Оставьте отзыв о скачивании PDF-файла.

Справочник по выражениям анализа данных (DAX)

Выражения анализа данных (DAX) — это библиотека функций и операторов, которые можно объединить для создания формул и выражений в Power BI, Analysis Services и PowerPivot в моделях данных Excel.

Функции

ССЫЛКА

функции DAX

операторы DAX

НОВОЕ

Новые функции DAX

Сведения о DAX

ОБЩИЕ СВЕДЕНИЯ

обзор DAX

VIDEO

Videos

ОБУЧЕНИЕ

Использование DAX в Power BI Desktop

Понимать

КОНЦЕПЦИЯ

глоссарий DAX

операторы DAX

запросы DAX

синтаксис DAX

Общие сведения о DAX

Статья • 20.10.2023

DAX — это язык формул, используемый в Analysis Services, Power BI и Power Pivot в Excel. Формулы DAX состоят из функций, операторов и значений и предназначены для выполнения сложных вычислений и запросов к данным в связанных таблицах и столбцах в табличных моделях данных.

Эта статья — лишь введение в наиболее важные понятия в DAX. Язык DAX в ней описывается в той форме, в которой он используется во всех соответствующих продуктах. В некоторых продуктах или сценариях некоторые функциональные возможности могут быть неприменимы. См. сведения о реализации DAX в конкретном продукте в документации по нему.

Вычисления

Формулы DAX используются в мерах, вычисляемых столбцах, вычисляемых таблицах и для обеспечения безопасности уровня строк.

Меры

Меры — это динамические формулы вычисления, результаты которых меняются в зависимости от контекста. Они применяются в отчетах, поддерживающих объединение и фильтрацию данных модели по нескольким атрибутам, например в отчетах Power BI и сводных таблицах и сводных диаграммах Excel. Меры создаются с помощью строки формул DAX в конструкторе моделей.

В формуле в мере могут использоваться стандартные статистические функции, автоматически создаваемые с помощью автосуммирования, например COUNT или SUM, либо вы можете определить собственную формулу с помощью строки формул DAX. Именованные меры можно передавать в качестве аргументов в другие меры.

При определении формулы для меры в строке формул в компоненте подсказки отображается предварительный просмотр будущих результатов во всем текущем контексте, но в других местах результаты не обновляются сразу же. Результаты вычисления (отфильтрованные) не отображаются немедленно потому, что результат меры нельзя определить без контекста. Для вычисления меры требуется клиентское приложение создания отчетов, которое может предоставить контекст, необходимый для получения данных для каждой из ячеек, а затем вычислить

выражения для ячеек. Таким клиентом может быть сводная таблица или сводная диаграмма Excel, отчет Power BI или табличное выражение в запросе DAX в SQL Server Management Studio (SSMS).

Независимо от клиента для каждой ячейки в результатах выполняется отдельный запрос. Это означает, что каждое сочетание заголовков строки и столбца в сводной таблице или каждый набор срезов и фильтров в отчете Power BI дает особое подмножество данных, на основе которых вычисляется мера. Например, рассмотрим следующую очень простую формулу для меры.

```
DAX

Total Sales = SUM([Sales Amount])
```

Когда пользователь помещает меру TotalSales в отчет, а затем помещает столбец "Категория продукта" из таблицы "Продукт" в окно "Фильтры", сумма продаж вычисляется и отображается для каждой категории продуктов.

В отличие от вычисляемых столбцов, синтаксис меры предусматривает размещение имени меры перед формулой. В приведенном примере перед формулой указано имя **Total Sales**. После создания меры ее имя и определение появляются в списке полей в клиентском приложении для создания отчетов. В зависимости от перспектив и ролей они доступны всем пользователям модели.

Дополнительные сведения см. на следующих ресурсах:
Меры в Power BI Desktop
Меры в Analysis Services
Меры в Power Pivot

Вычисляемые столбцы

Вычисляемый столбец — это столбец, добавляемый в существующую таблицу (в конструкторе моделей), для которого затем создается формула DAX, определяющая значения в столбце. Если вычисляемый столбец содержит допустимую формулу DAX, значения вычисляются для каждой строки сразу после ввода формулы. Затем значения сохраняются в модели данных в памяти. Например, в таблице дат в строке формул вводится следующая формула:

```
DAX

= [Calendar Year] & " Q" & [Calendar Quarter]
```

Для получения значения для каждой строки в таблице берется значение из столбца Calendar Year (Календарный год) в той же таблице дат, добавляется пробел и заглавная буква Q, а затем добавляется значение из столбца Calendar Quarter (Календарный квартал) в той же таблице дат. Результат для каждой строки в вычисляемом столбце вычисляется и отображается немедленно, например **2017 Q1**. Значения в столбце пересчитываются только в том случае, если обновляется таблица, в которой находится столбец, или любая связанная таблица либо если модель выгружается из памяти, а затем загружается снова, как, например, при закрытии и повторном открытии файла Power BI Desktop.

Дополнительные сведения:
Вычисляемые столбцы в Power BI Desktop
Вычисляемые столбцы в Analysis Services
Вычисляемые столбцы в Power Pivot

Вычисляемые таблицы

Вычисляемая таблица — это вычисляемый объект на основе выражения формулы, который формируется из всех других таблиц в той же модели или их части. Вместо запроса и загрузки значений в столбцы новой таблицы из источника данных значения таблицы определяются формулой DAX.

Вычисляемые таблицы могут быть полезны в ролевом измерении. Примером может служить использование таблицы дат в качестве таблицы дат заказов, дат отгрузки или дат выполнения в зависимости от связи по внешнему ключу. Создав вычисляемую таблицу для дат отгрузки явным образом, вы получаете отдельную таблицу с полной функциональностью, доступную для запросов. Вычисляемые таблицы также полезны при настройке отфильтрованного набора строк либо подмножества или надмножества столбцов из других существующих таблиц. Это позволяет создавать варианты таблицы для конкретных сценариев, сохраняя исходную таблицу неизменной.

Вычисляемые таблицы поддерживают связи с другими таблицами. Столбцы в вычисляемой таблице имеют типы данных и форматирование и могут относиться к категории данных. Вычисляемые таблицы можно именовать, предоставлять или скрывать так же, как любые другие таблицы. Вычисляемая таблица пересчитывается при изменении или обновлении данных в любой таблице, из которой она берет данные.

Дополнительные сведения:
Вычисляемые таблицы в Power BI Desktop
Вычисляемые таблицы в Analysis Services

Безопасность на уровне строк

При обеспечении безопасности на уровне строк формула DAX должна возвращать логическое значение true или false, которое определяет, какие строки могут возвращаться в результатах запроса, выполняемого членами той или иной роли. Например, рассмотрим таблицу Customers (Клиенты) со следующей формулой DAX для членов роли Sales (Продажи):

```
DAX
= Customers[Country] = "USA"
```

Члены роли Sales смогут просматривать данные только по клиентам в США. Результаты статистических вычислений, таких как SUM, также возвращаются только для клиентов в США. Безопасность на уровне строк недоступна в Power Pivot в Excel.

При определении безопасности на уровне строк с помощью формулы DAX создается разрешенный набор строк. При этом доступ к другим строкам не запрещается. Они просто не возвращаются в рамках разрешенного набора строк. Другие роли могут разрешать доступ к строкам, исключенным в формуле DAX. Если пользователь является членом другой роли и ее безопасность на уровне строк разрешает доступ к определенному набору строк, пользователь может просматривать данные из этих строк.

Формулы безопасности на уровне строк применяются к указанным строкам, а также к связанным с ними строкам. Если в таблице есть несколько связей, фильтры применяются к активной связи. Формулы безопасности на уровне строк будут пересекаться с другими формулами, определенными для связанных таблиц.

Дополнительные сведения:
Row-level security (RLS) with Power BI (Безопасность на уровне строк (RLS) в Power BI)
Роли в Analysis Services

Запросы

Запросы DAX можно создавать и выполнять в SQL Server Management Studio (SSMS) и в средствах с открытым кодом, таких как DAX Studio (daxstudio.org). В отличие от формул вычислений DAX, которые можно создавать только в табличных моделях данных, запросы DAX также можно выполнять к многомерным моделям Analysis Services. Запросы DAX зачастую проще в написании и эффективнее, чем запросы многомерных выражений (MDX).

Запрос DAX представляет собой инструкцию, наподобие инструкции SELECT в T-SQL. Простейший тип запроса DAX — инструкция *evaluate*. Например,

```
DAX

EVALUATE
 ( FILTER ( 'DimProduct', [SafetyStockLevel] < 200 ) )
ORDER BY [EnglishProductName] ASC
```

Этот запрос возвращает таблицу только с теми продуктами, у которых значение SafetyStockLevel меньше 200, отсортированными в порядке возрастания по полю EnglishProductName.

В рамках запроса можно создавать меры. Меры существуют, только пока выполняется запрос. Дополнительные сведения см. в статье Запросы DAX.

Формулы

Формулы DAX важны для создания вычислений в вычисляемых столбцах и мерах, а также для защиты данных с помощью безопасности на уровне строк. Для создания формул для вычисляемых столбцов и мер используется строка формул, расположенная вверху окна конструктора моделей или в редакторе DAX. Чтобы создать формулы для безопасности на уровне строк, используйте диспетчер ролей или диалоговое окно "Управление ролями". С помощью сведений в этом разделе вы получите представление об основных принципах формул DAX.

Основы формул

Формулы DAX могут быть как совсем простыми, так и довольно сложными. В таблице ниже приведен ряд примеров простых формул, которые можно использовать в вычисляемом столбце.

Формула	Определение
`= TODAY()`	Вставляет текущую дату в каждую строку вычисляемого столбца.
`= 3`	Вставляет значение 3 в каждую строку вычисляемого столбца.
`= [Column1] + [Column2]`	Складывает значения из столбцов [Column1] и [Column2] одной строки, а затем помещает результат в ту же строку в вычисляемом столбце.

Независимо от уровня сложности формулы для ее создания можно следовать приведенной ниже последовательности действий.

1. Каждая формула должна начинаться со знака равенства (=).
2. Можно ввести имя функции, выбрать его или ввести выражение.
3. Введите первые несколько букв имени, и функция автозаполнения выведет список доступных функций, таблиц и столбцов. Чтобы добавить элемент из списка автозаполнения в формулу, нажмите клавишу TAB.

   Чтобы отобразить список доступных функций, можно также нажать кнопку **Fx**. Чтобы выбрать функцию из раскрывающегося списка, используйте клавиши со стрелками для перехода по элементам и нажмите кнопку **OK**, чтобы добавить функцию в формулу.
4. Укажите аргументы функции, выбрав их в раскрывающемся списке возможных таблиц и столбцов или введя значения.
5. Проверьте наличие синтаксических ошибок: убедитесь в том, что все скобки закрыты, а столбцы, таблицы и значения указаны правильно.
6. Чтобы подтвердить ввод формулы, нажмите клавишу ВВОД.

> ⓘ **Примечание**
>
> Вычисляемый столбец заполняется значениями сразу после ввода формулы и ее проверки. Для меры нажатие клавиши ВВОД приводит к сохранению определения меры в таблице. Если формула недопустима, отображается ошибка.

Например, рассмотрим следующую формулу в мере с именем **Days in Current Quarter** (Дней в текущем квартале):

```DAX
Days in Current Quarter = COUNTROWS( DATESBETWEEN( 'Date'[Date],
STARTOFQUARTER( LASTDATE('Date'[Date])), ENDOFQUARTER('Date'[Date])))
```

Эта мера используется для определения соотношения между неполным текущим периодом и предыдущим периодом. Формула должна учитывать истекшую часть периода и сравнивать ее с той же частью предыдущего периода. В этом случае соотношение определяется по формуле [Дней с начала текущего квартала]/[Дней в текущем квартале].

Эта формула содержит следующие элементы:

Элемент формулы	Описание
`Days in Current Quarter`	Имя меры.
`=`	Формула начинается со знака равенства (=).
`COUNTROWS`	COUNTROWS подсчитывает число строк в таблице Date.
`()`	В круглых скобках указываются аргументы.
`DATESBETWEEN`	Функция DATESBETWEEN возвращает даты между первой и последней датами для каждого значения в столбце Date таблицы Date.
`'Date'`	Определяет таблицу Date. Имена таблиц заключаются в одинарные кавычки.
`[Date]`	Определяет столбец Date в таблице Date. Имена столбцов заключаются в квадратные кавычки.
`,`	
`STARTOFQUARTER`	Функция STARTOFQUARTER возвращает дату начала квартала.
`LASTDATE`	Функция LASTDATE возвращает последнюю дату квартала.
`'Date'`	Определяет таблицу Date.
`[Date]`	Определяет столбец Date в таблице Date.
`,`	
`ENDOFQUARTER`	Функция ENDOFQUARTER
`'Date'`	Определяет таблицу Date.
`[Date]`	Определяет столбец Date в таблице Date.

Использование автозаполнения формул

Автозаполнение помогает соблюдать правильный синтаксис формул, предлагая варианты для каждого из элементов в формуле.

- Функцию автозаполнения формул можно использовать в середине существующей формулы со вложенными функциями. Текст, расположенный непосредственно перед точкой вставки, используется для отображения значений раскрывающегося списка, а остальной текст остается без изменений.

- При использовании функции автозаполнения для функций закрывающая скобка не добавляется, а также не выполняется автоматическое добавление сходных скобок. Необходимо проверить синтаксическую правильность каждой функции, иначе ее будет нельзя сохранить и использовать.

Использование нескольких функций в формуле

Функции можно вкладывать, то есть использовать результаты одной функции в качестве аргумента другой функции. В вычисляемых столбцах поддерживается до 64 уровней вложенности функций. Однако вложенность функций может усложнить создание формул и диагностику ошибок. Многие функции предназначены для использования исключительно в качестве вложенных. Эти функции возвращают таблицу, которая не может быть непосредственно сохранена в качестве результата, но может быть передана табличной функции в качестве входного параметра. Например, в качестве первого аргумента функций SUMX, AVERAGEX и MINX требуется таблица.

Функции

Функция — это именованная формула в выражении. В большинстве функций в качестве входных данных используются обязательные и необязательные аргументы, также называемые параметрами. При выполнении функции возвращается значение. В DAX есть функции, с помощью которых можно производить вычисления с датами и временем, создавать условные значения, работать со строками, выполнять уточняющие запросы на основе связей и перебирать таблицы для рекурсивных вычислений. Многие из этих функций очень похожи на формулы Excel, однако формулы DAX отличаются в следующих важных аспектах.

- Функция DAX всегда ссылается на столбец или таблицу целиком. В формулу можно добавить фильтры, чтобы использовать только определенные значения из таблицы или столбца.

- Если нужно настроить вычисления на уровне строк, то в языке DAX имеются функции, использующие в зависимости от контекста либо текущее значение строки, либо связанное значение в качестве своего рода параметра. Чтобы понять, как работают эти функции, см. раздел Контекст этой статьи.

- В языке DAX имеется множество функций, возвращающих таблицу, а не значение. Таблица не отображается в клиенте отчетов, она служит для передачи данных другим функциям. Например, можно получить таблицу, а

затем подсчитать уникальные значения в ней или вычислить динамические суммы по отфильтрованным таблицам или столбцам.

- В число функций языка DAX входят и различные функции *логики операций со временем* . Они позволяют определять или выбирать диапазоны дат, а также выполнять динамические вычисления на основе этих дат или диапазонов. Например, можно сравнить суммы по параллельным периодам.

Агрегатные функции

Статистические функции вычисляют (скалярное) значение, например число, сумму, среднее, минимальное или максимальное значение для всех строк в столбце или таблице в соответствии с заданным выражением. Дополнительные сведения см. в статье Статистические функции.

Функции даты и времени

Функции даты и времени в DAX схожи с функциями даты и времени Microsoft Excel. Однако функции DAX основаны на типе данных **datetime** начиная с 1 марта 1900 г. Дополнительные сведения см. в разделе Функции даты и времени.

Функции фильтрации

Функции фильтра в DAX возвращают определенные типы данных, выполняют поиск значений в связанных таблицах и применяют фильтры по связанным значениям. Функции поиска работают с использованием таблиц и связей, как в базе данных. Функции фильтрации дают возможность управлять контекстом данных для создания динамических вычислений. Дополнительные сведения см. в разделе Функции фильтров.

Финансовые функции

Финансовые функции — эти функции используются в формулах, которые выполняют финансовые вычисления, такие как чистая приведенная стоимость и норма прибыли. Эти функции похожи на финансовые функции, используемые в Microsoft Excel. Дополнительные сведения см. в разделе Финансовые функции.

Информационные функции

Информационная функция проверяет ячейку или строку, указанные в качестве аргумента, и сообщает, соответствует ли значение ожидаемому типу. Например, функция ISERROR возвращает значение TRUE, если упоминаемое значение содержит ошибку. Дополнительные сведения см. в разделе Информационные функции.

Логические функции

Логические функции обрабатывают выражение и возвращают сведения о значениях в выражении. Например, функция TRUE позволяет определить, возвращает ли выражение значение TRUE. Дополнительные сведения см. в разделе Логические функции.

Математические и тригонометрические функции

Математические функции в DAX весьма схожи с математическими и тригонометрическими функциями Excel. Числовые типы данных, используемые в функциях DAX, имеют незначительные отличия. Дополнительные сведения см. в разделе Математические и тригонометрические функции.

Другие функции

Эти функции выполняют уникальные действия, и поэтому их нельзя отнести к другим категориям. Дополнительные сведения см. в разделе Другие функции.

Функции связей

Функции связей в DAX позволяют возвращать значения из другой связанной таблицы, указывать определенную связь для использования в выражении и указывать направление перекрестной фильтрации. Дополнительные сведения см. в статье Функции связей.

Статистические функции

Статистические функции вычисляют значения, связанные со статистическими распределениями и вероятностями, например стандартное отклонение и число перестановок. Дополнительные сведения см. в разделе Статистические функции.

Текстовые функции

Текстовые функции в DAX очень похожи на их аналоги в Excel. Можно вернуть часть строки, искать текст в строке или объединить строковые значения. DAX также предоставляет функции для управления форматами дат, времени и чисел. Дополнительные сведения см. в разделе Текстовые функции.

Функции операций со временем

Функции логики операций со временем в DAX позволяют выполнять вычисления с использованием встроенных наборов знаний о календарях и датах. Используя диапазоны времени и дат в сочетании с агрегатами или вычислениями, можно производить осмысленные сравнения объемов продаж, запасов и других показателей за сопоставимые периоды времени. Дополнительные сведения см. в разделе Функции операций со временем (DAX).

Функции обработки таблиц

Эти функции возвращают таблицу или обрабатывают существующие таблицы. Например, с помощью ADDCOLUMNS можно добавить вычисляемые столбцы в указанную таблицу либо вернуть сводную таблицу по набору групп с помощью функции SUMMARIZECOLUMNS. Дополнительные сведения см. в разделе Функции обработки таблиц.

Переменные

Переменные в выражении можно создавать с помощью функции VAR. VAR технически не является функцией. Это ключевое слово, применяемое для сохранения результата выражения в именованной переменной. Затем переменную можно передавать в качестве аргумента в другие выражения мер. Пример:

DAX

```
VAR
    TotalQty = SUM ( Sales[Quantity] )

Return

    IF (
        TotalQty > 1000,
        TotalQty * 0.95,
        TotalQty * 1.25
        )
```

В этом примере TotalQty можно передать в другие выражения как именованную переменную. Переменные могут быть любого скалярного типа данных, включая таблицы. Переменные значительно расширяют возможности формул DAX.

Типы данных

Данные в модель можно импортировать из множества различных источников данных, которые могут поддерживать различные типы данных. При импорте данных в модель данные преобразуются в один из типов данных табличной модели. При использовании данных модели в вычислениях тип данных меняется на DAX на время проведения вычислений и выведения результата. При создании формулы DAX термы, используемые в формуле, автоматически определяют тип возвращаемого значения данных.

DAX поддерживает следующие типы данных:

Тип данных в модели	**Тип данных в DAX**	**Описание**
Whole Number	64-разрядное (8-байтовое) целочисленное значение [1, 2]	Числа без десятичных разрядов. Целые числа могут быть положительными или отрицательными, но не могут содержать дробную часть в диапазоне -9,223,372,036,854,775,808 (-2^63) и 9,223,372,036,854,775,807 (2^63-1).
Десятичное число	64-разрядное (8 байтовое) вещественное число [1, 2]	Вещественные числа — это числа, которые могут иметь знаки после запятой. Вещественные числа включают широкий диапазон значений. Отрицательные числа от -1.79E +308 до -2.23E -308 Нуль Положительные числа от 2.23E -308 до 1.79E + 308 Однако количество значащих цифр ограничено 17 знаками после запятой.
Логический	Логический	Значение True или False.
Текстовый	Строковый	Строка символьных данных в Юникоде. Могут быть строками, числами или датами, представленными в текстовом формате.
Дата	Дата и время	Значения даты и времени в принятом представлении даты-времени.

Тип данных в модели	Тип данных в DAX	Описание
		Допустимый диапазон дат включает значения после 1 марта 1900г.
Валюта	Валюта	Тип данных "Валюта" включает значения в диапазоне от -922,337,203,685,477.5808 до 922,337,203,685,477.5807 с четырьмя десятичными знаками заданной точности.
Недоступно	Пусто	Тип данных с пустыми значениями в DAX представляет и заменяет пустые значения NULL в SQL. Пустое значение создается с помощью функции BLANK, а проверяется с помощью логической функции ISBLANK.

В табличных моделях данных в качестве входных или выходных данных для многих функций DAX также используется тип данных *Table*. Например, функция FILTER принимает в качестве входного аргумента таблицу и возвращает другую таблицу, которая содержит только строки, удовлетворяющие условиям фильтра. Применение табличных функций в сочетании с агрегатными функциями позволяет выполнять сложные вычисления с динамически определяемыми наборами данных.

Хотя типы данных обычно устанавливаются автоматически, важно понимать, как они работают, в особенности в формулах DAX. Например, ошибки в формулах или непредвиденных результаты часто связаны с использованием определенного оператора, который недопустим для указанного в аргументе типа данных. Например, формула `= 1 & 2` возвращает строковое значение 12. В то же время формула `= "1" + "2"` возвращает целочисленный результат 3.

Контекст

Контекст является важным понятием при создании формул DAX. Контекст позволяет выполнять динамический анализ, при котором результаты формулы могут изменяться в зависимости от выделенной строки или ячейки и любых взаимосвязанных данных. Понимание и эффективное использование контекста важно для построения высокопроизводительного динамического анализа и для устранения неполадок в формулах.

Формулы в табличных моделях могут вычисляться в разном контексте в зависимости от других структурных элементов, например на них влияет следующее:

- Фильтры, примененные в сводной таблице или отчете

- Фильтры, определенные в формуле
- Отношения, указанные с помощью специальных функций в формуле

Существуют различные типы контекста: *контекст строки*, *контекст запроса*и *контекст фильтра*.

Контекст строки

Контекст строки можно представить как "текущую строку". Если формула создана в вычисляемом столбце, то контекст строки для этой формулы включает в себя значения всех столбцов в текущей строке. Если таблица связана с другой таблицей, содержимое также включает в себя все значения из другой таблицы, связанные с текущей строкой.

Предположим, создается вычисляемый столбец `= [Freight] + [Tax]`, который складывает значения из двух столбцов Freight и Tax одной таблицы. Эта формула автоматически возвращает только значения из текущей строки в указанных столбцах.

Контекст строки также учитывает любые связи, определенные между таблицами, в том числе связи, заданные в пределах вычисляемого столбца при помощи формул DAX, чтобы определить, какие строки в связанных таблицах связаны с текущим рядом.

Например, в следующей формуле функция RELATED используется для выборки значения суммы налогов из связанной таблицы в зависимости от региона, в который отправлен заказ. Значение суммы налогов определяется с использованием значения для региона из текущей таблицы путем поиска этого региона в связанной таблице и получения ставки налога для этого региона из связанной таблицы.

```
DAX
```

```
= [Freight] + RELATED('Region'[TaxRate])
```

Эта формула получает налоговую ставку в текущем регионе из таблицы Region и складывает ее со значением столбца Freight. В формулах DAX не обязательно знать или задавать особые связи, соединяющие таблицы.

Контекст нескольких строк

Язык DAX включает функции, которые повторяют вычисления над таблицей. Эти функции могут содержать несколько текущих строк, каждая из которых может обладать собственным контекстом строки. В сущности, эти функции позволяют создавать формулы, выполняющие рекурсию по внутреннему или внешнему циклу.

Предположим, модель содержит таблицы **Products** и **Sales** . Пользователю может потребоваться просмотреть всю таблицу продаж, содержащую транзакции с множеством продуктов, и найти самый крупный заказ по каждому из продуктов в одной отдельной транзакции.

С помощью DAX можно создать одну формулу, возвращающую необходимое значение, причем ее результаты будут автоматически обновляться при каждом добавлении пользователем данных в таблицы.

DAX

```
= MAXX(FILTER(Sales,[ProdKey] = EARLIER([ProdKey])),Sales[OrderQty])
```

Подробный пример см. в описании функции EARLIER.

Иначе говоря, функция EARLIER сохраняет контекст строки из операции, предшествующей текущей. Функция всегда хранит в памяти два набора контекстов: один набор контекста представляет текущую строку для внутреннего цикла формулы, а другой набор контекста представляет текущую строку для внешнего цикла формулы. DAX автоматически формирует поток значений между двумя циклами, чтобы можно было создавать сложные агрегаты.

Контекст запроса

Контекст запроса представляет подмножество данных, которое неявным образом возвращается для формулы. Например, когда пользователь помещает в отчет меру или поле, обработчик анализирует заголовки строк и столбцов, срезы и фильтры отчетов для определения контекста. Затем выполняются необходимые запросы к модели данных для получения правильного подмножества данных, выполнения вычислений, определенных формулой, и последующего заполнения значений в отчете.

Так как контекст меняется в зависимости от того, где находится формула, ее результаты также могут меняться. Например, предположим, что вы создаете формулу, которая суммирует значения в столбце **Profit** (Прибыль) таблицы **Sales** (Продажи): `= SUM('Sales'[Profit])`. Если эта формула используется в вычисляемом столбце в таблице **Sales**, ее результаты для всей таблицы будут такими же, так как

контекстом запроса для формулы всегда является весь набор данных таблицы **Sales**. Результатом будет прибыль по всем регионам, всем продуктам, всем годам и т. д.

Однако обычно пользователям не нужны одни и те же результаты, повторяющиеся сотни раз. Вместо этого они хотят знать прибыль за определенный год, в определенной стране, по определенному продукту или их сочетанию, а затем получить общий итог.

В отчете контекст изменяется путем фильтрации, добавления или удаления полей и использования срезов. Для каждого изменения — контекст запроса, в котором вычисляется мера. Поэтому та же формула при использовании в мере вычисляется в различных *контекстах запроса* для каждой ячейки.

Контекст фильтра

Контекст фильтра — это набор значений, допустимых для каждого столбца или для конкретных значений, извлекаемых из связанной таблицы. Фильтры могут применяться к столбцу в конструкторе или на уровне представления (отчеты и сводные таблицы). Фильтры также можно определить явным образом с помощью критериев фильтров в формуле.

Контекст фильтра добавляется при задании ограничений фильтра для набора значений, допустимых в столбце или таблице, с помощью аргументов в формуле. Контекст фильтра применяется поверх других контекстов, например контекста строки или контекста запроса.

В табличных моделях существует много способов создания контекста фильтра. В клиентах, использующих модель, например отчетах Power BI, пользователи могут создавать фильтры в режиме реального времени, добавляя срезы или фильтры отчета к заголовкам строк и столбцов. Также можно указать критерии фильтров прямо в формуле, задав связанные значения для фильтрации таблиц, используемых в качестве входных, либо для динамического получения контекста для значений, используемых в вычислениях. Можно также полностью или выборочно очищать фильтры для определенных столбцов. Это очень полезная возможность при создании формул для вычисления итогов.

Дополнительные сведения о создании фильтров в формулах см. в разделе Функция FILTER (DAX).
Пример очистки фильтров для получения итоговых сумм см. в разделе Функция ALL (DAX).

Примеры выборочного снятия и применения фильтров в формулах см. в описании функции ALLEXCEPT.

Определение контекста в формулах

При создании формулы DAX сначала проверяется допустимость ее синтаксиса, после чего проверяется возможность найти имена столбцов и таблиц из формулы в текущем контексте. Если не удается найти какие-либо указанные в формуле столбец или таблицу, то возвращается ошибка.

Контекст во время проверки (а также во время операций повторного вычисления) определяется так, как описано в предыдущих разделах, с учетом доступных таблиц в модели, любых связей между таблицами и применяемых фильтров.

Например, если данные импортированы в новую таблицу и не связаны с любыми другими таблицами, но фильтры еще не применены, то в *текущий контекст* будет входить весь набор столбцов в таблице. Если между этой и другими таблицами существуют связи, то текущий контекст будет включать связанные таблицы. При добавлении столбца из таблицы в отчет, где есть срезы и, возможно, некоторые фильтры отчетов, контекст формулы представляет собой подмножество данных в каждой ячейке отчета.

Контекст — это мощная концепция, которая может также усложнять поиск и исправление ошибок в формулах. Рекомендуется начать работу с простых формул и связей, чтобы понять, как работает контекст. В следующем разделе приведены несколько примеров использования формул в различных типах контекстов для возвращения динамических результатов.

Операторы

В языке DAX используются четыре различных типа операторов вычислений в формулах:

- Операторы сравнения, которые сравнивают значения и возвращают логические значения (TRUE/FALSE).
- Арифметические операторы, которые выполняют арифметические вычисления и возвращают числовые значения.
- Операторы объединения текста, которые соединяют две и более текстовые строки.
- Логические операторы, которые объединяют два и более выражения, возвращая один результат.

Подробные сведения об операторах, используемых в формулах DAX, см. в статье Операторы DAX.

Работа с таблицами и столбцами

Таблицы в табличных моделях данных похожи на таблицы Excel, но отличаются способом обработки данных и формул.

- Формулы работают только с таблицами и столбцами, а не с отдельными ячейками, ссылками на диапазоны и массивами.
- В формулах можно использовать связи для получения значений из связанных таблиц. Возвращаемые значения всегда связаны со значением в текущей строке.
- Нельзя иметь неупорядоченные или неоднородные данные, как это возможно на листе Excel. Каждая строка в таблице должна содержать одинаковое количество столбцов. Однако некоторые столбцы могут иметь пустые значения. Таблицы данных Excel и таблицы данных табличной модели не являются взаимозаменяемыми.
- Поскольку тип данных задается для каждого столбца, все значения в столбце должны иметь один тип.

Ссылки на таблицы и столбцы в формулах

На любую таблицу и любой столбец можно ссылаться по имени. Например, следующая формула показывает, как ссылаться на столбцы из двух таблиц по *полному* имени:

DAX

```
= SUM('New Sales'[Amount]) + SUM('Past Sales'[Amount])
```

При вычислении формулы конструктор моделей сначала проверяет общий синтаксис, а затем сравнивает указанные имена столбцов и таблиц с возможными столбцами и таблицами в текущем контексте. Если имя, использованное в формуле, определено неоднозначно либо столбец или таблица не найдены, будет выдана ошибка (строка #ERROR вместо значения данных в ячейках, в которых произошла ошибка). Дополнительные сведения о требованиях к именованию таблиц, столбцов и других объектов см. в разделе "Требования к именованию" статьи Синтаксис DAX.

Связи между таблицами

Создавая связи между таблицами, вы получаете возможность использования связанных значений в других таблицах в вычислениях. Например, с помощью вычисляемого столбца можно найти все записи об отгрузке, связанные с текущим торговым посредником, а затем суммировать затраты на доставку. Однако во многих случаях связь не является обязательной. Используя функцию LOOKUPVALUE в формуле, можно получить значение *result_columnName* для строки, которая удовлетворяет условиям, указанным в аргументах *search_column* и *search_value*.

Многие функции DAX требуют наличия связи между двумя или несколькими таблицами, чтобы найти столбцы, на которые сделана ссылка, и возвратить осмысленные результаты. Некоторые функции пытаются определить такую связь, но для получения наилучших результатов нужно всегда создавать связь, если это возможно. Табличные модели данных поддерживают несколько связей между таблицами. Во избежание путаницы или неверных результатов только одна связь является активной в каждый момент времени. Однако активную связь можно менять по мере необходимости для использования различных подключений в вычислениях. С помощью функции USERELATIONSHIP можно указать одну или несколько связей для конкретного вычисления.

При использовании связей важно соблюдать приведенные ниже правила проектирования формул.

- Если таблицы соединены связью, в ключевых столбцах должны быть одинаковые значения. Целостность данных не обеспечивается принудительно, поэтому создать связь можно, даже если значения в ключевом столбце отличаются. В этом случае следует помнить, что наличие пустых или несоответствующих значений может повлиять на результаты вычисления формул.

- При соединении таблиц в модели с помощью связей увеличивается размер *контекста*, в котором вычисляются формулы. Изменения контекста, являющиеся итогом добавления новых таблиц, новых связей или смены активной связи, могут привести к совершенно непредвиденным изменениям результатов. Дополнительные сведения см. в разделе Контекст данной статьи.

Обработка и обновление

Обработка и *пересчет* — это две отдельные операции, которые однако связаны друг с другом. Ими необходимо уметь пользоваться при создании модели со

сложными формулами, большим объемом данных или данными из внешних источников данных.

Обработка (обновление) — это обновление данных в модели данными из внешнего источника.

Повторным вычислением называется процесс обновления результатов формул для отражения изменений в самих формулах, а также изменений в базовых данных. Повторное вычисление влияет на общую производительность в следующих случаях:

- Значения в вычисляемом столбце вычисляются и хранятся в модели. Чтобы обновить значения в вычисляемом столбце, необходимо обработать модель одной из трех команд обработки: полная обработка, обработка данных и повторное вычисление. При любом изменении формулы ее результат должен всегда повторно вычисляться для всего столбца.

- Значения, вычисляемые с помощью мер, динамически рассчитываются каждый раз, когда пользователь добавляет меру в сводную таблицу или открывает отчет. Когда пользователь изменяет контекст, значения, возвращаемые мерой, меняются. Результаты меры всегда отражают последнее состояние кэша в памяти.

Обработка и пересчет не влияют на формулы безопасности на уровне строк, если только в результате пересчета не возвращается другое значение, что делает строку доступной или не доступной для запроса членами роли.

Обновления

DAX постоянно улучшается. Новые и обновленные функции выпускаются в составе следующего доступного обновления, которое обычно предоставляется ежемесячно. Сначала обновляются службы, а затем устанавливаются такие приложения, как Power BI Desktop, Excel, SQL Server Management Studio (SSMS) и расширение проекта Analysis Services для Visual Studio (SSDT). Обновления для служб SQL Server Analysis Services содержатся в следующем накопительном обновлении. Новые функции впервые объявляются и описываются в ссылке на функцию DAX, совпадающую с обновлениями Power BI Desktop.

Не все функции поддерживаются в более ранних версиях SQL Server Analysis Services и Excel.

Устранение неполадок

Если во время определения формулы выводится ошибка, значит формула может содержать *синтаксическую ошибку*, *семантическую ошибку*или *ошибку вычисления*.

Синтаксические ошибки устранять проще всего. Они обычно вызваны пропущенной скобкой или запятой.

Ошибки другого типа возникают, когда синтаксис правилен, но значение или столбец, на который указывает ссылка, не имеют смысла в контексте формулы. Семантические ошибки и ошибки вычисления могут вызываться следующими причинами.

- Формула ссылается на несуществующий столбец, таблицу или функцию.
- Формула выглядит правильно, но когда подсистема обработки данных извлекает данные, она обнаруживает несоответствие типов и выдает ошибку.
- Формула передает в функцию неправильное количество аргументов или аргументы неправильных типов.
- Формула ссылается на другой столбец, который содержит ошибку, поэтому ее значения недопустимы.
- Формула обращается к необработанному столбцу, в котором есть метаданные, однако отсутствуют данные, пригодные для использования в вычислениях.

В первых четырех случаях DAX помечает весь столбец, содержащий недопустимую формулу. В последнем случае DAX выделяет имя столбца серым цветом, чтобы показать, что он находится в необработанном состоянии.

Приложения и средства

Power BI Desktop



Power BI Desktop — это бесплатное приложение для моделирования данных и создания отчетов. Конструктор моделей включает в себя редактор DAX для создания формул вычисления DAX.

Power Pivot в Excel

Конструктор моделей Power Pivot в Excel включает в себя редактор DAX для создания формул вычисления DAX.

Visual Studio

Visual Studio с расширением проектов Analysis Services (VSIX) используется для создания проектов модели Analysis Services. Конструктор табличных моделей, установленный вместе с расширением проектов, включает редактор DAX.

SQL Server Management Studio.

SQL Server Management Studio (SSMS) — это важный инструмент для работы с Analysis Services. SSMS включает в себя редактор запросов DAX для выполнения запросов как к табличным, так и к многомерным моделям.

DAX Studio

DAX Studio — это клиентское средство с открытым кодом для создания и выполнения запросов DAX к моделям Analysis Services, Power BI Desktop и Power Pivot в Excel.

Tabular Editor.

Табличный редактор — это средство с открытым кодом, которое обеспечивает интуитивно понятное иерархическое представление каждого объекта в метаданных табличной модели. Табличный редактор включает редактор DAX с выделением синтаксиса, который предоставляет простой способ изменения мер, вычисляемых столбцов и вычисляемых выражений таблиц.

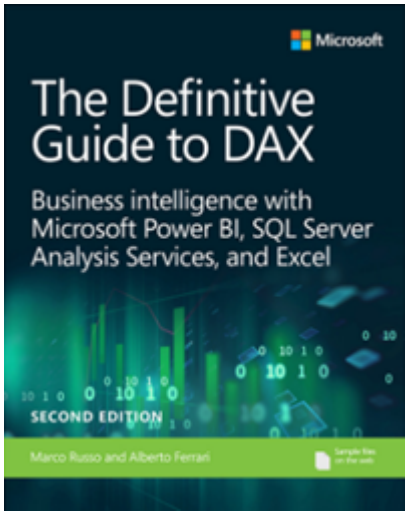
Учебные материалы

При изучении DAX лучше всего использовать приложение, в котором вы будете создавать свои модели данных. Для Analysis Services, Power BI Desktop и Power Pivot в Excel предлагаются статьи и учебники с уроками по созданию мер, вычисляемых столбцов и фильтров строк с помощью DAX. Ниже приведены некоторые дополнительные ресурсы.

Видео

Схема обучения Использование DAX в Power BI Desktop

Definitive Guide to DAX (Полное справочное руководство по DAX), Альберто Феррари (Alberto Ferrari) и Марко Руссо (Marco Russo), Microsoft Press. Доступное сейчас во втором выпуске, это подробное пошаговое руководство содержит как базовые сведения, так и более изощренные приемы для начинающих специалистов по моделированию данных и бизнес-аналитике.



Сообщество

В сфере DAX действует активное сообщество, члены которого всегда готовы поделиться своим опытом. В Сообществе Power BI (Майкрософт) есть специальный форум, посвященный командам и советам по DAX.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Videos

Статья • 21.04.2025

Независимо от того, используете ли вы Power BI Desktop, Power Pivot в Excel или Analysis Services, обучение выражений анализа данных (DAX) важно для создания эффективных моделей данных. Ниже приведены некоторые видеоролики, которые помогут вам приступить к использованию этого мощного языка выражений.

DAX 101

В этом видео DAX 101 партнер Microsoft Альберто Феррари представляет основные понятия в DAX. В практических и четких примерах вы узнаете о мерах, вычисляемых столбцах и основных выражениях моделирования данных с помощью DAX.
https://www.youtube-nocookie.com/embed/klQAZLr5vxA

Продвинутые DAX

В этом расширенном DAX видео партнер Microsoft, Альберто Феррари описывает DAX теорию, фильтр и контекст строк, и другие основные понятия в DAX.
https://www.youtube-nocookie.com/embed/6ncHnWMEdic

Практическое DAX для Power BI

В этом видео руководитель программы Microsoft Фил Seamark описывает некоторые рекомендации по практической работе с DAX в Power BI.
https://www.youtube-nocookie.com/embed/1fGfqzS37qs

Написание запросов DAX с помощью Copilot

Статья • 23.01.2025

Вы можете использовать Copilot для написания и объяснения запросов выражений анализа данных (DAX) в режиме запросов DAX для семантической модели Power BI.

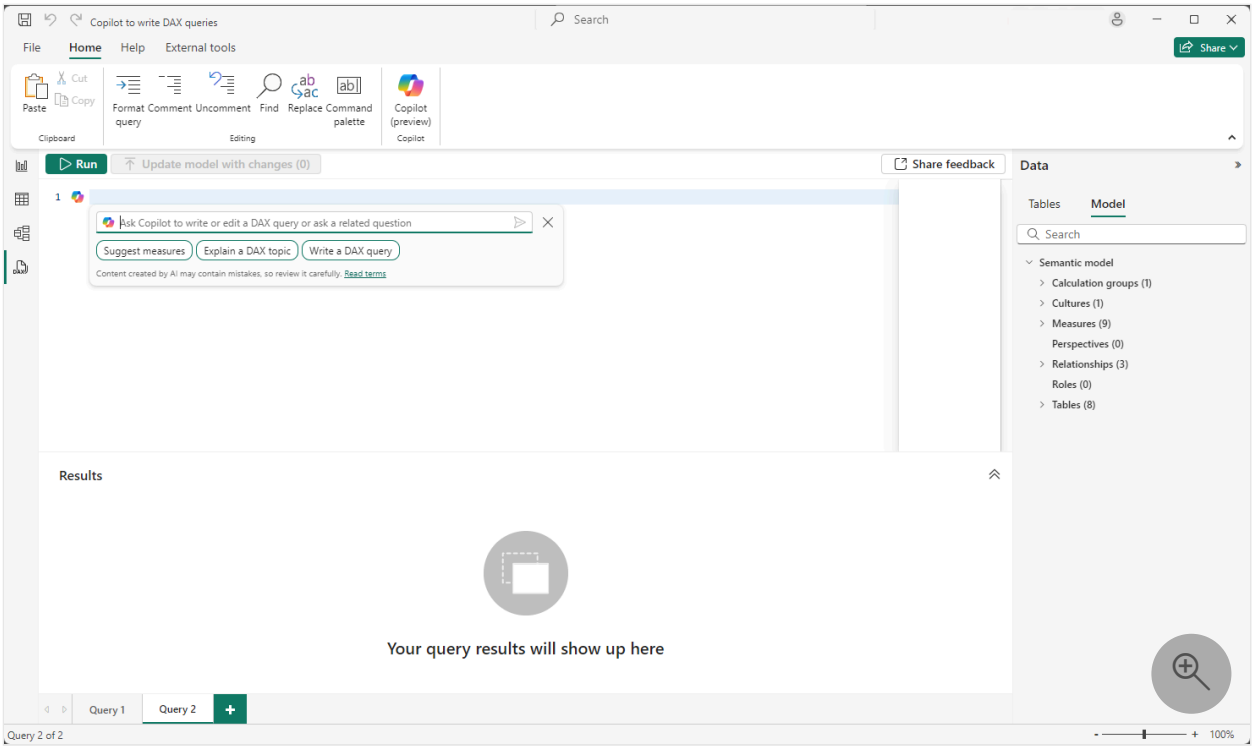
Включение Copilot

Проверьте, чтобы убедиться, что вы соответствуете требованиям Fabric Copilot.

Power BI Desktop

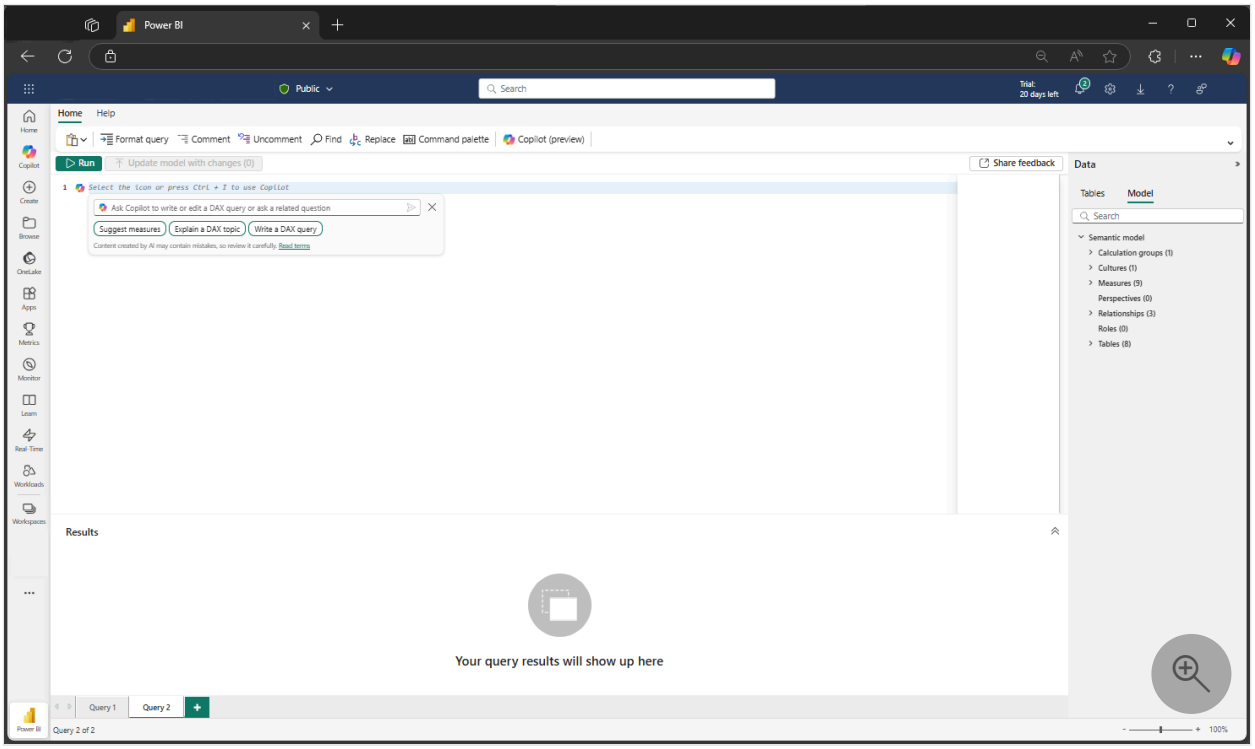
В Power BI Desktop в разделе предварительных версий функцийпараметров включите представление запросов DAX с помощью Copilot.

1. Откройте Power BI Desktop.
2. Откройте существующий отчет Power BI или подключитесь к опубликованной семантической модели в режиме редактирования в реальном времени или в режиме передачи данных в реальном времени. Кроме того, можно добавить данные и создать новую семантику модели и отчета.
3. Перейдите к **представлению запросов DAX**, это четвертое представление, доступное слева в Power BI Desktop.
4. Создайте вкладку запроса.
5. Откройте Copilot, нажав кнопку **Copilot** или клавиши `CTRL+I`.



служба или рабочая область Power BI

1. Выберите опубликованную семантическую модель в рабочей области.
2. В контекстном меню выберите **Написать запросы DAX**. Если записи запросов DAX неактивно, возможно, потребуется включить, чтобы пользователи могли изменять модели данных в службе PowerBI (предварительная версия) в разделе Общие параметров рабочей области.
3. Создайте вкладку запроса.
4. Откройте Copilot, нажав кнопку Copilot или используя `CTRL+ I`.



> ⓘ **Примечание**
>
> Авторы моделей DirectQuery и Direct Lake также могут использовать представление запросов DAX. Просматривайте данные в таблицах всякий раз, когда вы хотите.

Использование Copilot

Давайте рассмотрим, чем Copilot может вам помочь в режиме запросов DAX:

- **Естественный язык для запроса** DAX. Просто попросите у семантической модели, что вам нужно, и запрос DAX будет создан Copilot. Вы также можете выбрать существующий запрос DAX и поручить Copilot изменить его.
- **Запрос DAX к естественному языку**. Попросите Copilot объяснить, что делает любой из запросов DAX.
- **Общие сведения о концепциях** DAX. Может быть, вы слышали о определенной функции DAX или должны найти правильную функцию DAX для задания. Копилот также может помочь вам здесь.

Чтобы использовать Copilot для создания запроса DAX, введите подсказку, описывающую, какой запрос DAX вы хотите сгенерировать, и выберите **отправить** или нажмите клавишу ВВОД. Или можно нажать кнопку **Написать запрос DAX** пример. Чтобы запустить то, что возвращено, выберите **Выполнить** или нажмите F5, чтобы просмотреть результаты запроса DAX. Чтобы сохранить запрос и завершить работу с Copilot, выберите **Сохранить запрос**, чтобы добавить его на вкладку с запросами.

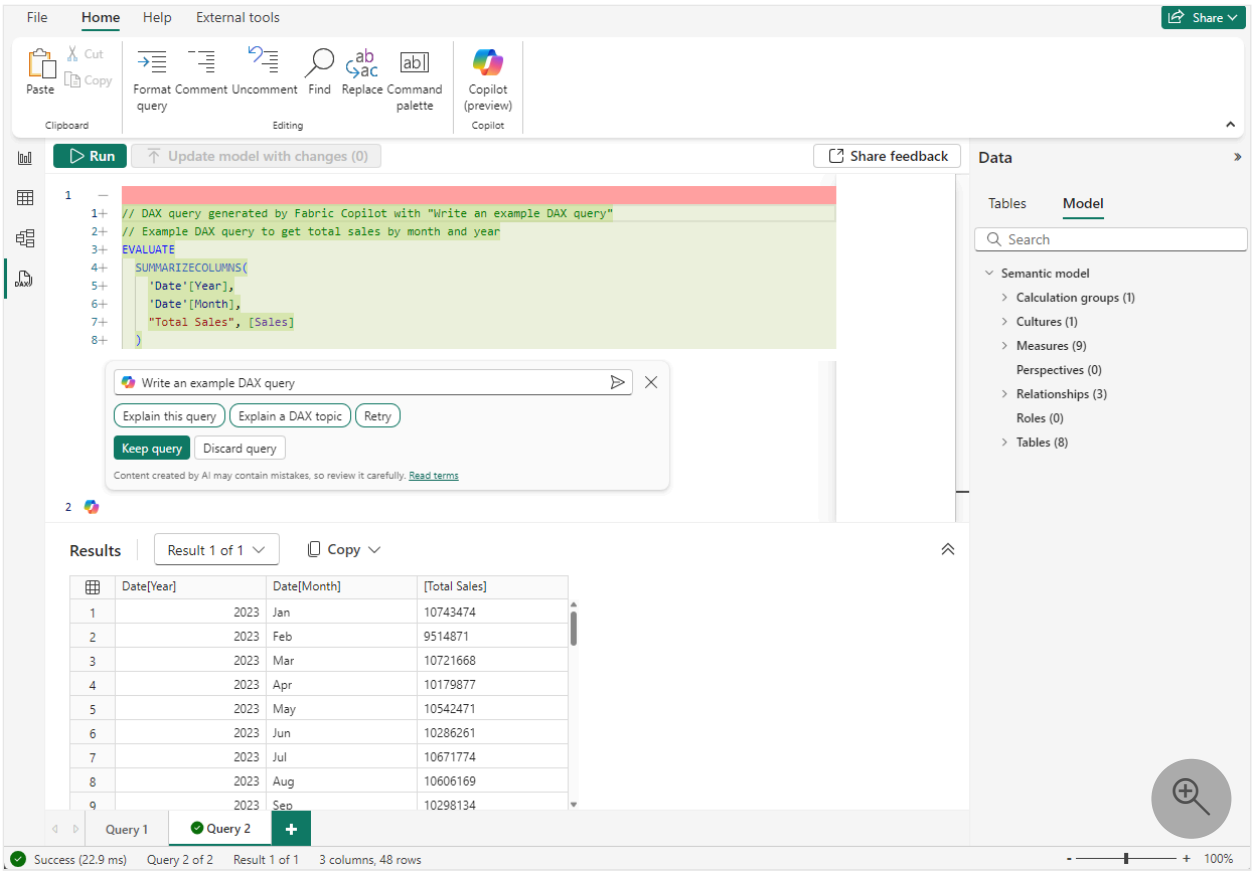
Чтобы использовать Copilot для объяснения запроса DAX, выберите запрос DAX и нажмите кнопку **Объяснить этот запрос**. Показано краткое объяснение с возможностью **показать больше** для просмотра более длинного объяснения.

Чтобы объяснить тему DAX с помощью Copilot, нажмите кнопку **Объяснить тему DAX** вдохновить или введите запрос для объяснения функции или темы DAX, затем выберите **Отправить** или нажмите клавишу ВВОД. Показано краткое объяснение с возможностью **показать больше** для просмотра более длинного объяснения.

Особенности интегрированного Copilot

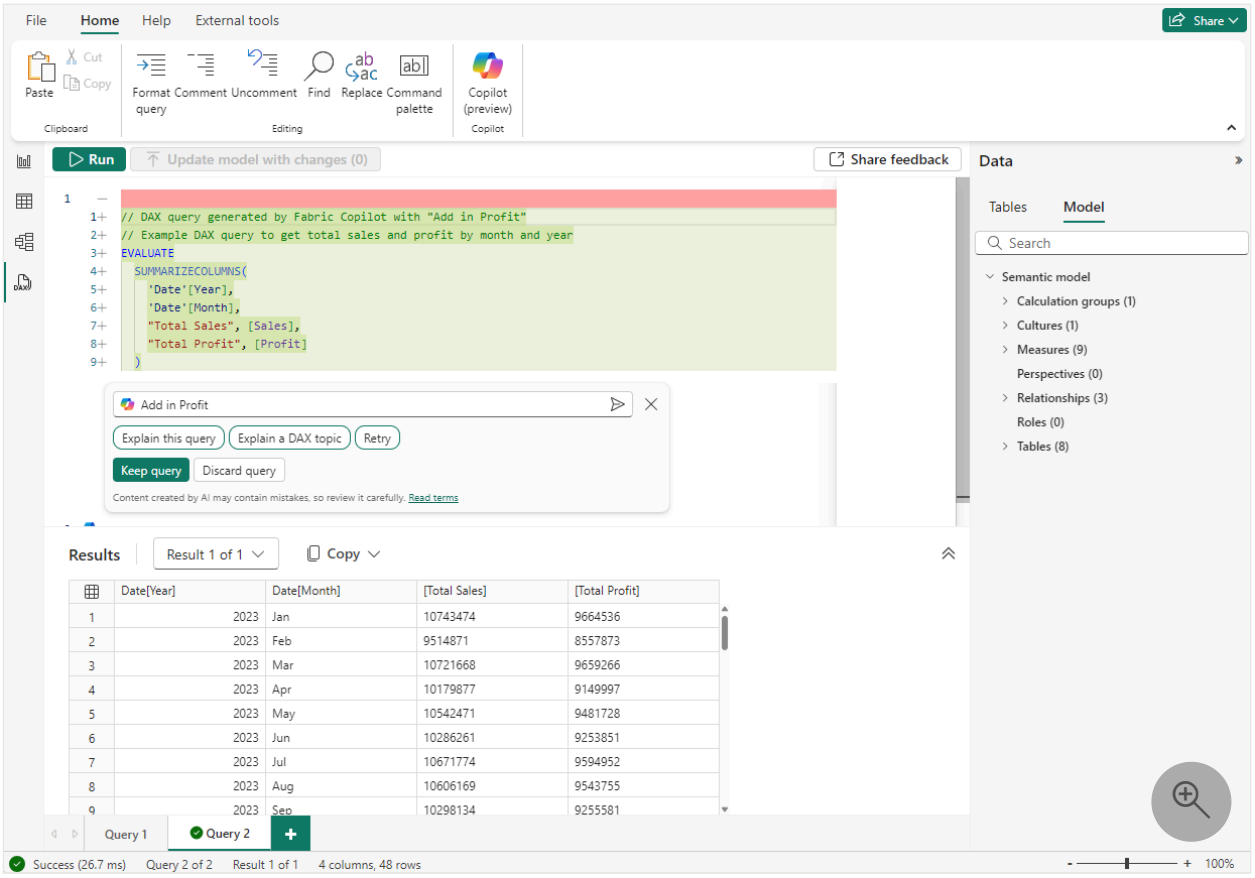
1. Выполните запрос DAX перед его сохранением

Вы можете **выполнить** запрос DAX, а затем решить, **сохранить** или **отклонить** запрос DAX.



2. Создание запроса DAX в диалоговом формате

После создания первоначального запроса DAX вы можете просто настроить его, вводя дополнительные подсказки пользователей.



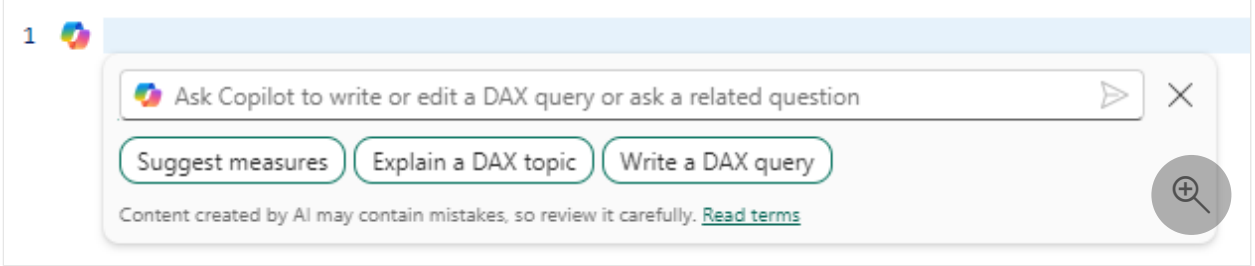
3. Проверки синтаксиса для созданного запроса DAX

Проверяется синтаксис созданного запроса DAX, и предупреждение автоматически повторяется один раз. Если повторная попытка также недопустима, созданный запрос DAX возвращается с примечанием о том, что возникла проблема, предоставив вам возможность перезаверять запрос или исправить созданный запрос DAX:

"Этот запрос содержит ошибки. Переформулируйте ваш запрос и попробуйте снова или попытайтесь исправить его самостоятельно.

4. Мотивирующие кнопки для начала работы с Copilot

Введите подсказку или выберите любую из этих кнопок, чтобы быстро узнать, что можно сделать в Copilot.



Примеры запросов

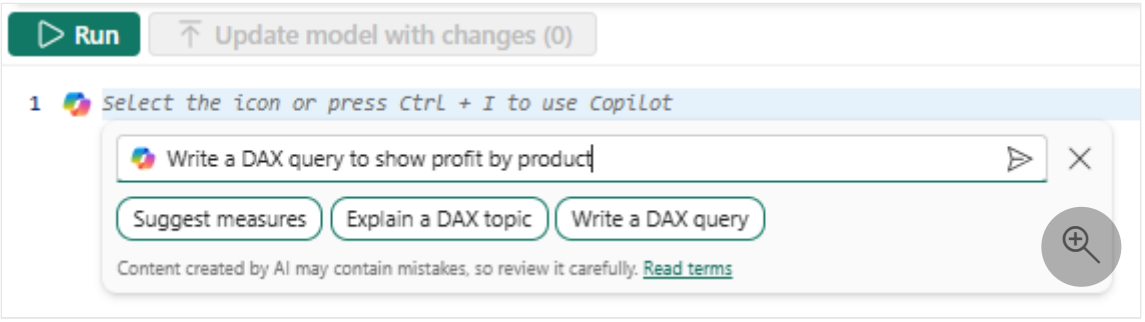
Эти образцы запросов могут проиллюстрировать использование Copilot для написания запросов DAX.

- напишите запрос DAX
- Добавление столбца с представлением диффа
- Объясните запрос DAX
- объяснить тему DAX

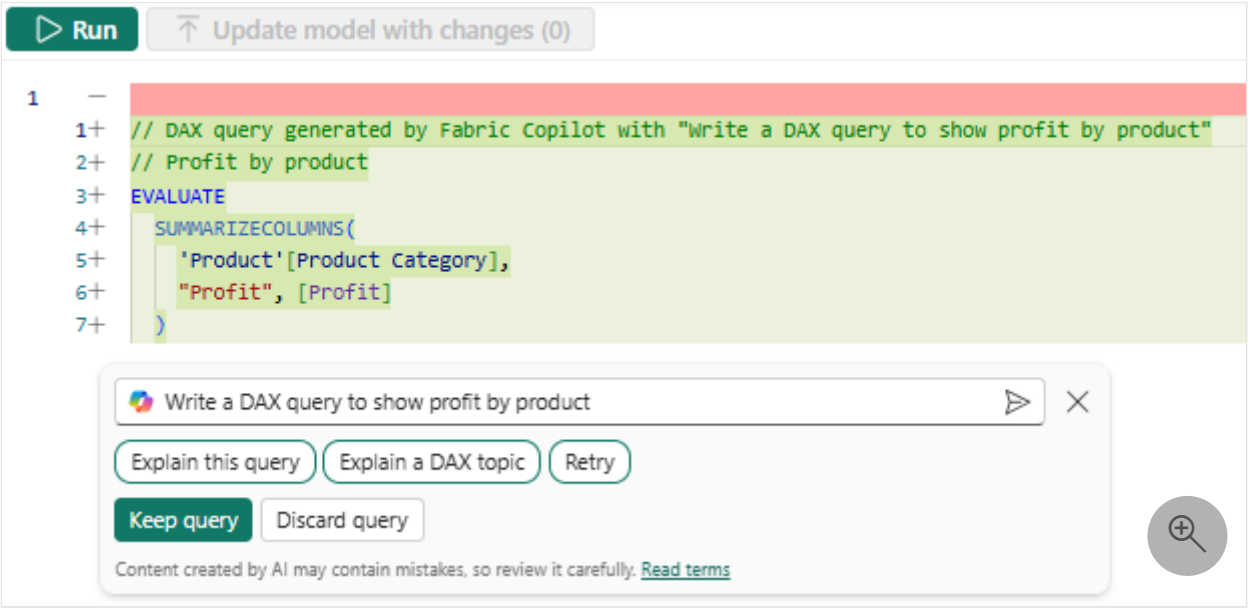
Написание запроса DAX

На пустой вкладке запроса вызовите Copilot с помощью `клавиш CTRL+I` и введите следующий запрос:

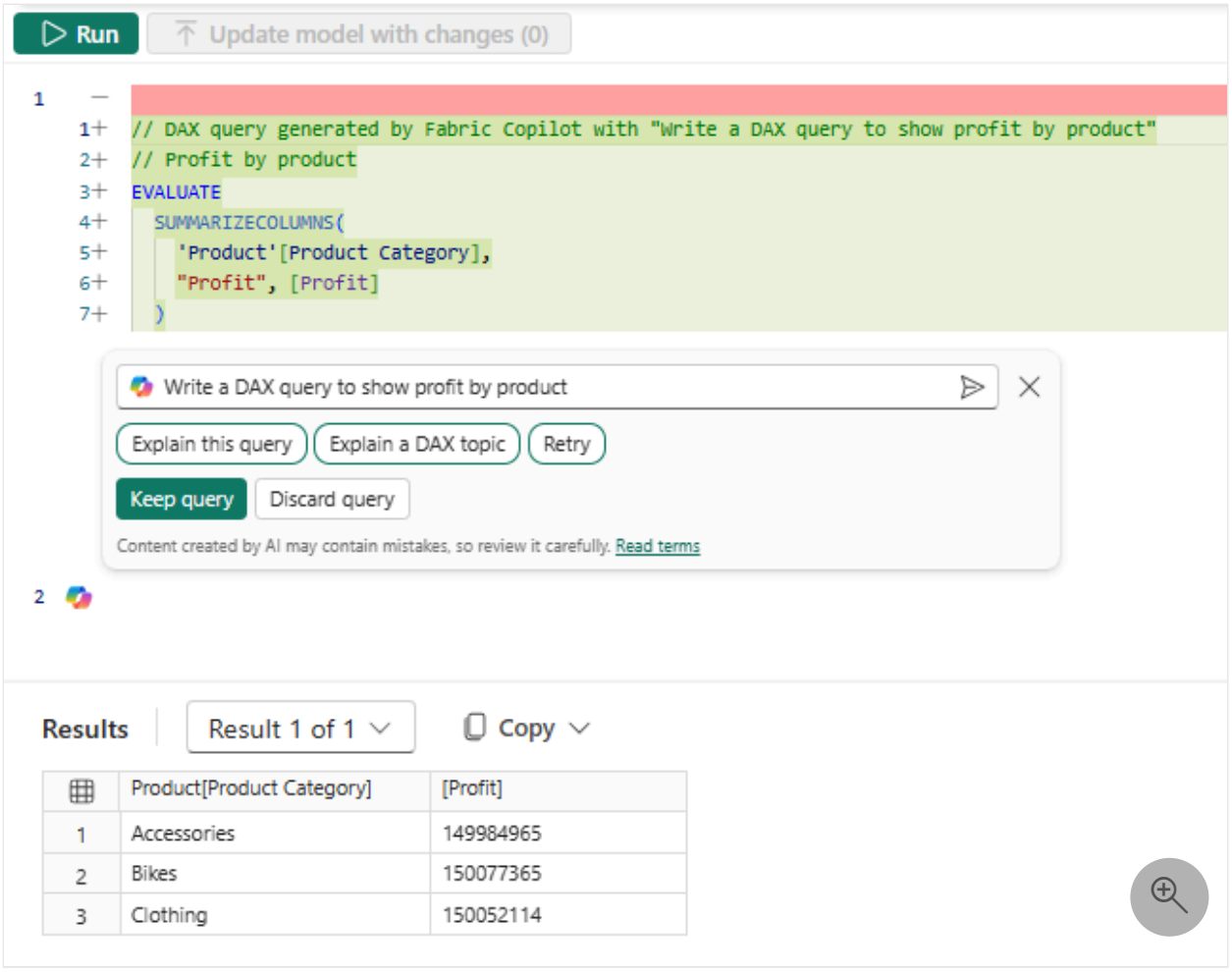
"Написание запроса DAX для отображения прибыли по продукту"



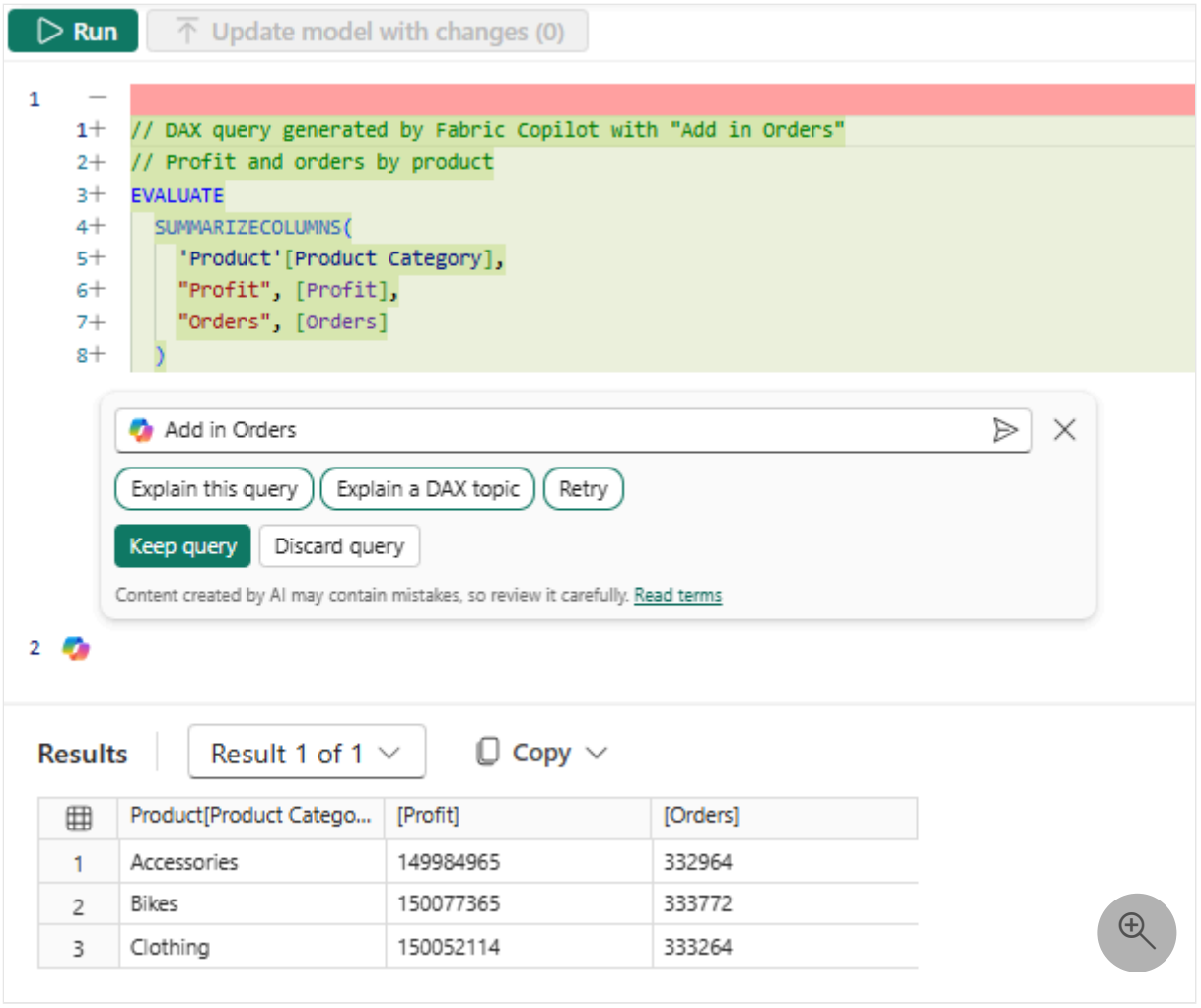
Через несколько минут я получаю запрос DAX.



Я могу выбрать **запустить**, чтобы просмотреть результаты перед сохранением запроса.



Я могу в разговорной форме вносить изменения в свой первоначальный запрос. Введите "Добавить в заказы" и нажмите `, затем введите`, и Copilot добавит дополнительную меру. Я могу снова **запустить** запрос, чтобы просмотреть результаты.

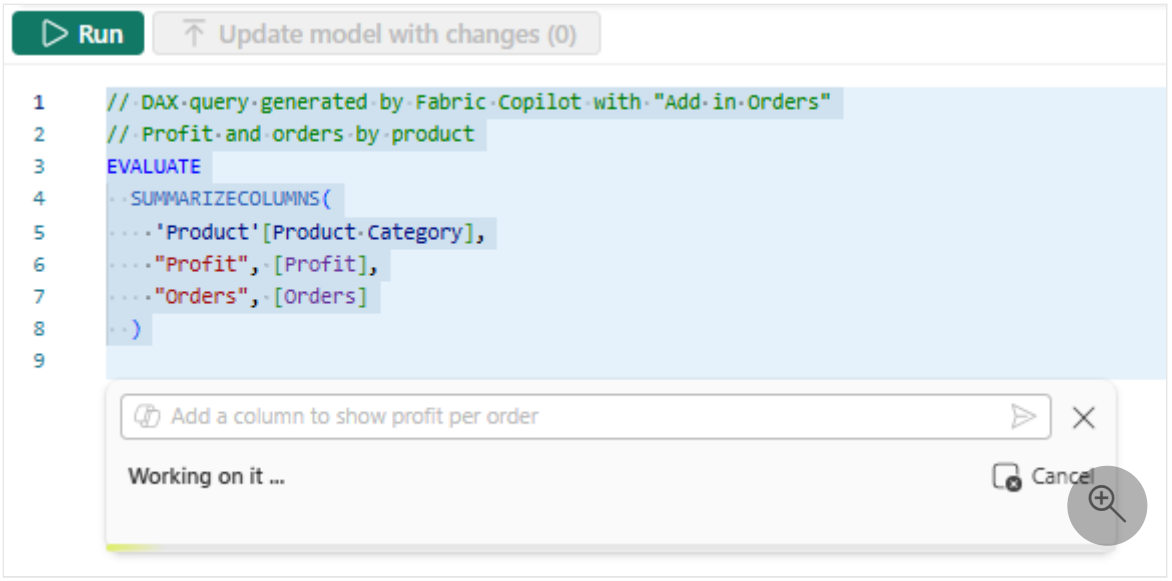


При выборе **и удержании запроса**, запрос добавляется на вкладку запросов.

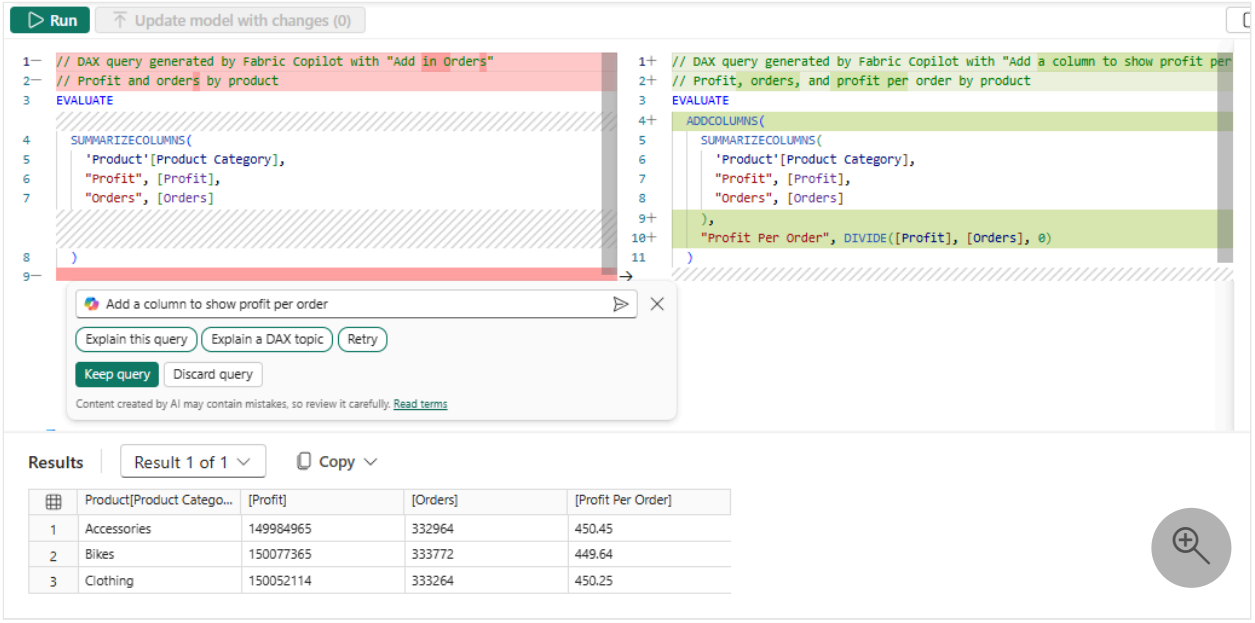
Добавьте столбец с представлением различий

Теперь добавим столбец в существующий запрос DAX. Сначала я выбираю запрос DAX, а затем вызываю Copilot. На этот раз выделение выполнено в соответствии с текстом, а фоновая заливка выделяет то, что отправляется в запросе к Copilot. Введите этот запрос и нажмите `введите`:

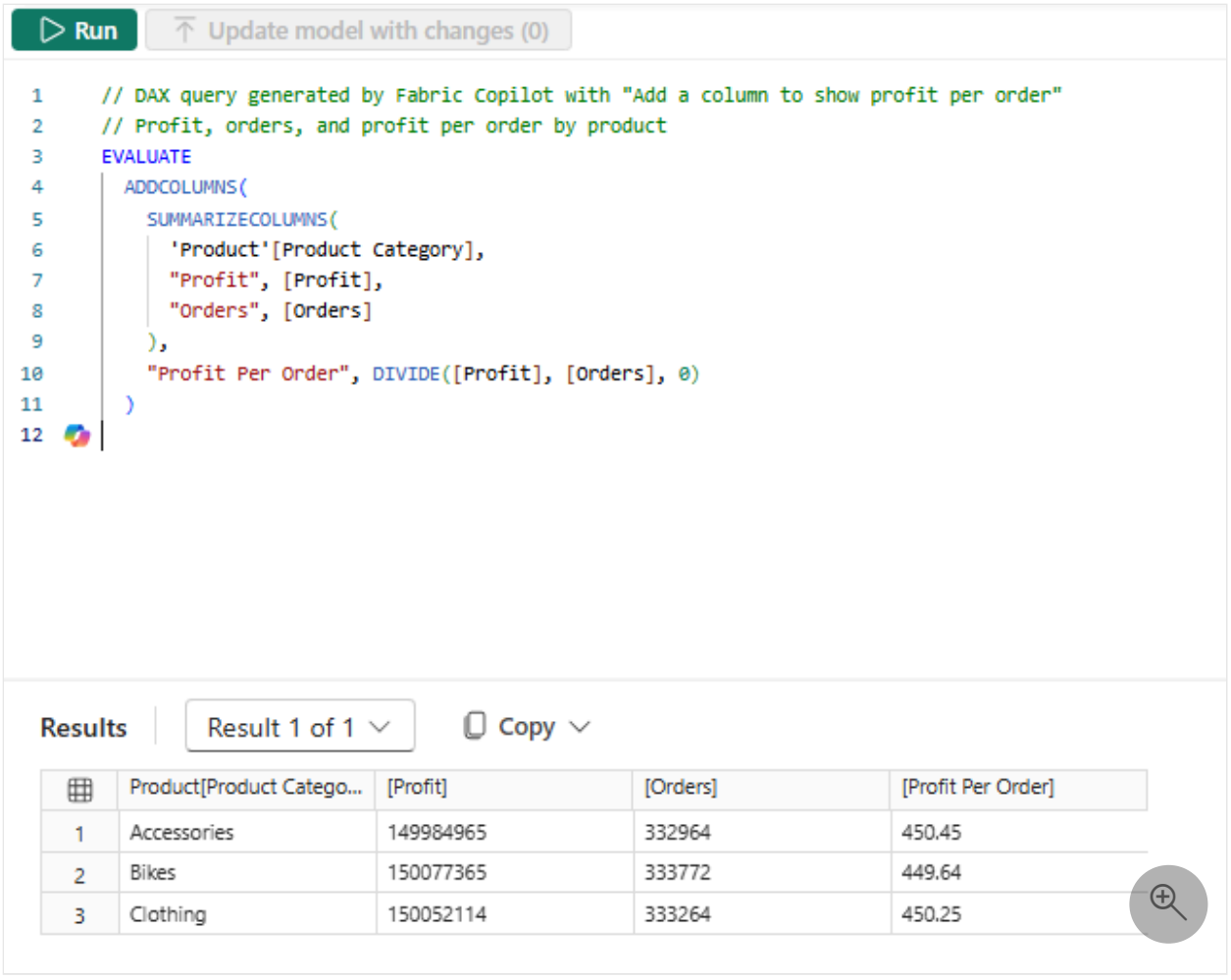
Добавьте столбец для отображения прибыли на каждый заказ



Теперь стилизация изменений в редакторе запросов DAX точно показывает, что именно изменил Copilot.

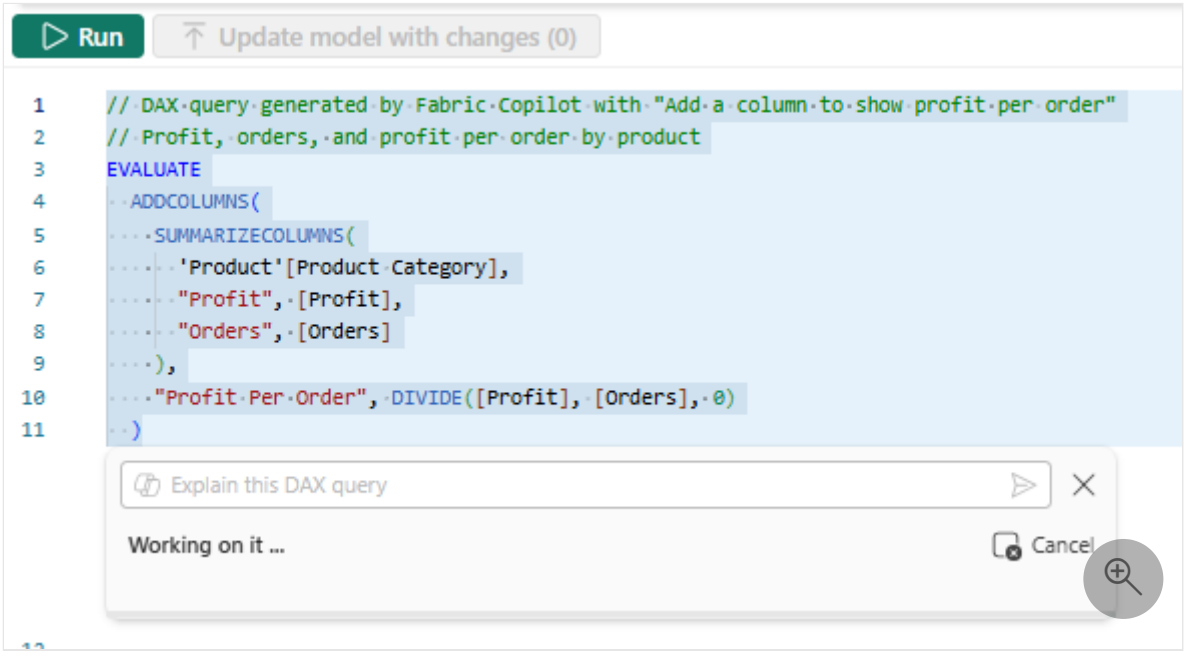


Я выбираю , **чтобы запустить** и посмотреть результаты, и **оставить запрос**, чтобы добавить его в редактор запросов.

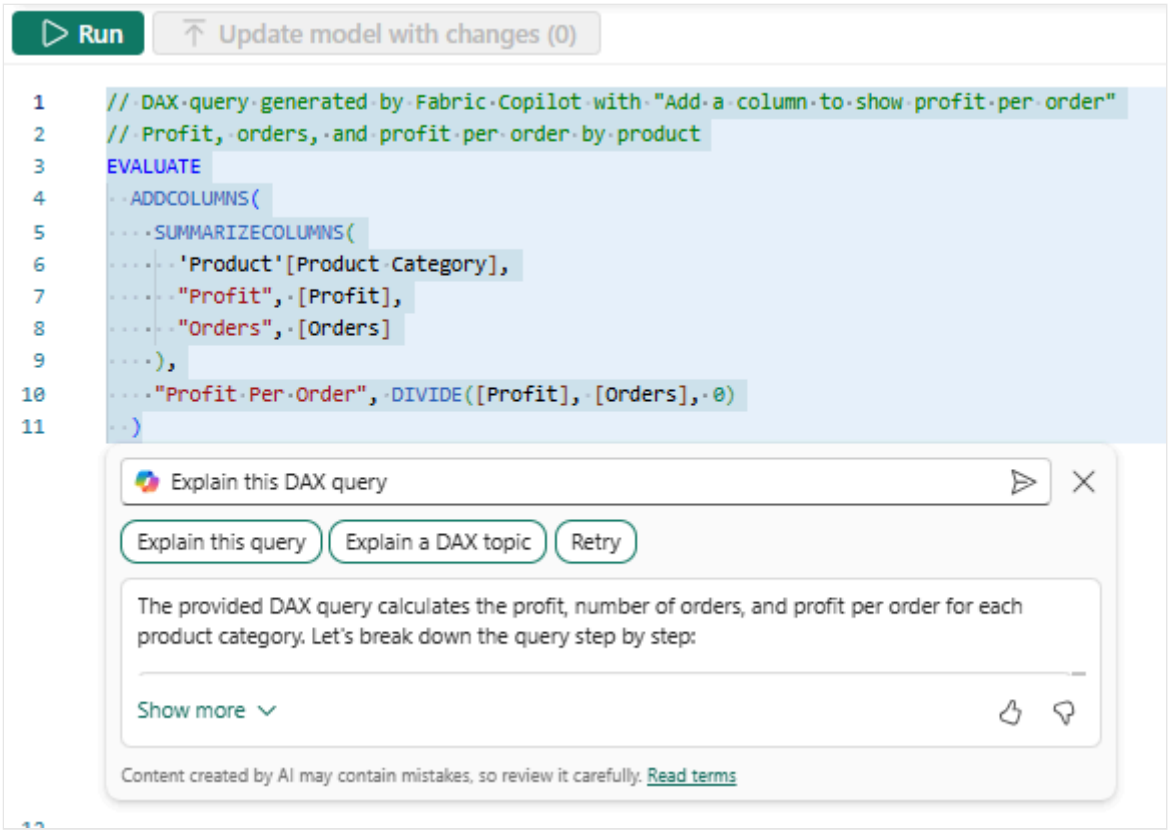


Объяснение запроса DAX

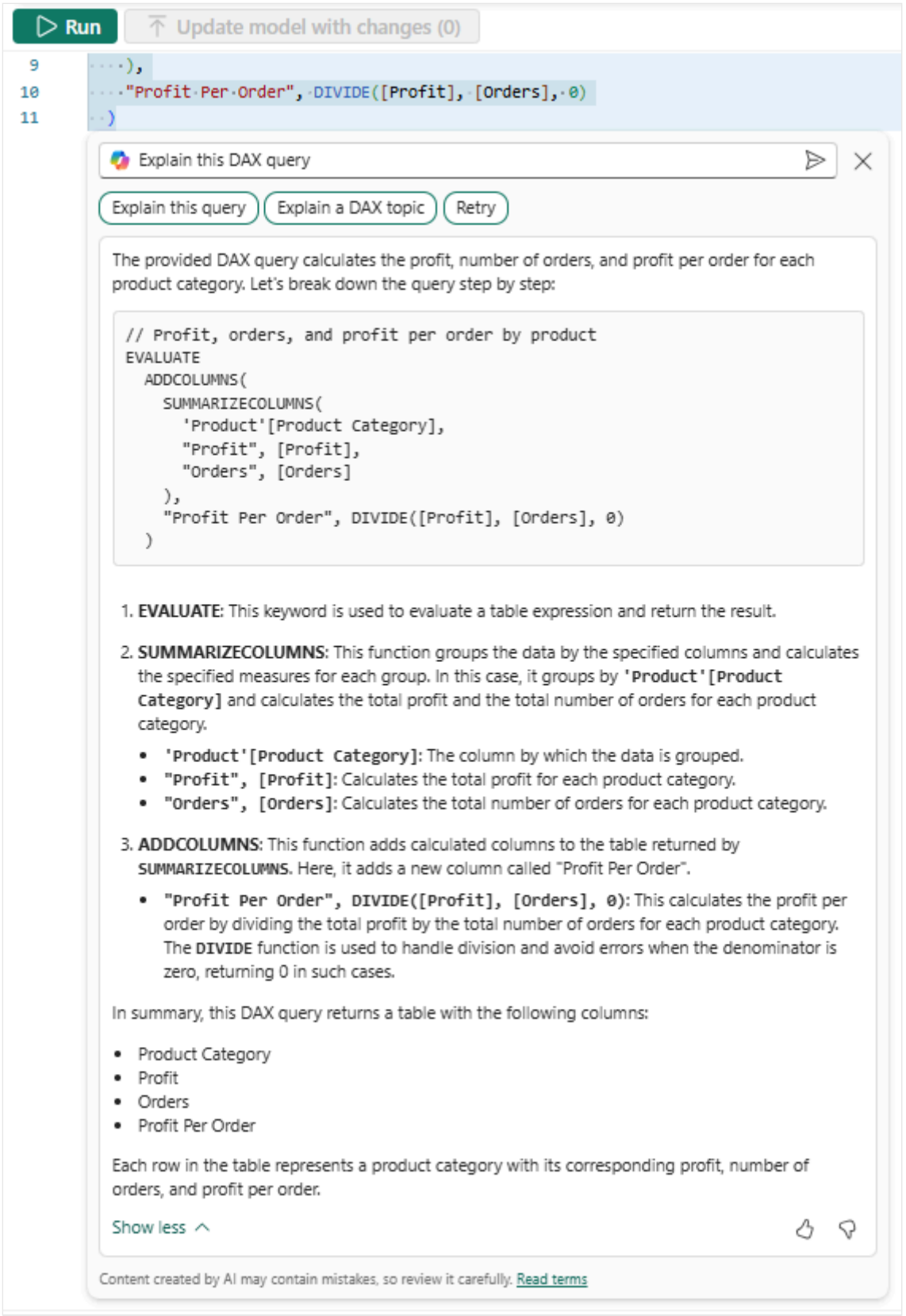
Copilot может предоставить подробное описание запроса DAX. Снова выберите весь запрос DAX и вызовите Copilot. На этот раз нажмите кнопку **Объяснить этот запрос**.



Теперь, вместо написания запросов DAX, он отвечает в встроенном Copilot.



Copilot создает несколько строк. Я нажимаю кнопку "**Показать больше**", чтобы увидеть полное объяснение.

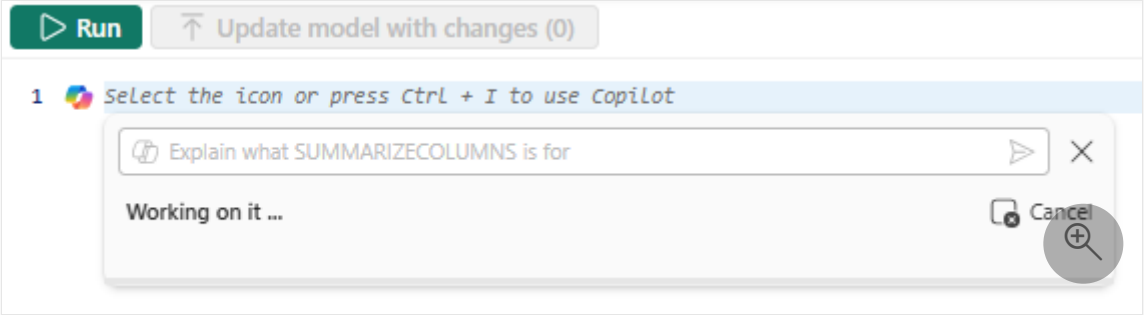


Теперь я понимаю каждую часть и используемые функции DAX.

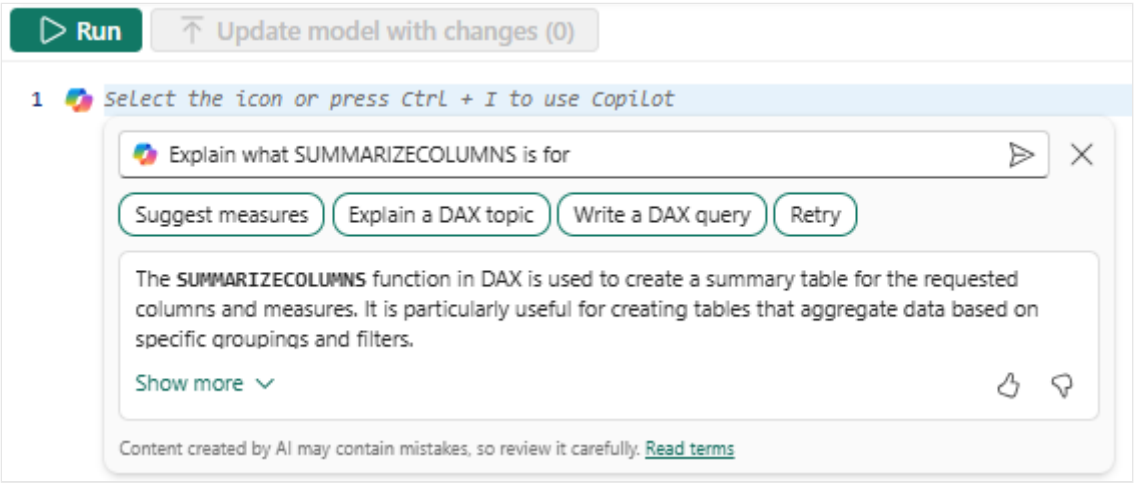
Объяснение темы DAX

Copilot также может более подробно объяснить функции DAX. Давайте спросим о SUMMARIZECOLUMNS. На новой вкладке запроса вызовите Copilot и отправьте следующий запрос:

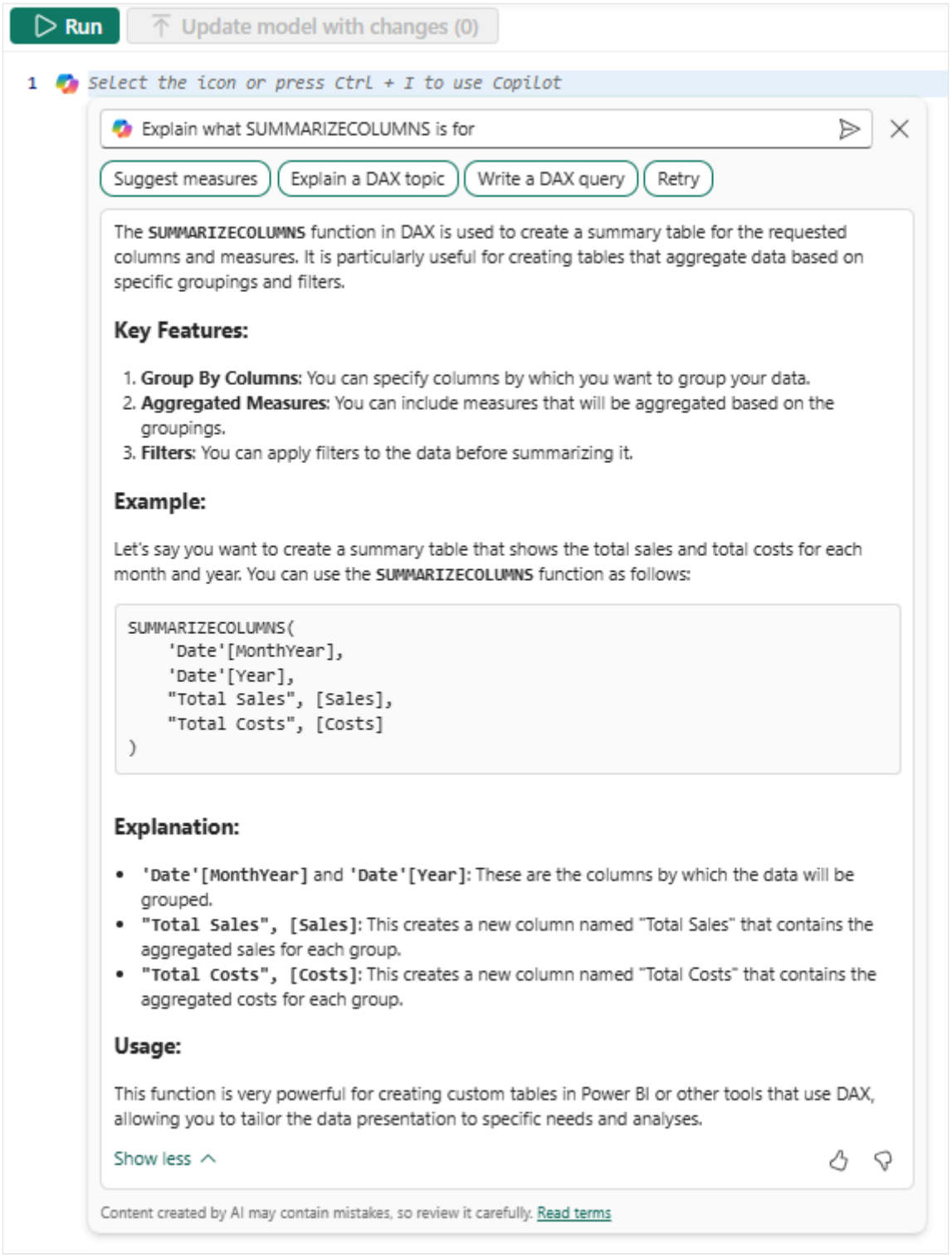
"Объясните, что такое SUMMARIZECOLUMNS"



Опять же, Copilot создает краткое описание:



Я могу выбрать **Показать больше**, чтобы увидеть более подробное объяснение с примером.



В примере также используются столбцы в моей модели.

Copilot для Fabric, помогающий писать DAX-запросы: Часто задаваемые вопросы об ответственном ИИ

Что такое Fabric Copilot, чтобы помочь писать запросы DAX?

Встроенный Copilot, доступный в представлении запросов DAX Power BI на рабочем столе или в рабочей области Power BI, помогает авторам и потребителям моделей запрашивать поддержку при создании запросов DAX. Эта помощь может включать объяснение запросов DAX и тем DAX, а также создание самих запросов DAX.

Что может сделать Fabric Copilot, чтобы помочь в написании запросов DAX?

Запрос пользователя может быть возвращен Copilot в виде запроса DAX, который пользователь может увидеть, а затем запустить, чтобы увидеть данные в сетке результатов. Можно выполнить дополнительные запросы и получить новый запрос DAX, сформировав его в формате диалога. Если запрос касается пояснения DAX-запроса или темы, во встроенном Copilot генерируется ответ на естественном языке. Разделы DAX могут включать в себя то, что выполняет функция DAX или какая функция DAX подходит для возврата данных определенным образом.

Каково предполагаемое использование Fabric Copilot для помощи в написании запросов DAX?

Предполагаемое использование Copilot заключается в создании запросов DAX для конкретной семантической модели, с помощью которую пользователь в настоящее время работает в Power BI.

Как была оценена помощь Fabric Copilot в написании запросов DAX? Какие метрики используются для измерения производительности?

Запросы DAX из нескольких семантических моделей Power BI, использующие различные пользовательские запросы, которые классифицируются по доступности данных в семантической модели и/или если для создания правильного ответа требуются дополнительные вычисления, а затем оцениваются по точности и производительности членами команды продукта. Это включало семантическую модель, соответствующую лучшим практикам. Пояснения из нескольких запросов DAX и разделов функций DAX, а затем проверенные на точность и удобочитаемость членами продуктовой команды.

Каковы ограничения Fabric Copilot для помощи в составлении запросов DAX? Как пользователи могут

свести к минимуму влияние Fabric Copilot на ограничения при написании DAX-запросов в системе?

- Чтобы использовать Copilot для записи запросов DAX, необходимо выбрать рабочую область с поддержкой Fabric.
- Copilot использует метаданные модели (таблицы, столбцы, меры), поэтому ответы запросов DAX отвечают только в контексте текущей семантической модели.
- Copilot использует метаданные модели (таблицы, столбцы, меры), поэтому предоставление богатых метаданных, которые являются описательными и понятными, даст лучшие результаты.
- Copilot может использовать min/max для получения примеров значений в режиме хранения импорта, но не может использовать другие данные в таблицах семантической модели.
- Copilot может быть недоступным или использовать сокращенную версию метаданных модели (таблицы, столбцы, меры) если число элементов метаданных модели превышает 500 таблиц, 1000 столбцов или 3000 мер в таблице, 2000 связей, 10 000 общих столбцов, 5 000 общих мер, 5 000 символов в одном выражении DAX или 256 символов в имени.
- Copilot может использовать свойство описания для метаданных модели, но ограничено только первыми 200 символами.
- Функции DAX, добавленные в последнее время, не могут быть известны в Copilot, и объяснения могут быть неверными.
- Запросы Copilot включают весь текст в текущем окне запроса; если текст выделен, информация о выделении учитывается, но весь текст в окне запроса всё равно включается.

Какие операционные факторы и параметры позволяют эффективно и ответственно использовать Fabric Copilot для записи запросов DAX?

- Операционные факторы и параметры включают текущую рабочую нагрузку на емкость Fabric и скорость сети.
- Copilot, помогающий писать запросы DAX, включен в как часть мер по обеспечению конфиденциальности, безопасности и ответственного использования в Fabric.

Как дать отзыв о Fabric Copilot для помощи в написании запросов DAX?

- Используйте пальцы вверх и пальцем вниз значки в встроенном Copilot для предоставления отзывов.
- Отправьте отзыв через службу поддержки Power BI.

Дополнительные сведения об этих ресурсах

- Обзор «Copilot» в Fabric и Power BI
- Представление запроса DAX
- Запросы DAX

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DAX пример модели

Статья • 10.04.2025

Образец модели Power BI Desktop **Adventure Works DW 2020** предназначен для содействия вашему DAX обучению. Модель основана на примере хранилища данных Adventure Works для AdventureWorksDW2017, однако данные были изменены в соответствии с целями примера модели.

Пример модели не содержит DAX формул. Однако она поддерживает сотни или даже тысячи потенциальных формул вычислений и запросов. Некоторые примеры функций, такие как CALCULATE, DATESBETWEEN, DATESIN PERIOD, IF и LOOKUPVALUE, могут быть добавлены в пример модели без изменений. Мы работаем над включением дополнительных примеров в другие справочные статьи о функциях, которые работают с образцом модели.

Сценарий



Компания Adventure Works представляет производителя велосипедов, который продает велосипеды и аксессуары на глобальных рынках. У компании есть данные хранилища данных, хранящиеся в базе данных SQL Azure.

Структура модели

Модель содержит семь таблиц:

Развернуть таблицу

таблица	Описание
клиент	Описывает клиентов и их географическое расположение. Клиенты покупают продукты через Интернет (интернет-продажи).
Дата	Существует три отношения между таблицами **даты** и **продаж**, датой заказа, датой доставки и сроком исполнения. Взаимосвязь даты заказа активна. Компания сообщает о продажах с использованием финансового года, который начинается

таблица	Описание
	1 июля каждого года. Таблица помечена как таблица дат, используя столбец **Дата**.
Продукт	Хранит только готовые продукты.
дистрибьютор	Описывает торговые посредники и их географическое расположение. Торговый посредник продаёт продукты своим клиентам.
продажи	Хранит строки с детализацией на уровне позиций заказа на продажу. Все финансовые значения находятся в долларах США (USD). Самая ранняя дата заказа — 1 июля 2017 г., а последняя дата заказа — 15 июня 2020 г.
заказ на продажу	Описываются номера заказов и номер позиции заказов, а также канал продаж: **Дилер** или **Интернет**. Эта таблица имеет отношение "один к одному" с таблицей **Sales**.
Торговая территория	Территории продаж организованы в группы (Северная Америка, Европа и Тихоокеанский регион), страны и регионы. Только Соединенные Штаты продают продукты на уровне региона.

Скачивание примера

Скачайте пример файла модели Power BI Desktop здесь.

Связанный контент

- Схема обучения: Использование DAX в Power BI Desktop
- Вопросы? Попробуйте обратиться к сообществу Power BI
- Предложения? Участие в разработке идей по улучшению Power BI

Общие сведения о функциях ORDERBY, PARTITIONBYи MATCHBY

Статья • 10.04.2025

Функции ORDERBY, PARTITIONBY и MATCHBY в DAX являются специальными функциями, которые можно использовать только вместе с функциями Window: INDEX, OFFSET, WINDOW, RANK, ROWNUMBER.

Понимание ORDERBY, PARTITIONBY и MATCHBY имеет критически важное значение для успешного использования функций Window. Примеры, приведенные здесь, используют OFFSET, но также применимы к другим оконным функциям.

Сценарий

Начнем с примера, который не использует функции Окна вообще. Ниже показана таблица, которая показывает общий объем продаж по цветам за календарный год. Существует несколько способов определения этой таблицы, но так как мы заинтересованы в понимании того, что происходит в DAX, мы будем использовать вычисляемую таблицу. Вот табличное выражение:

DAX

```
BasicTable =
    SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
    )
```

Вычисляемое табличное выражение использует SUMMARIZECOLUMNS, чтобы вычислить SUM столбца SalesAmount в таблице FactInternetSales, по столбцам Color из таблицы DimProduct и CalendarYear из таблицы DimDate. Вот результат:

Развернуть таблицу

Цвет	Календарный год	ПродажиТекущегоГода
"Черный"	2017	393885
"Черный"	2018	1818835
"Черный"	2019	3981638

Цвет	Календарный год	ПродажиТекущегоГода
"Черный"	2020	2644054
"Синий"	2019	994448
"Синий"	2020	1284648
Мульти	2019	48622
Мульти	2020	57849
"NA"	2019	207822
"NA"	2020	227295
"Красный"	2017	2961198
"Красный"	2018	3686935
"Красный"	2019	900175
"Красный"	2020	176022
Серебро	2017	326399
Серебро	2018	750026
Серебро	2019	2165176
Серебро	2020	1871788
Белый	2019	2517
Белый	2020	2589
"Желтый"	2018	163071
"Желтый"	2019	2072083
"Желтый"	2020	2621602

Теперь давайте предположим, что мы пытаемся решить бизнес-вопрос вычисления разницы в продажах, год за год для каждого цвета. На самом деле, нам нужен способ найти продажи для того же цвета в предыдущем году и вычитать их из продаж в текущем году, в контексте. Например, для сочетания [Red, 2019] мы ищем данные о продажах за [Red, 2018]. После этого мы можем вычитать его из текущих продаж и вернуть необходимое значение.

Использование OFFSET

OFFSET идеально подходит для типичного *сравнения с предыдущими* типами вычислений, необходимых для ответа на описанный выше бизнес-вопрос, так как это позволяет нам делать относительное движение. Наша первая попытка может быть:

DAX

```
1stAttempt =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
        )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "PreviousColorSales",
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation
            ),
            [CurrentYearSales]
        )
    )
```

Много происходит с этим выражением. Мы использовали ADDCOLUMNS для расширения предыдущей таблицы с добавлением столбца с именем PreviousColorSales. Содержимое этого столбца имеет значение CurrentYearSales, то есть SUM(FactInternetSales[SalesAmount]), для предыдущего цвета (извлекаемого с помощью OFFSET).

Результатом является:

Развернуть таблицу

Цвет	Календарный год	ПродажиТекущегоГода	ПредыдущиеColorSales
"Черный"	2017	393885	
"Черный"	2018	1818835	393885
"Черный"	2019	3981638	1818835
"Черный"	2020	2644054	3981638
"Синий"	2019	994448	2644054
"Синий"	2020	1284648	994448

Цвет	Календарный год	ПродажиТекущегоГода	ПредыдущиеColorSales
Мульти	2019	48622	1284648
Мульти	2020	57849	48622
"NA"	2019	207822	57849
"NA"	2020	227295	207822
"Красный"	2017	2961198	227295
"Красный"	2018	3686935	2961198
"Красный"	2019	900175	3686935
"Красный"	2020	176022	900175
Серебро	2017	326399	176022
Серебро	2018	750026	326399
Серебро	2019	2165176	750026
Серебро	2020	1871788	2165176
Белый	2019	2517	1871788
Белый	2020	2589	2517
"Желтый"	2018	163071	2589
"Желтый"	2019	2072083	163071
"Желтый"	2020	2621602	2072083

Это один шаг ближе к нашей цели, но если мы посмотрим внимательно, это не совсем соответствует тому, чего мы добиваемся. Например, для [Silver, 2017] для PreviousColorSales задано значение [Red, 2020].

Добавление ORDERBY

Это определение выше эквивалентно следующему:

DAX

```
1stAttemptWithORDERBY =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
```

```
        )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "PreviousColorSales",
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([Color], ASC, [CalendarYear], ASC, [CurrentYearSales],
ASC)
            ),
            [CurrentYearSales]
        )
    )
```

В этом случае вызов OFFSET использует ORDERBY для упорядочивания таблицы по Цвету и Календарному году в порядке возрастания, что определяет, какая строка считается предыдущей, возвращенной.

Причина, по которой эти два результата эквивалентны, заключается в том, что ORDERBY автоматически содержит все столбцы из связи, которые не находятся PARTITIONBY. Поскольку PARTITIONBY не указано, для параметра ORDERBY установлены значения Color, CalendarYear и CurrentYearSales. Однако, так как пары Color и CalendarYear в связи уникальны, добавление CurrentYearSales не изменяет результат. На самом деле, даже если бы мы только указали цвет в ORDERBY, результаты совпадают, так как CalendarYear будет автоматически добавлен. Это связано с тем, что функция добавит столько столбцов, сколько необходимо, к ORDERBY, чтобы обеспечить возможность уникальной идентификации каждой строки по столбцам ORDERBY и PARTITIONBY.

DAX

```
1stAttemptWithORDERBY =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
        )
    RETURN
    ADDCOLUMNS(
        vRelation,
        "PreviousColorSales",
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([Color])
            ),
            [CurrentYearSales]
```

```
        )
    )
```

Добавление PARTITIONBY

Теперь, чтобы *почти* получить нужный нам результат, мы можем использовать PARTITIONBY, как показано в следующем вычисляемом выражении таблицы.

DAX

```
UsingPARTITIONBY =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
        )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "PreviousColorSales",
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([CalendarYear]),
                PARTITIONBY ([Color])
            ),
            [CurrentYearSales]
        )
    )
```

Обратите внимание, что указание параметра ORDERBY здесь необязательно, так как ORDERBY автоматически включает все столбцы из набора данных, которые не указаны в PARTITIONBY. Таким образом, следующее выражение возвращает те же результаты, так как ORDERBY задано значение CalendarYear и CurrentYearSales автоматически:

DAX

```
UsingPARTITIONBYWithoutORDERBY =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
        )
    RETURN
    ADDCOLUMNS (
        vRelation,
```

```
        "PreviousColorSales",
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                PARTITIONBY ([Color])
            ),
            [CurrentYearSales]
        )
    )
```

> **Примечание**
>
> Хотя ORDERBY автоматически устанавливается как CalendarYear и CurrentYearSales, нет гарантии относительно порядка их добавления. Если CurrentYearSales добавляется перед CalendarYear, результирующий заказ не соответствует ожидаемому. **Указывайте явно ORDERBY и PARTITIONBY, чтобы избежать путаницы и неожиданных результатов**.

Оба выражения возвращают результат, которого мы добиваемся.

Развернуть таблицу

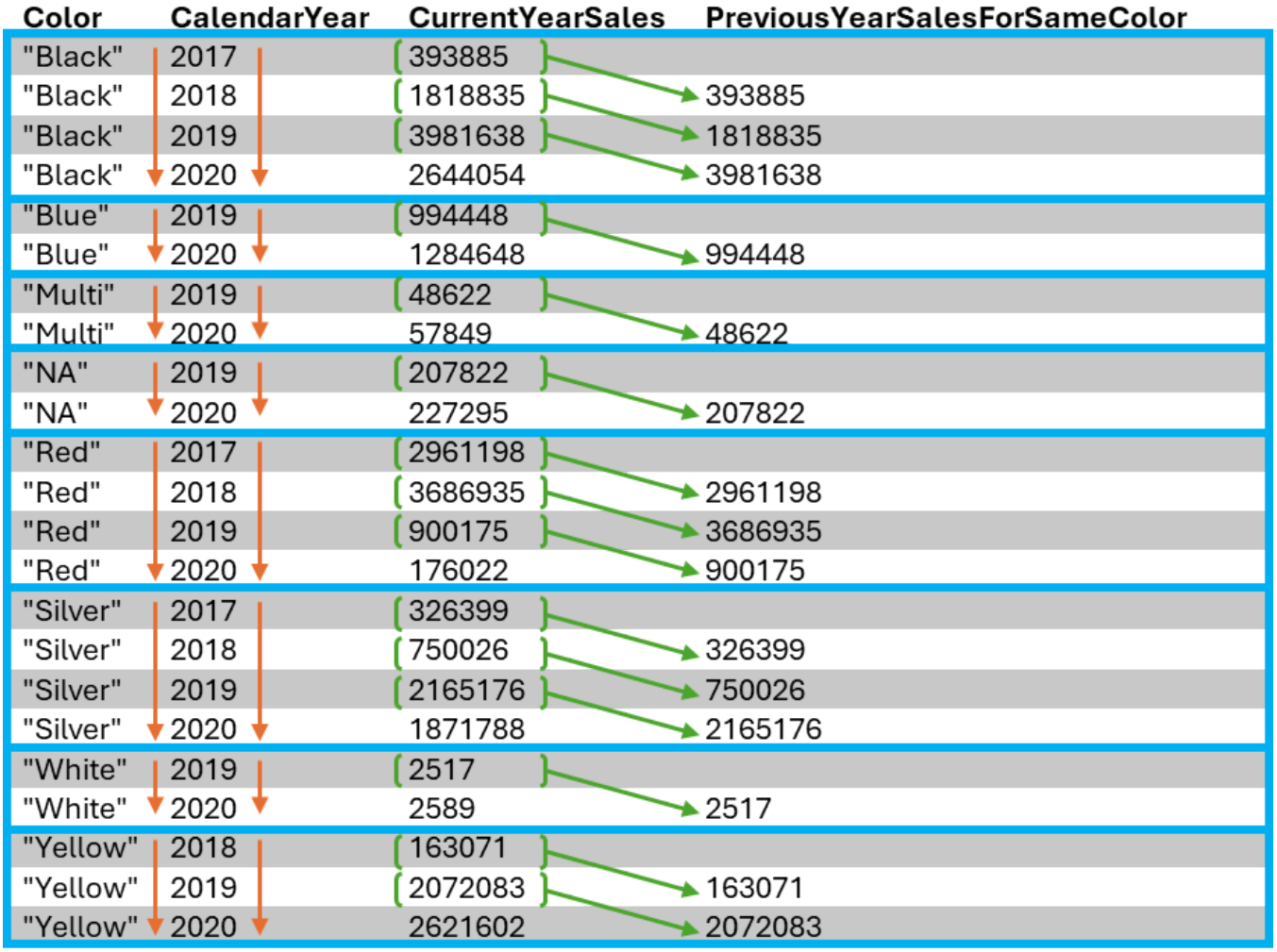
Цвет	Календарный год	ПродажиТекущегоГода	ПродажиЗаПрошлыйГодПоТомуЖеЦвету
"Черный"	2017	393885	
"Черный"	2018	1818835	393885
"Черный"	2019	3981638	1818835
"Черный"	2020	2644054	3981638
"Синий"	2019	994448	
"Синий"	2020	1284648	994448
Мульти	2019	48622	
Мульти	2020	57849	48622
"NA"	2019	207822	
"NA"	2020	227295	207822
"Красный"	2017	2961198	
"Красный"	2018	3686935	2961198

Цвет	Календарный год	ПродажиТекущегоГода	ПродажиЗаПрошлыйГодПоТомуЖеЦвету
"Красный"	2019	900175	3686935
"Красный"	2020	176022	900175
Серебро	2017	326399	
Серебро	2018	750026	326399
Серебро	2019	2165176	750026
Серебро	2020	1871788	2165176
Белый	2019	2517	
Белый	2020	2589	2517
"Желтый"	2018	163071	
"Желтый"	2019	2072083	163071
"Желтый"	2020	2621602	2072083

Как видно в этой таблице, столбец PreviousYearSalesForSameColor показывает продажи за предыдущий год для того же цвета. Для [Red, 2020] это возвращает продажи для [Red, 2019], и так далее. Если предыдущий год отсутствует, например в случае [Red, 2017], значение не возвращается.

Можно рассматривать PARTITIONBY как способ разделить таблицу на части для выполнения вычисления OFFSET. В приведенном выше примере таблица делится на столько частей, сколько есть цвета, по одному для каждого цвета. Затем в каждой части вычисляется OFFSET и сортируется по CalendarYear.

Визуально это происходит:



Во-первых, вызов PARTITIONBY приводит к тому, что таблица делится на части, каждая из которых соответствует определённому цвету. Это представлено светло-голубыми полями на изображении таблицы. Затем ORDERBY следит за тем, чтобы каждая часть была отсортирована по CalendarYear (что обозначено оранжевыми стрелками). Наконец, в каждой отсортированной части, для каждой строки OFFSET находит строку, расположенную выше, и возвращает это значение в столбце с названием PreviousYearSalesForSameColor. Так как для каждой первой строки в каждой части данных нет предыдущей строки в той же части, результат в этой строке для столбца PreviousYearSalesForSameColor будет пустым.

Чтобы добиться окончательного результата, мы просто должны вычесть CurrentYearSales из продаж предыдущего года за тот же цвет, возвращенный вызовом OFFSET. Поскольку нас интересуют не продажи за предыдущий год для того же цвета, а продажи за текущий год и их изменения по сравнению с прошлым годом. Ниже приведено окончательное вычисляемое табличное выражение:

DAX

```
FinalResult =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
```

```
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
        )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "YoYSalesForSameColor",
        [CurrentYearSales] -
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([CalendarYear]),
                PARTITIONBY ([Color])
            ),
            [CurrentYearSales]
        )
    )
```

И вот результат этого выражения:

Развернуть таблицу

Цвет	Календарный год	ПродажиТекущегоГода	Годовой объем продаж для одинакового цвета
"Черный"	2017	393885	393885
"Черный"	2018	1818835	1424950
"Черный"	2019	3981638	2162803
"Черный"	2020	2644054	-1337584
"Синий"	2019	994448	994448
"Синий"	2020	1284648	290200
Мульти	2019	48622	48622
Мульти	2020	57849	9227
"NA"	2019	207822	207822
"NA"	2020	227295	19473
"Красный"	2017	2961198	2961198
"Красный"	2018	3686935	725737
"Красный"	2019	900175	-2786760

Цвет	Календарный год	ПродажиТекущегоГода	Годовой объем продаж для одинакового цвета
"Красный"	2020	176022	-724153
Серебро	2017	326399	326399
Серебро	2018	750026	423627
Серебро	2019	2165176	1415150
Серебро	2020	1871788	-293388
Белый	2019	2517	2517
Белый	2020	2589	72
"Желтый"	2018	163071	163071
"Желтый"	2019	2072083	1909012
"Желтый"	2020	2621602	549519

Использование MATCHBY

Возможно, вы заметили, что мы вообще не указали MATCHBY . В этом случае это не обязательно. Столбцы в ORDERBY и PARTITIONBY (насколько они были указаны в приведенных выше примерах) достаточны для уникальной идентификации каждой строки. Так как мы не указали MATCHBY, столбцы, указанные в ORDERBY и PARTITIONBY, используются для уникальной идентификации каждой строки, чтобы их можно было сравнить и OFFSET дало значимый результат. Если столбцы в ORDERBY и PARTITIONBY не могут однозначно идентифицировать каждую строку, дополнительные столбцы можно добавить в ORDERBY предложение, если эти дополнительные столбцы позволяют каждой строке быть однозначно идентифицированы. Если это невозможно, возвращается ошибка. В этом последнем случае указание MATCHBY может помочь устранить ошибку.

Если MATCHBY задано, столбцы в MATCHBY и PARTITIONBY используются для уникальной идентификации каждой строки. Если это невозможно, возвращается ошибка. Даже если MATCHBY не требуется, явно укажите MATCHBY, чтобы избежать путаницы.

Далее из приведенных выше примеров ниже приведено последнее выражение:

DAX

```
FinalResult =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
```

```
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
        )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "YoYSalesForSameColor",
        [CurrentYearSales] -
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([CalendarYear]),
                PARTITIONBY ([Color])
            ),
            [CurrentYearSales]
        )
    )
```

Если мы хотим явно указать, как строку следует однозначно идентифицировать, мы можем задать MATCHBY, как показано в следующем эквивалентном выражении.

DAX

```
FinalResultWithExplicitMATCHBYOnColorAndCalendarYear =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
        )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "YoYSalesForSameColor",
        [CurrentYearSales] -
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([CalendarYear]),
                PARTITIONBY ([Color]),
                MATCHBY ([Color], [CalendarYear])
            ),
            [CurrentYearSales]
        )
    )
```

Так как MATCHBY указан, оба столбца, указанные в MATCHBY, а также в PARTITIONBY используются для уникальной идентификации строк. Поскольку цвет указан и в MATCHBY,

и в PARTITIONBY, следующее выражение эквивалентно предыдущему выражению:

DAX

```
FinalResultWithExplicitMATCHBYOnCalendarYear =
    VAR vRelation = SUMMARIZECOLUMNS (
        DimProduct[Color],
        DimDate[CalendarYear],
        "CurrentYearSales", ROUND ( SUM ( FactInternetSales[SalesAmount] ), 0 )
        )
    RETURN
    ADDCOLUMNS (
        vRelation,
        "YoYSalesForSameColor",
        [CurrentYearSales] -
        SELECTCOLUMNS (
            OFFSET (
                -1,
                vRelation,
                ORDERBY ([CalendarYear]),
                PARTITIONBY ([Color]),
                MATCHBY ([CalendarYear])
            ),
            [CurrentYearSales]
        )
    )
```

Так как указание MATCHBY не требуется в примерах, которые мы рассмотрели до сих пор, давайте рассмотрим немного другой пример, который требует MATCHBY. В этом случае у нас есть список позиций заказа. Каждая строка представляет собой позицию в заказе. Заказ может содержать несколько строк заказа, а строка заказа 1 отображается во многих заказах. Кроме того, для каждой строки заказа у нас есть ProductKey и SalesAmount. Пример соответствующих столбцов в таблице выглядит следующим образом:

Развернуть таблицу

НомерЗаказа	Номер строки заказа	Ключ продукта	СуммаПродаж
SO51900	1	528	4.99
SO51948	1	528	5.99
SO52043	1	528	4.99
SO52045	1	528	4,99
SO52094	1	528	4.99
SO52175	1	528	4.99

НомерЗаказа	Номер строки заказа	Ключ продукта	СуммаПродаж
SO52190	1	528	4,99
SO52232	1	528	4.99
SO52234	1	528	4.99
SO52234	2	529	3.99

Обратите внимание, что SalesOrderNumber и SalesOrderLineNumber требуются для уникальной идентификации строк.

Для каждого заказа мы хотим вернуть предыдущий объем продаж одного и того же продукта (представленного ProductKey), отсортировав заказы по SalesAmount в порядке убывания. Следующее выражение не будет работать, поскольку в vRelation может быть несколько строк, которые передаются в OFFSET.

DAX

```
ThisExpressionFailsBecauseMATCHBYIsMissing =
    ADDCOLUMNS (
        FactInternetSales,
        "Previous Sales Amount",
            SELECTCOLUMNS (
                OFFSET (
                    -1,
                    FactInternetSales,
                    ORDERBY ( FactInternetSales[SalesAmount], DESC ),
                    PARTITIONBY ( FactInternetSales[ProductKey] )
                ),
                FactInternetSales[SalesAmount]
            )
    )
```

Это выражение возвращает ошибку: "OFFSETПараметр отношения может иметь повторяющиеся строки, которые не разрешены".

Чтобы это выражение работало, необходимо, чтобы MATCHBY было указано и включало все столбцы, которые однозначно определяют строку. MATCHBY Здесь требуется, так как отношение FactInternetSales не содержит явных ключей или уникальных столбцов. Однако столбцы SalesOrderNumber и SalesOrderLineNumber вместе образуют *составной ключ*, где их существование вместе уникально в связи и поэтому может однозначно идентифицировать каждую строку. Просто указать SalesOrderNumber или SalesOrderLineNumber недостаточно, так как оба столбца содержат повторяющиеся значения. Следующее выражение решает проблему:

DAX

```
ThisExpressionWorksBecauseOfMATCHBY =
    ADDCOLUMNS (
        FactInternetSales,
        "Previous Sales Amount",
            SELECTCOLUMNS (
                OFFSET (
                    -1,
                    FactInternetSales,
                    ORDERBY ( FactInternetSales[SalesAmount], DESC ),
                    PARTITIONBY ( FactInternetSales[ProductKey] ),
                    MATCHBY ( FactInternetSales[SalesOrderNumber],
                                FactInternetSales[SalesOrderLineNumber] )
                ),
                FactInternetSales[SalesAmount]
            )
    )
```

И это выражение действительно возвращает те результаты, которые мы ищем.

Развернуть таблицу

НомерЗаказа	Номер строки заказа	Ключ продукта	СуммаПродаж	Предыдущая сумма продаж
SO51900	1	528	5.99	
SO51948	1	528	4.99	5.99
SO52043	1	528	4.99	4.99
SO52045	1	528	4.99	4.99
SO52094	1	528	4.99	4.99
SO52175	1	528	4,99 ₽	4,99
SO52190	1	528	4,99	4.99
SO52232	1	528	4.99	4.99
SO52234	1	528	4.99	4.99
SO52234	2	529	3.99	

Связанный контент

ORDERBY PARTITIONBY MATCHBY INDEX OFFSET WINDOW RANK ROWNUMBER

Соответствующее использование функций ошибок

Статья • 23.01.2025

В качестве моделиировщика данных при написании выражения DAX, которое может вызвать ошибку во время оценки, можно рассмотреть возможность использования двух полезных функций DAX.

- Функция ISERROR, которая принимает одно выражение и возвращает TRUE, если это выражение приводит к ошибке.
- Функция IFERROR, которая принимает два выражения. Если первое выражение приведет к ошибке, возвращается значение второго выражения. На самом деле это более оптимизированная реализация вложения функции ISERROR внутри функции IF.

Однако, хотя эти функции могут быть полезными и могут способствовать написанию простых выражений, они также могут значительно снизить производительность вычислений. Это может произойти, так как эти функции увеличивают количество необходимых проверок подсистемы хранилища.

Большинство ошибок во время оценки вызваны непредвиденными значениями BLANKs или нулевыми значениями или недопустимым преобразованием типов данных.

Рекомендации

Лучше избежать использования функций ISERROR и IFERROR. Вместо этого при разработке модели и написании выражений применяются оборонительные стратегии. Стратегии могут включать:

- **Обеспечение загрузки данных качества в модель:** использовать преобразования Power Query для удаления или замены недопустимых или отсутствующих значений, а также для задания правильных типов данных. Преобразование Power Query также можно использовать для фильтрации строк при возникновении ошибок, таких как недопустимое преобразование данных.

  Качество данных также можно контролировать, установив для столбца модели **свойство Is Nullable** в положение Off, что приведет к сбою обновления данных, если будут обнаружены пустые значения. В случае сбоя

данные, загруженные в результате успешного обновления, останутся в таблицах.

- **Использование функции IF:** логическое тестовое выражение функции IF может определить, произойдет ли ошибочный результат. Обратите внимание, что, как и функции ISERROR и IFERROR, эта функция может привести к дополнительным сканированиям подсистемы хранилища, но, скорее всего, будет лучше, чем они, так как не требуется вызывать ошибку.
- **Использование функций, терпимых к ошибкам:** некоторые функции DAX тестируют и компенсируют ошибочные состояния. Эти функции позволяют ввести альтернативный результат, возвращаемый вместо этого. Функция DIVIDE является одним из таких примеров. Дополнительные рекомендации по этой функции см. в статье DAX: DIVIDE функции и оператора деления (/).

Пример

Следующее выражение меры проверяет, вызовет ли это ошибку. Он возвращает BLANK в этом экземпляре (например, если функция IF не предоставляется с выражением value-if-false).

DAX

```
Profit Margin
= IF(ISERROR([Profit] / [Sales]))
```

Эта следующая версия выражения меры была улучшена с помощью функции IFERROR вместо функций IF и ISERROR.

DAX

```
Profit Margin
= IFERROR([Profit] / [Sales], BLANK())
```

Однако эта окончательная версия выражения меры достигает того же результата, но более эффективно и элегантно.

DAX

```
Profit Margin
= DIVIDE([Profit], [Sales])
```

Связанное содержимое

- Путь обучения: Использование DAX в Power BI Desktop
- Вопросы? Попробуйте обратиться к сообществу Power BI
- Предложения? Вносите идеи по улучшению Power BI

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Avoid converting BLANKs to values

Статья • 12.04.2025

В качестве моделиатора данных при написании выражений мер могут возникнуть случаи, когда не удается вернуть понятное значение. В таких случаях вы можете испытывать искушение вернуть значение, например, ноль. Мы рекомендуем тщательно определить, является ли этот дизайн эффективным и практическим.

Рассмотрим следующее определение меры, в которой явно преобразуются результаты BLANK в ноль.

DAX

```
Sales (No Blank) =
IF(
    ISBLANK([Sales]),
    0,
    [Sales]
)
```

Consider another measure definition that also converts BLANK results to zero.

DAX

```
Profit Margin =
DIVIDE([Profit], [Sales], 0)
```

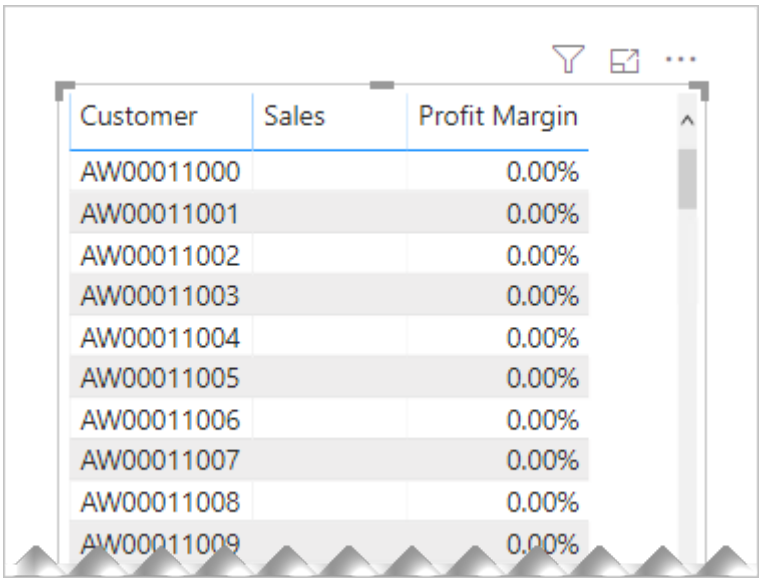
Функция DIVIDE делит меру **прибыли** на меру **продаж**. Если результат равен нулю или BLANK, третий аргумент — альтернативный результат (который является необязательным) возвращается. В этом примере, так как нуль передается в качестве альтернативного результата, мера гарантируется всегда возвращать значение.

Эти меры неэффективны и приводят к плохим конструкциям отчетов.

Когда они добавляются в визуализацию отчета, Power BI пытается получить все группировки в контексте фильтра. Оценка и получение больших результатов запросов часто приводит к медленному формированию отчетов. Каждый пример меры эффективно преобразует разреженное вычисление в плотное, заставляя Power BI использовать больше памяти, чем необходимо.

Кроме того, слишком большое количество группировок часто перегружает пользователей отчета.

Let's see what happens when the **Profit Margin** measure is added to a table visual, grouping by customer.



Customer	Sales	Profit Margin
AW00011000		0.00%
AW00011001		0.00%
AW00011002		0.00%
AW00011003		0.00%
AW00011004		0.00%
AW00011005		0.00%
AW00011006		0.00%
AW00011007		0.00%
AW00011008		0.00%
AW00011009		0.00%

The table visual displays an overwhelming number of rows. (На самом деле в модели есть 18 484 клиентов, поэтому таблица пытается отобразить всех из них.) Обратите внимание, что клиенты на экране не осуществили каких-либо продаж. Yet, because the **Profit Margin** measure always returns a value, they are displayed.

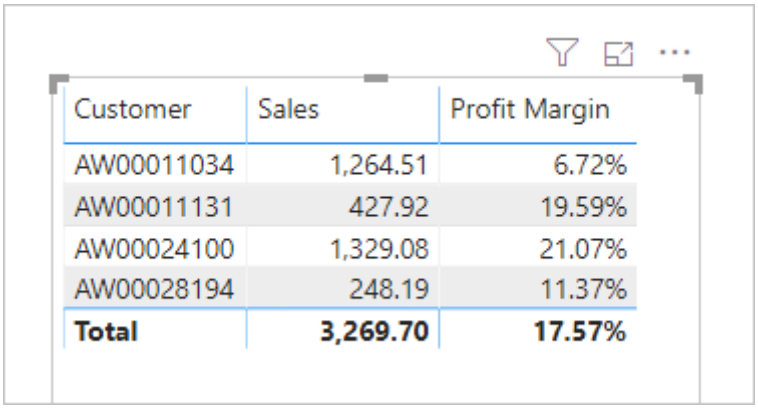
> ⓘ **Примечание**
>
> Если в визуальном элементе слишком много точек данных, Power BI может использовать стратегии сокращения данных для удаления или суммирования больших результатов запроса. For more information, see Data point limits and strategies by visual type.

Посмотрим, что произойдет, когда будет улучшено определение меры рентабельности . It now returns a value only when the **Sales** measure isn't BLANK (or zero).

DAX

```
Profit Margin =
DIVIDE([Profit], [Sales])
```

Визуальный элемент таблицы теперь отображает только клиентов, которые сделали продажи в текущем контексте фильтра. Улучшенная мера дает более эффективный и практический опыт для пользователей отчета.



Customer	Sales	Profit Margin
AW00011034	1,264.51	6.72%
AW00011131	427.92	19.59%
AW00024100	1,329.08	21.07%
AW00028194	248.19	11.37%
Total	**3,269.70**	**17.57%**

Совет

При необходимости можно настроить визуализацию для показа всех групп (возвращающих значения или BLANK) в контексте фильтра, включив параметр **Показывать элементы без данных**.

Рекомендация

Рекомендуется, чтобы ваши результаты измерений возвращали BLANK, когда невозможно вернуть значимое значение.

Этот подход к проектированию эффективен, что позволяет Power BI быстрее отображать отчеты. Also, returning BLANK is better because report visuals—by default—eliminate groupings when summarizations are BLANK.

Связанное содержимое

- Learning path: Use DAX in Power BI Desktop
- Вопросы? Попробуйте задать вопрос сообществу Power BI
- Предложения? Contribute ideas to improve Power BI

Избегайте использования FILTER в качестве аргумента фильтра

Статья • 13.04.2025

В качестве моделировщика данных вы обычно создаете выражения DAX, которые необходимо оценивать в измененном контексте фильтрации. Например, можно написать определение меры, чтобы вычислить продажи для "продуктов с высокой маржой". Далее мы рассмотрим этот расчет в этой статье.

> ⓘ **Примечание**
>
> Эта статья особенно актуальна для вычислений моделей, которые применяют фильтры к таблицам импорта.

Функции CALCULATE и CALCULATETABLEDAX являются важными и полезными функциями. Они позволяют создавать вычисления, которые удаляют или добавляют фильтры или изменяют пути связи. Это делается путем передачи аргументов фильтра, которые являются логическими выражениями, табличными выражениями или специальными функциями фильтра. В этой статье рассматриваются только логические и табличные выражения.

Рассмотрим следующее определение меры, которое вычисляет продажи красных продуктов с помощью табличного выражения. Он заменит все фильтры, которые могут применяться к таблице **Product**.

DAX

```
Red Sales =
CALCULATE(
    [Sales],
    FILTER('Product', 'Product'[Color] = "Red")
)
```

Функция CALCULATE принимает табличное выражение, возвращаемое функцией FILTERDAX, которая вычисляет его выражение фильтра для каждой строки таблицы **Product**. Он достигает правильного результата — результата продаж для красных продуктов. Однако его можно добиться гораздо эффективнее с помощью логического выражения.

Ниже приведено улучшенное определение меры, которое использует логическое выражение вместо табличного выражения. Функция KEEPFILTERSDAX гарантирует, что все

существующие фильтры, примененные к столбцу **Color**, сохраняются и не перезаписываются.

```
DAX
```

```
Red Sales =
CALCULATE(
    [Sales],
    KEEPFILTERS('Product'[Color] = "Red")
)
```

По возможности рекомендуется передавать аргументы фильтра в виде логических выражений. Это связано с тем, что таблицы модели импорта хранятся в памяти в виде колоночных хранилищ. Они явно оптимизированы для эффективного фильтрации столбцов таким образом.

Однако существуют ограничения, которые применяются к логическим выражениям, когда они используются в качестве аргументов фильтра. They:

- Не удается ссылаться на столбцы из нескольких таблиц
- Не удается ссылаться на меру
- Cannot use nested CALCULATE functions
- Не удается использовать функции, которые сканируют или возвращают таблицу

Это означает, что вам потребуется использовать табличные выражения для более сложных требований к фильтру.

Рассмотрим теперь другое определение меры. Требование заключается в расчете продаж, но только для тех месяцев, имеющих прибыль.

```
DAX
```

```
Sales for Profitable Months =
CALCULATE(
    [Sales],
    FILTER(
        VALUES('Date'[Month]),
        [Profit] > 0
    )
)
```

В этом примере необходимо использовать функцию FILTER. It's because it requires evaluating the **Profit** measure to eliminate those months that didn't achieve a profit. It's not possible to use a measure in a Boolean expression when it's used as a filter argument.

Рекомендации

Для оптимальной производительности рекомендуется использовать логические выражения в качестве аргументов фильтра по возможности.

Таким образом, FILTER функцию следует использовать только при необходимости. Его можно использовать для фильтрации сложных сравнений столбцов. Эти сравнения столбцов могут включать:

- Меры
- Другие столбцы
- Использование функции ORDAX или логического оператора OR (||)

Связанное содержимое

- функции фильтра (DAX)
- Learning path: Use DAX in Power BI Desktop
- Вопросы? Попробуйте задать вопрос сообществу Power BI
- Предложения? Вносите идеи по улучшению Power BI

Column and measure references

Статья • 12.04.2025

As a data modeler, your DAX expressions will refer to model columns and measures. Столбцы и меры всегда связаны с таблицами моделей, но такие связи отличаются, поэтому мы даём разные рекомендации, как на них ссылаться в выражениях.

Columns

Столбец — это объект уровня таблицы, а имена столбцов должны быть уникальными в таблице. Таким образом, возможно, что одно и то же имя столбца используется несколько раз в модели, обеспечивая, что они принадлежат разным таблицам. Существует еще одно правило: имя столбца не может иметь такое же имя, как имя меры или имя иерархии, которое существует в той же таблице.

In general, DAX will not force using a *fully qualified* reference to a column. Полностью квалифицированная ссылка означает, что имя таблицы предшествует имени столбца.

Ниже приведен пример определения вычисляемого столбца с использованием только ссылок на имена столбцов. Столбцы **Sales** и **Cost** относятся к таблице с именем **Orders**.

```
DAX

Profit = [Sales] - [Cost]
```

The same definition can be rewritten with fully qualified column references.

```
DAX

Profit = Orders[Sales] - Orders[Cost]
```

Sometimes, however, you'll be required to use fully qualified column references when Power BI detects ambiguity. When entering a formula, a red squiggly and error message will alert you. Also, some DAX functions like the LOOKUPVALUEDAX function, require the use of fully qualified columns.

It's recommended you always fully qualify your column references. Причины приведены в разделе рекомендаций .

Меры

Мера — это объект уровня модели. По этой причине имена мер должны быть уникальными в модели. However, in the **Fields** pane, report authors will see each measure associated with a single model table. This association is set for cosmetic reasons, and you can configure it by setting the **Home Table** property for the measure. For more information, see Measures in Power BI Desktop (Organizing your measures).

It's possible to use a fully qualified measure in your expressions. DAX intellisense will even offer the suggestion. Однако это не обязательно, и это не рекомендуется. If you change the home table for a measure, any expression that uses a fully qualified measure reference to it will break. Затем необходимо изменить каждую нерабочую формулу, чтобы удалить (или обновить) ссылку на измерение.

It's recommended you never qualify your measure references. Причины приведены в разделе рекомендаций .

Рекомендации

Наши рекомендации просты и просты для запоминания:

- Всегда используйте полностью квалифицированные ссылки на столбцы
- Никогда не используйте полностью квалифицированные ссылки на меры

Вот почему:

- **Formula entry**: Expressions will be accepted, as there won't be any ambiguous references to resolve. Also, you'll meet the requirement for those DAX functions that require fully qualified column references.
- **Robustness**: Expressions will continue to work, even when you change a measure home table property.
- **Readability**: Expressions will be quick and easy to understand—you'll quickly determine that it's a column or measure, based on whether it's fully qualified or not.

Связанное содержимое

- Learning path: Use DAX in Power BI Desktop
- Вопросы? Try asking the Power BI Community
- Предложения? Contribute ideas to improve Power BI

DIVIDE функция и оператор деления (/)

Статья • 23.01.2025

В качестве моделиатора данных при написании выражения DAX для деления числителя по знаменателю можно использовать функцию DIVIDE или оператор деления (/ — косую черту).

При использовании функции DIVIDE необходимо передать числовые и знаменательные выражения. При необходимости можно передать значение, представляющее *альтернативный результат*.

DAX

```
DIVIDE(<numerator>, <denominator> [,<alternateresult>])
```

Функция DIVIDE была разработана для автоматического обработки деления на нулевые случаи. Если альтернативный результат не передается, а знаменатель равен нулю или BLANK, функция возвращает BLANK. Когда передан альтернативный результат, он возвращается вместо BLANK.

Функция DIVIDE удобна, так как она сохраняет выражение от необходимости сначала протестировать значение знаменателя. Функция также лучше оптимизирована для тестирования значения знаменателя, чем функция IF. Повышение производительности значительно, так как проверка деления на ноль является затратной. Дальнейшее использование DIVIDE приводит к более краткому и элегантному выражению.

Пример

Следующее выражение меры создает безопасное деление, но включает в себя использование четырех DAX функций.

DAX

```
Profit Margin =
IF(
    OR(
        ISBLANK([Sales]),
        [Sales] == 0
    ),
    BLANK(),
```

```
    [Profit] / [Sales]
)
```

Это выражение меры достигает того же результата, но более эффективно и элегантно.

DAX

```
Profit Margin =
DIVIDE([Profit], [Sales])
```

Рекомендации

Рекомендуется использовать функцию DIVIDE всякий раз, когда знаменатель является выражением, которое *может* возвращать ноль или BLANK.

В случае, если знаменатель является константным значением, рекомендуется использовать оператор деления. В этом случае разделение гарантированно будет успешным, и ваше выражение станет эффективнее, так как это позволит избежать ненужных проверок.

Тщательно рассмотрите, должна ли функция DIVIDE возвращать альтернативное значение. Для мер обычно это лучший дизайн, который они возвращают BLANK. Возврат BLANK лучше, потому что визуальные элементы отчета по умолчанию устраняют группировки при суммировании BLANK. Это позволяет визуальному элементу сосредоточить внимание на группах, где существуют данные. При необходимости можно настроить визуализацию в Power BI для отображения всех групп (возвращающих значения или BLANK) в контексте фильтра, активировав опцию Показывать элементы без данных.

Связанное содержимое

- Схема обучения: Использование DAX в Power BI Desktop
- Вопросы? попробуйте задать вопрос сообществу Power BI
- Предложения? Вносите идеи по улучшению Power BI

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Используйте SELECTEDVALUE вместо VALUES

Статья • 23.01.2025

В качестве моделира данных иногда может потребоваться написать выражение DAX, которое проверяет, фильтруется ли столбец по определенному значению.

В более ранних версиях DAXэто требование было безопасно достигнуто с помощью шаблона с участием трех функций DAX; IF, HASONEVALUE и VALUES. В следующем определении меры представлен пример. Он вычисляет сумму налога на продажу, но только для продаж, сделанных австралийским клиентам.

DAX

```
Australian Sales Tax =
IF(
    HASONEVALUE(Customer[Country-Region]),
    IF(
        VALUES(Customer[Country-Region]) = "Australia",
        [Sales] * 0.10
    )
)
```

В этом примере функция HASONEVALUE возвращает TRUE только в том случае, если в текущем контексте фильтра отображается одно значение столбца "Страна-регион". При TRUEфункция VALUES сравнивается с литеральным текстом "Австралия". Когда функция возвращает , мера продаж умножается на 0,10 (представляющая 10%). Если функция HASONEVALUE возвращает FALSE(так как несколько значений фильтрует столбец), первая функция IF возвращает BLANK.

Использование HASONEVALUE является оборонительным методом. Это необходимо, так как возможно, что несколько значений фильтруют столбец **Страна-Регион**. В этом случае функция VALUES возвращает таблицу нескольких строк. Сравнение таблицы нескольких строк со скалярным значением приводит к ошибке.

Рекомендация

Рекомендуется использовать функцию SELECTEDVALUE. Он достигает того же результата, что и шаблон, описанный в этой статье, еще более эффективно и элегантно.

Используя функцию SELECTEDVALUE, теперь будет перезаписан пример определения меры.

DAX

```
Australian Sales Tax =
IF(
    SELECTEDVALUE(Customer[Country-Region]) = "Australia",
    [Sales] * 0.10
)
```

Совет

Можно передать альтернативное результата в функцию . Альтернативное значение результата возвращается, если к столбцу не применяются фильтры или несколько фильтров.

Связанное содержимое

- Путь обучения: Использование DAX в Power BI Desktop
- Вопросы? попробуйте обратиться к сообществу Power BI
- Предложения? Вносите идеи для улучшения Power BI

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Используйте COUNTROWS вместо COUNT

Статья • 23.01.2025

В качестве моделира данных иногда может потребоваться написать DAX выражение, которое подсчитывает строки таблицы. Таблица может быть таблицей модели или выражением, возвращающим таблицу.

Ваше требование может быть достигнуто двумя способами. Функцию COUNT можно использовать для подсчета значений столбцов или использовать функцию COUNTROWS для подсчета строк таблицы. Обе функции достигнут одного и того же результата, если в столбце нет пустых значений.

В следующем определении меры представлен пример. Он вычисляет количество значений столбцов **OrderDate**.

DAX

```
Sales Orders =
COUNT(Sales[OrderDate])
```

При условии, что степень детализации таблицы **продаж** составляет одну строку для заказа на продажу, а столбец **OrderDate** не содержит пустых значений, вычисляемая метрика вернет правильный результат.

Однако следующее определение меры является лучшим решением.

DAX

```
Sales Orders =
COUNTROWS(Sales)
```

Существует три причины, по которым второе определение меры лучше:

- Это более эффективно, и поэтому он будет работать лучше.
- Он не принимает во внимание BLANK, содержащиеся в любом столбце таблицы.
- Смысл формулы более ясен, до степени самоописания.

Рекомендация

Если вы собираетесь подсчитать строки таблицы, рекомендуется всегда использовать функцию COUNTROWS.

Связанное содержимое

- План обучения: Использование DAX в Power BI Desktop
- Вопросы? попробуйте обратиться к сообществу Power BI
- Предложения? Внесите идеи для улучшения Power BI

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Использование переменных для улучшения формул DAX

Статья • 23.01.2025

Как специалист по моделированию данных, написание и отладка некоторых вычислений на DAX может быть сложной задачей. Обычно сложные требования к вычислению часто включают написание составных или сложных выражений. Составные выражения могут включать использование многих вложенных функций и, возможно, повторное использование логики выражений.

Использование переменных в формулах DAX помогает создавать более сложные и эффективные вычисления. Переменные могут повысить производительность, надежность, удобочитаемость и уменьшить сложность.

В этой статье мы продемонстрируем первые три преимущества, используя пример измерения роста продаж по сравнению с годом (год к году, YoY). (Формула роста продаж YoY — это период продаж, минус продажи за тот же период прошлого года, разделенный на продажи за тот же период прошлого года.)

Начнем со следующего определения меры.

DAX

```
Sales YoY Growth % =
DIVIDE(
    ([Sales] - CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12,
MONTH))),
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))
)
```

Мера создает правильный результат, но теперь давайте посмотрим, как его можно улучшить.

Повышение производительности

Обратите внимание, что формула повторяет выражение, вычисляющее "тот же период прошлого года". Эта формула неэффективна, так как для вычисления одного выражения в два раза требуется Power BI. Определение меры можно сделать более эффективным с помощью переменной, VAR.

Следующее определение меры представляет улучшение. Он использует выражение для назначения результата "того же периода прошлого года"

переменной с именем **SalesPriorYear**. Затем переменная используется дважды в выражении RETURN.

```
DAX
```

```
Sales YoY Growth % =
VAR SalesPriorYear =
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))
RETURN
    DIVIDE(([Sales] - SalesPriorYear), SalesPriorYear)
```

Мера продолжает давать правильный результат и делает это примерно за половину времени, требуемого для запроса.

Улучшение удобочитаемости

В определении предыдущей меры обратите внимание, что выбор имени переменной упрощает выражение RETURN. Выражение является коротким и самоописывающим.

Упрощение отладки

Переменные также могут помочь выполнить отладку формулы. Чтобы проверить выражение, назначенное переменной, временно перезаписываете выражение RETURN, чтобы вывести переменную.

Следующее определение меры возвращает только переменную **SalesPriorYear**. Обратите внимание, как оно закомментирует ожидаемое выражение RETURN. Этот метод позволяет легко вернуть его обратно после завершения отладки.

```
DAX
```

```
Sales YoY Growth % =
VAR SalesPriorYear =
    CALCULATE([Sales], PARALLELPERIOD('Date'[Date], -12, MONTH))
RETURN
    --DIVIDE(([Sales] - SalesPriorYear), SalesPriorYear)
    SalesPriorYear
```

Уменьшение сложности

В более ранних версиях DAXпеременные пока не поддерживаются. Сложные выражения, вносившие новые контексты фильтров, должны были использовать

функции EARLIER или EARLIESTDAX для ссылки на контексты внешнего фильтра. К сожалению, моделиторы данных обнаружили, что эти функции трудно понять и использовать.

Переменные всегда вычисляются вне фильтров, которые применяет выражение RETURN. По этой причине при использовании переменной в измененном контексте фильтра он достигает того же результата, что и функция EARLIEST. Поэтому можно избежать использования функций EARLIER или EARLIEST. Это означает, что теперь можно писать формулы, которые являются менее сложными, и это проще понять.

Рассмотрим следующее определение вычисляемого столбца, добавленное в таблицу подкатегории . Он оценивает ранг для каждой подкатегории продукта на основе значений столбца **Продажи подкатегории**.

DAX

```
Subcategory Sales Rank =
COUNTROWS(
    FILTER(
        Subcategory,
        EARLIER(Subcategory[Subcategory Sales]) < Subcategory[Subcategory
Sales]
    )
) + 1
```

Функция используется для ссылки на значение столбца подкатегории продаж в текущем контексте строки.

Определение вычисляемого столбца можно улучшить с помощью переменной вместо функции EARLIER. Переменная CurrentSubcategorySales сохраняет значение столбца "Продажи " столбца в текущемконтексте строки, а выражение RETURN использует его в измененном контексте фильтра.

DAX

```
Subcategory Sales Rank =
VAR CurrentSubcategorySales = Subcategory[Subcategory Sales]
RETURN
    COUNTROWS(
        FILTER(
            Subcategory,
            CurrentSubcategorySales < Subcategory[Subcategory Sales]
        )
    ) + 1
```

Связанное содержимое

- статья VARDAX
- План обучения: Использование DAX в Power BI Desktop
- Вопросы? попробуйте обратиться к сообществу Power BI

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Справочник по функциям DAX

Статья • 17.03.2025

Справочник по функциям DAX содержит подробные сведения, включая синтаксис, параметры, возвращаемые значения и примеры для каждой из более чем 250 функций, используемых в формулах анализа данных (DAX).

> ⓘ **Важно!**
>
> Не все функции DAX поддерживаются или включены в более ранние версии Power BI Desktop, Analysis Services и Power Pivot в Excel.

В этом разделе

Новые функции DAX . Эти функции являются новыми или являются существующими функциями, которые были значительно обновлены.

функции агрегирования. Эти функции вычисляют (скалярное) значение, например количество, суммирование, среднее, минимальное или максимальное значение для всех строк в столбце или таблице, как определено выражением.

функции даты и времени . Эти функции в DAX похожи на функции даты и времени в Microsoft Excel. Однако DAX функции основаны на типах данных datetime, используемых Microsoft SQL Server.

функции фильтрации. Эти функции помогают возвращать определенные типы данных, искать значения в связанных таблицах и фильтровать по соответствующим значениям. Функции подстановки работают с помощью таблиц и связей между ними. Функции фильтрации позволяют управлять контекстом данных для создания динамических вычислений.

финансовые функции . Эти функции используются в формулах, выполняющих финансовые вычисления, такие как чистая текущая стоимость и ставка прибыли.

функции INFO. Эти функции возвращают метаданные о семантической модели, такие как таблицы, столбцы, связи и вычисления DAX формулах. Они помогут вам понять и задокументируйте модель. Они основаны на библиотеке динамических административных представлений в службах Analysis Services, которые были изменены для работы в качестве функций DAX.

функции сведений. Эти функции смотрят на таблицу или столбец, предоставленные в качестве аргумента другой функции, и возвращает, соответствует ли значение ожидаемому типу. Например, функция ISERROR возвращает `TRUE`, если указанное значение содержит ошибку.

логические функции . Эти функции возвращают сведения о значениях в выражении. Например, функция `TRUE` позволяет узнать, возвращается ли выражение, которое вы оцениваете, возвращает `TRUE` значение.

математические и триг-функции — математические функции в DAX похожи на математические и тригонометрические функции Excel. Однако существуют некоторые различия в числовых типах данных, используемых функциями DAX.

Другие функции. Эти функции выполняют уникальные действия, которые не могут быть определены любой из категорий, к которым относятся большинство других функций.

родительских и дочерних функций. Эти функции помогают пользователям управлять данными, представленными в модели данных в виде родительской или дочерней иерархии.

функции связи. Эти функции предназначены для управления и использования связей между таблицами. Например, можно указать определенную связь, используемую в вычислении.

Статистические функции. Эти функции вычисляют значения, связанные со статистическими распределениями и вероятностью, например стандартное отклонение и количество перемещений.

функции управления таблицами. Эти функции возвращают таблицу или управляют существующими таблицами.

текстовые функции . С помощью этих функций можно возвращать часть строки, искать текст в строке или объединять строковые значения. Дополнительные функции предназначены для управления форматами дат, времени и чисел.

функции аналитики времени. Эти функции помогают создавать вычисления, использующие встроенные знания о календарях и датах. Используя диапазоны времени и даты в сочетании с агрегатами или вычислениями, можно создавать значимые сравнения в сопоставимые периоды времени для продаж, инвентаризации и т. д.

Связанное содержимое

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Новые функции DAX

Статья • 20.05.2025

DAX постоянно улучшается с помощью новых функций и функций для поддержки новых функций. Новые функции и обновления включаются в обновления служб, приложений и инструментов, которые в большинстве случаев являются ежемесячными.

Хотя функции и функциональные возможности обновляются все время, описаны только те обновления, которые имеют видимые и функциональные изменения, предоставляемые пользователям. Здесь показаны новые функции и обновления существующих функций за прошлый год.

> **ⓘ Важно!**
>
> Не все функции поддерживаются во всех версиях Power BI Desktop, Analysis Services и Power Pivot в Excel. Новые и обновленные функции обычно представлены в Power BI Desktop, а затем в службах Analysis Services, Power Pivot в Excel и средствах.

Новые функции

Развернуть таблицу

Функция	Месяц	Описание
LOOKUPWITHTOTALS	Июнь 2025 г.	Используется только в визуальных вычислениях. Найдите значение при применении фильтров. Фильтры, не указанные, не будут выводиться.
ПОИСКА	Июнь 2025 г.	Используется только в визуальных вычислениях. Найдите значение при применении фильтров.
FIRST	Январь 2024 г.	Используется только в визуальных вычислениях. Извлекает значение в визуальной матрице из первой строки оси.
LAST	Январь 2024 г.	Используется только в визуальных вычислениях. Извлекает значение в визуальной матрице из последней строки оси.
NEXT	Январь 2024 г.	Используется только в визуальных вычислениях. Извлекает значение в следующей строке оси в визуальной матрице.
PREVIOUS	Январь 2024 г.	Используется только в визуальных вычислениях. Извлекает значение в предыдущей строке оси в визуальной матрице.
MATCHBY	Май 2023 г.	Определите столбцы, которые используются для сопоставления данных и определения текущей строки в выражении функции

Функция	Месяц	Описание
		окна.
RANK	Апрель 2023 г.	Возвращает рейтинг текущего контекста в указанной секции, отсортированный по указанному порядку.
ROWNUMBER	Апрель 2023 г.	Возвращает уникальное ранжирование текущего контекста в указанной секции, отсортированного по указанному порядку.
LINEST	Февраль 2023 г.	Использует метод "Наименьшие квадраты", чтобы вычислить прямую линию, которая лучше всего подходит для заданных данных.
LINESTX	Февраль 2023 г.	Использует метод "Наименьшие квадраты", чтобы вычислить прямую линию, которая лучше всего подходит для заданных данных. Результат данных из выражений, вычисляемых для каждой строки в таблице.

Функции агрегирования

Статья • 23.01.2025

Функции агрегирования вычисляют (скалярное) значение, например счетчик, сумм, среднее, минимальное или максимальное значение для всех строк в столбце или таблице, как определено выражением.

В этой категории

Развернуть таблицу

Функция	Описание
APPROXIMATEDISTINCTCOUNT	Возвращает *предполагаемое* количество уникальных значений в столбце.
AVERAGE	Возвращает среднее значение (арифметическое значение) всех чисел в столбце.
AVERAGEA	Возвращает среднее значение (арифметическое значение) значений в столбце.
AVERAGEX	Вычисляет среднее (арифметическое среднее) набора выражений, вычисляемых по таблице.
COUNT	Подсчитывает количество строк в указанном столбце, содержащих непустые значения.
COUNTA	Подсчитывает количество строк в указанном столбце, содержащих непустые значения.
COUNTAX	Подсчитывает непустые результаты при оценке результата выражения по таблице.
COUNTBLANK	Подсчитывает количество пустых ячеек в столбце.
COUNTROWS	Подсчитывает количество строк в указанной таблице или в таблице, определенной выражением.
COUNTX	Подсчитывает количество строк, содержащих число или выражение, которое оценивается числом при оценке выражения по таблице.
DISTINCTCOUNT	Подсчитывает количество уникальных значений в столбце.
DISTINCTCOUNTNOBLANK	Подсчитывает количество уникальных значений в столбце.

Функция	Описание
MAX	Возвращает наибольшее числовое значение в столбце или между двумя скалярными выражениями.
MAXA	Возвращает наибольшее значение в столбце.
MAXX	Вычисляет выражение для каждой строки таблицы и возвращает наибольшее числовое значение.
MIN	Возвращает наименьшее числовое значение в столбце или между двумя скалярными выражениями.
MINA	Возвращает наименьшее значение в столбце, включая все логические значения и числа, представленные в виде текста.
MINX	Возвращает наименьшее числовое значение, которое приводит к оценке выражения для каждой строки таблицы.
PRODUCT	Возвращает произведение чисел в столбце.
PRODUCTX	Возвращает продукт выражения, вычисляемого для каждой строки в таблице.
SUM	Добавляет все числа в столбце.
SUMX	Возвращает сумму выражения, вычисляемого для каждой строки в таблице.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

APPROXIMATEDISTINCTCOUNT

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Возвращает *предполагаемое* количество уникальных значений в столбце. Эта функция вызывает соответствующую операцию агрегирования в источнике данных, оптимизированную для производительности запросов, но с незначительной точностью. Эту функцию можно использовать со следующими источниками данных: sql Azure, выделенный пул SQL Azure Synapse, BigQuery, Databricks и Snowflake. Для этой функции требуется режим DirectQuery. Режим импорта и двойной режим хранения не поддерживаются.

Синтаксис

DAX

```
APPROXIMATEDISTINCTCOUNT(<columnName>)
```

Параметры

Развернуть таблицу

Срок	Описание
`column`	Столбец, содержащий значения, которые нужно подсчитать. Это не может быть выражением.

Возвращаемое значение

Приблизительное число уникальных значений в `column`.

Замечания

Единственным аргументом этой функции является столбец. Можно использовать столбцы, содержащие любой тип данных. Если функция не находит строк для подсчета, она возвращает BLANK, в противном случае возвращается количество различных значений.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

AVERAGE

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает среднее значение (арифметическое значение) всех чисел в столбце.

Синтаксис

DAX

```
AVERAGE(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, содержащий числа, для которых требуется среднее значение.

Возвращаемое значение

Возвращает десятичное число, представляющее арифметическое значение чисел в столбце.

Замечания

- Эта функция принимает указанный столбец в качестве аргумента и находит среднее значение в этом столбце. Если вы хотите найти среднее значение выражения, которое вычисляет набор чисел, используйте вместо этого функцию AVERAGEX.
- Ненумерные значения в столбце обрабатываются следующим образом:
  - Если столбец содержит текст, агрегирование не может быть выполнено, а функции возвращают пустые значения.
  - Если столбец содержит логические значения или пустые ячейки, эти значения игнорируются.

    - Ячейки с нулевым значением включаются.

- При среднем значении ячеек следует учитывать разницу между пустой ячейкой и ячейкой, содержащей значение 0 (ноль). Если ячейка содержит 0, она добавляется к сумме чисел, и строка учитывается среди числа строк, используемых в качестве разделителя. Однако если ячейка содержит пустое значение, строка не учитывается.

- Всякий раз, когда нет строк для статистической обработки, функция возвращает пустое значение. Однако если есть строки, но ни одна из них не соответствует указанным критериям, функция возвращает значение 0. Excel также возвращает ноль, если строки не найдены, соответствующие условиям.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула возвращает среднее значение в столбце ExtendedSalesAmount в таблице InternetSales.

```
DAX
= AVERAGE(InternetSales[ExtendedSalesAmount])
```

Связанные функции

Функция AVERAGEX может принимать в качестве аргумента выражение, вычисляемое для каждой строки в таблице. Это позволяет выполнять вычисления, а затем принимать среднее значение вычисляемых значений.

Функция AVERAGEA принимает столбец в качестве аргумента, но в противном случае это похоже на функцию Excel с тем же именем. Используя функцию AVERAGEA, можно вычислить среднее значение столбца, содержащего пустые значения.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

AVERAGEA

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает среднее значение (арифметическое значение) значений в столбце. Обрабатывает текстовые и нечисловые значения.

Синтаксис

DAX

```
AVERAGEA(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, содержащий значения, для которых требуется среднее значение.

Возвращаемое значение

Десятичное число.

Замечания

- Функция AVERAGEA принимает столбец и усредняет числа в нем, но также обрабатывает нечисловые типы данных в соответствии со следующими правилами:
  - Значения, которые оцениваются как `TRUE` считаются 1.
  - Значения, которые оцениваются как `FALSE` считаются 0 (ноль).
  - Значения, содержащие число нечисловых текстов, равное 0 (ноль).
  - Пустой текст ("") подсчитывается как 0 (ноль).
- Если вы не хотите включать логические значения и текстовые представления чисел в ссылку в рамках вычисления, используйте функцию AVERAGE.

- Всякий раз, когда нет строк для статистической обработки, функция возвращает пустое значение. Однако если есть строки, но ни одна из них не соответствует указанным критериям, функция возвращает значение 0. Microsoft Excel также возвращает ноль, если строки не найдены в соответствии с условиями.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере возвращается среднее значение непустых ячеек в столбце со ссылкой, учитывая следующую таблицу. Если вы использовали функцию AVERAGE, среднее значение будет 21/2; с функцией AVERAGEA результатом является 22/5.

Развернуть таблицу

Идентификатор транзакции	Количество	Результат
0000123	1	Подсчитывается как 1
0000124	20	Подсчитывается как 20
0000125	n/a	Подсчитывается как 0
0000126		Подсчитывается как 0
0000126	`TRUE`	Подсчитывается как 1

DAX

```
= AVERAGEA([Amount])
```

Связанное содержимое

функция AVERAGEAVERAGEXСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

AVERAGEX

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Вычисляет среднее (арифметическое среднее) набора выражений, вычисляемых по таблице.

Синтаксис

DAX

```
AVERAGEX(<table>,<expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Имя таблицы или выражение, указывающее таблицу, над которой можно выполнить агрегирование.
`expression`	Выражение со скалярным результатом, которое будет вычисляться для каждой строки таблицы в первом аргументе.

Возвращаемое значение

Десятичное число.

Замечания

- Функция AVERAGEX позволяет оценивать выражения для каждой строки таблицы, а затем принимать результирующий набор значений и вычислять его арифметическое среднее. Поэтому функция принимает таблицу в качестве первого аргумента и выражение в качестве второго аргумента.
- Во всех остальных отношениях AVERAGEX следует тем же правилам, что и AVERAGE. Нельзя включать нечисловые или пустые ячейки. Требуются

аргументы таблицы и выражения.

- Если для статистической обработки нет строк, функция возвращает пустое значение. Если есть строки, но ни одна из них не соответствует указанным критериям, функция возвращает значение 0.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере вычисляется средняя сумма фрахта и налог на каждый заказ в таблице InternetSales, сначала суммируя "Фрахт плюс TaxAmt" в каждой строке, а затем усчитывая эти суммы.

DAX

```
= AVERAGEX(InternetSales, InternetSales[Freight]+ InternetSales[TaxAmt])
```

При использовании нескольких операций в выражении, используемом в качестве второго аргумента, необходимо использовать скобки для управления порядком вычислений. Дополнительные сведения см. в DAX справочнике по синтаксису.

Связанное содержимое

функция AVERAGEAVERAGEACтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUNT

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Подсчитывает количество строк в указанном столбце, содержащих непустые значения.

Синтаксис

DAX

```
COUNT(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, содержащий значения, которые нужно подсчитать.

Возвращаемое значение

Целое число.

Замечания

- Единственным аргументом, допустимым для этой функции, является столбец. Функция COUNT подсчитывает строки, содержащие следующие типы значений:
  - Числа
  - Финики
  - Строки
- Когда функция не находит число строк, она возвращает пустое значение.
- Пустые значения пропускаются. `TRUE` / `FALSE` значения не поддерживаются.

- Если вы хотите оценить столбец `TRUE`/`FALSE` значений, используйте функцию COUNTA.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).
- Рекомендации по использованию COUNTсм. в разделе Использование COUNTROWS вместо COUNT.

Пример

В следующем примере показано, как подсчитать количество значений в столбце ShipDate.

DAX

```
= COUNT([ShipDate])
```

Чтобы подсчитать логические значения или текст, используйте функции COUNTA или COUNTAX.

Связанное содержимое

функция COUNTACOUNTAX функцииCOUNTXСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUNTA

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Подсчитывает количество строк в указанном столбце, содержащих непустые значения.

Синтаксис

DAX

```
COUNTA(<column>)
```

Параметры

Развернуть таблицу

Термин	Определение
`column`	Столбец, содержащий значения, которые нужно подсчитать.

Возвращаемое значение

целое число.

Замечания

- Если функция не находит число строк, функция возвращает пустое значение.
- В отличие от COUNT, COUNTA поддерживает логический тип данных.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере возвращается количество всех строк в таблице `Reseller` с любым типом значения в столбце, где хранятся номера телефонов.

DAX

```
= COUNTA(Reseller[Phone])
```

Связанное содержимое

функция COUNTCOUNTAX функцииCOUNTXСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUNTAX

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Функция COUNTAX подсчитывает непустые результаты при оценке результата выражения по таблице. То есть он работает так же, как и функция COUNTA, но используется для итерации строк в таблице и подсчета строк, в которых указанные выражения приводят к непустой результату.

Синтаксис

```
DAX

COUNTAX(<table>,<expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение, вычисляемого для каждой строки таблицы.

Возвращаемое значение

Целое число.

Замечания

- Как и функция COUNTA, функция COUNTAX подсчитывает ячейки, содержащие любую информацию, включая другие выражения. Например, если столбец содержит выражение, которое вычисляет пустую строку, функция COUNTAX обрабатывает результат как непустый. Обычно функция COUNTAX не подсчитывает пустые ячейки, но в данном случае ячейка содержит формулу, поэтому она подсчитывается.

- Всякий раз, когда функция не находит строк для статистической обработки, функция возвращает пустое значение.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере учитывается количество небланковых строк в столбце Phone, используя таблицу, которая приводит к фильтрации таблицы торгового посредника по `[Status] = Active`.

DAX

```
= COUNTAX(FILTER('Reseller',[Status]="Active"),[Phone])
```

Связанное содержимое

функция COUNTCOUNTA функцииCOUNTXСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUNTBLANK

Статья • 21.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Подсчитывает количество пустых ячеек в столбце.

Синтаксис

DAX

```
COUNTBLANK(<column>)
```

Параметры

⛶ Развернуть таблицу

Срок	Определение
`column`	Столбец, содержащий пустые ячейки для подсчета.

Возвращаемое значение

Целое число. Если строки не найдены, соответствующие условию, возвращаются пустые.

Замечания

- Единственным аргументом, допустимым для этой функции, является столбец. Можно использовать столбцы, содержащие любые типы данных, но учитываются только пустые ячейки. Ячейки с нулевым значением (0) не учитываются, так как ноль считается числовым значением, а не пустым.
- Всякий раз, когда нет строк для статистической обработки, функция возвращает пустое значение. Однако если есть строки, но ни одна из них не соответствует указанным критериям, функция возвращает значение 0. Microsoft Excel также возвращает ноль, если строки не найдены в соответствии с условиями.

- Другими словами, если функция COUNTBLANK не находит пустых, результат будет равен нулю, но если нет строк для проверки, результат будет пустым.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показано, как подсчитать количество строк в табличном торговом посреднике с пустыми значениями для BankName.

```
DAX

= COUNTBLANK(Reseller[BankName])
```

Чтобы подсчитать логические значения или текст, используйте функции COUNTA или COUNTAX.

Связанное содержимое

функция COUNTCOUNTA функцииCOUNTAXCOUNTXстатистических функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUNTROWS

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Функция COUNTROWS подсчитывает количество строк в указанной таблице или в таблице, определенной выражением.

Синтаксис

```DAX
COUNTROWS([<table>])
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	(Необязательно) Имя таблицы, содержащей строки для подсчета, или выражение, возвращающее таблицу. Если он не указан, значение по умолчанию является домашней таблицей текущего выражения.

Возвращаемое значение

Целое число.

Замечания

- Эту функцию можно использовать для подсчета количества строк в базовой таблице, но чаще используется для подсчета количества строк, которые приводят к фильтрации таблицы или применения контекста к таблице.
- Если аргумент таблицы не содержит строк, функция возвращает BLANK.
- Дополнительные сведения об использовании COUNT и COUNTROWSсм. в статье Использование COUNTROWS вместо COUNT в DAX.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

В следующем примере показано, как подсчитать количество строк в таблице Orders. Ожидаемый результат — 52761.

```
DAX

= COUNTROWS('Orders')
```

Пример 2

В следующем примере показано, как использовать COUNTROWS с контекстом строки. В этом сценарии существует два набора данных, связанных по номеру заказа. Торговый посредник таблицы содержит одну строку для каждого торгового посредника; Таблица ResellerSales содержит несколько строк для каждого заказа, каждая строка, содержащая один заказ для определенного торгового посредника. Таблицы связаны связью в столбце Торговых посредников.

Формула получает значение ResellerKey, а затем подсчитывает количество строк в связанной таблице с одинаковым идентификатором торгового посредника. Результат выводится в столбце **CalculatedColumn1**.

```
DAX

= COUNTROWS(RELATEDTABLE(ResellerSales))
```

В следующей таблице показана часть ожидаемых результатов:

Развернуть таблицу

Торговых посредников	CalculatedColumn1
1	73
2	70
3	394

Связанное содержимое

функция COUNTCOUNTA функцииCOUNTAXCOUNTXстатистических функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUNTX

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Подсчитывает количество строк, содержащих непустое значение или выражение, которое вычисляет непустое значение при оценке выражения по таблице.

Синтаксис

DAX

```
COUNTX(<table>,<expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки для подсчета.
`expression`	Выражение, возвращающее набор значений, содержащих значения, которые нужно подсчитать.

Возвращаемое значение

Целое число.

Замечания

- Функция COUNTX принимает два аргумента. Первый аргумент всегда должен быть таблицей или любым выражением, возвращающим таблицу. Второй аргумент — это столбец или выражение, искомые COUNTX.
- Функция COUNTX подсчитывает только значения, даты или строки. Если функция не находит число строк, возвращает пустое значение.

- Если вы хотите подсчитать логические значения, используйте функцию COUNTAX.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

Следующая формула возвращает количество всех строк в таблице Product, имеющих цену на список.

DAX

```
= COUNTX(Product,[ListPrice])
```

Пример 2

В следующей формуле показано, как передать отфильтрованную таблицу в COUNTX для первого аргумента. Формула использует выражение фильтра, чтобы получить только строки в таблице Product, которые соответствуют условию ProductSubCategory = Caps, а затем подсчитывает строки в результирующей таблице, которая имеет цену на список. Выражение FILTER применяется к таблицам Products, но использует значение, которое вы ищете в связанной таблице ProductSubCategory.

DAX

```
=
COUNTX(FILTER(Product,RELATED(ProductSubcategory[EnglishProductSubcategoryNa
me])="Caps"), Product[ListPrice])
```

Связанное содержимое

функция COUNTCOUNTA функцииCOUNTAXСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DISTINCTCOUNT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Подсчитывает количество уникальных значений в столбце.

Синтаксис

DAX

```
DISTINCTCOUNT(<column>)
```

Параметры

Развернуть таблицу

Срок	Описание
`column`	Столбец, содержащий значения, которые нужно подсчитать

Возвращаемое значение

Число различных значений в `column`.

Замечания

- Единственным аргументом, допустимым для этой функции, является столбец. Можно использовать столбцы, содержащие любой тип данных. Если функция не находит строк для подсчета, она возвращает BLANK, в противном случае возвращается количество различных значений.
- функция DISTINCTCOUNT подсчитывает значение BLANK. Чтобы пропустить значение BLANK, используйте функцию DISTINCTCOUNTNOBLANK.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показано, как подсчитать количество отдельных заказов на продажу в столбце ResellerSales_USD[SalesOrderNumber].

```
DAX

= DISTINCTCOUNT(ResellerSales_USD[SalesOrderNumber])
```

Использование приведенной выше меры в таблице с календарным годом в категории сторон и продуктов сверху возвращает следующие результаты:

Развернуть таблицу

Метки строк	Принадлежности	Велосипеды	Одежда	Компоненты	-	Общий итог
2005	135	345	242	205		366
2006	356	850	644	702		1015
2007	531	1234	963	1138		1521
2008	293	724	561	601		894
					1	1
общий	**1315**	**3153**	**2410**	**2646**	**1**	**3797**

В приведенном выше примере обратите внимание, что строки с общими номерами не добавляются, это происходит из-за того, что тот же порядок может содержать элементы строки в одном порядке из разных категорий продуктов.

Связанное содержимое

функция COUNTCOUNTA функцииCOUNTAXCOUNTXстатистических функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DISTINCTCOUNTNOBLANK

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Подсчитывает количество уникальных значений в столбце.

Синтаксис

```DAX
DISTINCTCOUNTNOBLANK(<column>)
```

Параметры

Развернуть таблицу

Срок	Описание
`column`	Столбец, содержащий значения, которые нужно подсчитать

Возвращаемое значение

Число различных значений в `column`.

Замечания

- В отличие от функции DISTINCTCOUNT, DISTINCTCOUNTNOBLANK не учитывает значение BLANK.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показано, как подсчитать количество отдельных заказов на продажу в столбце ResellerSales_USD[SalesOrderNumber].

DAX

```
= DISTINCTCOUNT(ResellerSales_USD[SalesOrderNumber])
```

запрос DAX

DAX

```
EVALUATE
    ROW(
        "DistinctCountNoBlank", DISTINCTCOUNTNOBLANK(DimProduct[EndDate]),
        "DistinctCount", DISTINCTCOUNT(DimProduct[EndDate])
    )
```

Развернуть таблицу

[DistinctCountNoBlank]	[DistinctCount]
2	3

Связанное содержимое

DISTINCTCOUNT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MAX

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает наибольшее значение в столбце или между двумя скалярными выражениями.

Синтаксис

```DAX
MAX(<column>)
```

```DAX
MAX(<expression1>, <expression2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, в котором требуется найти наибольшее значение.
`expression`	Любое DAX выражение, возвращающее одно значение.

Возвращаемое значение

Наибольшее значение.

Замечания

- При сравнении двух выражений пустое обрабатывается как 0 при сравнении. То есть max(1, Blank() возвращает значение 1, а max(-1, Blank() возвращает 0. Если оба аргумента пусты, MAX возвращает пустое значение. Если любое

выражение возвращает значение, которое не допускается, MAX возвращает ошибку.

- `TRUE` / `FALSE` значения не поддерживаются. Если вы хотите оценить столбец `TRUE`/`FALSE` значений, используйте функцию MAXA.

Пример 1

В следующем примере возвращается наибольшее значение, найденное в столбце ExtendedAmount таблицы InternetSales.

```
DAX

= MAX(InternetSales[ExtendedAmount])
```

Пример 2

В следующем примере возвращается наибольшее значение между результатом двух выражений.

```
DAX

= Max([TotalSales], [TotalPurchases])
```

Связанное содержимое

функция MAXAMAXXСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MAXA

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает наибольшее значение в столбце.

Синтаксис

```DAX
MAXA(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, в котором требуется найти наибольшее значение.

Возвращаемое значение

Наибольшее значение.

Замечания

- Функция MAXA принимает в качестве аргумента столбец и ищет наибольшее значение среди следующих типов значений:
  - Числа
  - Финики
- Логические значения, такие как `TRUE` и `FALSE`. Строки, которые оцениваются как `TRUE` число как 1; строки, которые оцениваются как `FALSE` число как 0 (ноль).
- Пустые ячейки игнорируются. Если столбец не содержит значений, которые можно использовать, MAXA возвращает значение 0 (ноль).

- Если вы хотите сравнить текстовые значения, используйте функцию MAX.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

В следующем примере возвращается наибольшее значение из вычисляемого столбца с именем **ResellerMargin**, которое вычисляет разницу между ценой на список и ценой торгового посредника.

DAX

```
= MAXA([ResellerMargin])
```

Пример 2

В следующем примере возвращается наибольшее значение из столбца, содержащего даты и время. Поэтому эта формула получает самую последнюю дату транзакции.

DAX

```
= MAXA([TransactionDate])
```

Связанное содержимое

функция MAXMAXXСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MAXX

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает наибольшее значение, которое приводит к оценке выражения для каждой строки таблицы.

Синтаксис

DAX

```
MAXX(<table>,<expression>,[<variant>])
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение, вычисляемого для каждой строки таблицы.
`variant`	(Необязательно) Если `TRUE` и есть вариант или смешанные типы значений, возвращается наибольшее значение на основе ORDER BYDESC.

Возвращаемое значение

Наибольшее значение.

Замечания

- Аргумент `table` функции MAXX может быть именем таблицы или выражением, вычисляющим таблицу. Второй аргумент указывает выражение для каждой строки таблицы.
- Из значений, которые необходимо оценить, учитываются только следующие значения:

    - Числа
    - Тексты
    - Финики

- Пустые значения пропускаются. `TRUE` / `FALSE` значения не поддерживаются.
- Если выражение имеет вариант или смешанные типы значений, такие как текст и число, по умолчанию MAXX учитывает только числа. Если `<variant> =` TRUE'' возвращается максимальное значение.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

Следующая формула использует выражение в качестве второго аргумента для вычисления общего объема налогов и доставки для каждого заказа в таблице InternetSales. Ожидаемый результат — 375,7184.

DAX

```
= MAXX(InternetSales, InternetSales[TaxAmt]+ InternetSales[Freight])
```

Пример 2

Следующая формула сначала фильтрует таблицу InternetSales с помощью выражения FILTER для возврата подмножества заказов для определенного региона продаж, определенного как [SalesTerritory] = 5. Затем функция MAXX вычисляет выражение, используемое в качестве второго аргумента для каждой строки отфильтрованной таблицы, и возвращает самую высокую сумму для налогов и доставки только для этих заказов. Ожидаемый результат — 250,3724.

DAX

```
= MAXX(FILTER(InternetSales,[SalesTerritoryCode]="5"),
InternetSales[TaxAmt]+ InternetSales[Freight])
```

Связанное содержимое

функция MAXMAXAСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MIN

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает наименьшее значение в столбце или между двумя скалярными выражениями.

Синтаксис

DAX

```
MIN(<column>)
```

DAX

```
MIN(<expression1>, <expression2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, в котором нужно найти наименьшее значение.
`expression`	Любое DAX выражение, возвращающее одно значение.

Возвращаемое значение

Наименьшее значение.

Замечания

- Функция MIN принимает столбец или два выражения в качестве аргумента и возвращает наименьшее значение. Учитываются следующие типы значений в столбцах:
  - Числа

    - Тексты
    - Финики
    - Пробелы

- При сравнении выражений пустое обрабатывается как 0 при сравнении. То есть min(1,Blank() возвращает значение 0, а min(-1, Blank() возвращает -1. Если оба аргумента пусты, MIN возвращает пустое значение. Если любое выражение возвращает значение, которое не допускается, MIN возвращает ошибку.

- `TRUE` / `FALSE` значения не поддерживаются. Если вы хотите оценить столбец `TRUE`/`FALSE` значений, используйте функцию MINA.

Пример 1

В следующем примере возвращается наименьшее значение из вычисляемого столбца ResellerMargin.

```
DAX
= MIN([ResellerMargin])
```

Пример 2

В следующем примере возвращается наименьшее значение из столбца, содержащего даты и время, TransactionDate. Поэтому эта формула возвращает дату, которая является самой ранней.

```
DAX
= MIN([TransactionDate])
```

Пример 3

В следующем примере возвращается наименьшее значение из результата двух скалярных выражений.

```
DAX
= Min([TotalSales], [TotalPurchases])
```

Связанное содержимое

функция MINAMINXСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MINA

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает наименьшее значение в столбце.

Синтаксис

DAX

```
MINA(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, для которого требуется найти минимальное значение.

Возвращаемое значение

Наименьшее значение.

Замечания

- Функция MINA принимает в качестве аргумента столбец, содержащий числа, и определяет наименьшее значение следующим образом:
  - Если столбец не содержит значений, MINA возвращает 0 (ноль).
  - Строки в столбце, вычисляемые логическими значениями, например `TRUE` и `FALSE`, обрабатываются как 1, если `TRUE` и 0 (ноль), если `FALSE`.
  - Пустые ячейки игнорируются.
- Если вы хотите сравнить текстовые значения, используйте функцию MIN.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на

уровне строк (RLS).

Пример 1

Следующее выражение возвращает минимальную плату за груз из таблицы InternetSales.

```
DAX

= MINA(InternetSales[Freight])
```

Пример 2

Следующее выражение возвращает минимальное значение в столбце "Почтовый индекс". Так как тип данных столбца является текстом, функция не находит никаких значений, а формула возвращает ноль (0).

```
DAX

= MINA([PostalCode])
```

Связанное содержимое

функция MINMINXСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MINX

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает наименьшее значение, которое приводит к оценке выражения для каждой строки таблицы.

Синтаксис

DAX

```
MINX(<table>, < expression>,[<variant>])
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение, вычисляемого для каждой строки таблицы.
`variant`	(Необязательно) Если `TRUE` и есть вариант или смешанные типы значений, возвращается наименьшее значение на основе ORDER BYASC.

Возвращаемое значение

Наименьшее значение.

Замечания

- Функция MINX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Второй аргумент содержит выражение, вычисляемое для каждой строки таблицы.
- Пустые значения пропускаются. `TRUE` / `FALSE` значения не поддерживаются.

- Если выражение имеет вариант или смешанные типы значений, такие как текст и число, по умолчанию MINX учитывает только числа. Если `<variant> = TRUE`, возвращается минимальное значение.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

В следующем примере фильтруется таблица InternetSales и возвращаются только строки для определенной территории продаж. Затем формула находит минимальное значение в столбце Freight.

DAX

```
= MINX( FILTER(InternetSales, [SalesTerritoryKey] = 5),[Freight])
```

Пример 2

В следующем примере используется та же отфильтроженная таблица, что и в предыдущем примере, но вместо простого поиска значений в столбце для каждой строки отфильтрованной таблицы функция вычисляет сумму двух столбцов, Freight и TaxAmt и возвращает наименьшее значение, полученное из этого вычисления.

DAX

```
= MINX( FILTER(InternetSales, InternetSales[SalesTerritoryKey] = 5),
InternetSales[Freight] + InternetSales[TaxAmt])
```

В первом примере имена столбцов не определены. Во втором примере имена столбцов являются полными.

Связанное содержимое

функция MINMINAСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ПРОДУКТ

Статья • 14.12.2024

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает произведение чисел в столбце.

Синтаксис

```DAX
PRODUCT(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, содержащий числа, для которых вычисляется продукт.

Возвращаемое значение

Десятичное число.

Замечания

- Чтобы вернуть продукт выражения, вычисляемого для каждой строки таблицы, используйте функцию PRODUCTX .
- Учитываются только числа в столбце. Пустые, логические значения и текст игнорируются. Например

  `PRODUCT( Table[Column] )` эквивалентен `PRODUCTX( Table, Table[Column] )`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Ниже вычислен продукт столбца AdjustedRates в таблице Annuity:

```DAX
= PRODUCT( Annuity[AdjustedRates] )
```

Связанное содержимое

PRODUCTX

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PRODUCTX

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает продукт выражения, вычисляемого для каждой строки в таблице.

Синтаксис

```
DAX

PRODUCTX(<table>, <expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение, вычисляемого для каждой строки таблицы.

Возвращаемое значение

Десятичное число.

Замечания

- Чтобы вернуть произведение чисел в столбце, используйте PRODUCT.
- Функция PRODUCTX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Второй аргумент — это столбец, содержащий числа, для которых требуется вычислить продукт, или выражение, которое вычисляет столбец.
- Учитываются только числа в столбце. Пустые, логические значения и текст игнорируются.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Ниже вычислена будущая стоимость инвестиций:

```
DAX

= [PresentValue] * PRODUCTX( AnnuityPeriods, 1+[FixedInterestRate] )
```

Связанное содержимое

PRODUCT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SUM

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Добавляет все числа в столбце.

Синтаксис

```
DAX
SUM(<column>)
```

Параметры

Развернуть таблицу

Термин	Определение
`column`	Столбец, содержащий числа для суммы.

Возвращаемое значение

Десятичное число.

Замечания

Если вы хотите отфильтровать суммарные значения, можно использовать функцию SUMX и указать выражение для суммирования.

Пример

В следующем примере добавляются все числа, содержащиеся в столбце Amt, из таблицы Sales.

```
DAX
```

```
= SUM(Sales[Amt])
```

Связанное содержимое

SUMX

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SUMX

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает сумму выражения, вычисляемого для каждой строки в таблице.

Синтаксис

DAX

```
SUMX(<table>, <expression>)
```

Параметры

Развернуть таблицу

Термин	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение, вычисляемого для каждой строки таблицы.

Возвращаемое значение

Десятичное число.

Замечания

- Функция SUMX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Второй аргумент — это столбец, содержащий числа, которые нужно суммировать, или выражение, которое вычисляется в столбце.
- — этофункция итератора .
- Учитываются только числа в столбце. Пустые, логические значения и текст игнорируются.

- Более сложные примеры SUMX в формулах см. в ALL и CALCULATETABLE.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере сначала фильтруется таблица, InternetSales, в выражении "InternetSales[SalesTerritoryID] = 5", а затем возвращается сумма всех значений в столбце Freight. Другими словами, выражение возвращает сумму расходов на груз только для указанной области продаж.

DAX

```
= SUMX(FILTER(InternetSales, InternetSales[SalesTerritoryID]=5),[Freight])
```

Если не нужно фильтровать столбец, используйте функцию SUM. Функция SUM похожа на функцию Excel с тем же именем, за исключением того, что она принимает столбец в качестве ссылки.

Связанное содержимое

SUM Статистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Функции даты и времени

Статья • 22.01.2025

Эти функции помогают создавать вычисления на основе дат и времени. Многие функции в DAX похожи на функции даты и времени Excel. Однако функции DAX используют тип данных `datetime` и могут принимать значения из столбца в качестве аргумента.

В этой категории

Развернуть таблицу

Функция	Описание
CALENDAR	Возвращает таблицу с одним столбцом с именем Date, который содержит непрерывный набор дат.
CALENDARAUTO	Возвращает таблицу с одним столбцом с именем Date, который содержит непрерывный набор дат.
DATE	Возвращает указанную дату в формате datetime.
DATEDIFF	Возвращает количество границ интервала между двумя датами.
DATEVALUE	Преобразует дату в виде текста в дату в формате datetime.
DAY	Возвращает день месяца, число от 1 до 31.
EDATE	Возвращает дату, которая является указанным числом месяцев до или после даты начала.
EOMONTH	Возвращает дату в формате даты и времени последнего дня месяца до или после указанного числа месяцев.
HOUR	Возвращает час в виде числа от 0 (12:00 утра) до 23 (11:00 вечера).
MINUTE	Возвращает минуту в виде числа от 0 до 59, учитывая значение даты и времени.
MONTH	Возвращает месяц как число от 1 (январь) до 12 (декабрь).
NETWORKDAYS	Возвращает количество целых рабочих дней между двумя датами.
NOW	Возвращает текущую дату и время в формате datetime.
QUARTER	Возвращает четверть в виде числа от 1 до 4.

Функция	Описание
SECOND	Возвращает секунды значения времени в виде числа от 0 до 59.
TIME	Преобразует часы, минуты и секунды в виде чисел в формат даты и времени.
TIMEVALUE	Преобразует время в текстовом формате в время в формате datetime.
TODAY	Возвращает текущую дату.
UTCNOW	Возвращает текущую дату и время в формате UTC
UTCTODAY	Возвращает текущую дату UTC.
WEEKDAY	Возвращает число от 1 до 7, определяющее день недели даты.
WEEKNUM	Возвращает число недели для заданной даты и года в соответствии с значением return_type.
YEAR	Возвращает год даты в виде четырехзначного целого числа в диапазоне 1900-9999.
YEARFRAC	Вычисляет долю года, представленную числом целых дней между двумя датами.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CALENDAR

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу с одним столбцом с именем Date, который содержит непрерывный набор дат. Диапазон дат — от указанной даты начала до указанной даты окончания, включительно с этими двумя датами.

Синтаксис

```DAX
CALENDAR(<start_date>, <end_date>)
```

Параметры

Развернуть таблицу

Срок	Определение
`start_date`	Любое выражение DAX, возвращающее значение даты и времени.
`end_date`	Любое выражение DAX, возвращающее значение даты и времени.

Возвращаемое значение

Возвращает таблицу с одним столбцом с именем Date, содержащим непрерывный набор дат. Диапазон дат — от указанной даты начала до указанной даты окончания, включительно с этими двумя датами.

Замечания

- Ошибка возвращается, если start_date больше end_date.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Следующая формула возвращает таблицу с датами от 1 января 2015 г. до 31 декабря 2021 г.

DAX

```
= CALENDAR (DATE (2015, 1, 1), DATE (2021, 12, 31))
```

Для модели данных, которая включает фактические данные о продажах и будущие прогнозы продаж, следующее выражение возвращает таблицу дат, охватывающую диапазон дат в таблицах "Продажи" и "Прогноз".

DAX

```
= CALENDAR (MINX (Sales, [Date]), MAXX (Forecast, [Date]))
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CALENDARAUTO

01.07.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу с одним столбцом с именем Date, который содержит непрерывный набор дат. Диапазон дат вычисляется автоматически на основе данных в модели.

Синтаксис

```DAX
CALENDARAUTO([fiscal_year_end_month])
```

Параметры

Развернуть таблицу

Срок	Определение
`fiscal_year_end_month`	Любое выражение DAX, которое возвращает целое число от 1 до 12. Если опущен, значение по умолчанию указано в шаблоне таблицы календаря для текущего пользователя, если оно присутствует; в противном случае значение по умолчанию — 12.

Возвращаемое значение

Возвращает таблицу с одним столбцом с именем Date, который содержит непрерывный набор дат. Диапазон дат вычисляется автоматически на основе данных в модели.

Замечания

- Диапазон дат вычисляется следующим образом:

    - Самая ранняя дата в модели, которая не находится в вычисляемом столбце или вычисляемой таблице, принимается как MinDate.
    - Последняя дата в модели, которая не находится в вычисляемом столбце или вычисляемой таблице, принимается как MaxDate.
    - Возвращаемый диапазон дат — это даты между началом финансового года, связанного с MinDate и окончанием финансового года, связанного с MaxDate.

- Ошибка возвращается, если модель не содержит значения даты и времени, которые не находятся в вычисляемых столбцах или вычисляемых таблицах.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В этом примере minDate и MaxDate в модели данных — 1 июля 2018 г. и 30 июня 2019 г. Эта модель содержит две вычисляемые таблицы:

- **CalendarAuto**. Определен как `CALENDARAUTO()`. Возвращает все даты между 1 января 2018 г. и 31 декабря 2019 г.
- **CalendarAuto3**. Определен как `CALENDARAUTO(3)`. Возвращает все даты между 1 апреля 2018 г. и 31 марта 2020 г. Как и в этом примере `fiscal_year_end_month` , 3, первые годы начинаются 1 апреля и заканчивается 31 марта. В результате диапазон определяется путем выбора первого дня финансового года или до года MinDate (1 апреля 2018 г.) и последнего дня финансового года после или в течение года MaxDate (31 марта 2020 г.).

Ниже приведены определения мер и их возвращаемые значения в примере модели:

DAX

```
MinDate = MIN('Date'[Date]) //returns July 1, 2018
MaxDate = MAX('Date'[Date]) //returns June 30, 2019
MinCalendarAuto = MIN('CalendarAuto'[Date]) //returns January 1, 2018
MaxCalendarAuto = MAX('CalendarAuto'[Date]) //returns December 31, 2019
MinCalendarAuto3 = MIN('CalendarAuto3'[Date]) //returns April 1, 2018
MaxCalendarAuto3 = MAX('CalendarAuto3'[Date]) //returns March 31, 2020
```

DATE

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает указанную дату в формате `datetime`.

Синтаксис

DAX

```
DATE(<year>, <month>, <day>)
```

Параметры

Развернуть таблицу

Срок	Определение
`year`	Число, представляющее год. Значение аргумента `year` может содержать одну до четырех цифр. Аргумент `year` интерпретируется в соответствии с системой дат, используемой компьютером. Поддерживаются даты, начиная с 1 марта 1900 г. Если ввести число с десятичным числом, число округляется. Для значений, превышающих 9999 или меньше нуля (отрицательные значения), функция возвращает ошибку `#VALUE!`. Если значение `year` составляет от 0 до 1899, значение добавляется к 1900, чтобы получить окончательное значение. См. приведенные ниже примеры. **Примечание.** следует использовать четыре цифры для аргумента `year` всякий раз, чтобы предотвратить нежелательные результаты. Например, при использовании 07 возвращается значение 1907 в качестве значения года.
`month`	Число, представляющее месяц или вычисление в соответствии со следующими правилами: Отрицательные целые числа не поддерживаются. Допустимые значения : 1–12. Если `month` число от 1 до 12, то оно представляет месяц года. 1 представляет 2

Срок	Определение
	января и т. д. до 12 декабря. Если ввести целое число, превышающее 12, происходит следующее вычисление: дата вычисляется путем добавления значения `month` в `year`. Например, если у вас есть DATE(2008, 18, 1), функция возвращает значение даты и времени, эквивалентное 1 июня 2009 года, так как 18 месяцев добавляются в начало 2008 года, что дает значение 2009 года. См. приведенные ниже примеры.
`day`	Число, представляющее день или вычисление в соответствии со следующими правилами: Отрицательные целые числа не поддерживаются. Допустимые значения : 1–31. Если `day` является числом от 1 до последнего дня данного месяца, он представляет день месяца. Если ввести целое число, превышающее последний день данного месяца, происходит следующее вычисление: дата вычисляется путем добавления значения `day` в `month`. Например, в формуле `DATE( 2008, 3, 32)` функция DATE возвращает значение `datetime`, эквивалентное 1 апреля 2008 года, так как 32 дня добавляются в начало марта, что дает значение 1 апреля. Если `day` содержит десятичную часть, округляется до ближайшего целочисленного значения.

Возвращаемое значение

Возвращает указанную дату (`datetime`).

Замечания

- Функция DATE принимает целые числа, входные в качестве аргументов, и создает соответствующую дату. Функция DATE наиболее полезна в ситуациях, когда год, месяц и день предоставляются формулами. Например, базовые данные могут содержать даты в формате, который не распознается как дата, например YYYYMMDDD. Функцию DATE можно использовать вместе с другими функциями для преобразования дат в число, которое можно распознать как дату.
- В отличие от Microsoft Excel, в котором хранятся даты в виде серийного номера, функции даты DAX всегда возвращают тип данных `datetime`. Однако при необходимости можно использовать форматирование для отображения дат в качестве серийных номеров.

- Дату и дату также можно указать в виде литерала в формате `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` или `dt"YYYY-MM-DD hh:mm:ss"`. При указании в качестве литерала использование функции DATE в выражении не требуется. Дополнительные сведения см. в DAX синтаксисе | Дата и время.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Простая дата

Следующая формула возвращает дату 8 июля 2009 г.

DAX

```
= DATE(2009,7,8)
```

Годы до 1899

Если значение, введенное для аргумента `year`, составляет от 0 до 1899 (включительно), это значение добавляется в 1900, чтобы вычислить год. Следующая формула возвращается 2 января 1908 г. (1900+08).

DAX

```
= DATE(08,1,2)
```

Годы после 1899

Если `year` составляет от 1900 до 9999 (включительно), это значение используется в качестве года. Следующая формула возвращается 2 января 2008 г.

DAX

```
= DATE(2008,1,2)
```

Месяцы

Если `month` больше 12, `month` добавляет это количество месяцев к первому месяцу в указанном году. Следующая формула возвращает дату 2 февраля 2009 г.

DAX

```
= DATE(2008,14,2)
```

Дни недели

Если `day` больше числа дней в указанном месяце, `day` добавляет это количество дней в первый день месяца. Следующая формула возвращает дату 4 февраля 2008 г.

DAX

```
= DATE(2008,1,35)
```

Связанное содержимое

функции даты и времениDAY функцииTODAY

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DATEDIFF

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает количество границ интервала между двумя датами.

Синтаксис

```DAX
DATEDIFF(<Date1>, <Date2>, <Interval>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Date1`	Скалярное значение datetime.
`Date2`	Скалярное значение datetime.
`Interval`	Интервал, используемый при сравнении дат. Значение может быть одним из следующих значений: - SECOND - MINUTE - HOUR - DAY -НЕДЕЛЯ - MONTH - QUARTER - YEAR

Возвращаемое значение

Количество границ интервала между двумя датами.

Замечания

Положительный результат возвращается, если date2 больше даты 1.
Отрицательный результат возвращается, если дата1 больше даты 2.

Пример

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

Следующий запрос DAX:

DAX

```
EVALUATE
VAR StartDate =  DATE ( 2019, 07, 01 )
VAR EndDate =    DATE ( 2021, 12, 31 )
RETURN
    {
        ( "Year",     DATEDIFF ( StartDate, EndDate, YEAR ) ),
        ( "Quarter",  DATEDIFF ( StartDate, EndDate, QUARTER ) ),
        ( "Month",    DATEDIFF ( StartDate, EndDate, MONTH ) ),
        ( "Week",     DATEDIFF ( StartDate, EndDate, WEEK ) ),
        ( "Day",      DATEDIFF ( StartDate, EndDate, DAY ) )
    }
```

Возвращает следующее:

Развернуть таблицу

Значение1	Значение2
Год	2
Четверть	9
Месяц	29
Неделя	130
День	914

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DATEVALUE

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Преобразует дату в текстовом формате в дату в формате datetime.

Синтаксис

```DAX
DATEVALUE(date_text)
```

Параметры

Развернуть таблицу

Срок	Определение
`date_text`	Текст, представляющий дату.

Значение свойства и возвращаемое значение

Дата в формате `datetime`.

Замечания

- При преобразовании DATEVALUE использует параметры языкового стандарта и даты и времени модели для определения значения даты. Если параметры даты и времени модели представляют даты в формате "Месяц/день/год", строка "1/8/2009" преобразуется в значение `datetime` эквивалентно 8 января 2009 года. Однако если параметры даты и времени модели представляют даты в формате Day/Month/Year, то та же строка преобразуется как значение `datetime` эквивалентно 1 августа 2009 года.

- Если преобразование с использованием языковых стандартов и параметров даты и времени модели завершается ошибкой, DATEVALUE попытается использовать другие форматы дат. В этом случае некоторые строки могут быть преобразованы с помощью одного формата, а другие строки преобразуются с помощью другого формата. Например, "5/4/2018" может преобразоваться в 4 мая 2018 года, а "20/4/2018" может преобразоваться в 20 апреля.

- Если часть года `date_text` аргумента опущена, функция DATEVALUE использует текущий год из встроенных часов компьютера. Сведения о времени в аргументе `date_text` игнорируются.

- Языковой стандарт модели и параметры времени и данных изначально определяются приложением и компьютером при создании модели.

- Дату и дату также можно указать в виде литерала в формате `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` или `dt"YYYY-MM-DD hh:mm:ss"`. При указании в качестве литерала использование функции DATEVALUE в выражении не требуется. Дополнительные сведения см. в DAX синтаксисе | Дата и время.

Пример

В следующем примере возвращается другое `datetime` значение в зависимости от языкового стандарта модели и параметров для представления дат и времени.

- В параметрах даты и времени, в которых день предшествует месяцу, в примере возвращается значение `datetime`, соответствующее 8 января 2009 года.

- В параметрах даты и времени, в которых месяц предшествует дню, в примере возвращается значение `datetime`, соответствующее 1 августа 2009 года.

DAX

```
= DATEVALUE("8/1/2009")
```

Связанное содержимое

функции даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DAY

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает день месяца, число от 1 до 31.

Синтаксис

DAX

```
DAY(<date>)
```

Параметры

Развернуть таблицу

Срок	Определение
`date`	Дата в формате `datetime` или текстовое представление даты.

Возвращаемое значение

Целочисленное число, указывающее день месяца.

Замечания

- Функция DAY принимает в качестве аргумента дату дня, который вы пытаетесь найти. Даты можно предоставить функции с помощью другой функции даты, используя выражение, которое возвращает дату или введите дату в формате `datetime`. Можно также ввести дату в одном из принятых форматов строк для дат.
- Значения, возвращаемые YEAR, MONTH и DAY функции будут григорианскими значениями независимо от формата отображения указанного значения даты. Например, если формат отображения указанной даты — Хиджри, возвращаемые значения для YEAR, MONTH и DAY функции будут значениями, связанными с эквивалентной датой Григориана.

- Если аргумент даты является текстовым представлением даты, функция day использует параметры языкового стандарта и даты и времени клиентского компьютера для понимания текстового значения для выполнения преобразования. Если текущие параметры даты и времени представляют даты в формате "Месяц/день/год", строка "1/8/2009", интерпретируется как значение `datetime` эквивалентно 8 января 2009 года, а функция возвращает 8. Однако если текущие параметры даты и времени представляют даты в формате Day/Month/Year, то та же строка будет интерпретирована как значение `datetime` эквивалентно 1 августа 2009 года, а функция возвращает 1.

Пример. Получение дня из столбца даты

Следующая формула возвращает день с даты в столбце [Birthdate].

DAX

```
= DAY([Birthdate])
```

Пример. Получение дня из строковой даты

Следующие формулы возвращают день, 4, используя даты, которые были предоставлены в виде строк в принятом текстовом формате.

DAX

```
= DAY("3-4-1007")
= DAY("March 4 2007")
```

Пример. Использование значения дня в качестве условия

Следующее выражение возвращает день размещения каждого заказа на продажу и помечает строку как рекламный элемент продажи, если заказ был помещен на 10-й месяц.

DAX

```
= IF( DAY([SalesDate])=10,"promotion","")
```

Связанное содержимое

функции даты и времениTODAY функцииDATE

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EDATE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает дату, которая является указанным числом месяцев до или после даты начала. Используйте EDATE для вычисления дат зрелости или дат выполнения, которые падают в тот же день месяца, что и дата выдачи.

Синтаксис

```DAX
EDATE(<start_date>, <months>)
```

Параметры

Развернуть таблицу

Срок	Определение
`start_date`	Дата в формате `datetime` или `text`, представляющая дату начала.
`months`	Целое число, представляющее количество месяцев до или после `start_date`.

Возвращаемое значение

Дата (`datetime`).

Замечания

- В отличие от Microsoft Excel, который сохраняет даты в виде последовательных серийных номеров, DAX работает с датами в формате `datetime`. Даты, хранящиеся в других форматах, преобразуются неявно.
- Если `start_date` не является допустимой датой, EDATE возвращает ошибку. Убедитесь, что ссылка на столбец или дата, указанная в качестве первого аргумента, является датой.

- Если `months` не является целым числом, оно усечено.
- Если аргумент даты является текстовым представлением даты, функция EDATE использует параметры языкового стандарта и даты клиентского компьютера для понимания текстового значения для выполнения преобразования. Если текущие параметры времени даты представляют дату в формате "Месяц/день/год", то следующая строка "1/8/2009" интерпретируется как значение даты и времени, эквивалентное 8 января 2009 года. Однако если текущие параметры даты представляют дату в формате Day/Month/Year, то та же строка будет интерпретирована как значение даты и времени, эквивалентное 1 августа 2009 года.
- Если запрошенная дата истекла за последний день соответствующего месяца, возвращается последний день месяца. Например, следующие функции: EDATE("2009-01-29", 1), EDATE("2009-01-30", 1), EDATE("2009-01-31", 1) возвращается 28 февраля 2009 года; соответствует одному месяцу после даты начала.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере возвращается дата трех месяцев после даты заказа, которая хранится в столбце [TransactionDate].

DAX

```
= EDATE([TransactionDate],3)
```

Связанное содержимое

функции EOMONTHфункции даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EOMONTH

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает дату в формате `datetime` последнего дня месяца, до или после указанного числа месяцев. Используйте EOMONTH для вычисления дат зрелости или дат выполнения, которые падают на последний день месяца.

Синтаксис

DAX

```
EOMONTH(<start_date>, <months>)
```

Параметры

Развернуть таблицу

Срок	Определение
`start_date`	Дата начала в формате `datetime` или в принятом текстовом представлении даты.
`months`	Число, представляющее количество месяцев до или после `start_date`. **Примечание.** Если ввести число, которое не является целым числом, число округляется вверх или вниз до ближайшего целого числа.

Возвращаемое значение

Дата (`datetime`).

Замечания

- В отличие от Microsoft Excel, который сохраняет даты в виде последовательных серийных номеров, DAX работает с датами в формате `datetime`. Функция EOMONTH может принимать даты в других форматах со следующими ограничениями:

- Если `start_date` не является допустимой датой, EOMONTH возвращает ошибку.
- Если `start_date` является числовым значением, которое не находится в формате `datetime`, EOMONTH преобразует число в дату. Чтобы избежать непредвиденных результатов, преобразуйте число в формат `datetime` перед использованием функции EOMONTH.
- Если `start_date` плюс месяцы дают недопустимую дату, EOMONTH возвращает ошибку. Даты до 1 марта 1900 года и после 31 декабря 9999 года недопустимы.
- Если аргумент даты является текстовым представлением даты, функция EDATE использует параметры языкового стандарта и даты на клиентском компьютере, чтобы понять текстовое значение для выполнения преобразования. Если текущие параметры времени даты представляют дату в формате месяца или дня или года, то следующая строка "1/8/2009" интерпретируется как значение даты и времени, эквивалентное 8 января 2009 года. Однако если текущие параметры даты представляют дату в формате Day/Month/Year, то та же строка будет интерпретирована как значение даты и времени, эквивалентное 1 августа 2009 года.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующее выражение возвращается 31 мая 2008 года, так как аргумент `months` округляется до 2.

DAX

```
= EOMONTH("March 3, 2008",1.5)
```

Связанное содержимое

функции EDATEфункции даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

HOUR

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает час в виде числа от 0 (12:00 утра) до 23 (11:00 вечера).

Синтаксис

DAX

```
HOUR(<datetime>)
```

Параметры

Развернуть таблицу

Срок	Определение
`datetime`	Значение `datetime`, например 16:48:00 или 4:48 вечера.

Возвращаемое значение

Целочисленное число от 0 до 23.

Замечания

- Функция HOUR принимает в качестве аргумента время, содержащее час поиска. Вы можете указать время с помощью функции даты и времени, выражения, возвращающего `datetime`, или путем ввода значения непосредственно в одном из принятых форматов времени. Также можно ввести время как любое принятое текстовое представление времени.

- Если аргумент `datetime` является текстовым представлением даты и времени, функция использует параметры языкового стандарта и даты и времени клиентского компьютера для понимания текстового значения для выполнения преобразования. Большинство языковых стандартов используют двоеточие (:) в качестве разделителя времени и любой входной текст с помощью

двоеточий в качестве разделителей времени будет правильно анализироваться. Просмотрите параметры языкового стандарта, чтобы понять результаты.

Пример 1

В следующем примере возвращается час из столбца **TransactionTime** таблицы с именем **Orders**.

```
DAX
= HOUR('Orders'[TransactionTime])
```

Пример 2

В следующем примере возвращается 15 часов, что означает час, соответствующий 3 ВЕЧЕРА в 24-часовом часовом режиме. Текстовое значение автоматически анализируется и преобразуется в значение даты и времени.

```
DAX
= HOUR("March 3, 2008 3:00 PM")
```

Связанное содержимое

функции функции функции даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MINUTE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает минуту в виде числа от 0 до 59, учитывая значение даты и времени.

Синтаксис

DAX

```
MINUTE(<datetime>)
```

Параметры

Развернуть таблицу

Срок	Определение
`datetime`	Значение `datetime` или текст в принятом формате времени, например 16:48:00 или 4:48 вечера.

Возвращаемое значение

Целочисленное число от 0 до 59.

Замечания

- В отличие от Microsoft Excel, в котором хранятся даты и время в последовательном числовом формате, DAX использует тип данных `datetime` для дат и времени. Вы можете указать значение `datetime` функции MINUTE путем ссылки на столбец, в который хранятся даты и время, с помощью функции даты и времени или с помощью выражения, возвращающего дату и время.
- Если аргумент `datetime` является текстовым представлением даты и времени, функция использует параметры языкового стандарта и даты и времени клиентского компьютера для понимания текстового значения для выполнения

преобразования. Большинство языковых стандартов используют двоеточие (:) в качестве разделителя времени и любой входной текст с помощью двоеточий в качестве разделителей времени будет правильно анализироваться. Проверьте параметры языкового стандарта, чтобы понять результаты.

Пример 1

В следующем примере возвращается минута из значения, хранящегося в столбце **TransactionTime** таблицы **Orders**.

DAX

```
= MINUTE(Orders[TransactionTime])
```

Пример 2

В следующем примере возвращается значение 45, которое является числом минут во время 1:45 вечера.

DAX

```
= MINUTE("March 23, 2008 1:45 PM")
```

Связанное содержимое

функции функции функции даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MONTH

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает месяц как число от 1 (январь) до 12 (декабрь).

Синтаксис

```
DAX
MONTH(<datetime>)
```

Параметры

Развернуть таблицу

Срок	Определение
`date`	Дата в `datetime` или текстовом формате.

Возвращаемое значение

Целочисленное число от 1 до 12.

Замечания

- В отличие от Microsoft Excel, который сохраняет даты в качестве серийных номеров, DAX использует формат `datetime` при работе с датами. Можно ввести дату, используемую в качестве аргумента для функции MONTH, введя принятый формат `datetime`, указав ссылку на столбец, содержащий даты, или с помощью выражения, возвращающего дату.
- Значения, возвращаемые YEAR, MONTH и DAY функции будут григорианскими значениями независимо от формата отображения указанного значения даты. Например, если формат отображения указанной даты — Хиджри, возвращаемые значения для YEAR, MONTH и DAY функции будут значениями, связанными с эквивалентной датой Григориана.

- Если аргумент даты является текстовым представлением даты, функция использует параметры языкового стандарта и даты клиентского компьютера для понимания текстового значения для выполнения преобразования. Если текущие параметры времени даты представляют дату в формате месяца или дня или года, то следующая строка "1/8/2009" интерпретируется как значение даты и времени, эквивалентное 8 января 2009 года, и функция возвращает результат 1. Однако если текущие параметры времени даты представляют дату в формате Day/Month/Year, то та же строка будет интерпретирована как значение даты и времени, эквивалентное 1 августа 2009 года, и функция дает результат 8.
- Если текстовое представление даты невозможно правильно преобразовать в значение datetime, функция возвращает ошибку.

Пример 1

Следующее выражение возвращает значение 3, которое является целым числом, соответствующим марту, месяцу в аргументе `date`.

```
DAX

= MONTH("March 3, 2008 3:45 PM")
```

Пример 2

Следующее выражение возвращает месяц из даты в столбце `TransactionDate` таблицы `Orders`.

```
DAX

= MONTH(Orders[TransactionDate])
```

Связанное содержимое

функции функции даты и времени функции функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NETWORKDAYS

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает количество целых рабочих дней между двумя датами (включительно). Параметры определяют, сколько дней в выходные дни. Выходные дни и дни, указанные как праздники, не считаются рабочими днями.

Синтаксис

DAX

```
NETWORKDAYS(<start_date>, <end_date>[, <weekend>, <holidays>])
```

Параметры

Развернуть таблицу

Срок	Определение
`start_date`	Дата, представляющая дату начала. Даты, для которых необходимо вычислить разницу. Start_date может быть выше, чем end_date.
`end_date`	Дата, представляющая дату окончания. Даты, для которых необходимо вычислить разницу. Start_date может быть выше, чем end_date.
`weekend`	Указывает дни недели, которые являются выходными днями и не включаются в число целых рабочих дней между start_date и end_date. Выходные — это число выходных, указывающее, когда происходят выходные дни. значения в выходные дни указывают следующие выходные дни: 1 или опущены: суббота, воскресенье 2: воскресенье, Понедельник 3: понедельник, вторник 4: вторник, среда 5: среда, четверг 6: четверг, пятница 7: пятница, суббота 11: воскресенье только 12: понедельник только 13: вторник только 14: среда только 15: Четверг только

Срок	Определение
	16: пятница только 17: только суббота
`holidays`	Таблица столбцов одной или нескольких дат, которые должны быть исключены из календаря рабочего дня.

Возвращаемое значение

Целое число целых рабочих дней.

Замечания

- Эта функция DAX аналогична функции Excel NETWORKDAYS. Функции INTL и NETWORKDAYS.
- Если start_date и end_date оба BLANK, выходное значение также BLANK.
- Если start_date или end_date BLANK, BLANK start_date или end_date будет рассматриваться как Дата(1899, 12, 30).
- Даты должны быть указаны с помощью функции DATE или в результате другого выражения. Например, `DATE ( 2022, 5, 30 )`, указывает 30 мая 2022 г. Даты также можно указать в виде литерала в формате `(dt”2022-05-30”)`. Не указывайте даты в виде текста.

Пример

Следующее выражение:

DAX

```
= NETWORKDAYS (
    DATE ( 2022, 5, 28 ),
    DATE ( 2022, 5, 30 ),
    1,
    {
        DATE ( 2022, 5, 30 )
    }
)
```

Возвращает:

Развернуть таблицу

[значение]
0

В этом примере возвращается значение 0, так как дата начала — суббота, а дата окончания — понедельник. Параметр выходных указывает, что выходные — суббота и воскресенье, поэтому они не рабочие дни. Параметр праздника помечает 30-ю (дату окончания) как праздник, поэтому рабочие дни не остаются.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NOW

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает текущую дату и время в формате `datetime`.

Функция NOW полезна, если необходимо отобразить текущую дату и время на листе или вычислить значение на основе текущей даты и времени, а также обновлять это значение при каждом открытии листа.

Синтаксис

```DAX
NOW()
```

Возвращаемое значение

Дата (`datetime`).

Замечания

- Результат функции `NOW` изменяется только при обновлении столбца, содержащего формулу. Он не обновляется постоянно.
- В службе Power BI результат функции `NOW` всегда находится в часовом поясе UTC.
- Функция `TODAY` возвращает ту же дату, но не является точной в отношении времени; Возвращаемое время всегда равно 12:00:00 и обновляется только дата.

Пример

В следующем примере возвращается текущая дата и время плюс 3,5 дня:

```DAX
```

```
= NOW()+3.5
```

Связанное содержимое

функция UTCNOWTODAY

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

QUARTER

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает четверть как число с 1 (январь – март) до 4 (октябрь – декабрь).

Синтаксис

```DAX
QUARTER(<date>)
```

Параметры

Развернуть таблицу

Срок	Определение
`date`	Дата.

Возвращаемое значение

Целочисленное число от 1 до 4.

Замечания

Если входное значение равно BLANK, выходное значение также BLANK.

Пример 1

Следующий запрос DAX:

```DAX
EVALUATE { QUARTER(DATE(2019, 2, 1)), QUARTER(DATE(2018, 12, 31)) }
```

Возвращает:

Развернуть таблицу

[значение]
1
4

Пример 2

Следующий запрос DAX:

DAX

```
EVALUATE
ADDCOLUMNS(
    FILTER(
        VALUES(
            FactInternetSales[OrderDate]),
            [OrderDate] >= DATE(2008, 3, 31) && [OrderDate] <= DATE(2008, 4,
1)
        ),
    "Quarter", QUARTER([OrderDate])
)
```

Возвращает:

Развернуть таблицу

FactInternetSales[OrderDate]	[Квартал]
3/31/2008	1
4/1/2008	2

Связанное содержимое

YEAR MONTH DAY

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SECOND

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает секунды значения времени в виде числа от 0 до 59.

Синтаксис

```
DAX

SECOND(<time>)
```

Параметры

Развернуть таблицу

Термин	Определение
`time`	Время в формате `datetime`, например 16:48:23 или 48:48:47.

Возвращаемое значение

Целочисленное число от 0 до 59.

Замечания

- В отличие от Microsoft Excel, в котором хранятся даты и время в виде серийных номеров, DAX использует формат `datetime` при работе с датами и временем. Если исходные данные не в этом формате, DAX неявно преобразует данные. Форматирование можно использовать для отображения дат и времени в качестве последовательного номера.
- Значение даты и времени, которое вы предоставляете в качестве аргумента функции SECOND, можно ввести в виде текстовой строки в кавычках (например, "6:45 PM"). Можно также указать значение времени в результате другого выражения или в качестве ссылки на столбец, содержащий время.

- Если вы предоставляете числовое значение другого типа данных, например 13.60, значение интерпретируется как серийный номер и представляется как тип данных `datetime` перед извлечением значения в секунды. Чтобы упростить понимание результатов, может потребоваться представить такие числа, как даты, прежде чем использовать их в функции SECOND. Например, если вы используете SECOND с столбцом, содержащим числовое значение, например, **25,56**, формула возвращает 24. Это связано с тем, что при форматировании в качестве даты значение 25,56 эквивалентно 25 января 1900 г., 1:26:24 вечера.

- Если аргумент `time` представляет собой текстовое представление даты и времени, функция использует параметры языкового стандарта и даты и времени клиентского компьютера для понимания текстового значения для выполнения преобразования. Большинство языковых стандартов используют двоеточие (:) в качестве разделителя времени и любой входной текст с помощью двоеточий в качестве разделителей времени будет правильно анализироваться. Просмотрите параметры языкового стандарта, чтобы понять результаты.

Пример 1

Следующая формула возвращает количество секунд в течение времени, содержащегося в столбце **TransactionTime** таблицы с именем **Orders**.

DAX

```
= SECOND('Orders'[TransactionTime])
```

Пример 2

Следующая формула возвращает значение 3, которое является числом секунд во времени, представленном значением, **3 марта 2008 г. 12:00:03**.

DAX

```
= SECOND("March 3, 2008 12:00:03")
```

Связанное содержимое

функции даты и времениHOURMINUTEYEAR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TIME

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует часы, минуты и секунды в виде чисел в формат `datetime`.

Синтаксис

DAX

```
TIME(hour, minute, second)
```

Параметры

Развернуть таблицу

Термин	Определение
`hour`	**режим импорта:** число от 0 до 32767, представляющее час. Любое значение больше 23 будет разделено на 24, а оставшаяся часть будет рассматриваться как значение часа, представленное как доля дня. Например, TIME(27 0,0) = TIME(3,0,0) = 3:00:00:00 **режим DirectQuery:** число от 0 до 23, представляющее час.
`minute`	**режим импорта:** число от 0 до 32767, представляющее минуту. Любое значение, больше 59 минут, преобразуется в часы и минуты. Любое значение, превышающее 1440 (24 часа), не изменяет часть даты. Вместо этого она будет разделена на 1440, а оставшаяся часть будет рассматриваться как значение минуты, представленное как доля дня. Например, TIME(0,2190,0) = TIME(0 750 0) = TIME(12 30 0) = 12:30:00 ВЕЧЕРА **режим DirectQuery:** число от 0 до 59, представляющее минуту.
`second`	**режим импорта:** число от 0 до 32767, представляющее второй. Любое значение, превышающее 59, будет преобразовано в часы, минуты и секунды. Например, TIME(0,0 2000) = TIME(033 20) = 12:33:20 **режим DirectQuery:** число от 0 до 59, представляющее второй.

Возвращаемое значение

Время (`datetime`) от 00:00:00 (12:00:00:00) до 23:59:59 (11:59:59:59 ВЕЧЕРА).

Замечания

- В отличие от Microsoft Excel, в котором хранятся даты и время в виде серийных номеров, DAX работает со значениями даты и времени в формате `datetime`. Числа в других форматах неявно преобразуются при использовании значения даты и времени в функции DAX. Если необходимо использовать серийные номера, можно использовать форматирование, чтобы изменить способ отображения чисел.
- Значения времени — это часть значения даты, а в системе серийных чисел представлены десятичным числом. Поэтому значение `datetime` 12:00 pm эквивалентно 0,5, так как это половина дня.
- Аргументы можно указать функции TIME в виде значений, которые вы вводите напрямую, в результате другого выражения или по ссылке на столбец, содержащий числовое значение.
- Дату и дату также можно указать в виде литерала в формате `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` или `dt"YYYY-MM-DD hh:mm:ss"`. При указании в качестве литерала использование функции TIME в выражении не требуется. Дополнительные сведения см. в DAX синтаксисе | Дата и время.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

В следующих примерах оба возвращают время: 3:00:

DAX

```
= TIME(27,0,0)
```

DAX

```
= TIME(3,0,0)
```

Пример 2

В следующих примерах оба возвращают время, 12:30:

```
DAX
= TIME(0,750,0)
```

```
DAX
= TIME(12,30,0)
```

Пример 3

В следующем примере создается время на основе значений в столбцах, `intHours`, `intMinutes`, `intSeconds`:

```
DAX
= TIME([intHours],[intMinutes],[intSeconds])
```

Связанное содержимое

DATE функции даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TIMEVALUE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует время в текстовом формате в время в формате datetime.

Синтаксис

DAX

```
TIMEVALUE(time_text)
```

Параметры

Развернуть таблицу

Термин	Определение
`time_text`	Текстовая строка, представляющая определенное время дня. Все сведения о дате, включенные в аргумент `time_text`, игнорируются.

Возвращаемое значение

Дата (`datetime`).

Замечания

- Значения времени — это часть значения даты и представлена десятичным числом. Например, 12:00 pm представлено как 0,5, так как это половина дня.
- Если аргумент `time_text` является текстовым представлением даты и времени, функция использует языковые параметры и параметры даты и времени модели для понимания текстового значения для выполнения преобразования. Большинство языковых стандартов используют двоеточие (:) в качестве разделителя времени, а любой входной текст, использующий двоеточия в качестве разделителей времени, правильно анализируется. Просмотрите параметры языкового стандарта, чтобы понять результаты.

- Дату и дату также можно указать в виде литерала в формате `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` или `dt"YYYY-MM-DD hh:mm:ss"`. При указании в качестве литерала использование функции TIMEVALUE в выражении не требуется. Дополнительные сведения см. в DAX синтаксисе | Дата и время.

Пример

DAX

```
= TIMEVALUE("20:45:30")
```

Связанное содержимое

функции даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TODAY

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает текущую дату.

Синтаксис

```DAX
TODAY()
```

Возвращаемое значение

Дата (`datetime`).

Замечания

- Функция TODAY полезна при необходимости отображения текущей даты на листе независимо от того, когда откроется книга. Кроме того, это полезно для вычисления интервалов.
- Если функция TODAY не обновляет дату, когда она ожидается, может потребоваться изменить параметры, управляющие обновлением столбца или книги.
- Функция NOW аналогична, но возвращает точное время, а TODAY возвращает значение времени 12:00:00 для всех дат.

Пример

Если вы знаете, что кто-то родился в 1963 году, вы можете использовать следующую формулу, чтобы найти возраст этого человека по состоянию на день рождения этого года:

```DAX
```

```
= YEAR(TODAY())-1963
```

Эта формула использует функцию TODAY в качестве аргумента для функции YEAR для получения текущего года, а затем вычитает 1963 год, возвращая возраст человека.

Связанное содержимое

функций даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

UTCNOW

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает текущую дату и время в формате UTC.

Синтаксис

```DAX
UTCNOW()
```

Возвращаемое значение

`datetime`.

Замечания

Результат функции UTCNOW изменяется только при обновлении формулы. Он не обновляется постоянно.

Пример

Следующее:

```DAX
EVALUATE { FORMAT(UTCNOW(), "General Date") }
```

Возвращает:

Развернуть таблицу

Ценность
2.2.2018 4:48:08

Связанное содержимое

функция NOWUTCTODAY

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

UTCTODAY

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает текущую дату UTC.

Синтаксис

```
DAX

UTCTODAY()
```

Возвращаемое значение

Дата.

Замечания

- UTCTODAY возвращает значение времени 12:00:00 для всех дат.
- Функция UTCNOW аналогична, но возвращает точное время и дату.

Пример

Следующее:

```
DAX

EVALUATE { FORMAT(UTCTODAY(), "General Date") }
```

Возвращает:

Развернуть таблицу

[значение]
2.02.2018

Связанное содержимое

функция NOWUTCNOW

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

WEEKDAY

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает число от 1 до 7, определяющее день недели даты. По умолчанию день составляет от 1 (воскресенье) до 7 (суббота).

Синтаксис

DAX

```
WEEKDAY(<date>, <return_type>)
```

Параметры

Развернуть таблицу

Термин	Определение
`date`	Дата в формате `datetime`. Даты следует вводить с помощью функции DATE, используя выражения, которые приводят к дате или в результате других формул.
`return_type`	Число, определяющее возвращаемое значение: Тип возврата: **1**, неделя начинается в воскресенье (1) и заканчивается в субботу (7). нумеровано от 1 до 7. Тип возврата: **2**, неделя начинается в понедельник (1) и заканчивается в воскресенье (7). Тип возврата: **3**, неделя начинается в понедельник (0) и заканчивается в воскресенье (6).нумерованный от 0 до 6.

Возвращаемое значение

Целочисленное число от 1 до 7.

Замечания

- В отличие от Microsoft Excel, который хранит даты в виде серийных номеров, DAX работает с датами и временем в формате `datetime`. Если необходимо отобразить даты в виде серийных номеров, можно использовать параметры форматирования в Excel.

- Вы также можете вводить даты в принятом текстовом представлении даты, но чтобы избежать непредвиденных результатов, лучше сначала преобразовать дату текста в формат `datetime`.

- Если аргумент даты является текстовым представлением даты, функция использует параметры языкового стандарта и даты и времени клиентского компьютера для понимания текстового значения для выполнения преобразования. Если текущие параметры даты и времени представляют даты в формате "Месяц/день/год", строка "1/8/2009" интерпретируется как значение `datetime` эквивалентно 8 января 2009 года. Однако если текущие параметры даты и времени представляют даты в формате Day/Month/Year, то та же строка будет интерпретирована как значение `datetime` эквивалентно 1 августа 2009 года.

Пример

В следующем примере возвращается дата из столбца [HireDate], добавляется 1 и отображается день недели, соответствующий этой дате. Так как аргумент `return_type` опущен, используется формат по умолчанию, в котором 1 — воскресенье и 7 — суббота. Если результат равен 4, день будет в среду.

DAX

```
= WEEKDAY([HireDate]+1)
```

Связанное содержимое

функции даты и времениWEEKNUM функцииYEARFRAC

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

WEEKNUM

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает номер недели для заданной даты в соответствии с значением `return_type`. Номер недели указывает, где неделя падает числовым образом в течение года.

Для этой функции используются две *системы* :

- **System 1** - Неделя, содержащая 1 января, является первой неделей года и нумеруется 1 неделя.
- **System 2** - Неделя, содержащая первый четверг года является первой неделей года и нумеруется как неделя 1. Эта система является методологией, указанной в ISO 8601, которая обычно называется европейской системой нумерования недель.

Синтаксис

DAX

```
WEEKNUM(<date>[, <return_type>])
```

Параметры

Развернуть таблицу

Термин	Определение
`date`	Дата в формате `datetime`.
`return_type`	(Необязательно) Число, определяющее, на какой день начинается неделя. По умолчанию 1. См. заметки.

Возвращаемое значение

Целочисленное число.

Замечания

- По умолчанию функция WEEKNUM использует соглашение календаря, в котором неделя, содержащая 1 января, считается первой неделей года. Однако стандарт календаря ISO 8601, широко используемый в Европе, определяет первую неделю как одну с большинством дней (четыре или более) падение в новом году. Это означает, что если `return_type` является допустимым значением, отличным от 21, в течение любых лет, в течение которых в течение первой недели января, функция WEEKNUM возвращает номера недель, отличные от определения ISO 8601.
- Для `return_type` следующие допустимые значения могут не поддерживаться некоторыми источниками данных DirectQuery:

Развернуть таблицу

return_type	Неделя начинается	Системные
1 или опущено	Воскресенье	1
2	Понедельник	1
11	Понедельник	1
12	Вторник	1
13	Среда	1
14	Четверг	1
15	Пятница	1
16	Суббота	1
17	Воскресенье	1
21	Понедельник	2

Пример 1

В следующем примере возвращается номер недели 14 февраля 2010 г. Это вычисление предполагает, что недели начинаются в понедельник.

```
DAX
```

```
= WEEKNUM("Feb 14, 2010", 2)
```

Пример 2

В следующем примере возвращается число недель, хранящихся в столбце, **HireDate**, из таблицы **Employees**. Это вычисление предполагает, что недели начинаются в воскресенье.

DAX

```
= WEEKNUM('Employees'[HireDate])
```

Связанное содержимое

функция YEARFRACWEEKDAY

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

YEAR

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает год даты в виде четырехзначного целого числа в диапазоне 1900-9999.

Синтаксис

DAX

```
YEAR(<date>)
```

Параметры

Развернуть таблицу

Термин	Определение
`date`	Дата в `datetime` или текстовом формате, содержащая год, который вы хотите найти.

Возвращаемое значение

Целое число в диапазоне 1900-9999.

Замечания

- В отличие от Microsoft Excel, который хранит даты в виде серийных номеров, DAX использует тип данных `datetime` для работы с датами и временем.
- Даты следует вводить с помощью функции DATE или в виде результатов других формул или функций. Вы также можете ввести даты в принятых текстовых представлениях даты, таких как 3 марта 2007 г. или Мар-3-2003.
- Значения, возвращаемые YEAR, MONTHи DAY функции будут григорианскими значениями независимо от формата отображения указанного значения даты. Например, если формат отображения указанной даты использует календарь

Хиджри, возвращаемые значения для функций YEAR, MONTHи DAY будут значениями, связанными с эквивалентной датой Григориана.

- Если аргумент даты является текстовым представлением даты, функция использует параметры языкового стандарта и даты клиентского компьютера для понимания текстового значения для выполнения преобразования. Ошибки могут возникать, если формат строк несовместим с текущими параметрами языкового стандарта. Например, если языковой стандарт определяет даты, отформатированные как месяц/день/год, а дата предоставляется как день/месяц/год, то 25/1.2009 не будет интерпретировано как 25 января 2009 года, но как недопустимая дата.

Пример

В следующем примере возвращается значение 2007.

```
DAX
= YEAR("March 2007")
```

Пример: дата в результате выражения

Description

В следующем примере возвращается год для сегодняшней даты.

```
DAX
= YEAR(TODAY())
```

Связанное содержимое

функции функции даты и времени функции функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

YEARFRAC

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Вычисляет долю года, представленную числом целых дней между двумя датами. Используйте функцию YEARFRAC листа, чтобы определить долю пособий или обязательств целого года для назначения определенного срока.

Синтаксис

DAX

```
YEARFRAC(<start_date>, <end_date>, <basis>)
```

Параметры

Развернуть таблицу

Термин	Определение
`start_date`	Дата начала в формате `datetime`.
`end_date`	Дата окончания в формате `datetime`.
`basis`	(Необязательно) Тип используемого числа дней. Все аргументы усечены в целые числа. Основы — описание 0 — US (NASD) 30/360 (значение по умолчанию) 1 — Фактический/фактический 2 — фактическая/360 3 — Фактический/365 4 - Европейский 30/360

Возвращаемое значение

Десятичное число. Внутренний тип данных — это подписанный номер 64-разрядной (8-байтовой) с двойной точностью с плавающей запятой.

Замечания

- В отличие от Microsoft Excel, который хранит даты в качестве серийных номеров, DAX использует формат `datetime` для работы с датами и временем. Если необходимо просмотреть даты в виде серийных номеров, можно использовать параметры форматирования в Excel.
- Если `start_date` или `end_date` недопустимы даты, YEARFRAC возвращает ошибку.
- Если `basis` < 0 или `basis` > 4, YEARFRAC возвращает ошибку.

Пример 1

В следующем примере возвращается доля года, представленная разницей между датами в двух столбцах, `TransactionDate` и `ShippingDate`:

DAX

```
= YEARFRAC(Orders[TransactionDate],Orders[ShippingDate])
```

Пример 2

В следующем примере возвращается доля года, представленная разницей между датами, 1 января и 1 марта:

DAX

```
= YEARFRAC("Jan 1 2007","Mar 1 2007")
```

Используйте четыре цифры лет, когда это возможно, чтобы избежать непредвиденных результатов. При усечении года предполагается текущий год. Если дата опущена или опущена, предполагается первая дата месяца.

Второй аргумент, `basis`, также был опущен. Таким образом, доля года вычисляется в соответствии со стандартом US (NASD) 30/360.

Связанное содержимое

функции функции функции даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Фильтрация функций

Статья • 23.01.2025

Функции фильтров и значений в DAX являются одними из самых сложных и мощных и значительно отличаются от функций Excel. Функции подстановки работают с помощью таблиц и связей, таких как база данных. Функции фильтрации позволяют управлять контекстом данных для создания динамических вычислений.

В этой категории

Развернуть таблицу

Функция	Описание
ALL	Возвращает все строки в таблице или все значения в столбце, игнорируя все фильтры, которые могли быть применены.
ALLCROSSFILTERED	Снимите все фильтры, применяемые к таблице.
ALLEXCEPT	Удаляет все фильтры контекста в таблице, кроме фильтров, которые были применены к указанным столбцам.
ALLNOBLANKROW	Из родительской таблицы связи возвращает все строки, но пустую строку или все отдельные значения столбца, но пустую строку, и игнорирует все фильтры контекста, которые могут существовать.
ALLSELECTED	Удаляет фильтры контекста из столбцов и строк в текущем запросе, сохраняя все остальные фильтры контекста или явные фильтры.
CALCULATE	Вычисляет выражение в измененном контексте фильтра.
CALCULATETABLE	Вычисляет табличное выражение в измененном контексте фильтра.
EARLIER	Возвращает текущее значение указанного столбца во внешнем проходе оценки указанного столбца.
EARLIEST	Возвращает текущее значение указанного столбца во внешнем проходе оценки указанного столбца.
FILTER	Возвращает таблицу, представляющую подмножество другой таблицы или выражения.
FIRST	Используется только в визуальных вычислениях. Извлекает значение в визуальной матрице из первой строки оси.
INDEX	Возвращает строку в абсолютной позиции, указанной параметром позиции, в пределах указанной секции, отсортированную по

Функция	Описание
	указанному порядку или по указанной оси.
KEEPFILTERS	Изменяет способ применения фильтров при оценке функции CALCULATE или CALCULATETABLE.
LAST	Используется только в визуальных вычислениях. Извлекает значение в визуальной матрице из последней строки оси.
LOOKUPVALUE	Возвращает значение строки, которая соответствует всем условиям поиска. Функция может применять одно или несколько условий поиска.
MATCHBY	В функциях окна определяет столбцы, которые используются для определения соответствия данных и определения *текущей строки*.
MOVINGAVERAGE	Возвращает скользящее среднее значение, вычисляемое по заданной оси визуальной матрицы.
NEXT	Используется только в визуальных вычислениях. Извлекает значение в следующей строке оси в визуальной матрице.
OFFSET	Возвращает одну строку, расположенную либо до, либо после *текущей строки* в той же таблице, с помощью заданного смещения.
ORDERBY	Определяет столбцы, определяющие порядок сортировки в каждой секции функции окна.
PARTITIONBY	Определяет столбцы, используемые для секционирования параметра `relation` функции окна.
PREVIOUS	Используется только в визуальных вычислениях. Извлекает значение в предыдущей строке оси в визуальной матрице.
RANGE	Возвращает интервал строк в заданной оси относительно текущей строки. Ярлык для WINDOW.
RANK	Возвращает ранжирование строки в пределах заданного интервала.
REMOVEFILTERS	Очищает фильтры из указанных таблиц или столбцов.
ROWNUMBER	Возвращает уникальное ранжирование строки в пределах заданного интервала.
RUNNINGSUM	Возвращает выполняемую сумму, вычисляемую вдоль заданной оси визуальной матрицы.
SELECTEDVALUE	Возвращает значение, когда контекст для columnName отфильтрован только до одного отдельного значения. В противном случае возвращает альтернативное значениеResult.
WINDOW	Возвращает несколько строк, расположенных в пределах заданного

Функция	Описание
	интервала.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ALL

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает все строки в таблице или все значения в столбце, игнорируя все фильтры, которые могли быть применены. Эта функция полезна для очистки фильтров и создания вычислений во всех строках таблицы.

Синтаксис

```
DAX

ALL( [<table> | <column>[, <column>[, <column>[,…]]]] )
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, в которой требуется очистить фильтры.
`column`	Столбец, на который требуется очистить фильтры.

Аргумент функции ALL должен быть ссылкой на базовую таблицу или ссылку на базовый столбец. Нельзя использовать табличные выражения или выражения столбцов с функцией ALL.

Возвращаемое значение

Таблица или столбец с удаленными фильтрами.

Замечания

- Эта функция сама по себе не используется, но служит промежуточной функцией, которая может использоваться для изменения набора результатов, по которым выполняется другое вычисление.

- Обычное поведение для выражений DAX, содержащих функцию ALL() заключается в том, что все примененные фильтры будут игнорироваться. Однако существуют некоторые сценарии, в которых это не так из-за `auto-exist`, технология DAX, которая оптимизирует фильтрацию для уменьшения объема обработки, необходимого для определенных DAX запросов. Пример, в котором автоматически существуют и ALL() предоставляют непредвиденные результаты, заключается в фильтрации по двум или нескольким столбцам одной таблицы (например, при использовании срезов) и в той же таблице, которая использует ALL(). В этом случае автоматическое существование *слиянием* нескольких фильтров в один и будет фильтровать только по существующим сочетаниям значений. Из-за этого слияния мера будет вычисляться по существующим сочетаниям значений, а результат будет основан на отфильтрованных значениях вместо всех ожидаемых значений. Дополнительные сведения об автоматическом существовании и его влиянии на вычисления см. в статье о sql.bi.comMicrosoft MVP Альберто Феррари основные сведения о DAX auto-Exist.

- В следующей таблице описывается, как использовать функции ALL и ALLEXCEPT в разных сценариях.

Развернуть таблицу

Функция и использование	Описание
ALL()	Удаляет все фильтры везде. ALL() можно использовать только для очистки фильтров, но не для возврата таблицы.
ALL(таблица)	Удаляет все фильтры из указанной таблицы. В результате ALL(таблица) возвращает все значения в таблице, удаляя все фильтры из контекста, которые в противном случае могли быть применены. Эта функция полезна при работе с множеством уровней группировки и требуется создать вычисление, которое создает соотношение агрегированного значения к общему значению. Первый пример демонстрирует этот сценарий.
ALL (столбец[, столбец[, ...]])	Удаляет все фильтры из указанных столбцов в таблице; все остальные фильтры для других столбцов в таблице по-прежнему применяются. Все аргументы столбцов должны поступать из одной таблицы. Вариант ALL(столбец) полезен, если требуется удалить фильтры контекста для одного или нескольких конкретных столбцов и сохранить все остальные фильтры контекста. Второй и третий примеры демонстрируют этот сценарий.

Функция и использование	Описание
ALLEXCEPT(таблица, столбец1 [,столбец2]...)	Удаляет все фильтры контекста в таблице, кроме фильтров, применяемых к указанным столбцам. Это удобное сочетание клавиш для ситуаций, в которых требуется удалить фильтры на многих, но не все столбцы в таблице.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

Вычисление соотношения продаж категории к общему объему продаж

Предположим, что вы хотите найти объем продаж для текущей ячейки в сводной таблице, разделенной на общую продажу для всех торговых посредников. Чтобы убедиться, что знаменатель совпадает независимо от того, как пользователь сводной таблицы может фильтровать или группировать данные, вы определяете формулу, которая использует ALL для создания правильного общего общего объема.

В следующей таблице показаны результаты при создании новой меры, **все коэффициенты продаж торговых посредников**, с помощью формулы, показанной в разделе кода. Чтобы узнать, как это работает, добавьте поле CalendarYear в область **метки строк** сводной таблицы и добавьте поле ProductCategoryName в область **Метки столбцов**. Затем перетащите меру **коэффициент продаж всех торговых посредников**в область **значений** таблицы сводной таблицы. Чтобы просмотреть результаты в процентах, используйте функции форматирования Excel для применения процентного форматирования чисел к ячейкам, содержащим меру.

Развернуть таблицу

Метки строк	Принадлежности	Велосипеды	Одежда	Компоненты	Общий итог
2005	0.02%	9.10%	0.04%	0.75%	9.91%
2006	0.11%	24.71%	0.60%	4.48%	29.90%
2007	0.36%	31.71%	1.07%	6.79%	39.93%
2008	0.20%	16.95%	0.48%	2.63%	20.26%

Метки строк	Принадлежности	Велосипеды	Одежда	Компоненты	Общий итог
Общий итог	0.70%	82.47%	2.18%	14.65%	100.00%

формулы

```DAX
= SUMX(ResellerSales_USD,
ResellerSales_USD[SalesAmount_USD])/SUMX(ALL(ResellerSales_USD),
ResellerSales_USD[SalesAmount_USD])
```

Формула создается следующим образом:

1. Числитель, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, — это сумма значений в ResellerSales_USD[SalesAmount_USD] для текущей ячейки сводной таблицы с фильтрами контекста, применяемыми к CalendarYear и ProductCategoryName.

2. Для знаменателя необходимо указать таблицу, ResellerSales_USD и использовать функцию ALL для удаления всех фильтров контекста в таблице.

3. Затем функция SUMX используется для суммирования значений в столбце ResellerSales_USD[SalesAmount_USD]. Другими словами, вы получаете сумму ResellerSales_USD[SalesAmount_USD] для всех торговых посредников продаж.

Пример 2

Вычисление соотношения продаж продуктов к общему объему продаж за текущий год

Предположим, что вы хотите создать таблицу, показывающую процент продаж по сравнению с годами для каждой категории продукта (ProductCategoryName). Чтобы получить процент за каждый год по каждому значению ProductCategoryName, необходимо разделить сумму продаж за этот конкретный год и категорию продуктов на сумму продаж для одной категории продуктов на протяжении всех лет. Другими словами, вы хотите сохранить фильтр на ProductCategoryName, но удалить фильтр в год при расчете знаменателя процента.

В следующей таблице показаны результаты, когда новая мера, **год продаж торгового посредника**, создается с помощью формулы, показанной в разделе кода. Чтобы узнать, как это работает, добавьте поле CalendarYear в область метки строк сводной таблицы и добавьте поле ProductCategoryName в область меток

столбцов . Чтобы просмотреть результаты в процентах, используйте функции форматирования Excel для применения процентного формата к ячейкам, содержащим меру, **год продаж торгового посредника**.

Развернуть таблицу

Метки строк	Принадлежности	Велосипеды	Одежда	Компоненты	Общий итог
2005	3.48%	11.03%	1.91%	5.12%	9.91%
2006	16.21%	29.96%	27.29%	30.59%	29.90%
2007	51.62%	38.45%	48.86%	46.36%	39.93%
2008	28.69%	20.56%	21.95%	17.92%	20.26%
Общий итог	100.00%	100.00%	100.00%	100.00%	100.00%

формулы

```DAX
= SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE(
SUM( ResellerSales_USD[SalesAmount_USD]), ALL(DateTime[CalendarYear]))
```

Формула создается следующим образом:

1. Числитель, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, — это сумма значений в ResellerSales_USD[SalesAmount_USD] для текущей ячейки в сводной таблице с фильтрами контекста, примененными к столбцам CalendarYear и ProductCategoryName.

2. Для знаменателя удалите существующий фильтр в CalendarYear с помощью функции ALL(столбец). Это вычисляет сумму по оставшимся строкам в таблице ResellerSales_USD после применения существующих фильтров контекста из меток столбцов. Чистый эффект заключается в том, что для знаменателя сумма вычисляется по выбранному объекту ProductCategoryName (подразумеваемый фильтр контекста) и для всех значений года.

Пример 3

Вычисление вклада категорий продуктов в общий объем продаж в год

Предположим, что вы хотите создать таблицу, которая показывает процент продаж для каждой категории продуктов по годам. Чтобы получить процент для каждой

категории продуктов в определенном году, необходимо вычислить сумму продаж для этой конкретной категории продуктов (ProductCategoryName) в году n, а затем разделить результирующее значение на сумму продаж за год n по всем категориям продуктов. Другими словами, вы хотите сохранить фильтр в год, но удалить фильтр на ProductCategoryName при вычислении знаменателя процента.

В следующей таблице показаны результаты, когда новая мера, **имя категории продаж торгового посредника**, создается с помощью формулы, показанной в разделе кода. Чтобы узнать, как это работает, добавьте поле CalendarYear в область **меток строк** сводной таблицы и добавьте поле ProductCategoryName в область **Метки столбцов**. Затем добавьте новую меру в область значений сводной таблицы. Чтобы просмотреть результаты в процентах, используйте функции форматирования Excel, чтобы применить процентный формат числа к ячейкам, содержащим новую меру, **имя категории продаж торгового посредника**.

Развернуть таблицу

Метки строк	Принадлежности	Велосипеды	Одежда	Компоненты	Общий итог
2005	0.25%	91.76%	0.42%	7.57%	100.00%
2006	0.38%	82.64%	1.99%	14.99%	100.00%
2007	0.90%	79.42%	2.67%	17.01%	100.00%
2008	0.99%	83.69%	2.37%	12.96%	100.00%
Общий итог	0.70%	82.47%	2.18%	14.65%	100.00%

формулы

DAX

```
= SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE(
SUM( ResellerSales_USD[SalesAmount_USD]),
ALL(ProductCategory[ProductCategoryName]))
```

Формула создается следующим образом:

1. Числитель, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, — это сумма значений в ResellerSales_USD[SalesAmount_USD] для текущей ячейки в сводной таблице с фильтрами контекста, примененными к полям, CalendarYear и ProductCategoryName.

2. Для знаменателя используется функция ALL(столбец), чтобы удалить фильтр по ProductCategoryName и вычислить сумму по оставшимся строкам в таблице ResellerSales_USD после применения существующих фильтров контекста из меток строк. Чистый эффект заключается в том, что для знаменателя сумма вычисляется за выбранный год (подразумеваемый фильтр контекста) и для всех значений ProductCategoryName.

Связанное содержимое

функция фильтрация функцийALLALLEXCEPT функцииFILTER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ALLCROSSFILTERED

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Снимите все фильтры, применяемые к таблице.

Синтаксис

```
DAX

ALLCROSSFILTERED(<table>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, в которой требуется очистить фильтры.

Возвращаемое значение

N/A. См. примечания.

Замечания

- ALLCROSSFILTERED можно использовать только для очистки фильтров, но не для возврата таблицы.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

```
DAX
```

```
DEFINE
MEASURE FactInternetSales[TotalQuantity1] =
    CALCULATE(SUM(FactInternetSales[OrderQuantity]),
ALLCROSSFILTERED(FactInternetSales))
MEASURE FactInternetSales[TotalQuantity2] =
    CALCULATE(SUM(FactInternetSales[OrderQuantity]), ALL(FactInternetSales))
EVALUATE
    SUMMARIZECOLUMNS(DimSalesReason[SalesReasonName],
        "TotalQuantity1", [TotalQuantity1],
        "TotalQuantity2", [TotalQuantity2])
    ORDER BY DimSalesReason[SalesReasonName]
```

Возвращает

Развернуть таблицу

DimSalesReason[SalesReasonName]	[TotalQuantity1]	[TotalQuantity2]
Демонстрационное событие	60398	
Реклама журнала	60398	
Изготовитель	60398	1818
На повышение	60398	7390
Другой	60398	3653
Цена	60398	47733
Качество	60398	1551
Обзор	60398	1640
Спонсорство	60398	
Телевизионная реклама	60398	730

Примечание

Существует прямая или непрямая связь между таблицей FactInternetSales и таблицей DimSalesReason.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ALLEXCEPT

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Удаляет все фильтры контекста в таблице, кроме фильтров, которые были применены к указанным столбцам.

Синтаксис

```
DAX

ALLEXCEPT(<table>,<column>[,<column>[,…]])
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, по которой удаляются все фильтры контекста, за исключением фильтров для этих столбцов, указанных в последующих аргументах.
`column`	Столбец, для которого необходимо сохранить фильтры контекста.

Первый аргумент функции ALLEXCEPT должен быть ссылкой на базовую таблицу. Все последующие аргументы должны быть ссылками на базовые столбцы. Нельзя использовать табличные выражения или выражения столбцов с функцией ALLEXCEPT.

Возвращаемое значение

Таблица со всеми фильтрами удалена, за исключением фильтров по указанным столбцам.

Замечания

- Эта функция сама по себе не используется, но служит промежуточной функцией, которая может использоваться для изменения набора результатов,

по которым выполняется другое вычисление.

- ALL и ALLEXCEPT можно использовать в различных сценариях:

Развернуть таблицу

Функция и использование	Описание
ALL(таблица)	Удаляет все фильтры из указанной таблицы. В результате ALL(таблица) возвращает все значения в таблице, удаляя все фильтры из контекста, которые в противном случае могли быть применены. Эта функция полезна при работе с множеством уровней группировки и требуется создать вычисление, которое создает соотношение агрегированного значения к общему значению.
ALL (столбец[, столбец[, ...]])	Удаляет все фильтры из указанных столбцов в таблице; все остальные фильтры для других столбцов в таблице по-прежнему применяются. Все аргументы столбцов должны поступать из одной таблицы. Вариант ALL(столбец) полезен, если требуется удалить фильтры контекста для одного или нескольких конкретных столбцов и сохранить все остальные фильтры контекста.
ALLEXCEPT(таблица, столбец1 [,столбец2]...)	Удаляет все фильтры контекста в таблице, кроме фильтров, применяемых к указанным столбцам. Это удобное сочетание клавиш для ситуаций, в которых требуется удалить фильтры на многих, но не все столбцы в таблице.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула измерения суммирует SalesAmount_USD и использует функцию ALLEXCEPT для удаления любых фильтров контекста в таблице DateTime, за исключением случаев, когда фильтр был применен к столбцу CalendarYear.

DAX

```
= CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]), ALLEXCEPT(DateTime,
DateTime[CalendarYear]))
```

Так как формула использует ALLEXCEPT, всякий раз, когда любой столбец, но CalendarYear из таблицы DateTime используется для среза визуализации, формула удаляет все фильтры среза, предоставляя значение, равное сумме SalesAmount_USD. Однако если столбец CalendarYear используется для среза визуализации, результаты отличаются. Так как CalendarYear указывается в качестве аргумента для ALLEXCEPT, когда данные срезаются в год, фильтр будет применяться на протяжении многих лет на уровне строк.

Связанное содержимое

Функция фильтрации функцийALLFILTER функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ALLNOBLANKROW

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Из родительской таблицы связи возвращает все строки, но пустую строку или все отдельные значения столбца, но пустую строку, и игнорирует все фильтры контекста, которые могут существовать.

Синтаксис

```
DAX

ALLNOBLANKROW( {<table> | <column>[, <column>[, <column>[,…]]]} )
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, по которой удаляются все фильтры контекста.
`column`	Столбец, по которому удаляются все фильтры контекста.

Необходимо передать только один параметр; параметр — это таблица или столбец.

Возвращаемое значение

Таблица, когда переданный параметр был таблицей или столбцом значений, когда переданный параметр был столбцом.

Замечания

- Функция ALLNOBLANKROW фильтрует только пустую строку, которую родительская таблица в связи отображает, если в дочерней таблице есть одна или несколько строк, не соответствующих значениям родительского столбца. Подробные объяснения см. в приведенном ниже примере.

- В следующей таблице перечислены варианты `ALL`, предоставляемые в DAX, и их различия.

  Развернуть таблицу

Функция и использование	Описание
`ALL(Column)`	Удаляет все фильтры из указанного столбца в таблице; все остальные фильтры в таблице, по-прежнему применяются к другим столбцам.
`ALL(Table)`	Удаляет все фильтры из указанной таблицы.
`ALLEXCEPT(Table,Col1,Col2...)`	Переопределяет все фильтры контекста в таблице, кроме указанных столбцов.
`ALLNOBLANK(table|column)`	Из родительской таблицы связи возвращает все строки, но пустую строку или все отдельные значения столбца, но пустую строку, и игнорирует все фильтры контекста, которые могут существовать.

  Общее описание работы функции ALL вместе с пошаговые примеры, использующие ALL(таблица) и ALL(столбец), см. ALL функции.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В примерах данных таблица ResellerSales_USD содержит одну строку, которая не имеет значений и поэтому не может быть связана с одной из родительских таблиц в отношениях в книге. Эта таблица будет использоваться в сводной таблице, чтобы увидеть пустое поведение строки и как обрабатывать количество несвязанных данных.

Шаг 1. Проверка несвязанных данных

Откройте окно **PowerPivot**, а затем выберите таблицу ResellerSales_USD. В столбце ProductKey отфильтруйте пустые значения. Одна строка останется. В этой строке все значения столбцов должны быть пустыми, за исключением SalesOrderLineNumber.

Шаг 2. Создание сводной таблицы

Создайте новую сводную таблицу, а затем перетащите столбец, datetime. [Календарный год], на панель меток строк. В следующей таблице показаны ожидаемые результаты:

Развернуть таблицу

Метки строк
2005
2006
2007
2008
Общий итог

Запишите пустую метку между **2008** и **грандиозным итогом**. Эта пустая метка представляет собой элемент Unknown, который является специальной группой, которая создается для учета любых значений в дочерней таблице, не имеющих соответствующего значения в родительской таблице, в этом примере datetime. Столбец [календарный год].

При отображении этой пустой метки в сводной таблице вы знаете, что в некоторых таблицах, связанных с столбцом, datetime. [Календарный год], имеются пустые значения или не соответствующие значения. Родительская таблица — это та, которая отображает пустую метку, но строки, которые не совпадают, находятся в одной или нескольких дочерних таблицах.

Строки, которые добавляются в эту пустую группу меток, являются либо значениями, которые не соответствуют ни одному значению родительской таблицы, например дате, которая не существует в значениях таблицы datetime или NULL, то есть нет значения для даты вообще. В этом примере мы разместили пустое значение во всех столбцах дочерней таблицы продаж. Наличие большего числа значений в родительской таблице, чем в дочерних таблицах, не вызывает проблемы.

Шаг 3. Подсчет строк с помощью ALL и ALLNOBLANK

Добавьте в таблицу datetime следующие два меры, чтобы подсчитать строки таблицы: **Countrows ALLNOBLANK даты и времени**, **Countrows ALL даты и времени**. Формулы, которые можно использовать для определения этих мер:

```
DAX
```

```
// Countrows ALLNOBLANK of datetime
= COUNTROWS(ALLNOBLANKROW('DateTime'))

// Countrows ALL of datetime
= COUNTROWS(ALL('DateTime'))

// Countrows ALLNOBLANKROW of ResellerSales_USD
= COUNTROWS(ALLNOBLANKROW('ResellerSales_USD'))

// Countrows ALL of ResellerSales_USD
= COUNTROWS(ALL('ResellerSales_USD'))
```

В пустой сводной таблице добавьте дату и время. Столбец [Календарный год] в метки строк, а затем добавьте только что созданные меры. Результаты должны выглядеть следующим образом:

Развернуть таблицу

Метки строк	Countrows ALLNOBLANK даты и времени	Счетчики ALL даты и времени
2005	1280	1281
2006	1280	1281
2007	1280	1281
2008	1280	1281
	1280	1281
Общий итог	1280	1281

Результаты показывают разницу в 1 строке в счетчике строк таблицы. Однако если открыть окно **PowerPivot** и выбрать таблицу datetime, вы не можете найти пустую строку в таблице, так как специальная пустая строка, указанная здесь, является неизвестным элементом.

Шаг 4. Убедитесь, что счетчик является точным

Чтобы доказать, что ALLNOBLANKROW не подсчитывает действительно пустые строки и обрабатывает только специальную пустую строку только в родительской таблице, добавьте следующие два меры в таблицу ResellerSales_USD: **Countrows ALLNOBLANKROW ResellerSales_USD**, **Countrows ALL ResellerSales_USD**.

Создайте сводную таблицу и перетащите столбец, datetime. [Календарный год], на панель меток строк. Теперь добавьте только что созданные меры. Результаты

должны выглядеть следующим образом:

Развернуть таблицу

Метки строк	Подсчет ALLNOBLANKROW ResellerSales_USD	Подсчет ALL ResellerSales_USD
2005	60856	60856
2006	60856	60856
2007	60856	60856
2008	60856	60856
	60856	60856
Общий итог	60856	60856

Теперь два меры имеют одинаковые результаты. Это связано с тем, что функция ALLNOBLANKROW не подсчитывает действительно пустые строки в таблице, а обрабатывает только пустую строку, созданную в родительской таблице, если одна или несколько дочерних таблиц в связи содержат несочетаемые значения или пустые значения.

Связанное содержимое

Функция фильтрации функцийALLFILTER функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ALLSELECTED

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Удаляет фильтры контекста из столбцов и строк в текущем запросе, сохраняя все остальные фильтры контекста или явные фильтры.

Функция ALLSELECTED получает контекст, представляющий все строки и столбцы в запросе, сохраняя явные фильтры и контексты, отличные от фильтров строк и столбцов. Эту функцию можно использовать для получения визуальных итогов в запросах.

Синтаксис

DAX

```
ALLSELECTED([<tableName> | <columnName>[, <columnName>[, <columnName>[,…]]]] )
```

Параметры

Развернуть таблицу

Срок	Определение
`tableName`	Имя существующей таблицы с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением. Этот параметр является необязательным.
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX, как правило, полное. Это не может быть выражением. Этот параметр является необязательным.

Возвращаемое значение

Контекст запроса без фильтров столбцов и строк.

Замечания

- Если есть один аргумент, аргумент либо `tableName`, либо `columnName`. Если имеется несколько аргументов, они должны быть столбцами из одной таблицы.
- Эта функция отличается от ALL(), так как она сохраняет все фильтры явным образом, заданные в запросе, и сохраняет все фильтры контекста, отличные от фильтров строк и столбцов.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показано, как создать различные уровни визуальных итогов в табличном отчете с помощью выражений DAX. В отчете два (2) предыдущих фильтра были применены к данным продаж торгового посредника; одна из групп территории продаж = *Европа* и другая по типу продвижения = *скидки на объем*. После применения фильтров визуальные итоги можно вычислить для всего отчета, для всех лет или для всех категорий продуктов. Кроме того, для иллюстрации общий итог для всех продаж торговых посредников также получается, удаляя все фильтры в отчете. При оценке следующего выражения DAX приводится таблица со всеми сведениями, необходимыми для создания таблицы с визуальными итогами.

DAX

```
define
measure 'Reseller Sales'[Reseller Sales Amount]=sum('Reseller Sales'[Sales Amount])
measure 'Reseller Sales'[Reseller Grand Total]=calculate(sum('Reseller Sales'[Sales
Amount]), ALL('Reseller Sales'))
measure 'Reseller Sales'[Reseller Visual Total]=calculate(sum('Reseller Sales'[Sales
Amount]), ALLSELECTED())
measure 'Reseller Sales'[Reseller Visual Total for All of Calendar
Year]=calculate(sum('Reseller Sales'[Sales Amount]), ALLSELECTED('Date'[Calendar Year]))
measure 'Reseller Sales'[Reseller Visual Total for All of Product Category
Name]=calculate(sum('Reseller Sales'[Sales Amount]), ALLSELECTED('Product
Category'[Product Category Name]))
evaluate
CalculateTable(
    //CT table expression
    summarize(
//summarize table expression
crossjoin(distinct('Product Category'[Product Category Name]), distinct('Date'[Calendar
Year]))
//First Group by expression
, 'Product Category'[Product Category Name]
//Second Group by expression
, 'Date'[Calendar Year]
//Summary expressions
, "Reseller Sales Amount", [Reseller Sales Amount]
, "Reseller Grand Total", [Reseller Grand Total]
, "Reseller Visual Total", [Reseller Visual Total]
, "Reseller Visual Total for All of Calendar Year", [Reseller Visual Total for All of
Calendar Year]
, "Reseller Visual Total for All of Product Category Name", [Reseller Visual Total for All
of Product Category Name]
)
//CT filters
, 'Sales Territory'[Sales Territory Group]="Europe", 'Promotion'[Promotion Type]="Volume
Discount"
)
order by [Product Category Name], [Calendar Year]
```

После выполнения приведенного выше выражения в SQL Server Management Studio для табличной модели AdventureWorks DW вы получите следующие результаты:

Развернуть таблицу

[Имя категории продукта]	[Календарный год]	[Объем продаж торгового посредника]	[Общий итог торгового посредника]	[Визуальный итог торгового посредника]	[Визуальный итог торгового посредника для всего календарного года]	[Визуальный итог торгового посредника для всех имен категории продуктов]
Принадлежности	2000		80450596.9823	877006.7987	38786.018	
Принадлежности	2001		80450596.9823	877006.7987	38786.018	
Принадлежности	2002	625.7933	80450596.9823	877006.7987	38786.018	91495.3104
Принадлежности	2003	26037.3132	80450596.9823	877006.7987	38786.018	572927.0136
Принадлежности	2004	12122.9115	80450596.9823	877006.7987	38786.018	212584.4747
Принадлежности	2005		80450596.9823	877006.7987	38786.018	
Принадлежности	2006		80450596.9823	877006.7987	38786.018	
Велосипеды	2000		80450596.9823	877006.7987	689287.7939	
Велосипеды	2001		80450596.9823	877006.7987	689287.7939	
Велосипеды	2002	73778.938	80450596.9823	877006.7987	689287.7939	91495.3104
Велосипеды	2003	439771.4136	80450596.9823	877006.7987	689287.7939	572927.0136
Велосипеды	2004	175737.4423	80450596.9823	877006.7987	689287.7939	212584.4747
Велосипеды	2005		80450596.9823	877006.7987	689287.7939	
Велосипеды	2006		80450596.9823	877006.7987	689287.7939	
Одежда	2000		80450596.9823	877006.7987	95090.7757	
Одежда	2001		80450596.9823	877006.7987	95090.7757	
Одежда	2002	12132.4334	80450596.9823	877006.7987	95090.7757	91495.3104
Одежда	2003	58234.2214	80450596.9823	877006.7987	95090.7757	572927.0136
Одежда	2004	24724.1209	80450596.9823	877006.7987	95090.7757	212584.4747
Одежда	2005		80450596.9823	877006.7987	95090.7757	
Одежда	2006		80450596.9823	877006.7987	95090.7757	
Компоненты	2000		80450596.9823	877006.7987	53842.2111	
Компоненты	2001		80450596.9823	877006.7987	53842.2111	
Компоненты	2002	4958.1457	80450596.9823	877006.7987	53842.2111	91495.3104
Компоненты	2003	48884.0654	80450596.9823	877006.7987	53842.2111	572927.0136
Компоненты	2004		80450596.9823	877006.7987	53842.2111	212584.4747

[Имя категории продукта]	[Календарный год]	[Объем продаж торгового посредника]	[Общий итог торгового посредника]	[Визуальный итог торгового посредника]	[Визуальный итог торгового посредника для всего календарного года]	[Визуальный итог торгового посредника для всех имен категории продуктов]
Компоненты	2005		80450596.9823	877006.7987	53842.2111	
Компоненты	2006		80450596.9823	877006.7987	53842.2111	

Столбцы отчета:

Объем продаж торгового посредника— фактическое значение продаж торговых посредников за год и категорию продуктов. Это значение отображается в ячейке в центре отчета на пересечении года и catergory.

Визуальный итог торгового посредника для всего календарного года— общее значение категории продуктов на протяжении всех лет. Это значение отображается в конце столбца или строки для заданной категории продукта и во всех годах отчета.

Визуальный итог торгового посредника для всех категорий продуктов— общее значение за год для всех категорий продуктов. Это значение отображается в конце столбца или строки за заданный год и во всех категориях продуктов в отчете.

Визуальный итог торгового посредника— общее значение для всех лет и категорий продуктов. Обычно это значение отображается в правом нижнем углу таблицы.

Общий итог торгового посредника это общий итог для всех продаж торговых посредников, прежде чем будет применен любой фильтр; Вы должны заметить разницу с [Визуальный итог торгового посредника]. Вы помните, что этот отчет включает два фильтра (2), один в группе категорий продуктов и другой в типе продвижения.

> ⓘ **Примечание**
>
> Если в выражении есть явные фильтры, эти фильтры также применяются к выражению.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CALCULATE

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Вычисляет выражение в измененном контексте фильтра.

> **Примечание**
>
> Есть также функция CALCULATETABLE. Он выполняет точно те же функции, за исключением того, что изменяет контекст фильтра , примененный к выражению, возвращающим объект таблицы .

Синтаксис

DAX

```
CALCULATE(<expression>[, <filter1> [, <filter2> [, …]]])
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Вычисляемого выражения.
`filter1, filter2,…`	(Необязательно) Логические выражения или табличные выражения, определяющие фильтры или функции модификатора фильтров.

Выражение, используемое в качестве первого параметра, по сути, совпадает с мерой.

Фильтры могут быть следующими:

- Логические выражения фильтра
- Выражения фильтра таблиц
- Функции изменения фильтра

При наличии нескольких фильтров их можно оценить с помощью логического оператора AND (`&&`) логических операторов, то есть все условия должны быть TRUEили логическимоператором OR (`||`), то есть любое условие может быть истинным.

Логические выражения фильтра

Логический фильтр выражений — это выражение, которое оценивается как `TRUE` или `FALSE`. Существует несколько правил, которые они должны соблюдать:

- Они могут ссылаться на столбцы из одной таблицы.
- Они не могут ссылаться на меры.
- Они не могут использовать вложенную функцию CALCULATE.

Начиная с выпуска Power BI Desktop за сентябрь 2021 г. также применяются следующие компоненты:

- Они не могут использовать функции, которые сканируют или возвращают таблицу, если они не передаются в качестве аргументов в агрегатные функции.
- Они *могут* содержать функцию агрегирования, возвращающую скалярное значение. Например

```DAX
Total sales on the last selected date =
CALCULATE (
    SUM ( Sales[Sales Amount] ),
    'Sales'[OrderDateKey] = MAX ( 'Sales'[OrderDateKey] )
)
```

Выражение фильтра таблицы

Фильтр выражений таблицы применяет объект таблицы в качестве фильтра. Это может быть ссылка на таблицу моделей, но, скорее всего, это функция, которая возвращает объект таблицы. Функцию FILTER можно использовать для применения сложных условий фильтрации, включая те, которые не могут быть определены логическим выражением фильтра.

Функции модификатора фильтра

Функции модификатора фильтров позволяют выполнять больше, чем просто добавлять фильтры. Они предоставляют дополнительный контроль при изменении контекста фильтра.

Развернуть таблицу

Функция	Цель
REMOVEFILTERS	Удалите все фильтры или фильтры из одной или нескольких столбцов таблицы или из всех столбцов одной таблицы.
ALL [1], ALLEXCEPT, ALLNOBLANKROW	Удалите фильтры из одного или нескольких столбцов или из всех столбцов одной таблицы.
KEEPFILTERS	Добавьте фильтр без удаления существующих фильтров в одних и том же столбцах.
USERELATIONSHIP	Включение неактивной связи между связанными столбцами, в этом случае активная связь автоматически станет неактивной.
CROSSFILTER	Измените направление фильтра (от одного до одного или из одного в оба) или отключите связь.

[1] Функция ALL и его варианты ведут себя как модификаторы фильтров, так и как функции, возвращающие объекты таблицы. Если функция REMOVEFILTERS поддерживается средством, лучше использовать ее для удаления фильтров.

Возвращаемое значение

Значение, которое является результатом выражения.

Замечания

- При предоставлении выражений фильтра функция CALCULATE изменяет контекст фильтра для оценки выражения. Для каждого выражения фильтра существует два возможных стандартных результата, если выражение фильтра не упаковано в функцию KEEPFILTERS:
  - Если столбцы (или таблицы) не содержатся в контексте фильтра, новые фильтры будут добавлены в контекст фильтра для оценки выражения.
  - Если столбцы (или таблицы) уже находятся в контексте фильтра, существующие фильтры будут перезаписаны новыми фильтрами для оценки выражения CALCULATE.

- Функция CALCULATE, используемая *без фильтров,* достигает определенного требования. Он преобразует контекст строки в фильтр контекст. Это необходимо, если выражение (а не мера модели), которое суммирует данные модели, необходимо оценить в контексте строки. Этот сценарий может произойти в формуле вычисляемого столбца или при вычислении выражения в функции итератора. Обратите внимание, что при использовании меры модели в контексте строки переход контекста выполняется автоматически.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Следующее **определение меры продаж** таблицы дает результат дохода, но только для продуктов с цветом синего цвета.

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

DAX

```
Blue Revenue =
CALCULATE(
    SUM(Sales[Sales Amount]),
    'Product'[Color] = "Blue"
)
```

Развернуть таблицу

Категория	Сумма продаж	Синий доход
Принадлежности	$1,272,057,89	$165,406,62
Велосипеды	$94,620,526.21	$8,374,313,88
Одежда	$2,117,613,45	$259,488,37
Компоненты	$11,799,076,66	$803,642.10
всего	**$ 109,809,274.20**	**$ 9602,850,97**

Функция оценивает сумму таблицы продаж суммы продаж столбца в измененном контексте фильтра. Новый фильтр добавляется в таблицу product color столбец или

фильтр перезаписывает любой фильтр, который уже применен к столбцу.

Следующее **определение меры продаж** таблицы создает соотношение продаж по продажам для всех каналов продаж.

Развернуть таблицу

Канал	Сумма продаж	Общий канал % выручки
Интернет	$29,358,677,22	26.74%
Перекупщик	$80,450,596,98	73.26%
всего	**$ 109,809,274.20**	**100.00%**

DAX

```
Revenue % Total Channel =
DIVIDE(
    SUM(Sales[Sales Amount]),
    CALCULATE(
        SUM(Sales[Sales Amount]),
        REMOVEFILTERS('Sales Order'[Channel])
    )
)
```

Функция делит выражение, которое суммирует значение столбца sales Sales Amount (в контексте фильтра) на то же выражение в измененном контексте фильтра. Это функция CALCULATE, которая изменяет контекст фильтра с помощью функции REMOVEFILTERS, которая является функцией модификатора фильтра. Он удаляет фильтры из столбца **Sales Order** table **Channel**.

Следующая **Customer** таблица вычисляемого столбца классифицирует клиентов в класс лояльности. Это очень простой сценарий: когда доход, полученный клиентом, меньше $ 2500, они классифицируются как *Низкий*; в противном случае они *High*.

DAX

```
Customer Segment =
IF(
    CALCULATE(SUM(Sales[Sales Amount]), ALLEXCEPT(Customer,
Customer[CustomerKey])) < 2500,
    "Low",
    "High"
)
```

В этом примере контекст строки преобразуется в контекст фильтра. Он называется *переходом контекста*. Функция удаляет фильтры из всех столбцов таблицы customer, кроме столбца CustomerKey.

Связанное содержимое

контекст фильтраконтекст строкCALCULATETABLE функцияфункции фильтрации

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CALCULATETABLE

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Вычисляет табличное выражение в измененном контексте фильтра.

> ⓘ **Примечание**
>
> Есть также функция CALCULATE. Он выполняет точно те же функции, за исключением того, что изменяет контекст фильтра , примененный к выражению, возвращающим скалярное значение.

Синтаксис

DAX

```
CALCULATETABLE(<expression>[, <filter1> [, <filter2> [, …]]])
```

Параметры

⛶ **Развернуть таблицу**

Срок	Определение
`expression`	Вычисляемая таблица.
`filter1, filter2,…`	(Необязательно) Логические выражения или табличные выражения, определяющие фильтры или функции модификатора фильтров.

Выражение, используемое в качестве первого параметра, должно быть таблицей моделей или функцией, возвращающей таблицу.

Фильтры могут быть следующими:

- Логические выражения фильтра
- Выражения фильтра таблиц
- Функции изменения фильтра

При наличии нескольких фильтров они оцениваются с помощью ANDлогического оператора. Это означает, что все условия должны быть `TRUE` одновременно.

Логические выражения фильтра

Логический фильтр выражений — это выражение, которое оценивается как `TRUE` или `FALSE`. Существует несколько правил, которые они должны соблюдать:

- Они могут ссылаться только на один столбец.
- Они не могут ссылаться на меры.
- Они не могут использовать вложенную функцию CALCULATE.

Начиная с выпуска Power BI Desktop за сентябрь 2021 г. также применяются следующие компоненты:

- Они не могут использовать функции, которые сканируют или возвращают таблицу, если они не передаются в качестве аргументов в агрегатные функции.
- Они *могут* содержать функцию агрегирования, возвращающую скалярное значение.

Выражение фильтра таблицы

Фильтр выражений таблицы применяет объект таблицы в качестве фильтра. Это может быть ссылка на таблицу моделей, но, скорее всего, это функция, которая возвращает объект таблицы. Функцию FILTER можно использовать для применения сложных условий фильтрации, включая те, которые не могут быть определены логическим выражением фильтра.

Функции модификатора фильтра

Функции модификатора фильтров позволяют выполнять больше, чем просто добавлять фильтры. Они предоставляют дополнительный контроль при изменении контекста фильтра.

Развернуть таблицу

Функция	Цель
REMOVEFILTERS	Удалите все фильтры или фильтры из одной или нескольких столбцов таблицы или из всех столбцов одной таблицы.

Функция	Цель
ALL [1], ALLEXCEPT, ALLNOBLANKROW	Удалите фильтры из одного или нескольких столбцов или из всех столбцов одной таблицы.
KEEPFILTERS	Добавьте фильтр без удаления существующих фильтров в одних и том же столбцах.
USERELATIONSHIP	Включение неактивной связи между связанными столбцами, в этом случае активная связь автоматически станет неактивной.
CROSSFILTER	Измените направление фильтра (от одного до одного или из одного в оба) или отключите связь.

[1] Функция ALL и его варианты ведут себя как модификаторы фильтров, так и как функции, возвращающие объекты таблицы. Если функция REMOVEFILTERS поддерживается средством, лучше использовать ее для удаления фильтров.

Возвращаемое значение

Таблица значений.

Замечания

- При предоставлении выражений фильтра функция CALCULATETABLE изменяет контекст фильтра для оценки выражения. Для каждого выражения фильтра существует два возможных стандартных результата, если выражение фильтра не упаковано в функцию KEEPFILTERS:
  - Если столбцы (или таблицы) не содержатся в контексте фильтра, новые фильтры будут добавлены в контекст фильтра для оценки выражения.
  - Если столбцы (или таблицы) уже находятся в контексте фильтра, существующие фильтры будут перезаписаны новыми фильтрами для оценки выражения CALCULATETABLE.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере функция CALCULATETABLE используется для получения суммы интернет-продаж за 2006 год. Это значение позже используется для

вычисления соотношения интернет-продаж по сравнению со всеми продажами за 2006 год.

Следующая формула:

DAX

```
= SUMX(
    CALCULATETABLE(
        'InternetSales_USD',
        'DateTime'[CalendarYear] = 2006
    ),
    [SalesAmount_USD]
)
```

Это приводит к следующей таблице:

Развернуть таблицу

Метки строк	Интернет SalesAmount_USD	CalculateTable 2006 Internet Sales	Соотношение интернет-продаж до 2006
2005	$2,627,031,40	$5,681,440,58	0.46
2006	$5,681,440,58	$5,681,440,58	1.00
2007	$8,705,066,67	$5,681,440,58	1.53
2008	$9,041,288,80	$5,681,440,58	1.59
Общий итог	$26,054,827,45	$5,681,440,58	4.59

Связанное содержимое

- контекст фильтра
- функция CALCULATE (DAX)
- функции фильтрации (DAX)

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EARLIER

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает текущее значение указанного столбца во внешнем проходе оценки указанного столбца.

EARLIER полезно для вложенных вычислений, в которых требуется использовать определенное значение в качестве входных данных и производить вычисления на основе этого ввода. В Microsoft Excel такие вычисления можно выполнять только в контексте текущей строки; однако в DAX можно сохранить значение входных данных, а затем выполнить вычисление с помощью данных из всей таблицы.

EARLIER в основном используется в контексте вычисляемых столбцов.

Синтаксис

```DAX
EARLIER(<column>, <number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец или выражение, разрешающееся столбцу.
`num`	(Необязательно) Положительное число для прохождения внешней оценки. Следующий уровень оценки представлен 1; два уровня представлены 2 и т. д. Если опущено значение по умолчанию равно 1.

Возвращаемое значение

Текущее значение строки из `column` по `number` прохождения внешней оценки.

Исключения

Описание ошибок

Замечания

- `EARLIER` успешно выполняется, если есть контекст строки до начала проверки таблицы. В противном случае возвращается ошибка.
- Производительность `EARLIER` может быть медленной, так как теоретически может потребоваться выполнить ряд операций, близких к общему количеству строк (в столбце), то же число (в зависимости от синтаксиса выражения). Например, если в столбце есть 10 строк, может потребоваться примерно 100 операций; Если у вас есть 100 строк, может быть выполнено около 10 000 операций.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

> ⓘ **Примечание**
>
> На практике подсистема аналитики VertiPaq в памяти выполняет оптимизацию для уменьшения фактического количества вычислений, но при создании формул, связанных с рекурсией, следует соблюдать осторожность.

Пример

Чтобы проиллюстрировать использование EARLIER, необходимо создать сценарий, вычисляющий значение ранжирования, а затем использовать это значение ранжирования в других вычислениях.

Следующий пример основан на этой простой таблице, `ProductSubcategory`, которая показывает общее количество продаж для каждого ProductSubcategory.

Последняя таблица, включая столбец ранжирования, показана здесь.

Развернуть таблицу

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	ПодкатегорияRanking
18	Bib-Shorts	$156,167,88	18
26	Велосипедные стойки	$220,720,70	14
27	Стенды велосипедов	$35,628.69	30
28	Бутылки и клетки	$59,342,43	24
5	Нижние скобки	$48,643,47	27
6	Тормоза	$62,113.16	23

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	ПодкатегорияRanking
19	Прописные буквы	$47,934,54	28
7	Путы	$ 8,847.08	35
29	Химчистка	$16,882.62	32
8	Cranksets	$191,522.09	15
9	Derailleurs	$64,965.33	22
30	Крылья	$41,974.10	29
10	Вилки	$74,727,66	21
20	Перчатки	$228,353,58	12
4	Руле	$163,257.06	17
11	Гарнитуры	$57,659,99	25
31	Шлемы	$451,192.31	9
32	Пакеты гидратации	$96,893,78	20
21	Трикотажные изделия	$699,429,78	7
33	Светофор		36
34	Замки	$15,059,47	33
1	Горные велосипеды	$34,305,864,29	2
12	Горные кадры	$4511,170,68	4
35	Корзины		36
13	Педали	$140,422.20	19
36	Лакированные бальные туфли	$12,695.18	34
2	Дорожные велосипеды	$40,551,696,34	1
14	Дорожные кадры	$3,636,398,71	5
15	Седла	$52,526,47	26
22	Шорты	$385,707,80	10
23	Носки	$28,337,85	31
24	Трико	$189,179,37	16
37	Шины и трубы	$224,832,81	13
3	Экскурсии велосипедов	$ 13,334,864.18	3
16	Обзор кадров	$1,545,344.02	6

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	ПодкатегорияRanking
25	Жилеты	$240,990,04	11
17	Колеса	$648,240,04	8

Создание значения ранжирования

Одним из способов получения значения ранжирования для заданного значения в строке является подсчет количества строк в той же таблице, которая имеет значение больше (или меньше), чем сравниваемое. Этот метод возвращает пустое или нулевое значение для наибольшего значения в таблице, в то время как равные значения будут иметь то же значение ранжирования и следующее значение (после равных значений) будет иметь непоследовательных ранговых значений. См. приведенный ниже пример.

Новый вычисляемый столбец **SubCategorySalesRanking**создается с помощью следующей формулы.

DAX

```
= COUNTROWS(FILTER(ProductSubcategory,
EARLIER(ProductSubcategory[TotalSubcategorySales])
<ProductSubcategory[TotalSubcategorySales]))+1
```

Ниже описан метод вычисления более подробно.

1. Функция `EARLIER` получает значение *TotalSubcategorySales* для текущей строки в таблице. В этом случае, так как процесс запускается, это первая строка в таблице
2. `EARLIER([TotalSubcategorySales])` оценивается как $156,167,88, текущая строка во внешнем цикле.
3. Функция `FILTER` теперь возвращает таблицу, в которой все строки имеют значение *TotalSubcategorySales* больше $ 156,167,88 (которое является текущим значением для `EARLIER`).
4. Функция `COUNTROWS` подсчитывает строки отфильтрованной таблицы и назначает это значение новому вычисляемого столбцу в текущей строке плюс 1. Добавление 1 необходимо, чтобы не допустить, чтобы значение верхнего ранжирования стало пустым.
5. Формула вычисляемого столбца перемещается к следующей строке и повторяет шаги 1–4. Эти шаги повторяются до тех пор, пока не будет достигнут конец таблицы.

Функция `EARLIER` всегда получает значение столбца до текущей операции таблицы. Если перед этим необходимо получить значение из цикла, задайте второй аргумент 2.

Связанное содержимое

функциифильтрации

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EARLIEST

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Возвращает текущее значение указанного столбца во внешнем проходе оценки указанного столбца.

Синтаксис

DAX

```
EARLIEST(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Ссылка на столбец.

Возвращаемое значение

Столбец с удаленными фильтрами.

Замечания

- Функция EARLIEST аналогична EARLIER, но позволяет указать один дополнительный уровень рекурсии.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Текущий пример данных не поддерживает этот сценарий.

DAX

```
= EARLIEST(<column>)
```

Связанное содержимое

функциифильтрации

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FILTER

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу, представляющую подмножество другой таблицы или выражения.

Синтаксис

```DAX
FILTER(<table>,<filter>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, отфильтрованной. Таблица также может быть выражением, которое приводит к таблице.
`filter`	Логическое выражение, которое необходимо оценить для каждой строки таблицы. Например, `[Amount] > 0` или `[Region] = "France"`

Возвращаемое значение

Таблица, содержащая только отфильтрованные строки.

Замечания

- Вы можете использовать FILTER для уменьшения количества строк в таблице, с которыми вы работаете, и использовать только определенные данные в вычислениях. FILTER не используется независимо, но в качестве функции, внедренной в другие функции, требующие таблицы в качестве аргумента.
- Рекомендации по использованию FILTERсм. в статье Избегайте использования FILTER в качестве аргумента фильтра.

Используйте COUNTROWS вместо COUNT в DAX

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере создается отчет о продажах в Интернете за пределами США с помощью меры, которая фильтрует продажи в США, а затем срезается по календарным годам и категориям продуктов. Чтобы создать эту меру, вы фильтруете таблицу, интернет-продажи USD, используя территорию продаж, а затем используйте отфильтрованную таблицу в функции SUMX.

В этом примере выражение:

DAX

```
FILTER('InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])
<>"United States")
```

Возвращает таблицу, которая является подмножеством интернет-продаж минус все строки, принадлежащие территории продаж США. Функция RELATED заключается в том, что связывает ключ "Территория" в таблице "Продажи в Интернете" с SalesTerritoryCountry в таблице SalesTerritory.

В следующей таблице показано доказательство концепции меры, NON США Интернет-продаж, формулы, для которой приведены в приведенном ниже разделе кода. В таблице сравниваются все интернет-продажи с не США интернет-продаж, чтобы показать, что выражение фильтра работает, за исключением продаж Сша из вычисления.

Чтобы повторно создать эту таблицу, добавьте поле SalesTerritoryCountry в область **меток строк** области отчета или сводной таблицы.

Таблица 1. Сравнение общих продаж для США и всех остальных регионов

Развернуть таблицу

Метки строк	Интернет-продажи	Не США интернет-продажи
Австралия	$499,021,84	$499,021,84
Канада	$1,343,109.10	$1,343,109.10
Франция	$2,490,944,57	$2,490,944,57
Германия	$2775,195,60	$2775,195,60
Великобритания	$5,057,076,55	$5,057,076,55
США	$9,389,479,79	
Общий итог	$26,054,827,45	$16,665,347,67

В последней таблице отчета отображаются результаты при создании сводной таблицы с помощью меры, НЕ США Интернет-продаж. Добавьте поле CalendarYear в область меток строк сводной таблицы и добавьте поле ProductCategoryName в область меток столбцов .

Таблица 2. Сравнение продаж, отличных от США по категориям продуктов

Развернуть таблицу

Метки строк	Принадлежности	Велосипеды	Одежда	Общий итог
2005		$1,526,481,95		$1,526,481,95
2006		$354,744.04		$354,744.04
2007	$156,480,18	$5,640,106,05	$70,142,77	$5,866,729,00
2008	$228,159,45	$ 5,386,558.19	$102,675.04	$5,717,392,68
Общий итог	$384,639,633	$16,107,890,23	$ 172,817.81	$16,665,347,67

DAX

```
SUMX(FILTER('InternetSales_USD',
RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States")
     ,'InternetSales_USD'[SalesAmount_USD])
```

Связанное содержимое

Функция фильтрации функцийALLALLEXCEPT функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FIRSTNONBLANK

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает первое значение в столбце, `<column>`, отфильтрованное текущим контекстом, где выражение не является пустым.

Синтаксис

DAX

```
FIRSTNONBLANK(<column>,<expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Выражение столбца.
`expression`	Выражение, вычисляемое для пустых значений каждого значения `column`.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку с вычисленным первым значением.

Замечания

- Аргумент `column` может быть одним из следующих:

    - Ссылка на любой столбец.
    - Таблица с одним столбцом.

- Логическое выражение, определяющее таблицу с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Эта функция обычно используется для возврата первого значения столбца, для которого выражение не является пустым. Например, можно получить последнее значение, для которого были продажи продукта.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

LASTNONBLANK функцииСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FIRSTNONBLANKVALUE

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ⚠️ визуального вычисления

> ⓘ **Примечание**
>
> Эта функция не рекомендуется использовать в **визуальных вычислениях**, так как, скорее всего, возвращает бессмысленные результаты.

Вычисляет выражение, отфильтрованное по отсортированных значениям столбца, и возвращает первое значение выражения, которое не является пустым.

Синтаксис

DAX

```
FIRSTNONBLANKVALUE(<column>, <expression>)
```

Параметры

⛶ **Развернуть таблицу**

Срок	Определение
`column`	Столбец или выражение, возвращающее таблицу с одним столбцом.
`expression`	Выражение, вычисляемое для каждого значения `column`.

Возвращаемое значение

Первое непустое значение `expression`, соответствующее отсортированных значений `column`.

Замечания

- Аргумент столбца может быть одним из следующих:

- Ссылка на любой столбец.
- Таблица с одним столбцом.

- Эта функция отличается от FIRSTNONBLANK в том, что `column` добавляется в контекст фильтра для оценки `expression`.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX

DAX

```
EVALUATE
SUMMARIZECOLUMNS(
  DimProduct[Class],
  "FNBV",
  FIRSTNONBLANKVALUE(
    DimDate[Date],
    SUM(FactInternetSales[SalesAmount])
   )
)
```

Возвращает

Развернуть таблицу

DimProduct[class]	[FNBV]
L	699.0982
H	13778.24
M	1000.4375
	533.83

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

INDEX

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает строку в абсолютной позиции, указанной параметром позиции в указанной секции, отсортированной по указанному порядку. Если текущая секция не может быть выведена в одну секцию, может быть возвращено несколько строк.

Синтаксис

DAX

```
INDEX(<position>[, <relation> or <axis>][, <orderBy>][, <blanks>][,
<partitionBy>][, <matchBy>][, <reset>] )
```

Параметры

Развернуть таблицу

Срок	Определение
`position`	Абсолютная позиция (1 на основе), из которой можно получить данные: - `position` положительно: 1 — первая строка, 2 — вторая строка и т. д. - `position` является отрицательной: -1 является последней строкой, -2 является второй последней строкой и т. д. Если `position` выходит из границы, или ноль, или BLANK(), INDEX возвращает пустую таблицу. Это может быть любое выражение DAX, возвращающее скалярное значение.
`relation`	(Необязательно) Табличное выражение, из которого возвращаются выходные данные. Если указано, все столбцы в `partitionBy` должны поступать из нее или связанной таблицы. Если опущено: - `orderBy` необходимо явно указать. . Все выражения `orderBy` и `partitionBy` должны быть полными именами столбцов и поступать из одной таблицы. — по умолчанию `ALLSELECTED()` всех столбцов в `orderBy` и `partitionBy`.
`axis`	(Необязательно) Ось в визуальной форме. Доступно только в визуальных вычислениях и заменяет `relation`.

Срок	Определение
`orderBy`	(Необязательно) Предложение ORDERBY(), содержащее выражения, определяющие порядок сортировки каждой секции. Если опущено: - `relation` необходимо явно указать. — по умолчанию упорядочивается по каждому столбцу в `relation`, который еще не указан в `partitionBy`.
`blanks`	(Необязательно) Перечисление, определяющее, как обрабатывать пустые значения при сортировке `relation` или `axis`. Поддерживаемые значения: • `DEFAULT` (значение по умолчанию), где поведение числовых значений является пустыми значениями, упорядочены от нуля до отрицательных значений. Поведение строк является пустым значением, упорядоченным перед всеми строками, включая пустые строки. • `FIRST`, пустые поля всегда упорядочены в начале независимо от возрастания или убывающего порядка сортировки. • `LAST`, пустые всегда упорядочены в конце независимо от возрастания или убывания сортировки. Примечание. Если параметр `blanks` и пустые значения в функции ORDERBY() для отдельных выражений указываются, `blanks` для отдельного выражения orderBy имеет приоритет для соответствующего выражения orderBy, а выражения orderBy без `blanks` заданы будут учитывать параметр `blanks` родительской функции.
`partitionBy`	(Необязательно) Предложение PARTITIONBY(), содержащее столбцы, определяющие секционирование `relation`. Если опущено, `relation` рассматривается как одна секция.
`matchBy`	(Необязательно) Предложение MATCHBY(), содержащее столбцы, определяющие сопоставление данных и определение текущей строки.
`reset`	(Необязательно) Доступно только в визуальных вычислениях. Указывает, сбрасывается ли вычисление и на каком уровне иерархии столбцов визуальной фигуры. Допустимые значения: ссылка на поле столбца в текущей визуальной форме, `NONE` (по умолчанию), `LOWESTPARENT`, `HIGHESTPARENT`или целое число. Поведение зависит от целочисленного знака: — если нулевая или опущенная, вычисление не сбрасывается. Эквивалентно `NONE`. — если положительный, целое число определяет столбец, начиная с самого высокого, независимо от зерна. `HIGHESTPARENT` эквивалентно 1. — если отрицательно, целое число определяет столбец, начиная с самого низкого, относительно текущего зерна. `LOWESTPARENT` эквивалентно -1.

Возвращаемое значение

Строка в абсолютной позиции.

Замечания

Каждый столбец `partitionBy` и `matchBy` должен иметь соответствующее внешнее значение, чтобы определить "текущую секцию", с помощью следующего поведения:

- Если есть ровно один соответствующий внешний столбец, используется его значение.
- Если нет соответствующего внешнего столбца:
  - INDEX сначала определит все `partitionBy` и `matchBy` столбцы, не имеющие соответствующего внешнего столбца.
  - Для каждого сочетания существующих значений для этих столбцов в родительском контексте INDEXвычисляется INDEX и возвращается строка.
  - INDEXконечные выходные данные являются объединением этих строк.
- Если существует несколько соответствующих внешних столбцов, возвращается ошибка.

Если `matchBy` присутствует, INDEX попытается использовать `matchBy` и `partitionBy` столбцы для идентификации строки. Если `matchBy` отсутствует, и столбцы, указанные в `orderBy` и `partitionBy`, не могут однозначно идентифицировать каждую строку в `relation`:

- INDEX попытается найти наименьшее количество дополнительных столбцов, необходимых для уникальной идентификации каждой строки.
- Если такие столбцы можно найти, INDEX автоматически добавит эти новые столбцы в `orderBy`, и каждая секция сортируется с помощью этого нового набора столбцов OrderBy.
- Если такие столбцы не удается найти, возвращается ошибка.

Если возвращается пустая таблица:

- Соответствующее внешнее значение столбца PartitionBy не существует в `relation`.
- Значение `position` ссылается на положение, которое не существует в разделе.

Если INDEX используется в вычисляемом столбце, определенном в той же таблице, что и `relation`, и `orderBy` опущен, возвращается ошибка.

`reset` можно использовать только в визуальных вычислениях и не может использоваться в сочетании с `orderBy` или `partitionBy`. Если `reset` присутствует, `axis` можно указать, но `relation` невозможно.

Пример 1— вычисляемый столбец

Следующий запрос DAX:

```
DAX

EVALUATE INDEX(1, ALL(DimDate[CalendarYear]))
```

Возвращает следующую таблицу:

Развернуть таблицу

DimDate[CalendarYear]
2005

Пример 2— вычисляемый столбец

Следующий запрос DAX:

```
DAX

EVALUATE
SUMMARIZECOLUMNS (
    FactInternetSales[ProductKey],
    DimDate[MonthNumberOfYear],
    FILTER (
            VALUES(FactInternetSales[ProductKey]),
            [ProductKey] < 222
    ),
    "CurrentSales", SUM(FactInternetSales[SalesAmount]),
    "LastMonthSales",
    CALCULATE (
        SUM(FactInternetSales[SalesAmount]),
        INDEX(-1, ORDERBY(DimDate[MonthNumberOfYear]))
    )
)
ORDER BY [ProductKey], [MonthNumberOfYear]
```

Возвращает следующую таблицу:

Развернуть таблицу

FactInternetSales[ProductKey]	DimDate[MonthNumberOfYear]	[CurrentSales]	[LastMonthSales]
214	1	5423.45	8047.7

FactInternetSales[ProductKey]	DimDate[MonthNumberOfYear]	[CurrentSales]	[LastMonthSales]
214	2	4968.58	8047.7
214	3	5598.4	8047.7
214	4	5073.55	8047.7
214	5	5248.5	8047.7
214	6	7487.86	8047.7
214	7	7382.89	8047.7
214	8	6543.13	8047.7
214	9	6788.06	8047.7
214	10	6858.04	8047.7
214	11	8607.54	8047.7
214	12	8047.7	8047.7
217	1	5353.47	7767.78
217	2	4268.78	7767.78
217	3	5773.35	7767.78
217	4	5738.36	7767.78
217	5	6158.24	7767.78
217	6	6998	7767.78
217	7	5563.41	7767.78
217	8	5913.31	7767.78
217	9	5913.31	7767.78
217	10	6823.05	7767.78
217	11	6683.09	7767.78
217	12	7767.78	7767.78

Пример 3. Визуальное вычисление

Следующие визуальные вычисления DAX запросах:

```
```

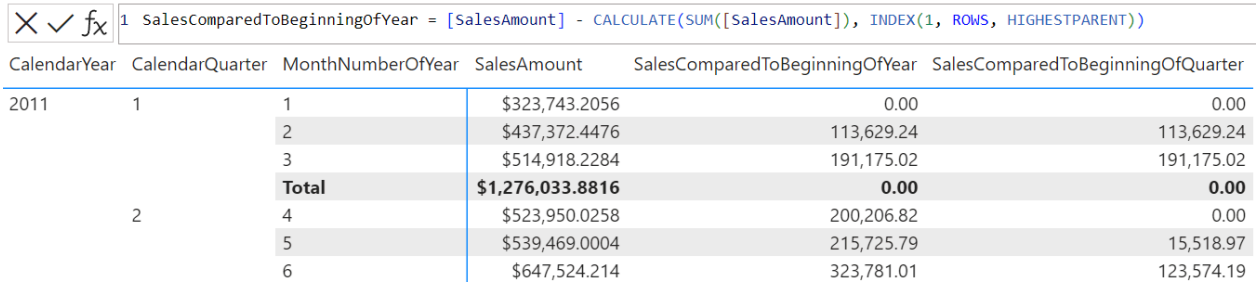
DAX

```
SalesComparedToBeginningOfYear = [SalesAmount] -
CALCULATE(SUM([SalesAmount]), INDEX(1, ROWS, HIGHESTPARENT))

SalesComparedToBeginningOfQuarter = [SalesAmount] -
CALCULATE(SUM([SalesAmount]), INDEX(1, , -1))
```

Улучшение таблицы таким образом, чтобы она содержала, за каждый месяц:
- общий объем продаж;
- разница в первом месяце соответствующего года;
- и разница в первом месяце соответствующего квартала.

Снимок экрана ниже: визуальная матрица и первое выражение визуального вычисления:



```
1 SalesComparedToBeginningOfYear = [SalesAmount] - CALCULATE(SUM([SalesAmount]), INDEX(1, ROWS, HIGHESTPARENT))
```

CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesComparedToBeginningOfYear	SalesComparedToBeginningOfQuarter
2011	1	1	$323,743.2056	0.00	0.00
		2	$437,372.4476	113,629.24	113,629.24
		3	$514,918.2284	191,175.02	191,175.02
		Total	**$1,276,033.8816**	**0.00**	**0.00**
	2	4	$523,950.0258	200,206.82	0.00
		5	$539,469.0004	215,725.79	15,518.97
		6	$647,524.214	323,781.01	123,574.19

Связанное содержимое

OFFSET ORDERBY PARTITIONBY WINDOW RANK ROWNUMBER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

KEEPFILTERS

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Изменяет способ применения фильтров при оценке функции CALCULATE или CALCULATETABLE.

Синтаксис

```DAX
KEEPFILTERS(<expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Любое выражение.

Возвращаемое значение

Таблица значений.

Замечания

- Вы используете KEEPFILTERS в контексте CALCULATE и CALCULATETABLE функций, чтобы переопределить стандартное поведение этих функций.
- По умолчанию аргументы фильтра в таких функциях, как CALCULATE, используются в качестве контекста для оценки выражения, а такие аргументы фильтра для CALCULATE заменить все существующие фильтры по тем же столбцам. Новый контекст, затронутый аргументом фильтра для CALCULATE, влияет только на существующие фильтры по столбцам, упомянутым в качестве части аргумента фильтра. Фильтры для столбцов, отличных от

упомянутых в аргументах CALCULATE или других связанных функций, остаются в силе и неуправляемы.

- Функция KEEPFILTERS позволяет изменить это поведение. При использовании KEEPFILTERSвсе существующие фильтры в текущем контексте сравниваются со столбцами в аргументах фильтра, а пересечение этих аргументов используется в качестве контекста для оценки выражения. Чистый эффект для любого столбца заключается в том, что оба набора аргументов применяются: как аргументы фильтра, используемые в CALCULATE, так и фильтры в аргументах функции KEEPFILTER. Другими словами, в то время как фильтры CALCULATE заменяют текущий контекст, KEEPFILTERS добавляет фильтры в текущий контекст.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере описаны некоторые распространенные сценарии, демонстрирующие использование функции KEEPFILTERS в рамках формулы CALCULATE или CALCULATETABLE.

Первые три выражения получают простые данные для сравнения:

- Интернет-продажи для штата Вашингтон.

- Интернет-продажи для штатов Вашингтон и Орегон (оба штата объединяются).

- Интернет-продажи для штата Вашингтон и провинции Британская Колумбия (оба региона объединяются).

Четвертое выражение вычисляет интернет-продажи для Вашингтона и Орегона, а фильтр для Вашингтона и Британской Колумбии применяется.

Следующее выражение вычисляет интернет-продажи для Вашингтона и Орегона, но использует KEEPFILTERS; Фильтр для Вашингтона и Британской Колумбии является частью предыдущего контекста.

DAX

```
EVALUATE ROW(
  "$$ in WA"
```

```
    , CALCULATE('Internet Sales'[Internet Total Sales]
                , 'Geography'[State Province Code]="WA"
      )
, "$$ in WA and OR"
    , CALCULATE('Internet Sales'[Internet Total Sales]
                , 'Geography'[State Province Code]="WA"
                  || 'Geography'[State Province Code]="OR"
      )
, "$$ in WA and BC"
    , CALCULATE('Internet Sales'[Internet Total Sales]
                , 'Geography'[State Province Code]="WA"
                  || 'Geography'[State Province Code]="BC"
      )
, "$$ in WA and OR ??"
    , CALCULATE(
          CALCULATE('Internet Sales'[Internet Total Sales]
                    ,'Geography'[State Province Code]="WA"
                      || 'Geography'[State Province Code]="OR"
          )
          , 'Geography'[State Province Code]="WA"
            || 'Geography'[State Province Code]="BC"
      )
, "$$ in WA !!"
    , CALCULATE(
          CALCULATE('Internet Sales'[Internet Total Sales]
                   , KEEPFILTERS('Geography'[State Province Code]="WA"
                                 || 'Geography'[State Province Code]="OR"
                     )
          )
          , 'Geography'[State Province Code]="WA"
            || 'Geography'[State Province Code]="BC"
      )
)
```

Если это выражение вычисляется для примера базы данных AdventureWorks DW, получены следующие результаты.

Развернуть таблицу

Столбец	Ценность
`[$$ in WA]`	$ 2,467,248,34
`[$$ in WA and OR]`	$ 3,638,239,88
`[$$ in WA and BC]`	$ 4,422,588,44
`[$$ in WA and OR ??]`	$ 3,638,239,88
`[$$ in WA !!]`	$ 2,467,248,34

> **Примечание**
>
> Приведенные выше результаты были отформатированы в таблицу вместо одной строки для образовательных целей.

Сначала изучите выражение, `[$$ in WA and OR ??]`. Вы можете задаться вопросом, как эта формула может вернуть значение для продаж в Вашингтоне и Орегоне, так как внешнее CALCULATE выражение включает фильтр для Вашингтона и Британской Колумбии. Ответ заключается в том, что поведение по умолчанию CALCULATE переопределяет внешние фильтры в "Geography"[Государственный код провинции] и заменяет свои собственные аргументы фильтра, так как фильтры применяются к одному столбцу.

Затем изучите выражение, `[$$ in WA !!]`. Вы можете задаться вопросом, как эта формула может вернуть значение для продаж в Вашингтоне и ничего другого, так как фильтр аргументов включает Орегон и внешнее выражение CALCULATE включает фильтр в Вашингтоне и Британской Колумбии. Ответ заключается в том, что KEEPFILTERS изменяет поведение CALCULATE по умолчанию и добавляет дополнительный фильтр. Так как используется пересечение фильтров, теперь внешний фильтр `'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="BC")` добавляется в аргумент фильтра `'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="OR"`.. Так как оба фильтра применяются к одному столбцу, результирующий фильтр `'Geography'[State Province Code]="WA"` является фильтром, применяемым при оценке выражения.

Связанное содержимое

Функция фильтрации функцийCALCULATECALCULATETABLE функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LASTNONBLANK

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает последнее значение в столбце `column`, отфильтрованное текущим контекстом, где выражение не является пустым.

Синтаксис

```
DAX

LASTNONBLANK(<column>,<expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Выражение столбца.
`expression`	Выражение, вычисляемое для пустых значений каждого значения `column`.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку с вычисляемого последнего значения.

Замечания

- Аргумент `column` может быть одним из следующих:

    - Ссылка на любой столбец.
    - Таблица с одним столбцом.
    - Логическое выражение, определяющее таблицу с одним столбцом

- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Эта функция обычно используется для возврата последнего значения столбца, для которого выражение не является пустым. Например, можно получить последнее значение, для которого были продажи продукта.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

FIRSTNONBLANK функцииСтатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LASTNONBLANKVALUE

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ⚠️ визуального вычисления

> ⓘ **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Вычисляет выражение, отфильтрованное по отсортированных значениям столбца, и возвращает последнее значение выражения, которое не является пустым.

Синтаксис

DAX

```
LASTNONBLANKVALUE(<column>, <expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец или выражение, возвращающее таблицу с одним столбцом.
`expression`	Выражение, вычисляемое для каждого значения `column`.

Возвращаемое значение

Последнее непустое значение `expression`, соответствующее отсортированных значениям `column`.

Замечания

- Аргумент столбца может быть одним из следующих:

- Ссылка на любой столбец.
- Таблица с одним столбцом.

- Эта функция отличается от LASTNONBLANK в том, что `column` добавляется в контекст фильтра для оценки `expression`.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX

DAX

```
EVALUATE
SUMMARIZECOLUMNS(
  DimProduct[Class],
  "LNBV",
  LASTNONBLANKVALUE(
    DimDate[Date],
    SUM(FactInternetSales[SalesAmount])
   )
)
```

Возвращает

Развернуть таблицу

DimProduct[class]	[LNBV]
L	132.44
H	137.6
M	84.97
	2288.6

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ПОИСКА

Статья • 20.05.2025

Применимо к: ⊗ Вычисляемый столбец ⊗ Вычисляемая таблица ⊗ Измерять ✓ Визуальное вычисление

Возвращает значение или вычисляет выражение из визуальной матрицы с помощью абсолютной навигации. Фильтры можно указать для любой оси в визуальной матрице. Все опущенные фильтры выводятся из контекста. Если одно значение не может быть определено, возвращается ошибка.

Синтаксис

```
DAX

LOOKUP(<expression>, <colref>, <expression>[, <colref>, <expression>]...)
```

Параметры

Развернуть таблицу

Срок	Определение
выражение	Выражение для вычисления.
colref	(Необязательно) Отфильтрованный столбец.
выражение	(Необязательно) Значение для фильтрации.

Возвращаемое значение

Значение **выражения** после применения фильтров.

Если совпадения нет, возвращается ошибка.

Если несколько строк соответствуют фильтрам, возвращается ошибка.

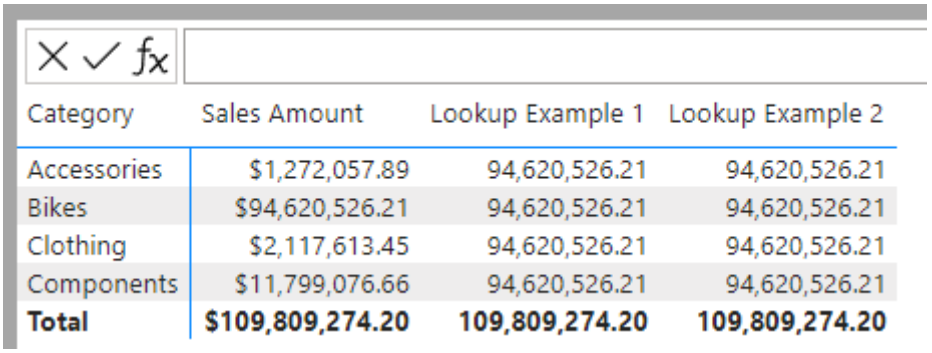
Пример 1

В этом примере LOOKUP извлекает сумму продаж для категории велосипедов. Первый аргумент может быть столбцом или скалярным выражением.

DAX

```
Lookup Example 1 = LOOKUP(SUM([Sales Amount]),  [Category], "Bikes")
Lookup Example 2 = LOOKUP([Sales Amount], [Category], "Bikes")
```

На снимка екрана ниже показана матрица с двумя визуальными вычислениями.



Category	Sales Amount	Lookup Example 1	Lookup Example 2
Accessories	$1,272,057.89	94,620,526.21	94,620,526.21
Bikes	$94,620,526.21	94,620,526.21	94,620,526.21
Clothing	$2,117,613.45	94,620,526.21	94,620,526.21
Components	$11,799,076.66	94,620,526.21	94,620,526.21
Total	**$109,809,274.20**	**109,809,274.20**	**109,809,274.20**

Пример 2

В этом примере визуальный расчет **LookupExample2** извлекает сумму продаж за финансовый год **FY2018**. Обратите внимание, что фильтр кварталов из строки будет использоваться, так как он не указан явно, что приводит к тому, что результаты не возвращаются для **FY2019 Q1** до **FY2019 Q4 с момента сочетания FY2018** и **FY2019 Q1** с **FY2019 Q4** не существует. **LookupExample3** явно указывает квартал, поэтому не учитывает текущий квартал строки.

DAX

```
LookupExample2 = LOOKUP([Sales Amount], [Fiscal Year], "FY2018")
LookupExample3 = LOOKUP([Sales Amount], [Fiscal Year], "FY2018", [Fiscal Quarter],
"FY2018 Q1")
```

Fiscal Year	Sales Amount	LookupExample2	LookupExample3
⊟ **FY2018**	**$23,860,891.17**	**23,860,891.17**	**6,005,207.32**
FY2018 Q1	$6,005,207.32	6,005,207.32	6,005,207.32
FY2018 Q2	$5,923,348.19	5,923,348.19	6,005,207.32
FY2018 Q3	$5,965,011.12	5,965,011.12	6,005,207.32
FY2018 Q4	$5,967,324.53	5,967,324.53	6,005,207.32
⊟ **FY2019**	**$34,070,108.50**	**23,860,891.17**	**6,005,207.32**
FY2019 Q1	$10,192,098.13		6,005,207.32
FY2019 Q2	$8,392,458.01		6,005,207.32
FY2019 Q3	$6,769,487.90		6,005,207.32
FY2019 Q4	$8,716,064.45		6,005,207.32
⊞ **FY2020**	**$51,878,274.54**	**23,860,891.17**	**6,005,207.32**
Total	**$109,809,274.20**	**23,860,891.17**	**6,005,207.32**

Связанный контент

LOOKUPWITHTOTALS

LOOKUPVALUE

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает значение строки, которая соответствует всем критериям, заданным одним или несколькими условиями поиска.

Синтаксис

DAX

```
LOOKUPVALUE (
    <result_columnName>,
    <search_columnName>,
    <search_value>
    [, <search2_columnName>, <search2_value>]…
    [, <alternateResult>]
)
```

Параметры

Развернуть таблицу

Срок	Определение
`result_columnName`	Имя существующего столбца, содержащего возвращаемое значение. Это не может быть выражением.
`search_columnName`	Имя существующего столбца. Он может находиться в той же таблице, что и result_columnName или в связанной таблице. Это не может быть выражением. Можно указать несколько пар search_columnName и search_value.
`search_value`	Значение для поиска в search_columnName. Можно указать несколько пар search_columnName и search_value.
`alternateResult`	(Необязательно) Значение, возвращаемое при фильтрации контекста для result_columnName до нуля или нескольких отдельных значений. Если функция не указана, функция возвращает BLANK, если result_columnName фильтруется до нуля значений или ошибки, если в контексте для result_columnName имеется несколько отдельных значений.

Возвращаемое значение

Значение `result_columnName` в строке, где все пары `search_columnName` и `search_value` имеют точное совпадение.

Если совпадение не соответствует всем значениям поиска, возвращается BLANK или `alternateResult` (если указано). Другими словами, функция не возвращает значение подстановки, если только некоторые из критериев соответствуют.

Если несколько строк соответствуют значениям поиска и значениям в `result_columnName` для этих строк идентичны, возвращается это значение. Однако если `result_columnName` возвращает разные значения, возвращается ошибка или `alternateResult` (если указано).

Замечания

- Если между таблицей, содержащей столбец результатов и таблицы, содержащие столбцы поиска, в большинстве случаев используется функция RELATED вместо LOOKUPVALUE эффективнее и обеспечивает более высокую производительность.
- Можно указать несколько пар `search_columnName` и `search_value`.
- Параметры `search_value` и `alternateResult` вычисляются перед итерацию функции по строкам таблицы поиска.
- Избегайте использования функций ISERROR или IFERROR для записи ошибки, возвращаемой LOOKUPVALUE. Если некоторые входные данные функции приводят к ошибке, если не удается определить одно выходное значение, предоставление параметра `alternateResult` является наиболее надежным и самым высоким способом обработки ошибки.
- Параметр `alternateResult` возвращает ошибку, если указан в вычисляемом столбце Power Pivot.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

В этом примере LOOKUPVALUE используется для поиска средней ставки для валюты, используемой для оплаты заказа на день размещения заказа:

```
DAX

Exchange Rate =
LOOKUPVALUE (
    'Currency Rate'[Average Rate],
    'Currency Rate'[CurrencyKey], [CurrencyKey],
    'Currency Rate'[DateKey], [OrderDateKey]
)
```

Для поиска даты и валюты заказа требуется средняя ставка для правильной даты и валюты. OrderDateKey и CurrencyKey — это ключи, используемые для поиска средней ставки в таблице "Ставка валюты".

Вы можете использовать курс обмена для вычисления суммы продаж в локальной валюте с помощью следующих способов:

```
DAX

Sales Amount Local Currency = [Sales Amount] * [Exchange Rate]
```

Пример 2

В этом примере следующий вычисляемый столбец, определенный в таблице sales Sales, использует функцию для возврата значений каналов из таблицы заказа на продажу.

```
DAX

CHANNEL =
LOOKUPVALUE (
    'Sales Order'[Channel],
    'Sales Order'[SalesOrderLineKey],
    [SalesOrderLineKey]
)
```

Однако в этом случае существует связь между **заказами на продажу** и таблицами **sales**, то более эффективно использовать функцию RELATED.

```
DAX
```

```
CHANNEL = RELATED('Sales Order'[Channel])
```

Связанное содержимое

функция RELATED (DAX)информационных функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LOOKUPWITHTOTALS

Статья • 20.05.2025

Применимо к: ⊗ Вычисляемый столбец ⊗ Вычисляемая таблица ⊗ Измерять ✓ Визуальное вычисление

Возвращает значение или вычисляет выражение из визуальной матрицы с помощью абсолютной навигации. Фильтры можно указать для любой оси в визуальной матрице. Все опущенные фильтры обрабатываются как общие. Если одно значение не может быть определено, возвращается ошибка.

Синтаксис

DAX

```
LOOKUPWITHTOTALS(<expression>, <colref>, <expression>[, <colref>,
<expression>]...)
```

Параметры

Развернуть таблицу

Срок	Определение
выражение	Выражение для вычисления
colref	(Необязательно) Отфильтрованный столбец.
выражение	(Необязательно) Значение для фильтрации.

Возвращаемое значение

Значение **выражения** после применения фильтров.

Если совпадения нет, возвращается ошибка.

Если несколько строк соответствуют фильтрам, возвращается ошибка.

Пример 1

Визуальное вычисление **Lookupwithtotals** извлекает значения для **FY2018** и все кварталы, так как значение для квартала не указано в качестве аргумента. С другой стороны,

визуальное вычисление **подстановки** учитывает текущий контекст. Таким образом, результаты отличаются, и визуальный расчет **подстановки** возвращает **сумму продаж** для **FY2018** за каждый квартал. Это также объясняет, почему результаты для визуального вычисления **подстановки** пусты, так как не существует сочетания **FY2019 Q1** с **FY2019 Q4** и **FY2018**.

DAX

```
LOOKUPWITHTOTALExample1 = LOOKUPWITHTOTALS([Sales Amount], [Fiscal Year],
"FY2018")

LookupExample1 = LOOKUP([Sales Amount], [Fiscal Year], "FY2018")
```

На снимка экрана ниже показана матрица с двумя визуальными вычислениями.

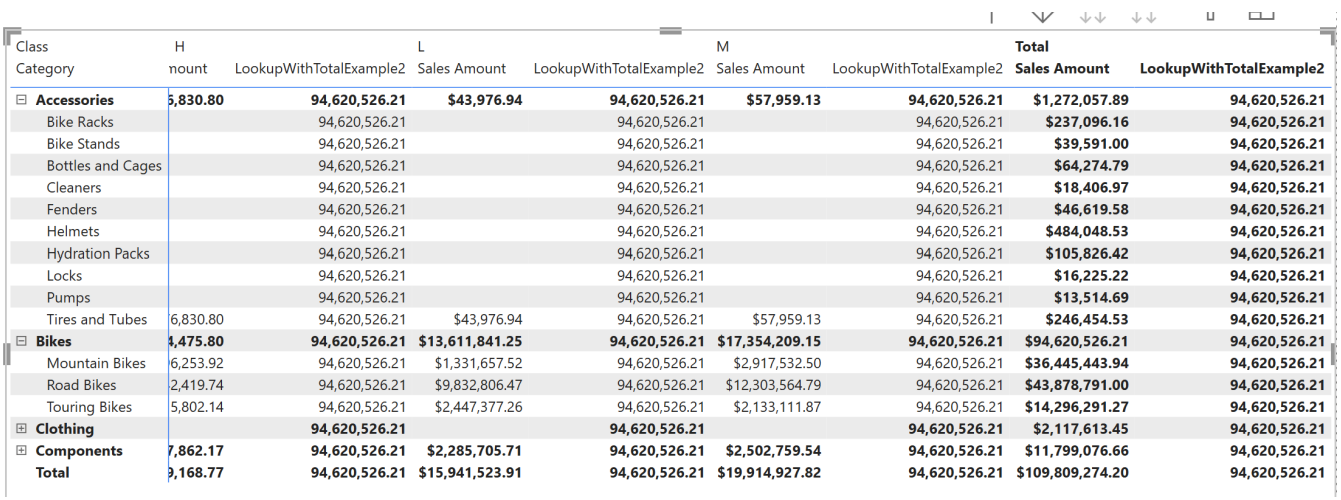
Fiscal Year	Sales Amount	LOOKUPWITHTOTALExample1	LookupExample1
⊟ **FY2018**	**$23,860,891.17**	**23,860,891.17**	**23,860,891.17**
FY2018 Q1	$6,005,207.32	23,860,891.17	6,005,207.32
FY2018 Q2	$5,923,348.19	23,860,891.17	5,923,348.19
FY2018 Q3	$5,965,011.12	23,860,891.17	5,965,011.12
FY2018 Q4	$5,967,324.53	23,860,891.17	5,967,324.53
⊟ **FY2019**	**$34,070,108.50**	**23,860,891.17**	**23,860,891.17**
FY2019 Q1	$10,192,098.13	23,860,891.17	
FY2019 Q2	$8,392,458.01	23,860,891.17	
FY2019 Q3	$6,769,487.90	23,860,891.17	
FY2019 Q4	$8,716,064.45	23,860,891.17	
⊞ **FY2020**	**$51,878,274.54**	**23,860,891.17**	**23,860,891.17**
Total	**$109,809,274.20**	**23,860,891.17**	**23,860,891.17**

Пример 2

В этом примере **LookupWithTotalExample2** извлекает сумму продаж для категории велосипедов. Обратите внимание, что использует общую сумму для других измерений, так как они не были указаны. Другими словами, эти измерения не фильтруются. Полезно, если мы хотим выполнить сравнение, так как результат заблокирован на одно значение.

DAX

```
LookupWithTotalExample2 = lookupWithTotals([Sales Amount], [Category], "Bikes")
```



Class Category	H nount	LookupWithTotalExample2	L Sales Amount	LookupWithTotalExample2	M Sales Amount	LookupWithTotalExample2	**Total** **Sales Amount**	**LookupWithTotalExample2**
⊟ **Accessories**	**6,830.80**	**94,620,526.21**	**$43,976.94**	**94,620,526.21**	**$57,959.13**	**94,620,526.21**	**$1,272,057.89**	**94,620,526.21**
Bike Racks		94,620,526.21		94,620,526.21		94,620,526.21	**$237,096.16**	**94,620,526.21**
Bike Stands		94,620,526.21		94,620,526.21		94,620,526.21	**$39,591.00**	**94,620,526.21**
Bottles and Cages		94,620,526.21		94,620,526.21		94,620,526.21	**$64,274.79**	**94,620,526.21**
Cleaners		94,620,526.21		94,620,526.21		94,620,526.21	**$18,406.97**	**94,620,526.21**
Fenders		94,620,526.21		94,620,526.21		94,620,526.21	**$46,619.58**	**94,620,526.21**
Helmets		94,620,526.21		94,620,526.21		94,620,526.21	**$484,048.53**	**94,620,526.21**
Hydration Packs		94,620,526.21		94,620,526.21		94,620,526.21	**$105,826.42**	**94,620,526.21**
Locks		94,620,526.21		94,620,526.21		94,620,526.21	**$16,225.22**	**94,620,526.21**
Pumps		94,620,526.21		94,620,526.21		94,620,526.21	**$13,514.69**	**94,620,526.21**
Tires and Tubes	6,830.80	94,620,526.21	$43,976.94	94,620,526.21	$57,959.13	94,620,526.21	**$246,454.53**	**94,620,526.21**
⊟ **Bikes**	**4,475.80**	**94,620,526.21**	**$13,611,841.25**	**94,620,526.21**	**$17,354,209.15**	**94,620,526.21**	**$94,620,526.21**	**94,620,526.21**
Mountain Bikes	6,253.92	94,620,526.21	$1,331,657.52	94,620,526.21	$2,917,532.50	94,620,526.21	**$36,445,443.94**	**94,620,526.21**
Road Bikes	2,419.74	94,620,526.21	$9,832,806.47	94,620,526.21	$12,303,564.79	94,620,526.21	**$43,878,791.00**	**94,620,526.21**
Touring Bikes	5,802.14	94,620,526.21	$2,447,377.26	94,620,526.21	$2,133,111.87	94,620,526.21	**$14,296,291.27**	**94,620,526.21**
⊞ **Clothing**		**94,620,526.21**		**94,620,526.21**		**94,620,526.21**	**$2,117,613.45**	**94,620,526.21**
⊞ **Components**	**7,862.17**	**94,620,526.21**	**$2,285,705.71**	**94,620,526.21**	**$2,502,759.54**	**94,620,526.21**	**$11,799,076.66**	**94,620,526.21**
Total	**9,168.77**	**94,620,526.21**	**$15,941,523.91**	**94,620,526.21**	**$19,914,927.82**	**94,620,526.21**	**$109,809,274.20**	**94,620,526.21**

Связанный контент

Поиск

MATCHBY

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

В функциях окна определяет столбцы, используемые для определения сопоставления данных и идентификации текущей строки.

Синтаксис

DAX

```
MATCHBY ( [<matchBy_columnName>[, matchBy_columnName [, …]]] )
```

Параметры

Развернуть таблицу

Срок	Определение
`matchBy_columnName`	(Необязательно) Имя существующего столбца, используемого для идентификации текущей строки в `relation` функции окна. RELATED() также можно использовать для ссылки на столбец в таблице, связанной с `relation`.

Возвращаемое значение

Эта функция не возвращает значение.

Замечания

Эту функцию можно использовать только в выражении функции окна.

Пример

См. OFFSET.

Связанное содержимое

Понимание ORDERBY, PARTITIONBYи MATCHBY функцийINDEXOFFSETORDERBYPARTITIONBYWINDOWRANKROWNUMBER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MOVINGAVERAGE

Статья • 22.01.2025

применяется: ⊗ вычисляемый столбец ⊗ вычисляемой таблицы ⊗ измерение ✓ визуального вычисления

Возвращает скользящее среднее значение, вычисляемое по заданной оси визуальной матрицы. То есть среднее значение заданного столбца, вычисляемого за последние `windowSize` строк.

Синтаксис

DAX

```
MOVINGAVERAGE ( <column>, <windowSize>[, <includeCurrent>][, <axis>][,
<blanks>][, <reset>] )
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, предоставляющий значение для каждого элемента.
`windowSize`	Количество строк, включаемых в вычисление. Должно быть константным значением.
`includeCurrent`	(Необязательно) Логическое значение, указывающее, следует ли включать текущую строку в диапазон. Значение по умолчанию — True.
`axis`	(Необязательно) Ссылка на оси, направление, по которому будет вычисляться скользящая средняя.
`blanks`	(Необязательно) Перечисление, определяющее, как обрабатывать пустые значения при сортировке `axis`. Поддерживаемые значения: • `DEFAULT` (значение по умолчанию), где поведение числовых значений является пустыми значениями, упорядочены от нуля до отрицательных значений. Поведение строк является пустым значением, упорядоченным перед всеми строками, включая пустые строки.

Срок	Определение
	• `FIRST`, пустые поля всегда упорядочены в начале независимо от возрастания или убывающего порядка сортировки. • `LAST`, пустые всегда упорядочены в конце независимо от возрастания или убывания сортировки.
`reset`	(Необязательно) Указывает, сбрасывается ли вычисление и на каком уровне иерархии столбцов визуальной фигуры. Допустимые значения: ссылка на поле столбца в текущей визуальной форме, `NONE` (по умолчанию), `LOWESTPARENT`, `HIGHESTPARENT` или целое число. Поведение зависит от целочисленного знака: — если нулевая или опущенная, вычисление не сбрасывается. Эквивалентно `NONE`. — если положительный, целое число определяет столбец, начиная с самого высокого, независимо от зерна. `HIGHESTPARENT` эквивалентно 1. — если отрицательно, целое число определяет столбец, начиная с самого низкого, относительно текущего зерна. `LOWESTPARENT` эквивалентно -1.

Возвращаемое значение

Скалярное значение, скользящее среднее в текущем элементе оси.

Замечания

Эту функцию можно использовать только в визуальных вычислениях.

Параметры `includeCurrent`, `axis`, `blanks` и `reset` могут быть опущены.

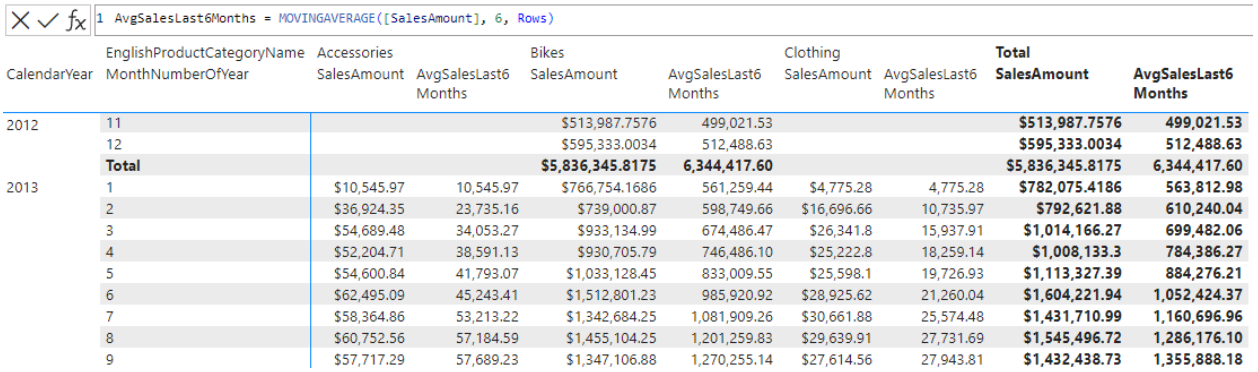
Пример 1

Учитывая таблицу, которая суммирует общий объем продаж для каждой категории продукта и календарного месяца, следующий запрос DAX добавляет столбец со средним показателем общих продаж для этой категории за последние 6 месяцев:

DAX

```
AvgSalesLast6Months = MOVINGAVERAGE([SalesAmount], 6, Rows)
```

Снимок экрана ниже: визуальная матрица и выражение визуального вычисления:



```
1 AvgSalesLast6Months = MOVINGAVERAGE([SalesAmount], 6, Rows)
```

CalendarYear	EnglishProductCategoryName MonthNumberOfYear	Accessories SalesAmount	 AvgSalesLast6 Months	Bikes SalesAmount	 AvgSalesLast6 Months	Clothing SalesAmount	 AvgSalesLast6 Months	**Total SalesAmount**	**AvgSalesLast6 Months**
2012	11			$513,987.7576	499,021.53			**$513,987.7576**	**499,021.53**
	12			$595,333.0034	512,488.63			**$595,333.0034**	**512,488.63**
	Total			**$5,836,345.8175**	**6,344,417.60**			**$5,836,345.8175**	**6,344,417.60**
2013	1	$10,545.97	10,545.97	$766,754.1686	561,259.44	$4,775.28	4,775.28	**$782,075.4186**	**563,812.98**
	2	$36,924.35	23,735.16	$739,000.87	598,749.66	$16,696.66	10,735.97	**$792,621.88**	**610,240.04**
	3	$54,689.48	34,053.27	$933,134.99	674,486.47	$26,341.8	15,937.91	**$1,014,166.27**	**699,482.06**
	4	$52,204.71	38,591.13	$930,705.79	746,486.10	$25,222.8	18,259.14	**$1,008,133.3**	**784,386.27**
	5	$54,600.84	41,793.07	$1,033,128.45	833,009.55	$25,598.1	19,726.93	**$1,113,327.39**	**884,276.21**
	6	$62,495.09	45,243.41	$1,512,801.23	985,920.92	$28,925.62	21,260.04	**$1,604,221.94**	**1,052,424.37**
	7	$58,364.86	53,213.22	$1,342,684.25	1,081,909.26	$30,661.88	25,574.48	**$1,431,710.99**	**1,160,696.96**
	8	$60,752.56	57,184.59	$1,455,104.25	1,201,259.83	$29,639.91	27,731.69	**$1,545,496.72**	**1,286,176.10**
	9	$57,717.29	57,689.23	$1,347,106.88	1,270,255.14	$27,614.56	27,943.81	**$1,432,438.73**	**1,355,888.18**

Пример 2

Учитывая ту же таблицу, следующий DAX запрос добавляет столбец со средним показателем общего объема продаж для этой категории за предыдущие 12 месяцев (не включая текущий месяц):

DAX

```
AvgSalesPrev12Months = MOVINGAVERAGE([SalesAmount], 12, FALSE, Rows, KEEP)
```

См. также

INDEX ORDERBY PARTITIONBY RUNNINGSUM WINDOW

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

OFFSET

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает одну строку, расположенную либо до, либо после *текущей строки* в той же таблице, с помощью заданного смещения. Если текущая строка не может быть выведена в одну строку, может быть возвращено несколько строк.

Синтаксис

DAX

```
OFFSET ( <delta>[, <relation> or <axis>][, <orderBy>][, <blanks>][,
<partitionBy>][, <matchBy>][, <reset>] )
```

Параметры

Развернуть таблицу

Срок	Определение
`delta`	Число строк до (отрицательное значение) или после (положительное значение) текущей строки, из которой необходимо получить данные. Это может быть любое выражение DAX, возвращающее скалярное значение.
`relation`	(Необязательно) Табличное выражение, из которого возвращается выходная строка. Если указано, все столбцы в `partitionBy` должны поступать из нее или связанной таблицы. Если опущено: - `orderBy` необходимо явно указать. . Все выражения `orderBy` и `partitionBy` должны быть полными именами столбцов и поступать из одной таблицы. — по умолчанию ALLSELECTED() всех столбцов в `orderBy` и `partitionBy`.
`axis`	(Необязательно) Ось в визуальной форме. Доступно только в визуальных вычислениях и заменяет `relation`.
`orderBy`	(Необязательно) Предложение ORDERBY(), содержащее выражения, определяющие порядок сортировки каждой секции. Если опущено: - `relation` необходимо явно указать.

Срок	Определение
	— по умолчанию упорядочивается по каждому столбцу в `relation`, который еще не указан в `partitionBy`.
`blanks`	(Необязательно) Перечисление, определяющее, как обрабатывать пустые значения при сортировке `relation` или `axis`. Поддерживаемые значения: • `DEFAULT` (значение по умолчанию), где поведение числовых значений является пустыми значениями, упорядочены от нуля до отрицательных значений. Поведение строк является пустым значением, упорядоченным перед всеми строками, включая пустые строки. • `FIRST`, пустые поля всегда упорядочены в начале независимо от возрастания или убывающего порядка сортировки. • `LAST`, пустые всегда упорядочены в конце независимо от возрастания или убывания сортировки. Примечание. Если параметр `blanks` и пустые значения в функции ORDERBY() для отдельных выражений указываются, `blanks` для отдельного выражения orderBy имеет приоритет для соответствующего выражения orderBy, а выражения orderBy без `blanks` заданы будут учитывать параметр `blanks` родительской функции.
`partitionBy`	(Необязательно) Предложение PARTITIONBY(), содержащее столбцы, определяющие секционирование `relation`. Если опущено, `relation` рассматривается как одна секция.
`matchBy`	(Необязательно) Предложение MATCHBY(), содержащее столбцы, определяющие сопоставление данных и определение текущей строки.
`reset`	(Необязательно) Доступно только в визуальных вычислениях. Указывает, сбрасывается ли вычисление и на каком уровне иерархии столбцов визуальной фигуры. Допустимые значения: ссылка на поле столбца в текущей визуальной форме, `NONE` (по умолчанию), `LOWESTPARENT`, `HIGHESTPARENT` или целое число. Поведение зависит от целочисленного знака: — если нулевая или опущенная, вычисление не сбрасывается. Эквивалентно `NONE`. — если положительный, целое число определяет столбец, начиная с самого высокого, независимо от зерна. `HIGHESTPARENT` эквивалентно 1. — если отрицательно, целое число определяет столбец, начиная с самого низкого, относительно текущего зерна. `LOWESTPARENT` эквивалентно -1.

Возвращаемое значение

Одна или несколько строк из `relation`.

Замечания

За исключением столбцов, добавленных DAX табличными функциями, каждый столбец в `relation`, если `matchBy` отсутствует, или каждый столбец в `matchBy` и `partitionBy`, если `matchBy` присутствует, должен иметь соответствующее внешнее значение, чтобы помочь определить текущую строку, в которой следует работать, с помощью следующего поведения:

- Если есть ровно один соответствующий внешний столбец, используется его значение.
- Если соответствующий внешний столбец отсутствует, то:
  - OFFSET сначала определит все столбцы, у которых нет соответствующего внешнего столбца.
  - Для каждого сочетания существующих значений для этих столбцов в родительском контексте OFFSETвычисляется OFFSET и возвращается строка.
  - OFFSETконечные выходные данные являются объединением этих строк.
- Если существует несколько соответствующих внешних столбцов, возвращается ошибка.

Если все столбцы `relation` были добавлены DAX табличными функциями, возвращается ошибка.

Если `matchBy` присутствует, OFFSET попытается использовать `matchBy` и `partitionBy` столбцы для идентификации строки. Если `matchBy` отсутствует, и столбцы, указанные в `orderBy` и `partitionBy`, не могут однозначно идентифицировать каждую строку в `relation`, а затем:

- OFFSET попытается найти наименьшее количество дополнительных столбцов, необходимых для уникальной идентификации каждой строки.
- Если такие столбцы можно найти, OFFSET автоматически добавит эти новые столбцы в `orderBy`, и каждая секция сортируется с помощью этого нового набора столбцов OrderBy.
- Если такие столбцы не удается найти, возвращается ошибка.

Если возвращается пустая таблица:

- Соответствующее внешнее значение столбца OrderBy или PartitionBy не существует в `relation`.
- Значение `delta` приводит к переходу к строке, которая не существует в разделе.

Если OFFSET используется в вычисляемом столбце, определенном в той же таблице, что и `relation`, и `orderBy` опущен, возвращается ошибка.

`reset` можно использовать только в визуальных вычислениях и не может использоваться в сочетании с `orderBy` или `partitionBy`. Если `reset` присутствует, `axis` можно указать, но `relation` невозможно.

Пример 1— вычисляемый столбец

Следующий запрос DAX:

DAX

```
DEFINE
VAR vRelation = SUMMARIZECOLUMNS (
                    DimProductCategory[EnglishProductCategoryName],
                    DimDate[CalendarYear],
                    "CurrentYearSales", SUM(FactInternetSales[SalesAmount])
                  )
EVALUATE
ADDCOLUMNS (
    vRelation,
    "PreviousYearSales",
    SELECTCOLUMNS(
        OFFSET (
                -1,
                vRelation,
                ORDERBY([CalendarYear]),
                PARTITIONBY([EnglishProductCategoryName])
        ),
        [CurrentYearSales]
    )
)
```

Возвращает таблицу, которая суммирует общий объем продаж для каждой категории продукта и календарного года, а также общий объем продаж для этой категории в предыдущем году.

Пример 2. Мера

Следующий запрос DAX:

DAX

```
DEFINE
MEASURE DimProduct[CurrentYearSales] = SUM(FactInternetSales[SalesAmount])
MEASURE DimProduct[PreviousYearSales] =
CALCULATE(SUM(FactInternetSales[SalesAmount]), OFFSET(-1, ,
ORDERBY(DimDate[CalendarYear])))
EVALUATE
SUMMARIZECOLUMNS (
    DimDate[CalendarYear],
    "CurrentYearSales", DimProduct[CurrentYearSales],
    "PreviousYearSales", DimProduct[PreviousYearSales]
)
```

Использует OFFSET() в меру для возврата таблицы, которая суммирует общий объем продаж за каждый календарный год и общий объем продаж за предыдущий год.

Пример 3. Вычисляемый столбец

Следующий запрос DAX:

DAX

```
EVALUATE
ADDCOLUMNS (
    FactInternetSales,
    "Previous Sales Amount",
        SELECTCOLUMNS (
            OFFSET (
                -1,
                FactInternetSales,
                ORDERBY ( FactInternetSales[SalesAmount], DESC ),
                PARTITIONBY ( FactInternetSales[ProductKey] ),
                MATCHBY( FactInternetSales[SalesOrderNumber],
FactInternetSales[SalesOrderLineNumber] )
            ),
            FactInternetSales[SalesAmount]
        )
)
```

Возвращает таблицу FactInternetSales с добавлением столбца, указывающего, для каждой продажи, суммы предыдущей продажи одного и того же продукта в порядке убывания объема продаж, при этом текущая продажа определяется его SalesOrderNumber и SalesOrderLineNumber. Без MATCHBYзапрос вернет ошибку, так как в таблице FactInternetSales отсутствуют ключевые столбцы.

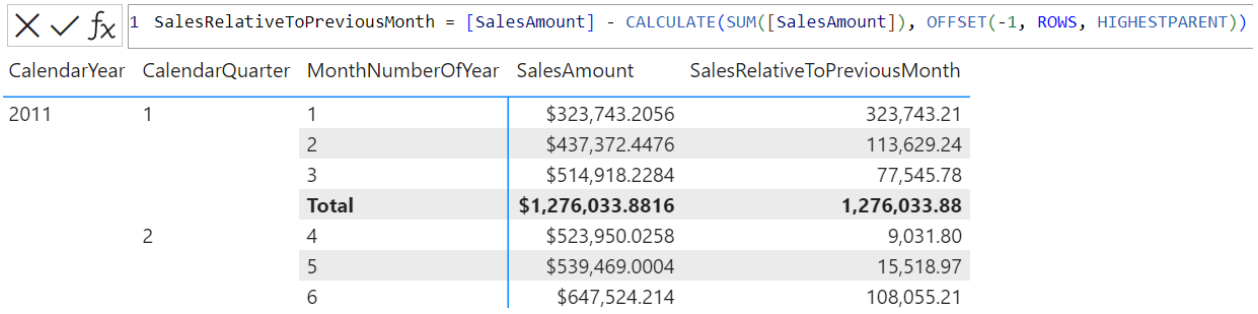
Пример 4. Визуальное вычисление

Следующий визуальный расчет DAX запроса:

```DAX
SalesRelativeToPreviousMonth = [SalesAmount] - CALCULATE(SUM([SalesAmount]),
OFFSET(-1, ROWS, HIGHESTPARENT))
```

Возвращает разницу в общем объеме продаж между каждым месяцем и предыдущим в течение одного года.

Снимок экрана ниже: визуальная матрица и выражение визуального вычисления:



CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesRelativeToPreviousMonth
2011	1	1	$323,743.2056	323,743.21
		2	$437,372.4476	113,629.24
		3	$514,918.2284	77,545.78
		Total	**$1,276,033.8816**	**1,276,033.88**
	2	4	$523,950.0258	9,031.80
		5	$539,469.0004	15,518.97
		6	$647,524.214	108,055.21

Связанное содержимое

INDEX ORDERBY PARTITIONBY MATCHBY WINDOW RANK ROWNUMBER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ORDERBY

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Определяет выражения, определяющие порядок сортировки в каждой секции функции окна.

Синтаксис

DAX

```
ORDERBY ( [<orderBy_expression>[, <order>[, <orderBy_expression>[, <order>]] …]] )
```

Параметры

Развернуть таблицу

Срок	Определение
`orderBy_expression`	(Необязательно) Любое скалярное выражение, которое будет использоваться для сортировки данных в каждой секции функции окна.
`order`	(Необязательно) Двух частей формы "`OrderDirection` [`BlankHandling`]". `OrderDirection` указывает, как сортировать `orderBy_expression` значения (т. е. возрастание или убывание). Допустимые значения: `DESC`. Альтернативное значение: `0` (ноль)/`FALSE`. Сортировка по убыванию значений `orderBy_expression`. `ASC`. Альтернативное значение: `1`/`TRUE`. Сортировка по возрастанию значений `orderBy_expression`. Это значение по умолчанию, если `order` опущен. `BlankHandling` часть является необязательной. Он определяет порядок упорядочения пустых значений. Допустимые значения: `BLANKS DEFAULT`. Это значение по умолчанию. Поведение числовых значений — пустые значения упорядочены от нуля до отрицательных значений. Поведение строк является пустым значением, упорядоченным перед всеми строками, включая пустые строки. `BLANKS FIRST`. Пустые поля всегда упорядочены в начале независимо от возрастания или убывающего порядка сортировки.

Срок	Определение
	`BLANKS LAST`. Пустые поля всегда упорядочены в конце независимо от возрастания или убывающего порядка сортировки.

Возвращаемое значение

Эта функция не возвращает значение.

Замечания

Эту функцию можно использовать только в выражении функции окна.

Пример

См. OFFSET.

Связанное содержимое

Понимание ORDERBY, PARTITIONBYи MATCHBY функцийINDEXOFFSETPARTITIONBYMATCHBYWINDOWRANKROWNUMBER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PARTITIONBY

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Определяет столбцы, используемые для секционирования параметра `relation` функции окна.

Синтаксис

DAX

```
PARTITIONBY ( [<partitionBy_columnName>[, partitionBy_columnName [, …]]] )
```

Параметры

Развернуть таблицу

Срок	Определение
`partitionBy_columnName`	(Необязательно) Имя существующего столбца, используемого для секционирования `relation` функции окна. RELATED() также можно использовать для ссылки на столбец в таблице, связанной с `relation`.

Возвращаемое значение

Эта функция не возвращает значение.

Замечания

Эту функцию можно использовать только в выражении функции окна.

Пример

См. OFFSET.

Связанное содержимое

Понимание ORDERBY, PARTITIONBYи MATCHBY функцийINDEXOFFSETORDERBYMATCHBYWINDOWRANKROWNUMBER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RANGE

Статья • 24.04.2025

Применимо к: ⊗ Вычисляемый столбец ⊗ Вычисляемая таблица ⊗ Измерять ✓ Визуальное вычисление

Возвращает интервал строк в заданной оси относительно текущей строки. Этот интервал будет состоять из последних `step` строк до текущего или первых `step` строк после текущего.

Синтаксис

```
DAX

RANGE ( <step>[, <includeCurrent>][, <axis>][, <blanks>][, <reset>] )
```

Параметры

Развернуть таблицу

Срок	Определение
`step`	Число строк до (отрицательное значение) или после (положительное значение) текущей строки, включаемой в диапазон. Должно быть константным значением. — если отрицательно, окно будет содержать последние -step строки до текущей строки. — в противном случае окно будет содержать строки первого шага после текущей строки.
`includeCurrent`	(Необязательно) Логическое значение, указывающее, следует ли включать текущую строку в диапазон. Значение по умолчанию — True.
`axis`	(Необязательно) Ссылка на оси, направление, по которому будет создан интервал.
`blanks`	(Необязательно) Перечисление, определяющее, как обрабатывать пустые значения при сортировке `axis`. Поддерживаемые значения: • `DEFAULT` (значение по умолчанию), где поведение числовых значений является пустыми значениями, упорядочены от нуля до отрицательных значений. Поведение строк является пустым значением, упорядоченным перед всеми строками, включая пустые строки. • `FIRST`, пустые поля всегда упорядочены в начале независимо от возрастания или убывающего порядка сортировки.

Срок	Определение
	• `LAST`, пустые всегда упорядочены в конце независимо от возрастания или убывания сортировки.
`reset`	(Необязательно) Указывает, сбрасывается ли вычисление и на каком уровне иерархии столбцов визуальной фигуры. Допустимые значения: ссылка на поле столбца в текущей визуальной форме, `NONE` (по умолчанию), `LOWESTPARENT`, `HIGHESTPARENT` или целое число. Поведение зависит от целочисленного знака: — если нулевая или опущенная, вычисление не сбрасывается. Эквивалентно `NONE`. — если положительный, целое число определяет столбец, начиная с самого высокого, независимо от зерна. `HIGHESTPARENT` эквивалентно 1. — если отрицательно, целое число определяет столбец, начиная с самого низкого, относительно текущего зерна. `LOWESTPARENT` эквивалентно -1.

Возвращаемое значение

Интервал строк данных.

Замечания

Эту функцию можно использовать только в визуальных вычислениях.

Параметры `includeCurrent`, `axis`, `blanks` и `reset` могут быть опущены.

Если значение `reset` является абсолютным (т. е. положительным целым числом `HIGHESTPARENT` или ссылкой на поле), а вычисление вычисляется на целевом уровне иерархии или выше, вычисление сбрасывается для каждого отдельного элемента. То есть функция оценивается в секции, содержащей только этот конкретный элемент.

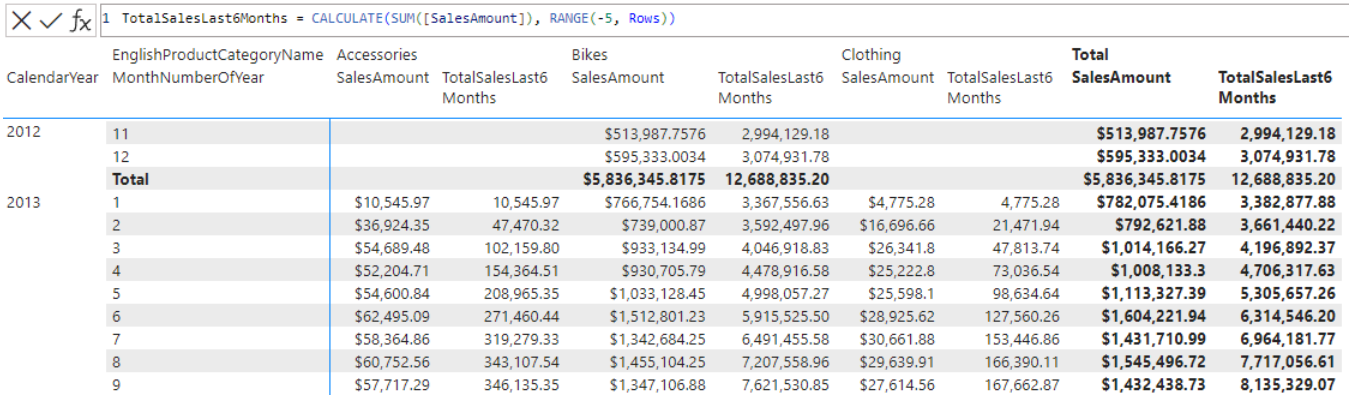
Пример 1

Учитывая таблицу, которая суммирует общий объем продаж для каждой категории продукта и месяца, следующий запрос DAX добавляет столбец с общим объемом продаж за последние 6 месяцев:

DAX

```
TotalSalesLast6Months = CALCULATE(SUM([SalesAmount]), RANGE(-5, Rows))
```

Снимок экрана ниже: визуальная матрица и выражение визуального вычисления:



CalendarYear	EnglishProductCategoryName MonthNumberOfYear	Accessories SalesAmount	 TotalSalesLast6 Months	Bikes SalesAmount	 TotalSalesLast6 Months	Clothing SalesAmount	 TotalSalesLast6 Months	**Total SalesAmount**	**TotalSalesLast6 Months**
2012	11			$513,987.7576	2,994,129.18			**$513,987.7576**	**2,994,129.18**
	12			$595,333.0034	3,074,931.78			**$595,333.0034**	**3,074,931.78**
	Total			**$5,836,345.8175**	**12,688,835.20**			**$5,836,345.8175**	**12,688,835.20**
2013	1	$10,545.97	10,545.97	$766,754.1686	3,367,556.63	$4,775.28	4,775.28	**$782,075.4186**	**3,382,877.88**
	2	$36,924.35	47,470.32	$739,000.87	3,592,497.96	$16,696.66	21,471.94	**$792,621.88**	**3,661,440.22**
	3	$54,689.48	102,159.80	$933,134.99	4,046,918.83	$26,341.8	47,813.74	**$1,014,166.27**	**4,196,892.37**
	4	$52,204.71	154,364.51	$930,705.79	4,478,916.58	$25,222.8	73,036.54	**$1,008,133.3**	**4,706,317.63**
	5	$54,600.84	208,965.35	$1,033,128.45	4,998,057.27	$25,598.1	98,634.64	**$1,113,327.39**	**5,305,657.26**
	6	$62,495.09	271,460.44	$1,512,801.23	5,915,525.50	$28,925.62	127,560.26	**$1,604,221.94**	**6,314,546.20**
	7	$58,364.86	319,279.33	$1,342,684.25	6,491,455.58	$30,661.88	153,446.86	**$1,431,710.99**	**6,964,181.77**
	8	$60,752.56	343,107.54	$1,455,104.25	7,207,558.96	$29,639.91	166,390.11	**$1,545,496.72**	**7,717,056.61**
	9	$57,717.29	346,135.35	$1,347,106.88	7,621,530.85	$27,614.56	167,662.87	**$1,432,438.73**	**8,135,329.07**

Пример 2

Учитывая ту же таблицу, следующий запрос DAX добавляет столбец с общим объемом продаж в течение следующих 12 месяцев, не включая текущий месяц:

DAX

```
TotalSalesFollowingYear = CALCULATE(SUM([SalesAmount]), RANGE(12, FALSE, Rows, KEEP))
```

См. также

INDEX ORDERBY PARTITIONBY WINDOW

RANK

Статья • 24.04.2025

Применимо к: Вычисляемый столбец Вычисляемая таблица Измерять Визуальное вычисление

Возвращает рейтинг текущего контекста в указанной секции, отсортированный по указанному порядку. Если совпадение не удается найти, то ранг пустой.

Синтаксис

DAX

```
RANK ( [<ties>][, <relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>]
[, <matchBy>][, <reset>] )
```

Параметры

Развернуть таблицу

Срок	Определение
`ties`	(Необязательно) Определяет способ обработки ранжирования при привязке двух или более строк. Если задано, поддерживаемое значение имеет значение DENSE или SKIP. Если опущено: — значение по умолчанию для SKIP
`relation`	(Необязательно) Табличное выражение, из которого возвращается выходная строка. При использовании в визуальных вычислениях этот параметр принимает ось в визуальной форме. Если указано, все столбцы в `orderBy` и `partitionBy` должны поступать из него. Если опущено: - `orderBy` необходимо явно указать. — все столбцы `orderBy` и `partitionBy` должны быть полностью квалифицированы и получены из одной таблицы. — по умолчанию ALLSELECTED() всех столбцов в `orderBy` и `partitionBy`.
`orderBy`	(Необязательно) Предложение ORDERBY(), содержащее столбцы, определяющие порядок сортировки каждой секции. Если опущено: - `relation` необходимо явно указать. — по умолчанию упорядочивается по каждому столбцу в `relation`, который еще не указан в `partitionBy`.

Срок	Определение
`blanks`	(Необязательно) Перечисление, определяющее, как обрабатывать пустые значения при сортировке `relation` или `axis`. Поддерживаемые значения: • `DEFAULT` (значение по умолчанию), где поведение числовых значений является пустыми значениями, упорядочены от нуля до отрицательных значений. Поведение строк является пустым значением, упорядоченным перед всеми строками, включая пустые строки. • `FIRST`, пустые поля всегда упорядочены в начале независимо от возрастания или убывающего порядка сортировки. • `LAST`, пустые всегда упорядочены в конце независимо от возрастания или убывания сортировки. Обратите внимание, что при `blanks` указании параметра и пустых `blanks` значений в функции () дляORDERBY отдельных выражений orderBy выражение имеет приоритет для соответствующего выражения orderBy, а выражения orderBy не `blanks` будут учитывать `blanks` параметр родительской функции.
`partitionBy`	(Необязательно) Предложение (PARTITIONBY) с столбцами, определяющими `relation` способ секционирования. Если опущено, `relation` рассматривается как одна секция.
`matchBy`	(Необязательно) Предложение (MATCHBY) с столбцами, определяющими сопоставление данных и определение текущей строки.
`reset`	(Необязательно) Доступно только в визуальных вычислениях. Указывает, сбрасывается ли вычисление и на каком уровне иерархии столбцов визуальной фигуры. Допустимые значения: ссылка на поле столбца в текущей визуальной форме, `NONE` (по умолчанию), `LOWESTPARENT`, `HIGHESTPARENT`или целое число. Поведение зависит от целочисленного знака: — если нулевая или опущенная, вычисление не сбрасывается. Эквивалентно `NONE`. — если положительный, целое число определяет столбец, начиная с самого высокого, независимо от зерна. `HIGHESTPARENT` эквивалентно 1. — если отрицательно, целое число определяет столбец, начиная с самого низкого, относительно текущего зерна. `LOWESTPARENT` эквивалентно -1.

Возвращаемое значение

Номер ранга для текущего контекста.

Замечания

- Каждый столбец `orderBy`, `partitionBy`и `matchBy` должен иметь соответствующее внешнее значение, чтобы помочь определить текущую строку, с помощью

следующего поведения:
    - Если есть ровно один соответствующий внешний столбец, используется его значение.
    - Если соответствующий внешний столбец отсутствует, то:
        - RANK сначала определит все `orderBy`, `partitionBy` и `matchBy` столбцы, не имеющие соответствующего внешнего столбца.
        - Для каждого сочетания существующих значений для этих столбцов в родительском контексте RANK вычисляется RANK и возвращается строка.
        - RANKконечные выходные данные — это номер ранга.

- Если `matchBy` присутствует, RANK попытается использовать столбцы в `matchBy` и `partitionBy` для idenfity текущей строки.

- Если столбцы, указанные в `orderBy` и `partitionBy`, не могут однозначно идентифицировать каждую строку в `relation`, то два или более строк могут иметь одинаковый рейтинг, а ранжирование определяется параметром связей.

- RANK возвращает пустое значение для общих строк. Рекомендуется тщательно протестировать выражение.

- RANK не сравнивается с RANKX, так как SUM сравнивается с SUMX.

- `reset` можно использовать только в визуальных вычислениях и не может использоваться в сочетании с `orderBy` или `partitionBy`. Если `reset` присутствует, `axis` можно указать, но `relation` невозможно.

- Если значение `reset` является абсолютным (т. е. положительным целым числом `HIGHESTPARENT` или ссылкой на поле), а вычисление вычисляется на целевом уровне иерархии или выше, вычисление сбрасывается для каждого отдельного элемента. То есть функция оценивается в секции, содержащей только этот конкретный элемент.

Пример 1— вычисляемый столбец

Следующий запрос DAX:

DAX

```
EVALUATE
ADDCOLUMNS(
    'DimGeography',
    "Rank",
    RANK(
        DENSE,
        'DimGeography',
```

```
        ORDERBY(
            'DimGeography'[StateProvinceName], desc,
            'DimGeography'[City], asc),
        LAST,
        PARTITIONBY(
            'DimGeography'[EnglishCountryRegionName])))
ORDER BY [EnglishCountryRegionName] asc, [StateProvinceName] desc, [City] asc
```

Возвращает таблицу, которая ранжирует каждую географию с одинаковым значением EnglishCountryRegionName, по stateProvinceName и City. Пустые значения столбцов `orderBy` отсортированы в конце.

Пример 2. Визуальное вычисление

Следующие визуальные вычисления DAX запросах:

DAX

```
SalesRankWithinYear = RANK(DENSE, ORDERBY([SalesAmount], DESC),
PARTITIONBY([CalendarYear]))

SalesRankAllHistory = RANK(DENSE, ORDERBY([SalesAmount], DESC))
```

Создайте два столбца, которые ранжировать каждый месяц по общему объему продаж, как в течение каждого года, так и всей истории.

Снимок экрана ниже: визуальная матрица и первое выражение визуального вычисления:



CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesRankWithinYear	SalesRankAllHistory
2011	1	1	$323,743.2056	12.00	37.00
		2	$437,372.4476	11.00	34.00
		3	$514,918.2284	10.00	25.00
		Total	**$1,276,033.8816**	**4.00**	**11.00**
	2	4	$523,950.0258	9.00	24.00
		5	$539,469.0004	8.00	23.00
		6	$647,524.214	4.00	16.00
		Total	**$1,710,943.2402**	**3.00**	**7.00**
	3	7	$648,343.8132	3.00	15.00
		8	$599,874.2732	7.00	20.00
		9	$609,121.865	6.00	19.00
		Total	**$1,857,339.9514**	**2.00**	**6.00**
	4	10	$655,170.3294	2.00	14.00
		11	$716,748.7178	1.00	13.00
		12	$636,253.2642	5.00	17.00
		Total	**$2,008,172.3114**	**1.00**	**5.00**
	Total		**$6,852,489.3846**	**1.00**	**2.00**

Связанное содержимое

INDEX ORDERBY PARTITIONBY WINDOW ROWNUMBER

REMOVEFILTERS

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Очистка фильтров из указанных таблиц или столбцов.

Синтаксис

DAX

```
REMOVEFILTERS([<table> | <column>[, <column>[, <column>[,…]]]])
```

Параметры

Развернуть таблицу

Термин	Определение
table	Таблица, в которой требуется очистить фильтры.
столбец	Столбец, на который требуется очистить фильтры.

Возвращаемое значение

Недоступно См. примечания.

Замечания

- REMOVEFILTERS можно использовать только для очистки фильтров, но не для возврата таблицы.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

запрос DAX

DAX

```
DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],
CALCULATE([TotalSales],REMOVEFILTERS()))

EVALUATE
    SUMMARIZECOLUMNS(
      ROLLUPADDISSUBTOTAL(DimProductCategory[EnglishProductCategoryName],
"IsGrandTotal"),
      "TotalSales", [TotalSales],
      "%Sales", [%Sales]
    )
ORDER BY
  [IsGrandTotal] DESC, [TotalSales] DESC
```

Возвраты

Развернуть таблицу

DimProductCategory[EnglishProductCategoryName]	[IsGrandTotal]	[TotalSales]	[%Sales]
Строка1	Истинный	29358677.2207	1
Велосипеды	Ложный	28318144.6507	0.964557920570538
Аксессуары	Ложный	700759.96	0.023868921434441
Clothing	Ложный	339772.61	0.0115731579950215

Пример 2

запрос DAX

DAX

```
DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],
CALCULATE([TotalSales],REMOVEFILTERS(DimProductSubcategory[EnglishProductSubcategory
Name])))

EVALUATE
    SUMMARIZECOLUMNS(
      DimProductCategory[EnglishProductCategoryName],
      DimProductSubcategory[EnglishProductSubcategoryName],
      "TotalSales", [TotalSales],
      "%Sales", [%Sales]
    )
ORDER BY
  DimProductCategory[EnglishProductCategoryName] ASC,
  DimProductSubcategory[EnglishProductSubcategoryName] ASC
```

Возвраты

Развернуть таблицу

DimProductCategory [EnglishProductCategoryName]	DimProductSubcategory [EnglishProductSubcategoryName]	[TotalSales]	[%Sales]
Аксессуары	Багажники для велосипедов	39360	0.05616759
Аксессуары	Велосипедные стойки	39591	0.05649723
Аксессуары	Бутылки и клетки	56798.19	0.08105228
Аксессуары	Очистители	7218.6	0.0103011
Аксессуары	Крылья	46619.58	0.06652717
Аксессуары	Шлемы	225335.6	0.3215589
Аксессуары	Питьевые системы	40307.67	0.05751994
Аксессуары	Шины и трубы	245529.32	0.35037578
Велосипеды	Горные велосипеды	9952759.564	0.35146228
Велосипеды	Шоссейные велосипеды	14520584.04	0.51276608
Велосипеды	Туристические велосипеды	3844801.05	0.13577164
Clothing	Каски	19688.1	0.05794493
Clothing	Перчатки	35020.7	0.10307099
Clothing	Майки	172950.68	0.5090189
Clothing	Шорты	71319.81	0.20990453
Clothing	Носки	5106.32	0.01502864
Clothing	Жилеты	35687	0.10503201

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ROWNUMBER

Статья • 24.04.2025

Применимо к: Вычисляемый столбец Вычисляемая таблица Измерять Визуальное вычисление

Возвращает уникальное ранжирование текущего контекста в указанной секции, отсортированного по указанному порядку. Если совпадение не удается найти, строка пуста.

Синтаксис

DAX

```
ROWNUMBER ( [<relation> or <axis>][, <orderBy>][, <blanks>][, <partitionBy>][,
<matchBy>][, <reset>] )
```

Параметры

Развернуть таблицу

Термин	Определение
`relation`	(Необязательно) Табличное выражение, из которого возвращается выходная строка. При использовании в визуальных вычислениях этот параметр принимает ось в визуальной форме. Если указано, все столбцы в `orderBy` и `partitionBy` должны поступать из него. Если опущено: - `orderBy` необходимо явно указать. — все столбцы `orderBy` и `partitionBy` должны быть полностью квалифицированы и получены из одной таблицы. — по умолчанию ALLSELECTED() всех столбцов в `orderBy` и `partitionBy`.
`orderBy`	(Необязательно) Предложение ORDERBY(), содержащее столбцы, определяющие порядок сортировки каждой секции. Если опущено: - `relation` необходимо явно указать. — по умолчанию упорядочивается по каждому столбцу в `relation`, который еще не указан в `partitionBy`.
`blanks`	(Необязательно) Перечисление, определяющее, как обрабатывать пустые значения при сортировке `relation` или `axis`. Поддерживаемые значения:

Термин	Определение
	• `DEFAULT` (значение по умолчанию), где поведение числовых значений является пустыми значениями, упорядочены от нуля до отрицательных значений. Поведение строк является пустым значением, упорядоченным перед всеми строками, включая пустые строки. • `FIRST`, пустые поля всегда упорядочены в начале независимо от возрастания или убывающего порядка сортировки. • `LAST`, пустые всегда упорядочены в конце независимо от возрастания или убывания сортировки. Обратите внимание, что при `blanks` указании параметра и пустых `blanks` значений в функции () дляORDERBY отдельных выражений orderBy выражение имеет приоритет для соответствующего выражения orderBy, а выражения orderBy не `blanks` будут учитывать `blanks` параметр родительской функции.
`partitionBy`	(Необязательно) Предложение (PARTITIONBY) с столбцами, определяющими `relation` способ секционирования. Если опущено, `relation` рассматривается как одна секция.
`matchBy`	(Необязательно) Предложение (MATCHBY) с столбцами, определяющими сопоставление данных и определение текущей строки.
`reset`	(Необязательно) Доступно только в визуальных вычислениях. Указывает, сбрасывается ли вычисление и на каком уровне иерархии столбцов визуальной фигуры. Допустимые значения: ссылка на поле столбца в текущей визуальной форме, `NONE` (по умолчанию), `LOWESTPARENT`, `HIGHESTPARENT` или целое число. Поведение зависит от целочисленного знака: — если нулевая или опущенная, вычисление не сбрасывается. Эквивалентно `NONE`. — если положительный, целое число определяет столбец, начиная с самого высокого, независимо от зерна. `HIGHESTPARENT` эквивалентно 1. — если отрицательно, целое число определяет столбец, начиная с самого низкого, относительно текущего зерна. `LOWESTPARENT` эквивалентно -1.

Возвращаемое значение

Номер строки для текущего контекста.

Замечания

Каждый столбец `orderBy`, `partitionBy` и `matchBy` должен иметь соответствующее внешнее значение, чтобы помочь определить текущую строку, с помощью следующего поведения:

- Если есть ровно один соответствующий внешний столбец, используется его значение.
- Если соответствующий внешний столбец отсутствует, то:

    - ROWNUMBER сначала определит все `orderBy`, `partitionBy` и `matchBy` столбцы, не имеющие соответствующего внешнего столбца.
    - Для каждого сочетания существующих значений для этих столбцов в родительском контексте ROWNUMBER вычисляется ROWNUMBER и возвращается строка.
    - ROWNUMBERконечные выходные данные являются объединением этих строк.
- Если существует несколько соответствующих внешних столбцов, возвращается ошибка.

Если `matchBy` присутствует, ROWNUMBER попытается использовать столбцы в `matchBy` и `partitionBy` для idenfity текущей строки. Если столбцы, указанные в `orderBy` и `partitionBy`, не могут однозначно идентифицировать каждую строку в `relation`, то:

- ROWNUMBER попытается найти наименьшее количество дополнительных столбцов, необходимых для уникальной идентификации каждой строки.
- Если такие столбцы можно найти, ROWNUMBER
    - Попробуйте найти минимальное количество дополнительных столбцов, необходимых для уникальной идентификации каждой строки.
    - Автоматически добавляйте эти новые столбцы в предложение `orderBy`.
    - Сортируйте каждую секцию с помощью этого нового набора столбцов orderBy.
- Если такие столбцы не удается найти и функция обнаруживает связь во время выполнения, возвращается ошибка.

`reset` можно использовать только в визуальных вычислениях и не может использоваться в сочетании с `orderBy` или `partitionBy`. Если `reset` присутствует, `axis` можно указать, но `relation` невозможно.

Если значение `reset` является абсолютным (т. е. положительным целым числом `HIGHESTPARENT` или ссылкой на поле), а вычисление вычисляется на целевом уровне иерархии или выше, вычисление сбрасывается для каждого отдельного элемента. То есть функция оценивается в секции, содержащей только этот конкретный элемент.

Пример 1— вычисляемый столбец

Следующий запрос DAX:

DAX

```
EVALUATE
ADDCOLUMNS(
    'DimGeography',
    "UniqueRank",
    ROWNUMBER(
```

```
        'DimGeography',
        ORDERBY(
            'DimGeography'[StateProvinceName], desc,
            'DimGeography'[City], asc),
        PARTITIONBY(
            'DimGeography'[EnglishCountryRegionName])))
ORDER BY [EnglishCountryRegionName] asc, [StateProvinceName] desc, [City] asc
```

Возвращает таблицу, которая однозначно ранжирует каждую географию с одинаковым значением EnglishCountryRegionName, по состоянию StateProvinceName и City.

Пример 2. Визуальное вычисление

Следующие визуальные вычисления DAX запросах:

DAX

```
SalesRankWithinYear = ROWNUMBER(ORDERBY([SalesAmount], DESC),
PARTITIONBY([CalendarYear]))

SalesRankAllHistory = ROWNUMBER(ORDERBY([SalesAmount], DESC))
```

Создайте два столбца, которые однозначно ранжировать каждый месяц по общему объему продаж, как в течение каждого года, так и всей истории.

Снимок экрана ниже: визуальная матрица и первое выражение визуального вычисления:

```
1 SalesRankWithinYear = ROWNUMBER(ORDERBY([SalesAmount], DESC), PARTITIONBY([CalendarYear]))
```

CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	SalesRankWithinYear	SalesRankAllHistory
2011	1	1	$323,743.2056	12.00	37.00
		2	$437,372.4476	11.00	34.00
		3	$514,918.2284	10.00	25.00
		Total	**$1,276,033.8816**	**4.00**	**11.00**
	2	4	$523,950.0258	9.00	24.00
		5	$539,469.0004	8.00	23.00
		6	$647,524.214	4.00	16.00
		Total	**$1,710,943.2402**	**3.00**	**7.00**
	3	7	$648,343.8132	3.00	15.00
		8	$599,874.2732	7.00	20.00
		9	$609,121.865	6.00	19.00
		Total	**$1,857,339.9514**	**2.00**	**6.00**
	4	10	$655,170.3294	2.00	14.00
		11	$716,748.7178	1.00	13.00
		12	$636,253.2642	5.00	17.00
		Total	**$2,008,172.3114**	**1.00**	**5.00**
	Total		**$6,852,489.3846**	**1.00**	**2.00**

Связанное содержимое

INDEX ORDERBY PARTITIONBY WINDOW RANK

RUNNINGSUM

Статья • 24.04.2025

Применимо к: ⊗ Вычисляемый столбец ⊗ Вычисляемая таблица ⊗ Измерять ✓ Визуальное вычисление

Возвращает выполняемую сумму, вычисляемую вдоль заданной оси визуальной матрицы. То есть сумма заданного столбца, вычисляемая по всем элементам до текущего элемента оси.

Синтаксис

```
DAX

RUNNINGSUM ( <column>[, <axis>][, <blanks>][, <reset>] )
```

Параметры

Развернуть таблицу

Термин	Определение
`column`	Столбец, предоставляющий значение для каждого элемента.
`axis`	(Необязательно) Ссылка на оси, направление, по которому будет вычисляться сумма выполнения.
`blanks`	(Необязательно) Перечисление, определяющее, как обрабатывать пустые значения при сортировке `axis`. Поддерживаемые значения: • `DEFAULT` (значение по умолчанию), где поведение числовых значений является пустыми значениями, упорядочены от нуля до отрицательных значений. Поведение строк является пустым значением, упорядоченным перед всеми строками, включая пустые строки. • `FIRST`, пустые поля всегда упорядочены в начале независимо от возрастания или убывающего порядка сортировки. • `LAST`, пустые всегда упорядочены в конце независимо от возрастания или убывания сортировки.
`reset`	(Необязательно) Указывает, сбрасывается ли вычисление и на каком уровне иерархии столбцов визуальной фигуры. Допустимые значения: ссылка на поле столбца в текущей визуальной форме, `NONE` (по умолчанию), `LOWESTPARENT`, `HIGHESTPARENT` или целое число. Поведение зависит от целочисленного знака:

Термин	Определение
	— если нулевая или опущенная, вычисление не сбрасывается. Эквивалентно `NONE`. — если положительный, целое число определяет столбец, начиная с самого высокого, независимо от зерна. `HIGHESTPARENT` эквивалентно 1. — если отрицательно, целое число определяет столбец, начиная с самого низкого, относительно текущего зерна. `LOWESTPARENT` эквивалентно -1.

Возвращаемое значение

Скалярное значение, выполняемое суммирование до текущего элемента оси.

Замечания

Эту функцию можно использовать только в визуальных вычислениях.

Параметры `axis`, `blanks` и `reset` могут быть опущены.

Если значение `reset` является абсолютным (т. е. положительным целым числом `HIGHESTPARENT` или ссылкой на поле), а вычисление вычисляется на целевом уровне иерархии или выше, вычисление сбрасывается для каждого отдельного элемента. То есть функция оценивается в секции, содержащей только этот конкретный элемент.

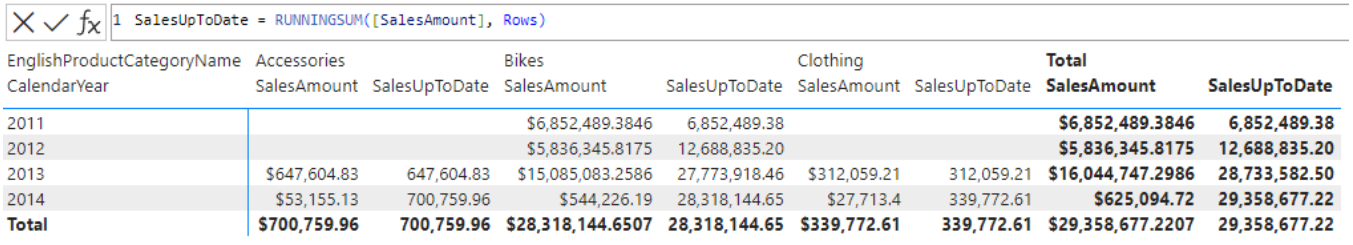
Пример

Учитывая таблицу, которая суммирует общий объем продаж для каждой категории продуктов и календарного года, следующий запрос DAX добавляет столбец с общим объемом продаж для этой категории до заданного года.

DAX

```
SalesUpToDate = RUNNINGSUM([SalesAmount], Rows)
```

Снимок экрана ниже: визуальная матрица и выражение визуального вычисления:



EnglishProductCategoryName	Accessories		Bikes		Clothing		Total	
CalendarYear	SalesAmount	SalesUpToDate	SalesAmount	SalesUpToDate	SalesAmount	SalesUpToDate	**SalesAmount**	**SalesUpToDate**
2011			$6,852,489.3846	6,852,489.38			**$6,852,489.3846**	**6,852,489.38**
2012			$5,836,345.8175	12,688,835.20			**$5,836,345.8175**	**12,688,835.20**
2013	$647,604.83	647,604.83	$15,085,083.2586	27,773,918.46	$312,059.21	312,059.21	**$16,044,747.2986**	**28,733,582.50**
2014	$53,155.13	700,759.96	$544,226.19	28,318,144.65	$27,713.4	339,772.61	**$625,094.72**	**29,358,677.22**
Total	**$700,759.96**	**700,759.96**	**$28,318,144.6507**	**28,318,144.65**	**$339,772.61**	**339,772.61**	$29,358,677.2207	29,358,677.22

См. также

INDEX MOVINGAVERAGE ORDERBY PARTITIONBY WINDOW

SELECTEDVALUE

Статья • 04.07.2024

Область применения: вычисление вычисляемого столбца вычисляемой таблицы Меры визуального элемента

Возвращает значение, когда контекст для columnName отфильтрован только до одного отдельного значения. В противном случае возвращает альтернативное значениеResult.

Синтаксис

```DAX
SELECTEDVALUE(<columnName>[, <alternateResult>])
```

Параметры

Развернуть таблицу

Термин	Определение
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Это не может быть выражением.
alternateResult	(Необязательно) Значение, возвращаемое при фильтрации контекста для columnName до нуля или нескольких отдельных значений. Если не указано, значение по умолчанию является выражением BLANK().

Возвращаемое значение

Значение, когда контекст для columnName отфильтрован только до одного отдельного значения. Else, alternateResult.

Замечания

- Эквивалентное выражение для `SELECTEDVALUE(<columnName>, <alternateResult>)` .`IF(HASONEVALUE(<columnName>), VALUES(<columnName>), <alternateResult>)`

- Дополнительные сведения о рекомендациях при использовании SELECTEDVALUE см. в разделе "Использование SELECTEDVALUE" вместо ЗНАЧЕНИй в DAX.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX:

DAX

```
DEFINE
 MEASURE DimProduct[Selected Color] = SELECTEDVALUE(DimProduct[Color], "No
Single Selection")
EVALUATE
 SUMMARIZECOLUMNS
   (ROLLUPADDISSUBTOTAL(DimProduct[Color], "Is Total"),
   "Selected Color", [Selected Color])ORDER BY [Is Total] ASC,
   [Color] ASC
```

Возвращает следующее:

Развернуть таблицу

DimProduct[Color]	[Итого]	[Выбранный цвет]
Черный	FALSE	Черный
Синий	FALSE	Синий
Серый	FALSE	Серый
Разноцветный	FALSE	Разноцветный
Неприменимо	FALSE	Неприменимо
Красный	FALSE	Красный
Серебряная	FALSE	Серебряная
Серебристый или черный	FALSE	Серебристый или черный
Белый	FALSE	Белый
Желтый	FALSE	Желтый

DimProduct[Color]	[Итого]	[Выбранный цвет]
	TRUE	Нет единого выбора

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

WINDOW

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает несколько строк, расположенных в пределах заданного интервала.

Синтаксис

DAX

```
WINDOW ( from[, from_type], to[, to_type][, <relation> or <axis>][,
<orderBy>][, <blanks>][, <partitionBy>][, <matchBy>][, <reset>] )
```

Параметры

Развернуть таблицу

Термин	Определение
`from`	Указывает, где начинается окно. Это может быть любое выражение DAX, возвращающее скалярное значение. Поведение зависит от параметра `from_type`: — если `from_type` REL, количество строк, возвращаемых (отрицательное значение) или вперед (положительное значение) из текущей строки, чтобы получить первую строку в окне. — если `from_type` ABS, а `from` положительным, то это позиция начала окна с начала раздела. Индексирование основано на 1, а 0 интерпретируется как 1. Например, 0 и 1 означают, что окно начинается с начала раздела. Если `from` отрицательно, то это позиция начала окна с конца раздела. -1 означает последнюю строку в разделе.
`from_type`	Изменяет поведение параметра `from`. Возможные значения: ABS (абсолютные) и REL (относительные). Значение по умолчанию — REL.
`to`	То же, что и `from`, но указывает конец окна. Последняя строка включена в окно.
`to_type`	То же, что и `from_type`, но изменяет поведение `to`.
`relation`	(Необязательно) Табличное выражение, из которого возвращаются выходные строки. Если указано, все столбцы в `partitionBy` должны поступать из нее или

Термин	Определение
	связанной таблицы. Если опущено: - `orderBy` необходимо явно указать. . Все выражения `orderBy` и `partitionBy` должны быть полными именами столбцов и поступать из одной таблицы. — по умолчанию ALLSELECTED() всех столбцов в `orderBy` и `partitionBy`.
`axis`	(Необязательно) Ось в визуальной форме. Доступно только в визуальных вычислениях и заменяет `relation`.
`orderBy`	(Необязательно) Предложение ORDERBY(), содержащее выражения, определяющие порядок сортировки каждой секции. Если опущено: - `relation` необходимо явно указать. — по умолчанию упорядочивается по каждому столбцу в `relation`, который еще не указан в `partitionBy`.
`blanks`	(Необязательно) Перечисление, определяющее, как обрабатывать пустые значения при сортировке `relation` или `axis`. Поддерживаемые значения: • `DEFAULT` (значение по умолчанию), где поведение числовых значений является пустыми значениями, упорядочены от нуля до отрицательных значений. Поведение строк является пустым значением, упорядоченным перед всеми строками, включая пустые строки. • `FIRST`, пустые поля всегда упорядочены в начале независимо от возрастания или убывающего порядка сортировки. • `LAST`, пустые всегда упорядочены в конце независимо от возрастания или убывания сортировки. Примечание. Если параметр `blanks` и пустые значения в функции ORDERBY() для отдельных выражений указываются, `blanks` для отдельного выражения orderBy имеет приоритет для соответствующего выражения orderBy, а выражения orderBy без `blanks` заданы будут учитывать параметр `blanks` родительской функции.
`partitionBy`	(Необязательно) Предложение PARTITIONBY(), содержащее столбцы, определяющие секционирование `relation`. Если опущено, `relation` рассматривается как одна секция.
`matchBy`	(Необязательно) Предложение MATCHBY(), содержащее столбцы, определяющие сопоставление данных и определение текущей строки.
`reset`	(Необязательно) Доступно только в визуальных вычислениях. Указывает, сбрасывается ли вычисление и на каком уровне иерархии столбцов визуальной фигуры. Допустимые значения: ссылка на поле столбца в текущей визуальной форме, `NONE` (по умолчанию), `LOWESTPARENT`, `HIGHESTPARENT` или целое число. Поведение зависит от целочисленного знака:

Термин	Определение
	— если нулевая или опущенная, вычисление не сбрасывается. Эквивалентно `NONE`. — если положительный, целое число определяет столбец, начиная с самого высокого, независимо от зерна. `HIGHESTPARENT` эквивалентно 1. — если отрицательно, целое число определяет столбец, начиная с самого низкого, относительно текущего зерна. `LOWESTPARENT` эквивалентно -1.

Возвращаемое значение

Все строки из окна.

Замечания

За исключением столбцов, добавленных функциями таблицы DAX, каждый столбец в `relation`, если `matchBy` отсутствует, или каждый столбец в `matchBy` и `partitionBy`, если `matchBy` присутствует, должен иметь соответствующее внешнее значение, чтобы помочь определить текущую строку, с которой следует работать. Если `from_type` и `to_type` оба значения ABS, то следующие действия применяются только к столбцам `partitionBy`:

- Если есть ровно один соответствующий внешний столбец, используется его значение.
- Если нет соответствующего внешнего столбца:
  - WINDOW сначала определит все столбцы, у которых нет соответствующего внешнего столбца.
  - Для каждого сочетания существующих значений для этих столбцов в родительском контексте WINDOWвычисляется WINDOW и возвращаются соответствующие строки.
  - WINDOW окончательные выходные данные — это объединение этих строк.
- Если существует несколько соответствующих внешних столбцов, возвращается ошибка.

Если все столбцы `relation` были добавлены DAX табличными функциями, возвращается ошибка.

Если `matchBy` присутствует, WINDOW попытается использовать `matchBy` и `partitionBy` столбцы для идентификации строки. Если `matchBy` отсутствует, а столбцы, указанные в `orderBy`, и `partitionBy` не могут однозначно идентифицировать каждую строку в `relation`, а затем:

- WINDOW попытается найти наименьшее количество дополнительных столбцов, необходимых для уникальной идентификации каждой строки.
- Если такие столбцы можно найти, WINDOW автоматически добавит эти новые столбцы в `orderBy`, и каждая секция сортируется с помощью этого нового набора столбцов orderBy.
- Если такие столбцы не удается найти, возвращается ошибка.

Если возвращается пустая таблица:

- Соответствующее внешнее значение столбца `orderBy` или `partitionBy` не существует в `relation`.
- Все окно находится за пределами секции, или начало окна после его окончания.

Если WINDOW используется в вычисляемом столбце, определенном в той же таблице, что и `relation`, и `orderBy` опущен, возвращается ошибка.

Если начало окна выходит перед первой строкой, оно устанавливается в первую строку. Аналогичным образом, если конец окна находится после последней строки секции, то для нее задано значение последней строки.

`reset` можно использовать только в визуальных вычислениях и не может использоваться в сочетании с `orderBy` или `partitionBy`. Если `reset` присутствует, `axis` можно указать, но `relation` невозможно.

Пример 1 — мера

Следующая мера:

DAX

```
3-day Average Price =
AVERAGEX(
    WINDOW(
        -2,REL,0,REL,
        SUMMARIZE(ALLSELECTED('Sales'), 'Date'[Date], 'Product'[Product]),
        ORDERBY('Date'[Date]),
        KEEP,
        PARTITIONBY('Product'[Product])
    ),
    CALCULATE(AVERAGE(Sales[Unit Price]))
)
```

Возвращает 3-дневный средний показатель цен на единицу для каждого продукта. Обратите внимание, что 3-дневное окно состоит из трех дней, в которых продукт имеет продажи, не обязательно три дня подряд.

Пример 2. Мера

Следующая мера:

DAX

```
RunningSum =
SUMX (
    WINDOW (
        1, ABS, 0, REL,
        ALLSELECTED (
            'Date'[Fiscal Year],
            'Date'[Month Number Of Year]
        ),
        PARTITIONBY ( 'Date'[Fiscal Year] )
    ),
    [Total Sales]
)
```

Возвращает беговую сумму для общего объема продаж по количеству месяцев года, перезапуская за каждый финансовый год:

Развернуть таблицу

Год	Число месяцев года	Сумма продаж	RunningSum
ФГ2018	1	$1,327,675	$1,327,675
ФГ2018	2	$3,936,463	$5,264,138
ФГ2018	3	$700,873	$5 965 011
ФГ2018	4	$1519,275	$7,484,286
ФГ2018	5	$2,960,378	$10,444,664
ФГ2018	6	$1,487,671	$11,932,336
ФГ2018	7	$1,423,357	$13,355,693
ФГ2018	8	$2057,902	$15,413,595
ФГ2018	9	$2,523,948	$17,937,543

Год	Число месяцев года	Сумма продаж	RunningSum
ФГ2018	10	$561,681	$18,499,224
ФГ2018	11	$4764,920	$23,264,145
ФГ2018	12	$596,747	$23,860,891
2019 финансовый год	1	$1,847,692	$1,847,692
2019 финансовый год	2	$2,829,362	$4677 054
2019 финансовый год	3	$2092,434	$6 769 488
2019 финансовый год	4	$2405,971	$9,175,459
2019 финансовый год	5	$3,459,444	$12,634,903
2019 финансовый год	6	$2,850,649	$15,485,552
2019 финансовый год	7	$2,939,691	$18,425,243
2019 финансовый год	8	$3,964,801	$22,390,045
2019 финансовый год	9	$3,287,606	$25,677,650
2019 финансовый год	10	$2,157,287	$27,834,938
2019 финансовый год	11	$ 3611,092	$31,446,030
2019 финансовый год	12	$ 2624 078	$34,070,109
2020-й финансовый год	1	$3,235,187	$3,235,187
2020-й финансовый год	2	$ 4070 046	$7,305,233
2020-й финансовый год	3	$4429,833	$11,735,066
2020-й финансовый год	4	$4002,614	$15,737,680
2020-й финансовый год	5	$5,265,797	$21,003,477
2020-й финансовый год	6	$3,465,241	$24,468,717
2020-й финансовый год	7	$3,513 064	$27,981,781
2020-й финансовый год	8	$5,247,165	$33,228,947
2020-й финансовый год	9	$5,104,088	$38,333,035
2020-й финансовый год	10	$3542,150	$41,875,184
2020-й финансовый год	11	$5,151,897	$47,027,081

Год	Число месяцев года	Сумма продаж	RunningSum
2020-й финансовый год	12	$4,851,194	$51,878,275

Пример 3. Визуальное вычисление

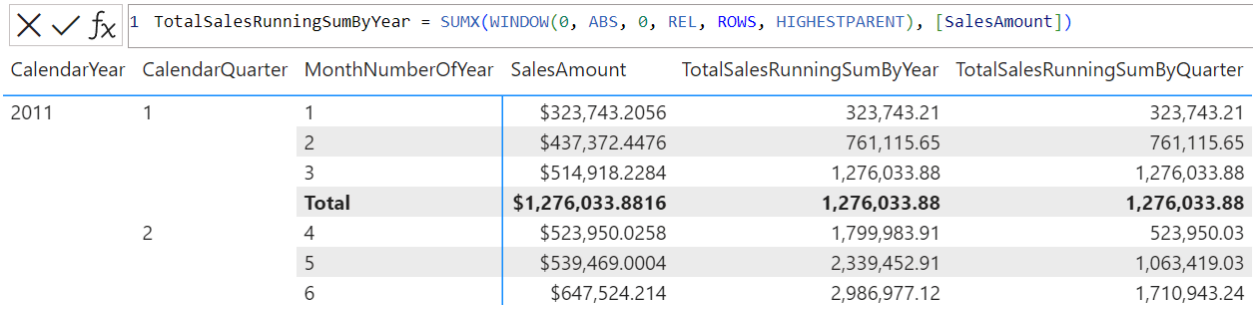
Следующий визуальный расчет DAX запроса:

```
DAX

TotalSalesRunningSumByYear = SUMX(WINDOW(0, ABS, 0, REL, ROWS,
HIGHESTPARENT), [SalesAmount])
```

Возвращает совокупный общий объем продаж по месяцам, вычисляемым по каждому году. Значения 1 и -2 можно использовать вместо `HIGHESTPARENT` с одинаковыми результатами.

Снимок экрана ниже: визуальная матрица и выражение визуального вычисления:



CalendarYear	CalendarQuarter	MonthNumberOfYear	SalesAmount	TotalSalesRunningSumByYear	TotalSalesRunningSumByQuarter
2011	1	1	$323,743.2056	323,743.21	323,743.21
		2	$437,372.4476	761,115.65	761,115.65
		3	$514,918.2284	1,276,033.88	1,276,033.88
		Total	**$1,276,033.8816**	**1,276,033.88**	**1,276,033.88**
	2	4	$523,950.0258	1,799,983.91	523,950.03
		5	$539,469.0004	2,339,452.91	1,063,419.03
		6	$647,524.214	2,986,977.12	1,710,943.24

Пример 4. Визуальное вычисление

Следующий визуальный расчет DAX запроса:

```
DAX

TotalSalesRunningSumByQuarter = SUMX(WINDOW(0, ABS, 0, REL, , -1),
[SalesAmount])
```

Возвращает совокупный общий объем продаж по месяцам, вычисляемый по каждому кварталу.

Связанное содержимое

INDEX MOVINGAVERAGE OFFSET ORDERBY PARTITIONBY RANGE RANK ROWNUMBER RUNNINGSUM

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Финансовые функции

Статья • 23.01.2025

Финансовые функции в DAX используются в формулах, выполняющих финансовые вычисления, такие как чистая текущая стоимость и ставка прибыли. Эти функции похожи на финансовые функции, используемые в Microsoft Excel.

В этой категории

Развернуть таблицу

Функция	Описание
ACCRINT	Возвращает накопленный процент за обеспечение, которое платит периодический процент.
ACCRINTM	Возвращает накопленный процент для обеспечения безопасности, которая платит проценты по истечении срока действия.
AMORDEGRC	Возвращает амортизацию для каждого отчетного периода. Аналогично AMORLINC, за исключением коэффициента амортизации, применяется в зависимости от жизни активов.
AMORLINC	Возвращает амортизацию для каждого отчетного периода.
COUPDAYBS	Возвращает количество дней с начала периода купона до даты его урегулирования.
COUPDAYS	Возвращает количество дней в период купона, содержащий дату урегулирования.
COUPDAYSNC	Возвращает количество дней от даты урегулирования до следующей даты купона.
COUPNCD	Возвращает следующую дату купона после даты урегулирования.
COUPNUM	Возвращает количество купонов, выплачиваемых между датой урегулирования и датой погашения, округленным до ближайшего всего купона.
COUPPCD	Возвращает предыдущую дату купона до даты урегулирования.
CUMIPMT	Возвращает совокупный процент, выплачиваемый за кредит между start_period и end_period.
CUMPRINC	Возвращает накопительный субъект, оплаченный по кредиту между start_period и end_period.

Функция	Описание
DB	Возвращает амортизацию актива в течение указанного периода с помощью метода фиксированного снижения баланса.
DDB	Возвращает амортизацию актива в течение указанного периода с помощью метода двойного снижения баланса или другого метода, который вы указали.
DISC	Возвращает ставку скидки для безопасности.
DOLLARDE	Преобразует цену доллара, выраженную как целочисленную часть и дробную часть, например 1,02, в цену доллара, выраженную как десятичное число.
DOLLARFR	Преобразует цену доллара, выраженную как целочисленную часть и дробную часть, например 1,02, в цену доллара, выраженную как десятичное число.
DURATION	Возвращает длительность Macauley для предполагаемого парного значения $ 100.
EFFECT	Возвращает эффективную годовую процентную ставку, учитывая номинальную годовую процентную ставку и количество составных периодов в год.
FV	Вычисляет будущую стоимость инвестиций на основе постоянной процентной ставки.
INTRATE	Возвращает процентную ставку для полностью инвестируемой безопасности.
IPMT	Возвращает процентный платеж за указанный период для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.
ISPMT	Вычисляет проценты, выплаченные (или полученные) за указанный период кредита (или инвестиции) с даже основными платежами.
MDURATION	Возвращает измененную длительность Macauley для безопасности с предполагаемой номинальной стоимостью $ 100.
NOMINAL	Возвращает номинальную годовую процентную ставку, учитывая эффективную ставку и количество составных периодов в год.
NPER	Возвращает количество периодов для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.
ODDFPRICE	Возвращает цену за $100 лицевой стоимости безопасности с нечетным (коротким или длинным) первым периодом.

Функция	Описание
ODDFYIELD	Возвращает доходность безопасности, которая имеет нечетный (короткий или длинный) первый период.
ODDLPRICE	Возвращает цену за $ 100 лицевой стоимости безопасности с нечетным (коротким или длинным) прошлым купонным периодом.
ODDLYIELD	Возвращает доходность безопасности, которая имеет нечетный (короткий или длинный) последний период.
PDURATION	Возвращает количество периодов, необходимых инвестиции для достижения указанной стоимости.
PMT	Вычисляет платеж за кредит на основе постоянных платежей и постоянной процентной ставки.
PPMT	Возвращает платеж по субъекту в течение заданного периода для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.
PRICE	Возвращает цену за $ 100 лицевой стоимости безопасности, которая платит периодический процент.
PRICEDISC	Возвращает цену за $ 100 лицевой стоимости скидок безопасности.
PRICEMAT	Возвращает цену за $ 100 лицевой стоимости безопасности, которая платит проценты по зрелости.
PV	Вычисляет текущую стоимость кредита или инвестиций на основе постоянной процентной ставки.
RATE	Возвращает процентную ставку за период аннуитета.
RECEIVED	Возвращает сумму, полученную по сроку погашения для полностью инвестируемой безопасности.
RRI	Возвращает эквивалентную процентную ставку для роста инвестиций.
SLN	Возвращает прямую амортизацию актива в течение одного периода.
SYD	Возвращает цифры амортизации актива в течение указанного периода.
TBILLEQ	Возвращает эквивалентную облигацию доходность для казначейского счета.
TBILLPRICE	Возвращает цену за $ 100 лиц для казначейского счета.
TBILLYIELD	Возвращает доходность казначейского счета.
VDB	Возвращает амортизацию актива в течение любого указанного периода, включая частичные периоды, используя метод двойного снижения баланса или другой указанный метод.

Функция	Описание
XIRR	Возвращает внутреннюю ставку прибыли для расписания денежных потоков, которые не обязательно периодически.
XNPV	Возвращает текущее значение для расписания денежных потоков, которые не обязательно периодически.
YIELD	Возвращает доходность по безопасности, которая платит периодический процент.
YIELDDISC	Возвращает годовую доходность для скидок на скидку.
YIELDMAT	Возвращает годовую доходность безопасности, которая платит проценты по зрелости.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ACCRINT

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает накопленный процент за обеспечение, которое платит периодический процент.

Синтаксис

DAX

```
ACCRINT(<issue>, <first_interest>, <settlement>, <rate>, <par>, <frequency>
[, <basis>[, <calc_method>]])
```

Параметры

Развернуть таблицу

Срок	Определение
`issue`	Дата проблемы безопасности.
`first_interest`	Дата первого интереса безопасности.
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`rate`	Годовая ставка купона на безопасность.
`par`	Значение парного значения безопасности.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.
`calc_method`	(Необязательно) Логическое значение, указывающее способ вычисления общего начисленного процента, когда дата урегулирования превышает дату first_interest. Если calc_method опущен, предполагается, что `TRUE`. — Если calc_method оценивается `TRUE` или опущен, ACCRINT возвращает общий накопленный процент от вопроса к урегулированию.

Срок	Определение
	— Если calc_method оценивается `FALSE`, ACCRINT возвращает накопленные проценты от first_interest до урегулирования.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	подсчета дней
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Накопленный процент.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- ACCRINT вычисляется следующим образом:

  $$\text{ACCRINT} = \text{par} \times \frac{\text{rate}}{\text{frequency}} \times \sum^{\text{NC}_{i=1}\frac{\text{A}_{i}}{\text{NL}_{i}_{i}}$$

  где:
  - A_i = количество накопленных дней в течение i^{th} квази-купонного периода в нечетном периоде.
  - NC = количество квази-купонных периодов, которые соответствуют нечетным периодам. Если это число содержит дробь, доведите его до следующего целого числа.
  - NL_i = обычная длина в днях квази-купонного периода в нечетном периоде.

- проблемы, first_interest и урегулирования усечены по целым числам.
- частота и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - проблема, first_interest или урегулирование не является допустимой датой.
  - проблема ≥ урегулирования.
  - скорость ≤ 0.
  - par ≤ 0.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Развернуть таблицу

данных	описание
1-март 2007 г.	Дата проблемы
31 августа 2008 г.	Дата первого интереса
1-май-2008 г.	Дата урегулирования
10%	Ставка купона
1000	Значение par
2	Частота является полунуальной (см. выше)
0	30/360 базис (см. выше)

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
{
```

```
  ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0)
}
```

Возвращает накопленные проценты от вопроса к урегулированию для обеспечения безопасности с указанными выше условиями.

Развернуть таблицу

[значение]
116.944444444444

Пример 2

Следующий запрос DAX:

DAX

```
EVALUATE
{
  ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0,
FALSE)
}
```

Возвращает накопленные проценты от first_interest до урегулирования для обеспечения безопасности с указанными выше условиями.

Развернуть таблицу

[значение]
66.9444444444445

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ACCRINTM

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает накопленный процент для обеспечения безопасности, которая платит проценты по истечении срока действия.

Синтаксис

DAX

```
ACCRINTM(<issue>, <maturity>, <rate>, <par>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`issue`	Дата проблемы безопасности.
`maturity`	Дата окончания срока действия безопасности.
`rate`	Годовая ставка купона на безопасность.
`par`	Значение парного значения безопасности.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический

`Basis`	**подсчета дней**
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Накопленный процент.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- ACCRINTM вычисляется следующим образом:

  $$\mathrm{ACCRINTM} = \mathrm{par} \times \mathrm{rate} \times \frac{\mathrm{A}}{\mathrm{D}}$$

  где:
  - A = количество накопленных дней, засчитываемых в соответствии с ежемесячной базой. Для процентов по пунктам зрелости используется количество дней с даты выпуска до даты зрелости.
  - D = годовая база данных.
- проблема и зрелость усечены в целые числа.
- база округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - проблема или срок действия не является допустимой датой.
  - проблема ≥ зрелости.
  - скорость ≤ 0.
  - par ≤ 0.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
1 апреля 2008 г.	Дата проблемы
15 июня 2008 г.	Дата зрелости
10%	Процентный купон
1000	Значение par
3	Фактическая/365 база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  ACCRINTM(DATE(2008,4,1), DATE(2008,6,15), 0.1, 1000, 3)
}
```

Возвращает накопленный процент для обеспечения безопасности с указанными выше условиями.

Развернуть таблицу

[значение]
20.5479452054795

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

AMORDEGRC

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает амортизацию для каждого отчетного периода. Эта функция предоставляется для французской системы бухгалтерского учета. Если актив приобретается в середине бухгалтерского периода, учитывается пропорциональное амортизация. Функция аналогична AMORLINC, за исключением того, что коэффициент амортизации применяется в расчете в зависимости от жизни активов.

Синтаксис

DAX

```
AMORDEGRC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>,
<rate>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`cost`	Стоимость ресурса.
`date_purchased`	Дата покупки актива.
`first_period`	Дата окончания первого периода.
`salvage`	Значение спасения в конце срока жизни актива.
`period`	Период.
`rate`	Скорость амортизации.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**система даты**
0 или опущено	360 дней (метод NASD)
1	Действительный
3	365 дней в год
4	360 дней в год (европейский метод)

Возвращаемое значение

Амортизация для каждого учетного периода.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Эта функция вернет амортизацию до последнего периода жизни активов или до тех пор, пока не будет обученной стоимости амортизации больше, чем стоимость активов минус стоимость спасения.
- Коэффициенты амортизации:

Развернуть таблицу

Жизни активов (1/ставка)	**коэффициент амортизации**
От 3 до 4 лет	1.5
От 5 до 6 лет	2
Более 6 лет	2.5

- Ставка амортизации будет расти до 50 процентов за период, предшествующий последнему периоду, и будет расти до 100 процентов за последний период.
- период и база округляются до ближайшего целого числа.
- Если возвращается ошибка:

- стоимость < 0.
- first_period или date_purchased не является допустимой датой.
- date_purchased > first_period.
- < спасения < 0 или > стоимости.
- период < 0.
- скорость ≤ 0.
- Срок жизни активов составляет от 0 до 1, 1, 1 и 2, 2 и 3 или 4 и 5.
- основанием является любое число, отличное от 0, 1, 3 или 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
2400	Стоить
19 августа 2008 г.	Дата покупки
31 декабря 2008 г.	Окончание первого периода
300	Значение salvage
1	Период
15%	Коэффициент амортизации
1	Фактическая база (см. выше)

Следующий запрос DAX:

DAX

```
EVALUATE
{
  AMORDEGRC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)
}
```

Возвращает амортизацию первого периода, учитывая указанные выше условия.

Развернуть таблицу

[значение]
776

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

AMORLINC

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает амортизацию для каждого отчетного периода. Эта функция предоставляется для французской системы бухгалтерского учета. Если актив приобретается в середине бухгалтерского периода, учитывается пропорциональное амортизация.

Синтаксис

DAX

```
AMORLINC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>,
<rate>[, <basis>])
```

Параметры

⛶ Развернуть таблицу

Срок	Определение
`cost`	Стоимость ресурса.
`date_purchased`	Дата покупки актива.
`first_period`	Дата окончания первого периода.
`salvage`	Значение спасения в конце срока жизни актива.
`period`	Период.
`rate`	Скорость амортизации.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

⛶ Развернуть таблицу

`Basis`	**система даты**
0 или опущено	360 дней (метод NASD)
1	Действительный
3	365 дней в год
4	360 дней в год (европейский метод)

Возвращаемое значение

Амортизация для каждого учетного периода.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- период и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - `cost` < 0.
  - `first_period` или `date_purchased` не является допустимой датой.
  - `date_purchased` > `first_period`.
  - `salvage` < 0 или `salvage` > `cost`.
  - `period` < 0.
  - `rate` ≤ 0.
  - `basis` любое число, отличное от 0, 1, 3 или 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
2400	Стоить

данных	описание
19 августа 2008 г.	Дата покупки
31 декабря 2008 г.	Окончание первого периода
300	Значение salvage
1	Период
15%	Коэффициент амортизации
1	Фактическая база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  AMORLINC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)
}
```

Возвращает амортизацию первого периода, учитывая указанные выше условия.

Развернуть таблицу

[значение]
360

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUPDAYBS

Статья • 04.07.2024

Область применения: вычисление вычисляемого столбца вычисляемой таблицы Меры визуального элемента

Возвращает количество дней с начала периода купона до даты его урегулирования.

Синтаксис

```
DAX

COUPDAYBS(<settlement>, <maturity>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Термин	Определение
поселение	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
зрелость	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
обновления	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
базовых	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Базовый параметр принимает следующие значения:

Развернуть таблицу

Базис	Подсчет дней
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360

Базис	Подсчет дней
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Количество дней с начала периода купона до даты его урегулирования.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 г. 30 лет после 1 января 2008 г.
- расчет и зрелость усечены в целые числа.
- частота и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - частота — любое число, отличное от 1, 2 или 4.
  - basis < 0 или basis > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Данные	Description
25-январь-11	Дата урегулирования
15 ноября-11	Дата зрелости
2	Semiannual купон (см. выше)
1	Фактическая/фактическая база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  COUPDAYBS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Возвращает количество дней с начала купонного периода до даты урегулирования для облигации с условиями выше.

Развернуть таблицу

[значение]
71

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUPDAYS

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает количество дней в период купона, содержащий дату урегулирования.

Синтаксис

DAX

```
COUPDAYS(<settlement>, <maturity>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360

`Basis`	**подсчета дней**
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Количество дней в период купона, содержащий дату урегулирования.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 г., а дата погашения — 1 января 2038 г. 30 лет после даты выдачи 1 января 2008 г.
- расчет и зрелость усечены в целые числа.
- частота и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
25-январь-11	Дата урегулирования
15 ноября-11	Дата зрелости
2	Semiannual купон (см. выше)
1	Фактическая/фактическая база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  COUPDAYS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Возвращает количество дней в период купона, содержащий дату урегулирования, для облигации с указанными выше условиями.

Развернуть таблицу

[значение]
181

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUPDAYSNC

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает количество дней от даты урегулирования до следующей даты купона.

Синтаксис

DAX

```
COUPDAYSNC(<settlement>, <maturity>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360

`Basis`	**подсчета дней**
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Количество дней от даты урегулирования до следующей даты купона.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- расчет и зрелость усечены в целые числа.
- частота и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
25-январь-11	Дата урегулирования
15 ноября-11	Дата зрелости
2	Semiannual купон (см. выше)
1	Фактическая/фактическая база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  COUPDAYSNC(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Возвращает количество дней от даты урегулирования до следующей даты купона для облигации с указанными выше условиями.

Развернуть таблицу

[значение]
110

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUPNCD

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает следующую дату купона после даты урегулирования.

Синтаксис

```
DAX

COUPNCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360

`Basis`	**подсчета дней**
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Следующая дата купона после даты урегулирования.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 г., а дата погашения — 1 января 2038 г. 30 лет после даты выдачи 1 января 2008 г.
- расчет и зрелость усечены в целые числа.
- частота и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
25-январь-11	Дата урегулирования
15 ноября-11	Дата зрелости
2	Semiannual купон (см. выше)
1	Фактическая/фактическая база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  COUPNCD(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Возвращает следующую дату купона после даты урегулирования для облигации с указанными выше условиями.

Развернуть таблицу

[значение]
5.15.2011 12:00:00 AM

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUPNUM

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает количество купонов, выплачиваемых между датой урегулирования и датой погашения, округленным до ближайшего всего купона.

Синтаксис

DAX

```
COUPNUM(<settlement>, <maturity>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический

`Basis`	**подсчета дней**
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Количество купонов, выплачиваемых между датой расчетов и датой погашения.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- расчет и зрелость усечены в целые числа.
- частота и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
25-январь-07	Дата урегулирования
15 ноября-08	Дата зрелости
2	Semiannual купон (см. выше)
1	Фактическая/фактическая база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  COUPNUM(DATE(2007,1,25), DATE(2008,11,15), 2, 1)
}
```

Возвращает количество купонных платежей за облигацию с указанными выше условиями.

Развернуть таблицу

[значение]
4

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COUPPCD

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает предыдущую дату купона до даты урегулирования.

Синтаксис

DAX

```
COUPPCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360

`Basis`	**подсчета дней**
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Предыдущая дата купона до даты урегулирования.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- расчет и зрелость усечены в целые числа.
- частота и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
25-январь-11	Дата урегулирования
15 ноября-11	Дата зрелости
2	Semiannual купон (см. выше)
1	Фактическая/фактическая база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  COUPPCD(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Возвращает предыдущую дату купона до даты урегулирования для облигации с использованием указанных выше условий.

Развернуть таблицу

[значение]
11.15.2010 12:00:00

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CUMIPMT

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает совокупный процент, выплачиваемый за кредит между start_period и end_period.

Синтаксис

```
DAX

CUMIPMT(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка.
`nper`	Общее количество периодов оплаты.
`pv`	Текущее значение.
`start_period`	Первый период вычисления. Должно быть от 1 до end_period (включительно).
`end_period`	Последний период вычисления. Должно быть между start_period и nper (включительно).
`type`	Время оплаты. Допустимые значения перечислены ниже этой таблицы.

Параметр `type` принимает следующие значения:

Развернуть таблицу

`Type`	**времени**
0 (ноль)	Платеж в конце периода

Type	времени
1	Оплата в начале периода

Возвращаемое значение

Совокупный процент, выплачиваемый за указанный период.

Замечания

- Убедитесь, что вы согласованы с единицами, используемыми для указания скорости и nper. Если вы вносите ежемесячные платежи по четырехлетнему кредиту на годовую процентную ставку 10 процентов, используйте 0,1/12 для ставки и 4*12 для nper. Если вы вносите ежегодные платежи по одному кредиту, используйте 0,1 для ставки и 4 для nper.
- start_period, end_period и тип округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - скорость ≤ 0.
  - nper < 1.
  - pv ≤ 0.
  - start_period < 1 или start_period > end_period.
  - end_period < start_period или end_period > nper.
  - тип — любое число, отличное от 0 или 1.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Развернуть таблицу

данных	описание
9%	Годовая процентная ставка
30	Годы кредита
125000	Текущее значение

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
{
  CUMIPMT(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

Возвращает общий процент, выплачиваемый во второй год платежей, периодов 13–24, предполагая, что платежи выполняются в начале каждого месяца.

Развернуть таблицу

[значение]
-11052.3395838718

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE
{
  CUMIPMT(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

Возвращает проценты, выплачиваемые за один платеж в первом месяце, при условии, что платеж производится в конце месяца.

Развернуть таблицу

[значение]
-937.5

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CUMPRINC

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает накопительный субъект, оплаченный по кредиту между start_period и end_period.

Синтаксис

DAX

```
CUMPRINC(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка.
`nper`	Общее количество периодов оплаты.
`pv`	Текущее значение.
`start_period`	Первый период вычисления. Должно быть от 1 до end_period (включительно).
`end_period`	Последний период вычисления. Должно быть между start_period и nper (включительно).
`type`	Время оплаты. Допустимые значения перечислены ниже этой таблицы.

Параметр `type` принимает следующие значения:

Развернуть таблицу

`Type`	**времени**
0 (ноль)	Платеж в конце периода

Type	времени
1	Оплата в начале периода

Возвращаемое значение

Совокупный субъект, оплачиваемый за указанный период.

Замечания

- Убедитесь, что вы согласованы с единицами, используемыми для указания скорости и nper. Если вы вносите ежемесячные платежи по четырехлетнему кредиту на годовую процентную ставку 10 процентов, используйте 0,1/12 для ставки и 4*12 для nper. Если вы вносите ежегодные платежи по одному кредиту, используйте 0,1 для ставки и 4 для nper.
- start_period, end_period и тип округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - скорость ≤ 0.
  - nper < 1.
  - pv ≤ 0.
  - start_period < 1 или start_period > end_period.
  - end_period < start_period или end_period > nper.
  - тип — любое число, отличное от 0 или 1.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Развернуть таблицу

данных	описание
9%	Годовая процентная ставка
30	Срок в годах
125000	Текущее значение

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
{
  CUMPRINC(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

Возвращает общий объем выплат в течение второго года платежей, периодов 13–24, при условии, что платежи выполняются в начале каждого месяца.

Развернуть таблицу

[значение]
-927.153472378062

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE
{
  CUMPRINC(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

Возвращает субъект, оплачиваемый в одном платеже в первом месяце, при условии, что платеж производится в конце месяца.

Развернуть таблицу

[значение]
-68.2782711809784

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DB

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает амортизацию актива в течение указанного периода с помощью метода фиксированного снижения баланса.

Синтаксис

DAX

```
DB(<cost>, <salvage>, <life>, <period>[, <month>])
```

Параметры

Развернуть таблицу

Срок	Определение
`cost`	Начальная стоимость ресурса.
`salvage`	Значение в конце амортизации (иногда называется значением спасения ресурса). Это значение может быть равно 0.
`life`	Количество периодов, в течение которых ресурс не рекомендуется (иногда называется полезной жизнью актива).
`period`	Период, для которого требуется вычислить амортизацию. Период должен использовать те же единицы, что и время жизни. Должно быть от 1 до жизни (включительно).
`month`	(Необязательно) Количество месяцев в первом году. Если месяц опущен, предполагается, что он равен 12.

Возвращаемое значение

Амортизация за указанный период.

Замечания

- Метод фиксированного снижения баланса вычисляет амортизацию с фиксированной скоростью. DB использует следующие формулы для вычисления амортизации за период:

$$(\text{cost} - \text{total амортизация от предыдущих периодов}) \times \text{rate}$$

где:
  - $\text{rate} = 1 - (((\frac{\text{salvage}}}{\text{cost}})^{{\frac{1}{\text{life}}})\text{, округленный до трех десятичных знаков}$

- Амортизация для первых и последних периодов является особым случаем.
  - В течение первого периода DB использует следующую формулу:

$$\frac{\text{cost} \times \text{rate} \times \text{month}}{12}$$

  - В течение последнего периода DB использует следующую формулу:

$$\frac{(\text{cost} - \text{total амортизация от предыдущих периодов}) \times \text{rate} \times (12 - \text{month})}{12}$$

- период и месяц округляются до ближайшего целого числа.

- Если возвращается ошибка:
  - стоимость < 0.
  - salvage < 0.
  - жизнь < 1.
  - период < 1 или период > жизни.
  - месяц < 1 или месяц > 12.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
{
```

```
  DB(1000000, 0, 6, 1, 2)
}
```

Возвращает амортизацию актива за последние два месяца первого года, предполагая, что она будет стоить \0 через 6 лет.

Развернуть таблицу

[значение]
166666.666666667

Пример 2

Ниже вычислится общая амортизация всех активов в разных годах за их время существования. Здесь, первый год включает только 7 месяцев амортизации, и в прошлом году только 5 месяцев.

DAX

```
DEFINE
VAR NumDepreciationPeriods = MAX(Asset[LifeTimeYears])+1
VAR DepreciationPeriods = GENERATESERIES(1, NumDepreciationPeriods)
EVALUATE
ADDCOLUMNS (
  DepreciationPeriods,
  "Current Period Total Depreciation",
  SUMX (
    FILTER (
      Asset,
      [Value] <= [LifetimeYears]+1
    ),
    DB([InitialCost], [SalvageValue], [LifetimeYears], [Value], 7)
  )
)
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DDB

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает амортизацию актива в течение указанного периода с помощью метода двойного снижения баланса или другого метода, который вы указали.

Синтаксис

DAX

```
DDB(<cost>, <salvage>, <life>, <period>[, <factor>])
```

Параметры

Развернуть таблицу

Срок	Определение
`cost`	Начальная стоимость ресурса.
`salvage`	Значение в конце амортизации (иногда называется значением спасения ресурса). Это значение может быть равно 0.
`life`	Количество периодов, в течение которых ресурс не рекомендуется (иногда называется полезной жизнью актива).
`period`	Период, для которого требуется вычислить амортизацию. Период должен использовать те же единицы, что и время жизни. Должно быть от 1 до жизни (включительно).
`factor`	(Необязательно) Скорость снижения баланса. Если фактор опущен, предполагается, что значение равно 2 (метод двойного уменьшения баланса).

Возвращаемое значение

Амортизация за указанный период.

Замечания

- Метод двойного снижения баланса вычисляет амортизацию с ускоренной скоростью. Амортизация является самой высокой за первый период и уменьшается в последующие периоды. DDB использует следующую формулу для вычисления амортизации за период:

  $$\text{Min}(\text{cost} - \text{total амортизация от предыдущих периодов}) \times (\frac{\text{{factor}}{text{life}}),(\text{cost} - \text{salvage} - \text{total амортизация с предыдущих периодов})$$

- Коэффициент изменения, если вы не хотите использовать метод двойного уменьшения баланса.

- Используйте функцию VDB, если вы хотите переключиться на метод амортизации прямой линии, если амортизация больше, чем вычисление уменьшения баланса.

- период округляется до ближайшего целого числа.

- Если возвращается ошибка:
  - стоимость < 0.
  - salvage < 0.
  - жизнь < 1.
  - период < 1 или период > жизни.
  - фактор ≤ 0.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Пример 1

Следующий запрос DAX:

DAX

```
EVALUATE
{
  DDB(1000000, 0, 10, 5, 1.5)
}
```

Возвращает амортизацию актива в 5th год, предполагая, что он будет стоить $0 после 10 лет. Этот расчет использует коэффициент 1,5.

Развернуть таблицу

[значение]
78300.9375

Пример 2

Ниже вычислится общая амортизация всех активов в разных годах за их время существования. Этот расчет использует коэффициент по умолчанию 2 (метод двойного уменьшения баланса).

DAX

```
DEFINE
VAR NumDepreciationPeriods = MAX(Asset[LifeTimeYears])
VAR DepreciationPeriods = GENERATESERIES(1, NumDepreciationPeriods)
EVALUATE
  ADDCOLUMNS (
  DepreciationPeriods,
  "Current Period Total Depreciation",
  SUMX (
    FILTER (
      Asset,
      [Value] <= [LifetimeYears]
    ),
    DDB([InitialCost], [SalvageValue], [LifetimeYears], [Value])
  )
)
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DISC

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает ставку скидки для безопасности.

Синтаксис

DAX

```
DISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`pr`	Цена на цену безопасности на $100 лицовую стоимость.
`redemption`	Стоимость активации безопасности за $100 лиц.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический

`Basis`	**подсчета дней**
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Скидка.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выпущена 1 января 2018 года и приобретена покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2018 г., дата урегулирования будет 1 июля 2018 года, а дата погашения будет 1 января 2048 г. 30 лет после 1 января 2018 г.
- DISC вычисляется следующим образом:

  $$\text{DISC} = \frac{\text{активация} - \text{par}}{\text{активация}} \times \frac{\text{B}}{\text{DSM}}$$

  где:

  - B = количество дней в год в зависимости от года.
  - DSM = количество дней между расчетом и зрелости.
- расчет и зрелость усечены в целые числа.
- база округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - pr ≤ 0.

- активация ≤ 0.
- базовый < 0 или базовый > 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
07/01/2018	Дата урегулирования
01/01/2048	Дата зрелости
97.975	Цена
100	Значение активации
1	Фактическая/фактическая база (см. выше)

Следующий запрос DAX:

DAX

```
EVALUATE
{
  DISC(DATE(2018,7,1), DATE(2048,1,1), 97.975, 100, 1)
}
```

Возвращает ставку скидок на облигацию с указанными выше условиями.

Развернуть таблицу

[значение]
0.000686384169121348

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DOLLARDE

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Преобразует цену доллара, выраженную как целочисленную часть и дробную часть, например 1,02, в цену доллара, выраженную как десятичное число. Дробные номера долларов иногда используются для цен на безопасность.

Синтаксис

```
DAX

DOLLARDE(<fractional_dollar>, <fraction>)
```

Параметры

Развернуть таблицу

Срок	Определение
`fractional_dollar`	Число, выраженное как целочисленная часть и часть дроби, разделенная десятичным символом.
`fraction`	Целое число, используемое в знаменателе дроби.

Возвращаемое значение

Десятичное значение `fractional_dollar`.

Замечания

- Часть дроби значения делится на указанное целое число. Например, если вы хотите, чтобы цена была выражена точностью 1/16 доллара, вы делите часть дроби на 16. В этом случае 1.02 представляет $1,125 ($1 + 2/16 = $1,125).
- дробь округляется до ближайшего целого числа.
- Если возвращается ошибка:

    - доля < 1.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX:

DAX

```
EVALUATE
{
  DOLLARDE(1.02, 16)
}
```

Возвращает значение 1.125, десятичная цена исходной дробной цены, 1,02, считывается как 1 и 2/16. Так как значение дроби равно 16, цена имеет точность 1/16 доллара.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DOLLARFR

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует цену доллара, выраженную как десятичное число, в цену доллара, выраженную как целочисленная часть и доля, например 1,02. Дробные номера долларов иногда используются для цен на безопасность.

Синтаксис

```DAX
DOLLARFR(<decimal_dollar>, <fraction>)
```

Параметры

Развернуть таблицу

Срок	Определение
`decimal_dollar`	Десятичное число.
`fraction`	Целое число, используемое в знаменателе дроби.

Возвращаемое значение

Дробное значение `decimal_dollar`, выраженное как целочисленная часть и дробная часть.

Замечания

- дробь округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - доля < 1.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на

уровне строк (RLS).

Пример

Следующий запрос DAX:

```
DAX

EVALUATE
{
  DOLLARFR(1.125, 16)
}
```

Возвращает значение 1.02, считываемое как 1 и 2/16, которое является соответствующей ценой дроби исходной десятичной цены, 1,125. Так как значение дроби равно 16, цена имеет точность 1/16 доллара.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DURATION

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает длительность Macauley для предполагаемого парного значения $100. Длительность определяется как взвешенный средний показатель текущей стоимости денежных потоков и используется в качестве меры ответа цены облигаций на изменения в доходности.

Синтаксис

```
DAX

DURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`coupon`	Годовая ставка купона на безопасность.
`yld`	Годовая доходность безопасности.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Длительность Макули.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения — 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- расчет и зрелость усечены в целые числа.
- частота и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - купон < 0.
  - yld < 0
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на

уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
07/01/2018	Дата урегулирования
01/01/2048	Дата зрелости
8.0%	Процентный купон
9.0%	Процент доходности
2	Частота является полунуальной (см. выше)
1	Фактическая/фактическая база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  DURATION(DATE(2018,7,1), DATE(2048,1,1), 0.08, 0.09, 2, 1)
}
```

Возвращает длительность Макули для связи с указанными выше условиями.

Развернуть таблицу

[значение]
10.9191452815919

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EFFECT

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает эффективную годовую процентную ставку, учитывая номинальную годовую процентную ставку и количество составных периодов в год.

Синтаксис

DAX

```
EFFECT(<nominal_rate>, <npery>)
```

Параметры

Развернуть таблицу

Срок	Определение
`nominal_rate`	Номинальная процентная ставка.
`npery`	Количество составных периодов в год.

Возвращаемое значение

Эффективная годовая процентная ставка.

Замечания

- EFFECT вычисляется следующим образом:

$$\text{EFFECT} = \left(1 + \frac{\text{nominal_rate}}{\text{npery}}\right)^{\text{npery}} - 1$$

- npery округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - nominal_rate ≤ 0.

- nper < 1.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
5.25%	Номинальная процентная ставка
4	Количество составных периодов в год

Следующий запрос DAX:

```
DAX

EVALUATE
{
  EFFECT(0.0525, 4)
}
```

Возвращает эффективную процентную ставку, используя указанные выше условия.

Развернуть таблицу

[значение]
0.0535426673707584

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FV

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Вычисляет будущую стоимость инвестиций на основе постоянной процентной ставки. Вы можете использовать FV с периодическими, постоянными платежами и (или) с одной суммой оплаты.

Синтаксис

DAX

```
FV(<rate>, <nper>, <pmt>[, <pv>[, <type>]])
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка на период.
`nper`	Общее количество периодов оплаты в аннуитете.
`pmt`	Платеж, сделанный каждый период; он не может измениться в течение жизни аннуитета. Как правило, pmt содержит субъект и проценты, но никаких других сборов или налогов.
`pv`	(Необязательно) Текущее значение или сумма единовременной суммы, которая стоит ряд будущих платежей прямо сейчас. Если pv опущен, предполагается, что BLANK.
`type`	(Необязательно) Число 0 или 1, указывающее, когда должны быть выплаты. Если тип опущен, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `type` принимает следующие значения:

Развернуть таблицу

Задать тип, равный	Если платежи должны быть
0 или опущено	В конце периода
1	В начале периода

Примечание. Более полное описание аргументов в FV и дополнительные сведения о функциях аннуитета см. в PV функции.

Возвращаемое значение

Будущая стоимость инвестиций.

Замечания

- Убедитесь, что вы согласованы с единицами, используемыми для указания скорости и nper. Если вы вносите ежемесячные платежи по четырехлетнему кредиту на 12 процентов годовых процентов, используйте 0,12/12 для ставки и 4*12 для nper. Если вы вносите ежегодные платежи по тому же кредиту, используйте 0,12 для ставки и 4 для nper.
- Для всех аргументов денежные средства, которые вы оплачиваете, такие как депозиты для сбережений, представлено отрицательными числами; полученные денежные средства, такие как дивидендные проверки, представлены положительными числами.
- тип округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - nper < 1
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
6%	Годовая процентная ставка

данных	описание
10	Количество платежей
-200	Сумма платежа
-500	Текущее значение
1	Платеж должен быть в начале периода (0 указывает, что платеж должен быть в конце периода)

Следующий запрос DAX:

DAX

```
EVALUATE
{
  FV(0.06/12, 10, -200, -500, 1)
}
```

Возвращает будущую стоимость инвестиций с помощью указанных выше условий.

Развернуть таблицу

[значение]
2581.40337406012

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

INTRATE

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает процентную ставку для полностью инвестируемой безопасности.

Синтаксис

DAX

```
INTRATE(<settlement>, <maturity>, <investment>, <redemption>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`investment`	Сумма, вложенная в безопасность.
`redemption`	Сумма, полученная по истечении срока действия.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	подсчета дней
0 или опущено	США (NASD) 30/360
1	Фактический/фактический

`Basis`	**подсчета дней**
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Процентная ставка.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения — 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- INTRATE вычисляется следующим образом:

  $$\text{INTRATE} = \frac{\text{активация} - \text{инвестиции}{\text{investment}} \times \frac{\text{B}{\text{{DIM}}}$$

  где:
  - B = количество дней в год в зависимости от года.
  - DIM = количество дней от расчетов до зрелости.
- расчет и зрелость усечены в целые числа.
- база округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - инвестиции ≤ 0.

- активация ≤ 0.
- базовый < 0 или базовый > 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
2/15/2008	Дата урегулирования
5/15/2008	Дата зрелости
$1000 000	Инвестиция
$1,014,420	Значение активации
2	Фактическая/360 база

Следующий запрос DAX:

```
DAX

EVALUATE
{
  INTRATE(DATE(2008,2,15), DATE(2008,5,15), 1000000, 1014420, 2)
}
```

Возвращает ставку скидки для облигации, используя указанные выше условия.

Развернуть таблицу

[значение]
0.05768

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

IPMT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает процентный платеж за указанный период для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.

Синтаксис

DAX

```
IPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка на период.
`per`	Период, для которого вы хотите найти интерес. Должно быть от 1 до nper (включительно).
`nper`	Общее количество периодов оплаты в аннуитете.
`pv`	Текущее значение или сумма единовременной суммы, которая стоит ряд будущих платежей прямо сейчас.
`fv`	(Необязательно) Будущая стоимость или денежный баланс, который вы хотите достичь после последнего платежа. Если fv опущен, предполагается, что BLANK.
`type`	(Необязательно) Число 0 или 1, указывающее, когда должны быть выплаты. Если тип опущен, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `type` принимает следующие значения:

Развернуть таблицу

Задать `type` равным	Если платежи должны быть
0 или опущено	В конце периода
1	В начале периода

Возвращаемое значение

Выплата процентов за указанный период.

Замечания

- Убедитесь, что вы согласованы с единицами, используемыми для указания скорости и nper. Если вы вносите ежемесячные платежи по четырехлетнему кредиту на 12 процентов годовых процентов, используйте 0,12/12 для ставки и 4*12 для nper. Если вы вносите ежегодные платежи по тому же кредиту, используйте 0,12 для ставки и 4 для nper.
- Для всех аргументов денежные средства, которые вы оплачиваете, такие как депозиты для сбережений, представлено отрицательными числами; полученные денежные средства, такие как дивидендные проверки, представлены положительными числами.
- тип округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - per < 1 или per > nper
  - nper < 1
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Развернуть таблицу

данных	описание
10.00%	Годовые проценты
3	Годы кредита

данных	описание
$8000	Текущая стоимость кредита

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
{
  IPMT(0.1/12, 1, 3*12, 8000)
}
```

Возвращает ежемесячный процент за первый месяц для кредита с указанными выше условиями.

Развернуть таблицу

[значение]
-66.6666666666667

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE
{
  IPMT(0.1, 3, 3, 8000)
}
```

Возвращает ежегодные проценты за последний год для кредита с указанными выше условиями, где платежи выполняются ежегодно.

Развернуть таблицу

[значение]
-292.447129909366

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISPMT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Вычисляет проценты, выплаченные (или полученные) за указанный период кредита (или инвестиции) с даже основными платежами.

Синтаксис

DAX

```
ISPMT(<rate>, <per>, <nper>, <pv>)
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка для инвестиций.
`per`	Период, для которого вы хотите найти интерес. Должно быть от 0 до nper-1 (включительно).
`nper`	Общее количество периодов оплаты для инвестиций.
`pv`	Текущая стоимость инвестиций. Для кредита, pv является суммой кредита.

Возвращаемое значение

Проценты, выплачиваемые (или полученные) за указанный период.

Замечания

- Убедитесь, что вы согласованы с единицами, используемыми для указания скорости и nper. Если вы вносите ежемесячные платежи по четырехлетнему кредиту на годовую процентную ставку 12 процентов, используйте 0,12/12 для

ставки и 4*12 для nper. Если вы вносите ежегодные платежи по тому же кредиту, используйте 0,12 для ставки и 4 для nper.

- Для всех аргументов денежные средства, которые вы оплачиваете, такие как депозиты для сбережений или других выводов, представлены отрицательными числами; полученные денежные средства, такие как дивидендные проверки и другие депозиты, представлены положительными числами.
- ISPMT подсчитывает каждый период, начиная с нуля, а не один.
- Большинство кредитов используют график погашения даже с периодическими платежами. Функция IPMT возвращает процентную оплату за указанный период для этого типа кредита.
- Некоторые кредиты используют график погашения даже с основными платежами. Функция ISPMT возвращает процентную оплату за указанный период для этого типа кредита.
- Если возвращается ошибка:
  - nper = 0.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
$4000	Текущее значение
4	Количество периодов
10%	Скорость

Чтобы проиллюстрировать, когда использовать ISPMT, в таблице амортизации ниже используется план выплаты четных субъектов с указанными выше условиями. Плата за проценты в каждом периоде равна времени неоплачиваемого баланса за предыдущий период. И платеж каждый период равен даже субъекту плюс процент за период.

Развернуть таблицу

Период	Основной платеж	Платеж по проценту	Итоговый платеж	Равновесие
				4,000.00
1	1,000.00	400.00	1,400.00	3,000.00
2	1,000.00	300.00	1,300.00	2,000.00
3	1,000.00	200.00	1,200.00	1,000.00
4	1,000.00	100.00	1,100.00	0.00

Следующий запрос DAX:

DAX

```
DEFINE
VAR NumPaymentPeriods = 4
VAR PaymentPeriods = GENERATESERIES(0, NumPaymentPeriods-1)
EVALUATE
ADDCOLUMNS (
  PaymentPeriods,
  "Interest Payment",
  ISPMT(0.1, [Value], NumPaymentPeriods, 4000)
)
```

Возвращает проценты, выплачиваемые в течение каждого периода, используя план выплаты четных субъектов и указанные выше условия. Значения являются отрицательными, чтобы указать, что он является платным, а не получен.

Развернуть таблицу

[значение]	[Платеж по проценту]
0	-400
1	-300
2	-200
3	-100

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MDURATION

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает измененную длительность Macauley для безопасности с предполагаемым значением пар $100.

Синтаксис

DAX

```
MDURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

Параметры

⛶ **Развернуть таблицу**

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`coupon`	Годовая ставка купона на безопасность.
`yld`	Годовая доходность безопасности.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

⛶ **Развернуть таблицу**

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Измененная длительность Macauley.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения — 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- Измененная длительность определяется следующим образом:

  $$\text{MDURATION} = \frac{\text{DURATION}}{1 + (\frac{\text{доходность рынка}}{\text{купонные платежи в год}}$$

- расчет и зрелость усечены в целые числа.
- частота и база округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - купон < 0.
  - yld < 0

- частота — любое число, отличное от 1, 2 или 4.
- базовый < 0 или базовый > 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
1/1/2008	Дата урегулирования
1/1/2016	Дата зрелости
8%	Процентный купон
9%	Процент доходности
2	Частота является полунуальной (см. выше)
1	Фактическая/фактическая база (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  MDURATION(DATE(2008,1,1), DATE(2016,1,1), 0.08, 0.09, 2, 1)
}
```

Возвращает измененную длительность связи Macauley с помощью указанных выше условий.

Развернуть таблицу

[значение]
5.73566981391884

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NOMINAL

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает номинальную годовую процентную ставку, учитывая эффективную ставку и количество составных периодов в год.

Синтаксис

DAX

```
NOMINAL(<effect_rate>, <npery>)
```

Параметры

Развернуть таблицу

Срок	Определение
`effect_rate`	Эффективная процентная ставка.
`npery`	Количество составных периодов в год.

Возвращаемое значение

Номинальная годовая процентная ставка.

Замечания

- Связь между NOMINAL и EFFECT показана в следующем уравнении:

$$\text{EFFECT} = \left(1 + \frac{\text{nominal_rate}}{\text{npery}}\right)^{\text{npery}} - 1$$

- npery округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - effect_rate ≤ 0.

- npery < 1.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
5.3543%	Эффективная процентная ставка
4	Количество составных периодов в год

Следующий запрос DAX:

```
DAX

EVALUATE
{
  NOMINAL(0.053543, 4)
}
```

Возвращает номинальную процентную ставку, используя указанные выше условия.

Развернуть таблицу

[значение]
0.052500319868356

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NPER

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает количество периодов для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.

Синтаксис

DAX

```
NPER(<rate>, <pmt>, <pv>[, <fv>[, <type>]])
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка на период.
`pmt`	Платеж, сделанный каждый период; он не может измениться в течение жизни аннуитета. Как правило, pmt содержит субъект и проценты, но никаких других сборов или налогов.
`pv`	Текущее значение или сумма единовременной суммы, которая стоит ряд будущих платежей прямо сейчас.
`fv`	(Необязательно) Будущая стоимость или денежный баланс, который вы хотите достичь после последнего платежа. Если fv опущен, предполагается, что BLANK.
`type`	(Необязательно) Число 0 или 1 и указывает, когда должны быть выплаты. Если тип опущен, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `type` принимает следующие значения:

Развернуть таблицу

Задать тип, равный	Если платежи должны быть
0 или опущено	В конце периода
1	В начале периода

Возвращаемое значение

Количество периодов для инвестиций.

Замечания

- тип округляется до ближайшего целого числа.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
12%	Годовая процентная ставка
-100	Платеж, сделанный каждый период
-1000	Текущее значение
10000	Будущее значение
1	Платеж должен быть в начале периода (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  NPER(0.12/12, -100, -1000, 10000, 1)
}
```

Возвращает количество периодов для инвестиций, описанных выше.

Развернуть таблицу

[значение]
59.6738656742946

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ODDFPRICE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает цену за $ 100 лицевой стоимости безопасности с нечетным (коротким или длинным) первым периодом.

Синтаксис

DAX

```
ODDFPRICE(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <yld>,
<redemption>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`issue`	Дата проблемы безопасности.
`first_coupon`	Дата первого купона безопасности.
`rate`	Процентная ставка безопасности.
`yld`	Годовая доходность безопасности.
`redemption`	Стоимость активации безопасности за $100 лиц.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Цена за $100 лицовая стоимость.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- ODDFPRICE вычисляется следующим образом:

  Нечетный короткий первый купон:

  $$\text{ODDFPRICE} = \bigg[\frac{\text}{{1 + \frac{\text{yld}}{\text{{frequency}^{N - 1 + \frac{\text{DSC}}{\text{E}}}} \bigg] + \bigg[\frac{100 \times \frac{\text{rate}}{\text{частота}} \times \frac{\text{DFC}}{\text{E}}{(1 + \frac{\text\text{yld}}{\text{frequency}})^{{\frac{\text{DSC}}{\text{E}}} \size] + \bigg[\sum^{N}_{k=2} \frac{100 \times \frac{\text{rate}}{\text{frequency}{(1 + \frac{\text{yld}}

{\text{frequency}})^{k - 1 + \frac{\text{DSC}}{\text{E}}}} \ big] - \Big[100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}} \Big] $$

где:

- A = количество дней с начала периода купона до даты расчетов (накопленные дни).
- DSC = количество дней от расчетов до следующей даты купона.
- DFC = количество дней с начала нечетного первого купона до первой даты купона.
- E = количество дней в период купона.
- N = количество купонов, выплачиваемых между датой урегулирования и датой активации. (Если это число содержит дробь, оно поднимается до следующего целого числа.)

Нечетный длинный первый купон:

$$\text{ODDFPRICE} = \bigg[\frac{\text{активация}{{1 + \frac{\text{yld}}{\text{frequency}^{(\text{N} + \text{N}_{q} + \frac{\text{DSC}}{\text{E}}}} \big] + \big[\frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \Big[\sum^{\text{NC}_{i=1} \frac{\text{DC}_{i}{\text{NL}_{i}}{Big] }{{1 + \frac{\text{yld}}{\text{frequency}}^{{\text{N}_{q} \frac{\text{DSC}}{\text{E}}}} \bigg] + \bigg[\sum^{\text{N}_{k=1} \frac{100 \times \frac{\text{rate}}{\text{ frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{k - \text{N}_{q} + \frac{\text{DSC}}{\text{E}}}} \big] - \Big[100 \times \frac{\text{rate}}{\text{frequency}} \times \sum^{\text{NC}}_{i=1} \frac{\text{A}_{i}{i}{\text{NL}_{i}} \Big]$$

где:

- A_i = количество дней с начала i^{th}, или последний, квази-купонный период в нечетном периоде.
- DC_i = количество дней от даты (или даты выдачи) до первого квази-купона ($i = 1$) или количества дней в квази-купоне ($i = 2$,..., $i = \mathrm{NC}$).
- DSC = количество дней от расчета до следующей даты купона.
- E = количество дней в период купона.
- N = количество купонов, выплачиваемых между первой датой реального купона и датой активации. (Если это число содержит дробь, оно поднимается до следующего целого числа.)
- NC = количество квази-купонных периодов, которые соответствуют нечетным периодам. (Если это число содержит дробь, оно поднимается до следующего целого числа.)
- NL_i = обычная длина в днях полной i^{th}, или последний, квази-купонный период в нечетном периоде.

- N_q = количество целых квази-купонных периодов между датой расчетов и первым купоном.

- расчет, зрелость, проблема и first_coupon усечены в целые числа.
- база и частота округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет, зрелость, проблема или first_coupon не является допустимой датой.
  - проблема > first_coupon > урегулирования > не удовлетворена.
  - скорость < 0.
  - yld < 0.
  - активация ≤ 0.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание аргумента
11/11/2008	Дата урегулирования
3/1/2021	Дата зрелости
10/15/2008	Дата проблемы
3/1/2009	Дата первого купона
7.85%	Процентный купон
6.25%	Процент доходности
$100.00	Повторное значение
2	Частота является полунауальной
1	Фактическая/фактическая основа

Следующий запрос DAX:

```
DAX
```

```
EVALUATE
{
  ODDFPRICE(DATE(2008,11,11), DATE(2021,3,1), DATE(2008,10,15),
DATE(2009,3,1), 0.0785, 0.0625, 100.00, 2, 1)
}
```

Возвращает цену за $100 лицевой стоимости безопасности с нечетным (коротким или длинным) первым периодом, используя указанные выше условия.

Развернуть таблицу

[значение]
113.597717474079

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ODDFYIELD

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает доходность безопасности, которая имеет нечетный (короткий или длинный) первый период.

Синтаксис

DAX

```
ODDFYIELD(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <pr>,
<redemption>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`issue`	Дата проблемы безопасности.
`first_coupon`	Дата первого купона безопасности.
`rate`	Процентная ставка безопасности.
`pr`	Цена на безопасность.
`redemption`	Стоимость активации безопасности за $100 лиц.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Доходность безопасности.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- `settlement` дата является датой, когда покупатель покупает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- ODDFYIELD вычисляется с помощью итеративного метода. Он использует метод Newton на основе формулы, используемой для функции ODDFPRICE. Доходность изменяется до 100 итераций до тех пор, пока оценочная цена с данной доходностью близка к цене. См. ODDFPRICE формулы, которая ODDFYIELD используется.
- расчет, зрелость, проблема и first_coupon усечены в целые числа.
- база и частота округляются до ближайшего целого числа.

- Если возвращается ошибка:
  - `settlement`, `maturity`, `issue` или `first_coupon` не является допустимой датой.
  - `maturity` > `first_coupon` > `settlement` > `issue` не удовлетворено.
  - `rate` < 0.
  - `pr` ≤ 0.
  - `redemption` ≤ 0.
  - `frequency` любое число, отличное от 1, 2 или 4.
  - `basis` < 0 или `basis` > 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание аргумента
11 ноября 2008 г.	Дата урегулирования
1 марта 2021 г.	Дата зрелости
15 октября 2008 г.	Дата проблемы
1 марта 2009 г.	Дата первого купона
5.75%	Процентный купон
84.50	Цена
100	Повторное значение
2	Частота является полунауальной
0	30/360 базис

Следующий запрос DAX:

DAX

```
EVALUATE
{
  ODDFYIELD(DATE(2008,11,11), DATE(2021,3,1), DATE(2008,10,15),
DATE(2009,3,1), 0.0575, 84.50, 100, 2, 0)
}
```

Возвращает доходность безопасности, которая имеет нечетный (короткий или длинный) первый период, используя указанные выше условия.

Развернуть таблицу

[значение]
0.0772455415972989

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ODDLPRICE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает цену за $ 100 лицевой стоимости безопасности с нечетным (коротким или длинным) прошлым купонным периодом.

Синтаксис

DAX

```
ODDLPRICE(<settlement>, <maturity>, <last_interest>, <rate>, <yld>,
<redemption>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`last_interest`	Дата последнего купона безопасности.
`rate`	Процентная ставка безопасности.
`yld`	Годовая доходность безопасности.
`redemption`	Стоимость активации безопасности за $100 лиц.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Цена за $100 лицовая стоимость.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- расчет, зрелость и last_interest усечены в целые числа.
- база и частота округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет, зрелость или last_interest не является допустимой датой.
  - > урегулирования > last_interest > не удовлетворяется.
  - скорость < 0.
  - yld < 0.
  - активация ≤ 0.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX:

Развернуть таблицу

данных	описание аргумента
7 февраля 2008 г.	Дата урегулирования
15 июня 2008 г.	Дата зрелости
15 октября 2007 г.	Дата последнего интереса
3.75%	Процентный купон
4.05%	Процент доходности
$100	Повторное значение
2	Частота является полунауальной
0	30/360 базис

DAX

```
EVALUATE
{
  ODDLPRICE(DATE(2008,2,7), DATE(2008,6,15), DATE(2007,10,15), 0.0375,
0.0405, 100, 2, 0)
}
```

Возвращает цену на $100 лицевой стоимости безопасности, которая имеет нечетный (короткий или длинный) последний период купона, используя указанные выше условия.

Развернуть таблицу

[значение]
99.8782860147213

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ODDLYIELD

Статья • 14.12.2024

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает доходность безопасности, которая имеет нечетный (короткий или длинный) последний период.

Синтаксис

DAX

```
ODDLYIELD(<settlement>, <maturity>, <last_interest>, <rate>, <pr>,
<redemption>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`last_interest`	Дата последнего купона безопасности.
`rate`	Процентная ставка безопасности.
`pr`	Цена на безопасность.
`redemption`	Стоимость активации безопасности за $100 лиц.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Доходность безопасности.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.

- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.

- ODDLYIELD вычисляется следующим образом:

  $$\text{ODDLYIELD} = \bigg[\frac{(\text{активация} + ((\sum^{\text{NC}}_{i=1} \frac{\text{DC}_{i}}{\text{NL}_{i}{i}}) \times \frac{100 \times \text{rate}}{\text{frequency}})) - (\text{par} + ((\sum^{\text{NC}}_{i=1} \frac{\text{A}__{i}}{\text{NL}_{i}}) \times \frac{100 \times \text{rate}}{\text{frequency}}))}{\text{par} + ((\sum^{\text{NC}}_{i=1} \frac{\text{A}_{i}}{\text{NL}_{i}}) \times \frac{100 \times \text{rate}}{\text{frequency}}}} \size] \times \size[\frac{\text{frequency}}{(\sum^{\text{NC}}_{i=1} \frac{\text{DSC}_{i}}{\text{NL}_{i}}}} \bigg]$$

  где:

- A_i = количество накопленных дней для i^{th}, или последний, квази-купонный период в нечетном периоде подсчета от последней даты процента до активации.
- DC_i = количество дней, отсчитываемых в i^{th}, или последний, квази-купонный период, разделенный по длине фактического периода купона.
- NC = число квази-купонных периодов, которые соответствуют нечетным периодам; Если это число содержит дробь, она будет поднята до следующего целого числа.
- NL_i = обычная длина в днях i^{th}, или последний, квази-купонный период в нечетном периоде купона.

- расчет, зрелость, last_interest усечены по целым числам.
- база и частота округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет, зрелость, last_interest не является допустимой датой.
  - > урегулирования > last_interest > не удовлетворяется.
  - скорость < 0.
  - pr ≤ 0.
  - активация ≤ 0.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX:

Развернуть таблицу

данных	описание аргумента
4/20/2008	Дата урегулирования
6/15/2008	Дата зрелости
12/24/2007	Дата последнего интереса
3.75%	Процентный купон

данных	описание аргумента
$99,875	Цена
$100	Значение активации
2	Частота является полунауальной
0	30/360 базис

DAX

```
EVALUATE
{
  ODDLYIELD(DATE(2008,4,20), DATE(2008,6,15), DATE(2007,12,24), 0.0375,
99.875, 100, 2, 0)
}
```

Возвращает доходность безопасности, которая имеет нечетный (короткий) последний период, используя указанные выше условия.

Развернуть таблицу

[значение]
0.0451922356291692

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PDURATION

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает количество периодов, необходимых инвестиции для достижения указанной стоимости.

Синтаксис

DAX

```
PDURATION(<rate>, <pv>, <fv>)
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка на период.
`pv`	Текущая стоимость инвестиций.
`fv`	Желаемая будущая стоимость инвестиций.

Возвращаемое значение

Количество периодов.

Замечания

- PDURATION использует следующее уравнение:

$$\text{PDURATION} = \frac{log(\text{fv}) - log(\text{pv})}{log(1 + \text{rate})}$$

- Если возвращается ошибка:
  - скорость ≤ 0.

- pv ≤ 0.
- fv ≤ 0.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
{
  PDURATION(0.025, 2000, 2200)
}
```

Возвращает количество лет, необходимых для инвестиций $2000, зарабатывая 2,5% ежегодно, чтобы достичь $2200.

Развернуть таблицу

[значение]
3.85986616262266

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE
{
  PDURATION(0.025/12, 1000, 1200)
}
```

Возвращает количество месяцев, необходимых для инвестиций в размере $1000, заработав 2,5% ежегодно, чтобы достичь $1200.

Развернуть таблицу

[значение]
87.6054764193714

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PMT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Вычисляет платеж за кредит на основе постоянных платежей и постоянной процентной ставки.

Синтаксис

```
DAX
PMT(<rate>, <nper>, <pv>[, <fv>[, <type>]])
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка для кредита.
`nper`	Общее количество платежей за кредит.
`pv`	Текущее значение или общая сумма, которую стоит ряд будущих платежей; также называется субъектом.
`fv`	(Необязательно) Будущая стоимость или денежный баланс, который вы хотите достичь после последнего платежа. Если fv опущен, предполагается, что BLANK.
`type`	(Необязательно) Число 0 или 1, указывающее, когда должны быть выплаты. Если тип опущен, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `type` принимает следующие значения:

Развернуть таблицу

Задать `type` равным	Если платежи должны быть
0 или опущено	В конце периода

Задать `type` равным	Если платежи должны быть
1	В начале периода

Примечание. Более полное описание аргументов в PMTсм. в PV функции.

Возвращаемое значение

Сумма единого платежа по кредиту.

Замечания

- Платеж, возвращаемый PMT включает в себя субъект и проценты, но не налоги, резервные платежи или сборы иногда связаны с кредитами.
- Убедитесь, что вы согласованы с единицами, используемыми для указания скорости и nper. Если вы вносите ежемесячные платежи по четырехлетнему кредиту на годовую процентную ставку 12 процентов, используйте 0,12/12 для ставки и 4*12 для nper. Если вы вносите ежегодные платежи по тому же кредиту, используйте 0,12 для ставки и 4 для nper.
- `type` округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - `nper < 1`

совет: чтобы найти общую сумму, выплачиваемую в течение срока кредита, умножьте возвращаемое PMT значение на nper.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Пример 1

Развернуть таблицу

данных	описание
8%	Годовая процентная ставка
10	Количество месяцев платежей
$10,000	Сумма кредита

Следующий запрос DAX:

```
DAX

EVALUATE
{
  PMT(0.08/12, 10, 10000, 0, 1)
}
```

Возвращает ежемесячную сумму платежа, выплачиваемую в начале месяца, за кредит с указанными выше условиями.

Развернуть таблицу

[значение]
-1030.16432717797

Примечание. 1030.16432717797 — это платеж за период. В результате общая сумма, выплачиваемая в течение срока кредита, составляет примерно 1030,16 * 10 = \10,301,60 долл. США. Другими словами, оплачивается примерно $301,60 процентов.

Пример 2

Развернуть таблицу

данных	описание
6%	Годовая процентная ставка
18	Количество лет платежей
$50,000	Сумма кредита

Следующий запрос DAX:

```
DAX
```

```
EVALUATE
{
  PMT(0.06/12, 18*12, 0, 50000)
}
```

Развернуть таблицу

[значение]
-129.081160867991

Возвращает сумму для экономии каждый месяц, чтобы иметь \50 000 долларов в конце 18 лет, используя указанные выше условия.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PPMT

Статья • 16.12.2024

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает платеж по субъекту в течение заданного периода для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.

Синтаксис

DAX

```
PPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка для кредита.
`per`	Указывает период. Должно быть от 1 до nper (включительно).
`nper`	Общее количество периодов оплаты в аннуитете.
`pv`	Текущее значение — общая сумма, которую стоит ряд будущих платежей.
`fv`	(Необязательно) Будущая стоимость или денежный баланс, который вы хотите достичь после последнего платежа. Если fv опущен, предполагается, что fv будет пустым.
`type`	(Необязательно) Число 0 или 1, указывающее, когда должны быть выплаты. Если тип опущен, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `type` принимает следующие значения:

Развернуть таблицу

Задать `type` равным	Если платежи должны быть
0 или опущено	В конце периода
1	В начале периода

Примечание. Более полное описание аргументов в PPMT см. в функции PV.

Возвращаемое значение

Платеж по субъекту в течение заданного периода.

Замечания

- Убедитесь, что вы согласованы с единицами, используемыми для указания `rate` и `nper`. Если вы вносите ежемесячные платежи по четырехлетнему кредиту на годовую процентную ставку 12 процентов, используйте 0,12/12 для ставки и 4*12 для nper. Если вы вносите ежегодные платежи по одному кредиту, используйте 0,12 для `rate` и 4 для `nper`.
- тип округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - `per < 1` или `per > nper`
  - `nper < 1`
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

Развернуть таблицу

данных	описание аргумента
10%	Годовая процентная ставка
2	Количество лет для кредита
$2000,000	Сумма кредита

Следующий запрос DAX:

DAX

```
EVALUATE
{
  PPMT(0.1/12, 1, 2*12, 2000.00)
}
```

Возвращает основной платеж, сделанный в первом месяце для кредита с указанными выше условиями.

Развернуть таблицу

[значение]
-75.6231860083663

Пример 2

Развернуть таблицу

данных	описание аргумента
8%	Годовая процентная ставка
10	Количество лет для кредита
$200,000,00	Сумма кредита

Следующий запрос DAX:

DAX

```
EVALUATE
{
  PPMT(0.08, 10, 10, 200000.00)
}
```

Возвращает основной платеж, сделанный в 10-м году за кредит с указанными выше условиями.

Развернуть таблицу

[значение]
-27598.0534624214

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ЦЕНА

Статья • 15.12.2024

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает цену на $100 лицевой стоимости безопасности, которая платит периодический процент.

Синтаксис

DAX

```
PRICE(<settlement>, <maturity>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`rate`	Годовая ставка купона на безопасность.
`yld`	Годовая доходность безопасности.
`redemption`	Стоимость активации безопасности за $100 лиц.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Цена за $100 лицовая стоимость.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- расчет и зрелость усечены в целые числа.
- база и частота округляются до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - скорость < 0.
  - yld < 0.
  - активация ≤ 0.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Важно:

- Если N > 1 (N — это количество купонов, выплачиваемых между датой урегулирования и датой активации), **PRICE** вычисляется следующим образом:

  $$\text{PRICE} = \bigg[\frac{\text{активация}}{(1 + \frac{\text{yld}}{\text{частота}})^{(N - 1 + \frac{\text{DSC}}{\text{E}})}} \bigg] + \bigg[\sum^{N}_{k=1} \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{k - 1 + \frac{\text{DSC}}{\text{E}}}} \bigg] - \bigg[100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}} \bigg]$$

- Если N = 1 (N — это количество купонов, выплачиваемых между датой урегулирования и датой активации), **PRICE** вычисляется следующим образом:

  $$\mathrm{DSR} = \mathrm{E} - \mathrm{A}$$

  $$\mathrm{T1} = 100 \times \frac{\mathrm{rate}}{\mathrm{frequency}} + \text{активация}$$

  $$\mathrm{T2} = \frac{\mathrm{yld}}{\mathrm{frequency}} \times \frac{\mathrm{DSR}}{\mathrm{E}} + 1$$

  $$\text{T3} = 100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}}$$

  $$\mathrm{PRICE} = \frac{\mathrm{T1}}{\mathrm{T2}} - \mathrm{T3}$$

  где:
  - DSC = количество дней от расчета до следующей даты купона.
  - E = количество дней в период купона, в котором падает дата урегулирования.
  - A = количество дней с начала периода купона до даты урегулирования.

Пример

Развернуть таблицу

данных	описание аргумента
2/15/2008	Дата урегулирования
11/15/2017	Дата зрелости
5.75%	Процент полуфиналового купона
6.50%	Процент доходности
$100	Значение активации
2	Частота является полунауальной
0	30/360 базис

Следующий запрос DAX:

DAX

```
EVALUATE
{
  PRICE(DATE(2008,2,15), DATE(2017,11,15), 0.0575, 0.065, 100, 2, 0)
}
```

Возвращает цену облигации для облигации, используя указанные выше условия.

Развернуть таблицу

[значение]
94.6343616213221

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PRICEDISC

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает цену за $100 с учетом стоимости скидок.

Синтаксис

```
DAX

PRICEDISC(<settlement>, <maturity>, <discount>, <redemption>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`discount`	Скидка на безопасность.
`redemption`	Стоимость активации безопасности за $100 лиц.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	`Day count basis`
0 или опущено	США (NASD) 30/360
1	Фактический/фактический

`Basis`	`Day count basis`
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Цена за $100 лицовая стоимость.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выпущена 1 января 2018 года и приобретена покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2018 г., дата урегулирования будет 1 июля 2018 года, а дата погашения будет 1 января 2048 г. 30 лет после 1 января 2018 г.
- PRICEDISC вычисляется следующим образом:

$$\text{PRICEDISC} = \text{активация} - \text{скидка} \times \text{активация} \times \frac{\text{DSM}}{\text{B}}$$

  где:
  - B = количество дней в году в зависимости от года.
  - DSM = количество дней от расчета до срока действия.
- расчет и зрелость усечены в целые числа.
- база округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - скидка ≤ 0.
  - активация ≤ 0.
  - базовый < 0 или базовый > 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание аргумента
2/16/2008	Дата урегулирования
3/1/2008	Дата зрелости
5.25%	Процент скидки
$100	Значение активации
2	Фактическая/360 база

Следующий запрос DAX:

```
DAX

EVALUATE
{
  PRICEDISC(DATE(2008,2,16), DATE(2008,3,1), 0.0525, 100, 2)
}
```

Возвращает цену облигации на $100 лицовую стоимость для облигации с указанными выше условиями.

Развернуть таблицу

[значение]
99.7958333333333

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PRICEMAT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает цену на $100 лицевой стоимости безопасности, которая платит проценты по зрелости.

Синтаксис

DAX

```
PRICEMAT(<settlement>, <maturity>, <issue>, <rate>, <yld>[, <basis>])
```

Параметры

Развернуть таблицу

Срок	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`issue`	Дата проблемы безопасности.
`rate`	Процентная ставка безопасности на дату выдачи.
`yld`	Годовая доходность безопасности.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Цена за $100 лицовая стоимость.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.

- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.

- PRICEMAT вычисляется следующим образом:

  $$\text{PRICEMAT} = \frac{100 + (\frac{\text{DIM}}{\text{B}} \times \text{rate} \times 100)}{1 + (\frac{()\text{DSM}}{\text{B}} \times \text{yld})} - (\frac{\text{A}}{\text{B}} \times \text{rate} \times 100)$$

  где:
  - B = количество дней в году в зависимости от года.
  - DSM = количество дней от расчета до срока действия.
  - DIM = количество дней от выдачи до зрелости.
  - A = количество дней от выдачи до урегулирования.

- расчет, зрелость и проблема усечены в целые числа.

- база округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - дата урегулирования, зрелости или выдачи не является допустимой датой.
  - Проблема > урегулирования >> не удовлетворена.
  - скорость < 0.
  - yld < 0.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX:

Развернуть таблицу

данных	описание
2/15/2008	Дата урегулирования
4/13/2008	Дата зрелости
11/11/2007	Дата проблемы
6.10%	Процент полуфиналового купона
6.10%	Процент доходности
0	30/360 базис

DAX

```
EVALUATE
{
  PRICEMAT(DATE(2008,2,15), DATE(2008,4,13), DATE(2007,11,11), 0.061, 0.061,
0)
}
```

Возвращает цену за $100 лицевой стоимости безопасности с указанными выше условиями.

Развернуть таблицу

[значение]
99.9844988755569

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PV

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Вычисляет текущую стоимость кредита или инвестиций на основе постоянной процентной ставки. Вы можете использовать PV с периодическими, постоянными платежами (например, ипотекой или другим кредитом) и /или будущей стоимостью, которая является вашей инвестиционной целью.

Синтаксис

DAX

```
PV(<rate>, <nper>, <pmt>[, <fv>[, <type>]])
```

Параметры

Развернуть таблицу

Срок	Определение
`rate`	Процентная ставка на период. Например, если вы получаете кредит на автомобиле на 10 процентов годовых процентных ставок и вносите ежемесячные платежи, процентная ставка в месяц составляет 0,1/12 или 0,0083. Вы введите 0.1/12 или 0,0083 в формулу в качестве скорости.
`nper`	Общее количество периодов оплаты в аннуитете. Например, если вы получаете четырехлетний кредит на автомобиль и вносите ежемесячные платежи, ваш кредит имеет 4*12 (или 48) периодов. Вы введете 48 в формулу для nper.
`pmt`	Платеж сделал каждый период, который не может измениться за всю жизнь аннуитета. Как правило, pmt включает в себя субъект и проценты, но никаких других сборов или налогов. Например, ежемесячные платежи по $10 000, четырехлетний кредит на автомобиль на 12 процентов составляет $263,33. Вы введите -263.33 в формулу в качестве pmt.
`fv`	(Необязательно) Будущая стоимость или денежный баланс, который вы хотите достичь после последнего платежа. Если fv опущен, предполагается, что BLANK. Например, если вы хотите сэкономить \50 000 долларов, чтобы заплатить за специальный проект в течение 18 лет, то $ 50 000 является будущим значением.

Срок	Определение
	Затем вы можете сделать консервативное предположение на процентной ставке и определить, сколько вы должны сэкономить каждый месяц.
`type`	(Необязательно) Число 0 или 1, указывающее, когда должны быть выплаты. Если тип опущен, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `type` принимает следующие значения:

Развернуть таблицу

Задать `type` равным	Если платежи должны быть
0 или опущено	В конце периода
1	В начале периода

Возвращаемое значение

Текущая стоимость кредита или инвестиций.

Замечания

- Убедитесь, что вы согласованы с единицами, используемыми для указания скорости и nper. Если вы вносите ежемесячные платежи по четырехлетнему кредиту на 12 процентов годовых процентов, используйте 0,12/12 для ставки и 4*12 для nper. Если вы вносите ежегодные платежи по тому же кредиту, используйте 0,12 для ставки и 4 для nper.
- Следующие функции применяются к аннуитетам:
  - `CUMIPMT`
  - `CUMPRINC`
  - `FV`
  - `IPMT`
  - `PMT`
  - `PPMT`
  - `PV`
  - `RATE`
  - `XIRR`
  - `XNPV`

- Аннуитет — это ряд постоянных денежных платежей, сделанных в течение непрерывного периода. Например, кредит на автомобиль или ипотеку является аннуитетом. Дополнительные сведения см. в описании каждой функции аннуитета.
- В функциях аннуитета денежные средства, которые вы оплачиваете, например депозит для сбережений, представлено отрицательным числом; полученные вами деньги, такие как дивидендный контроль, представлены положительным числом. Например, депозит $1000 в банк будет представлен аргументом -1000, если вы являетесь вкладчиком и аргументом 1000, если вы банк.
- Один финансовый аргумент решается с точки зрения других.
  - Если ставка не равен 0, то:

    $$\text{pv} \times (1 + \text{rate})^{\text{nper}} + \text{pmt}(1 + \text{rate} \times \text{type}) \times \bigg(\frac{{1 + \text{rate}})^{\text{nper}} - 1} {text{rate}} \bigg) + \text{fv} = 0$$
  - Если значение равно 0, то:

$$(\text{pmt} \times \text{nper}) + \text{pv} + \text{fv} = 0$$

- тип округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - nper < 1 или пусто
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

данных	описание
$500,00	Деньги выплачиваются из страхового аннуитета в конце каждого месяца.
8%	Процентная ставка, заработанную на деньгах, выплаченных.
20	Годы деньги будут выплачены.

Следующий запрос DAX:

DAX

```
EVALUATE
{
  PV(0.08/12, 12*20, 500.00, 0, 0)
}
```

Возвращает текущее значение аннуитета, используя указанные выше термины.

Развернуть таблицу

[значение]
-59777.1458511878

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

СКОРОСТЬ

Статья • 04.07.2024

Область применения: вычисление вычисляемого столбца вычисляемой таблицы Меры визуального элемента

Возвращает процентную ставку за период аннуитета. Rate вычисляется путем итерации и может иметь ноль или более решений. Если последовательные результаты RATE не конвергентируются в пределах 0,0000001 после 20 итерации, возвращается ошибка.

Синтаксис

DAX

```
RATE(<nper>, <pmt>, <pv>[, <fv>[, <type>[, <guess>]]])
```

Параметры

Развернуть таблицу

Термин	Определение
nper	Общее количество периодов оплаты в аннуитете.
pmt	Платеж сделал каждый период и не может измениться в течение срока действия аннуитета. Как правило, pmt включает в себя субъект и проценты, но никаких других сборов или налогов.
pv	Текущее значение — общая сумма, которую стоит ряд будущих платежей.
fv	(Необязательно) Будущая стоимость или денежный баланс, который вы хотите достичь после последнего платежа. Если fv опущен, предполагается, что значение 0 (в будущем значение кредита, например, равно 0).
type	(Необязательно) Число 0 или 1, указывающее, когда должны быть выплаты. Если тип опущен, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.
догадка	(Необязательно) Ваше предположение о том, что будет ставка. — Если опущено, предполагается, что 10 %. — Если скорость не конвергентна, попробуйте использовать разные значения для угадок. СКОРОСТЬ обычно конвергентно, если угадывание составляет от 0 до 1.

Параметр **типа** принимает следующие значения:

Развернуть таблицу

Задать тип, равный	Если платежи должны быть в счете
0 или опущено	В конце периода
1	В начале периода

Возвращаемое значение

Процентная ставка за период.

Замечания

- Убедитесь, что вы согласованы с единицами, используемыми для указания угадок и nper. Если вы вносите ежемесячные платежи по четырехлетнему кредиту на 12 процентов годовых процентов, используйте 0,12/12 для угадок и 4*12 для nper. Если вы делаете ежегодные платежи по одному кредиту, используйте 0,12 для угадок и 4 для nper.
- тип округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - nper ≤ 0.
  - RATE не конвергентирует в пределах 0,0000001 после 20 итераций
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Развернуть таблицу

Данные	Description
4	Годы кредита
-200	Ежемесячный платеж
8000	Сумма кредита

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
{
  RATE(4*12, -200, 8000)
}
```

Возвращает ежемесячную ставку кредита, используя указанные выше условия.

Развернуть таблицу

[значение]
0.00770147248820137

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE
{
  RATE(4*12, -200, 8000) * 12
}
```

Возвращает годовую ставку кредита, используя указанные выше условия.

Развернуть таблицу

[значение]
0.0924176698584164

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RECEIVED

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает сумму, полученную по сроку погашения для полностью инвестируемой безопасности.

Синтаксис

```
DAX

RECEIVED(<settlement>, <maturity>, <investment>, <discount>[, <basis>])
```

Параметры

Развернуть таблицу

Термин	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`investment`	Сумма, вложенная в безопасность.
`discount`	Скидка на безопасность.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	подсчета дней
0 или опущено	США (NASD) 30/360

`Basis`	**подсчета дней**
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Сумма, полученная по истечении срока действия.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- RECEIVED вычисляется следующим образом:

  $$\text{RECEIVED} = \frac{\text{инвестиции}}{1 - (\text{скидка} \times \frac{\text{DIM}}{\text{B}})} $$

  где:
  - B = количество дней в год в зависимости от года.
  - DIM = количество дней от выдачи до зрелости.
- расчет и зрелость усечены в целые числа.
- база округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.

- инвестиции ≤ 0.
- скидка ≤ 0.
- базовый < 0 или базовый > 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX:

Развернуть таблицу

Данные	Description
15-февраль-08	Дата урегулирования (проблема)
15-май-08	Дата зрелости
1 000 000,00 долл. США	Инвестиции
5.75%	Процент скидки
2	Фактическая/360 база

DAX

```
EVALUATE
{
  RECEIVED(DATE(2008,2,15), DATE(2008,5,15), 1000000.00, 0.0575, 2)
}
```

Возвращает общую сумму, полученную при зрелости, для облигации с указанными выше условиями.

Развернуть таблицу

[значение]
1014584.6544071

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RRI

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает эквивалентную процентную ставку для роста инвестиций.

Синтаксис

```
DAX
RRI(<nper>, <pv>, <fv>)
```

Параметры

Развернуть таблицу

Термин	Определение
`nper`	Количество периодов для инвестиций.
`pv`	Текущая стоимость инвестиций.
`fv`	Будущая стоимость инвестиций.

Возвращаемое значение

Эквивалентная процентная ставка.

Замечания

- RRI возвращает процентную ставку, указанную nper (количество периодов), pv (текущее значение) и fv (будущее значение), вычисляемое с помощью следующего уравнения:

$$\left(\frac{\mathrm{fv}}{\mathrm{pv}}\right)^{\left(\frac{1}{nper}\right)} - 1$$

- Если возвращается ошибка:

- nper ≤ 0.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Данные	Description
10 000 долл. США	Текущее значение
21 000 долл. США	Будущее значение
4	Годы инвестиций

Следующий запрос DAX:

```
DAX

EVALUATE
{
  RRI(4*12, 10000, 21000)
}
```

Возвращает эквивалентную процентную ставку для роста инвестиций с указанными выше условиями.

Развернуть таблицу

[значение]
0.0155771057566627

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SLN

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает прямую амортизацию актива в течение одного периода.

Синтаксис

```
DAX
SLN(<cost>, <salvage>, <life>)
```

Параметры

Развернуть таблицу

Термин	Определение
`cost`	Начальная стоимость актива.
`salvage`	Значение в конце амортизации (иногда называется значением спасения ресурса).
`life`	Количество периодов, за которые ресурс не рекомендуется (иногда называется полезной жизнью актива).

Возвращаемое значение

Амортизация прямой линии в течение одного периода.

Замечания

- Ошибка возвращается, если: life = 0.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Данные	Description
30 000 долл. США	Себестоимость
7500 долл. США	Значение salvage
10	Годы полезной жизни

Следующий запрос DAX:

```
DAX

EVALUATE
{
  SLN(30000, 7500, 10)
}
```

Возвращает ежегодное амортизационные пособия с использованием указанных выше условий.

Развернуть таблицу

[значение]
2250

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SYD

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает цифры амортизации актива в течение указанного периода.

Синтаксис

DAX

```
SYD(<cost>, <salvage>, <life>, <per>)
```

Параметры

Развернуть таблицу

Термин	Определение
`cost`	Начальная стоимость актива.
`salvage`	Значение в конце амортизации (иногда называется значением спасения ресурса).
`life`	Количество периодов, за которые ресурс не рекомендуется (иногда называется полезной жизнью актива).
`per`	Период. Должен использовать те же единицы, что и жизнь. Должно быть от 1 до жизни (включительно).

Возвращаемое значение

Суммарные цифры амортизации за указанный период.

Замечания

- SYD вычисляется следующим образом:

$$\text{SYD} = \frac{(\text{cost} - \text{salvage}) \times (\text{life} - \text{per} + 1) \times 2}{(\text{life}) \times (\text{life} + 1)}$$

- Если возвращается ошибка:
  - life < 1.
  - за < 1 или за > жизни.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Развернуть таблицу

Данные	Description
30 000,00 долл. США	Первоначальная стоимость
7500,00 долл. США	Значение salvage
10	Срок жизни в течение многих лет

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
{
  SYD(30000.00, 7500.00, 10, 1)
}
```

Возвращает суммарные цифры амортизации актива в течение первого года, учитывая указанные выше условия.

Развернуть таблицу

[значение]
4090.90909090909

Пример 2

Следующий запрос DAX:

DAX

```
EVALUATE
{
  SYD(30000.00, 7500.00, 10, 10)
}
```

Возвращает суммарные цифры амортизации актива за десятый (последний) год, учитывая указанные выше условия.

Развернуть таблицу

[значение]
409.090909090909

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TBILLEQ

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает эквивалентную облигацию доходность для казначейского счета.

Синтаксис

```DAX
TBILLEQ(<settlement>, <maturity>, <discount>)
```

Параметры

Развернуть таблицу

Термин	Определение
`settlement`	Дата урегулирования казначейского счета. Дата урегулирования безопасности — это дата после даты выдачи, когда казначейские счета торгуются покупателю.
`maturity`	Дата зрелости казначейского законопроекта. Дата окончания срока действия казначейского счета.
`discount`	Ставка скидки казначейского счета.

Возвращаемое значение

Эквивалентная доходность облигаций казначейского законопроекта.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- TBILLEQ вычисляется следующим образом:

$$\text{TBILLEQ} = \frac{365 \times \text{скидка}}{360 - (\text{скидка} \times \text{DSM})}$$

где:

- DSM — это количество дней между расчетом и зрелости, вычисляемой в соответствии с 360 днями в год.

- расчет и зрелость усечены в целые числа.

- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости или зрелости составляет более одного года после урегулирования.
  - скидка ≤ 0.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Данные	Description
3/31/2008	Дата урегулирования
6/1/2008	Дата зрелости
9.14%	Процент скидки

Следующий запрос DAX:

```
DAX

EVALUATE
{
  TBILLEQ(DATE(2008,3,31), DATE(2008,6,1), 0.0914)
}
```

Возвращает эквивалентную облигацию доходность для казначейского счета, используя указанные выше условия.

Развернуть таблицу

[значение]
0.094151493565943

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TBILLPRICE

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает цену за $100 лицовую стоимость для казначейского счета.

Синтаксис

DAX

```
TBILLPRICE(<settlement>, <maturity>, <discount>)
```

Параметры

Развернуть таблицу

Термин	Определение
`settlement`	Дата урегулирования казначейского счета. Дата урегулирования безопасности — это дата после даты выдачи, когда казначейские счета торгуются покупателю.
`maturity`	Дата зрелости казначейского законопроекта. Дата окончания срока действия казначейского счета.
`discount`	Ставка скидки казначейского счета.

Возвращаемое значение

Цена казначейского счета за $100 лицевой стоимости.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- TBILLPRICE вычисляется следующим образом:

$$\text{TBILLPRICE} = 100 \times (1 - \frac{\text{скидка} \times \text{DSM}}{360})$$

где:

- DSM = количество дней от расчетов до срока погашения, за исключением любой даты погашения, которая составляет более одного календарного года после даты урегулирования.

- расчет и зрелость усечены в целые числа.

- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости или зрелости составляет более одного года после урегулирования.
  - скидка ≤ 0.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Данные	Description
3/31/2008	Дата урегулирования
6/1/2008	Дата зрелости
9.0%	Процент скидки

Следующий запрос DAX:

```
DAX
EVALUATE
{
  TBILLPRICE(DATE(2008,3,31), DATE(2008,6,1), 0.09)
}
```

Возвращает цену казначейского счета за $100 лиц, учитывая указанные выше условия.

Развернуть таблицу

[значение]
98.45

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TBILLYIELD

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает доходность казначейского счета.

Синтаксис

```
DAX

TBILLYIELD(<settlement>, <maturity>, <pr>)
```

Параметры

Развернуть таблицу

Термин	Определение
`settlement`	Дата урегулирования казначейского счета. Дата урегулирования безопасности — это дата после даты выдачи, когда казначейские счета торгуются покупателю.
`maturity`	Дата зрелости казначейского законопроекта. Дата окончания срока действия казначейского счета.
`pr`	Цена казначейского счета за $100 лиц.

Возвращаемое значение

Доходность казначейского законопроекта.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- TBILLYIELD вычисляется следующим образом:

$$\text{TBILLYIELD} = \frac{100 - \text{pr}}{\text{pr}} \times \frac{360}{\text{DSM}}$$

где:

- DSM = количество дней от расчетов до срока погашения, за исключением любой даты погашения, которая составляет более одного календарного года после даты урегулирования.

- расчет и зрелость усечены в целые числа.

- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости или зрелости составляет более одного года после урегулирования.
  - pr ≤ 0.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX:

Развернуть таблицу

Данные	Description
3/31/2008	Дата урегулирования
6/1/2008	Дата зрелости
98,45 долл. США	Цена за $100 лицовая стоимость

DAX

```
EVALUATE
{
  TBILLYIELD(DATE(2008,3,31), DATE(2008,6,1), 98.45)
}
```

Возвращает доходность казначейского счета, используя указанные выше условия.

Развернуть таблицу

[значение]
0.0914169629253426

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

VDB

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает амортизацию актива в течение любого указанного периода, включая частичные периоды, используя метод двойного снижения баланса или другой указанный метод. VDB означает снижение баланса переменных.

Синтаксис

```
DAX

VDB(<cost>, <salvage>, <life>, <start_period>, <end_period>[, <factor>[,
<no_switch>]])
```

Параметры

Развернуть таблицу

Термин	Определение
`cost`	Начальная стоимость актива.
`salvage`	Значение в конце амортизации (иногда называется значением спасения ресурса). Это значение может быть равно 0.
`life`	Количество периодов, в течение которых ресурс не рекомендуется (иногда называется полезной жизнью актива).
`start_period`	Начальный период, для которого требуется вычислить амортизацию. Start_period должны использовать те же единицы, что и жизнь. Должно быть от 1 до жизни (включительно).
`end_period`	Конечный период, для которого требуется вычислить амортизацию. End_period должны использовать те же единицы, что и жизнь. Должно быть между start_period и жизнью (включительно).
`factor`	(Необязательно) Скорость снижения баланса. Если фактор опущен, предполагается, что значение равно 2 (метод двойного уменьшения баланса). Коэффициент изменения, если вы не хотите использовать метод двойного уменьшения баланса. Описание метода двойного уменьшения баланса см. в DDB.

Термин	Определение
`no_switch`	(Необязательно) Логическое значение, указывающее, следует ли переключаться на амортизацию прямой линии, если амортизация больше, чем вычисление уменьшения баланса. Если опущено, предполагается, что `FALSE`. - Если no_switch оценивается как `TRUE` VDB не переключается на амортизацию прямой линии, даже если амортизация больше, чем вычисление снижения баланса. — Если no_switch оценивается как `FALSE` или опущена, VDB переключается на амортизацию прямой линии, если амортизация больше, чем вычисление снижения баланса.

Возвращаемое значение

Амортизация за указанный период.

Замечания

- Если возвращается ошибка:
  - cost < 0.
  - salvage < 0.
  - life < 1.
  - start_period < 1 или start_period > end_period.
  - end_period < start_period или end_period > жизни.
  - factor < 0.
  - no_switch не оценивается как `TRUE`, так и `FALSE`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Развернуть таблицу

Данные	Description
2400	Первоначальная стоимость
300	Значение salvage

Данные	Description
10	Время существования в течение многих лет

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
{
  VDB(2400, 300, 10*365, 0, 1)
}
```

Возвращает амортизацию актива первого дня с помощью коэффициента 2.

Развернуть таблицу

[значение]
1.31506849315068

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE
{
  VDB(2400, 300, 10*12, 6, 18, 3)
}
```

Возвращает амортизацию актива между месяцем 6th и 18th месяц. Этот расчет использует коэффициент 3.

Развернуть таблицу

[значение]
540.185558199698

Пример 3

Следующий запрос DAX:

```
DAX

EVALUATE
{
  VDB(2400, 300, 10, 0, 0.875, 1.5)
}
```

Возвращает амортизацию актива в первый финансовый год, который вы владеете им, при условии, что налоговые законы ограничивают вас до 150% амортизации снижения баланса. Актив приобретается в середине первого квартала финансового года.

Развернуть таблицу

[значение]
315

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

XIRR

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает внутреннюю ставку прибыли для расписания денежных потоков, которые не обязательно периодически.

Синтаксис

DAX

```
XIRR(<table>, <values>, <dates>, [, <guess>[, <alternateResult>]])
```

Параметры

Развернуть таблицу

Термин	Определение
`table`	Таблица, для которой должны вычисляться значения и выражения дат.
`values`	Выражение, возвращающее значение денежного потока для каждой строки таблицы.
`dates`	Выражение, возвращающее дату денежного потока для каждой строки таблицы.
`guess`	(Необязательно) Начальное предположение для внутренней ставки возврата. Если опущено, используется предположение по умолчанию 0.1.
`alternateResult`	(Необязательно) Значение, возвращаемое вместо ошибки, если решение не может быть определено.

Возвращаемое значение

Внутренняя ставка возврата для заданных входных данных. Если вычисление не возвращает допустимый результат, возвращается ошибка или значение, указанное как альтернативныйResult.

Замечания

- Значение вычисляется как скорость, которая удовлетворяет следующей функции:

$$0 = \sum_{j=1}^{N} \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

  Где:
  - P_j — $j^{ü}$ платеж
  - d_j — это дата оплаты j^{th}
  - d_1 является первой датой оплаты

- Ряд значений денежного потока должен содержать по крайней мере одно положительное число и одно отрицательное число.

- Избегайте использования функций ISERROR или IFERROR для записи ошибки, возвращаемой XIRR. Если некоторые входные данные функции не могут привести к ошибке решения, предоставление альтернативного параметраResult является самым надежным и самым высоким способом обработки ошибки.

- Чтобы получить дополнительные сведения об использовании параметра alternateResult, посмотрите этот видеоролик.

- Если абсолютное значение первоначального платежа невелико, расчет, скорее всего, не возвращает допустимый результат.

- Избегайте включения 0 значений в платежи. Они не будут влиять на окончательный результат, и использование 0 в качестве первоначального платежа всегда завершится сбоем XIRR() вычислений.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула вычисляет внутреннюю ставку возврата таблицы CashFlows:

```
DAX
```

```
= XIRR( CashFlows, [Payment], [Date] )
```

Развернуть таблицу

Дата	Оплата
01.01.2014	10000–
3/1/2014	2750
10/30/2014	4250
2/15/2015	3 250
4/1/2015	2750

Ставка возврата = 37,49%

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

XNPV

Статья • 14.12.2024

Область применения: вычисление вычисляемого столбца вычисляемой таблицы Меры визуального элемента

Возвращает текущее значение для расписания денежных потоков, которые не обязательно периодически.

Синтаксис

```
DAX
XNPV(<table>, <values>, <dates>, <rate>)
```

Параметры

Развернуть таблицу

Термин	Определение
`table`	Таблица, для которой должны вычисляться значения и выражения дат.
`values`	Выражение, возвращающее значение денежного потока для каждой строки таблицы.
`dates`	Выражение, возвращающее дату денежного потока для каждой строки таблицы.
`rate`	Ставка скидки, применяемая к денежному потоку для каждой строки таблицы.

Возвращаемое значение

Чистое текущее значение.

Замечания

- Значение вычисляется как следующее суммирование:

$$\sum_{j=1}^{N} \frac{P_j}{(1+\text{rate})^{\frac{d_j - d_1}{365}}}$$

Где:
  - P_j — $j^{ый}$ платеж
  - d_j — дата $j^{го}$ платежа
  - d_1 — дата первого платежа

- Ряд значений денежного потока должен содержать по крайней мере одно положительное число и одно отрицательное число.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Ниже вычисляется текущее значение таблицы CashFlows:

```
DAX
= XNPV( CashFlows, [Payment], [Date], 0.09 )
```

Развернуть таблицу

`Date`	**Оплата**
`1/1/2014`	10000–
`3/1/2014`	2750
`10/30/2014`	4250
`2/15/2015`	3 250
`4/1/2015`	2750

Текущее значение = 2089.50

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

YIELD

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает доходность по безопасности, которая платит периодический процент. Используйте YIELD для вычисления доходности облигаций.

Синтаксис

DAX

```
YIELD(<settlement>, <maturity>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

Параметры

Развернуть таблицу

Термин	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`rate`	Годовая ставка купона на безопасность.
`pr`	Цена на цену безопасности на $100 лицовую стоимость.
`redemption`	Стоимость активации безопасности за $100 лиц.
`frequency`	Количество купонных платежей в год. Для ежегодных платежей частота = 1; для полуналога, частота = 2; для ежеквартально, частота = 4.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Доходность безопасности.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.

- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.

- Если существует один период купона или меньше, YIELD вычисляется следующим образом:

  $$\text{YIELD} = \frac{(\frac{\text{активация}}{100} + \frac{\text{{rate}}{\text{frequency}}) - (\frac{\text{par}}{100} + (\frac{\text{A}}{\text{E}} \times \frac{\text{\text}}{\text{frequency}{\frac{\text{par}}{100} + (\frac{\text{A}}{\text{E}} \times \frac{\text{{rate}}{\text{frequency}}})} \times \frac{\text{frequency} \times \text{E}{\text{DSR}}$$

  где:
  - A = количество дней с начала периода купона до даты расчетов (накопленные дни).

- DSR = количество дней с даты урегулирования до даты активации.
- E = количество дней в период купона.

- Если существует более одного периода купона до активации, YIELD вычисляется через сотни итераций. Решение использует метод Newton, основанный на формуле, используемой для функции PRICE. Доходность изменяется до тех пор, пока оценочная цена, учитываемая доходность, близка к цене.

- расчет и зрелость усечены в целые числа.

- частота и база округляются до ближайшего целого числа.

- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - скорость < 0.
  - pr ≤ 0.
  - активация ≤ 0.
  - частота — любое число, отличное от 1, 2 или 4.
  - базовый < 0 или базовый > 4.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Данные	Description
15-февраль-08	Дата урегулирования
15 ноября-16	Дата зрелости
5.75%	Процентный купон
95.04287	Цена
100 долл. США	Значение активации
2	Частота является полунуальной (см. выше)
0	30/360 базис (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  YIELD(DATE(2008,2,15), DATE(2016,11,15), 0.0575, 95.04287, 100, 2,0)
}
```

Возвращает доходность облигации с указанными выше условиями.

Развернуть таблицу

[значение]
0.0650000068807314

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

YIELDDISC

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает годовую доходность для скидок на скидку.

Синтаксис

DAX

```
YIELDDISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

Параметры

Развернуть таблицу

Термин	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`pr`	Цена на цену безопасности на $100 лицовую стоимость.
`redemption`	Стоимость активации безопасности за $100 лиц.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический

`Basis`	**подсчета дней**
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Годовая доходность.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- расчет и зрелость усечены в целые числа.
- база округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - расчет или срок действия не является допустимой датой.
  - расчет ≥ зрелости.
  - pr ≤ 0.
  - активация ≤ 0.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Данные	-
16-февраль-08	Дата урегулирования
1-Мар-08	Дата зрелости
99.795	Цена
100 долл. США	Значение активации
2	Фактическая/360 база

Следующий запрос DAX:

```
DAX

EVALUATE
{
  YIELDDISC(DATE(2008,2,16), DATE(2008,3,1), 99.795, 100, 2)
}
```

Возвращает годовую доходность безопасности, учитывая указанные выше условия.

Развернуть таблицу

[значение]
0.0528225719868583

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

YIELDMAT

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает годовую доходность безопасности, которая платит проценты по зрелости.

Синтаксис

```
DAX

YIELDMAT(<settlement>, <maturity>, <issue>, <rate>, <pr>[, <basis>])
```

Параметры

Развернуть таблицу

Термин	Определение
`settlement`	Дата урегулирования безопасности. Дата урегулирования безопасности — это дата после даты выдачи, когда безопасность торгуется покупателем.
`maturity`	Дата окончания срока действия безопасности. Дата зрелости — это дата истечения срока действия безопасности.
`issue`	Дата проблемы безопасности.
`rate`	Процентная ставка безопасности на дату выдачи.
`pr`	Цена на цену безопасности на $100 лицовую стоимость.
`basis`	(Необязательно) Тип используемого числа дней. Если база опущена, предполагается, что значение равно 0. Допустимые значения перечислены ниже этой таблицы.

Параметр `basis` принимает следующие значения:

Развернуть таблицу

`Basis`	**подсчета дней**
0 или опущено	США (NASD) 30/360
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский 30/360

Возвращаемое значение

Годовая доходность.

Замечания

- Даты хранятся в виде последовательных серийных номеров, чтобы их можно было использовать в вычислениях. В DAX, 30 декабря 1899 г. день 0, а 1 января 2008 г. — 39448, так как 39 448 дней после 30 декабря 1899 г.
- Дата урегулирования — это дата, когда покупатель приобретает купон, например облигацию. Дата зрелости — это дата истечения срока действия купона. Например, предположим, что 30-летняя облигация выдается 1 января 2008 года и приобретается покупателем шесть месяцев спустя. Дата выдачи будет 1 января 2008 года, дата урегулирования будет 1 июля 2008 года, а дата погашения будет 1 января 2038 года, которая составляет 30 лет после 1 января 2008 года, дата выдачи.
- расчет, зрелость и проблема усечены в целые числа.
- база округляется до ближайшего целого числа.
- Если возвращается ошибка:
  - дата урегулирования, зрелости или выдачи не является допустимой датой.
  - Проблема > урегулирования >> не удовлетворена.
  - скорость < 0.
  - pr ≤ 0.
  - базовый < 0 или базовый > 4.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Данные	Description
15-Мар-08	Дата урегулирования
3-ноябрь-08	Дата зрелости
8-ноябрь-07	Дата проблемы
6,25%	Процент полуфиналового купона
100.0123	Цена
0	30/360 базис (см. выше)

Следующий запрос DAX:

```
DAX

EVALUATE
{
  YIELDMAT(DATE(2008,3,15), DATE(2008,11,3), DATE(2007,11,8), 0.0625,
100.0123, 0)
}
```

Возвращает доходность для безопасности с помощью указанных выше условий.

Развернуть таблицу

Ценность
0.0609543336915387

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Функции INFO

Статья • 01.04.2025

Выражения анализа данных (DAX) включают набор информационных функций на основе библиотеки динамических представлений управления (DMV) в службах Analysis Services, которые были изменены для работы в качестве функций DAX. INFO DAX функции выводятся в виде типа данных таблицы. В качестве таблиц их можно использовать с другими функциями DAX, такими как FILTER, SELECTCOLUMNS, ADDCOLUMNSи другие. В этом разделе описаны функции INFO, доступные на языке DAX.

Для семантических моделей Power BI, как и для DMVs, функции INFO DAX требуют прав администратора семантической модели. Для некоторых также требуются разрешения администратора рабочей области.

Как функции DAX, эти функции INFO превосходят возможности динамических административных представлений, которые используют синтаксис, подобный SQL, и возвращают набор строк. Поскольку эти новые функции являются функциями DAX, они могут использоваться как любая таблица в запросе DAX — дальше комбинируются и структурируются в запросе DAX.

DAX

```
EVALUATE
    ADDCOLUMNS(
        SELECTCOLUMNS(
            INFO.VIEW.MEASURES(),
            "Measure", [Name],
            [Description],
            "DAX Formula", [Expression],
            "State", [State]
        ),
        "Model name", "My semantic model",
        "As of date", NOW()
    )
```

Функции INFO поддерживаются в семантических моделях Power BI, но не в моделях SQL Server Analysis Services, Azure Analysis Services или PowerPivot. Функции INFO.VIEW DAX могут выполняться в вычисляемых таблицах, столбцах, мерах и DAX запросах, но другие INFO-функции DAX могут выполняться только в DAX запросах.

ИНФО.ВЬЮ DAX функции

Четыре функции INFO DAX также имеют аналогичные функции INFO.VIEW DAX. Понятные имена, такие как имя таблицы вместо идентификатора таблицы, используются и добавляются новые столбцы, чтобы упростить использование без необходимости присоединять другие таблицы функций INFO DAX. ИНФОРМАЦИЯ. Функции VIEW DAX можно использовать в вычислениях внутри семантической модели. При включении в вычисляемую таблицу они

могут сами задокументировать модель, чтобы другие люди могли использовать ее для создания отчетов или DAX запросов.

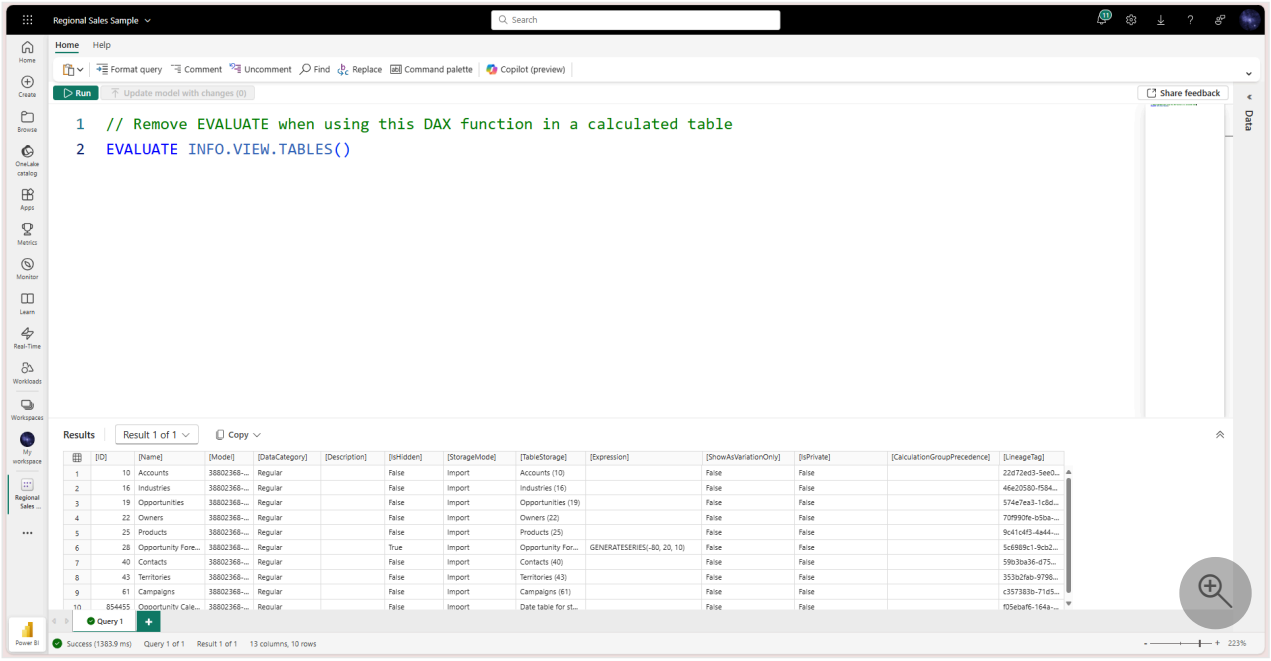
ИНФОРМАЦИЯ.ВИД.ТАБЛИЦЫ

INFO.VIEW.TABLES содержит информацию о таблицах в модели, такую как имя таблицы, описание, режим хранения и сведения о том, скрыта ли она.

DAX

```
// Remove EVALUATE when using this DAX function in a calculated table
EVALUATE INFO.VIEW.TABLES()
```

На следующем снимке экрана показан пример использования этой функции DAX в представлении запроса DAX с образцом региональных продаж , доступном из **Learn** в службе Power BI.



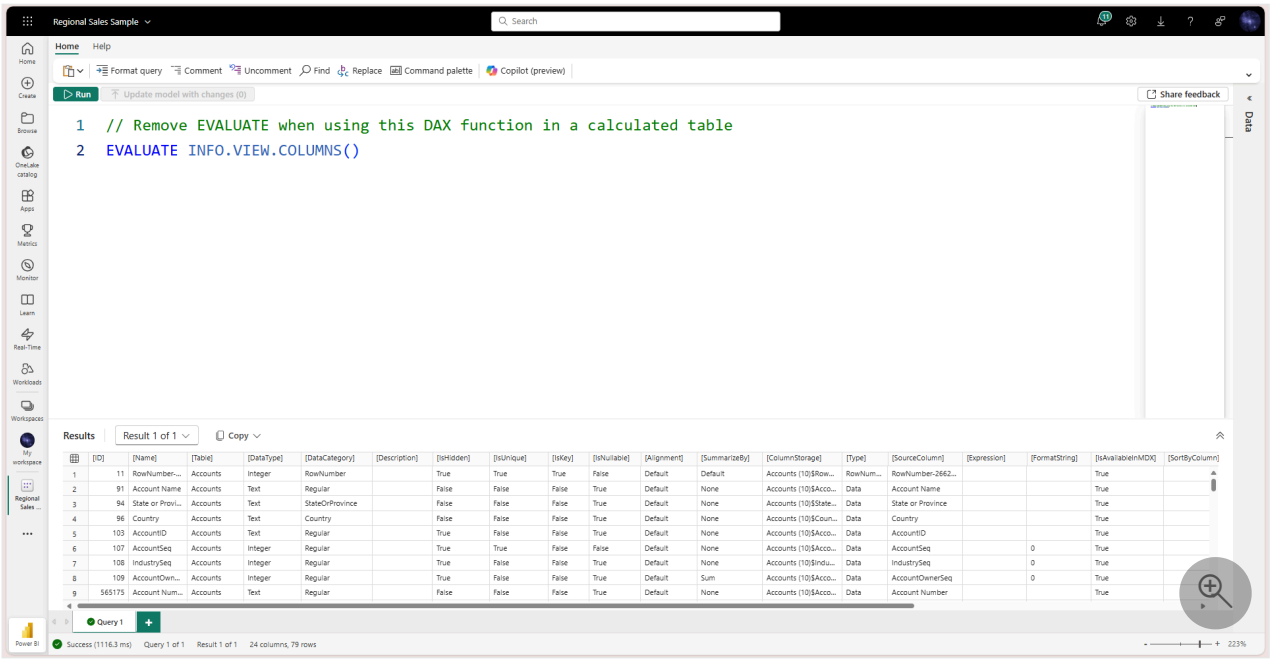
ИНФОРМАЦИЯ. ОБЗОР. СТОЛБЦЫ

INFO.VIEW.COLUMNS содержит информацию о столбцах в модели, например, имя столбца, тип данных и то, скрыт он или нет.

DAX

```
// Remove EVALUATE when using this DAX function in a calculated table
EVALUATE INFO.VIEW.COLUMNS()
```

На следующем снимке экрана показан пример использования этой функции DAX в представлении запроса DAX с образцом региональных продаж , доступном из **Learn** в службе Power BI.



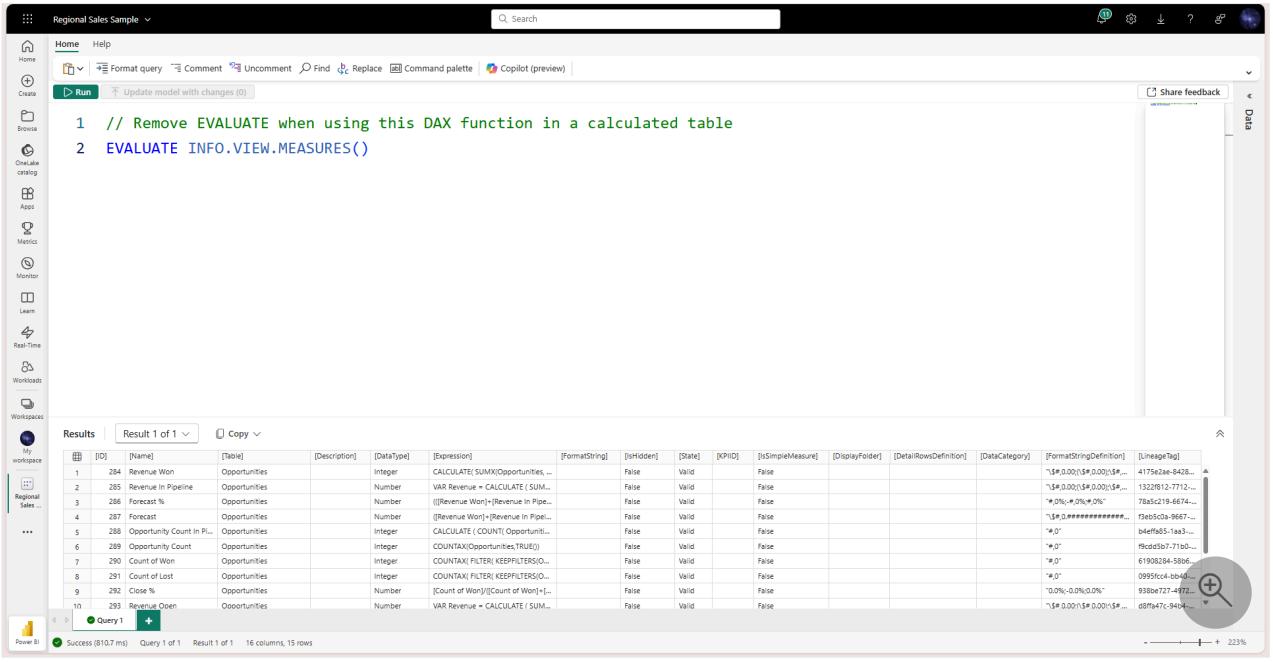
ИНФОРМАЦИЯ. ВИД. МЕРЫ

INFO.VIEW.MEASURES содержит сведения о мерах в модели, таких как название, выражение и строка форматирования.

DAX

```
// Remove EVALUATE when using this DAX function in a calculated table
EVALUATE INFO.VIEW.MEASURES()
```

На следующем снимке экрана показан пример использования этой функции DAX в представлении запроса DAX с образцом региональных продаж , доступном из **Learn** в службе Power BI.



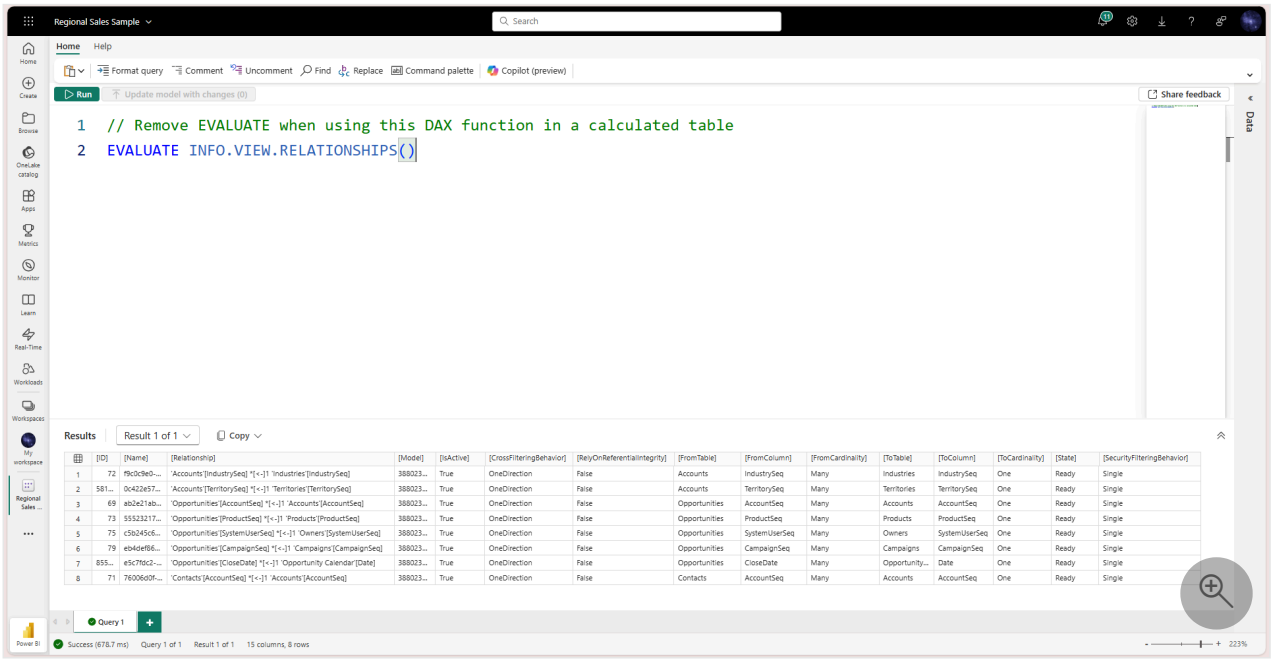
ИНФОРМАЦИЯ. ВИД. ОТНОШЕНИЯ

ИНФО.ВИД.RELATIONSHIPS содержит сведения о связях в модели, таких как откуда и куда таблицы и столбцы, кардинальность и направление перекрестного фильтра.

DAX

```
// Remove EVALUATE when using this DAX function in a calculated table
EVALUATE INFO.VIEW.RELATIONSHIPS()
```

На следующем снимке экрана показан пример использования этой функции DAX в представлении запроса DAX с образцом региональных продаж , доступном из **Learn** в службе Power BI.



Функции раздела INFO DAX

Ниже приведен список функций INFO. Некоторые работают только на определенных уровнях компата и для определенных режимов хранения или других свойств семантической модели.

Развернуть таблицу

[Функция]	[Описание]
ИНФОРМАЦИЯ. ОБЗОР. СТОЛБЦЫ	Возвращает список всех столбцов в текущей модели. Можно использовать в вычислениях, включая вычисляемые таблицы.
ИНФОРМАЦИЯ. ВИД. МЕРЫ	Возвращает список всех мер в текущей модели. Можно использовать в вычислениях, включая вычисляемые таблицы.
ИНФОРМАЦИЯ.ОБЗОР.ВЗАИМОСВЯЗИ	Возвращает список всех связей в текущей модели. Можно

[Функция]	[Описание]
	использовать в вычислениях, включая вычисляемые таблицы.
ИНФОРМАЦИЯ.ВИД.ТАБЛИЦЫ	Возвращает список всех таблиц в текущей модели. Можно использовать в вычислениях, включая вычисляемые таблицы.
ИНФОРМАЦИЯ.ОПРЕДЕЛЕНИЯАЛЬТЕРНАТИВЫ	
ИНФОРМАЦИЯ, АННОТАЦИИ	Возвращает список всех заметок в текущей модели со столбцами, соответствующими набору строк схемы для объектов заметки.
ИНФОРМАЦИЯ. ИЕРАРХИИ АТРИБУТОВ	Представляет функцию запроса TMSCHEMA_ATTRIBUTE_HIERARCHIES DMV.
ИНФОРМАЦИЯ. ИЕРАРХИЧЕСКОЕ ХРАНИЛИЩЕ АТРИБУТОВ	
ИНФОРМАЦИЯ. ЗАВИСИМОСТЬ_РАСЧЕТА	Возвращает сведения о зависимости вычисления для запроса DAX.
ИНФО.КАЛЬКУЛЯЦИОННЫЕ ГРУППЫ	
ИНФОРМАЦИЯ. ЭЛЕМЕНТЫ РАСЧЕТА	
ИНФОРМАЦИЯ. КАТАЛОГИ	Представляет функцию запроса DMV для DBSCHEMA_CATALOGS.
ИНФОРМАЦИЯ. ИЗМЕНЕННЫЕ СВОЙСТВА	Представляет функцию запроса DMV для TMSCHEMA_CHANGED_PROPERTIES.
ИНФОРМАЦИЯ.ОБЪЕКТЫХРАНЕНИЯЧАСТЕЙСТОЛБЦОВ	
INFO.COLUMNPERMISSIONS	Возвращает список всех разрешений столбцов в текущей модели с столбцами, соответствующими набору строк схемы для объектов разрешений столбцов.
ИНФО_СТОЛБЦЫ	Возвращает список всех столбцов в текущей модели с столбцами, соответствующими набору строк схемы для объектов столбцов.
ИНФОРМАЦИЯ. ХРАНИЛИЩА СТОЛБЦОВ	Возвращает список всех хранилищ столбцов в текущей модели с столбцами, соответствующими набору строк схемы для объектов хранилища столбцов.

[Функция]	[Описание]
ИНФОРМАЦИЯ. CSDLMETADATA	Возвращает сведения о метаданных базы данных в формате XML.
ИНФОРМАЦИЯ. КУЛЬТУРЫ	Возвращает список всех культур в текущей модели со столбцами, соответствующими строкам схемы для объектов культур.
ИНФОРМАЦИЯ. ОПРЕДЕЛЕНИЯПОКРЫТИЯДАННЫХ	
ИНФОРМАЦИЯ. ИСТОЧНИКИ_ДАННЫХ	Представляет функцию запроса TMSCHEMA_DATASOURCES DMV.
ИНФОРМАЦИЯ.DELTATABLEMETADATASTORAGES	
ИНФОРМАЦИЯ О ЗАВИСИМОСТЯХ	Возвращает сведения о зависимости вычисления для запроса DAX.
ИНФОРМАЦИЯ.ОПРЕДЕЛЕНИЯСТРОКПОДРОБНОСТЕЙ	Возвращает список всех определений строк подробных данных в текущей модели со столбцами, соответствующими набору строк схемы для объектов этих определений.
ИНФОРМАЦИЯ. ХРАНИЛИЩА СЛОВАРЕЙ	
ИНФОРМАЦИЯ. Исключенные артефакты	Представляет функцию TMSCHEMA_EXCLUDED_ARTIFACTS для запросов DMV.
ИНФО. ЭКСПРЕССИИ	Возвращает список всех выражений в текущей модели со столбцами, соответствующими набору строк схемы для объектов выражений.
ИНФОРМАЦИЯ. РАСШИРЕННЫЕ_СВОЙСТВА	Возвращает список всех расширенных свойств в текущей модели со столбцами, соответствующими набору строк схемы для объектов расширенных свойств.
ИНФОРМАЦИЯ. ОПРЕДЕЛЕНИЯ ФОРМАТА СТРОК	
ИНФО.ФУНКЦИИ	Возвращает сведения о функциях, которые в настоящее время доступны для использования на языке программирования DAX.
ИНФОРМАЦИЯ.ГЕНЕРАЛЬНАЯКАРТАСЕГМЕНТАМЕТАДАННЫХХРАНИЛИЩ	
ИНФО.GROUPBYCOLUMNS	

[Функция]	[Описание]
ИНФОРМАЦИЯ О ИЕРАРХИЯХ	Представляет функцию запроса TMSCHEMA_HIERARCHIES DMV.
ИНФОРМАЦИЯ. ИЕРАРХИЯ ХРАНИЛИЩ	
Инфо.КПЭ	Возвращает список всех ключевых показателей эффективности в текущей модели со столбцами, соответствующими набору строк схемы для объектов ключевого показателя эффективности.
ИНФО.УРОВНИ	Возвращает список всех уровней в текущей модели со столбцами, соответствующими набору строк схемы для объектов уровня.
ИНФОРМАЦИЯ ЛИНГВИСТИЧЕСКИЕ МЕТАДАННЫЕ	Представляет функцию запроса TMSCHEMA_LINGUISTIC_METADATA DMV.
ИНФОРМАЦИЯ.МЕРЫ	Возвращает список всех мер в текущей модели со столбцами, соответствующими набору строк схемы для объектов мер.
ИНФО.МОДЕЛЬ	Представляет функцию запроса TMSCHEMA_MODEL DMV.
ИНФОРМАЦИЯ. ПЕРЕВОДЫ ОБЪЕКТОВ	Возвращает список всех переводов объектов в текущей модели с столбцами, соответствующими набору строк схемы для объектов перевода объектов.
ИНФОРМАЦИЯ.PARQUETFILESTORAGES	
ИНФОРМАЦИЯ О РАЗДЕЛАХ	Представляет собой функцию запроса DMV для TMSCHEMA_PARTITIONS.
ИНФОРМАЦИЯ.ПАРТИЦИОННЫЕХРАНИЛИЩА	Возвращает список всех хранилищ секций в текущей модели со столбцами, соответствующими набору строк схемы для объектов хранилища секций.
ИНФОРМАЦИЯ.ПЕРСПЕКТИВНЫЕСТОЛБЦЫ	Возвращает список всех колонок перспектив в текущей модели, где столбцы соответствуют набору строк схемы для объектов столбцов перспектив.
ИНФОРМАЦИЯ. ИЕРАРХИИ ПЕРСПЕКТИВ	Возвращает список всех иерархий перспектив в текущей модели со

[Функция]	[Описание]
	столбцами, соответствующими набору строк схемы для объектов иерархий перспектив.
ИНФОРМАЦИЯ. МЕРЫ ПЕРСПЕКТИВЫ	Возвращает список всех перспективных мер в текущей модели со столбцами, соответствующими строковой схеме для объектов перспективных мер.
ИНФОРМАЦИЯ, ПЕРСПЕКТИВЫ	Возвращает список всех перспектив в текущей модели с столбцами, соответствующими набору строк схемы для объектов перспективы.
ИНФОРМАЦИЯ. ПЕРСПЕКТИВНЫЕ ТАБЛИЦЫ	Возвращает список всех таблиц перспективы в текущей модели со столбцами, соответствующими набору строк схемы для объектов перспектив.
ИНФО.СВОЙСТВА	Представляет функцию запроса DISCOVER_PROPERTIES DMV.
ИНФО. ЗАПРОСЫГРУПП	
ИНФОРМАЦИЯ. ПОЛИТИКИ ОБНОВЛЕНИЯ	
ИНФОРМАЦИЯ. ДЕТАЛИ СВЯЗАННОЙ КОЛОНКИ	
ИНФОРМАЦИЯ.ОТНОШЕНИЕ_ИНДЕКС_ХРАНИЛИЩЕ	
ИНФОРМАЦИЯ О СВЯЗЯХ	Представляет функцию запроса TMSCHEMA_RELATIONSHIPS DMV.
ИНФОРМАЦИЯ. ХРАНИЛИЩА ОТНОШЕНИЙ	
ИНФОРМАЦИЯ. ЧЛЕНСТВА В РОЛЯХ	Возвращает список всех ролевых членств в текущей модели со столбцами, соответствующими набору строк схемы для объектов ролевого членства.
ИНФОРМАЦИЯ О РОЛЯХ	Возвращает список всех ролей в текущей модели с столбцами, соответствующими набору строк схемы для объектов ролей.
ИНФОРМАЦИЯ. ХРАНИЛИЩА КАРТЫ СЕГМЕНТОВ	Возвращает список всех хранилищ карт сегментов в текущей модели с столбцами, соответствующими набору строк схемы для объектов хранилища карты сегментов.

[Функция]	[Описание]
ИНФОРМАЦИЯ. SEGMENTSTORAGES	
ИНФОРМАЦИЯ. ФАЙЛЫ_ХРАНИЛИЩА	Возвращает список всех файлов хранилища в текущей модели со столбцами, соответствующими набору строк схемы для объектов файлов хранилища.
ИНФОРМАЦИЯ. ПАПКИ_ХРАНЕНИЯ	Возвращает список всех папок хранилища в текущей модели с столбцами, соответствующими набору строк схемы для объектов папок хранилища.
ИНФОРМАЦИЯ.STORAGETABLECOLUMNS	Возвращает статистику о столбцах таблиц в памяти.
ИНФОРМАЦИЯ. СЕГМЕНТЫСТОЛБЦОВТАБЛИЦЫХРАНЕНИЯ	Возвращает сведения о сегментах столбцов, используемых для хранения данных для таблиц в памяти.
ИНФОРМАЦИЯ.СТОРАДЖТЕЙБЛС	Возвращает статистику о таблицах в памяти.
ИНФО.ТАБЛИЧНЫЕ_РАЗРЕШЕНИЯ	Возвращает список всех разрешений таблицы в текущей модели со столбцами, соответствующими набору строк схемы для объектов разрешений таблицы.
ИНФОТАБЛИЦЫ	Возвращает список всех таблиц в текущей модели со столбцами, соответствующими набору строк схемы для объектов таблиц.
ИНФОРМАЦИЯ. ХРАНИЛИЩА ТАБЛИЦ	Возвращает список всех хранилищ таблиц в текущей модели с столбцами, соответствующими набору строк схемы для объектов хранилища таблиц.
ИНФОРМАЦИЯ. ВАРИАЦИИ	Возвращает список всех вариантов в текущей модели с столбцами, соответствующими набору строк схемы для объектов вариантов.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ИНФОРМАЦИЯ. ВИД. СТОЛБЦЫ

Статья • 17.03.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу с информацией о каждом столбце в семантической модели, например имя, описание и строку форматирования. Эти сведения помогают понять модель и самостоятельно документировать модель при использовании в вычисляемых таблицах.

Синтаксис

```
DAX

INFO.VIEW.COLUMNS()
```

Возвращаемое значение

Таблица со следующими столбцами:

Развернуть таблицу

Столбец	Описание
[ИДЕНТИФИКАТОР]	Уникальный идентификатор для каждого столбца в этой семантической модели в виде целого числа.
[Имя]	Имя каждого столбца в этой семантической модели в виде строки.
[Таблица]	Таблица каждого столбца в этой семантической модели в виде строки.
[DataType]	Тип данных каждого столбца в этой семантической модели в виде строки.
[DataCategory]	Категория данных каждого столбца в этой семантической модели в виде строки.
[Описание]	Описание каждого столбца в этой семантической модели в виде строки.
[IsHidden]	Скрытое состояние каждого столбца в этой семантической модели как True или False.

Столбец	Описание
[IsUnique]	Это уникальность каждого столбца в этой семантической модели как True от False.
[IsKey]	Ключ каждого столбца в этой семантической модели как True или False.
[IsNullable]	Значение NULL для каждого столбца в этой семантической модели как True или False.
[выравнивание]	Выравнивание каждого столбца в этой семантической модели в виде строки.
[SummarizeBy]	Сводка по каждому столбцу в этой семантической модели в виде строки.
[ColumnStorage]	Хранилище столбцов каждого столбца в этой семантической модели в виде строкового сочетания имени и идентификатора.
[Тип]	Тип каждого столбца в этой семантической модели в виде строки.
[SourceColumn]	Исходный столбец каждого столбца в этой семантической модели в виде строки.
[Выражение]	Формула DAX вычисляемых столбцов.
[FormatString]	Строка формата каждого столбца в этой семантической модели в виде строки.
[IsAvailableInMDX]	Он доступен в многомерных выражениях каждого столбца в этой семантической модели как True или False. Анализ в таблицах сводной таблицы Excel будет отображать только столбцы, равные True.
[SortByColumn]	Сортировка по столбцу каждого столбца в этой семантической модели в виде строки. Отображается как пустое при сортировке по себе.
[GroupingBehavior]	Поведение группировки каждого столбца в этой семантической модели в виде строки.
[SourceProviderType]	Тип поставщика источника каждого столбца в этой семантической модели в виде строки.
[DisplayFolder]	Папка отображения каждого столбца в этой семантической модели в виде строки. Вложенные папки, отображаемые с /и несколькими папками, разделенными ;.
[AlternateOf]	Альтернативное свойство каждого столбца в этой семантической модели в виде строки.

Столбец	Описание
[LineageTag]	Тег происхождения каждого столбца в этой семантической модели в виде строки.

Замечания

Пользователи могут запускаться только с разрешением на запись в семантической модели, а не при динамическом подключении к семантической модели в Power BI Desktop. Эта функция может использоваться в вычисляемых таблицах, столбцах и мерах семантической модели и будет обновляться при обновлении модели.

Пример 1. запрос DAX

Следующий DAX запрос можно запустить в представлении запросов DAX:

```
DAX

EVALUATE
    INFO.VIEW.COLUMNS()
```

Этот запрос DAX возвращает таблицу со всеми столбцами этой функции DAX.

Пример 2. запрос DAX с SELECTCOLUMNS и FILTER

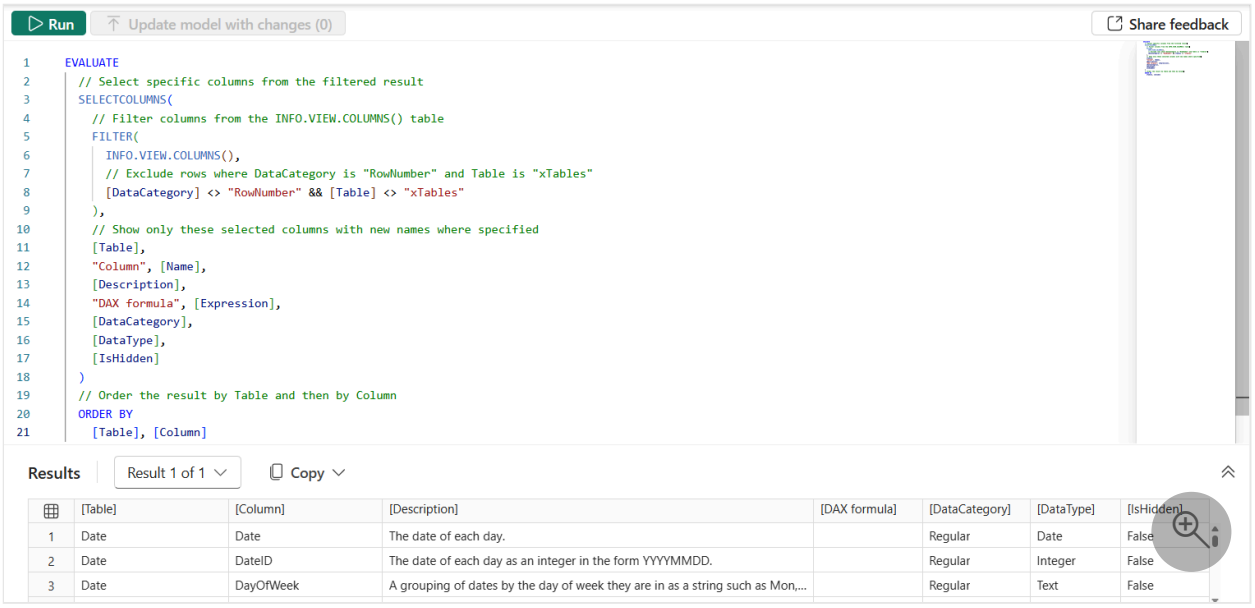
Следующий DAX запрос можно запустить в представлении запросов DAX:

```
DAX

EVALUATE
  // Select specific columns from the filtered result
  SELECTCOLUMNS(
    // Filter columns from the INFO.VIEW.COLUMNS() table
    FILTER(
      INFO.VIEW.COLUMNS(),
      // Exclude rows where DataCategory is "RowNumber" and Table is
"xTables"
      [DataCategory] <> "RowNumber" && [Table] <> "xTables"
    ),
    // Show only these selected columns with new names where specified
    [Table],
    "Column", [Name],
    [Description],
```

```
    "DAX formula", [Expression],
    [DataCategory],
    [DataType],
    [IsHidden]
  )
  // Order the result by Table and then by Column
  ORDER BY
    [Table], [Column]
```

Этот запрос DAX возвращает таблицу только с указанными столбцами и строками, отвечающими условию фильтра с формулой DAX.



Пример 3. Вычисляемая таблица с SELECTCOLUMNS и FILTER

Любой из приведенных выше примеров работает в вычисляемой таблице при удалении ключевых слов EVALUATE и ORDER BY и добавлении имени таблицы. Ниже приведен пример 2 в вычисляемой таблице:

DAX

```
Columns in this semantic model =
// Select specific columns from the filtered result
  SELECTCOLUMNS(
    // Filter columns from the INFO.VIEW.COLUMNS() table
    FILTER(
      INFO.VIEW.COLUMNS(),
      // Exclude rows where DataCategory is "RowNumber" and Table is
"xTables"
      [DataCategory] <> "RowNumber" && [Table] <> "xTables"
    ),
    // Show only these selected columns with new names where specified
```

```
    [Table],
    "Column", [Name],
    [Description],
    "DAX formula", [Expression],
    [DataCategory],
    [DataType],
    [IsHidden]
  )
```

Эта вычисляемая таблица отображает те же сведения, что и запрос DAX в примере 2 в таблице в самой модели.

Пример 4. Мера

Для подсчета количества текстовых столбцов в семантической модели можно добавить следующую меру:

DAX

```
Number of text columns =
COUNTROWS(
    FILTER(
        INFO.VIEW.COLUMNS(),
            [DataType] = "Text"
    )
)
```

Это будет отображать скалярное значение с количеством текстовых столбцов в моей модели.

Это можно показать в визуальном элементе:

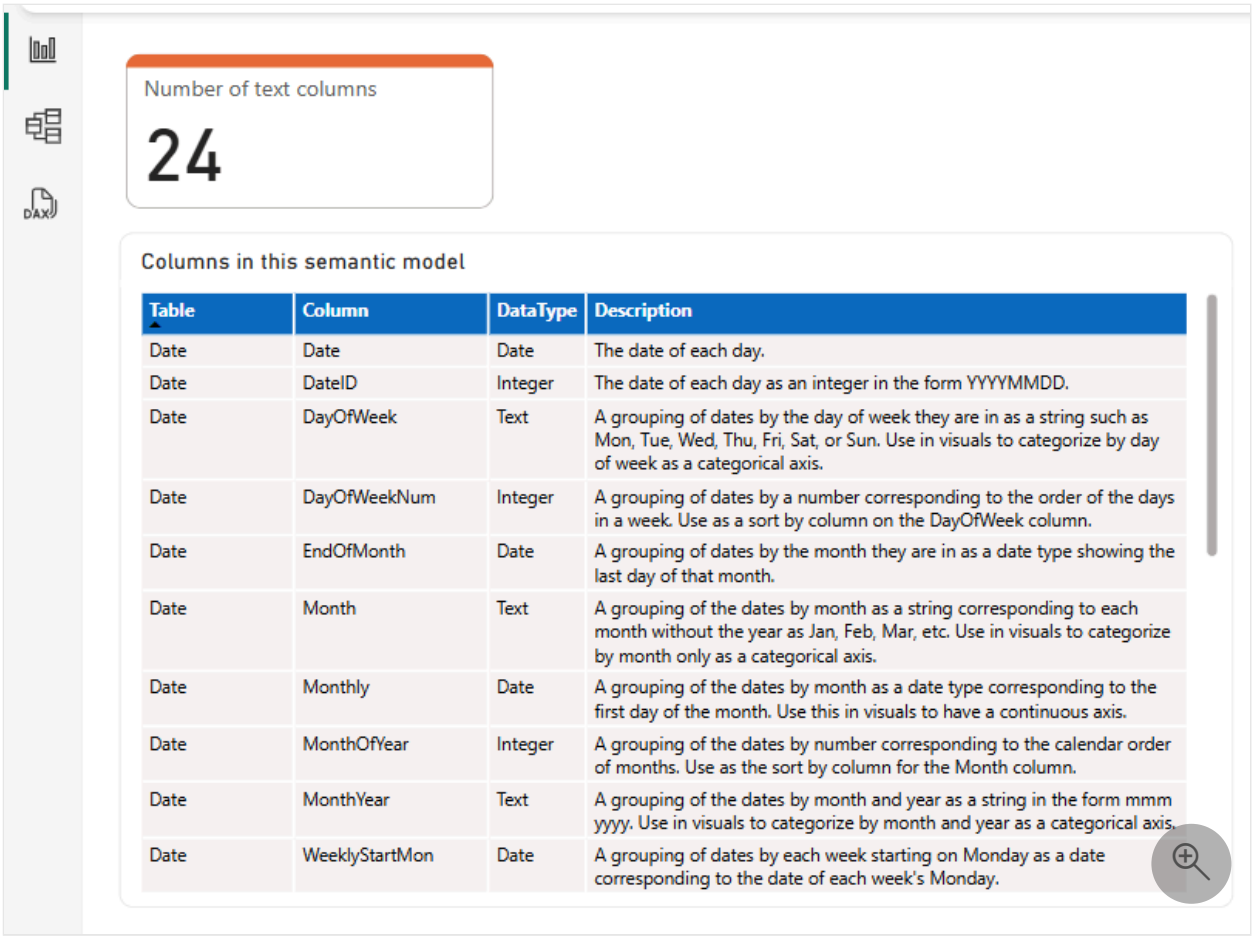


Table	Column	DataType	Description
Date	Date	Date	The date of each day.
Date	DateID	Integer	The date of each day as an integer in the form YYYYMMDD.
Date	DayOfWeek	Text	A grouping of dates by the day of week they are in as a string such as Mon, Tue, Wed, Thu, Fri, Sat, or Sun. Use in visuals to categorize by day of week as a categorical axis.
Date	DayOfWeekNum	Integer	A grouping of dates by a number corresponding to the order of the days in a week. Use as a sort by column on the DayOfWeek column.
Date	EndOfMonth	Date	A grouping of dates by the month they are in as a date type showing the last day of that month.
Date	Month	Text	A grouping of the dates by month as a string corresponding to each month without the year as Jan, Feb, Mar, etc. Use in visuals to categorize by month only as a categorical axis.
Date	Monthly	Date	A grouping of the dates by month as a date type corresponding to the first day of the month. Use this in visuals to have a continuous axis.
Date	MonthOfYear	Integer	A grouping of the dates by number corresponding to the calendar order of months. Use as the sort by column for the Month column.
Date	MonthYear	Text	A grouping of the dates by month and year as a string in the form mmm yyyy. Use in visuals to categorize by month and year as a categorical axis.
Date	WeeklyStartMon	Date	A grouping of dates by each week starting on Monday as a date corresponding to the date of each week's Monday.

Или запрос DAX в представлении запросов DAX:

DAX

```
DEFINE
    MEASURE 'Columns in this semantic model'[Number of text columns] =
    COUNTROWS(
        FILTER(
            INFO.VIEW.COLUMNS(),
                [DataType] = "Text"
        )
    )

EVALUATE
    SUMMARIZECOLUMNS(
        "Number of text columns", [Number of text columns]
    )
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ИНФОРМАЦИЯ. ВИД. СИСТЕМА ОДНОРОДНЫХ ПЛАСТОВ

Статья • 17.03.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу с информацией о каждой мере в семантической модели, например имя, описание и формулу DAX. Эти сведения помогают понять модель и самостоятельно документировать модель при использовании в вычисляемых таблицах.

Синтаксис

```
DAX

INFO.VIEW.MEASURES()
```

Возвращаемое значение

Таблица со следующими столбцами:

Развернуть таблицу

Столбец	Описание
[ИДЕНТИФИКАТОР]	Уникальный идентификатор для каждой меры в этой семантической модели в виде целого числа.
[Имя]	Имя каждой меры в этой семантической модели в виде строки.
[Таблица]	Домашняя таблица каждой меры в этой семантической модели в виде строки.
[Описание]	Описание каждой меры в этой семантической модели в виде строки.
[DataType]	Тип данных каждой меры в этой семантической модели в виде строки. Меры обычно являются вариантным типом данных.
[Выражение]	Формула DAX каждой меры в этой семантической модели.

Столбец	Описание
[FormatString]	Строка формата каждой меры в этой семантической модели в виде строки.
[IsHidden]	Скрытое состояние каждой меры в этой семантической модели как True или False.
[Состояние]	Состояние (например, допустимое или ошибка) каждой меры в этой семантической модели в виде строки.
[KPIID]	Идентификатор ключевого показателя эффективности каждой меры в этой семантической модели в виде целого числа.
[IsSimpleMeasure]	Простой флаг меры каждой меры в этой семантической модели как True от False.
[DisplayFolder]	Папка отображения каждой меры в этой семантической модели в виде строки. Вложенные папки, отображаемые с /и несколькими папками, разделенными ;.
[DetailRowsDefinition]	Определение строк сведений для каждой меры в этой семантической модели.
[DataCategory]	Категория данных каждой меры в этой семантической модели в виде строки.
[FormatStringDefinition]	Строка динамического формата каждой меры в этой семантической модели.
[LineageTag]	Тег происхождения каждой меры в этой семантической модели в виде строки.

Замечания

Пользователи могут запускаться только с разрешением на запись в семантической модели, а не при динамическом подключении к семантической модели в Power BI Desktop. Эта функция может использоваться в вычисляемых таблицах, столбцах и мерах семантической модели и будет обновляться при обновлении модели.

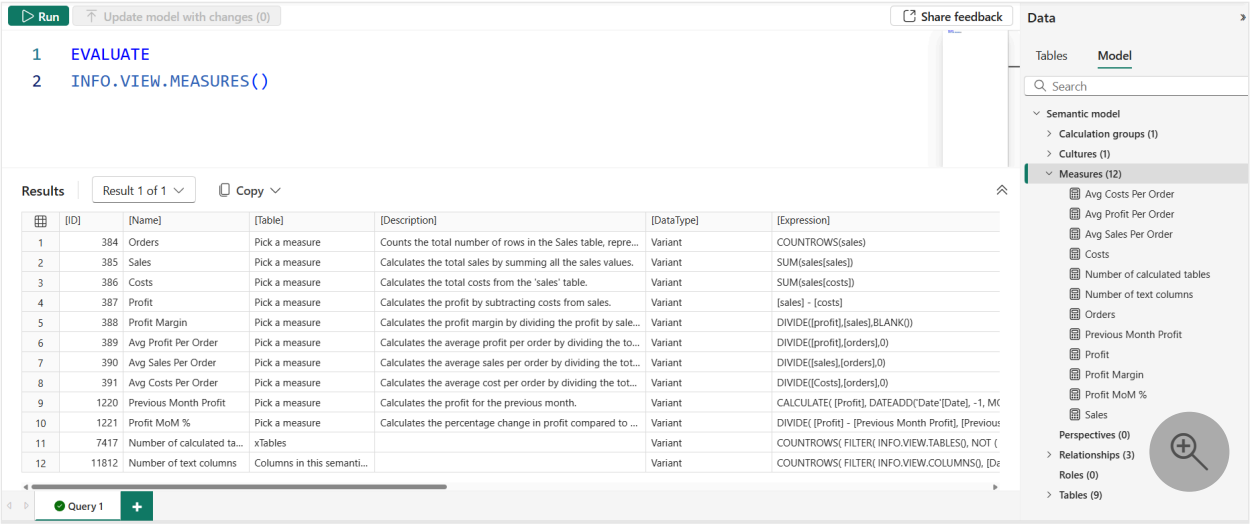
Пример 1. запрос DAX

Следующий DAX запрос можно запустить в представлении запросов DAX:

```
DAX
```

```
EVALUATE
    INFO.VIEW.MEASURES()
```

Этот запрос DAX возвращает таблицу со всеми столбцами этой функции DAX.



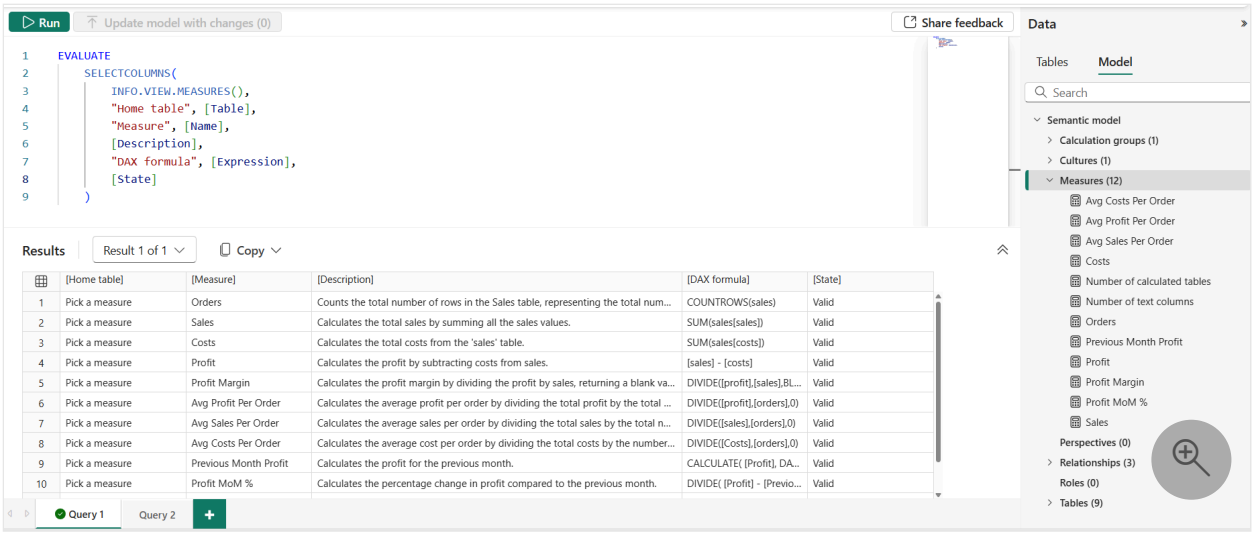
Пример 2. запрос DAX с SELECTCOLUMNS

Следующий DAX запрос можно запустить в представлении запросов DAX:

DAX

```
EVALUATE
    SELECTCOLUMNS(
        INFO.VIEW.MEASURES(),
        "Home table", [Table],
        "Measure", [Name],
        [Description],
        "DAX formula", [Expression],
        [State]
    )
```

Этот запрос DAX возвращает таблицу только с указанными столбцами.



Пример 3. Вычисляемая таблица с SELECTCOLUMNS

Любой из приведенных выше примеров работает в вычисляемой таблице при удалении ключевого слова EVALUATE и добавлении имени таблицы. Ниже приведен пример 2 в вычисляемой таблице:

DAX

```
Measures in this semantic model =
SELECTCOLUMNS(
        INFO.VIEW.MEASURES(),
        "Home table", [Table],
        "Measure", [Name],
        [Description],
        "DAX formula", [Expression],
        [State]
    )
```

Эта вычисляемая таблица отображает те же сведения, что и запрос DAX в примере 2 в таблице в самой модели.

Пример 4. Мера

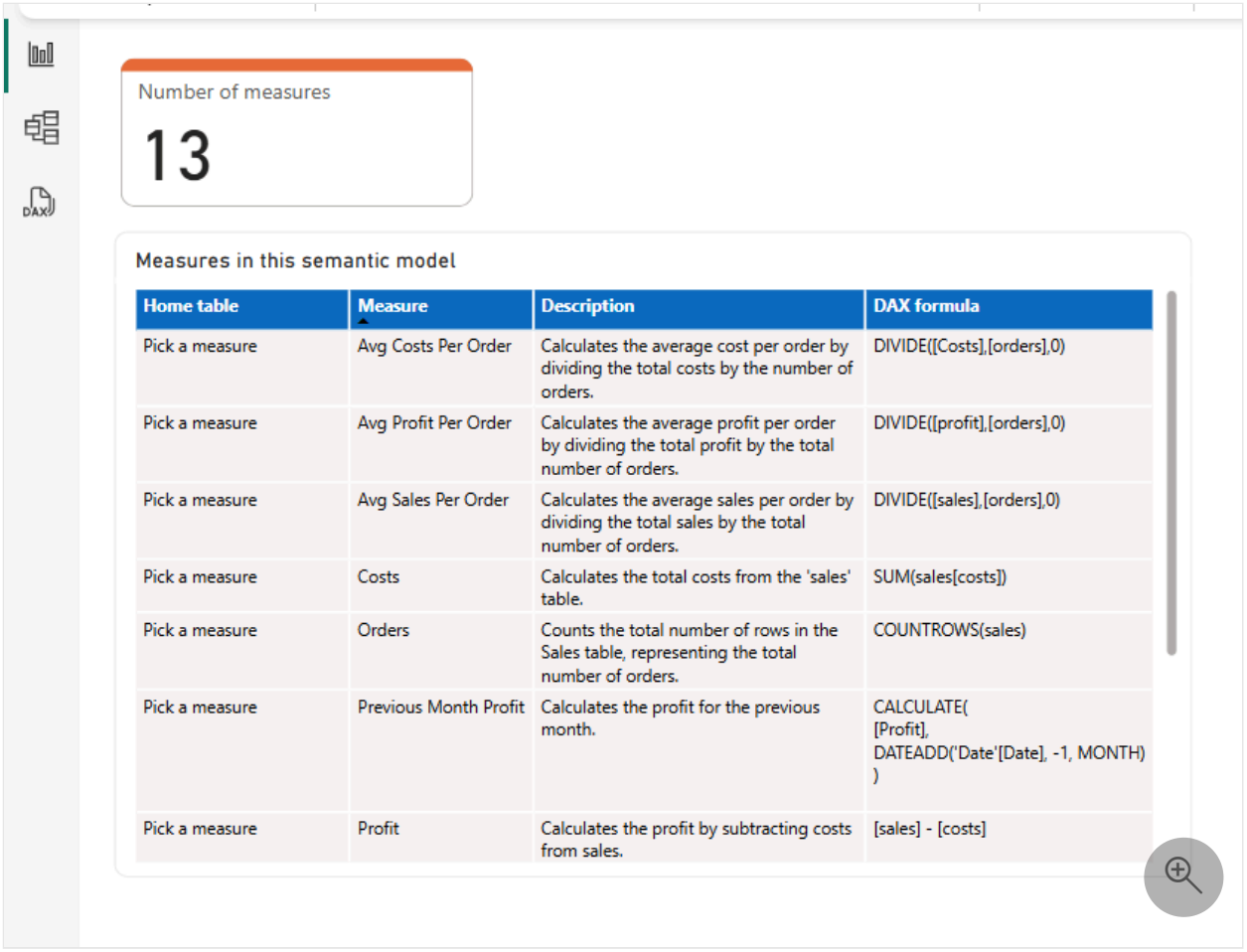
Для подсчета количества текстовых столбцов в семантической модели можно добавить следующую меру:

DAX

```
Number of measures =
COUNTROWS( INFO.VIEW.MEASURES() )
```

Это будет отображать скалярное значение с количеством мер в моей модели.

Это можно показать в визуальном элементе:



Home table	Measure	Description	DAX formula
Pick a measure	Avg Costs Per Order	Calculates the average cost per order by dividing the total costs by the number of orders.	DIVIDE([Costs],[orders],0)
Pick a measure	Avg Profit Per Order	Calculates the average profit per order by dividing the total profit by the total number of orders.	DIVIDE([profit],[orders],0)
Pick a measure	Avg Sales Per Order	Calculates the average sales per order by dividing the total sales by the total number of orders.	DIVIDE([sales],[orders],0)
Pick a measure	Costs	Calculates the total costs from the 'sales' table.	SUM(sales[costs])
Pick a measure	Orders	Counts the total number of rows in the Sales table, representing the total number of orders.	COUNTROWS(sales)
Pick a measure	Previous Month Profit	Calculates the profit for the previous month.	CALCULATE([Profit], DATEADD('Date'[Date], -1, MONTH))
Pick a measure	Profit	Calculates the profit by subtracting costs from sales.	[sales] - [costs]

Или запрос DAX в представлении запросов DAX:

DAX

```
DEFINE
    MEASURE 'Measures in this semantic model'[Number of measures] =
COUNTROWS( INFO.VIEW.MEASURES() )

EVALUATE
    SUMMARIZECOLUMNS(
        "Number of measures", [Number of measures]
    )
```

Run | Update model with changes (0)

```
DEFINE
    MEASURE 'Measures in this semantic model'[Number of measures] = COUNTROWS( INFO.VIEW.MEASURES() )

EVALUATE
    SUMMARIZECOLUMNS(
        "Number of measures", [Number of measures]
    )
```

Results | Result 1 of 1 | Copy

	[Number of measures]
1	13



Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ИНФОРМАЦИЯ. ВИД. ОТНОШЕНИЯ

Статья • 17.03.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу с информацией о каждой связи в семантической модели, например имя, кратность и поведение перекрестной фильтрации. Эти сведения помогают понять модель и самостоятельно документировать модель при использовании в вычисляемых таблицах.

Синтаксис

```
DAX

INFO.VIEW.RELATIONSHIPS()
```

Возвращаемое значение

Таблица со следующими столбцами:

Развернуть таблицу

Столбец	Описание
[ИДЕНТИФИКАТОР]	Уникальный идентификатор для каждой связи в этой семантической модели в качестве целого числа.
[Имя]	Имя каждой связи в этой семантической модели в виде строки. Может быть GUID.
[Связь]	Описательное имя каждой связи в этой семантической модели в виде строки. Включает из таблицы и столбца в таблицу и столбец с кратностью и перекрестным направлением фильтра.
[Модель]	Идентификатор семантической модели связи, как правило, GUID.
[IsActive]	Активное свойство каждой связи в этой семантической модели как True или False.
[CrossFilteringBehavior]	Перекрестное поведение или направление каждой связи в этой семантической модели в виде строки.

Столбец	Описание
[RelyOnReferentialIntegrity]	В качестве строки используется свойство ссылочной целостности каждой связи в этой семантической модели. Также называется предполагается, что целостности ссылок в редакторе связей, так как предполагается, что все строки в столбце во многих таблицах соответствуют строке в одной боковой таблице.
[FromTable]	Имя таблицы каждой связи в этой семантической модели в виде строки.
[FromColumn]	Имя столбца каждой связи в этой семантической модели в виде строки.
[FromCardinality]	Кратность столбцов каждой связи в этой семантической модели в виде строки.
[ToTable]	Имя таблицы каждой связи в этой семантической модели в виде строки.
[ToColumn]	Имя столбца каждой связи в этой семантической модели в виде строки.
[ToCardinality]	Кратность столбцов каждой связи в этой семантической модели в виде строки.
[Состояние]	Состояние каждой связи в этой семантической модели в виде строки.
[SecurityFilteringBehavior]	Поведение фильтрации безопасности для каждой связи в этой семантической модели в виде строки. Важное значение для ролей безопасности на уровне строк.

Замечания

Пользователи могут запускаться только с разрешением на запись в семантической модели, а не при динамическом подключении к семантической модели в Power BI Desktop. Эта функция может использоваться в вычисляемых таблицах, столбцах и мерах семантической модели и будет обновляться при обновлении модели.

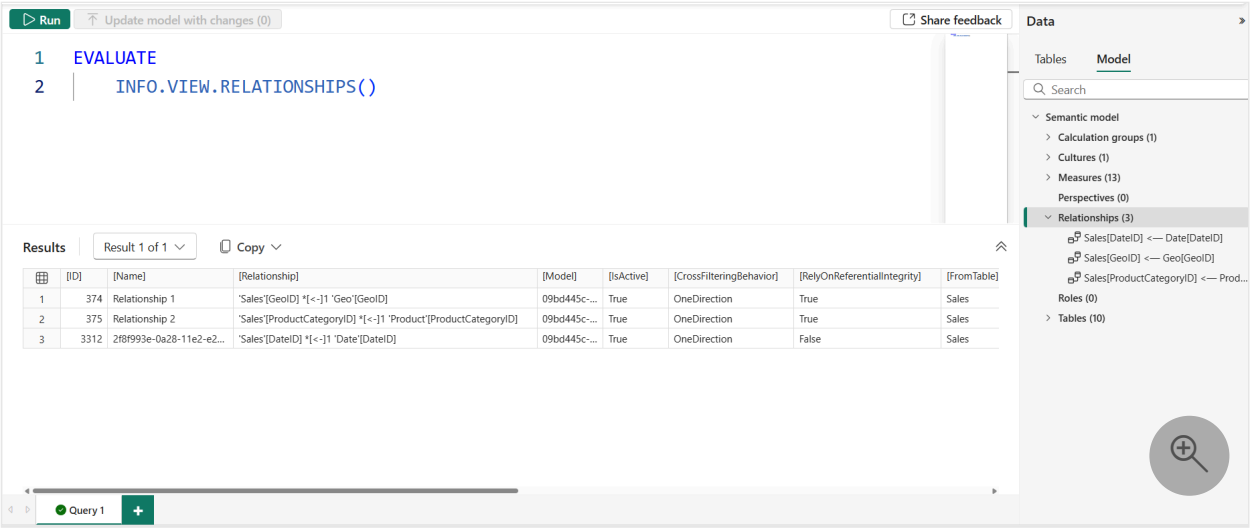
Пример 1. запрос DAX

Следующий DAX запрос можно запустить в представлении запросов DAX:

```
DAX
```

```
EVALUATE
    INFO.VIEW.RELATIONSHIPS()
```

Этот запрос DAX возвращает таблицу со всеми столбцами этой функции DAX.



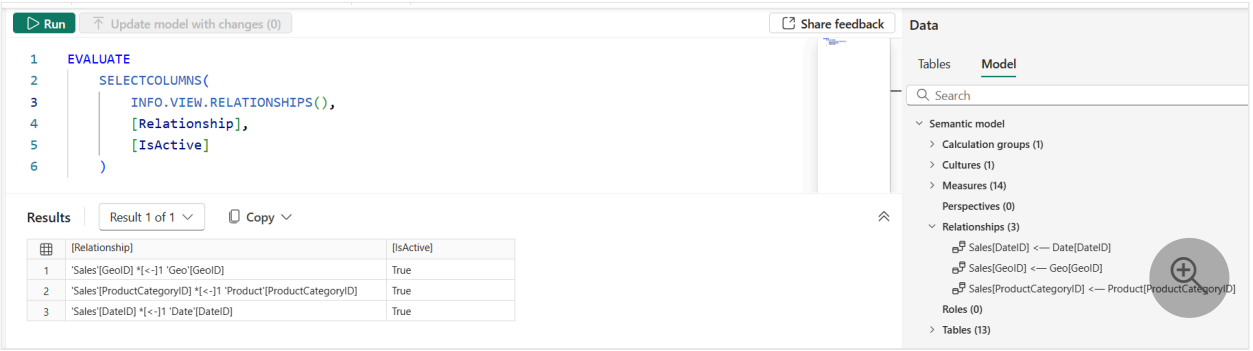
Пример 2. запрос DAX с SELECTCOLUMNS

Следующий DAX запрос можно запустить в представлении запросов DAX:

DAX

```
EVALUATE
    SELECTCOLUMNS(
        INFO.VIEW.RELATIONSHIPS(),
        [Relationship],
        [IsActive]
    )
```

Этот запрос DAX возвращает таблицу только с указанными столбцами.



Пример 3. Вычисляемая таблица с SELECTCOLUMNS

Любой из приведенных выше примеров работает в вычисляемой таблице при удалении ключевых слов EVALUATE и ORDER BY и добавлении имени таблицы. Ниже приведен пример 2 в вычисляемой таблице:

```
DAX

Relationships in this semantic model =
SELECTCOLUMNS(
    INFO.VIEW.RELATIONSHIPS(),
    [Relationship],
    [IsActive]
)
```

Эта вычисляемая таблица отображает те же сведения, что и запрос DAX в примере 2 в таблице в самой модели.

Пример 4. Мера

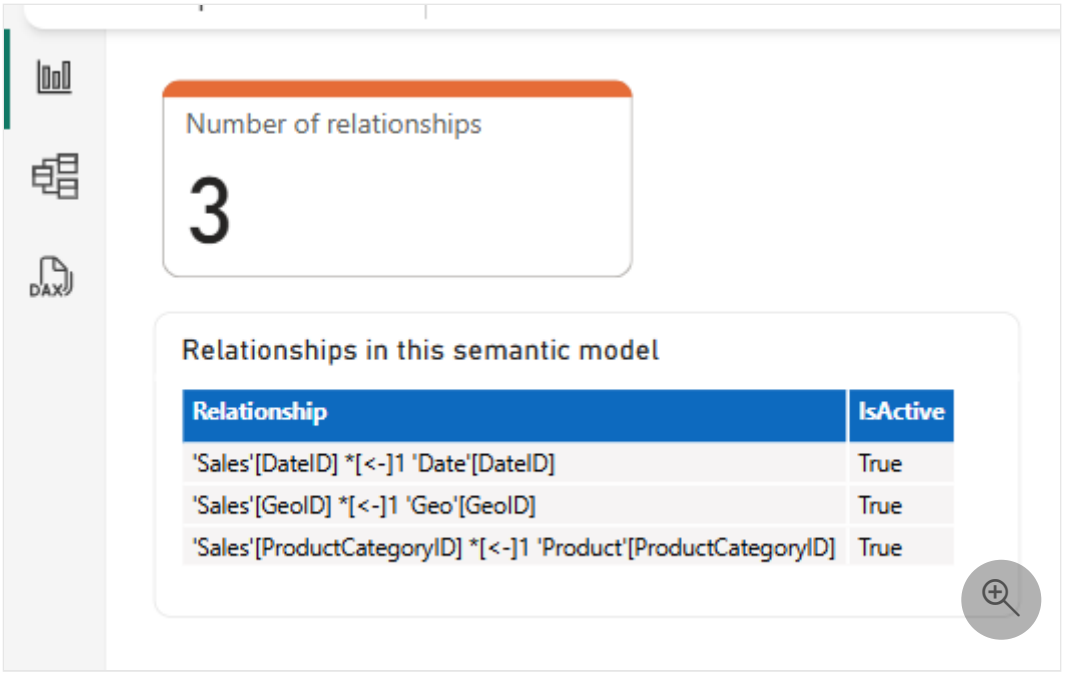
Для подсчета количества связей в семантической модели можно добавить следующую меру:

```
DAX

Number of relationships =
COUNTROWS( INFO.VIEW.RELATIONSHIPS() )
```

Это будет отображать скалярное значение с количеством связей в моей модели.

Это можно показать в визуальном элементе:

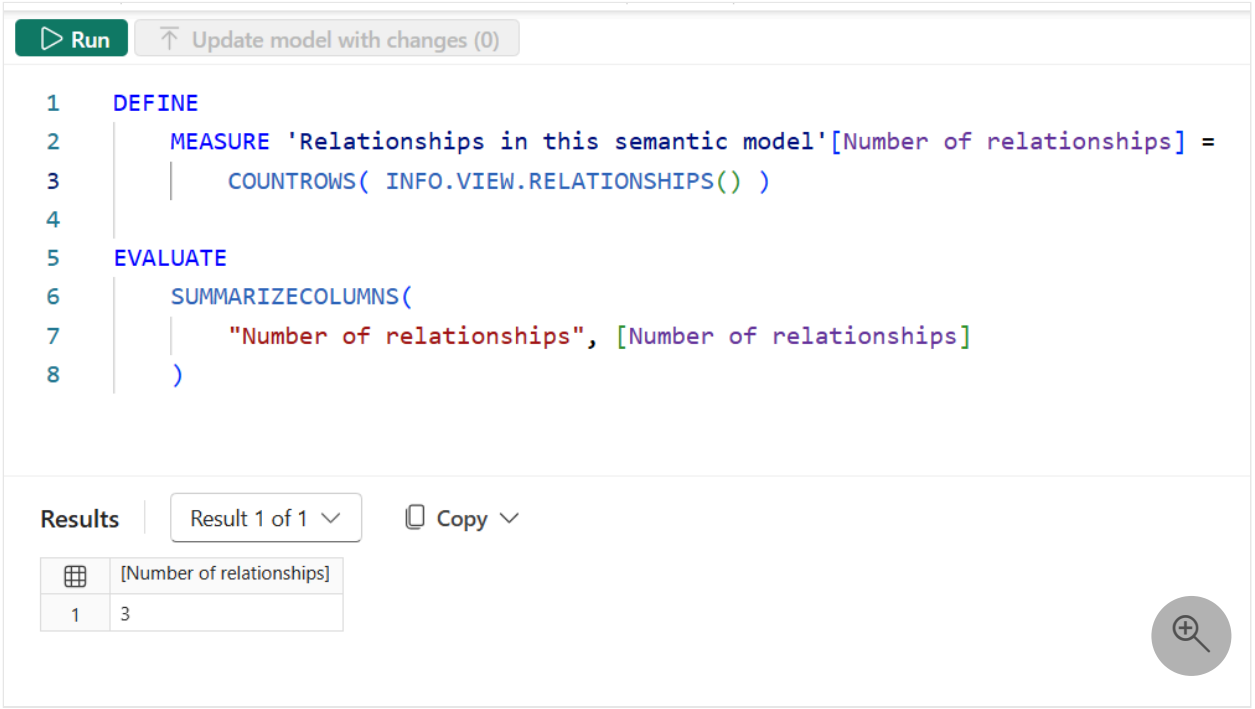


Или запрос DAX в представлении запросов DAX:

DAX

```
DEFINE
    MEASURE 'Measures in this semantic model'[Number of measures] =
COUNTROWS( INFO.VIEW.MEASURES() )

EVALUATE
    SUMMARIZECOLUMNS(
        "Number of measures", [Number of measures]
    )
```



Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ИНФОРМАЦИЯ. ВИД. ТАБЛИЦЫ

Статья • 17.03.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу с информацией о каждой таблице в семантической модели, например имя таблицы, описание и режим хранения. Эти сведения помогают понять модель и самостоятельно документировать модель при использовании в вычисляемых таблицах.

Синтаксис

```
DAX

INFO.VIEW.TABLES()
```

Возвращаемое значение

Таблица со следующими столбцами:

Развернуть таблицу

Столбец	Описание
[ИДЕНТИФИКАТОР]	Уникальный идентификатор каждой таблицы в этой семантической модели в виде целого числа.
[Имя]	Имя каждой таблицы в этой семантической модели в виде строки.
[Модель]	Идентификатор семантической модели таблицы в качестве GUID.
[DataCategory]	Категория данных каждой таблицы в этой семантической модели в виде строки.
[Описание]	Описание каждой таблицы в этой семантической модели в виде строки.
[IsHidden]	Скрытое состояние каждой таблицы в этой семантической модели как TRUE или FALSE.

Столбец	Описание
[StorageMode]	Режим хранения каждой таблицы в этой семантической модели в виде строки.
[TableStorage]	Имя и уникальный идентификатор каждой таблицы в этой семантической модели в виде строки.
[Выражение]	Формула DAX каждой таблицы в этой семантической модели в виде строки. Применяется только к вычисляемым таблицам.
[ShowAsVariationOnly]	Отображение в качестве вариантов только состояния каждой таблицы в этой семантической модели как TRUE или FALSE.
[IsPrivate]	Частное состояние каждой таблицы в этой семантической модели как TRUE или FALSE.
[CalculationGroupPrecedence]	Приоритет группы вычислений каждой таблицы в этой семантической модели в качестве целого числа. Применяется только к группам вычислений.
[LineageTag]	Тег происхождения каждой таблицы в этой семантической модели в виде строки.

Замечания

Пользователи могут запускаться только с разрешением на запись в семантической модели, а не при динамическом подключении к семантической модели в Power BI Desktop. Эта функция может использоваться в вычисляемых таблицах, столбцах и мерах семантической модели и будет обновляться при обновлении модели.

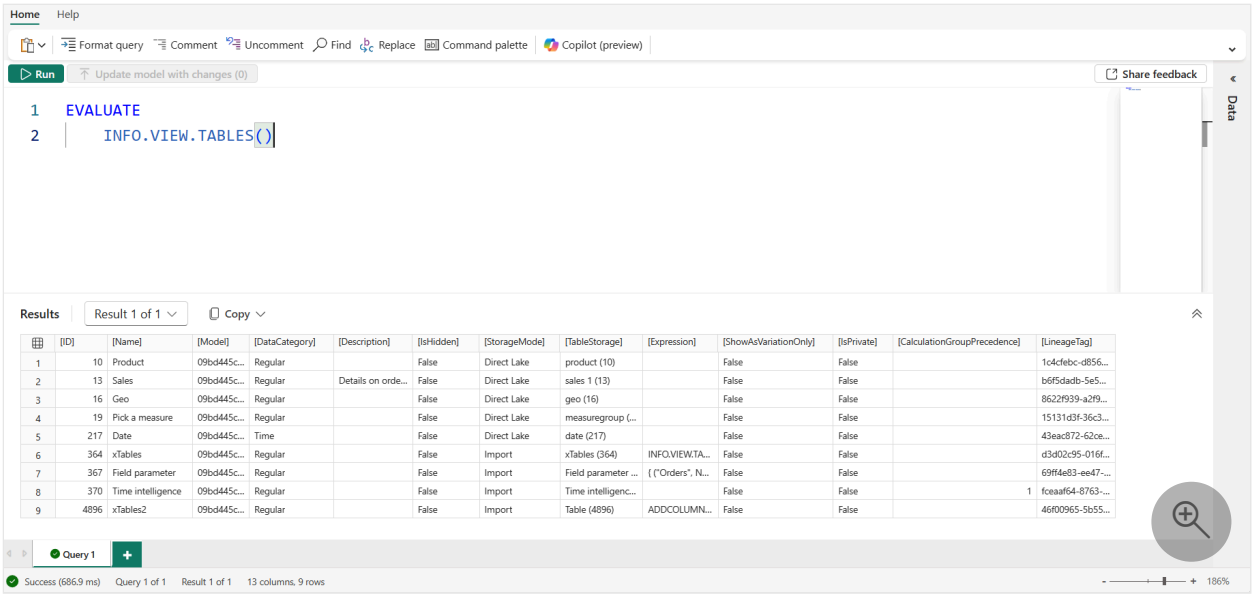
Пример 1. запрос DAX

Следующий DAX запрос можно запустить в представлении запросов DAX:

DAX

```
EVALUATE
    INFO.VIEW.TABLES()
```

Этот запрос DAX возвращает таблицу со всеми столбцами этой функции DAX.



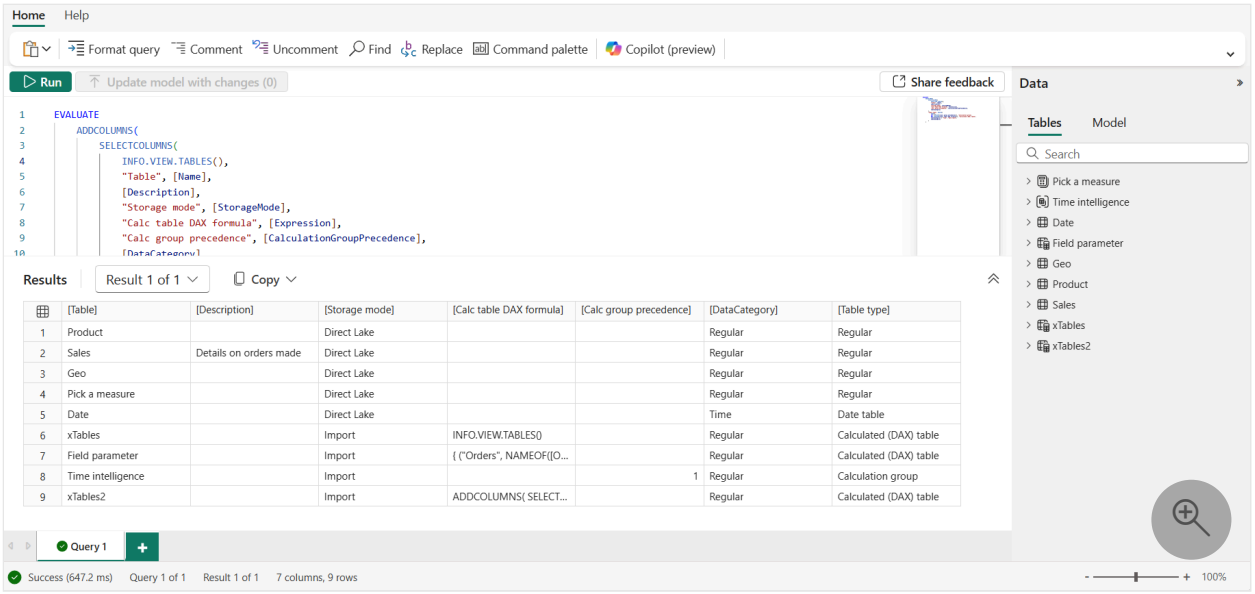
Пример 2. запрос DAX с SELECTCOLUMNS и ADDCOLUMNS

Следующий DAX запрос можно запустить в представлении запросов DAX:

DAX

```
EVALUATE
    ADDCOLUMNS(
        SELECTCOLUMNS(
            INFO.VIEW.TABLES(),
            "Table", [Name],
            [Description],
            "Storage mode", [StorageMode],
            "Calc table DAX formula", [Expression],
            "Calc group precedence", [CalculationGroupPrecedence],
            [DataCategory]
        ),
        "Table type", SWITCH(
            TRUE(),
            NOT (ISBLANK([Calc group precedence])), "Calculation group",
            NOT (ISBLANK([Calc table DAX formula])), "Calculated (DAX)
table",
            [DataCategory] = "Time", "Date table",
            [DataCategory]
        )
    )
```

Этот запрос DAX возвращает таблицу только с указанными столбцами и новым столбцом для классификации таблиц с помощью формулы DAX.



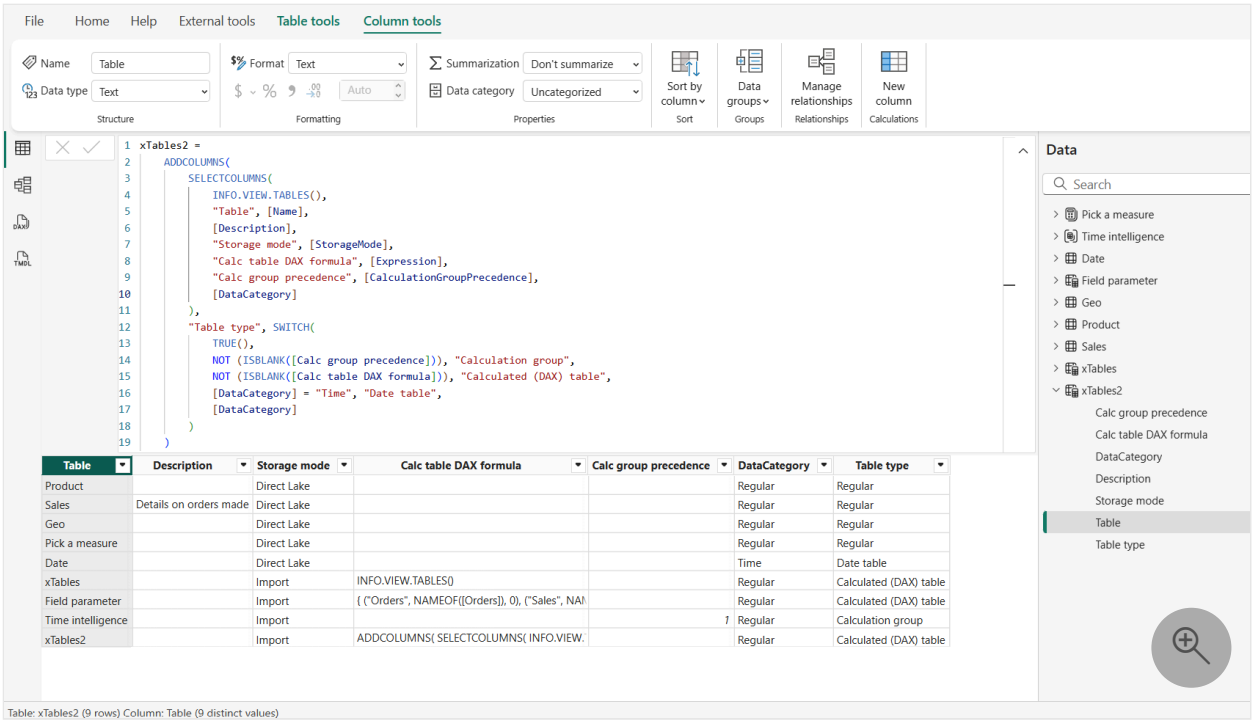
Пример 3. Вычисляемая таблица с SELECTCOLUMNS и ADDCOLUMNS

Любой из приведенных выше примеров работает в вычисляемой таблице при удалении ключевого слова EVALUATE и добавлении имени таблицы. Ниже приведен пример 2 в вычисляемой таблице:

DAX

```
xTables2 =
    ADDCOLUMNS(
        SELECTCOLUMNS(
            INFO.VIEW.TABLES(),
            "Table", [Name],
            [Description],
            "Storage mode", [StorageMode],
            "Calc table DAX formula", [Expression],
            "Calc group precedence", [CalculationGroupPrecedence],
            [DataCategory]
        ),
        "Table type", SWITCH(
            TRUE(),
            NOT (ISBLANK([Calc group precedence])), "Calculation group",
            NOT (ISBLANK([Calc table DAX formula])), "Calculated (DAX)
table",
            [DataCategory] = "Time", "Date table",
            [DataCategory]
        )
    )
```

Эта вычисляемая таблица отображает те же сведения, что и запрос DAX в примере 2 в таблице в самой модели.



Пример 4. Мера

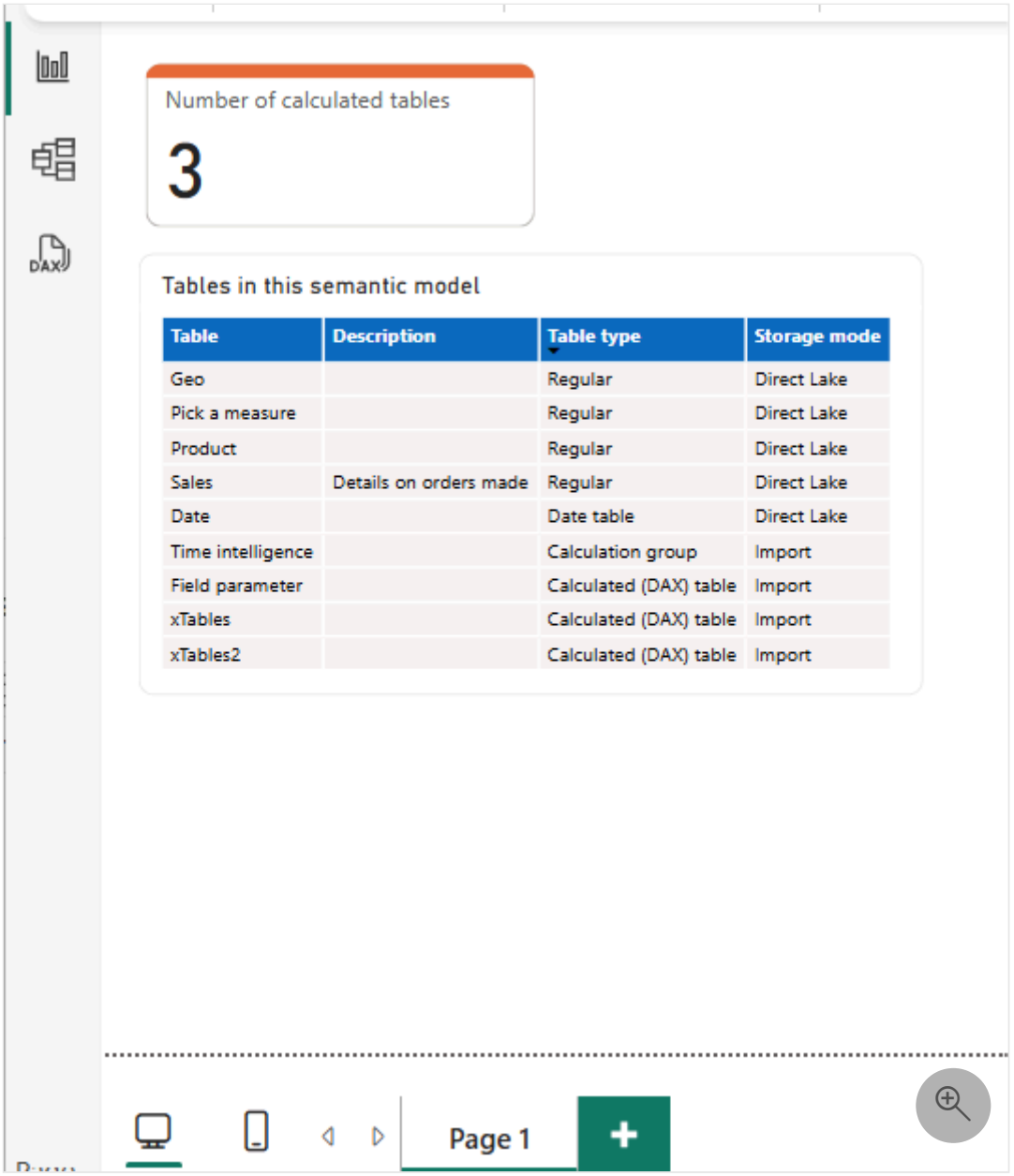
Для подсчета количества вычисляемых столбцов в семантической модели можно добавить следующую меру:

DAX

```
Number of calculated tables =
COUNTROWS(
    FILTER(
        INFO.VIEW.TABLES(),
            NOT ( ISBLANK( [Expression] ) )
    )
)
```

Это будет отображать скалярное значение с количеством вычисляемых таблиц в моей модели.

Это можно показать в визуальном элементе:

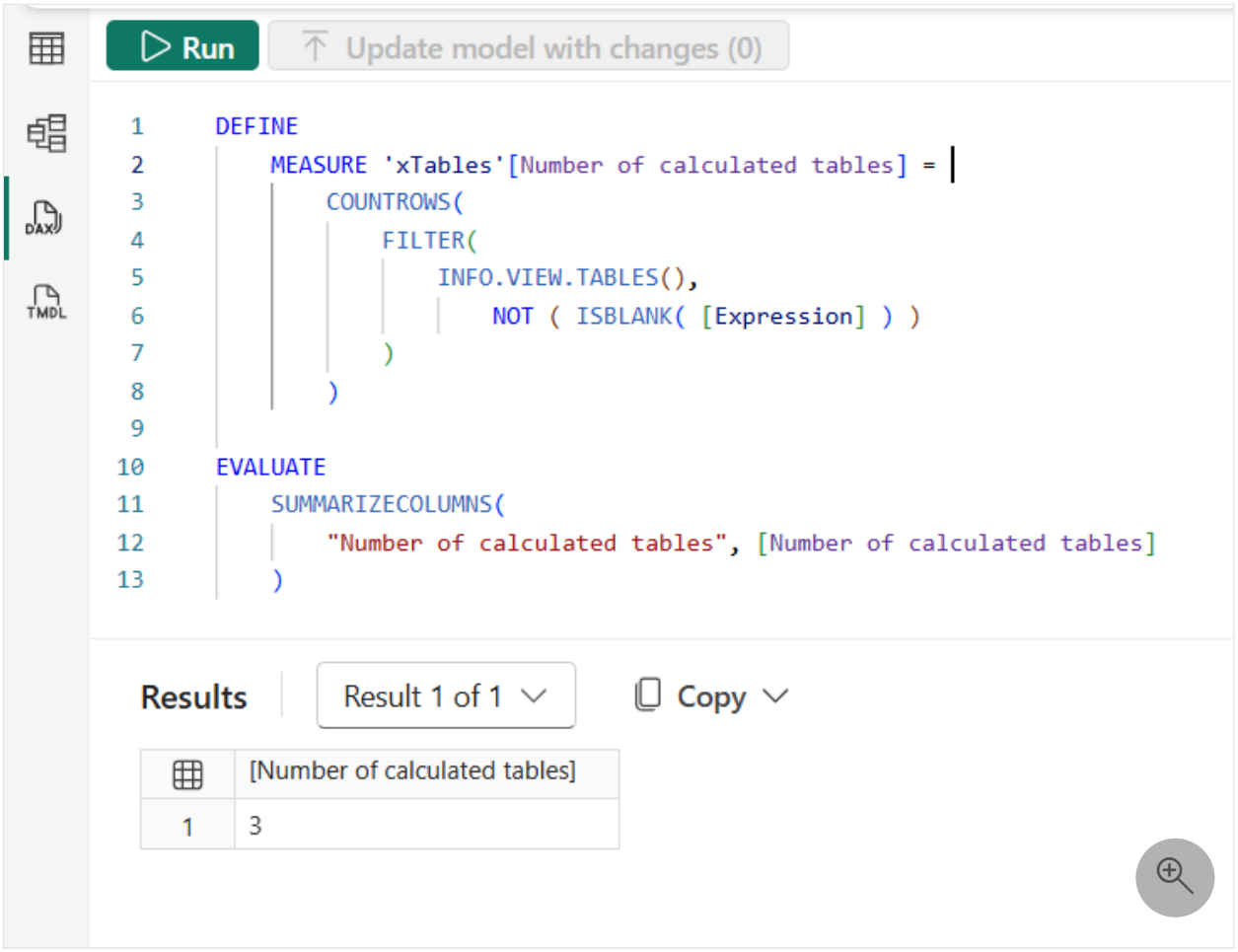


Или запрос DAX в представлении запросов DAX:

DAX

```
DEFINE
    MEASURE 'xTables'[Number of calculated tables] =
        COUNTROWS(
            FILTER(
                INFO.VIEW.TABLES(),
                    NOT ( ISBLANK( [Expression] ) )
            )
        )

EVALUATE
    SUMMARIZECOLUMNS(
        "Number of calculated tables", [Number of calculated tables]
    )
```



Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ИНФОРМАЦИЯ. АННОТАЦИИ

Статья • 18.03.2025

Область применения: ⊗ вычисляемый столбец ⊗ вычисляемой таблицы ⊗ ⊗ визуального вычисления ✓ DAX запроса

Возвращает таблицу с информацией о каждой заметке в семантической модели. Эта информация помогает понять модель.

Синтаксис

DAX

```
INFO.ANNOTATIONS([<Restriction name>, <Restriction value>], ...)
```

Параметры

Параметры являются необязательными для этой функции DAX. При использовании параметров необходимо задать оба параметра. Допускается несколько пар параметров. Имя ограничения и значение — текст и введены в двойные кавычки.

⛶ **Развернуть таблицу**

Срок	Определение
Имя ограничения	Имя ограничения, используемого для фильтрации результатов.
Значение ограничения	Значение, используемое для фильтрации результатов ограничения.

Ограничения

Как правило, все столбцы результатов функции DAX можно использовать в качестве ограничения. Также могут быть разрешены дополнительные ограничения.

Возвращаемое значение

Таблица со следующими столбцами:

⛶ **Развернуть таблицу**

Столбец	Описание
[ИДЕНТИФИКАТОР]	Ссылка на объект. Идентификаторы обычно создаются автоматически и не должны быть изменены после создания модели. Тип данных не имеет знака (4 байта).
[ObjectID]	Ссылка на объект, принадлежащий этому заметке.
[ObjectType]	Ссылка на тип объекта, которому принадлежит эта заметка.
[Имя]	Имя объекта, используемого в коде, скрипте и запросах.
[значение]	Значение заметки.
[ModifiedTime]	Время последнего изменения объекта.

Замечания

Пользователи могут запускаться только с разрешением на запись в семантической модели, а не при динамическом подключении к семантической модели в Power BI Desktop. Эту функцию можно использовать в DAX запросахи не может использоваться в вычислениях.

Пример 1. запрос DAX

Следующий DAX запрос можно запустить в представлении запросов DAX:

```
DAX

EVALUATE
    INFO.ANNOTATIONS()
```

Этот запрос DAX возвращает таблицу со всеми столбцами этой функции DAX.

Пример 2. запрос DAX с соединениями

Следующий DAX запрос можно запустить в представлении запросов DAX:

```
DAX

EVALUATE
    VAR _ObjectTypeEnum =
    DATATABLE(
    "ObjectType",INTEGER,
    "Object",STRING,
```

```
    {
    {0,"Null"},
    {1,"Model"},
    {2,"DataSource"},
    {3,"Table"},
    {4,"Column"},
    {5,"AttributeHierarchy"},
    {6,"Partition"},
    {7,"Relationship"},
    {8,"Measure"},
    {9,"Hierarchy"},
    {10,"Level"},
    {11,"Annotation"},
    {12,"KPI"},
    {13,"Culture"},
    {14,"ObjectTranslation"},
    {15,"LinguisticMetadata"},
    {29,"Perspective"},
    {30,"PerspectiveTable"},
    {31,"PerspectiveColumn"},
    {32,"PerspectiveHierarchy"},
    {33,"PerspectiveMeasure"},
    {34,"Role"},
    {35,"RoleMembership"},
    {36,"TablePermission"},
    {37,"Variation"},
    {38,"Set"},
    {39,"PerspectiveSet"},
    {40,"ExtendedProperty"},
    {41,"Expression"},
    {42,"ColumnPermission"},
    {43,"DetailRowsDefinition"},
    {44,"RelatedColumnDetails"},
    {45,"GroupByColumn"},
    {46,"CalculationGroup"},
    {47,"CalculationItem"},
    {48,"AlternateOf"},
    {49,"RefreshPolicy"},
    {50,"FormatStringDefinition"}
    }
    )

    VAR _INFO =
    INFO.ANNOTATIONS()

    VAR _CombinedTable =
    NATURALLEFTOUTERJOIN(
        _INFO,
        _ObjectTypeEnum
    )

    VAR _ModelObjects =
    UNION(
        SELECTCOLUMNS(
            INFO.MEASURES(),
```

```
            "ObjectID", [ID],
            "Object Name", [Name]
        ),
        SELECTCOLUMNS(
            INFO.TABLES(),
            "ObjectID", [ID],
            "Object Name", [Name]
        ),
        SELECTCOLUMNS(
            INFO.COLUMNS(),
            "ObjectID", [ID],
            "Object Name", [ExplicitName]
        ),
        SELECTCOLUMNS(
            INFO.EXPRESSIONS(),
            "ObjectID", [ID],
            "Object Name", [Name]
        ),
        SELECTCOLUMNS(
            INFO.MODEL(),
            "ObjectID", [ID],
            "Object Name", [Name]
        )
    )

    VAR _CombinedTable2 =
    NATURALLEFTOUTERJOIN(
        _CombinedTable,
        _ModelObjects
    )

    RETURN
        SELECTCOLUMNS(
            _CombinedTable2,
            "Object type", [Object],
            "Object", [Object Name],
            "Annotation name", [Name],
            "Annotation value", [Value]
        )
    ORDER BY [Annotation name]
```

Этот запрос DAX возвращает таблицу только с указанными столбцами и присоединением к другим функциям INFO DAX и таблице перечисления.

Перечисления

Столбцы, возвращаемые в этой функции INFO, предоставляют идентификатор перечисления или таблицы подстановки. Ниже приведены значения для каждого перечисления в этой функции DAX.

ObjectType

Развернуть таблицу

Число	Ценность
0	Недействительный
1	Модель
2	DataSource
3	Стол
4	Столбец
5	AttributeHierarchy
6	Раздел
7	Связь
8	Измерять
9	Иерархия
10	Уровень
11	Аннотация
12	Ключевой показатель эффективности
13	Культура
14	ObjectTranslation
15	Лингвистические метаданные
29	Перспектива
30	PerspectiveTable
31	PerspectiveColumn
32	PerspectiveHierarchy
33	PerspectiveMeasure
34	Роль
35	RoleMembership
36	TablePermission

Число	Ценность
37	Вариация
38	Набор
39	PerspectiveSet
40	ExtendedProperty
41	Выражение
42	ColumnPermission
43	DetailRowsDefinition
44	RelatedColumnDetails
45	GroupByColumn
46	Вычисляемая группа
47	CalculationItem
48	AlternateOf
49	RefreshPolicy
50	FormatStringDefinition

Эта таблица основана на официальной документации.

Чтобы присоединиться к функциям INFO, используйте этот DAX запрос.

DAX

```
EVALUATE
    DATATABLE(
    "ObjectType",INTEGER,
    "Object",STRING,
    {
    {0,"Null"},
    {1,"Model"},
    {2,"DataSource"},
    {3,"Table"},
    {4,"Column"},
    {5,"AttributeHierarchy"},
    {6,"Partition"},
    {7,"Relationship"},
    {8,"Measure"},
    {9,"Hierarchy"},
    {10,"Level"},
    {11,"Annotation"},
```

```
    {12,"KPI"},
    {13,"Culture"},
    {14,"ObjectTranslation"},
    {15,"LinguisticMetadata"},
    {29,"Perspective"},
    {30,"PerspectiveTable"},
    {31,"PerspectiveColumn"},
    {32,"PerspectiveHierarchy"},
    {33,"PerspectiveMeasure"},
    {34,"Role"},
    {35,"RoleMembership"},
    {36,"TablePermission"},
    {37,"Variation"},
    {38,"Set"},
    {39,"PerspectiveSet"},
    {40,"ExtendedProperty"},
    {41,"Expression"},
    {42,"ColumnPermission"},
    {43,"DetailRowsDefinition"},
    {44,"RelatedColumnDetails"},
    {45,"GroupByColumn"},
    {46,"CalculationGroup"},
    {47,"CalculationItem"},
    {48,"AlternateOf"},
    {49,"RefreshPolicy"},
    {50,"FormatStringDefinition"}
    }
    )
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ИНФОРМАЦИЯ. CALCDEPENDENCY

Статья • 18.03.2025

Область применения: ⊗ вычисляемый столбец ⊗ вычисляемой таблицы ⊗ ⊗ визуального вычисления ✓ DAX запроса

Возвращает таблицу с информацией о каждой зависимости вычислений в семантической модели. Эта информация помогает понять модель.

Синтаксис

DAX

```
INFO.CALCDEPENDENCY([<Restriction name>, <Restriction value>], ...)
```

Параметры

Параметры являются необязательными для этой функции DAX. При использовании параметров необходимо задать оба параметра. Допускается несколько пар параметров. Имя ограничения и значение — текст и введены в двойные кавычки.

⛶ **Развернуть таблицу**

Срок	Определение
Имя ограничения	Имя ограничения, используемого для фильтрации результатов.
Значение ограничения	Значение, используемое для фильтрации результатов ограничения.

Ограничения

Как правило, все столбцы результатов функции DAX можно использовать в качестве ограничения. Также могут быть разрешены дополнительные ограничения.

Возвращаемое значение

Таблица со следующими столбцами:

⛶ **Развернуть таблицу**

Столбец	Описание
[DATABASE_NAME]	Имя семантической модели.
[OBJECT_TYPE]	Тип объекта.
[TABLE]	Имя таблицы объекта.
[OBJECT]	Имя объекта.
[ВЫРАЖЕНИЕ]	Формула DAX объекта.
[REFERENCED_OBJECT_TYPE]	Тип объекта, на который ссылается этот объект. Объект зависит от этого объекта.
[REFERENCED_TABLE]	Имя таблицы объекта, на который ссылается ссылка.
[REFERENCED_OBJECT]	Имя объекта, на который ссылается ссылка.
[REFERENCED_EXPRESSION]	Формула DAX объекта, на который ссылается ссылка.
[ЗАПРОС]	Запрос, если он указан как ограничение.

Замечания

Пользователи могут запускаться только с разрешением на запись в семантической модели, а не при динамическом подключении к семантической модели в Power BI Desktop. Эту функцию можно использовать в DAX запросахи не может использоваться в вычислениях.

Вы также можете вызвать эту функцию DAX с помощью INFO. ЗАВИСИМОСТИ.

Пример 1. запрос DAX

Следующий DAX запрос можно запустить в представлении запросов DAX:

```
DAX

EVALUATE
    INFO.CALCDEPENDENCY()
```

Этот запрос DAX возвращает таблицу со всеми столбцами этой функции DAX.

Пример 2. запрос DAX с ограничением

Следующий DAX запрос можно запустить в представлении запросов DAX:

DAX

```
EVALUATE
    INFO.CALCDEPENDENCY("Query", "EVALUATE { [Orders] }")
```

Этот запрос DAX возвращает таблицу с объектами в семантической модели, необходимой для выполнения указанного запроса DAX. Имя ограничения запроса используется со значением запроса DAX. Этот DAX запрос получает выходные данные меры с именем Orders.

Если запрос имеет двойные кавычки, они могут быть экранированы с другой двойной кавычки, и вы можете при необходимости использовать VAR для хранения значения.

DAX

```
EVALUATE
    VAR _query =
        "EVALUATE
        SELECTCOLUMNS(
            'Date',
            ""Date"", [Date]
        )"
    RETURN
        INFO.CALCDEPENDENCY(
            "Query",
            _query
        )
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Информационные функции

Статья • 22.01.2025

DAX информационные функции смотрят на ячейку или строку, предоставляемую в качестве аргумента, и сообщает, соответствует ли значение ожидаемому типу. Например, функция ISERROR возвращает `TRUE`, если указанное значение содержит ошибку.

В этой категории

Развернуть таблицу

Функция	Описание
COLUMNSTATISTICS	Возвращает таблицу статистики по каждому столбцу в каждой таблице модели.
CONTAINS	Возвращает значение true, если значения для всех указанных столбцов существуют или содержатся в этих столбцах; в противном случае функция возвращает значение false.
CONTAINSROW	Возвращает `TRUE`, если строка значений существует или содержится в таблице, в противном случае возвращается `FALSE`.
CONTAINSSTRING	Возвращает `TRUE` или `FALSE`, указывающие, содержит ли одна строка другую строку.
CONTAINSSTRINGEXACT	Возвращает `TRUE` или `FALSE`, указывающие, содержит ли одна строка другую строку.
CUSTOMDATA	Возвращает содержимое свойства CustomData в строке подключения.
HASONEFILTER	Возвращает TRUE, если число напрямую отфильтрованных значений в `columnName` равно одному; в противном случае возвращает `FALSE`.
HASONEVALUE	Возвращает `TRUE`, если контекст для `columnName` отфильтрован только до одного отдельного значения. В противном случае `FALSE`.
ISAFTER	Логическая функция, которая эмулирует поведение предложения Start At и возвращает значение true для

Функция	Описание
	строки, которая соответствует всем параметрам условия.
ISBLANK	Проверяет, является ли значение пустым и возвращается `TRUE` или `FALSE`.
ISCROSSFILTERED	Возвращает `TRUE` при фильтрации `columnName` или другого столбца в той же или связанной таблице.
ISEMPTY	Проверяет, пуста ли таблица.
ISERROR	Проверяет, является ли значение ошибкой, и возвращает `TRUE` или `FALSE`.
ISEVEN	Возвращает `TRUE`, если число даже или `FALSE`, если число нечетно.
ISFILTERED	Возвращает `TRUE` при непосредственной фильтрации `columnName`.
ISINSCOPE	Возвращает значение true, если указанный столбец является уровнем в иерархии уровней.
ISLOGICAL	Проверяет, является ли значение логическим значением (`TRUE` или `FALSE`), и возвращает `TRUE` или `FALSE`.
ISNONTEXT	Проверяет, не является ли значение текстом (пустые ячейки не являются текстом) и возвращает `TRUE` или `FALSE`.
ISNUMBER	Проверяет, является ли значение числом и возвращается `TRUE` или `FALSE`.
ISODD	Возвращает `TRUE`, если число нечетно или `FALSE`, если число даже.
ISONORAFTER	Логическая функция, которая эмулирует поведение предложения Start At и возвращает значение true для строки, которая соответствует всем параметрам условия.
ISSELECTEDMEASURE	Используется выражениями для элементов вычисления, чтобы определить меру, которая находится в контексте, является одной из тех, которые указаны в списке мер.
ISSUBTOTAL	Создает другой столбец в выражении SUMMARIZE, возвращающего значение True, если строка содержит промежуточные значения столбца, заданного в

Функция	Описание
	качестве аргумента, в противном случае возвращает значение False.
ISTEXT	Проверяет, является ли значение текстом, и возвращает `TRUE` или `FALSE`.
NONVISUAL	Помечает фильтр значений в выражении SUMMARIZECOLUMNS как не визуальный элемент.
SELECTEDMEASURE	Используется выражениями для элементов вычисления для ссылки на меру, которая находится в контексте.
SELECTEDMEASUREFORMATSTRING	Используется выражениями для элементов вычисления для получения строки формата меры, которая находится в контексте.
SELECTEDMEASURENAME	Используется выражениями для элементов вычисления для определения меры, которая находится в контексте по имени.
USERCULTURE	Возвращает языковой стандарт для текущего пользователя.
USERNAME	Возвращает доменное имя и имя пользователя из учетных данных, предоставленных системе во время подключения.
USEROBJECTID	Возвращает идентификатор объекта текущего пользователя или идентификатор идентификатора безопасности.
USERPRINCIPALNAME	Возвращает имя участника-пользователя.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COLUMNSTATISTICS

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу статистики по каждому столбцу в каждой таблице модели.

Синтаксис

```
DAX

COLUMNSTATISTICS ()
```

Параметры

Эта функция не принимает никаких параметров.

Возвращаемое значение

Таблица статистики. Каждая строка этой таблицы представляет другой столбец в модели. К столбцам таблицы относятся:

- `Table Name`: таблица текущего столбца.
- `Column Name`: имя текущего столбца.
- `Min`: минимальное значение, найденное в текущем столбце.
- `Max`: максимальное значение, найденное в текущем столбце.
- `Cardinality`: количество уникальных значений, найденных в текущем столбце.
- `Max Length`: длина самой длинной строки, найденной в текущем столбце (применимо только для строковых столбцов).

Замечания

- Столбцы в состоянии ошибки и столбцы из вычисляемых таблиц области запроса не отображаются в результирующих таблицах.
- Если фильтр из контекста фильтра применяется к COLUMNSTATISTICS(), возвращается ошибка.
- Для двоичных типизированных столбцов статистика Min и Max будет иметь значения BLANK.

Пример

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

Следующий запрос DAX:

DAX

```
DEFINE
    TABLE FilteredProduct =
        FILTER (
            Product,
            [Color] == "Blue"
        )
    COLUMN Customer[Location] = [State-Province] & " " & [Country-Region]

EVALUATE
COLUMNSTATISTICS ()
```

Возвращает таблицу со статистикой по всем столбцам из всех таблиц в модели. Таблица также содержит статистику для вычисляемого столбца области запроса, Customer[Location]. Однако таблица не включает столбцы из вычисляемой таблицы области запроса FilteredProduct.

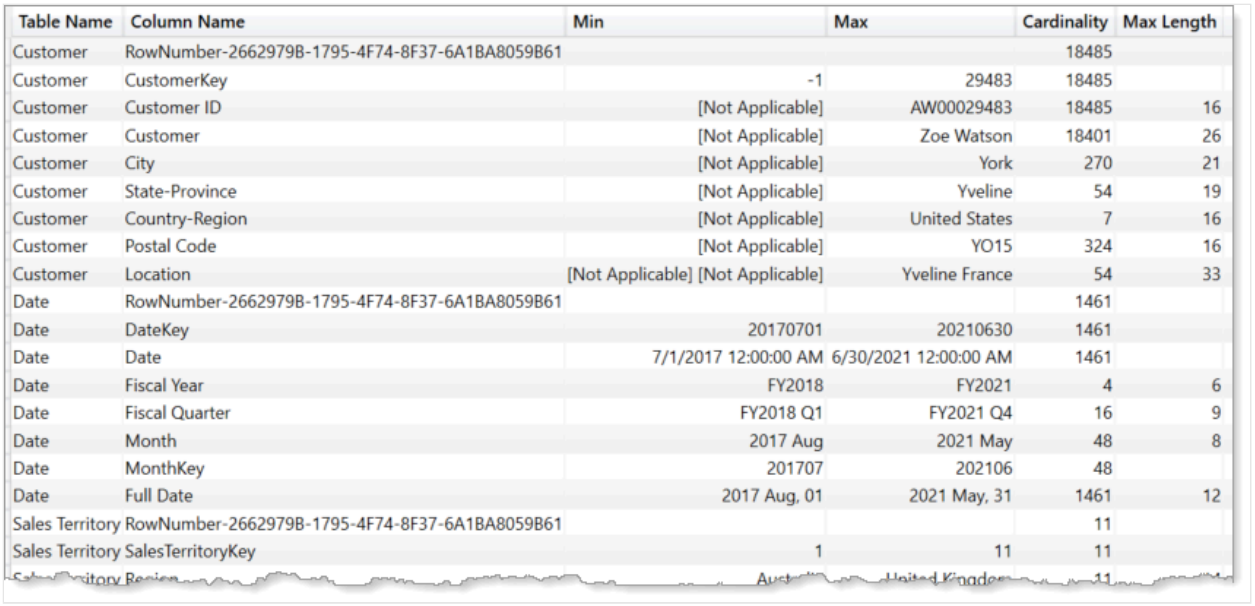


Table Name	Column Name	Min	Max	Cardinality	Max Length
Customer	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			18485	
Customer	CustomerKey	-1	29483	18485	
Customer	Customer ID	[Not Applicable]	AW00029483	18485	16
Customer	Customer	[Not Applicable]	Zoe Watson	18401	26
Customer	City	[Not Applicable]	York	270	21
Customer	State-Province	[Not Applicable]	Yveline	54	19
Customer	Country-Region	[Not Applicable]	United States	7	16
Customer	Postal Code	[Not Applicable]	YO15	324	16
Customer	Location	[Not Applicable]	[Not Applicable] Yveline France	54	33
Date	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			1461	
Date	DateKey	20170701	20210630	1461	
Date	Date	7/1/2017 12:00:00 AM	6/30/2021 12:00:00 AM	1461	
Date	Fiscal Year	FY2018	FY2021	4	6
Date	Fiscal Quarter	FY2018 Q1	FY2021 Q4	16	9
Date	Month	2017 Aug	2021 May	48	8
Date	MonthKey	201707	202106	48	
Date	Full Date	2017 Aug, 01	2021 May, 31	1461	12
Sales Territory	RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61			11	
Sales Territory	SalesTerritoryKey	1	11	11	
Sales Territory	Region	Aust	United Kingdom	11	

Связанное содержимое

функцияCALCULATETABLE контекста фильтра

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CONTAINS

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает значение true, если значения для всех указанных столбцов существуют или содержатся в этих столбцах; в противном случае функция возвращает значение false.

Синтаксис

```DAX
CONTAINS(<table>, <columnName>, <value>[, <columnName>, <value>]…)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Любое выражение DAX, возвращающее таблицу данных.
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX. Это не может быть выражением.
`value`	Любое выражение DAX, возвращающее одно скалярное значение, которое необходимо искать в `columnName`. Выражение вычисляется ровно один раз и перед передачей в список аргументов.

Возвращаемое значение

Значение `TRUE`, если каждый указанный `value` можно найти в соответствующем `columnName` или содержится в этих столбцах; в противном случае функция возвращает `FALSE`.

Замечания

- Аргументы `columnName` и `value` должны поступать парами; в противном случае возвращается ошибка.
- `columnName` должен принадлежать указанной `table` или таблице, связанной с `table`.
- Если `columnName` ссылается на столбец в связанной таблице, он должен быть полностью квалифицирован; в противном случае возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере создается мера, указывающая, были ли интернет-продажи продукта 214 и клиенту 11185 одновременно.

DAX

```
= CONTAINS(InternetSales, [ProductKey], 214, [CustomerKey], 11185)
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Функция CONTAINSROW

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает `TRUE`, если существует по крайней мере одна строка, где все столбцы имеют указанные значения.

Синтаксис

DAX

```
CONTAINSROW(<Table>, <Value> [, <Value> [, …] ] )
```

Параметры

Развернуть таблицу

Срок	Определение
`Table`	Таблица для тестирования.
`Value`	Любое допустимое выражение DAX, возвращающее скалярное значение.

Возвращаемое значение

`TRUE` или `FALSE`.

Замечания

- Кроме синтаксиса, оператор `IN` и функция CONTAINSROW функционально эквивалентны.

  DAX

  ```
  <scalarExpr> IN <tableExpr>
  ( <scalarExpr1>, <scalarExpr2>, … ) IN <tableExpr>
  ```

- Число скалярныхexprN должно соответствовать количеству столбцов в tableExpr.
- NOT IN не является оператором в DAX. Чтобы выполнить логическое отрицание оператора IN, поместите NOT перед всем выражением. Например, NOT [цвет] IN { "Красный", "Желтый", "Синий" }.

- В отличие от оператора = оператор IN и функция CONTAINSROW выполняют строгое сравнение. Например, значение BLANK не соответствует 0.

Примеры

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

Пример 1

Следующие запросы DAX:

DAX

```
EVALUATE
FILTER (
    ALL ( Product[Color] ),
    ( [Color] )
        IN {
        "Red",
        "Yellow",
        "Blue"
    }
)
ORDER BY [Color]
```

и

DAX

```
EVALUATE
FILTER (
    ALL ( Product[Color] ),
    CONTAINSROW (
        {
            "Red",
            "Yellow",
            "Blue"
        },
```

```
        [Color]
    )
)
ORDER BY [Color]
```

Верните следующую таблицу с одним столбцом:

Развернуть таблицу

[Цвет]
Синий
Красный
Жёлтый

Пример 2

Следующие эквивалентные запросы DAX:

DAX

```
EVALUATE
FILTER (
    ALL ( Product[Color] ),
    NOT [Color]
        IN {
        "Red",
        "Yellow",
        "Blue"
    }
)
ORDER BY [Color]
```

и

DAX

```
EVALUATE
FILTER (
    ALL ( Product[Color] ),
    NOT CONTAINSROW (
        {
            "Red",
            "Yellow",
            "Blue"
        },
        [Color]
```

```
    )
)
ORDER BY [Color]
```

Верните следующую таблицу с одним столбцом:

Развернуть таблицу

[Цвет]
Чёрный
Серый
Мульти
NA
Серебро
Silver\Black
Белый

Связанное содержимое

оператор INDAX запросов

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CONTAINSSTRING

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает `TRUE` или `FALSE`, указывающие, содержит ли одна строка другую строку.

Синтаксис

DAX

```
CONTAINSSTRING(<within_text>, <find_text>)
```

Параметры

Развернуть таблицу

Срок	Определение
`within_text`	Текст, в котором требуется выполнить поиск find_text.
`find_text`	Текст, который нужно найти.

Возвращаемое значение

`TRUE`, если find_text является подстрокой within_text; в противном случае `FALSE`.

Замечания

- CONTAINSSTRING не учитывает регистр.
- Вы можете использовать `?` и `*` подстановочные знаки. Используйте `~` для экранирования подстановочных знаков.

Пример

запрос DAX

DAX

```
EVALUATE
    ROW(
        "Case 1", CONTAINSSTRING("abcd", "bc"),
        "Case 2", CONTAINSSTRING("abcd", "BC"),
        "Case 3", CONTAINSSTRING("abcd", "a*d"),
        "Case 4", CONTAINSSTRING("abcd", "ef")
    )
```

Возвращает

Развернуть таблицу

[Случай 1]	[Случай 2]	[Случай 3]	[Случай 4]
TRUE	TRUE	TRUE	FALSE

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CONTAINSSTRINGEXACT

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает `TRUE` или `FALSE`, указывающие, содержит ли одна строка другую строку.

Синтаксис

DAX

```
CONTAINSSTRINGEXACT(<within_text>, <find_text>)
```

Параметры

Развернуть таблицу

Срок	Определение
`within_text`	Текст, в котором требуется выполнить поиск find_text.
`find_text`	Текст, который нужно найти.

Возвращаемое значение

`TRUE`, если find_text является подстрокой within_text; в противном случае `FALSE`.

Замечания

CONTAINSSTRINGEXACT учитывает регистр.

Пример

запрос DAX

DAX

```
EVALUATE
    ROW(
        "Case 1", CONTAINSSTRINGEXACT("abcd", "bc"),
        "Case 2", CONTAINSSTRINGEXACT("abcd", "BC"),
        "Case 3", CONTAINSSTRINGEXACT("abcd", "a*d"),
        "Case 4", CONTAINSSTRINGEXACT("abcd", "ef")
    )
```

Возвращает

Развернуть таблицу

[Случай 1]	[Случай 2]	[Случай 3]	[Случай 4]
TRUE	FALSE	FALSE	FALSE

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CUSTOMDATA

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает содержимое свойства `CustomData` в строке подключения.

Синтаксис

DAX

```
CUSTOMDATA()
```

Возвращаемое значение

Содержимое свойства `CustomData` в строке подключения.

Пусто, если свойство `CustomData` не определено во время подключения.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула DAX проверяет, задано ли свойство CustomData `OK`.

DAX

```
= IF(CUSTOMDATA()="OK", "Correct Custom data in connection string", "No
custom data in connection string property or unexpected value")
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

HASONEFILTER

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает `TRUE`, если число напрямую отфильтрованных значений в `columnName` равно одному; в противном случае возвращает `FALSE`.

Синтаксис

```
DAX

HASONEFILTER(<columnName>)
```

Параметры

Развернуть таблицу

Срок	Определение
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX. Это не может быть выражением.

Возвращаемое значение

`TRUE`, если число напрямую отфильтрованных значений в `columnName` равно одному; в противном случае возвращает `FALSE`.

Замечания

- Эта функция похожа на HASONEVALUE() с разницей, что HASONEVALUE() работает на основе перекрестных фильтров, а HASONEFILTER() работает прямым фильтром.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показано, как использовать HASONEFILTER() для возврата фильтра для ResellerSales_USD[ProductKey], если существует один фильтр, или возвратить BLANK, если нет фильтров или нескольких фильтров на ResellerSales_USD[ProductKey]).

DAX

```
=
IF(HASONEFILTER(ResellerSales_USD[ProductKey]),FILTERS(ResellerSales_USD[Pro
ductKey]),BLANK())
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

HASONEVALUE

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает `TRUE`, если контекст для `columnName` отфильтрован только до одного отдельного значения. В противном случае `FALSE`.

Синтаксис

```
HASONEVALUE(<columnName>)
```

Параметры

Развернуть таблицу

Срок	Определение
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Это не может быть выражением.

Возвращаемое значение

`TRUE`, если контекст для `columnName` отфильтрован только до одного отдельного значения. В противном случае `FALSE`.

Замечания

- Эквивалентное выражение для HASONEVALUE() `COUNTROWS(VALUES(<columnName>)) = 1`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула меры проверяет, срезает ли контекст по одному значению, чтобы оценить процент по предопределенному сценарию; В этом случае вы хотите сравнить продажи торговых посредников с продажами в 2007 году, необходимо знать, фильтруется ли контекст по одному году. Кроме того, если сравнение бессмысленно, вы хотите вернуть BLANK.

DAX

```
=
IF(HASONEVALUE(DateTime[CalendarYear]),SUM(ResellerSales_USD[SalesAmount_USD
])/CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),DateTime[CalendarYear]=
2007),BLANK())
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISAFTER

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Логическая функция, которая эмулирует поведение предложения Start At и возвращает значение true для строки, которая соответствует всем параметрам условия.

В зависимости от порядка сортировки первый параметр сравнивается со вторым параметром. Если порядок сортировки возрастает, то сравнение, выполняемого, является первым параметром, превышающим второй параметр. Если порядок сортировки убывается, то сравнение выполняется вторым параметром меньше первого параметра.

Синтаксис

DAX

```
ISAFTER(<scalar_expression>, <scalar_expression>[, sort_order [,
<scalar_expression>, <scalar_expression>[, sort_order]]…)
```

Параметры

Развернуть таблицу

Срок	Определение
`scalar expression`	Любое выражение, возвращающее скалярное значение, например ссылку на столбец или целое число или строковое значение. Как правило, первый параметр является ссылкой на столбец, а второй — скалярным значением.
`sort order`	(необязательно) Порядок сортировки столбца. Может быть возрастание (ASC) или убывание (ДЕК). По умолчанию порядок сортировки возрастает.

Возвращаемое значение

True или false.

Замечания

Эта функция аналогична ISONORAFTER. Разница ISAFTER возвращает значение true для значений, отсортированных строго *после* значений фильтра, где ISONORAFTER возвращает значение true для значений, отсортированных *или после* значений фильтра.

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Имя таблицы: "Info"

Развернуть таблицу

Страна или регион	Государство	Считать	Итог
IND	JK	20	800
IND	МГН	25	1000
IND	WB	10	900
США	Центр сертификации	5	500
США	WA	10	900

Следующее выражение:

DAX

```
FILTER (
    Info,
    ISAFTER (
        Info[Country], "IND", ASC,
        Info[State], "MH", ASC )
)
```

Возвращает:

Развернуть таблицу

Страна	Государство	Считать	Итог
IND	WB	10	900
США	Центр сертификации	5	500
США	WA	10	900

Связанное содержимое

ISONORAFTER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISBLANK

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Проверяет, является ли значение пустым и возвращается `TRUE` или `FALSE`.

Синтаксис

DAX

```
ISBLANK(<value>)
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Значение или выражение, которое требуется проверить.

Возвращаемое значение

Логическое значение `TRUE`, если значение пусто; в противном случае `FALSE`.

Замечания

Дополнительные сведения о рекомендациях при работе с BLANKS см. в статье Избегайте преобразования BLANKs в значения в DAX.

Пример

Эта формула вычисляет коэффициент увеличения или уменьшения объема продаж по сравнению с предыдущим годом. В этом примере функция IF используется для проверки значения продаж предыдущего года, чтобы избежать нулевой ошибки.

DAX

```
//Sales to Previous Year Ratio

= IF( ISBLANK('CalculatedMeasures'[PreviousYearTotalSales])
   , BLANK()
   , ( 'CalculatedMeasures'[Total Sales]-
'CalculatedMeasures'[PreviousYearTotalSales] )
      /'CalculatedMeasures'[PreviousYearTotalSales])
```

Результат

Развернуть таблицу

Метки строк	Всего продаж	Общий объем продаж за предыдущий год	Соотношение продаж к предыдущему году
2005	$10,209,985.08		
2006	$28,553,348,43	$10,209,985.08	179.66%
2007	$39,248,847,52	$28,553,348,43	37.46%
2008	$24,542,444.68	$39,248,847,52	-37.47%
Общий итог	$102,554,625.71		

Связанное содержимое

функции информационных функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISCROSSFILTERED

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает `TRUE`, когда указанная таблица или столбец перекрестно фильтруется.

Синтаксис

DAX

```
ISCROSSFILTERED(<TableNameOrColumnName>)
```

Параметры

Развернуть таблицу

Срок	Определение
`TableNameOrColumnName`	Имя существующей таблицы или столбца. Это не может быть выражением.

Возвращаемое значение

`TRUE`, если `ColumnName` или столбец `TableName` перекрестно фильтруется. В противном случае возвращается `FALSE`.

Замечания

- Столбец или таблица, как правило, перекрестно фильтруется при применении фильтра к `ColumnName`, любому столбцу `TableName` или любому столбцу связанной таблицы.
- Как сообщается, столбец или таблица фильтруется непосредственно при применении фильтра к `ColumnName` или к любому столбцу `TableName`. Поэтому функция ISFILTERED также возвращает `TRUE` при фильтрации `ColumnName` или любого столбца `TableName`.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

функция ISFILTEREDFILTERS функцииHASONEFILTERHASONEVALUE функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISEMPTY

Статья • 21.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Проверяет, пуста ли таблица.

Синтаксис

```DAX
ISEMPTY(<table_expression>)
```

Параметры

⛶ Развернуть таблицу

Срок	Определение
`table_expression`	Ссылка на таблицу или выражение DAX, возвращающее таблицу.

Возвращаемое значение

Значение True, если таблица пуста (не имеет строк), если в противном случае — false.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Для приведенной ниже таблицы с именем Info:

⛶ Развернуть таблицу

Страна или регион	Государство	Графство	Итог
IND	JK	20	800
IND	МГН	25	1000
IND	WB	10	900
США	Центр сертификации	5	500
США	WA	10	900

DAX

```
EVALUATE
ROW("Any countries with count > 25?", NOT(ISEMPTY(FILTER(Info,
[County]>25))))
```

Возвращаемое значение: 'FALSE''

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISERROR

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Проверяет, является ли значение ошибкой, и возвращает `TRUE` или `FALSE`.

Синтаксис

```
DAX

ISERROR(<value>)
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Значение, которое требуется проверить.

Возвращаемое значение

Логическое значение `TRUE`, если значение является ошибкой; в противном случае `FALSE`.

Замечания

- Рекомендации по использованию ISERRORсм. в разделе Соответствующее использование функций ошибок.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере вычисляется соотношение общих продаж Через Интернет к общему объему продаж торгового посредника. Функция ISERROR используется для проверки ошибок, таких как деление по нулю. Если возвращается сообщение об ошибке, возвращается пустое значение, в противном случае возвращается отношение.

DAX

```
= IF( ISERROR(
       SUM('ResellerSales_USD'[SalesAmount_USD])
       /SUM('InternetSales_USD'[SalesAmount_USD])
            )
    , BLANK()
    , SUM('ResellerSales_USD'[SalesAmount_USD])
      /SUM('InternetSales_USD'[SalesAmount_USD])
    )
```

Связанное содержимое

функцииIFERROR функцииIF

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISEVEN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает `TRUE`, если число даже или `FALSE`, если число нечетно.

Синтаксис

DAX

```
ISEVEN(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Значение для тестирования. Если число не является целым числом, оно усечено.

Возвращаемое значение

Возвращает `TRUE`, если число даже или `FALSE`, если число нечетно.

Замечания

- Если число не является числом, ISEVEN возвращает значение ошибки `#VALUE!`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISFILTERED

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает `TRUE`, когда указанная таблица или столбец фильтруется напрямую.

Синтаксис

DAX

```
ISFILTERED(<TableNameOrColumnName>)
```

Параметры

Развернуть таблицу

Срок	Определение
`TableNameOrColumnName`	Имя существующей таблицы или столбца. Это не может быть выражением.

Возвращаемое значение

`TRUE`, если `ColumnName` или столбец `TableName` фильтруется напрямую. В противном случае возвращается `FALSE`.

Замечания

- Как сообщается, столбец или таблица фильтруются непосредственно при применении фильтра к `ColumnName` или любому столбцу `TableName`.
- Столбец или таблица, как правило, перекрестно фильтруется при применении фильтра к `ColumnName`, любому столбцу `TableName` или любому столбцу связанной таблицы. Поэтому функция ISCROSSFILTERED также возвращает `TRUE` при `ColumnName`, любой столбец `TableName` или столбец связанной таблицы фильтруется.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

функция ISCROSSFILTEREDFILTERS функцииHASONEFILTERHASONEVALUE функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISINSCOPE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает значение true, если указанный столбец является уровнем в иерархии уровней.

Синтаксис

DAX

```
ISINSCOPE(<columnName>)
```

Параметры

Развернуть таблицу

Срок	Определение
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX. Это не может быть выражением.

Возвращаемое значение

`TRUE`, если указанный столбец является уровнем в иерархии уровней.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

DAX

```
DEFINE
MEASURE FactInternetSales[% of Parent] =
  SWITCH (TRUE(),
    ISINSCOPE(DimProduct[Subcategory]),
      DIVIDE(
        SUM(FactInternetSales[Sales Amount]),
        CALCULATE(
          SUM(FactInternetSales[Sales Amount]),
          ALLSELECTED(DimProduct[Subcategory]))
      ),
    ISINSCOPE(DimProduct[Category]),
      DIVIDE(
        SUM(FactInternetSales[Sales Amount]),
        CALCULATE(
          SUM(FactInternetSales[Sales Amount]),
          ALLSELECTED(DimProduct[Category]))
      ),
    1
  ) * 100
```

```
EVALUATE
  SUMMARIZECOLUMNS
  (
    ROLLUPADDISSUBTOTAL
    (
      DimProduct[Category], "Category Subtotal",
      DimProduct[Subcategory], "Subcategory Subtotal"
    ),
    TREATAS(
      {"Bike Racks", "Bike Stands", "Mountain Bikes", "Road Bikes", "Touring Bikes"},
      DimProduct[Subcategory]),
    "Sales", SUM(FactInternetSales[Sales Amount]),
    "% of Parent", [% of Parent]
  )
  ORDER BY
    [Category Subtotal] DESC, [Category],
    [Subcategory Subtotal] DESC, [Subcategory]
```

Возвращает

Развернуть таблицу

DimProduct[Категория]	DimProduct[SubCategory]	[Промежуточный итог категории]	[Промежуточный итог подкатегории]	[Продажи]	[% родительского]
		TRUE	TRUE	28,397,095.65	100.00
Принадлежности		FALSE	TRUE	78,951.00	0.28
Принадлежности	Велосипедные стойки	FALSE	FALSE	39,360.00	49.85
Принадлежности	Стенды велосипедов	FALSE	FALSE	39,591.00	50.15
Велосипеды		FALSE	TRUE	28,318,144.65	99.72
Велосипеды	Горные велосипеды	FALSE	FALSE	9,952,759.56	35.15
Велосипеды	Дорожные велосипеды	FALSE	FALSE	14,520,584.04	51.28
Велосипеды	Экскурсии велосипедов	FALSE	FALSE	3,844,801.05	13.58

Связанное содержимое

функция SUMMARIZECOLUMNSCALCULATE

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISLOGICAL

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Проверяет, является ли значение логическим значением (`TRUE` или `FALSE`), и возвращает `TRUE` или `FALSE`.

Синтаксис

```
DAX

ISLOGICAL(<value>)
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Значение, которое требуется проверить.

Возвращаемое значение

`TRUE`, если значение является логическим значением; `FALSE`, если любое значение, отличное от `TRUE` OR `FALSE`.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующих трех примерах показано поведение ISLOGICAL.

DAX

```
//RETURNS: Is Boolean type or Logical
= IF(ISLOGICAL(true), "Is Boolean type or Logical", "Is different type")

//RETURNS: Is Boolean type or Logical
= IF(ISLOGICAL(false), "Is Boolean type or Logical", "Is different type")

//RETURNS: Is different type
= IF(ISLOGICAL(25), "Is Boolean type or Logical", "Is different type")
```

Связанное содержимое

функции информационных функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISNONTEXT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Проверяет, не является ли значение текстом (пустые ячейки не являются текстом) и возвращает `TRUE` или `FALSE`.

Синтаксис

DAX

```
ISNONTEXT(<value>)
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Значение, которое требуется проверить.

Возвращаемое значение

`TRUE`, если значение не является текстом или пустым; `FALSE`, если значение является текстом.

Замечания

- Пустая строка считается текстом.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующих примерах показано поведение функции ISNONTEXT.

DAX

```
//RETURNS: Is Non-Text
= IF(ISNONTEXT(1), "Is Non-Text", "Is Text")

//RETURNS: Is Non-Text
= IF(ISNONTEXT(BLANK()), "Is Non-Text", "Is Text")

//RETURNS: Is Text
= IF(ISNONTEXT(""), "Is Non-Text", "Is Text")
```

Связанное содержимое

функции информационных функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISNUMBER

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Проверяет, является ли значение числом и возвращается `TRUE` или `FALSE`.

Синтаксис

```
DAX

ISNUMBER(<value>)
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Значение, которое требуется проверить.

Возвращаемое значение

`TRUE`, если значение числовое; в противном случае `FALSE`.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующих трех примерах показано поведение ISNUMBER.

```
DAX
```

```
//RETURNS: Is number
= IF(ISNUMBER(0), "Is number", "Is Not number")

//RETURNS: Is number
= IF(ISNUMBER(3.1E-1),"Is number", "Is Not number")

//RETURNS: Is Not number
= IF(ISNUMBER("123"), "Is number", "Is Not number")
```

Связанное содержимое

функции информационных функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISODD

Статья • 15.12.2024

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает `TRUE`, если число нечетно или `FALSE`, если число даже.

Синтаксис

DAX

```
ISODD(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Значение для тестирования. Если число не является целым числом, оно усечено.

Возвращаемое значение

Возвращает `TRUE`, если число нечетно или `FALSE`, если число даже.

Замечания

- Если число не является числом, ISODD возвращает значение ошибки `#VALUE!`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISONORAFTER

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Логическая функция, которая эмулирует поведение предложения Start At и возвращает значение true для строки, которая соответствует всем параметрам условия.

В зависимости от порядка сортировки первый параметр сравнивается со вторым параметром. Если порядок сортировки возрастает, то сравнение, выполняемого, является первым параметром, превышающим второй параметр. Если порядок сортировки убывается, то сравнение выполняется вторым параметром меньше первого параметра.

Синтаксис

DAX

```
ISONORAFTER(<scalar_expression>, <scalar_expression>[, sort_order [,
<scalar_expression>, <scalar_expression>[, sort_order]]…)
```

Параметры

Развернуть таблицу

Срок	Определение
`scalar expression`	Любое выражение, возвращающее скалярное значение, например ссылку на столбец или целое число или строковое значение. Как правило, первый параметр является ссылкой на столбец, а второй — скалярным значением.
`sort order`	(необязательно) Порядок сортировки столбца. Может быть возрастание (ASC) или убывание (DESC). По умолчанию порядок сортировки возрастает.

Возвращаемое значение

True или false.

Замечания

Эта функция аналогична ISAFTER. Разница ISONORAFTER возвращает значение true для значений, отсортированных *или после* значений фильтра, где ISAFTER возвращает значение true для значений, отсортированных строго *после* значений фильтра.

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Для следующей таблицы с именем Info:

Развернуть таблицу

Страна или регион	Государство	Считать	Итог
IND	JK	20	800
IND	МГН	25	1000
IND	WB	10	900
США	Центр сертификации	5	500
США	WA	10	900

Следующее выражение:

DAX

```
FILTER (
    Info,
    ISONORAFTER (
        Info[Country], "IND", ASC,
        Info[State], "MH", ASC )
)
```

Возвращает:

Развернуть таблицу

Страна или регион	Государство	Считать	Итог
IND	МГН	25	1000
IND	WB	10	900
США	Центр сертификации	5	500
США	WA	10	900

Связанное содержимое

ISAFTER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISSELECTEDMEASURE

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Используется выражениями для элементов вычисления, чтобы определить меру, которая находится в контексте, является одной из тех, которые указаны в списке мер.

Синтаксис

```DAX
ISSELECTEDMEASURE( M1, M2, ... )
```

Параметры

Развернуть таблицу

Срок	Определение
`M1, M2, ...`	Список мер.

Возвращаемое значение

Логическое значение, указывающее, является ли мера, которая в настоящее время находится в контексте, одной из тех, которые указаны в списке параметров.

Замечания

- Можно ссылаться только в выражении для элемента вычисления.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующее выражение элемента вычисления проверяет, является ли текущая мера одной из тех, которые указаны в списке параметров. Если меры переименованы, исправление формул будет отражать изменения имени в выражении.

DAX

```
IF (
    ISSELECTEDMEASURE ( [Expense Ratio 1], [Expense Ratio 2] ),
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

Связанное содержимое

SELECTEDMEASURE SELECTEDMEASURENAME

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISSUBTOTAL

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Создает другой столбец в выражении SUMMARIZE, возвращающего значение True, если строка содержит промежуточные значения столбца, заданного в качестве аргумента, в противном случае возвращает значение False.

Синтаксис

DAX

```
ISSUBTOTAL(<columnName>)
```

С SUMMARIZE,

DAX

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]…[,
ROLLUP(<groupBy_columnName>[,< groupBy_columnName>…])][, <name>,
{<expression>|ISSUBTOTAL(<columnName>)}]…)
```

Параметры

Развернуть таблицу

Срок	Определение
`columnName`	Имя любого столбца в таблице функции SUMMARIZE или любого столбца в связанной таблице.

Возвращаемое значение

Значение True, если строка содержит значение промежуточных итогов для столбца, заданного в качестве аргумента, в противном случае возвращает значение False.

Замечания

- Эту функцию можно использовать только в выражении функции SUMMARIZE.
- Эта функция должна предшествовать имени логического столбца.

Пример

См. SUMMARIZE.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISTEXT

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Проверяет, является ли значение текстом, и возвращает `TRUE` или `FALSE`.

Синтаксис

DAX

```
ISTEXT(<value>)
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Значение, которое требуется проверить.

Возвращаемое значение

`TRUE`, если значение является текстом; в противном случае `FALSE`.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующих примерах показано поведение функции ISTEXT.

DAX

```
//RETURNS: Is Text
= IF(ISTEXT("text"), "Is Text", "Is Non-Text")

//RETURNS: Is Text
= IF(ISTEXT(""), "Is Text", "Is Non-Text")

//RETURNS: Is Non-Text
= IF(ISTEXT(1), "Is Text", "Is Non-Text")

//RETURNS: Is Non-Text
= IF(ISTEXT(BLANK()), "Is Text", "Is Non-Text")
```

Связанное содержимое

функции информационных функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NONVISUAL

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Помечает фильтр значений в выражении SUMMARIZECOLUMNS как не визуальный элемент. Эту функцию можно использовать только в выражении SUMMARIZECOLUMNS.

Синтаксис

```DAX
NONVISUAL(<expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Любое выражение DAX, возвращающее одно значение (а не таблицу).

Возвращаемое значение

Таблица значений.

Замечания

- Помечает фильтр значений в SUMMARIZECOLUMNS так же, как не влияет на значения мер, но применяется только к столбцам по группам.
- Эту функцию можно использовать только в выражении SUMMARIZECOLUMNS. Он используется в качестве аргумента filterTable функции SUMMARIZECOLUMNS или аргумента groupLevelFilter функции ROLLUPADDISSUBTOTAL или ROLLUPISSUBTOTAL.

Пример

См. SUMMARIZECOLUMNS.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SELECTEDMEASURE

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Используется выражениями для элементов вычислений или строк динамического формата для ссылки на меру, которая находится в контексте.

Синтаксис

```DAX
SELECTEDMEASURE()
```

Параметры

Никакой

Возвращаемое значение

Ссылка на меру, которая в настоящее время находится в контексте при вычислении элемента вычисления или строки форматирования.

Замечания

- Можно ссылаться только в выражении для элемента вычисления или строки форматирования.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующее выражение элемента вычисления вычисляет дату года независимо от того, что мера находится в контексте.

```DAX
```

```
CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))
```

Следующее выражение можно использовать для динамической настройки строки формата меры на основе того, является ли значение сотнями, тысячами или миллионами.

DAX

```
SWITCH(
TRUE(),
SELECTEDMEASURE() < 1000,"$#,##0",            //Values less than 1000 have
no text after them
SELECTEDMEASURE() < 1000000, "$#,##0,.0 K",   //Values between 1000 and
1000000 are formatted as #.## K
"$#,##0,,.0 M"                                //Values greater than 1000000
are formatted as #.## M
)
```

Связанное содержимое

SELECTEDMEASURENAME ISSELECTEDMEASURE

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SELECTEDMEASUREFORMATSTRING

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Используется выражениями для элементов вычисления для получения строки формата меры, которая находится в контексте.

Синтаксис

```
DAX

SELECTEDMEASUREFORMATSTRING()
```

Параметры

Никакой

Возвращаемое значение

Строка, содержащая строку формата меры, которая в настоящее время находится в контексте при вычислении элемента вычисления.

Замечания

- Эта функция может ссылаться только в выражениях для элементов вычислений в группах вычислений. Он предназначен для использования свойством **Format String Expression** элементов вычислений.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующее выражение вычисляется свойством Format String Expression для элемента вычисления. Если в контексте фильтра существует одна валюта, строка

формата извлекается из столбца DimCurrency[FormatString]; в противном случае используется строка формата меры в контексте.

```
DAX
SELECTEDVALUE( DimCurrency[FormatString], SELECTEDMEASUREFORMATSTRING() )
```

Связанное содержимое

SELECTEDMEASURE ISSELECTEDMEASURE

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SELECTEDMEASURENAME

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Используется выражениями для элементов вычисления для определения меры, которая находится в контексте по имени.

Синтаксис

```DAX
SELECTEDMEASURENAME()
```

Параметры

Никакой

Возвращаемое значение

Строковое значение, которое содержит имя меры, которая в настоящее время находится в контексте при вычислении элемента вычисления.

Замечания

- Можно ссылаться только в выражении для элемента вычисления.
- Эта функция часто используется для отладки при создании групп вычислений.

Пример

Следующее выражение элемента вычисления проверяет, является ли текущая мера коэффициентом расходов и условно применяет логику вычисления. Так как проверка основана на сравнении строк, она не подлежит исправлению формул и не будет получать выгоду от переименования объекта автоматически отраженным. Аналогичное сравнение, которое будет использовать исправление формул, см. вместо этого функцию ISSLECTEDMEASURE.

DAX

```
IF (
    SELECTEDMEASURENAME = "Expense Ratio",
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

Связанное содержимое

SELECTEDMEASURE ISSELECTEDMEASURE

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

USERCULTURE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает языковой стандарт (код страны языка) для текущего пользователя, определяемого операционной системой, параметрами браузера или службой Power BI.

Синтаксис

DAX

```
USERCULTURE()
```

Параметры

Это выражение не имеет параметров.

Возвращаемое значение

Языковой стандарт в виде строки.

Замечания

- В службе Power BI языковой стандарт определяется **параметры**>**языка**>**языковых параметров**. Значение по умолчанию определяется языковым параметром браузера пользователя.
- При использовании в вычисляемой таблице и вычисляемых выражениях столбцов результат может отличаться в зависимости от того, находится ли таблица в режиме DirectQuery или import. В режиме DirectQuery результат определяется языком (языковым стандартом), указанным в параметрах языка в службе Power BI. Значение по умолчанию в параметрах языка определяется языковым стандартом браузера пользователя, что означает, что одна вычисляемая таблица или столбец может возвращать разные результаты в зависимости от параметров языка браузера для каждого пользователя. При использовании режима импорта результат статически определяется во время

обновления и не будет изменяться во время запроса. Для управляемых обновлений, таких как запланированный или интерактивный, языковой стандарт не основан на параметре языка браузера пользователя, а вместо этого использует инвариантный языковой стандарт. Однако инвариантный языковой стандарт можно переопределить с помощью конечной точки XMLA, чтобы указать пользовательский языковой стандарт.

- В сочетании с функцией параметров поля в Power BI USERCULTURE можно использовать для надежного перевода заголовков и подписей динамической визуализации при использовании в выражениях объектов уровня строк (RLS) в одной модели. Однако выражения, содержащие USERCULTURE, вызываемые извне модели, такие как запросы и меры отчета live-connect, не должны полагаться на правильно переведенные заголовки и субтитры.
- USERCULTURE возвращает правильный языковой стандарт пользователя при использовании в выражениях объектов, вызываемых в модели, таких как меры, безопасность на уровне строк (RLS) и элементы вычисления. Однако он может не возвращать правильный языковой стандарт пользователя при использовании в выражениях извне модели, таких как запросы и меры отчета live-connect.
- В отчетах Live-connect USERCULTURE могут не возвращать правильный языковой стандарт пользователя при вызове из выражения меры отчета.

Пример

Для следующего выражения

DAX

```
FORMAT(TODAY(), "dddd", USERCULTURE())
```

В зависимости от параметра языка для текущего пользователя USERCULTURE возвращает текущий день, например:

Развернуть таблицу

Локаль	Форматированный рабочий день
de-DE	Dienstag
en-US	Tuesday

Локаль	Форматированный рабочий день
es-ES_tradnl	martes
eu-ES	asteartea
it-IT	martedì
nl-NL	dinsdag
pl-PL	wtorek
ro-RO	marți
ru-RU	вторник
uk-UA	вівторок

Связанное содержимое

заголовки на основе выражений в Power
BIUSERNAMEUSERPRINCIPALNAMEUSEROBJECTID

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

USERNAME

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает доменное имя и имя пользователя из учетных данных, предоставленных системе во время подключения.

Синтаксис

```DAX
USERNAME()
```

Параметры

Это выражение не имеет параметров.

Возвращаемое значение

Имя пользователя из учетных данных, предоставленных системе во время подключения

Пример

Следующая формула проверяет, является ли имя входа пользователя частью UsersTable.

```DAX
= IF(CONTAINS(UsersTable,UsersTable[login], USERNAME()), "Allowed", BLANK())
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

USEROBJECTID

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает идентификатор объекта текущего пользователя из идентификатора Microsoft Entra ID или идентификатора безопасности (SID).

Синтаксис

```DAX
USEROBJECTID()
```

Параметры

Это выражение не имеет параметров.

Возвращаемое значение

Идентификатор объекта текущего пользователя из идентификатора Microsoft Entra для моделей Power BI или Azure Analysis Services или sid для моделей SQL Server Analysis Services.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

USERPRINCIPALNAME

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает имя участника-пользователя.

Синтаксис

```
DAX

USERPRINCIPALNAME()
```

Параметры

Это выражение не имеет параметров.

Возвращаемое значение

Имя пользователяprincipalname во время подключения.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Логические функции

Статья • 22.01.2025

Логические функции действуют на выражение для возврата сведений о значениях или наборах в выражении. Например, можно использовать функцию IF для проверки результата выражения и создания условных результатов.

В этой категории

Развернуть таблицу

Функция	Описание
AND	Проверяет, `TRUE` ли оба аргумента, и возвращает `TRUE`, если оба аргумента `TRUE`.
BITAND	Возвращает битовое значение "AND" из двух чисел.
BITLSHIFT	Возвращает число, смещенное влево указанным числом битов.
BITOR	Возвращает битовое значение "OR" из двух чисел.
BITRSHIFT	Возвращает число, смещенное вправо по указанному числу битов.
BITXOR	Возвращает битовое значение XOR из двух чисел.
COALESCE	Возвращает первое выражение, которое не оценивается BLANK.
FALSE	Возвращает логическое значение `FALSE`.
IF	Проверяет условие и возвращает одно значение при `TRUE`, в противном случае возвращается второе значение.
IF.EAGER	Проверяет условие и возвращает одно значение при `TRUE`, в противном случае возвращается второе значение. Использует план выполнения, который всегда выполняет выражения ветви независимо от выражения условия.
IFERROR	Вычисляет выражение и возвращает указанное значение, если выражение возвращает ошибку
NOT	Изменяет `FALSE` на `TRUE` или `TRUE` на `FALSE`.
OR	Проверяет, является ли один из аргументов `TRUE` возвращать `TRUE`.
SWITCH	Вычисляет выражение по списку значений и возвращает одно из нескольких возможных выражений результатов.
TRUE	Возвращает логическое значение `TRUE`.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

AND

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Проверяет, `TRUE` ли оба аргумента, и возвращает `TRUE`, если оба аргумента `TRUE`. В противном случае возвращается `FALSE`.

Синтаксис

DAX

```
AND(<logical1>,<logical2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`logical_1, logical_2`	Логические значения, которые требуется проверить.

Возвращаемое значение

Возвращает значение true или false в зависимости от сочетания тестируемых значений.

Замечания

Функция `AND` в DAX принимает только два аргумента (2). Если необходимо выполнить операцию AND для нескольких выражений, можно создать ряд вычислений или лучше использовать оператор `AND` (`&&`) для объединения всех из них в более простом выражении.

Пример 1

В следующей формуле показан синтаксис функции `AND`.

DAX

```
= IF(AND(10 > 9, -10 < -1), "All true", "One or more false"
```

Так как оба условия, передаваемые в качестве аргументов, в функцию `AND` имеют значение true, формула возвращает значение All True.

Пример 2

В следующем примере используется функция `AND` с вложенными формулами для сравнения двух наборов вычислений одновременно. Для каждой категории продуктов формула определяет, превышают ли продажи в текущем году и предыдущие продажи канала Интернета в течение одного и того же периода. Если оба условия имеют значение true, для каждой категории формула возвращает значение "Хит в Интернете".

DAX

```
= IF( AND(  SUM( 'InternetSales_USD'[SalesAmount_USD])
           >SUM('ResellerSales_USD'[SalesAmount_USD])
          , CALCULATE(SUM('InternetSales_USD'[SalesAmount_USD]),
PREVIOUSYEAR('DateTime'[DateKey] ))
           >CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),
PREVIOUSYEAR('DateTime'[DateKey] ))
          )
     , "Internet Hit"
     , ""
     )
```

Возвращает

Развернуть таблицу

Метки строк	2005	2006	2007	2008	-	Общий итог
Bib-Shorts						
Велосипедные стойки						
Стенды велосипедов				Интернет-хит		
Бутылки и клетки				Интернет-хит		
Нижние скобки						
Тормоза						

Метки строк	2005	2006	2007	2008	-	Общий итог
Прописные буквы						
Путы						
Химчистка						
Cranksets						
Derailleurs						
Крылья				Интернет-хит		
Вилки						
Перчатки						
Руле						
Гарнитуры						
Шлемы						
Пакеты гидратации						
Трикотажные изделия						
Светофор						
Замки						
Горные велосипеды						
Горные кадры						
Корзины						
Педали						
Лакированные бальные туфли						
Дорожные велосипеды						
Дорожные кадры						
Седла						
Шорты						
Носки						
Трико						

Метки строк	2005	2006	2007	2008	-	Общий итог
Шины и трубы				Интернет-хит		
Экскурсии велосипедов						
Обзор кадров						
Жилеты						
Колеса						
Общий итог						

Связанное содержимое

логические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

BITAND

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает побитовое AND двух чисел.

Синтаксис

```DAX
BITAND(<number>, <number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Number`	Любое скалярное выражение, возвращающее число. Если не целое число, оно усечено.

Возвращаемое значение

Побитовая AND двух чисел.

Замечания

- Эта функция поддерживает как положительные, так и отрицательные числа.

Пример

Следующий запрос DAX:

```DAX
EVALUATE { BITAND(13, 11) }
```

Возвращает значение 9.

Связанное содержимое

BITLSHIFT BITRSHIFT BITOR BITXOR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

BITLSHIFT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает число, смещенное влево указанным числом битов.

Синтаксис

DAX

```
BITLSHIFT(<Number>, <Shift_Amount>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Number`	Любое DAX выражение, возвращающее целочисленное выражение.
`Shift_Amount`	Любое DAX выражение, возвращающее целочисленное выражение.

Возвращаемое значение

Целочисленное значение.

Замечания

- Не забудьте понять характер операций bitshift и переполнения или переполнения целых чисел перед использованием DAX функций bitshift.
- Если Shift_Amount отрицательный, он будет двигаться в противоположном направлении.
- Если абсолютное значение Shift_Amount больше 64, ошибка не будет, но приведет к переполнению или переполнению.
- Нет ограничений на число, но результат может переполнения или переполнения.

Примеры

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
    { BITLSHIFT(2, 3) }
```

Возвращает 16.

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE
    { BITLSHIFT(128, -1) }
```

Возвращает значение 64.

Пример 3

Следующий запрос DAX:

```
DAX

Define
    Measure Sales[LeftShift] = BITLSHIFT(SELECTEDVALUE(Sales[Amount]), 3)

EVALUATE
SUMMARIZECOLUMNS(
    Sales[Amount],
    "LEFTSHIFT",
    [LeftShift]
)
```

Сдвиги оставили каждую сумму продаж с 3 битами и возвращает бит смещенную сумму продаж.

Связанное содержимое

BITRSHIFT BITAND BITOR BITXOR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

BITOR

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает побитовое OR двух чисел.

Синтаксис

```
DAX
BITOR(<number>, <number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Number`	Любое скалярное выражение, возвращающее число. Если не целое число, оно усечено.

Возвращаемое значение

Побитовая OR двух чисел.

Замечания

- Эта функция поддерживает как положительные, так и отрицательные числа.

Пример

Следующий запрос DAX:

```
DAX
EVALUATE
    { BITOR(9, 10) }
```

Возвращает значение 11.

Связанное содержимое

BITAND BITXOR BITLSHIFT BITRSHIFT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

BITRSHIFT

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает число, смещенное вправо по указанному числу битов.

Синтаксис

DAX

```
BITRSHIFT(<Number>, <Shift_Amount>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Number`	Любое DAX выражение, возвращающее целочисленное выражение.
`Shift_Amount`	Любое DAX выражение, возвращающее целочисленное выражение.

Возвращаемое значение

Целочисленное значение.

Замечания

- Не забудьте понять характер операций bitshift и переполнения или переполнения целых чисел перед использованием DAX функций bitshift.
- Если Shift_Amount отрицательный, он будет двигаться в противоположном направлении.
- Если абсолютное значение Shift_Amount больше 64, ошибка не будет, но приведет к переполнению или переполнению.
- Нет ограничений на число, но результат может переполнения или переполнения.

Примеры

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE
    { BITRSHIFT(16, 3) }
```

Возвращает 2.

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE
    { BITRSHIFT(1024, -3) }
```

Возвращает значение 8192.

Пример 3

Следующий запрос DAX:

```
DAX

Define
    Measure Sales[RightShift] = BITRSHIFT(SELECTEDVALUE(Sales[Amount]), 3)

EVALUATE
SUMMARIZECOLUMNS(
    Sales[Amount],
    "RIGHTSHIFT",
    [RightShift]
)
```

Сдвигает вправо каждую сумму продаж с 3 битами и возвращает бит смещенную сумму продаж.

Связанное содержимое

BITLSHIFT BITAND BITOR BITXOR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

BITXOR

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает битовое XOR из двух чисел.

Синтаксис

```DAX
BITXOR(<number>, <number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Number`	Любое скалярное выражение, возвращающее число. Если не целое число, оно усечено.

Возвращаемое значение

Побитовый XOR двух чисел.

Замечания

- Эта функция поддерживает как положительные, так и отрицательные числа.

Пример

Следующий запрос DAX:

```DAX
EVALUATE { BITXOR(9, 10) }
```

Возвращает 3.

Связанное содержимое

BITOR BITAND BITLSHIFT BITRSHIFT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COALESCE

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает первое выражение, которое не оценивается BLANK. Если все выражения оцениваются как BLANK, возвращается BLANK.

Синтаксис

DAX

```
COALESCE(<expression>, <expression>[, <expression>]…)
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Любое выражение DAX, возвращающее скалярное выражение.

Возвращаемое значение

Скалярное значение, исходящее из одного из выражений или BLANK, если все выражения оцениваются BLANK.

Замечания

Входные выражения могут быть различными типами данных.

Пример 1

Следующий запрос DAX:

DAX

```
EVALUATE { COALESCE(BLANK(), 10, DATE(2008, 3, 3)) }
```

Возвращает `10`, которое является первым выражением, которое не оценивается BLANK.

Пример 2

Следующее выражение DAX:

DAX

```
= COALESCE(SUM(FactInternetSales[SalesAmount]), 0)
```

Возвращает сумму всех значений в столбце SalesAmount в таблице FactInternetSales или `0`. Это можно использовать для преобразования BLANK значений общего объема продаж в `0`.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FALSE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает логическое значение `FALSE`.

Синтаксис

```DAX
FALSE()
```

Возвращаемое значение

Всегда `FALSE`.

Замечания

Слово `FALSE` также интерпретируется как логическое значение `FALSE`.

Пример

Формула возвращает логическое значение `FALSE`, если значение в столбце "InternetSales_USD"[SalesAmount_USD], меньше или равно 200000.

```DAX
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

В следующей таблице показаны результаты, когда пример формулы используется с ProductCategory[ProductCategoryName] в метках строк и DateTime[CalendarYear] в метках столбцов.

Развернуть таблицу

Метки строк	2005	2006	2007	2008	-	Общий итог
Принадлежности	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
Велосипеды	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
Одежда	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Компоненты	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Общий итог	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE

Связанное содержимое

функция TRUENOT функцииIF

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

IF

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Проверяет условие и возвращает одно значение, если оно `TRUE`, в противном случае возвращается второе значение.

Синтаксис

DAX

```
IF(<logical_test>, <value_if_true>[, <value_if_false>])
```

Параметры

Развернуть таблицу

Срок	Определение
`logical_test`	Любое значение или выражение, которое можно оценить как `TRUE` или `FALSE`.
`value_if_true`	Значение, возвращаемое, если логический тест `TRUE`.
`value_if_false`	(Необязательно) Значение, возвращаемое, если логический тест `FALSE`. Если опущено, возвращается BLANK.

Возвращаемое значение

Либо `value_if_true`, `value_if_false` или `BLANK`.

Замечания

- Функция IF может возвращать вариантный тип данных, если `value_if_true` и `value_if_false` имеют разные типы данных, но функция пытается вернуть один тип данных, если оба `value_if_true` и `value_if_false` имеют числовые типы данных. В последнем случае функция IF неявно преобразует типы данных для размещения обоих значений.

Например, формула `IF(<condition>, TRUE(), 0)` возвращает `TRUE` или 0, но формула `IF(<condition>, 1.0, 0)` возвращает только десятичные значения, даже если `value_if_false` имеет весь тип данных чисел. Дополнительные сведения о неявном преобразовании типов данных см. в типах данных.

- Чтобы выполнить выражения ветви независимо от выражения условия, используйте вместо этого IF.EAGER.

Примеры

Следующие определения вычисляемых столбцов **Product** используют функцию IF различными способами классификации каждого продукта на основе его цены на список.

В первом примере проверяется, является ли значение столбца цен на меньше 500. Если это условие имеет значение true, возвращается значение **low**. Так как нет значения `value_if_false`, возвращается BLANK.

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

DAX

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low"
)
```

Второй пример использует тот же тест, но на этот раз включает значение `value_if_false`. Таким образом, формула классифицирует каждый продукт как `Low` или `High`.

DAX

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low",
    "High"
)
```

Третий пример использует тот же тест, но на этот раз вложена функция IF для выполнения дополнительного теста. Таким образом, формула классифицирует каждый продукт как `Low`, `Medium` или `High`.

DAX

```
Price Group =
IF(
    'Product'[List Price] < 500,
    "Low",
    IF(
        'Product'[List Price] < 1500,
        "Medium",
        "High"
    )
)
```

Совет

Если необходимо вложить несколько функций IF, функция SWITCH может оказаться лучшей. Эта функция предоставляет более элегантный способ записи выражения, возвращающего более двух возможных значений.

Связанное содержимое

функция IF.EAGERSWITCH (DAX)логические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

IF.EAGER

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Проверяет условие и возвращает одно значение при `TRUE`, в противном случае возвращается второе значение. Он использует план выполнения, который всегда выполняет выражения ветви независимо от выражения условия.

Синтаксис

DAX

```
IF.EAGER(<logical_test>, <value_if_true>[, <value_if_false>])
```

Параметры

Развернуть таблицу

Срок	Определение
`logical_test`	Любое значение или выражение, которое можно оценить как `TRUE` или `FALSE`.
`value_if_true`	Значение, возвращаемое, если логический тест `TRUE`.
`value_if_false`	(Необязательно) Значение, возвращаемое, если логический тест `FALSE`. Если опущено, возвращается `BLANK`.

Возвращаемое значение

Либо `value_if_true`, `value_if_false` или `BLANK`.

Замечания

- Функция IF.EAGER может возвращать вариантный тип данных, если value_if_true и value_if_false имеют разные типы данных, но функция пытается вернуть один тип данных, если оба `value_if_true` и `value_if_false` имеют

числовые типы данных. В последнем случае функция IF.EAGER неявно преобразует типы данных для размещения обоих значений.

Например, формула `IF.EAGER(<condition>, TRUE(), 0)` возвращает `TRUE` или 0, но формула `IF.EAGER(<condition>, 1.0, 0)` возвращает только десятичные значения, даже если `value_if_false` имеет весь тип данных чисел. Дополнительные сведения о неявном преобразовании типов данных см. в типах данных.

- IF.EAGER имеет то же функциональное поведение, что и функция IF, но производительность может отличаться из-за различий в планах выполнения. `IF.EAGER(<logical_test>, <value_if_true>, <value_if_false>)` имеет тот же план выполнения, что и следующее выражение DAX:

  DAX

  ```
  VAR _value_if_true = <value_if_true>
  VAR _value_if_false = <value_if_false>
  RETURN
  IF (<logical_test>, _value_if_true, _value_if_false)
  ```

  Примечание. Два выражения ветви вычисляются независимо от выражения условия.

Примеры

См. IF примеры.

Связанное содержимое

функция IFлогических функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

IFERROR

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Вычисляет выражение и возвращает указанное значение, если выражение возвращает ошибку; в противном случае возвращает значение самого выражения.

Синтаксис

DAX

```
IFERROR(value, value_if_error)
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Любое значение или выражение.
`value_if_error`	Любое значение или выражение.

Возвращаемое значение

Скаляр того же типа, что и `value`

Замечания

- Функцию IFERROR можно использовать для перехвата и обработки ошибок в выражении.
- Если `value` или `value_if_error` является пустой ячейкой, IFERROR обрабатывает ее как пустое строковое значение ("").
- Функция IFERROR основана на функции IF и использует те же сообщения об ошибках, но имеет меньше аргументов. Связь между функцией IFERROR и функцией IF следующим образом:

```
IFERROR(A,B) := IF(ISERROR(A), B, A)
```

Значения, возвращаемые для A и B, должны иметь одинаковый тип данных; Таким образом, столбец или выражение, используемое для `value`, и значение, возвращаемое для `value_if_error`, должно быть одинаковым типом данных.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).
- Рекомендации по использованию IFERRORсм. в разделе Соответствующее использование функций ошибок.

Пример

В следующем примере возвращается значение 9999, если выражение 25/0 оценивается ошибкой. Если выражение возвращает значение, отличное от ошибки, это значение передается в вызывающее выражение.

DAX

```
= IFERROR(25/0,9999)
```

Связанное содержимое

логические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NOT

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Изменяет `FALSE` на `TRUE` или `TRUE` на `FALSE`.

Синтаксис

```
DAX

NOT(<logical>)
```

Параметры

Развернуть таблицу

Срок	Определение
`logical`	Значение или выражение, которое можно оценить как `TRUE` или `FALSE`.

Возвращаемое значение

`TRUE` OR `FALSE`.

Пример

В следующем примере извлекаются значения из вычисляемого столбца, созданного для иллюстрации функции IF. Например, вычисляемый столбец был назван с помощью имени по умолчанию, **вычисляемого столбца1**и содержит следующую формулу: `= IF([Orders]<300,"true","false")`

Формула проверяет значение в столбце [Заказы] и возвращает значение true, если число заказов составляет 300.

Теперь создайте вычисляемый столбец, **вычисляемый столбец 2**и введите следующую формулу.

```
```

DAX

```
= NOT([CalculatedColumn1])
```

Для каждой строки в **вычисляемого столбца1**значения "true" и "false" интерпретируются как логические значения `TRUE` или `FALSE`, а функция NOT возвращает логические противоположности этого значения.

Связанное содержимое

функция TRUEFALSE функцииIF

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

OR

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Проверяет, является ли один из аргументов `TRUE` возвращать `TRUE`. Функция возвращает `FALSE`, если оба аргумента `FALSE`.

Синтаксис

DAX

```
OR(<logical1>,<logical2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`logical_1, logical_2`	Логические значения, которые требуется проверить.

Возвращаемое значение

Логическое значение. Значение `TRUE`, если один из двух аргументов `TRUE`; значение FALSE, если оба аргумента `FALSE`.

Замечания

- Функция `OR` в DAX принимает только два аргумента (2). Если необходимо выполнить операцию OR для нескольких выражений, можно создать ряд вычислений или лучше использовать оператор OR (||) для объединения всех из них в более простом выражении.
- Функция вычисляет аргументы до первого `TRUE` аргумента, а затем возвращает `TRUE`.

Пример

В следующем примере показано, как использовать функцию OR для получения сотрудников по продажам, принадлежащих кругу превосходства. Круг превосходства признает тех, кто достиг более миллиона долларов в туринге велосипедов продаж или продаж более двух с половиной миллионов долларов в 2007 году.

DAX

```
IF(   OR(   CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),
'ProductSubcategory'[ProductSubcategoryName]="Touring Bikes") > 1000000
         ,   CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),
'DateTime'[CalendarYear]=2007) > 2500000
         )
   , "Circle of Excellence"
   , ""
   )
```

Возвращает

Развернуть таблицу

Метки строк	2005	2006	2007	2008	-	Общий итог
Аббас, Сид Э						
Альбертс, Эми Е						
Ансман-Вольф, Памела О						
Блит, Майкл G	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства
Кэмпбелл, Дэвид R						
Карсон, Джиллиан	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства
Ито, Шу K						
Цзян, Стивен Y						
Менса-Аннан, Тете A						
Митчелл, Линда C	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства
Пак, Джей Б	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства
Рейтер, Цви Майкл						

Метки строк	2005	2006	2007	2008	-	Общий итог
Сараива, Хосе Эдвалдо	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства
Цуфлияс, Линн Н						
Валдес, Рейчел Б						
Vargas, Garrett R						
Varkey Chudukatil, Ranjit R						Круг превосходства
Общий итог	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства	Круг превосходства

Связанное содержимое

логические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SWITCH

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Вычисляет выражение по списку значений и возвращает одно из нескольких возможных выражений результатов. Эту функцию можно использовать, чтобы избежать нескольких вложенных операторов IF.

Синтаксис

```
DAX

SWITCH(<expression>, <value>, <result>[, <value>, <result>]…[, <else>])
```

Параметры

Развернуть таблицу

Термин	Определение
`expression`	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).
`value`	Константное значение, соответствующее результатам `expression`.
`result`	Любое скалярное выражение, вычисляемое, если результаты `expression` соответствуют соответствующим `value`.
`else`	Любое скалярное выражение, которое необходимо оценить, если результат `expression` не соответствует ни одному из аргументов `value`.

Возвращаемое значение

Если имеется совпадение с `value`, возвращается скалярное значение из соответствующего `result`. Если нет совпадения с `value`, возвращается значение из `else`. Если ни один из `values` совпадений и `else` не указан, возвращается BLANK.

Замечания

- Вычисляемая `expression` может быть константным значением или выражением. Обычно эта функция используется для установки первого параметра на `TRUE`. См. приведенные ниже примеры.
- Все выражения `result` и выражение `else` должны иметь одинаковый тип данных.
- Порядок условий имеет значение. Как только возвращается один `value` совпадений, возвращается соответствующая `result`, а другие последующие `values` не оцениваются. Убедитесь, что наиболее строгие `values` для оценки указаны до менее строгих `values`. См. приведенные ниже примеры.

Примеры

Часто используется SWITCH для сравнения `expression` с константными `values`. В следующем примере создается вычисляемый столбец имен месяцев:

DAX

```
= SWITCH (
         [Month Number Of Year],
         1, "January",
         2, "February",
         3, "March",
         4, "April",
         5, "May",
         6, "June",
         7, "July",
         8, "August",
         9, "September",
         10, "October",
         11, "November",
         12, "December",
         "Unknown month number"
        )
```

Еще одним общим использованием SWITCH является замена нескольких вложенных операторов IF. Это достигается путем задания выражения для `TRUE`, как показано в следующем примере, который сравнивает уровень переупорядочения точки и безопасности акций на продуктах для выявления потенциальных рисков выхода из запасов:

DAX

```
= SWITCH (
        TRUE,
        [Reorder Point] > [Safety Stock Level], "Good: Safety stock level
```

```
exceeded",
        [Reorder Point] = [Safety Stock Level], "Minimal: Safety stock level
met",
        [Reorder Point] < [Safety Stock Level], "At risk: Safety stock level
not met",
        ISBLANK ( [Reorder Point] ), "Incomplete: Reorder point not set",
        ISBLANK ( [Safety Stock Level] ), "Incomplete: Safety stock level
not set",
        "Unknown"
        )
```

Порядок `values` вопросов. В следующем примере второй `result` никогда не возвращается, так как первое значение является менее строгим, чем второе. Результат в этом примере всегда `A` или `C`, но никогда не `B`.

DAX

```
= SWITCH (
         TRUE,
         Product[Standard Cost] < 100, "A",
         Product[Standard Cost] < 10, "B",
         "C"
         )
```

Следующая инструкция возвращает ошибку, так как типы данных в аргументах `result` отличаются. Помните, что типы данных во всех `result` и `else` аргументах должны быть одинаковыми.

DAX

```
= SWITCH (
        [Class],
        "L", "Large",
        "H", 0.1
        )
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TRUE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает логическое значение `TRUE`.

Синтаксис

```DAX
TRUE()
```

Возвращаемое значение

Всегда `TRUE`.

Замечания

Слово `TRUE` также интерпретируется как логическое значение `TRUE`.

Пример

Формула возвращает логическое значение `TRUE`, если значение в столбце "InternetSales_USD"[SalesAmount_USD], больше 200000.

```DAX
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

В следующей таблице показаны результаты, когда в отчете используется пример формулы с "ProductCategory"[ProductCategoryName] в метках строк и "DateTime"[CalendarYear] в метках столбцов.

Развернуть таблицу

Метки строк	2005	2006	2007	2008	-	Общий итог
Аксессуары	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
Велосипеды	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
Clothing	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Компоненты	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Общий итог	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE

Связанное содержимое

FALSE NOT IF

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Математические и триг-функции

Статья • 23.01.2025

Математические функции в выражениях анализа данных (DAX) очень похожи на математические и тригонометрические функции Excel. В этом разделе перечислены математические функции, предоставляемые DAX.

В этой категории

Развернуть таблицу

Функция	Описание
ABS	Возвращает абсолютное значение числа.
ACOS	Возвращает arccosine или обратное косинус числа.
ACOSH	Возвращает обратное гиперболическое косинус числа.
ACOT	Возвращает arccotangent или обратное котангентное число.
ACOTH	Возвращает обратное гиперболическое котангентное число.
ASIN	Возвращает арксин или инверсионный синус числа.
ASINH	Возвращает обратный гиперболический синус числа.
ATAN	Возвращает арктангентный или обратный тангенс числа.
ATANH	Возвращает обратный гиперболический тангенс числа.
CEILING	Округляет число до ближайшего целого числа или до ближайшего нескольких значений.
CONVERT	Преобразует выражение одного типа данных в другой.
COS	Возвращает косинус заданного угла.
COSH	Возвращает гиперболический косинус числа.
COT	Возвращает котангент угла, указанного в радианах.
COTH	Возвращает гиперболический котангент гиперболического угла.
CURRENCY	Вычисляет аргумент и возвращает результат в виде типа данных валюты.
DEGREES	Преобразует радианы в градусы.

Функция	Описание
DIVIDE	Выполняет деление и возвращает альтернативный результат или BLANK() на деление на 0.
EVEN	Возвращает число, округленное до ближайшего даже целого числа.
EXP	Возвращает значение e, возвращающее значение заданного числа.
FACT	Возвращает факториал числа, равный серии 1*2*3*** , заканчивая заданным числом.
FLOOR	Округляет число вниз, к нулю, до ближайшего нескольких значений.
GCD	Возвращает наибольший общий делитель двух или более целых чисел.
INT	Округляет число до ближайшего целого числа.
ISO.CEILING	Округляет число до ближайшего целого числа или до ближайшего нескольких значений.
LCM	Возвращает наименьшее число целых чисел.
LN	Возвращает естественный логарифм числа.
LOG	Возвращает логарифм числа в указанную базу.
LOG10	Возвращает логарифм базового-10 числа.
MOD	Возвращает оставшуюся часть после того, как число делится делителем. Результат всегда имеет тот же знак, что и делитель.
MROUND	Возвращает число, округленное до требуемого числа.
ODD	Возвращает число, округленное до ближайшего нечетного целого числа.
PI	Возвращает значение Pi, 3.14159265358979, точное до 15 цифр.
POWER	Возвращает результат числа, поднятого в силу.
QUOTIENT	Выполняет деление и возвращает только целочисленную часть результата деления.
RADIANS	Преобразует градусы в радианы.
RAND	Возвращает случайное число, больше или равное 0 и меньше 1, равномерно распределенное.
RANDBETWEEN	Возвращает случайное число в диапазоне между двумя указанными числами.
ROUND	Округляет число до указанного числа цифр.

Функция	Описание
ROUNDDOWN	Округляет число вниз, в сторону нуля.
ROUNDUP	Округляет число вверх, от 0 (ноль).
SIGN	Определяет знак числа, результат вычисления или значение в столбце.
SIN	Возвращает синус заданного угла.
SINH	Возвращает гиперболический синус числа.
SQRT	Возвращает квадратный корень числа.
SQRTPI	Возвращает квадратный корень (число * pi).
TAN	Возвращает тангенс заданного угла.
TANH	Возвращает гиперболический тангенс числа.
TRUNC	Усечение числа до целого числа путем удаления десятичного или дробного числа.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ABS

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает абсолютное значение числа.

Синтаксис

```DAX
ABS(<number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Число, для которого требуется абсолютное значение.

Возвращаемое значение

Десятичное число.

Замечания

Абсолютное значение числа — десятичное число, целое или десятичное без его знака. Функцию ABS можно использовать, чтобы гарантировать, что только не отрицательные числа возвращаются из выражений при вложенных в функциях, требующих положительного числа.

Пример

В следующем примере возвращается абсолютное значение разницы между ценовой категорией и ценой дилера, которую можно использовать в новом вычисляемом столбце, **ДилерMarkup**.

DAX

```
= ABS([DealerPrice]-[ListPrice])
```

Связанное содержимое

функции Math и TrigSIGN

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ACOS

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает arccosine или обратное косинус числа. Arccosine — это угол, косинус которого `number`. Возвращаемый угол задается в радианах в диапазоне 0 (ноль) на pi.

Синтаксис

DAX

```
ACOS(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`Number`	Косинус нужного угла и должен быть от -1 до 1.

Возвращаемое значение

Возвращает arccosine или обратное косинус числа.

Замечания

Если вы хотите преобразовать результат из радианов в градусы, умножьте его на 180/PI() или используйте функцию DEGREES.

Пример

Развернуть таблицу

Формула	Описание	Результат
`= ACOS(-0.5)`	Arccosine от -0,5 в радианах, 2*pi/3.	2.094395102
`= ACOS(-0.5)*180/PI()`	Arccosine от -0,5 в градусах.	120

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ACOSH

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает обратное гиперболическое косинус числа. Число должно быть больше или равно 1. Обратный гиперболический косинус является значением, гиперболическое косинус которого `number`, поэтому ACOSH(COSH(число)) равно числу.

Синтаксис

DAX

```
ACOSH(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Любое реальное число, равное или больше 1.

Возвращаемое значение

Возвращает обратное гиперболическое косинус числа.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`= ACOSH(1)`	Обратный гиперболический косинус 1.	0
`= ACOSH(10)`	Обратный гиперболический косинус 10.	2.993228

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ACOT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает основное значение arccotangent или обратное котангентное число.

Синтаксис

```DAX
ACOT(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`Number`	Косинус нужного угла. Должно быть реальным числом.

Возвращаемое значение

Одно десятичное значение.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ACOTH

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает обратное гиперболическое котангентное число.

Синтаксис

```DAX
ACOTH(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`Number`	Абсолютное значение number должно быть больше 1.

Возвращаемое значение

Одно десятичное значение.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ASIN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает арксин или инверсионный синус числа. Арксин — это угол, синус которого `number`. Возвращаемый угол задается в радианах в диапазоне -pi/2 до pi/2.

Синтаксис

DAX

```
ASIN(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Синус нужного угла и должен быть от -1 до 1.

Возвращаемое значение

Возвращает арксин или инверсионный синус числа.

Замечания

Чтобы выразить арксин в градусах, умножьте результат на 180/PI() или используйте функцию DEGREES.

Пример

Развернуть таблицу

Формула	Описание	Результат
`= ASIN(-0.5)`	Arcsine от -0,5 в радианах, -pi/6	-0.523598776
`= ASIN(-0.5)*180/PI()`	Арксин из -0,5 в градусах	-30
`= DEGREES(ASIN(-0.5))`	Арксин из -0,5 в градусах	-30

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ASINH

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает обратный гиперболический синус числа. Обратный гиперболический синус — это значение, гиперболический синус которого `number`, поэтому ASINH(SINH(число)) равно `number`.

Синтаксис

DAX

```
ASINH(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Любое реальное число.

Возвращаемое значение

Возвращает обратный гиперболический синус числа.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`= ASINH(-2.5)`	Обратный гиперболический синус -2,5	-1.647231146
`= ASINH(10)`	Обратный гиперболический синус 10	2.99822295

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ATAN

12.06.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает арктангентный или обратный тангенс числа. Арктангент — это угол, тангенс которого `number`. Возвращаемый угол задается в радианах в диапазоне -pi/2 до pi/2.

Синтаксис

```
DAX

ATAN(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Тангенс нужного угла.

Возвращаемое значение

Возвращает обратный тангенс числа.

Замечания

Чтобы выразить арктангент в градусах, умножьте результат на 180/PI() или используйте функцию DEGREES.

Пример

Развернуть таблицу

Формула	Описание	Результат
`= ATAN(1)`	Arctangent от 1 в радианах, pi/4	0.785398163

Формула	Описание	Результат
`= ATAN(1)*180/PI()`	Arctangent от 1 градусов	45

ATANH

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает обратный гиперболический тангенс числа. Число должно быть между -1 и 1 (за исключением -1 и 1). Обратный гиперболический тангенс является значением, гиперболический тангенс которого `number`, поэтому ATANH(TANH(число)) равно `number`.

Синтаксис

DAX

```
ATANH(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Любое реальное число от 1 до -1.

Возвращаемое значение

Возвращает обратный гиперболический тангенс числа.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`= ATANH(0.76159416)`	Обратный гиперболический тангенс 0,76159416	1.00000001
`= ATANH(-0.1)`		-0.100335348

Связанное содержимое

ATAN функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CEILING

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Округляет число до ближайшего целого числа или до ближайшего нескольких значений.

Синтаксис

```DAX
CEILING(<number>, <significance>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Число, которое нужно округить или ссылку на столбец, содержащий числа.
`significance`	Несколько значений, к которым вы хотите округлить. Например, для округления до ближайшего целого числа введите 1.

Возвращаемое значение

Число округлено, как указано.

Замечания

- В CEILINGесть две функции DAX с следующими различиями:
  - Функция CEILING эмулирует поведение функции CEILING в Excel.
  - Функция ISO.CEILING следует стандарту ISO для определения значения потолка.
- Две функции возвращают одно и то же значение для положительных чисел, но разные значения отрицательных чисел. При использовании положительного значения как CEILING, так и ISO.CEILING округление

отрицательных чисел вверх (к положительной бесконечности). При использовании отрицательного значения CEILING округляет отрицательные числа вниз (к отрицательной бесконечности), а ISO.CEILING округляет отрицательные числа вверх (к положительной бесконечности).

- Возвращаемый тип обычно имеет тот же тип значительного аргумента, что и следующие исключения:
  - Если тип аргумента число является валютой, возвращаемый тип является валютой.
  - Если тип аргумента значения является логическим, возвращаемый тип является целым числом.
  - Если тип аргумента значения не является числовым, возвращаемый тип является реальным.

Пример 1

Следующая формула возвращает значение 4.45. Это может быть полезно, если вы хотите избежать использования небольших единиц в ценах. Если существующий продукт стоит на $ 4,42, вы можете использовать CEILING для округления цен до ближайшей единицы пяти центов.

DAX

```
= CEILING(4.42,0.05)
```

Пример 2

Следующая формула возвращает аналогичные результаты, как предыдущий пример, но использует числовые значения, хранящиеся в столбце, **ProductPrice**.

DAX

```
= CEILING([ProductPrice],0.05)
```

Связанное содержимое

функции Math и TrigFLOOR функцииISO.CEILINGROUNDUP функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CONVERT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует выражение одного типа данных в другой.

Синтаксис

DAX

```
CONVERT(<Expression>, <Datatype>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Expression`	Любое допустимое выражение.
`Datatype`	Перечисление, которое включает в себя: INTEGER(Целое число), DOUBLE(десятичное число), STRING(Text), BOOLEAN(True/False), CURRENCY(фиксированное десятичное число), DATETIME(дата, время и т. д.).

Возвращаемое значение

Возвращает значение `Expression`, преобразованное в `Datatype`.

Замечания

- Функция возвращает ошибку, если значение не может быть преобразовано в указанный тип данных.
- DAX вычисляемые столбцы должны иметь один тип данных. Так как функции MEDIAN и MEDIANX для целочисленного столбца возвращают смешанные типы данных(целое или двойное), следующее вычисляемое выражение столбца возвращает ошибку в результате: `MedianNumberCarsOwned = MEDIAN(DimCustomer[NumberCarsOwned])`.

- Чтобы избежать смешанных типов данных, измените выражение, чтобы всегда возвращать двойной тип данных, например, `MedianNumberCarsOwned = MEDIANX(DimCustomer, CONVERT([NumberCarsOwned], DOUBLE))`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

запрос DAX

```
DAX

EVALUATE { CONVERT(DATE(1900, 1, 1), INTEGER) }
```

Возвращает

Развернуть таблицу

[значение]
2

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COS

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает косинус заданного угла.

Синтаксис

```
DAX
COS(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Обязательно. Угол в радианах, для которого требуется косинус.

Возвращаемое значение

Возвращает косинус заданного угла.

Замечания

Если угол находится в градусах, умножьте угол на PI()/180 или используйте функцию RADIANS для преобразования угла в радианы.

Пример

Развернуть таблицу

Формула	Описание	Результат
`= COS(1.047)`	Косинус 1.047 радиан	0.5001711

Формула	Описание	Результат
`= COS(60*PI()/180)`	Косинус 60 градусов	0.5
`= COS(RADIANS(60))`	Косинус 60 градусов	0.5

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COSH

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает гиперболический косинус числа.

Синтаксис

```
DAX
COSH(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Обязательно. Любое реальное число, для которого вы хотите найти гиперболический косинус.

Возвращаемое значение

Гиперболический косинус числа.

Замечания

- Формула гиперболического косина:

$$\text{COSH}(z) = \frac{e^z + e^{-z}}{2}$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

выражение DAX	Описание	Результат
`= COSH(4)`	Гиперболический косинус 4	27.308233
`= COSH(EXP(1))`	Гиперболический косинус основы естественного логарифма.	7.6101251

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает котангент угла, указанного в радианах.

Синтаксис

```
DAX

COT (<number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Угол в радианах, для которого требуется котангент.

Возвращаемое значение

Котангент заданного угла.

Замечания

- Абсолютное значение числа должно быть меньше 2^27 и не может иметь значение 0.
- Если число выходит за рамки ограничений, возвращается ошибка.
- Если число является нечисленным значением, возвращается ошибка.

Пример

Следующий запрос DAX

DAX

```
EVALUATE { COT(30) }
```

Возвращает

Развернуть таблицу

[значение]
-0.156119952161659

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COTH

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает гиперболический котангенс гиперболического угла.

Синтаксис

```
DAX

COTH (<number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Гиперболический угол в радианах, для которого требуется гиперболический котангенс.

Возвращаемое значение

Гиперболический котангенс заданного угла.

Замечания

- Гиперболический котангенс является аналогом обычного (циклического) котангенса.
- Абсолютное значение числа должно быть меньше 2^{27} и не может быть 0.
- Если число выходит за пределы ограничений, возвращается ошибка.
- Если число является нечисленным значением, возвращается ошибка.
- Используется следующее уравнение:

$$\mathrm{COTH}(N) = \frac{1}{\mathrm{TANH}(N)} = \frac{\mathrm{COSH(N)}}{\mathrm{SINH(N)}} = \frac{e^N + e^{-N}}{e^N - e^{-N}}$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX

```
DAX

EVALUATE { COTH(2) }
```

Возвращает

Развернуть таблицу

[значение]
1.03731472072755

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CURRENCY

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Вычисляет аргумент и возвращает результат в виде типа данных валюты.

Синтаксис

```
DAX

CURRENCY(<value>)
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение должно оцениваться ровно один раз перед всеми остальными операциями.

Возвращаемое значение

Значение выражения, вычисляемое и возвращаемое как значение типа валюты.

Замечания

- Функция CURRENCY округляет 5-й значимый десятичный знак в значении, чтобы вернуть 4-ю десятичную цифру. Округление происходит, если 5-й значимый десятичный знак равен или больше 5. Например, если значение равно 3,66666666666666, то преобразование в валюту возвращает \3,6667 долл. США. Однако если значение равно 3.0123456789, то преобразование в валюту возвращает \3,0123 долл. США.
- Если тип данных выражения имеет значение TrueFalse, то CURRENCY(`TrueFalse`) возвращает $1,0000 для значений True и $0,0000 для

значений False.

- Если тип данных выражения — Text, то CURRENCY(`Text`) попытается преобразовать текст в число. Если преобразование завершается успешно, число будет преобразовано в валюту, в противном случае возвращается ошибка.

- Если тип данных выражения — DateTime, то CURRENCY(`DateTime`) преобразует значение даты и времени в число, а это число в валюту. Значения DateTime имеют целочисленную часть, представляющую количество дней между заданной датой и 1900-03-01 и дробной частью, представляющей долю дня (где 12 часов или полдень составляет 0,5 дня). Если значение выражения не является правильным значением DateTime, возвращается ошибка.

Пример

Преобразуйте номер 1234.56 в тип данных валюты.

```
DAX
= CURRENCY(1234.56)
```

Возвращает значение 1234.56000.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DEGREES

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует радианы в градусы.

Синтаксис

```
DAX
DEGREES(angle)
```

Параметры

Развернуть таблицу

Срок	Определение
`angle`	Обязательно. Угол в радианах, которые требуется преобразовать.

Пример

Развернуть таблицу

Формула	Описание	Результат
`= DEGREES(PI())`	Градусы пи радиан	180

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DIVIDE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Выполняет деление и возвращает альтернативный результат или BLANK() на деление на 0.

Синтаксис

DAX

```
DIVIDE(<numerator>, <denominator> [,<alternateresult>])
```

Параметры

Развернуть таблицу

Срок	Определение
`numerator`	Дивиденды или число, которые необходимо разделить.
`denominator`	Делитель или число, на которые нужно разделить.
`alternateresult`	(Необязательно) Значение, возвращаемое при делении на ноль, приводит к ошибке. Если это не указано, значение по умолчанию BLANK().

Возвращаемое значение

- Десятичное число.

Замечания

- Альтернативный результат на деление на 0 должен быть константой.
- Рекомендации по использованию DIVIDEсм. в разделе DIVIDE функции и оператора деления (/) в DAX.

Пример

В следующем примере возвращается значение 2.5.

```
DAX
= DIVIDE(5,2)
```

Пример 1

В следующем примере возвращается BLANK.

```
DAX
= DIVIDE(5,0)
```

Пример 2

В следующем примере возвращается значение 1.

```
DAX
= DIVIDE(5,0,1)
```

Связанное содержимое

QUOTIENT функциифункции Math и Trig

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EVEN

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает число, округленное до ближайшего даже целого числа. Эту функцию можно использовать для обработки элементов, поступающих в два. Например, ячейка упаковки принимает строки одного или двух элементов. Скорость заполнена, когда количество элементов округляется до ближайшего двух, соответствует емкости ящика.

Синтаксис

```DAX
EVEN(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Значение для округления.

Возвращаемое значение

Возвращает число, округленное до ближайшего даже целого числа.

Замечания

- Если `number` не является числовыми, EVEN возвращает значение ошибки `#VALUE!`.
- Независимо от знака числа, значение округляется при корректировке от нуля. Если число является даже целым числом, округление не происходит.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`= EVEN(1.5)`	Округляет 1,5 до ближайшего даже целого числа	2
`= EVEN(3)`	Округляет 3 до ближайшего даже целого числа	4
`= EVEN(2)`	Округляет 2 до ближайшего даже целого числа	2
`= EVEN(-1)`	Округляет -1 до ближайшего даже целого числа	-2

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EXP

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает значение e, возвращающее значение заданного числа. Константа e равна 2.71828182845904, основа естественного логарифма.

Синтаксис

```DAX
EXP(<number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Экспонент, примененный к базовому e. Константа e равна 2.71828182845904, основа естественного логарифма.

Возвращаемое значение

Десятичное число.

Исключения

Замечания

- EXP является обратным LN, который является естественным логарифмом заданного числа.
- Чтобы вычислить полномочия баз, отличных от e, используйте оператор экспонентации (^). Дополнительные сведения см. в DAX справочнике по оператору.

Пример

Следующая формула вычисляет e, поднятую до значения числа, содержащегося в столбце, `[Power]`.

```
DAX

= EXP([Power])
```

Связанное содержимое

функции Math и TrigLN функцииEXPLOG функцииLOG

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FACT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает факториал числа, равный серии 1*2*3*** , заканчивая заданным числом.

Синтаксис

DAX

```
FACT(<number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Неотрицательное число, для которого требуется вычислить факториал.

Возвращаемое значение

Десятичное число.

Замечания

- Если число не является целым числом, оно усечено и возвращается ошибка. Если результат слишком велик, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула возвращает факториал для ряда целых чисел в столбце `[Values]`.

```
DAX
= FACT([Values])
```

В следующей таблице показаны ожидаемые результаты:

Развернуть таблицу

Значения	Результаты
0	1
1	1
2	2
3	6
4	24
5	120
170	7.257415615308E+306

Связанное содержимое

функции Math и TrigTRUNC

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FLOOR

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Округляет число вниз, к нулю, до ближайшего нескольких значений.

Синтаксис

```DAX
FLOOR(<number>, <significance>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Числовое значение, которое требуется округлить.
`significance`	Число, к которому требуется округление. Аргументы `number` и `significance` должны быть положительными или отрицательными.

Возвращаемое значение

Десятичное число.

Замечания

- Если любой аргумент не является числовыми, FLOOR возвращает значение ошибки `#VALUE!`.
- Если число и значение имеют разные признаки, FLOOR возвращает значение ошибки `#NUM!`.
- Независимо от знака числа, значение округляется по нулю. Если число является точным кратным значением, округление не происходит.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула принимает значения в столбце [Total Product Cost] из таблицы, InternetSales и округляется до ближайшего числа 1.

```
DAX
= FLOOR(InternetSales[Total Product Cost],.1)
```

В следующей таблице показаны ожидаемые результаты для некоторых примеров значений:

Развернуть таблицу

Значения	Ожидаемый результат
10.8423	10.8
8.0373	8
2.9733	2.9

Связанное содержимое

функции Math и Trig

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

GCD

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает наибольший общий делитель двух или более целых чисел. Наибольший общий делитель — это наибольшее целое число, которое делит как число1, так и число2 без остатка.

Синтаксис

```
DAX

GCD(number1, [number2], ...)
```

Параметры

Развернуть таблицу

Срок	Определение
`number1, number2, ...`	Число1 является обязательным, последующие числа являются необязательными. От 1 до 255 значений. Если любое значение не является целым числом, оно усечено.

Возвращаемое значение

Величайший общий делитель двух или более целых чисел.

Замечания

- Если какой-либо аргумент не является числовыми, GCD возвращает значение ошибки `#VALUE!`.
- Если любой аргумент меньше нуля, GCD возвращает значение ошибки `#NUM!`.
- Один разделяет любое значение равномерно.
- Простое число имеет только себя и один, как даже делители.

- Если параметр GCD равен >=2^53, GCD возвращает значение ошибки `#NUM!`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

выражение DAX	Описание	Результат
`= GCD(5, 2)`	Величайший общий делитель 5 и 2.	1
`= GCD(24, 36)`	Величайший общий делитель 24 и 36.	12
`= GCD(7, 1)`	Величайший общий делитель 7 и 1.	1

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

INT

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Округляет число до ближайшего целого числа.

Синтаксис

```
DAX
INT(<number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Число, которое нужно округить до целого числа

Возвращаемое значение

Целое число.

Замечания

TRUNC и INT похожи в том, что оба возвращают целые числа. TRUNC удаляет дробную часть числа. INT округляет числа до ближайшего целого числа в зависимости от значения дробной части числа. INT и TRUNC отличаются только при использовании отрицательных чисел: `TRUNC(-4.3)` возвращает -4, но `INT(-4.3)` возвращает -5, так как -5 меньшее число.

Пример

Следующее выражение округляет значение до 1. Если вы используете функцию ROUND, результатом будет 2.

DAX

```
= INT(1.5)
```

Связанное содержимое

функции Math и TrigROUND функцииROUNDUPROUNDDOWN функцииMROUND

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ISO.CEILING

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Округляет число до ближайшего целого числа или до ближайшего нескольких значений.

Синтаксис

DAX

```
ISO.CEILING(<number>[, <significance>])
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Число, которое нужно округить или ссылку на столбец, содержащий числа.
`significance`	(необязательно) Несколько значений, к которым вы хотите округлить. Например, для округления до ближайшего целого числа введите 1. Если единица значения не указана, число округляется до ближайшего целого числа.

Возвращаемое значение

Число того же типа, что и аргумент `number`, округленное, как указано.

Замечания

В CEILINGесть две функции DAX с следующими различиями:

- Функция CEILING эмулирует поведение функции CEILING в Excel.
- Функция ISO.CEILING следует стандарту ISO для определения значения потолка.

Две функции возвращают одно и то же значение для положительных чисел, но разные значения отрицательных чисел. При использовании положительного значения как CEILING, так и ISO.CEILING округление отрицательных чисел вверх (к положительной бесконечности). При использовании отрицательного значения CEILING округляет отрицательные числа вниз (к отрицательной бесконечности), а ISO.CEILING округляет отрицательные числа вверх (к положительной бесконечности).

Тип результата обычно совпадает с типом значения, используемого в качестве аргумента со следующими исключениями:

- Если первый аргумент имеет тип валюты, результат будет типом валюты.
- Если необязательный аргумент не включен в результат целочисленного типа.
- Если аргумент значения имеет логический тип, результатом является целочисленный тип.
- Если аргумент значения является нечисленным, результатом является реальный тип.

Пример: положительные числа

Следующая формула возвращает значение 4.45. Это может быть полезно, если вы хотите избежать использования небольших единиц в ценах. Если существующий продукт стоит на $ 4,42, вы можете использовать ISO.CEILING для округления цен до ближайшей единицы пяти центов.

```
DAX
= ISO.CEILING(4.42,0.05)
```

Пример: отрицательные числа

Следующая формула возвращает значение потолочного значения ISO -4,40.

```
DAX
= ISO.CEILING(-4.42,0.05)
```

Связанное содержимое

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LCM

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает наименьшее число целых чисел. Наименьшее число является наименьшим положительным целым числом, которое является несколькими из всех целых аргументов number1, number2 и т. д. Используйте LCM для добавления дробей с разными знаменателями.

Синтаксис

DAX

```
LCM(number1, [number2], ...)
```

Параметры

Развернуть таблицу

Срок	Определение
`number1, number2,...`	Число1 является обязательным, последующие числа являются необязательными. От 1 до 255 значений, для которых требуется наименьшее общее несколько. Если значение не является целым числом, оно усечено.

Возвращаемое значение

Возвращает наименьшее число целых чисел.

Замечания

- Если какой-либо аргумент не является числовыми, LCM возвращает значение ошибки `#VALUE!`.
- Если любой аргумент меньше нуля, LCM возвращает значение ошибки `#NUM!`.
- Если LCM(a,b) >=2^53, LCM возвращает значение ошибки `#NUM!`.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`= LCM(5, 2)`	Наименее распространено несколько из 5 и 2.	10
`= LCM(24, 36)`	Наименее распространено несколько из 24 и 36.	72

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает естественный логарифм числа. Естественные логарифмы основаны на константе e (2.71828182845904).

Синтаксис

DAX

```
LN(<number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Положительное число, для которого требуется естественный логарифм.

Возвращаемое значение

Десятичное число.

Замечания

LN является обратной функцией EXP.

Пример

В следующем примере возвращается естественный логарифм числа в столбце `[Values]`.

DAX

```
= LN([Values])
```

Связанное содержимое

функции Math и TrigEXP

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LOG

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает логарифм числа в указанную базу.

Синтаксис

```
DAX
LOG(<number>,<base>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Положительное число, для которого требуется логарифм.
`base`	Основа логарифма. Если опущено, база составляет 10.

Возвращаемое значение

Десятичное число.

Замечания

Если отображаемое значение слишком велико, может появиться сообщение об ошибке.

Функция LOG10 аналогична, но всегда возвращает общий логарифм, то есть логарифм для базового 10.

Пример

Следующие формулы возвращают тот же результат, 2.

DAX

```
= LOG(100,10)
= LOG(100)
= LOG10(100)
```

Связанное содержимое

функции Math и TrigEXP функцииLOGLOG функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LOG10

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает логарифм базового-10 числа.

Синтаксис

```
DAX

LOG10(<number>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Положительное число, для которого требуется логарифм base-10.

Возвращаемое значение

Десятичное число.

Замечания

Функция LOG позволяет изменить базу логарифма, а не использовать базовую 10.

Пример

Следующие формулы возвращают тот же результат, 2:

```
DAX

= LOG(100,10)
= LOG(100)
= LOG10(100)
```

Связанное содержимое

функции Math и TrigEXP функцииLOGLOG функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MOD

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает оставшуюся часть после того, как число делится делителем. Результат всегда имеет тот же знак, что и делитель.

Синтаксис

```DAX
MOD(<number>, <divisor>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Число, для которого нужно найти оставшуюся часть после выполнения деления.
`divisor`	Число, по которому нужно разделить.

Возвращаемое значение

Целое число.

Замечания

- Если разделитель равен 0 (ноль), MOD возвращает ошибку. Нельзя разделить на 0.
- Функция MOD может быть выражена с точки зрения функции INT: MOD(n, d) = n - d*INT(n/d)

Пример 1

Следующая формула возвращает 1, оставшуюся часть 3, разделенную на 2.

```
DAX
= MOD(3,2)
```

Пример 2

Следующая формула возвращает значение -1, оставшаяся часть 3, разделенная на 2. Обратите внимание, что знак всегда совпадает с знаком делителя.

```
DAX
= MOD(-3,-2)
```

Связанное содержимое

функции Math и TrigROUND функцииROUNDUP функцииROUNDDOWNMROUND функцииINT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MROUND

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает число, округленное до требуемого числа.

Синтаксис

```
DAX

MROUND(<number>, <multiple>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Число для округления.
`multiple`	Несколько значений, к которым требуется округление числа.

Возвращаемое значение

Десятичное число.

Замечания

MROUND округляется вверх, от нуля, если оставшаяся часть деления `number` по указанному `multiple` больше или равно половине значения `multiple`.

Пример: десятичные разряды

Следующее выражение округляет 1.3 до ближайшего числа .2. Ожидаемый результат — 1.4.

```
DAX
```

```
= MROUND(1.3,0.2)
```

Пример: отрицательные числа

Следующее выражение округляет -10 до ближайшего числа -3. Ожидаемый результат равен -9.

DAX

```
= MROUND(-10,-3)
```

Пример: ошибка

Следующее выражение возвращает ошибку, так как числа имеют разные признаки.

DAX

```
= MROUND(5,-2)
```

Связанное содержимое

функции Math и TrigROUND функцииROUNDUP функцииROUNDDOWNMROUND функцииINT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ODD

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает число, округленное до ближайшего нечетного целого числа.

Синтаксис

DAX

```
ODD(number)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Обязательно. Значение для округления.

Возвращаемое значение

Возвращает число, округленное до ближайшего нечетного целого числа.

Замечания

- Если `number` не является числовыми, ODD возвращает значение ошибки `#VALUE!`.
- Независимо от знака числа, значение округляется при корректировке от нуля. Если число является нечетным целым числом, округление не происходит.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`= ODD(1.5)`	Округляет до ближайшего нечетного целого числа до 1,5.	3
`= ODD(3)`	Округляет 3 до ближайшего нечетного целого числа.	3
`= ODD(2)`	Округляет 2 до ближайшего нечетного целого числа.	3
`= ODD(-1)`	Округляет -1 до ближайшего нечетного целого числа.	-1
`= ODD(-2)`	Округляет -2 вверх (от 0) до ближайшего нечетного целого числа.	-3

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PI

Статья • 21.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает значение Pi, 3.14159265358979, точное до 15 цифр.

Синтаксис

```DAX
PI()
```

Возвращаемое значение

Десятичное число со значением Pi, 3.14159265358979, точное до 15 цифр.

Замечания

Pi является математической константой. В DAX, Pi представлен как реальное число, точное до 15 цифр, то же самое, что и Excel.

Пример

Следующая формула вычисляет область круга, заданного радиусом в столбце, `[Radius]`.

```DAX
= PI()*([Radius]*2)
```

Связанное содержимое

функции Math и Trig

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

POWER

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает результат числа, поднятого в силу.

Синтаксис

DAX

```
POWER(<number>, <power>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Базовое число, которое может быть любым реальным числом.
`power`	Экспонент, к которому вызывается базовое число.

Возвращаемое значение

Десятичное число.

Пример

В следующем примере возвращается 25.

DAX

```
= POWER(5,2)
```

Связанное содержимое

функции Math и Trig

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

QUOTIENT

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Выполняет деление и возвращает только целочисленную часть результата деления. Используйте эту функцию, если вы хотите отменить оставшуюся часть деления.

Синтаксис

```DAX
QUOTIENT(<numerator>, <denominator>)
```

Параметры

Развернуть таблицу

Срок	Определение
`numerator`	Дивиденды или число для деления.
`denominator`	Делитель или число, на которые нужно разделить.

Возвращаемое значение

Целое число.

Замечания

- Если любой из аргументов не является числовым, QUOTIENT возвращает значение ошибки `#VALUE!`.
- Для любого аргумента можно использовать ссылку на столбец вместо литерального значения. Однако если столбец, на который ссылается ссылка, содержит значение 0 (ноль), возвращается ошибка для всего столбца значений.

Пример

Следующие формулы возвращают тот же результат, 2.

```
DAX
```

```
= QUOTIENT(5,2)
```

```
DAX
```

```
= QUOTIENT(10/2,2)
```

Связанное содержимое

функции Math и Trig

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RADIANS

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует градусы в радианы.

Синтаксис

DAX

```
RADIANS(angle)
```

Параметры

Развернуть таблицу

Срок	Определение
`angle`	Обязательно. Угол в градусах, которые требуется преобразовать.

Пример

Развернуть таблицу

Формула	Описание	Результат
`= RADIANS(270)`	270 градусов как радианы (4,712389 или 3π/2 радианы)	4.712389

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RAND

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает случайное число, больше или равное 0 и меньше 1, равномерно распределенное. Число, возвращаемое при каждом пересчете ячейки, содержащей эту функцию.

Синтаксис

```DAX
RAND()
```

Возвращаемое значение

Десятичное число.

Замечания

- Пересчет зависит от различных факторов, в том числе от того, установлена ли модель `Manual` или `Automatic` режим пересчета, а также от того, обновлены ли данные.
- RAND и другие переменные функции, не имеющие фиксированных значений, не всегда пересчитываются. Например, выполнение запроса или фильтрации обычно не приводит к повторной оценке таких функций. Однако результаты этих функций будут пересчитываться при перерасчете всего столбца. Эти ситуации включают обновление из внешнего источника данных или ручного редактирования данных, что приводит к повторной оценке формул, содержащих эти функции.
- RAND всегда пересчитывается, если функция используется в определении меры.
- RAND функция не может возвращать результат нуля, чтобы предотвратить такие ошибки, как деление на ноль.

Примеры

Чтобы создать случайное реальное число между двумя другими числами, используйте:

```
DAX
= RAND()*(b-a)+a
```

Чтобы создать случайное число, превышающее 0 и менее 1, выполните приведенные ниже действия.

```
DAX
= RAND()
```

Создание случайного числа больше 0 и менее 100

```
DAX
= RAND()*100
```

Создание случайного целого числа больше 0 и менее 100

```
DAX
INT(RAND()*100)
```

Связанное содержимое

функции Math и Trigстатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RANDBETWEEN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает случайное число в диапазоне между двумя указанными числами.

Синтаксис

```DAX
RANDBETWEEN(<bottom>,<top>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Bottom`	Наименьшее целое число, возвращаемое функцией.
`Top`	Наибольшее целое число, возвращаемое функцией.

Возвращаемое значение

Целое число.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула возвращает случайное число от 1 до 10.

```DAX
```

```
= RANDBETWEEN(1,10)
```

Связанное содержимое

функции Math и Trigстатистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ROUND

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Округляет число до указанного числа цифр.

Синтаксис

```DAX
ROUND(<number>, <num_digits>)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Число, которое требуется округлить.
`num_digits`	Число цифр, к которым требуется округление. Отрицательное значение округляет цифры слева от десятичной запятой; значение нулевых округляется до ближайшего целого числа.

Возвращаемое значение

Десятичное число.

Замечания

- Если `num_digits` больше 0 (ноль), число округляется до указанного числа десятичных разрядов.
- Если значение `num_digits` равно 0, число округляется до ближайшего целого числа.
- Если `num_digits` меньше 0, число округляется слева от десятичной запятой.

- Связанные функции
  - Чтобы всегда округлить (от нуля), используйте функцию ROUNDUP.
  - Чтобы всегда округлить (к нулю), используйте функцию ROUNDDOWN.
  - Чтобы округить число до определенного числа (например, округление до ближайшего числа 0,5), используйте функцию MROUND.
  - Используйте функции TRUNC и INT для получения целочисленной части числа.

Пример 1

Следующая формула округляет 2,15 до одного десятичного разряда. Ожидаемый результат — 2.2.

DAX

```
= ROUND(2.15,1)
```

Пример 2

Следующая формула округляет 21,5 до одного десятичного знака слева от десятичной точки. Ожидаемый результат — 20.

DAX

```
= ROUND(21.5,-1)
```

Связанное содержимое

функции Math и TrigROUNDROUNDDOWNMROUNDINTTRUNC

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ROUNDDOWN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Округляет число вниз, в сторону нуля.

Синтаксис

DAX

```
ROUNDDOWN(<number>, <num_digits>)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Реальное число, которое требуется округлить.
`num_digits`	Число цифр, к которым требуется округление. Отрицательные округления слева от десятичной запятой; ноль до ближайшего целого числа.

Возвращаемое значение

Десятичное число.

Замечания

- Если `num_digits` больше 0 (ноль), то значение в `number` округляется до указанного числа десятичных разрядов.
- Если `num_digits` равно 0, то значение в `number` округляется до ближайшего целого числа.
- Если `num_digits` меньше 0, то значение в `number` округляется до левой части десятичной запятой.

- ROUNDDOWN ведет себя как ROUND, за исключением того, что он всегда округляет число вниз. Функция INT также округляется, но при INT результат всегда является целым числом, в то время как с ROUNDDOWN можно управлять точностью результата.

Пример 1

В следующем примере округляется 3.14159 до трех десятичных разрядов. Ожидаемый результат — 3.141.

```
DAX

= ROUNDDOWN(3.14159,3)
```

Пример 2

В следующем примере округляется значение 31415,92654 до 2 десятичных разрядов слева от десятичного разряда. Ожидаемый результат — 31400.

```
DAX

= ROUNDDOWN(31415.92654, -2)
```

Связанное содержимое

функции Math и TrigROUNDROUNDUPROUNDDOWNMROUNDINT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ROUNDUP

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Округляет число вверх, от 0 (ноль).

Синтаксис

```
DAX
ROUNDUP(<number>, <num_digits>)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Реальное число, которое требуется округление.
`num_digits`	Число цифр, к которым требуется округление. Отрицательное значение для `num_digits` округляется слева от десятичной запятой; Если `num_digits` равно нулю или опущено, `number` округляется до ближайшего целого числа.

Возвращаемое значение

Десятичное число.

Замечания

- Если `num_digits` больше 0 (ноль), то число округляется до указанного числа десятичных разрядов.
- Если значение `num_digits` равно 0, число округляется до ближайшего целого числа.
- Если `num_digits` меньше 0, число округляется до левой части десятичной запятой.

- ROUNDUP ведет себя как ROUND, за исключением того, что он всегда округляет число вверх.

Пример

Следующая формула округляет Pi до четырех десятичных разрядов. Ожидаемый результат — 3.1416.

```
DAX

= ROUNDUP(PI(),4)
```

Пример: десятичные разряды в качестве второго аргумента

Следующая формула округляет 1,3 до ближайшего числа 0,2. Ожидаемый результат равен 2.

```
DAX

= ROUNDUP(1.3,0.2)
```

Пример: отрицательное число в качестве второго аргумента

Следующая формула округляет значение в столбце **FreightCost**с ожидаемыми результатами, показанными в следующей таблице:

```
DAX

= ROUNDUP([Values],-1)
```

Если `num_digits` меньше нуля, число мест слева от десятичного знака увеличивается на указанное значение.

Развернуть таблицу

FreightCost	Ожидаемый результат
13.25	20
2.45	10
25.56	30
1,34	10
345.01	350

Связанное содержимое

функции Math и TrigROUNDROUNDDOWNMROUNDINT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SIGN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Определяет знак числа, результат вычисления или значение в столбце. Функция возвращает значение 1, если число положительное, 0 (ноль), если число равно нулю или -1, если число отрицательное.

Синтаксис

```DAX
SIGN(<number>)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Любое реальное число, столбец, содержащий числа или выражение, которое оценивается числом.

Возвращаемое значение

целое число. Возможные значения Return: 1, 0 и -1.

Развернуть таблицу

Возвращаемое значение	Description
1	Число положительное
0	Число равно нулю
-1	Число отрицательное

Пример

Следующая формула возвращает знак результата выражения, вычисляющего цену продажи минус затрат.

```DAX
= SIGN( ([Sale Price] - [Cost]) )
```

Связанное содержимое

функции Math и Trig

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SIN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает синус заданного угла.

Синтаксис

```DAX
SIN(number)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Обязательный. Угол в радианах, для которого требуется синус.

Возвращаемое значение

Возвращает синус заданного угла.

Замечания

Если аргумент находится в градусах, умножьте его на `PI()/180` или используйте функцию RADIANS для преобразования его в радианы.

Пример

Развернуть таблицу

Формула	Description	Результат
`= SIN(PI())`	Синус пи радиан (0, приблизительно).	0,0

Формула	Description	Результат
`= SIN(PI()/2)`	Синус пи/2 радианы.	1.0
`= SIN(30*PI()/180)`	Синус из 30 градусов.	0,5
`= SIN(RADIANS(30))`	Синус из 30 градусов.	0,5

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SINH

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает гиперболический синус числа.

Синтаксис

DAX

```
SINH(number)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Обязательный. Любое реальное число.

Возвращаемое значение

Возвращает гиперболический синус числа.

Замечания

- Формула гиперболического синуса:

$$\text{SINH}(z) = \frac{e^z - e^{-z}}{2}$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Вероятность получения результата менее 1,03 секунды.

```
DAX
```

```
= 2.868*SINH(0.0342\*1.03)
```

Возвращает значение 0.1010491

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SQRT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает квадратный корень числа.

Синтаксис

```DAX
SQRT(<number>)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Число, для которого требуется квадратный корень, столбец, содержащий числа или выражение, которое оценивается в число.

Возвращаемое значение

Десятичное число.

Замечания

Если число отрицательное, функция SQRT возвращает ошибку.

Пример

Следующая формула:

```DAX
= SQRT(25)
```

Связанное содержимое

функции Math и Trig

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SQRTPI

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает квадратный корень (число * pi).

Синтаксис

```
DAX

SQRTPI(number)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Обязательный. Число, на которое умножается пи.

Возвращаемое значение

Возвращает квадратный корень (число * pi).

Пример

Развернуть таблицу

Формула	Description	Результат
`= SQRTPI(1)`	Квадратный корень pi.	1.772454
`= SQRTPI(2)`	Квадратный корень 2 * pi.	2.506628

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TAN

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает тангенс заданного угла.

Синтаксис

DAX

```
TAN(number)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Обязательный. Угол в радианах, для которого требуется тангенс.

Возвращаемое значение

Возвращает тангенс заданного угла.

Замечания

Если аргумент находится в градусах, умножьте его на PI()/180 или используйте функцию RADIANS для преобразования его в радианы.

Пример

Развернуть таблицу

Формула	Description	Результат
`= TAN(0.785)`	Тангенс из 0,785 радиан (0,99920)	0.99920

Формула	Description	Результат
`= TAN(45*PI()/180)`	Тангенс из 45 градусов (1)	1
`= TAN(RADIANS(45))`	Тангенс из 45 градусов (1)	1

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TANH

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает гиперболический тангенс числа.

Синтаксис

```
DAX

TANH(number)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Обязательный. Любое реальное число.

Возвращаемое значение

Возвращает гиперболический тангенс числа.

Замечания

- Формула гиперболического тангента:

$$\text{TANH}(z) = \frac{\text{SINH}(z)}{\text{COSH}(z)}$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Description	Результат
`= TANH(-2)`	Гиперболический тангенс -2 (-0,96403)	-0.964028
`= TANH(0)`	Гиперболический тангенс 0 (0)	0
`= TANH(0.5)`	Гиперболический тангенс 0,5 (0.462117)	0.462117

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TRUNC

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Усечение числа до целого числа путем удаления десятичного или дробного числа.

Синтаксис

DAX

```
TRUNC(<number>,<num_digits>)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Число, которое требуется усечь.
`num_digits`	Число, указывающее точность усечения; Значение 0 (ноль)

Возвращаемое значение

целое число.

Замечания

TRUNC и INT похожи в том, что оба возвращают целые числа. TRUNC удаляет дробную часть числа. INT округляет числа до ближайшего целого числа в зависимости от значения дробной части числа. INT и TRUNC отличаются только при использовании отрицательных чисел: `TRUNC(-4.3)` возвращает -4, но `INT(-4.3)` возвращает -5, так как -5 меньшее число.

Пример 1

Следующая формула возвращает 3, целочисленную часть pi.

DAX

```
= TRUNC(PI())
```

Пример 2

Следующая формула возвращает -8, целочисленную часть -8.9.

DAX

```
= TRUNC(-8.9)
```

Связанное содержимое

функции Math и TrigROUNDROUNDUPROUNDDOWNMROUNDINT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Другие функции

Статья • 22.01.2025

Эти функции выполняют уникальные действия, которые не могут быть определены любой из категорий.

В этой категории

Развернуть таблицу

Функция	Описание
BLANK	Возвращает пустое значение.
ERROR	Вызывает ошибку с сообщением об ошибке.
EVALUATEANDLOG	Возвращает значение первого аргумента и регистрирует его в событии профилировщика журнала оценки DAX.
TOCSV	Возвращает таблицу в виде строки в формате CSV.
TOJSON	Возвращает таблицу в виде строки в формате JSON.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

BLANK

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает пустое значение.

Синтаксис

DAX

```
BLANK()
```

Возвращаемое значение

Пустое.

Замечания

- Пустые значения не эквивалентны значениям NULL. DAX использует пустые значения для базы данных NULL и пустых ячеек в Excel.
- Некоторые функции DAX обрабатывают пустые ячейки по-другому от Microsoft Excel. Пустые и пустые строки ("") не всегда эквивалентны, но некоторые операции могут рассматривать их как такие.

Пример

В следующем примере показано, как работать с пустыми полями в формулах. Формула вычисляет соотношение продаж между торговыми посредниками и каналами Интернета. Однако прежде чем пытаться вычислить соотношение знаменателя, следует проверить для нуля значений. Если знаменатель равен нулю, возвращается пустое значение; в противном случае коэффициент вычисляется.

DAX

```
= IF( SUM(InternetSales_USD[SalesAmount_USD])= 0   , BLANK()   ,
SUM(ResellerSales_USD[SalesAmount_USD])/SUM(InternetSales_USD[SalesAmount_US
```

```
D])   )
```

В таблице показаны ожидаемые результаты, когда эта формула используется для создания визуализации таблицы.

Развернуть таблицу

Метки строк	Принадлежности	Велосипеды	Одежда	Общий итог
2005		2.65		2.89
2006		3.33		4.03
2007	1.04	2.92	6.63	3.51
2008	0.41	1.53	2.00	1.71
Общий итог	0.83	2.51	5.45	2.94

В исходном источнике данных столбец, вычисляемый функцией BLANK, может включать текст, пустые строки или значения NULL. Если исходный источник данных был базой данных SQL Server, значения NULL и пустые строки являются различными типами данных. Однако для этой операции выполняется неявное приведение типов и DAX обрабатывает их как те же.

Связанное содержимое

функции текстовых функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ERROR

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Вызывает ошибку с сообщением об ошибке.

Синтаксис

```DAX
ERROR(<text>)
```

Параметры

Развернуть таблицу

Срок	Определение
`text`	Текстовая строка, содержащая сообщение об ошибке.

Возвращаемое значение

Никакой

Замечания

- Функция ERROR может быть помещена в выражение DAX в любом месте скалярного значения.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

Следующий запрос DAX:

DAX

```
DEFINE
MEASURE DimProduct[Measure] =
        IF(
            SELECTEDVALUE(DimProduct[Color]) = "Red",
            ERROR("red color encountered"),
            SELECTEDVALUE(DimProduct[Color])
        )
EVALUATE SUMMARIZECOLUMNS(DimProduct[Color], "Measure", [Measure])
ORDER BY [Color]
```

Завершается сбоем и возникает сообщение об ошибке, содержащее "красный цвет обнаружен".

Пример 2

Следующий запрос DAX:

DAX

```
DEFINE
MEASURE DimProduct[Measure] =
        IF(
            SELECTEDVALUE(DimProduct[Color]) = "Magenta",
            ERROR("magenta color encountered"),
            SELECTEDVALUE(DimProduct[Color])
        )
EVALUATE SUMMARIZECOLUMNS(DimProduct[Color], "Measure", [Measure])
ORDER BY [Color]
```

Возвращает следующую таблицу:

Развернуть таблицу

DimProduct[Color]	[Мера]
Чёрный	Чёрный
Синий	Синий
Серый	Серый
Мульти	Мульти
NA	NA
Красный	Красный

DimProduct[Color]	[Мера]
Серебро	Серебро
Silver\Black	Silver\Black
Белый	Белый
Жёлтый	Жёлтый

Так как Magenta не является одним из цветов продукта, функция ERROR не выполняется.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EVALUATEANDLOG

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает значение первого аргумента и регистрирует его в событии профилировщика журнала оценки DAX. Эта функция полностью функциональна только в Power BI Desktop. Она выступает в качестве простой сквозной функции в других средах.

Синтаксис

```
DAX

EVALUATEANDLOG(<Value>, [Label], [MaxRows])
```

Параметры

Развернуть таблицу

Срок	Определение
`Value`	Любое скалярное выражение или табличное выражение для вычисления и ведения журнала.
`Label`	(Необязательно) Константная строка, включенная как в текст JSON, так и столбец Label события журнала оценки DAX, который можно использовать для легкого определения экземпляра вызова функции.
`MaxRows`	(Необязательно) Максимальное количество строк в тексте json события журнала оценки DAX, когда первый аргумент является табличным выражением. Значение по умолчанию — 10.

Возвращаемое значение

Значение первого аргумента.

Структура JSON, зарегистрированная в событии профилировщика журнала оценки DAX, включает:

- `expression` — это текстовая версия первого аргумента.

- `label` — это параметр Label, указанный в выражении.
- `inputs` — это список столбцов в контексте оценки, который влияет на значения первого аргумента.
- `outputs` — это список одного столбца [значение], когда первый аргумент является скалярным выражением и списком выходных столбцов, когда первый аргумент является табличным выражением.
- `data` — это список входных значений и выходных значений, когда первый аргумент является скалярным выражением, а также список входных значений и соответствующих выходных строк, когда первый аргумент является табличным выражением.
- `rowCount` — это количество строк, когда первый аргумент является табличным выражением. Несмотря на то, что количество строк в выходных данных JSON усечено параметром MaxRows, rowCount — это реальное количество строк без усечения.

Замечания

- События трассировки можно записать с помощью SQL Server Profiler и средства DAX отладки выходных данных с открытым исходным кодом.
- Эта функция может использоваться практически с любым вложенным выражением в выражении DAX, и все выражение по-прежнему будет допустимым.
- Когда первый аргумент вычисляется несколько раз в одном запросе, функция создает одно событие журнала оценки DAX, содержащее как входные значения, так и соответствующие выходные значения.
- Если указан параметр метки, его значение возвращается как в выходных данных JSON, так и в столбце Label события журнала оценки DAX.
- Если первый аргумент является табличным выражением, в событии журнала оценки DAX отображаются только верхние строки MaxRows.
- В некоторых случаях эта функция не выполняется из-за оптимизации.
- Если событие журнала оценки DAX больше одного миллиона символов, оно усечено для сохранения правильной структуры JSON.

Пример 1

Следующий запрос DAX:

DAX

```
evaluate
SUMMARIZE(
    EVALUATEANDLOG(FILTER(Sales, [ProductKey] = 528)),
    Sales[SalesTerritoryKey],
    "sum",
    sum(Sales[Sales Amount])
)
```

Возвращает следующее событие журнала оценки DAX:

JSON


```
{
    "expression": "FILTER(Sales, [ProductKey] = 528)",
    "inputs": [],
    "outputs": ["'Sales'[SalesOrderLineKey]", "'Sales'[ResellerKey]",
"'Sales'[CustomerKey]", "'Sales'[ProductKey]", "'Sales'[OrderDateKey]",
"'Sales'[DueDateKey]", "'Sales'[ShipDateKey]", "'Sales'[SalesTerritoryKey]",
"'Sales'[Order Quantity]", "'Sales'[Unit Price]", "'Sales'[Extended
Amount]", "'Sales'[Product Standard Cost]", "'Sales'[Total Product Cost]",
"'Sales'[Sales Amount]", "'Sales'[Unit Price Discount Pct]"],
    "data": [
        {
            "input": [],
            "rowCount": 3095,
            "output": [
                [52174001, -1, 23785, 528, 20190707, 20190717, 20190714, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                [52173001, -1, 26278, 528, 20190707, 20190717, 20190714, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                [52082001, -1, 23831, 528, 20190705, 20190715, 20190712, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                [52054002, -1, 11207, 528, 20190704, 20190714, 20190711, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                [52036001, -1, 25337, 528, 20190704, 20190714, 20190711, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                [51939002, -1, 23670, 528, 20190702, 20190712, 20190709, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                [51911002, -1, 11746, 528, 20190701, 20190711, 20190708, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                [51379003, -1, 13745, 528, 20190612, 20190622, 20190619, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                [51264002, -1, 11282, 528, 20190605, 20190615, 20190612, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0],
                [51184003, -1, 11263, 528, 20190531, 20190610, 20190607, 1,
1, 4.99, 4.99, 1.8663, 1.8663, 4.99, 0.0]
            ]
        }
    ]
}
```

Пример 2

Следующий запрос DAX с скалярным аргументом и различными атрибутами:

DAX

```
evaluate
SELECTCOLUMNS(
    TOPN(5, Customer),
    [Customer],
    "Customer",
    EVALUATEANDLOG([Customer] & ", " & [Country-Region], "customerLog")
)
```

Возвращает следующее событие журнала оценки DAX:

JSON

```
{
    "expression": "[Customer] & \", \" & [Country-Region]",
    "label": "customerLog",
    "inputs": ["'Customer'[Customer]", "'Customer'[Country-Region]"],
    "data": [
        {
            "input": ["Russell Xie", "United States"],
            "output": "Russell Xie, United States"
        },
        {
            "input": ["Savannah Baker", "United States"],
            "output": "Savannah Baker, United States"
        },
        {
            "input": ["Maurice Tang", "United States"],
            "output": "Maurice Tang, United States"
        },
        {
            "input": ["Emily Wood", "United States"],
            "output": "Emily Wood, United States"
        },
        {
            "input": ["Meghan Hernandez", "United States"],
            "output": "Meghan Hernandez, United States"
        }
    ]
}
```

Связанное содержимое

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TOCSV

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу в виде строки в формате CSV.

Синтаксис

```
DAX
TOCSV(<Table>, [MaxRows], [Delimiter], [IncludeHeaders])
```

Параметры

Развернуть таблицу

Термин	Определение
`Table`	Таблица, преобразованная в CSV-файл.
`MaxRows`	(Необязательно) Максимальное число строк fo для преобразования. Значение по умолчанию — 10 строк.
`Delimiter`	(Необязательно) Разделитель столбцов. По умолчанию используется запятая ",".
`IncludeHeaders`	(Необязательно) Задает заголовок с именем столбца в качестве первой строки. Значение по умолчанию — True.

Возвращаемое значение

Строка с представлением CSV таблицы.

Пример

Следующий запрос DAX:

```
DAX
```

```
EVALUATE
{TOCSV(DimSalesTerritory)}
```

Возвращает:

```
'DimSalesTerritory'[SalesTerritoryKey],'DimSalesTerritory'[SalesTerritoryAlt
ernateKey],'DimSalesTerritory'[SalesTerritoryRegion],'DimSalesTerritory'[Sal
esTerritoryCountry],'DimSalesTerritory'[SalesTerritoryGroup]
1,1,Northwest,United States,North America
2,2,Northeast,United States,North America
3,3,Central,United States,North America
4,4,Southwest,United States,North America
5,5,Southeast,United States,North America
6,6,Canada,Canada,North America
7,7,France,France,Europe
8,8,Germany,Germany,Europe
9,9,Australia,Australia,Pacific
10,10,United Kingdom,United Kingdom,Europe
```

Связанное содержимое

TOJSON EVALUATEANDLOG

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TOJSON

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу в виде строки с использованием формата JSON.

Синтаксис

```
DAX
TOJSON(<Table>, [MaxRows])
```

Параметры

Развернуть таблицу

Термин	Определение
`Table`	Таблица, преобразованная в JSON.
`MaxRows`	(Необязательно) Максимальное число строк fo для преобразования. Значение по умолчанию — 10 строк.

Возвращаемое значение

Строка с представлением JSON таблицы. Представление содержит имена столбцов как "заголовок", количество строк как "rowCount" и значения как "data".

Пример

Следующий запрос DAX:

```
DAX
EVALUATE
{TOJSON(DimSalesTerritory)}
```

Возвращает:

JSON


```
{
"header": ["'DimSalesTerritory'[SalesTerritoryKey]",
"'DimSalesTerritory'[SalesTerritoryAlternateKey]",
"'DimSalesTerritory'[SalesTerritoryRegion]",
"'DimSalesTerritory'[SalesTerritoryCountry]",
"'DimSalesTerritory'[SalesTerritoryGroup]"],<br>
"rowCount": 11,
"data": [
[1, 1, "Northwest", "United States", "North America"],
[2, 2, "Northeast", "United States", "North America"],
[3, 3, "Central", "United States", "North America"],
[4, 4, "Southwest", "United States", "North America"],
[5, 5, "Southeast", "United States", "North America"],
[6, 6, "Canada", "Canada", "North America"],
[7, 7, "France", "France", "Europe"],
[8, 8, "Germany", "Germany", "Europe"],
[9, 9, "Australia", "Australia", "Pacific"],
[10, 10, "United Kingdom", "United Kingdom", "Europe"]
]
}
```


Связанное содержимое

TOCSV EVALUATEANDLOG

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Родительские и дочерние функции

Статья • 22.01.2025

Эти функции управляют данными, представленными как родительские или дочерние иерархии. Дополнительные сведения см. в разделе Основные функции для Parent-Child иерархий в DAX.

В этой категории

Развернуть таблицу

Функция	Описание
PATH	Возвращает текстовую строку с разделителями с идентификаторами всех родителей текущего идентификатора.
PATHCONTAINS	Возвращает `TRUE`, если указанный `item` существует в указанной `path`.
PATHITEM	Возвращает элемент по указанному `position` из строки, полученной при оценке функции PATH.
PATHITEMREVERSE	Возвращает элемент по указанному `position` из строки, полученной при оценке функции PATH.
PATHLENGTH	Возвращает число родителей указанному элементу в заданном PATH результатах, включая себя.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Общие сведения о функциях иерархий родительского-дочернего элемента в DAX

Статья • 22.01.2025

DAX предоставляет пять функций, помогающих пользователям управлять данными, представленными в качестве иерархии родительского-дочернего элемента в своих моделях. С помощью этих функций пользователь может получить всю строку происхождения родителей, сколько уровней имеет происхождение верхнего родителя, который является родительским n-уровнем над текущей строкой, который является n-потомком из верхней части текущей иерархии строк и является родительским родителем в текущей иерархии строк?

Родительские дочерние функции в DAX

В следующей таблице содержится иерархия Parent-Child столбцов: **EmployeeKey** и **ParentEmployeeKey**, которая используется во всех примерах функций.

Развернуть таблицу

EmployeeKey	ParentEmployeeKey
112	
14	112
3	14
11	3
13	3
162	3
117	162
221	162
81	162

В приведенной выше таблице видно, что у сотрудника 112 нет родительского определения, сотрудник 14 имеет сотрудников 112 в качестве руководителя (ParentEmployeeKey), сотрудник 3 имеет сотрудников 14 в качестве руководителя и

сотрудников 11, 13 и 162 имеют сотрудников 3 в качестве руководителя. Приведенный выше помогает понять, что у сотрудника 112 нет руководителя над ней/ей, и она/она является топ-менеджером для всех сотрудников, показанных здесь; кроме того, сотрудник 3 сообщает сотрудникам 14 и сотрудникам 11, 13, 162 отчета до 3.

В следующей таблице представлены доступные функции, краткое описание функции и пример функции над теми же данными, которые показаны выше.

PATH функция — возвращает разделенный текст с идентификаторами всех родителей в текущую строку, начиная с самого старого или верхнего до текущего.

Развернуть таблицу

EmployeeKey	ParentEmployeeKey	Путь
112		112
14	112	112|14
3	14	112|14|3
11	3	112|14|3|11
13	3	112|14|3|13
162	3	112|14|3|162
117	162	112|14|3|162|117
221	162	112|14|3|162|221
81	162	112|14|3|162|81

PATHLENGTH функция — возвращает количество уровней в заданной PATH(), начиная с текущего уровня до самого старого или верхнего родительского уровня. В следующем примере столбец PathLength определяется как "`= PATHLENGTH([Path])`"; В этом примере содержатся все данные из примера Path(), которые помогут понять, как работает эта функция.

Развернуть таблицу

EmployeeKey	ParentEmployeeKey	Путь	PathLength
112		112	1
14	112	112|14	2

EmployeeKey	ParentEmployeeKey	Путь	PathLength
3	14	112|14|3	3
11	3	112|14|3|11	4
13	3	112|14|3|13	4
162	3	112|14|3|162	4
117	162	112|14|3|162|117	5
221	162	112|14|3|162|221	5
81	162	112|14|3|162|81	5

PATHITEM функция — возвращает элемент в указанной позиции из PATH(например, результат, подсчет слева направо). В следующем примере столбец PathItem — 4-й слева определяется как "`= PATHITEM([Path], 4)`"; В этом примере возвращается значение "Нанимайки" в четвертой позиции в строке пути слева, используя те же образцы данных из примера Path().

Развернуть таблицу

EmployeeKey	ParentEmployeeKey	Путь	PathItem — 4-й слева
112		112	
14	112	112|14	
3	14	112|14|3	
11	3	112|14|3|11	11
13	3	112|14|3|13	13
162	3	112|14|3|162	162
117	162	112|14|3|162|117	162
221	162	112|14|3|162|221	162
81	162	112|14|3|162|81	162

PATHITEMREVERSE функция — возвращает элемент в `position` из PATH(например, результата функции), подсчитывая обратно справа налево. В следующем примере столбец PathItemReverse — 3rd from right определяется как "`= PATHITEMREVERSE([Path], 3)`"; в этом примере возвращается функция UsingeKey в

третьей позиции в строке path справа, используя те же примеры данных из примера Path().

Развернуть таблицу

EmployeeKey	ParentEmployeeKey	Путь	PathItemReverse — 3-е место справа
112		112	
14	112	112|14	
3	14	112|14|3	112
11	3	112|14|3|11	14
13	3	112|14|3|13	14
162	3	112|14|3|162	14
117	162	112|14|3|162|117	3
221	162	112|14|3|162|221	3
81	162	112|14|3|162|81	3

функция PATHCONTAINS — возвращает `TRUE`, если указанный `item` существует в пределах указанного `path`. В следующем примере столбец PathContains — employee 162 определяется как "`= PATHCONTAINS([Path], "162")`"; в этом примере возвращается `TRUE`, если заданный путь содержит сотрудник 162. В этом примере используются результаты из приведенного выше примера Path().

Развернуть таблицу

EmployeeKey	ParentEmployeeKey	Путь	PathContains - сотрудник 162
112		112	`FALSE`
14	112	112|14	`FALSE`
3	14	112|14|3	`FALSE`
11	3	112|14|3|11	`FALSE`
13	3	112|14|3|13	`FALSE`
162	3	112|14|3|162	`TRUE`
117	162	112|14|3|162|117	`TRUE`

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PATH

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает текстовую строку с разделителями с идентификаторами всех родителей текущего идентификатора, начиная с самого старого и продолжающегося до текущего.

Синтаксис

DAX

```
PATH(<ID_columnName>, <parent_columnName>)
```

Параметры

Развернуть таблицу

Срок	Определение
`ID_columnName`	Имя существующего столбца, содержащего уникальный идентификатор строк в таблице. Это не может быть выражением. Тип данных значения в `ID_columnName` должен быть текстом или целым числом, а также должен быть тем же типом данных, что и столбец, на который ссылается столбец в `parent_columnName`.
`parent_columnName`	Имя существующего столбца, содержащего уникальный идентификатор для родительского элемента текущей строки. Это не может быть выражением. Тип данных в `parent_columnName` тип данных должен быть текстом или целым числом, и должен быть таким же типом данных, что и значение в `ID_columnName`.

Возвращаемое значение

Текстовая строка с разделителями, содержащая идентификаторы всех родителей текущего идентификатора.

Замечания

- Эта функция используется в таблицах, имеющих определенную внутреннюю иерархию, для возврата элементов, связанных с текущим значением строки. Например, в таблице "Сотрудники", содержащей сотрудников, руководителей сотрудников и руководителей менеджеров, можно вернуть путь, который связывает сотрудника с его руководителем.

- Путь не ограничивается одним уровнем отношений "родительский-дочерний"; он может возвращать связанные строки, которые находятся на нескольких уровнях из указанной начальной строки.
  - Разделитель, используемый для разделения возрастания, является вертикальной полосой, `|`.
  - Значения в `ID_columnName` и `parent_columnName` должны иметь одинаковый тип данных, текст или целое число.
  - Значения в `parent_columnName` должны присутствовать в `ID_columnName`. То есть вы не можете искать родителя, если на дочернем уровне нет значения.
  - Если `parent_columnName` `BLANK`, `PATH()` возвращает значение `ID_columnName`. Другими словами, если вы ищете руководителя сотрудника, но столбец `parent_columnName` не имеет данных, функция PATH возвращает только идентификатор сотрудника.
  - Если `ID_columnName` имеет дубликаты и `parent_columnName` совпадает с этими дубликатами, `PATH()` возвращает общее значение `parent_columnName`; Однако если `parent_columnName` значение отличается для этих дубликатов, `PATH()` возвращает ошибку. Другими словами, если у вас есть два списка для одного и того же идентификатора сотрудника, и у них есть один и тот же идентификатор руководителя, функция PATH возвращает идентификатор для этого руководителя. Однако при наличии двух идентичных идентификаторов сотрудников с разными идентификаторами руководителя функция PATH возвращает ошибку.
  - Если `ID_columnName` `BLANK` `PATH()` возвращается `BLANK`.
  - Если `ID_columnName` содержит вертикальную полосу `|`, `PATH()` возвращает ошибку.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере создается вычисляемый столбец, который содержит список всех руководителей для каждого сотрудника.

DAX

```
= PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey])
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PATHCONTAINS

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает `TRUE`, если указанный `item` существует в указанной `path`.

Синтаксис

DAX

```
PATHCONTAINS(<path>, <item>)
```

Параметры

Развернуть таблицу

Срок	Определение
`path`	Строка, созданная в результате оценки функции PATH.
`item`	Текстовое выражение для поиска в результате пути.

Возвращаемое значение

Значение `TRUE`, если `item` существует в `path`; в противном случае `FALSE`.

Замечания

- Если `item` является целым числом, оно преобразуется в текст, а затем вычисляется функция. Если преобразование завершается ошибкой, функция возвращает ошибку.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере создается вычисляемый столбец, который принимает идентификатор руководителя и проверяет набор сотрудников. Если идентификатор руководителя находится в списке руководителей, возвращаемых функцией PATH, функция PATHCONTAINS возвращает значение true; в противном случае возвращает значение false.

DAX

```
= PATHCONTAINS(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]),
"23")
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PATHITEM

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает элемент по указанному `position` из строки, полученной при оценке функции PATH. Позиции учитываются слева направо.

Синтаксис

DAX

```
PATHITEM(<path>, <position>[, <type>])
```

Параметры

Развернуть таблицу

Срок	Определение
`path`	Текстовая строка в виде результатов функции PATH.
`position`	Целочисленное выражение с положением возвращаемого элемента.
`type`	(Необязательно)Перечисление, определяющее тип данных результата:

Перечисление типов

Развернуть таблицу

Перечисление	Альтернативное перечисление	Описание
`TEXT`	0	Результаты возвращаются с текстом типа данных. (по умолчанию).
`INTEGER`	1	Результаты возвращаются в виде целых чисел.

Возвращаемое значение

Идентификатор, возвращаемый функцией PATH в указанной позиции в списке идентификаторов. Элементы, возвращаемые функцией PATH, упорядочены наиболее удаленными к текущим.

Замечания

- Эту функцию можно использовать для возврата определенного уровня из иерархии, возвращаемой функцией PATH. Например, можно вернуть только руководителей на уровне пропуска для всех сотрудников.
- Если указать число для `position` меньше одного (1) или больше числа элементов в `path`, функция PATHITEM возвращает BLANK
- Если `type` не является допустимым элементом перечисления, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере возвращается диспетчер третьего уровня текущего сотрудника; Он принимает идентификаторы сотрудников и руководителей в качестве входных данных в функцию PATH, которая возвращает строку с иерархией родителей текущему сотруднику. Из этой строки PATHITEM возвращает третью запись в виде целого числа.

DAX

```
= PATHITEM(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PATHITEMREVERSE

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает элемент по указанному `position` из строки, полученной при оценке функции PATH. Позиции подсчитываются назад справа налево.

Синтаксис

DAX

```
PATHITEMREVERSE(<path>, <position>[, <type>])
```

Параметры

Развернуть таблицу

Срок	Определение
`path`	Текстовая строка, полученная при оценке функции PATH.
`position`	Целочисленное выражение с положением возвращаемого элемента. Позиция подсчитывается назад справа налево.
`type`	(Необязательно)Перечисление, определяющее тип данных результата:

Перечисление типов

Развернуть таблицу

Перечисление	Альтернативное перечисление	Описание
`TEXT`	0	Результаты возвращаются с текстом типа данных. (по умолчанию).
`INTEGER`	1	Результаты возвращаются в виде целых чисел.

Возвращаемое значение

N-позицию по возрастанию в заданном пути, подсчитывая от текущего до самого старого.

Замечания

- Эту функцию можно использовать для получения отдельного элемента из иерархии, полученной из PATH функции.
- Эта функция изменяет стандартный порядок иерархии, чтобы ближайшие элементы были перечислены сначала, например, если функция PATh возвращает список руководителей над сотрудником в иерархии, функция PATHITEMREVERSE возвращает непосредственного руководителя сотрудника в позиции 2, так как позиция 1 содержит идентификатор сотрудника.
- Если число, указанное для `position`, меньше одного (1) или больше числа элементов в `path`, функция PATHITEM возвращает BLANK.
- Если `type` не является допустимым элементом перечисления, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере в качестве входных данных в функцию PATH используется столбец идентификатора сотрудника, который возвращает список возвращаемых элементов бабушки и дедушки. Указана позиция 3, а возвращаемый тип — 1; Таким образом, функция PATHITEMREVERSE возвращает целое число, представляющее руководителя два уровня от сотрудника.

DAX

```
= PATHITEMREVERSE(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]),
3, 1)
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PATHLENGTH

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает число родителей указанному элементу в заданном PATH результатах, включая себя.

Синтаксис

DAX

```
PATHLENGTH(<path>)
```

Параметры

Развернуть таблицу

Срок	Определение
`path`	Текстовое выражение, полученное от оценки функции PATH.

Возвращаемое значение

Количество элементов, которые являются родителями указанного элемента в заданном PATH результата, включая указанный элемент.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере используется идентификатор сотрудника в качестве входных данных для функции PATH и возвращается список руководителей выше, с

которыми работает сотрудник в иерархии, функция PATHLENGTH принимает этот результат и подсчитывает различные уровни сотрудников и руководителей, включая сотрудников, с которыми вы начали работу.

DAX

```
= PATHLENGTH(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]))
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Функции связи

Статья • 21.01.2025

Функции в этой категории предназначены для управления и использования связей между таблицами.

В этой категории

Развернуть таблицу

Функция	Описание
CROSSFILTER	Указывает направление перекрестной фильтрации, используемое в вычислении связи, которая существует между двумя столбцами.
RELATED	Возвращает связанное значение из другой таблицы.
RELATEDTABLE	Вычисляет табличное выражение в контексте, измененном заданными фильтрами.
USERELATIONSHIP	Указывает связь, используемую в определенном вычислении в качестве той, которая существует между columnName1 и columnName2.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CROSSFILTER

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Указывает направление перекрестной фильтрации, используемое в вычислении связи, которая существует между двумя столбцами.

Синтаксис

```DAX
CROSSFILTER(<columnName1>, <columnName2>, <direction>)
```

Параметры

Развернуть таблицу

Срок	Определение
`columnName1`	Имя существующего столбца, использующего стандартный синтаксис DAX и полный синтаксис, который обычно представляет многие стороны используемой связи; Если аргументы задаются в обратном порядке, функция переключится перед их использованием. Этот аргумент не может быть выражением.
`columnName2`	Имя существующего столбца, используя стандартный синтаксис DAX и полный синтаксис, который обычно представляет одну сторону или сторону подстановки используемой связи; Если аргументы задаются в обратном порядке, функция переключится перед их использованием. Этот аргумент не может быть выражением.
`Direction`	Направление перекрестной фильтрации, используемое. Должно быть одним из следующих элементов: `None` — по этой связи не выполняется перекрестная фильтрация. `Both` — фильтры на обеих сторонах фильтруют другую сторону. `OneWay` — фильтры на одной стороне или стороне подстановки фильтра отношений с другой стороны. Этот параметр нельзя использовать с связью "один к одному". Не используйте этот параметр в отношениях "многие ко многим", потому что неясно, какая сторона является стороной подстановки;

Срок	Определение
	вместо этого используйте OneWay_LeftFiltersRight или OneWay_RightFiltersLeft. `OneWay_LeftFiltersRight` — фильтры на стороне `columnName1` фильтруют сторону `columnName2`. Этот параметр нельзя использовать с отношением "один к одному" или "ко многим". `OneWay_RightFiltersLeft` — фильтры на стороне `columnName2` фильтруют сторону `columnName1`. Этот параметр нельзя использовать с отношением "один к одному" или "ко многим".

Возвращаемое значение

Функция не возвращает значения; Функция задает только направление перекрестной фильтрации для указанной связи в течение длительности запроса.

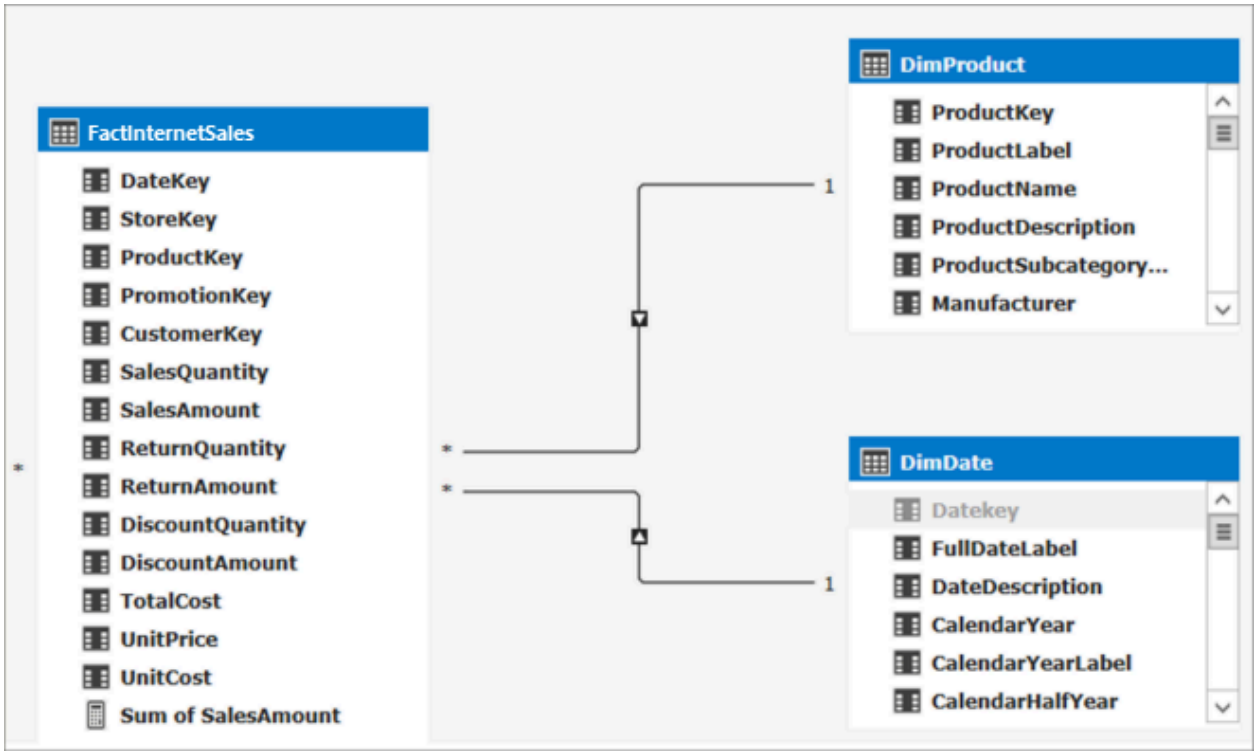
Замечания

- В случае отношения 1:1 нет разницы между одним и обоими направлениями.
- CROSSFILTER можно использовать только в функциях, которые принимают фильтр в качестве аргумента, например, CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTHCLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD и TOTALYTD функций.
- CROSSFILTER использует существующие связи в модели, определяя связи по их конечным столбцам точек.
- В CROSSFILTERпараметр перекрестной фильтрации связи не важен; То есть, настроена ли связь для фильтрации одной или обоих направлений в модели, не влияет на использование функции. CROSSFILTER переопределит любой существующий параметр перекрестной фильтрации.
- Ошибка возвращается, если любой из столбцов, именованных как аргумент, не является частью связи или аргументы принадлежат разным связям.
- Если выражения CALCULATE вложены, а несколько выражений CALCULATE содержат функцию CROSSFILTER, то самым внутренним CROSSFILTER является тот, который преобладает в случае конфликта или неоднозначности.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на

уровне строк (RLS).

Пример

На следующей схеме модели как DimProduct, так и DimDate имеют одно направление связи с FactInternetSales.



По умолчанию мы не можем получить количество продуктов, проданных по году:

Row Labels	Sum of SalesAmount	Distinct Count of ProductKey
2005	$3,266,373.66	606
2006	$6,530,343.53	606
2007	$9,791,060.30	606
2008	$9,770,899.74	606
2009		606
2010		606
Grand Total	**$29,358,677.22**	**606**

Количество продуктов по годам можно получить двумя способами:

- Включите двунаправленную перекрестную фильтрацию для связи. Это изменит способ работы фильтров для всех данных между этими двумя таблицами.
- Используйте функцию CROSSFILTER, чтобы изменить способ работы связей только для этой меры.

При использовании DAXможно использовать функцию CROSSFILTER для изменения поведения направления перекрестной фильтрации между двумя столбцами, определенными связью. В этом случае выражение DAX выглядит следующим образом:

DAX

```
BiDi:= CALCULATE([Distinct Count of ProductKey],
CROSSFILTER(FactInternetSales[ProductKey], DimProduct[ProductKey] , Both))
```

Используя функцию CROSSFILTER в выражении меры, мы получаем ожидаемые результаты:

Row Labels	Sum of SalesAmount	Distinct Count of ProductKey	BiDi
2005	$3,266,373.66	606	25
2006	$6,530,343.53	606	56
2007	$9,791,060.30	606	133
2008	$9,770,899.74	606	102
2009		606	
2010		606	
Grand Total	**$29,358,677.22**	**606**	**606**

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RELATED

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Возвращает связанное значение из другой таблицы.

Синтаксис

```DAX
RELATED(<column>)
```

Параметры

Развернуть таблицу

Термин	Определение
`column`	Столбец, содержащий значения, которые требуется извлечь.

Возвращаемое значение

Одно значение, связанное с текущей строкой.

Замечания

- Функция RELATED требует, чтобы связь существовала между текущей таблицей и таблицей со связанными сведениями. Укажите столбец, содержащий нужные данные, и функция следует существующей связи "многие ко одному", чтобы получить значение из указанного столбца в связанной таблице. Если связь не существует, необходимо создать связь.
- Когда функция RELATED выполняет поиск, он проверяет все значения в указанной таблице независимо от любых фильтров, которые могли быть применены.

- Функция RELATED нуждается в контексте строки; поэтому его можно использовать только в вычисляемом выражении столбца, где текущий контекст строки является однозначной или вложенной функцией в выражении, использующим функцию сканирования таблиц. Функция сканирования таблиц, например SUMX, получает значение текущего значения строки, а затем сканирует другую таблицу для экземпляров этого значения.
- Функцию RELATED нельзя использовать для получения столбца в ограниченной связи.

Пример

В следующем примере создается мера , не США Интернет-продажи, чтобы создать отчет о продажах, который исключает продажи в Соединенных Штатах. Чтобы создать меру, необходимо отфильтровать таблицу InternetSales_USD, чтобы исключить все продажи, принадлежащие Соединенным Штатам в таблице SalesTerritory. Соединенные Штаты, как страна, появляются в таблице SalesTerritory 5 раз; один раз для каждого из следующих регионов: Северо-Запад, Северо-Восток, Центральная, Юго-Западная и Юго-Восточная.

Первый подход к фильтрации интернет-продаж, чтобы создать меру, может быть добавлено выражение фильтра, как показано ниже:

DAX

```
FILTER('InternetSales_USD'
, 'InternetSales_USD'[SalesTerritoryKey]<>1 &&
'InternetSales_USD'[SalesTerritoryKey]<>2 &&
'InternetSales_USD'[SalesTerritoryKey]<>3 &&
'InternetSales_USD'[SalesTerritoryKey]<>4 &&
'InternetSales_USD'[SalesTerritoryKey]<>5)
```

Однако этот подход является контринтуитивным, склонным к вводу ошибок, и может не работать, если какой-либо из существующих регионов разделен в будущем.

Лучше всего использовать существующие отношения между InternetSales_USD и SalesTerritory и явно утверждать, что страна должна отличаться от Соединенных Штатов. Для этого создайте выражение фильтра, как показано ниже:

DAX

```
FILTER( 'InternetSales_USD',
```

```
RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States")
```

Это выражение использует функцию RELATED для поиска значения страны в таблице SalesTerritory, начиная со значения ключевого столбца SalesTerritoryKey в таблице InternetSales_USD. Результат подстановки используется функцией фильтра, чтобы определить, фильтруется ли строка InternetSales_USD.

> **Примечание**
>
> Если пример не работает, может потребоваться создать связь между таблицами.

DAX

```
= SUMX(FILTER( 'InternetSales_USD'
            ,  RELATED('SalesTerritory'[SalesTerritoryCountry])
               <>"United States"
             )
     ,'InternetSales_USD'[SalesAmount_USD])
```

В следующей таблице показаны только итоги для каждого региона, чтобы доказать, что выражение фильтра в измерении, отличном от США интернет-продаж, работает должным образом.

Развернуть таблицу

Row Labels	Internet Sales	Non USA Internet Sales
Австралия	$499,021,84	$499,021,84
Канада	$1,343,109.10	$1,343,109.10
Франция	$2,490,944,57	$2,490,944,57
Германия	$2775,195,60	$2775,195,60
Соединенное Королевство	$5,057,076,55	$5,057,076,55
Соединенные Штаты	$9,389,479,79	
Общий итог	$26,054,827,45	$16,665,347,67

Ниже показано, что можно получить, если вы использовали эту меру в визуальном элементе таблицы отчетов:

Развернуть таблицу

Row Labels	Accessories	Bikes	Clothing	Grand Total
2005		$1,526,481,95		$1,526,481,95
2006		$354,744.04		$354,744.04
2007	$156,480,18	$5,640,106,05	$70,142,77	$5,866,729,00
2008	$228,159,45	$ 5,386,558.19	$102,675.04	$5,717,392,68
Общий итог	$384,639,633	$16,107,890,23	$ 172,817.81	$16,665,347,67

Связанное содержимое

RELATEDTABLE функции фильтрации

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RELATEDTABLE

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Вычисляет табличное выражение в контексте, измененном заданными фильтрами.

Синтаксис

DAX

```
RELATEDTABLE(<tableName>)
```

Параметры

Развернуть таблицу

Термин	Определение
`tableName`	Имя существующей таблицы с использованием стандартного синтаксиса DAX. Это не может быть выражением.

Возвращаемое значение

Таблица значений.

Замечания

- Функция RELATEDTABLE изменяет контекст, в котором фильтруются данные, и вычисляет выражение в новом контексте, который вы указали.
- Эта функция является ярлыком для функции CALCULATETABLE без логического выражения.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере функция RELATEDTABLE используется для создания вычисляемого столбца с интернет-продажами в таблице "Категория продукта".

DAX

```
= SUMX( RELATEDTABLE('InternetSales_USD')
     , [SalesAmount_USD])
```

В следующей таблице показаны результаты:

Развернуть таблицу

Product Category Key	Product Category AlternateKey	Product Category Name	Internet Sales
1	1	Велосипеды	$28,318,144.65
2	2	Компоненты	
3	3	Clothing	$339,772,61
4	4	Аксессуары	$700,759,96

Связанное содержимое

CALCULATETABLE функции фильтрации

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

USERELATIONSHIP

Статья • 23.01.2025

Область применения: вычисление вычисляемого столбца вычисляемой таблицы Меры визуального элемента

Указывает связь, используемую в определенном вычислении в качестве той, которая существует между columnName1 и columnName2.

Синтаксис

```DAX
USERELATIONSHIP(<columnName1>,<columnName2>)
```

Параметры

Развернуть таблицу

Термин	Определение
`columnName1`	Имя существующего столбца, использующего стандартный синтаксис DAX и полный синтаксис, который обычно представляет многие стороны используемой связи; Если аргументы задаются в обратном порядке, функция переключится перед их использованием. Этот аргумент не может быть выражением.
`columnName2`	Имя существующего столбца, используя стандартный синтаксис DAX и полный синтаксис, который обычно представляет одну сторону или сторону подстановки используемой связи; Если аргументы задаются в обратном порядке, функция переключится перед их использованием. Этот аргумент не может быть выражением.

Возвращаемое значение

Функция не возвращает значения; Функция включает только указанную связь в течение длительности вычисления.

Замечания

- USERELATIONSHIP можно использовать только в функциях, которые принимают фильтр в качестве аргумента, например, CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTHCLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD и TOTALYTD функций.

- USERELATIONSHIP нельзя использовать при определении безопасности на уровне строк для таблицы, в которой включена мера. Например, возвращает ошибку, `CALCULATE(SUM([SalesAmount]), USERELATIONSHIP(FactInternetSales[CustomerKey], DimCustomer[CustomerKey]))` если для DimCustomer определена безопасность на уровне строк.

- USERELATIONSHIP использует существующие связи в модели, определяя связи по их конечным столбцам точек.

- В USERELATIONSHIPсостояние связи не важно; То есть, является ли связь активной или не влияет на использование функции. Даже если связь неактивна, она будет использоваться и переопределяет любые другие активные связи, которые могут присутствовать в модели, но не упоминаются в аргументах функции.

- Ошибка возвращается, если любой из столбцов, именованных как аргумент, не является частью связи или аргументы принадлежат разным связям.

- Если требуется несколько связей для объединения таблицы A к таблице B в вычислении, каждая связь должна быть указана в другой функции USERELATIONSHIP.

- Если выражения CALCULATE вложены, а несколько выражений CALCULATE содержат функцию USERELATIONSHIP, то самым внутренним USERELATIONSHIP является тот, который преобладает в случае конфликта или неоднозначности.

- Можно вложить до 10 USERELATIONSHIP функций; однако выражение может иметь более глубокий уровень вложения, т. е. Следующее примере выражения вложено 3 уровня глубоко, но только 2 для USERELATIONSHIP: `=CALCULATE(CALCULATE( CALCULATE( <anyExpression>, USERELATIONSHIP( t1[colA], t2[colB])), t99[colZ]=999), USERELATIONSHIP( t1[colA], t2[colA]))`.

- Для связей 1–1 USERELATIONSHIP активирует связь только в одном направлении. В частности, фильтрация будет направлена от таблицы со столбцом *columnName2* к таблице со столбцом *columnName1*. Если нужно реализовать двунаправленную фильтрацию, вы можете использовать две

функции USERELATIONSHIP с противоположными направлениями в одном вычислении. Например, `CALCULATE(..., USERELATIONSHIP(T1[K], T2[K]), USERELATIONSHIP(T2[K], T1[K]))`.

Пример

В следующем примере показано, как переопределить значения по умолчанию, активные, связи между InternetSales и таблицами DateTime. Связь по умолчанию существует между столбцом OrderDate в таблице InternetSales и столбцом Date в таблице DateTime.

Чтобы вычислить сумму интернет-продаж и разрешить срезы по shippingDate вместо традиционного заказа, создайте меру [InternetSales by ShippingDate], используя следующее выражение:

DAX

```
= CALCULATE(SUM(InternetSales[SalesAmount]),
USERELATIONSHIP(InternetSales[ShippingDate], DateTime[Date]))
```

Связи между InternetSales[ShipmentDate] и DateTime[Date] должны существовать и не должны быть активными связями; Кроме того, связь между InternetSales[OrderDate] и DateTime[Date] должна существовать и должна быть активной связью.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Статистические функции

Статья • 23.01.2025

Статистические функции вычисляют значения, связанные со статистическими распределениями и вероятностью, например стандартное отклонение и количество перемутов.

В этой категории

Развернуть таблицу

Функция	Описание
BETA.DIST	Возвращает бета-версию.
BETA.INV	Возвращает обратную функцию плотности бета-накопительной вероятности (BETA.DIST).
CHISQ.DIST	Возвращает распределение чи-квадратов.
CHISQ.DIST.RT	Возвращает правостороннюю вероятность распределения чи-квадратов.
CHISQ.INV	Возвращает обратное значение левого хвоста вероятности распределения чи-квадрата.
CHISQ.INV.RT	Возвращает обратное значение правой вероятности распределения хи-квадрата.
COMBIN	Возвращает количество сочетаний для заданного количества элементов.
COMBINA	Возвращает количество сочетаний (с повторениями) для заданного количества элементов.
CONFIDENCE.NORM	Интервал доверия — это диапазон значений.
CONFIDENCE.T	Возвращает доверительный интервал для среднего числа населения, используя распределение учащихся.
EXPON.DIST	Возвращает экспоненциальное распределение.
GEOMEAN	Возвращает геометрическое среднее числа в столбце.
GEOMEANX	Возвращает геометрическое значение выражения, вычисляемого для каждой строки таблицы.

Функция	Описание
LINEST	Использует метод "Наименьшие квадраты", чтобы вычислить прямую линию, которая лучше всего подходит для заданных данных.
LINESTX	Использует метод "Наименьшие квадраты", чтобы вычислить прямую линию, которая лучше всего подходит для заданных данных. Результат данных из выражений, вычисляемых для каждой строки в таблице.
MEDIAN	Возвращает медиану чисел в столбце.
MEDIANX	Возвращает медианово число выражения, вычисляемого для каждой строки в таблице.
NORM.DIST	Возвращает нормальное распределение для указанного среднего и стандартного отклонения.
NORM.INV	Обратное нормальное совокупное распределение для указанного среднего и стандартного отклонения.
NORM.S.DIST	Возвращает стандартное нормальное распределение (имеет среднее значение нуля и стандартное отклонение одного).
NORM.S.INV	Возвращает обратное стандартное нормальное накопительное распределение.
PERCENTILE.EXC	Возвращает процентиль k-th значений в диапазоне, где k находится в диапазоне 0,1, монопольный.
PERCENTILE.INC	Возвращает процентиль k-th значений в диапазоне, где k находится в диапазоне 0.1, включительно.
PERCENTILEX.EXC	Возвращает процентиль выражения, вычисляемого для каждой строки в таблице.
PERCENTILEX.INC	Возвращает процентиль выражения, вычисляемого для каждой строки в таблице.
PERMUT	Возвращает количество переметов для заданного количества объектов, которые можно выбрать из числовых объектов.
POISSON.DIST	Возвращает распределение Poisson.
RANK.EQ	Возвращает рейтинг числа в списке чисел.
RANKX	Возвращает ранжирование числа в списке чисел для каждой строки в аргументе `table`.
SAMPLE	Возвращает пример N строк из указанной таблицы.
STDEV.P	Возвращает стандартное отклонение всего населения.

Функция	Описание
STDEV.S	Возвращает стандартное отклонение выборки.
STDEVX.P	Возвращает стандартное отклонение всего населения.
STDEVX.S	Возвращает стандартное отклонение выборки.
T.DIST	Возвращает левое распределение t-распределения студента.
T.DIST.2T	Возвращает двухстороннее распределение учащихся.
T.DIST.RT	Возвращает правостороннее распределение учащегося.
T.INV	Возвращает левый хвост обратного распределения t-распределения студента.
T.INV.2t	Возвращает двухстороннее обратное распределение учащихся.
VAR.P	Возвращает дисперсию всего населения.
VAR.S	Возвращает дисперсию выборки.
VARX.P	Возвращает дисперсию всего населения.
VARX.S	Возвращает дисперсию выборки.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

BETA.DIST

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает бета-версию. Бета-распределение обычно используется для изучения вариантов в процентах от чего-то в выборках, таких как доля день людей проводит смотреть телевизор.

Синтаксис

```DAX
BETA.DIST(x,alpha,beta,cumulative,[A],[B])
```

Параметры

Развернуть таблицу

Срок	Определение
`x`	Значение между A и B, с помощью которого необходимо оценить функцию
`Alpha`	Параметр распределения.
`Beta`	Параметр распределения.
`A`	Необязательный. Нижняя граница интервала x.
`B`	Необязательный. Верхняя граница интервала x.

Возвращаемое значение

Возвращает бета-версию.

Замечания

- Если какой-либо аргумент не является числовыми, BETA.DIST возвращает значение ошибки `#VALUE!`.

- Если какой-либо аргумент не является целым числом, округляется.
- Если `alpha ≤ 0` или бета-версия ≤ 0, BETA.DIST возвращает значение ошибки `#NUM!`.
- Если x < A, x > B или A = B, BETA.DIST возвращает значение ошибки `#NUM!`.
- Если значения A и B опущены, BETA.DIST использует стандартное накопительное бета-распределение, чтобы A = 0 и B = 1.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

BETA.INV

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает обратную функцию плотности бета-накопительной вероятности (BETA.DIST).

Если вероятность `= BETA.DIST(x,...TRUE)`, `BETA.INV(probability,...) = x`. Бета-распределение можно использовать в планировании проекта для моделирования возможного времени завершения с учетом ожидаемого времени завершения и изменчивости.

Синтаксис

DAX

```
BETA.INV(probability,alpha,beta,[A],[B])
```

Параметры

Развернуть таблицу

Срок	Определение
`Probability`	Вероятность, связанная с бета-распределением.
`Alpha`	Параметр распределения.
`Beta`	Параметр распределения.
`A`	Необязательный. Нижняя граница интервала x.
`B`	Необязательный. Верхняя граница интервала x.

Возвращаемое значение

Возвращает обратную функцию плотности бета-накопительной вероятности (BETA.DIST).

Замечания

- Если какой-либо аргумент не является числовыми, `BETA.INV` возвращает значение ошибки `#VALUE!`.
- Если какой-либо аргумент не является целым числом, округляется.
- Если `alpha` ≤ 0 или `beta` ≤ 0, `BETA.INV` возвращает значение ошибки `#NUM!`.
- Если вероятность ≤ 0 или вероятность > 1, `BETA.INV` возвращает #NUM! значение ошибки.
- Если значения A и B опущены, `BETA.INV` использует стандартное накопительное бета-распределение, чтобы A = 0 и B = 1.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CHISQ.DIST

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает распределение чи-квадратов.

Распределение чи-квадрат обычно используется для изучения вариации в процентах от чего-то в выборках, таких как доля дня люди тратят на просмотр телевизора.

Синтаксис

```DAX
CHISQ.DIST(<x>, <deg_freedom>, <cumulative>)
```

Параметры

Развернуть таблицу

Срок	Определение
`x`	Значение, по которому требуется оценить распределение.
`Deg_freedom`	Количество степеней свободы.
`cumulative`	Логическое значение, определяющее форму функции. Если накопительный `TRUE`, CHISQ.DIST возвращает накопительную функцию распределения; Если `FALSE`, он возвращает функцию плотности вероятности.

Возвращаемое значение

Распределение чи-квадратов.

Замечания

- Если `x` или `deg_freedom` не является числом, возвращается ошибка.
- Если `deg_freedom` не является целым числом, округляется.

- Если `x` < 0, возвращается ошибка.
- Если `deg_freedom` < 1 или `deg_freedom` > 10^10, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX

```
DAX

EVALUATE { CHISQ.DIST(2, 2, TRUE) }
```

Возвращает

Развернуть таблицу

[значение]
0.632120558828558

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CHISQ.DIST.RT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает правостороннюю вероятность распределения чи-квадратов.

Распределение чи-квадратов связано с тестом chi-squared. Используйте тест chi-squared для сравнения наблюдаемых и ожидаемых значений. Например, генетический эксперимент может предположить, что следующее поколение растений будет проявлять определенный набор цветов. Сравнивая наблюдаемые результаты с ожидаемыми, вы можете решить, является ли исходная гипотеза допустимой.

Синтаксис

```
DAX

CHISQ.DIST.RT(<x>, <deg_freedom>)
```

Параметры

Развернуть таблицу

Срок	Определение
`x`	Значение, по которому требуется оценить распределение.
`Deg_freedom`	Количество степеней свободы.

Возвращаемое значение

Правая вероятность распределения хи-квадрата.

Замечания

- Если `x` или `deg_freedom` не является числом, возвращается ошибка.
- Если `deg_freedom` не является целым числом, округляется.

- Если `x` < 0, возвращается ошибка.
- Если `deg_freedom` < 1 или `deg_freedom` > 10^10, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX

```
DAX

EVALUATE { CHISQ.DIST.RT(2, 5) }
```

Возвращает

Развернуть таблицу

[значение]
0.84914503608461

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CHISQ.INV

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает обратное значение левого хвоста вероятности распределения чи-квадрата.

Распределение чи-квадрат обычно используется для изучения вариации в процентах от чего-то в выборках, таких как доля дня люди тратят на просмотр телевизора.

Синтаксис

```
DAX

CHISQ.INV(probability,deg_freedom)
```

Параметры

Развернуть таблицу

Срок	Определение
`Probability`	Вероятность, связанная с распределением чи-квадратов.
`Deg_freedom`	Количество степеней свободы.

Возвращаемое значение

Возвращает обратное значение левого хвоста вероятности распределения чи-квадрата.

Замечания

- Если аргумент не является числом, CHISQ.INV возвращает значение ошибки `#VALUE!`.

- Если вероятность < 0 или вероятность > 1, CHISQ.INV возвращает значение ошибки `#NUM!`.
- Если `deg_freedom` не является целым числом, округляется.
- Если `deg_freedom` < 0 или `deg_freedom` > 10^10, CHISQ.INV возвращает значение ошибки `#NUM!`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`= CHISQ.INV(0.93,1)`	Вместо левого хвоста вероятность распределения хи-квадрата для 0,93, используя 1 степень свободы.	5.318520074
`= CHISQ.INV(0.6,2)`	Вместо левого хвоста вероятность хи-квадратного распределения для 0,6, используя 2 градуса свободы.	1.832581464

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CHISQ.INV.RT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает обратное значение правой вероятности распределения хи-квадрата.

Если вероятность = CHISQ.DIST.RT(x,...), то CHISQ.INV.RT(вероятность,...) = x. Используйте эту функцию для сравнения наблюдаемых результатов с ожидаемыми, чтобы решить, является ли исходная гипотеза допустимой.

Синтаксис

```DAX
CHISQ.INV.RT(probability,deg_freedom)
```

Параметры

Развернуть таблицу

Срок	Определение
`Probability`	Вероятность, связанная с распределением чи-квадратов.
`Deg_freedom`	Количество степеней свободы.

Возвращаемое значение

Возвращает обратное значение правой вероятности распределения хи-квадрата.

Замечания

- Если любой аргумент не является числовыми, CHISQ.INV.RT возвращает значение ошибки `#VALUE!`.
- Если `probability` < 0 или `probability` > 1, CHISQ.INV.RT возвращает значение ошибки `#NUM!`.

- Если `deg_freedom` не является целым числом, округляется.
- Если `deg_freedom` < 1, CHISQ.INV.RT возвращает значение ошибки `#NUM!`.
- Учитывая значение вероятности, CHISQ.INV.RT ищет значение x таким образом, чтобы CHISQ.DIST.RT(x, deg_freedom) = вероятность. Таким образом, точность CHISQ.INV.RT зависит от точности CHISQ.DIST.RT. CHISQ.INV.RT использует метод итеративного поиска. Если поиск не конвергентировался после 64 итерации, функция возвращает значение ошибки #N/A.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COMBIN

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает количество сочетаний для заданного количества элементов. Используйте COMBIN, чтобы определить общее возможное количество групп для заданного количества элементов.

Синтаксис

```DAX
COMBIN(number, number_chosen)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Количество элементов.
`number_chosen`	Количество элементов в каждой комбинации.

Возвращаемое значение

Возвращает количество сочетаний для заданного количества элементов.

Замечания

- Числовые аргументы усечены до целых чисел.
- Если любой аргумент не является числовыми, COMBIN возвращает значение ошибки `#VALUE!`.
- Если число < 0, number_chosen < 0 или число < number_chosen, COMBIN возвращает значение ошибки `#NUM!`.

- Сочетание — это любой набор или подмножество элементов независимо от их внутреннего порядка. Сочетания отличаются от перемутации, для которых внутренний порядок имеет значительное значение.
- Число сочетаний выглядит следующим образом: число = n и number_chosen = k:

  $${n \choose k} = \frac{P_{k,n}}{k!} = \frac{n!}{k! (n-k)!}$$

  Где

  $$P_{k,n} = \frac{n!}{(n-k)!}$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`= COMBIN(8,2)`	Возможные два человека команды, которые могут быть сформированы из 8 кандидатов.	28

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COMBINA

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает количество сочетаний (с повторениями) для заданного количества элементов.

Синтаксис

DAX

```
COMBINA(number, number_chosen)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Должно быть больше или равно 0 и больше или равно Number_chosen. Не целочисленные значения усечены.
`number_chosen`	Должно быть больше или равно 0. Не целочисленные значения усечены.

Возвращаемое значение

Возвращает количество сочетаний (с повторениями) для заданного количества элементов.

Замечания

- Если значение любого аргумента выходит за пределы ограничений, COMBINA возвращает значение ошибки `#NUM!`.
- Если любой аргумент является нечисленным значением, COMBINA возвращает значение ошибки `#VALUE!`.
- Используется следующее уравнение, где N — Число, а M — Number_chosen:

$$\binom{N+M-1}{N-1}$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`= COMBINA(4,3)`	Возвращает количество сочетаний (с повторениями) для 4 и 3.	20
`= COMBINA(10,3)`	Возвращает количество сочетаний (с повторениями) для 10 и 3.	220

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CONFIDENCE.NORM

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Интервал доверия — это диапазон значений. Пример среднего значения x находится в центре этого диапазона, а диапазон — x ± CONFIDENCE.NORM. Например, если x является примером среднего значения времени доставки для продуктов, упорядоченных по почте, x ± CONFIDENCE.NORM является диапазоном средств заполнения. Для любого значения популяции в этом диапазоне вероятность получения образца меньше м0, чем x, больше альфа; для любой совокупности, м0, а не в этом диапазоне, вероятность получения выборки означают дальше от м0, чем x меньше альфа. Иными словами, предположим, что мы используем x, standard_dev и размер для создания двухсторонней проверки на уровне важности альфа гипотезы, что среднее население равно м0. Тогда мы не отклоним этот гипотеза, если μ0 находится в доверительном интервале и отклонит эта гипотеза, если м0 не находится в доверительном интервале. Доверительный интервал не позволяет нам определить, что существует вероятность 1 - альфа, что следующий пакет займет время доставки, которое находится в доверительном интервале.

Синтаксис

```
DAX
CONFIDENCE.NORM(alpha,standard_dev,size)
```

Параметры

Развернуть таблицу

Срок	Определение
`alpha`	Уровень важности, используемый для вычисления уровня достоверности. Уровень достоверности равен 100*(1 - альфа)%, или, другими словами, альфа-значение 0,05 указывает на уровень достоверности на 95 процентов.
`standard_dev`	Стандартное отклонение популяции для диапазона данных и считается известным.

Срок	Определение
`standard_dev,size`	Размер образца.

Возвращаемое значение

Диапазон значений

Замечания

- Если какой-либо аргумент не является числовыми, CONFIDENCE.NORM возвращает значение ошибки `#VALUE!`.
- Если `alpha ≤ 0` или `alpha ≥ 1`, CONFIDENCE.NORM возвращает значение ошибки `#NUM!`.
- Если `standard_dev ≤ 0`, CONFIDENCE.NORM возвращает значение ошибки `#NUM!`.
- Если `size` не является целым числом, округляется.
- Если `size < 1`, CONFIDENCE.NORM возвращает значение ошибки `#NUM!`.
- Если предположим, что альфа равно 0,05, необходимо вычислить область под стандартной нормальной кривой, равной (1 - альфа), или 95 процентов. Это значение ± 1.96. Таким образом, доверительный интервал:

$$\overline{x} \pm 1.96\left(\frac{\sigma}{\sqrt{n}}\right)$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CONFIDENCE.T

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает доверительный интервал для среднего числа населения, используя распределение учащихся.

Синтаксис

DAX

```
CONFIDENCE.T(alpha,standard_dev,size)
```

Параметры

Развернуть таблицу

Срок	Определение
`alpha`	Уровень важности, используемый для вычисления уровня достоверности. Уровень достоверности равен 100*(1 - альфа)%, или, другими словами, альфа-значение 0,05 указывает на уровень достоверности на 95 процентов.
`standard_dev`	Стандартное отклонение популяции для диапазона данных и считается известным.
`size`	Размер образца.

Возвращаемое значение

Возвращает доверительный интервал для среднего числа населения, используя распределение учащихся.

Замечания

- Если какой-либо аргумент не является числовыми, CONFIDENCE.T возвращает значение ошибки `#VALUE!`.

- Если `alpha ≤ 0` или `alpha ≥ 1`, CONFIDENCE.T возвращает значение ошибки `#NUM!`.
- Если `standard_dev ≤ 0`, CONFIDENCE.T возвращает значение ошибки `#NUM!`.
- Если размер не является целым числом, округляется.
- Если размер равен 1, CONFIDENCE.T возвращает #DIV/0! значение ошибки.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Развернуть таблицу

Формула	Описание	Результат
`=CONFIDENCE.T(0.05,1,50)`	Доверительный интервал для среднего числа населения на основе размера выборки 50, с уровнем важности 5% и стандартным отклонением от 1. Это основано на нераспространителях учащегося.	0.284196855

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EXPON.DIST

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает экспоненциальное распределение. Используйте EXPON.DIST для моделирования времени между событиями, например, как долго автоматизированный банковский помощник принимает для доставки денежных средств. Например, можно использовать EXPON.DIST для определения вероятности, что процесс занимает не более 1 минуты.

Синтаксис

```DAX
EXPON.DIST(x,lambda,cumulative)
```

Параметры

Развернуть таблицу

Срок	Определение
`x`	Обязательно. Значение функции.
`lambda`	Обязательно. Значение параметра.
`cumulative`	Обязательно. Логическое значение, указывающее, какая форма экспоненциальной функции требуется предоставить. Если накопительный `TRUE`, EXPON.DIST возвращает накопительную функцию распределения; Если `FALSE`, он возвращает функцию плотности вероятности.

Возвращаемое значение

Возвращает экспоненциальное распределение.

Замечания

- Если `x` или `lambda` не является числовыми, EXPON.DIST возвращает значение ошибки `#VALUE!`.
- Если `x` или `lambda` не является целым числом, округляется.
- Если `x` < 0, EXPON.DIST возвращает значение ошибки `#NUM!`.
- Если `lambda` ≤ 0, EXPON.DIST возвращает значение ошибки `#NUM!`.
- Уравнение для функции плотности вероятности:

$$f(x;\lambda) = \lambda e^{-\lambda x}$$

- Уравнение для накопительной функции распределения:

$$F(x;\lambda) = 1 - e^{-\lambda x}$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

GEOMEAN

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает геометрическое среднее числа в столбце.

Чтобы вернуть геометрическое среднее выражение, вычисляемое для каждой строки таблицы, используйте GEOMEANX функцию.

Синтаксис

```DAX
GEOMEAN(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, содержащий числа, для которых вычисляется геометрическое среднее значение.

Возвращаемое значение

Десятичное число.

Замечания

- Учитываются только числа в столбце. Пустые, логические значения и текст игнорируются.
- GEOMEAN(таблица[столбец]) эквивалентен GEOMEANX(таблица, таблица[столбец])
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на

уровне строк (RLS).

Пример

В следующей таблице вычисляется геометрическое значение столбца Return в таблице "Инвестиции".

```
DAX

= GEOMEAN( Investment[Return] )
```

Связанное содержимое

GEOMEANX функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

GEOMEANX

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает геометрическое значение выражения, вычисляемого для каждой строки таблицы.

Чтобы вернуть геометрическое среднее числа в столбце, используйте функцию GEOMEAN.

Синтаксис

```
DAX

GEOMEANX(<table>, <expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение, вычисляемого для каждой строки таблицы.

Возвращаемое значение

Десятичное число.

Замечания

- Функция GEOMEANX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Второй аргумент — это столбец, содержащий числа, для которых требуется вычислить геометрическое среднее значение, или выражение, которое вычисляет столбец.

- Учитываются только числа в столбце. Пустые, логические значения и текст игнорируются.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В таблице "Инвестиции" вычисляется геометрическое среднее столбца ReturnPct:

DAX

```
= GEOMEANX( Investments, Investments[ReturnPct] + 1 )
```

Связанное содержимое

GEOMEAN функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LINEST

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Использует метод "Наименьшие квадраты" для вычисления прямой линии, которая лучше всего соответствует заданным данным, а затем возвращает таблицу, описывающую линию. Уравнение для линии состоит из формы: y = **Наклон1***x1 + **Наклон2***x2 + ... + **Перехват**.

Синтаксис

```
DAX

LINEST ( <columnY>, <columnX>[, …][, <const>] )
```

Параметры

Развернуть таблицу

Срок	Определение
`columnY`	Столбец известных значений y. Должен иметь скалярный тип.
`columnX`	Столбцы известных значений x. Должен иметь скалярный тип. Необходимо предоставить хотя бы один.
`const`	(Необязательно) Константное значение, указывающее, следует ли принудительно перехватывать равным 0.Если или опущен, значение перехвата вычисляется нормально; Если , значение перехвата перехвата равно нулю.

Возвращаемое значение

Таблица с одной строкой, описывающая строку, а также дополнительную статистику. Это доступные столбцы:

- **Наклон1**, **Наклон2**, ..., **НаклонN**: коэффициенты, соответствующие каждому значению x;
- **Перехват**: перехват значения;
- **StandardErrorSlope1**, **StandardErrorSlope2**, ..., **StandardErrorSlopeN**: стандартные значения ошибок для коэффициентов **Наклон1**, **Наклон2**, ..., **НаклонN**;
- **StandardErrorIntercept**: стандартное значение ошибки для константы **Перехват**;
- **коэффициентOfDetermination**: коэффициент определения (r 2). Сравнивает предполагаемые и фактические значения y и диапазоны значений от 0 до 1: чем выше значение, тем выше корреляция в образце;
- **StandardError**: стандартная ошибка для оценки y;
- **FStatistic**: статистика F или наблюдаемое значение F. Используйте статистику F, чтобы определить, возникает ли наблюдаемая связь между зависимыми и независимыми переменными случайно;
- **DegreesOfFreedom**: степень свободы. Используйте это значение, чтобы помочь найти критически важные значения F в статистической таблице и определить уровень достоверности для модели;
- **регрессииSumOfSquares**: сумма регрессии квадратов;

- **Остаточные**ОфСварес : остаточная сумма квадратов.

Замечания

`columnY` и все `columnX` должны принадлежать одной таблице.

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE LINEST(
    'FactInternetSales'[SalesAmount],
    'FactInternetSales'[TotalProductCost]
)
```

Возвращает таблицу с одной строкой с десятью столбцами:

Развернуть таблицу

Наклон1	Перехватывать	StandardErrorSlope1	StandardErrorIntercept	КоэффициентOfDetermination
1.67703250456677	6.34550460373026	0.000448675725548806	0.279131821917317	0.995695557281456

Развернуть таблицу

StandardError	FStatistic	DegreesOfFreedom	РегрессияSumOfSquares	ОстаткиumOfSquares
60.9171030357485	13970688.6139993	60396	51843736761.658	224123120.339218

- **Наклон1** и **Перехват**: коэффициенты вычисляемой линейной модели;
- **StandardErrorSlope1** и **StandardErrorIntercept**: стандартные значения ошибок для приведенных выше коэффициентов;
- **КоэффициентOfDetermination**, **StandardError**, **FStatistic**, **DegreesOfFreedom**, **РегрессионSumOfSquares** и **ОстатокSumOfSquares**: статистика регрессии по модели.

Для данной интернет-продажи эта модель прогнозирует сумму продажи по следующей формуле:

```
SalesAmount = Slope1 * TotalProductCost + Intercept
```

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE LINEST(
    'DimCustomer'[TotalSalesAmount],
    'DimCustomer'[YearlyIncome],
```

```
    'DimCustomer'[TotalChildren],
    'DimCustomer'[BirthDate]
)
```

Возвращает таблицу с одной строкой с четырнадцатью столбцами:

- Наклон1
- Наклон2
- Наклон3
- Перехватывать
- StandardErrorSlope1
- StandardErrorSlope2
- StandardErrorSlope3
- StandardErrorIntercept
- КоэффициентOfDetermination
- StandardError
- FStatistic
- DegreesOfFreedom
- РегрессияSumOfSquares
- ОстаткиumOfSquares

Для данного клиента эта модель прогнозирует общий объем продаж по следующей формуле (дата рождения автоматически преобразуется в число):

```
TotalSalesAmount = Slope1 * YearlyIncome + Slope2 * TotalChildren + Slope3 * BirthDate +
Intercept
```

Связанное содержимое

LINESTX Статистические функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LINESTX

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Использует метод "Наименьшие квадраты" для вычисления прямой линии, которая лучше всего соответствует заданным данным, а затем возвращает таблицу, описывающую линию. Результат данных из выражений, вычисляемых для каждой строки в таблице. Уравнение для линии состоит из формы: y = **Наклон1***x1 + **Наклон2***x2 + ... + **Перехват**.

Синтаксис

```
DAX

LINESTX ( <table>, <expressionY>, <expressionX>[, …][, <const>] )
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будут оцениваться выражения.
`expressionY`	Выражение, вычисляемого для каждой строки таблицы, для получения известных значений y. Должен иметь скалярный тип.
`expressionX`	Выражения, которые необходимо оценить для каждой строки таблицы, для получения известных x-значений. Должен иметь скалярный тип. Необходимо предоставить хотя бы один.
`const`	(Необязательно) Константное значение, указывающее, следует ли принудительно перехватывать равным 0.Если или опущен, значение перехвата вычисляется нормально; Если , значение перехвата перехвата равно нулю.

Возвращаемое значение

Таблица с одной строкой, описывающая строку, а также дополнительную статистику. Это доступные столбцы:

- **Наклон1**, **Наклон2**, ..., **НаклонN**: коэффициенты, соответствующие каждому значению x;
- **Перехват**: перехват значения;
- **StandardErrorSlope1**, **StandardErrorSlope2**, ..., **StandardErrorSlopeN**: стандартные значения ошибок для коэффициентов **Наклон1**, **Наклон2**, ..., **НаклонN**;
- **StandardErrorIntercept**: стандартное значение ошибки для константы **Перехват**;
- **коэффициентOfDetermination**: коэффициент определения (r 2). Сравнивает предполагаемые и фактические значения y и диапазоны значений от 0 до 1: чем выше значение, тем выше

корреляция в образце;

- **StandardError**: стандартная ошибка для оценки y;
- **FStatistic**: статистика F или наблюдаемое значение F. Используйте статистику F, чтобы определить, возникает ли наблюдаемая связь между зависимыми и независимыми переменными случайно;
- **DegreesOfFreedom**: степень свободы. Используйте это значение, чтобы помочь найти критически важные значения F в статистической таблице и определить уровень достоверности для модели;
- **регрессииSumOfSquares**: сумма регрессии квадратов;
- **Остаточные**ОфСварес : остаточная сумма квадратов.

Пример 1

Следующий запрос DAX:

```
DAX

DEFINE VAR TotalSalesByRegion = SUMMARIZECOLUMNS(
    'Sales Territory'[Sales Territory Key],
    'Sales Territory'[Population],
    "Total Sales", SUM(Sales[Sales Amount])
)
EVALUATE LINESTX(
    'TotalSalesByRegion',
    [Total Sales],
    [Population]
)
```

Возвращает таблицу с одной строкой с десятью столбцами:

Развернуть таблицу

Наклон1	Перехватывать	StandardErrorSlope1	StandardErrorIntercept	КоэффициентOfDetermination
6.42271517588	-410592.76216	0.24959467764561	307826.343996223	0.973535860750193

Развернуть таблицу

StandardError	FStatistic	DegreesOfFreedom	РегрессияSumOfSquares	ОстаткиumOfSquares
630758.1747292	662.165707642	18	263446517001130	7161405749781.07

- **Наклон1** и **Перехват**: коэффициенты вычисляемой линейной модели;
- **StandardErrorSlope1** и **StandardErrorIntercept**: стандартные значения ошибок для приведенных выше коэффициентов;
- **КоэффициентOfDetermination**, **StandardError**, **FStatistic**, **DegreesOfFreedom**, **РегрессионSumOfSquares** и **ОстатокSumOfSquares**: статистика регрессии по модели.

Для данной территории продаж эта модель прогнозирует общий объем продаж по следующей формуле:

```
Total Sales = Slope1 * Population + Intercept
```

Пример 2

Следующий запрос DAX:

```DAX
DEFINE VAR TotalSalesByCustomer = SUMMARIZECOLUMNS(
    'Customer'[Customer ID],
    'Customer'[Age],
    'Customer'[NumOfChildren],
    "Total Sales", SUM(Sales[Sales Amount])
)
EVALUATE LINESTX(
    'TotalSalesByCustomer',
    [Total Sales],
    [Age],
    [NumOfChildren]
)
```

Возвращает таблицу с одной строкой с двенадцатью столбцами:

Развернуть таблицу

Наклон1	Наклон2	Перехватывать	StandardErrorSlope1
69.0435458093763	33.005949841721	-871.118539339539	0.872588875481658

Развернуть таблицу

StandardErrorSlope2	StandardErrorIntercept	КоэффициентOfDetermination	StandardError
6.21158863903435	26.726292527427	0.984892920482022	68.5715034014342

Развернуть таблицу

FStatistic	DegreesOfFreedom	РегрессияSumOfSquares	ОстаткиumOfSquares
3161.91535144391	97	29734974.9782379	456098.954637092

Для данного клиента эта модель прогнозирует общий объем продаж по следующей формуле:

```
Total Sales = Slope1 * Age + Slope2 * NumOfChildren + Intercept
```

Связанное содержимое

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MEDIAN

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает медиану чисел в столбце.

Чтобы вернуть медиану выражения, вычисляемую для каждой строки в таблице, используйте функцию MEDIANX.

Синтаксис

```DAX
MEDIAN(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, содержащий числа, для которых вычисляется медиана.

Возвращаемое значение

Десятичное число.

Замечания

- Учитываются только числа в столбце. Пробелы игнорируются. Логические значения, даты и текст не поддерживаются.
- MEDIAN(таблица[столбец] эквивалентен MEDIANX(таблица, таблица[столбец]).
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Ниже вычисляется медиана столбца с именем Age в таблице с именем Customers:

```
DAX
```

```
= MEDIAN( Customers[Age] )
```

Связанное содержимое

MEDIANX функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MEDIANX

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает медианное число выражения, вычисляемого для каждой строки в таблице.

Чтобы вернуть медиану чисел в столбце, используйте функцию MEDIAN.

Синтаксис

DAX

```
MEDIANX(<table>, <expression>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение, вычисляемого для каждой строки таблицы.

Возвращаемое значение

Десятичное число.

Замечания

- Функция MEDIANX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Второй аргумент — это столбец, содержащий числа, для которых требуется вычислить медиану или выражение, которое вычисляет столбец.
- Учитываются только числа в столбце.

- Логические значения и текст игнорируются.
- MEDIANX не игнорирует пустые значения; однако MEDIAN игнорирует пустые
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Ниже вычислен средний возраст клиентов, живущих в США.

DAX

```
= MEDIANX( FILTER(Customers, RELATED( Geography[Country]="USA" ) ),
Customers[Age] )
```

Связанное содержимое

MEDIAN функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NORM.DIST

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает нормальное распределение для указанного среднего и стандартного отклонения.

Синтаксис

```DAX
NORM.DIST(X, Mean, Standard_dev, Cumulative)
```

Параметры

Развернуть таблицу

Срок	Определение
`X`	Значение, для которого требуется распределение.
`Mean`	Арифметическое среднее распределения.
`Standard_dev`	Стандартное отклонение распределения.
`Cumulative*`	Логическое значение, определяющее форму функции. Если накопительный `TRUE`, NORM.DIST возвращает накопительную функцию распределения; Если `FALSE`, он возвращает функцию плотности вероятности.

Возвращаемое значение

Обычное распределение для указанного среднего и стандартного отклонения.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

```
DAX
EVALUATE { NORM.DIST(42, 40, 1.5, TRUE) }
```

Возвращает

Развернуть таблицу

[значение]
0.908788780274132

Связанное содержимое

функция функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NORM.INV

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Обратное нормальное совокупное распределение для указанного среднего и стандартного отклонения.

Синтаксис

```
DAX

NORM.INV(Probability, Mean, Standard_dev)
```

Параметры

Развернуть таблицу

Срок	Определение
`Probability`	Вероятность, соответствующая нормальному распределению.
`Mean`	Арифметическое среднее распределения.
`Standard_dev`	Стандартное отклонение распределения.

Возвращаемое значение

Возвращает обратное нормальное совокупное распределение для указанного среднего и стандартного отклонения.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

DAX

```
EVALUATE { NORM.INV(0.908789, 40, 1.5) }
```

Возвращает

Развернуть таблицу

[значение]
42.00000200956628780274132

Связанное содержимое

функция функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NORM.S.DIST

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает стандартное нормальное распределение (имеет среднее значение нуля и стандартное отклонение одного).

Синтаксис

```
DAX

NORM.S.DIST(Z, Cumulative)
```

Параметры

Развернуть таблицу

Срок	Определение
`Z`	Значение, для которого требуется распределение.
`Cumulative`	Накопительный — это логическое значение, определяющее форму функции. Если накопительный `TRUE`, NORM.S.DIST возвращает накопительную функцию распределения; Если `FALSE`, он возвращает функцию плотности вероятности.

Возвращаемое значение

Стандартное нормальное распределение (имеет среднее значение нуля и стандартное отклонение одного.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

DAX

```
EVALUATE { NORM.S.DIST(1.333333, TRUE) }
```

Возвращает

Развернуть таблицу

[значение]
0.908788725604095

Связанное содержимое

функция функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NORM.S.INV

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает обратное стандартное нормальное накопительное распределение. Распределение имеет среднее значение нуля и стандартное отклонение одного.

Синтаксис

```
DAX
NORM.S.INV(Probability)
```

Параметры

Развернуть таблицу

Срок	Определение
`Probability`	Вероятность, соответствующая нормальному распределению.

Возвращаемое значение

Обратное стандартное нормальное накопительное распределение. Распределение имеет среднее значение нуля и стандартное отклонение одного.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

```
DAX
```

```
EVALUATE { NORM.S.INV(0.908789) }
```

Возвращает

Развернуть таблицу

[значение]
1.33333467304411

Связанное содержимое

функция функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PERCENTILE.EXC

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает процентиль k-th значений в диапазоне, где k находится в диапазоне 0,1, монопольный.

Чтобы вернуть процентиль выражения, вычисляемого для каждой строки таблицы, используйте функцию PERCENTILEX.EXC.

Синтаксис

```DAX
PERCENTILE.EXC(<column>, <k>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, содержащий значения, определяющие относительную стоящую.
`k`	Значение процентиля в диапазоне 0.1, эксклюзивное.

Возвращаемое значение

K-th процентиль значений в диапазоне, где k находится в диапазоне 0,1, эксклюзивный.

Замечания

- Если столбец пуст, возвращается BLANK().
- Если k равен нулю или пусто, процентиль ранг 1/(n+1) возвращает наименьшее значение. Если ноль, он выходит за пределы диапазона и возвращается ошибка.

- Если k не является числом или за пределами диапазона 0 до 1, возвращается ошибка.
- Если k не является нескольким из 1/(n + 1), PERCENTILE.EXC интерполяется, чтобы определить значение на процентиле k-th.
- PERCENTILE.EXC интерполяется, если значение указанного процентиля находится между двумя значениями в массиве. Если он не может интерполироваться для указанного процента k, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

PERCENTILEX.EXC

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PERCENTILE.INC

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает процентиль k-th значений в диапазоне, где k находится в диапазоне 0.1, включительно.

Чтобы вернуть процентиль выражения, вычисляемого для каждой строки в таблице, используйте PERCENTILEX.INC.

Синтаксис

```DAX
PERCENTILE.INC(<column>, <k>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, содержащий значения, определяющие относительную стоящую.
`k`	Значение процентиля в диапазоне 0.1 включительно.

Возвращаемое значение

K-th процентиль значений в диапазоне, где k находится в диапазоне 0.1, включительно.

Замечания

- Если столбец пуст, возвращается BLANK().
- Если k равен нулю или пусто, процентиль ранг 1/(n+1) возвращает наименьшее значение. Если ноль, он выходит за пределы диапазона и возвращается ошибка.

- Если k не является числом или за пределами диапазона 0 до 1, возвращается ошибка.
- Если k не является нескольким из 1/(n + 1), PERCENTILE.INC интерполяется, чтобы определить значение на процентиле k-th.
- PERCENTILE.INC интерполяется, если значение указанного процентиля находится между двумя значениями в массиве. Если он не может интерполироваться для указанного процента k, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

PERCENTILEX.INC

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PERCENTILEX.EXC

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает процентиль выражения, вычисляемого для каждой строки в таблице.

Чтобы вернуть процентиль чисел в столбце, используйте функцию PERCENTILE.EXC.

Синтаксис

```
DAX

PERCENTILEX.EXC(<table>, <expression>, k)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение, вычисляемого для каждой строки таблицы.
`k`	Требуемое значение процентиля в диапазоне от 0 до 1 монопольное.

Возвращаемое значение

Процентильное число выражения, вычисляемого для каждой строки в таблице.

Замечания

- Если k равен нулю или пусто, процентиль ранг 1/(n+1) возвращает наименьшее значение. Если ноль, он выходит за пределы диапазона и возвращается ошибка.
- Если k не является числом или за пределами диапазона 0 до 1, возвращается ошибка.

- Если k не является нескольким из 1/(n + 1), PERCENTILEX.EXC интерполяется, чтобы определить значение на процентиле k-th.
- PERCENTILEX.EXC интерполяется, если значение указанного процентиля находится между двумя значениями в массиве. Если он не может интерполироваться для указанного процента k, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

PERCENTILE.EXC

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PERCENTILEX.INC

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает процентиль выражения, вычисляемого для каждой строки в таблице.

Чтобы вернуть процентиль чисел в столбце, используйте PERCENTILE.INC.

Синтаксис

```
DAX

PERCENTILEX.INC(<table>, <expression>;, k)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение, вычисляемого для каждой строки таблицы.
`k`	Требуемое значение процентиля в диапазоне от 0 до 1 включительно.

Возвращаемое значение

Процентильное число выражения, вычисляемого для каждой строки в таблице.

Замечания

- Если k равен нулю или пусто, процентиль ранг 1/(n - 1) возвращает наименьшее значение. Если ноль, он выходит за пределы диапазона и возвращается ошибка.
- Если k не является числом или за пределами диапазона 0 до 1, возвращается ошибка.

- Если k не равен 1/(n - 1), PERCENTILEX.EXC интерполяется, чтобы определить значение на процентиль k-th.
- PERCENTILEX.INC интерполяется, если значение указанного процентиля находится между двумя значениями в массиве. Если он не может интерполироваться для указанного процента k, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

PERCENTILE.INC

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PERMUT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает количество переметов для заданного количества объектов, которые можно выбрать из числовых объектов. Перемещение — это любой набор или подмножество объектов или событий, в которых имеется значительный внутренний порядок. Перемутации отличаются от сочетаний, для которых внутренний порядок не является значительным. Используйте эту функцию для вычислений вероятностей в стиле лотереи.

Синтаксис

DAX

```
PERMUT(number, number_chosen)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Обязательно. Целое число, описывающее количество объектов.
`number_chosen`	Обязательно. Целое число, описывающее количество объектов в каждой перемутации.

Возвращаемое значение

Возвращает количество перемутов для заданного количества объектов, которые можно выбрать из числовых объектов.

Замечания

- Оба аргумента усечены в целые числа.

- Если число или number_chosen не является числом, PERMUT возвращает значение ошибки `#VALUE!`.
- Если число ≤ 0 или значение number_chosen < 0, PERMUT возвращает значение ошибки `#NUM!`.
- Если число < number_chosen, PERMUT возвращает значение ошибки `#NUM!`.
- Уравнение для числа перемутов:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующей формуле перемутации можно использовать для группы из 3 объектов, где выбрано 2:

```
DAX
= PERMUT(3,2)
```

Результат

6

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

POISSON.DIST

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает распределение Poisson. Общее применение распределения Poisson прогнозирует количество событий в течение определенного времени, таких как количество автомобилей, прибывающих на платный плаза в 1 минуту.

Синтаксис

```DAX
POISSON.DIST(x,mean,cumulative)
```

Параметры

Развернуть таблицу

Срок	Определение
`x`	Обязательно. Количество событий.
`mean`	Обязательно. Ожидаемое числовое значение.
`cumulative`	Обязательно. Логическое значение, определяющее форму возвращаемого распределения вероятностей. Если накопительный `TRUE`, POISSON.DIST возвращает совокупную вероятность Poisson, что число случайных событий будет от нуля до x включительно; Если `FALSE`, он возвращает массовую функцию вероятности Poisson, что число событий, происходящих, будет точно x.

Возвращаемое значение

Возвращает распределение Poisson.

Замечания

- Если x не является целым числом, округляется.

- Если значение x или среднее не является числом, POISSON.DIST возвращает значение ошибки `#VALUE!`.
- Если x < 0, POISSON.DIST возвращает значение ошибки `#NUM!`.
- Если среднее значение < 0, POISSON.DIST возвращает значение ошибки `#NUM!`.
- POISSON.DIST вычисляется следующим образом.
  - Для накопительных = `FALSE`:

    $$\text{POISSON} = \frac{e^{-\lambda} \lambda^{x}{x!}$$

  - Для накопительных = `TRUE`:

$$\text{CUMPOISSON} = \sum_{k=0}^{x} \frac{e^{-\лямбда} \lambda^{x}}{k!}$$

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RANK.EQ

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает рейтинг числа в списке чисел.

Синтаксис

DAX

```
RANK.EQ(<value>, <columnName>[, <order>])
```

Параметры

⛶ Развернуть таблицу

Срок	Определение
`value`	Любое DAX выражение, возвращающее одно скалярное значение, ранг которого необходимо найти. Выражение должно быть вычисляется ровно один раз перед вычислением функции, а значение передается в список аргументов.
`columnName`	Имя существующего столбца, для которого будут определены ряды. Это не может быть выражение или столбец, созданный с помощью этих функций: `ADDCOLUMNS`, `ROW` или `SUMMARIZE`.
`order`	(Необязательно) Значение, указывающее, как ранжировать *числах*, низкий до высокого или высокого уровня:

Значения заказа

⛶ Развернуть таблицу

ценность	альтернативное значение	Описание
`0` (ноль)	`FALSE`	Ранжирует в порядке убывания `columnName`. Если `value` равно наибольшему числу в `columnName` то `RANK.EQ` равно 1.

ценность	альтернативное значение	Описание
`1`	`TRUE`	Ранжирует в порядке возрастания `columnName`. Если `value` равно наименьшему числу в `columnName` то `RANK.EQ` равно 1.

Возвращаемое значение

Число, указывающее ранг `value` среди чисел в `columnName`.

Замечания

- `columnName` не может ссылаться на любой столбец, созданный с помощью этих функций: ADDCOLUMNS, ROW или SUMMARIZE. Я
- Если `value` не находится в `columnName` или значении является пустым, *RANK.EQ* возвращает пустое значение.
- Повторяющиеся значения `value` получают то же значение ранжирования; Следующее значение ранга, назначенное, будет значением ранжирования и числом повторяющихся значений. Например, если пять (5) значений привязаны к рангу 11, то следующее значение получит ранг 16 (11 + 5).
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

В следующем примере создается вычисляемый столбец, который ранжирует значения в SalesAmount_USD из таблицы *InternetSales_USD* по всем числам в одном столбце.

DAX

```
= RANK.EQ(InternetSales_USD[SalesAmount_USD],
InternetSales_USD[SalesAmount_USD])
```

Пример 2

В следующем примере выполняется подмножество значений по заданному образцу. Предположим, что у вас есть таблица местных студентов с их производительностью в определенном национальном тесте, а также у вас есть весь набор показателей в этом национальном тесте. Следующий вычисляемый столбец даст вам национальный рейтинг для каждого из местных студентов.

DAX

```
= RANK.EQ(Students[Test_Score], NationalScores[Test_Score])
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RANKX

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает ранжирование числа в списке чисел для каждой строки в аргументе `table`.

Синтаксис

DAX

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

Параметры

`table` любое выражение DAX, которое возвращает таблицу данных, по которым вычисляется выражение.

`expression` любое выражение DAX, возвращающее одно скалярное значение. Выражение вычисляется для каждой строки `table`, чтобы создать все возможные значения для ранжирования. Ознакомьтесь с разделом примечаний, чтобы понять поведение функции, когда `expression` оценивается BLANK.

`value` (необязательно) Любое выражение DAX, которое возвращает одно скалярное значение, ранг которого необходимо найти. Ознакомьтесь с разделом примечаний, чтобы понять поведение функции, если `value` не найден в выражении.

Если параметр `value` опущен, вместо него используется значение выражения в текущей строке.

`order` (необязательно) Значение, указывающее, как ранжировать `value`, низкую до высокой или высокой:

Развернуть таблицу

ценность	альтернативное значение	Description
0 (ноль)	`FALSE`	Ранжирует в порядке убывания значений выражения. Если значение равно наибольшему числу в выражении, RANKX возвращает 1. Это значение по умолчанию при опущении параметра заказа.
1	`TRUE`	Ранжирует по возрастанию выражения. Если значение равно наименьшему числу в выражении, RANKX возвращает 1.

`ties` (необязательно) Перечисление, определяющее способ определения ранжирования при наличии связей.

Развернуть таблицу

перечисление	Description
`Skip`	Следующее значение ранжирования после галстука — это значение ранга галстука плюс количество связанных значений. Например, если пять (5) значений привязаны к рангу 11, то следующее значение получит ранг 16 (11 + 5). Это значение по умолчанию при опущении параметра `ties`.
`Dense`	Следующее значение ранжирования после галстука — это следующее значение ранжирования. Например, если пять (5) значений привязаны к рангу 11, то следующее значение получит ранг 12.

Возвращаемое значение

Число рангов `value` среди всех возможных значений `expression` вычисляется для всех строк `table` чисел.

Замечания

- Если `expression` или `value` оценивается как BLANK она обрабатывается как 0 (ноль) для всех выражений, которые приводят к числу или как пустой текст для всех текстовых выражений.

- Если `value` не является одним из всех возможных значений `expression`, RANKX временно добавляет `value` к значениям из `expression` и повторно оценивает RANKX, чтобы определить соответствующий ранг `value`.
- Необязательные аргументы могут быть пропущены путем размещения пустой запятой (,) в списке аргументов, например `RANKX(Inventory, [InventoryCost],,,"Dense")`.
- Сравнения с равенством (`=`, `<`, `>`, `>=` и `<=`) между значениями с типом данных *десятичного числа* могут потенциально возвращать непредвиденные результаты при использовании функции RANKX. Неправильные результаты могут возникать, так как значения с *десятичным числом* тип данных хранятся в виде чисел с плавающей запятой СТАНДАРТА IEEE 754 и имеют ограничения в их точности. Чтобы избежать непредвиденных результатов, измените тип данных на *фиксированное десятичное число* или выполните принудительное округление с помощью ROUND.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий вычисляемый столбец в таблице "Продукты" вычисляет рейтинг продаж для каждого продукта в канале Интернета.

DAX

```
= RANKX(ALL(Products), SUMX(RELATEDTABLE(InternetSales), [SalesAmount]))
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SAMPLE

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает пример N строк из указанной таблицы.

Синтаксис

DAX

```
SAMPLE(<n_value>, <table>, <orderBy_expression>, [<order>[,
<orderBy_expression>, [<order>]]…])
```

Параметры

Развернуть таблицу

Термин	Определение
`n_value`	Количество возвращаемых строк. Это любое выражение DAX, которое возвращает одно скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста). Если введено не целочисленное значение (или выражение), результат приводится в виде целого числа.
`table`	Любое выражение DAX, возвращающее таблицу данных, из которой извлекаются примеры строк n.
`orderBy_expression`	Любое скалярное DAX выражение, где значение результата вычисляется для каждой строки `table`.
`order`	(Необязательно) Значение, указывающее, как сортировать `orderBy_expression` значения, возрастание или убывание: 0 (ноль), сортирует по убыванию значений `order_by`. 1, ранжирует в порядке возрастания `order_by`.

Возвращаемое значение

Таблица, состоящая из образца строк N `table` или пустой таблицы, если `n_value` равно 0 (ноль) или меньше. Если указаны аргументы OrderBy, пример будет

стабильным и детерминированным, возвращая первую строку, последнюю строку и равномерно распределенные строки между ними. Если порядок не указан, выборка будет случайной, не стабильной и не детерминированной.

Замечания

- Если `n_value` равно 0 (ноль) или меньше, SAMPLE возвращает пустую таблицу.
- Чтобы избежать повторяющихся значений в примере, таблица, указанная в качестве второго аргумента, должна сгруппироваться по столбцу, используемому для сортировки.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SAMPLECARTESIANPOINTSBYCOVER

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает пример подмножества из таблицы, полученной путем построения строк в виде точек в 2D-пространстве и удаления перекрывающихся точек.

Синтаксис

```
DAX

SAMPLECARTESIANPOINTSBYCOVER(<Size>, <Table>, <XAxis>, <YAxis>[, <Radius>]
[, <MaxMinRatio>] [, <MaxBlankRatio>] )
```

Параметры

Развернуть таблицу

Срок	Определение
Размер	Количество строк в примере для возврата
Стол	Любое выражение DAX, возвращающее таблицу данных, из которой нужно вернуть образец подмножества.
XAxis	Числовый столбец XAxis из таблицы.
YAxis	Числовой столбец YAxis из таблицы.
Радиус	(Необязательно) Числовый столбец Radius из таблицы.
MaxMinRatio	(Необязательно) При указании радиуса необходимо указать MaxMinRatio и определить соотношение между максимальным и минимальным радиусом отрисованными точками. Дополнительные сведения см. в примечаниях.
MaxBlankRatio	(Необязательно) При указании radius необходимо указать MaxBlankRatio и определить соотношение между максимальными и пустыми радиусами отрисовки. Дополнительные сведения см. в примечаниях.

Возвращаемое значение

Пример из таблицы, состоящий из числа строк размера.

Замечания

- Если размер меньше или равен 0, SAMPLECARTESIANPOINTSBYCOVER возвращает пустую таблицу.
- Если задан радиус, необходимо указать MaxMinRatio и MaxBlankRatio и задать значение, превышающее 1. Если нет, SAMPLECARTESIANPOINTSBYCOVER возвращает ошибку
- Задайте MaxMinRatio максимальным значением radius, разделенным на минимальное значение Radius. Например, если минимальный радиус равен 20, а максимальный радиус равен 60, параметр MaxMinRatio должен иметь значение 60/20 = 3.
- Задайте maxBlankRatio максимальное значение Radius, разделенное радиусом для пустых значений. Например, если максимальный радиус равен 60, а для пустых значений — 6, то maxBlankRatio должен иметь значение 60/6 = 10.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий запрос DAX:

DAX

```
EVALUATE
SAMPLECARTESIANPOINTSBYCOVER (
    1000,
    SELECTCOLUMNS ( Sales, "x", [Unit Price], "y", [Sales Amount] ),
    [x],
    [y]
)
```

Возвращает пример строки 1000 из уникальных точек из таблицы Sales в 2D-пространстве, определяемом ценой единицы в виде оси X и суммы продаж в виде оси Y.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

STDEV.S

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает стандартное отклонение выборки.

Синтаксис

```DAX
STDEV.S(<ColumnName>)
```

Параметры

Развернуть таблицу

Термин	Определение
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX, как правило, полное. Это не может быть выражением.

Возвращаемое значение

Число, представляющее стандартное отклонение выборки.

Исключения

Замечания

- STDEV.S предполагает, что столбец ссылается на выборку населения. Если данные представляют всю популяцию, вычислить стандартное отклонение с помощью STDEV.P.
- STDEV.S использует следующую формулу:

  $\sqrt{[\sum(x - x - \tilde{x})^2/(n-1)]}$

где $\tilde{x}$ — среднее значение x для выборки населения и n — это размер совокупности.

- Пустые строки отфильтровываются из `columnName` и не рассматриваются в вычислениях.
- Ошибка возвращается, если `columnName` содержит менее 2 непустых строк.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показана формула для меры, которая вычисляет стандартное отклонение столбца, SalesAmount_USD, когда таблица InternetSales_USD является выборкой.

DAX

```
= STDEV.S(InternetSales_USD[SalesAmount_USD])
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

STDEV.P

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает стандартное отклонение всего населения.

Синтаксис

```DAX
STDEV.P(<ColumnName>)
```

Параметры

Развернуть таблицу

Термин	Определение
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX, как правило, полное. Это не может быть выражением.

Возвращаемое значение

Число, представляющее стандартное отклонение всего населения.

Замечания

- STDEV.P предполагает, что столбец ссылается на всю популяцию. Если данные представляют образец совокупности, вычислить стандартное отклонение с помощью STDEV.S.
- STDEV.P использует следующую формулу:

  $\sqrt{[\sum(x - \tilde{x})^2/n]}$

  где $\tilde{x}$ — среднее значение x для всей совокупности, а n — это размер совокупности.

- Пустые строки отфильтровываются из `columnName` и не рассматриваются в вычислениях.
- Ошибка возвращается, если `columnName` содержит менее 2 непустых строк
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показана формула для меры, которая вычисляет стандартное отклонение столбца, SalesAmount_USD, когда таблица InternetSales_USD является всей популяцией.

DAX

```
= STDEV.P(InternetSales_USD[SalesAmount_USD])
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

STDEVX.S

Статья • 14.12.2024

Область применения: вычисление вычисляемого столбца вычисляемой таблицы Меры визуального элемента

Возвращает стандартное отклонение выборки.

Синтаксис

DAX

```
STDEVX.S(<table>, <expression>)
```

Параметры

Развернуть таблицу

Термин	Определение
`table`	Любое выражение DAX, возвращающее одно скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста).
`expression`	Любое выражение DAX, возвращающее одно скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста).

Возвращаемое значение

Число со стандартным отклонением выборки.

Исключения

Замечания

- STDEVX. S оценивает `expression` для каждой строки `table` и возвращает стандартное отклонение `expression` при условии, что `table` относится к выборке совокупности. Если `table` представляет всю популяцию, вычислить стандартное отклонение с помощью STDEVX.P.

- STDEVX. S использует следующую формулу:

  $\sqrt{[\sum(x - x - \tilde{x})^2/(n-1)]}$

  где $\tilde{x}$ — среднее значение x для всей совокупности, а n — это размер совокупности.

- Пустые строки отфильтровываются из `columnName` и не рассматриваются в вычислениях.

- Ошибка возвращается, если `columnName` содержит менее 2 непустых строк.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показана формула для вычисляемого столбца, который оценивает стандартное отклонение цены на единицу на продукт для выборки населения, когда формула используется в таблице Product.

DAX

```
= STDEVX.S(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD]
- (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

STDEVX.P

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает стандартное отклонение всего населения.

Синтаксис

DAX

```
STDEVX.P(<table>, <expression>)
```

Параметры

Развернуть таблицу

Термин	Определение
`table`	Любое выражение DAX, возвращающее одно скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста).
`expression`	Любое выражение DAX, возвращающее одно скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста).

Возвращаемое значение

Число, представляющее стандартное отклонение всего населения.

Замечания

- STDEVX.P вычисляет `expression` для каждой строки `table` и возвращает стандартное отклонение выражения, если таблица ссылается на всю популяцию. Если данные в `table` представляют образец совокупности, следует вычислить стандартное отклонение с помощью STDEVX.S.
- STDEVX.P использует следующую формулу:

  $\sqrt{[\sum(x - \tilde{x})^2/n]}$

где $\tilde{x}$ — среднее значение x для всей совокупности, а n — это размер совокупности.

- Пустые строки отфильтровываются из `columnName` и не рассматриваются в вычислениях.
- Ошибка возвращается, если `columnName` содержит менее 2 непустых строк
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показана формула вычисляемого столбца, который вычисляет стандартное отклонение цены на единицу на продукт, если формула используется в таблице *Product*.

DAX

```
= STDEVX.P(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD]
- (InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

T.DIST

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает левое распределение t-распределения студента.

Синтаксис

```
DAX
T.DIST(X,Deg_freedom,Cumulative)
```

Параметры

Развернуть таблицу

Термин	Определение
`X`	Числовое значение, по которому необходимо оценить распределение.
`Deg_freedom`	Целое число, указывающее количество градусов свободы.
`Cumulative`	Логическое значение, определяющее форму функции. Если накопительный `TRUE`, T.DIST возвращает накопительную функцию распределения; Если `FALSE`, он возвращает функцию плотности вероятности.

Возвращаемое значение

Левое распределение t-распределения студента.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

DAX

```
EVALUATE { T.DIST(60, 1, TRUE) }
```

Возвращает

Развернуть таблицу

[значение]
0.994695326367377

Связанное содержимое

T.DIST.2T T.DIST.RT T.INV T.INV.2t

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

T.DIST.2T

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает двухстороннее распределение учащихся.

Синтаксис

DAX

```
T.DIST.2T(X,Deg_freedom)
```

Параметры

Развернуть таблицу

Термин	Определение
`X`	Числовое значение, по которому необходимо оценить распределение.
`Deg_freedom`	Целое число, указывающее количество градусов свободы.

Возвращаемое значение

Двухстороннее распределение студента.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

DAX

```
EVALUATE { T.DIST.2T(1.959999998, 60) }
```

Возвраты

Развернуть таблицу

[значение]
0.054644929975921

Связанное содержимое

T.DIST T.DIST.RT T.INV T.INV.2t

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

T.DIST.RT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает правостороннее распределение учащегося.

Синтаксис

DAX

```
T.DIST.RT(X,Deg_freedom)
```

Параметры

Развернуть таблицу

Термин	Определение
`X`	Числовое значение, по которому необходимо оценить распределение.
`Deg_freedom`	Целое число, указывающее количество градусов свободы.

Возвращаемое значение

Правостороннее распределение учащегося.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

DAX

```
EVALUATE { T.DIST.RT(1.959999998, 60) }
```

Возвраты

Развернуть таблицу

[значение]
0.0273224649879605

Связанное содержимое

T.DIST T.DIST.2T T.INV T.INV.2t

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

T.INV

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает левый хвост обратного распределения t-распределения студента.

Синтаксис

```DAX
T.INV(Probability,Deg_freedom)
```

Параметры

Развернуть таблицу

Термин	Определение
`Probability`	Вероятность, связанная с распределением учащихся.
`Deg_freedom`	Число градусов свободы, с которыми следует охарактеризовать распределение.

Возвращаемое значение

Левый хвост обратного распределения учащегося.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

```DAX
```

```
EVALUATE { T.INV(0.75, 2) }
```

Возвраты

Развернуть таблицу

[значение]
0.816496580927726

Связанное содержимое

T.INV.2T T.DIST T.DIST.2T T.DIST.RT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

T.INV.2T

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает двухстороннее обратное распределение учащихся.

Синтаксис

```
DAX
T.INV.2T(Probability,Deg_freedom)
```

Параметры

Развернуть таблицу

Термин	Определение
`Probability`	Вероятность, связанная с распределением учащихся.
`Deg_freedom`	Число градусов свободы, с которыми следует охарактеризовать распределение.

Возвращаемое значение

Двухстороннее обратное распределение студента.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

```
DAX
```

```
EVALUATE { T.INV.2T(0.546449, 60) }
```

Возвраты

Развернуть таблицу

[значение]
0.606533075825759

Связанное содержимое

T.INV T.DIST T.DIST.2T T.DIST.RT

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

VAR.S

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает дисперсию выборки.

Синтаксис

```DAX
VAR.S(<columnName>)
```

Параметры

Развернуть таблицу

Термин	Определение
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX, как правило, полное. Это не может быть выражением.

Возвращаемое значение

Число с дисперсией выборки.

Замечания

- VAR.S предполагает, что столбец ссылается на выборку населения. Если данные представляют всю популяцию, вычислить дисперсию с помощью VAR.P.
- VAR.S использует следующую формулу:

  $\sum(x - x - \tilde{x})^2/(n-1)$

  где $\tilde{x}$ — среднее значение x для выборки

  и n — это размер совокупности

- Пустые строки отфильтровываются из `columnName` и не рассматриваются в вычислениях.
- Ошибка возвращается, если `columnName` содержит менее 2 непустых строк.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показана формула для меры, которая вычисляет дисперсию столбца SalesAmount_USD из InternetSales_USD для выборки.

DAX

```
= VAR.S(InternetSales_USD[SalesAmount_USD])
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

VAR.P

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает дисперсию всего населения.

Синтаксис

```DAX
VAR.P(<columnName>)
```

Параметры

Развернуть таблицу

Термин	Определение
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX, как правило, полное. Это не может быть выражением.

Возвращаемое значение

Число с дисперсией всего населения.

Замечания

- VAR.P предполагает, что столбец ссылается на всю популяцию. Если данные представляют образец совокупности, вычислить дисперсию с помощью VAR.S.
- VAR.P использует следующую формулу:

  $\sum(x - x - \tilde{x})^2/n$

  где $\tilde{x}$ — среднее значение x для всей совокупности

  и n — это размер совокупности

- Пустые строки отфильтровываются из `columnName` и не рассматриваются в вычислениях.
- Ошибка возвращается, если `columnName` содержит менее 2 непустых строк
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показана формула для меры, которая оценивает дисперсию столбца SalesAmount_USD из таблицы InternetSales_USD для всей совокупности.

DAX

```
= VAR.P(InternetSales_USD[SalesAmount_USD])
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

VARX.S

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает дисперсию выборки.

Синтаксис

DAX

```
VARX.S(<table>, <expression>)
```

Параметры

Развернуть таблицу

Термин	Определение
`table`	Любое выражение DAX, возвращающее таблицу данных.
`expression`	Любое выражение DAX, возвращающее одно скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста).

Возвращаемое значение

Число, представляющее дисперсию выборки.

Замечания

- VARX.S вычисляет `expression` для каждой строки `table` и возвращает дисперсию `expression`; при предположении, что `table` относится к выборке населения. Если `table` представляет всю популяцию, следует вычислить дисперсию с помощью VARX.P.
- VAR.S использует следующую формулу:

  $\sum(x - x - \tilde{x})^2/(n-1)$

где $\tilde{x}$ — среднее значение x для выборки

и n — это размер совокупности

- Пустые строки отфильтровываются из `columnName` и не рассматриваются в вычислениях.
- Ошибка возвращается, если `columnName` содержит менее 2 непустых строк.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показана формула для вычисляемого столбца, который оценивает дисперсию цены на единицу на продукт для выборки населения, если формула используется в таблице Product.

DAX

```
= VARX.S(InternetSales_USD, InternetSales_USD[UnitPrice_USD] –
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

VARX.P

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает дисперсию всего населения.

Синтаксис

DAX

```
VARX.P(<table>, <expression>)
```

Параметры

Развернуть таблицу

Термин	Определение
`table`	Любое выражение DAX, возвращающее таблицу данных.
`expression`	Любое выражение DAX, возвращающее одно скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста).

Возвращаемое значение

Число с дисперсией всего населения.

Замечания

- VARX.P вычисляет><выражения для каждой строки <таблицы> и возвращает дисперсию выражения <> при условии, что <таблица> ссылается на всю популяцию. Если <таблица> представляет образец совокупности, вычислить дисперсию с помощью VARX.S.
- VARX.P использует следующую формулу:

  $\sum(x - x - \tilde{x})^2/n$

где $\tilde{x}$ — среднее значение x для всей совокупности

и n — это размер совокупности

- Пустые строки отфильтровываются из `columnName` и не рассматриваются в вычислениях.
- Ошибка возвращается, если `columnName` содержит менее 2 непустых строк
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показана формула для вычисляемого столбца, который вычисляет дисперсию цены на единицу на продукт, когда формула используется в таблице Product

DAX

```
= VARX.P(InternetSales_USD, InternetSales_USD[UnitPrice_USD] –
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Функции обработки таблиц

Статья • 23.01.2025

Эти функции возвращают таблицу или управляют существующими таблицами.

В этой категории

Развернуть таблицу

Функция	Описание
ADDCOLUMNS	Добавляет вычисляемые столбцы в заданное выражение таблицы или таблицы.
ADDMISSINGITEMS	Добавляет сочетания элементов из нескольких столбцов в таблицу, если они еще не существуют.
CROSSJOIN	Возвращает таблицу, содержащую декартовую продукцию всех строк из всех таблиц в аргументах.
CURRENTGROUP	Возвращает набор строк из аргумента таблицы выражения GROUPBY.
DATATABLE	Предоставляет механизм объявления встроенного набора значений данных.
DETAILROWS	Вычисляет выражение подробных строк, определенное для меры, и возвращает данные.
столбца DISTINCT	Возвращает таблицу с одним столбцом, содержащую различные значения из указанного столбца.
ТАБЛИЦА DISTINCT	Возвращает таблицу, удаляя повторяющиеся строки из другой таблицы или выражения.
EXCEPT	Возвращает строки одной таблицы, которая не отображается в другой таблице.
FILTERS	Возвращает таблицу значений, непосредственно применяемых в качестве фильтров для `columnName`.
GENERATE	Возвращает таблицу с декартовским продуктом между каждой строкой в *таблице1* и таблицей, которая приводит к оценке *таблицы 2* в контексте текущей строки из *таблицы 1*.
GENERATEALL	Возвращает таблицу с декартовским продуктом между каждой строкой в *таблице1* и таблицей, которая приводит к оценке *таблицы 2* в контексте текущей строки из *таблицы 1*.

Функция	Описание
GENERATESERIES	Возвращает одну таблицу столбцов, содержащую значения арифметического ряда.
GROUPBY	Как и функция SUMMARIZE, GROUPBY не выполняет неявную CALCULATE для столбцов расширений, которые он добавляет.
IGNORE	Изменяет SUMMARIZECOLUMNS путем пропуска определенных выражений из вычисления BLANK/NULL.
INTERSECT	Возвращает пересечение строк двух таблиц, сохраняя дубликаты.
NATURALINNERJOIN	Выполняет внутреннее соединение таблицы с другой таблицей.
NATURALLEFTOUTERJOIN	Выполняет присоединение левойtable к rightTable.
ROLLUP	Изменяет поведение SUMMARIZE путем добавления строк свертки в результат для столбцов, определенных параметром groupBy_columnName.
ROLLUPADDISSUBTOTAL	Изменяет поведение SUMMARIZECOLUMNS путем добавления свертки или промежуточных строк в результат на основе столбцов groupBy_columnName.
ROLLUPISSUBTOTAL	Объединяет группы свертки с столбцом, добавленным ROLLUPADDISSUBTOTAL в выражении ADDMISSINGITEMS.
ROLLUPGROUP	Изменяет поведение SUMMARIZE и SUMMARIZECOLUMNS путем добавления строк свертки в результат для столбцов, определенных параметром groupBy_columnName.
ROW	Возвращает таблицу с одной строкой, содержащей значения, полученные из выражений, заданных каждому столбцу.
SELECTCOLUMNS	Добавляет вычисляемые столбцы в заданное выражение таблицы или таблицы.
SUBSTITUTEWITHINDEX	Возвращает таблицу, представляющую левый полусоединен из двух таблиц, предоставленных в качестве аргументов.
SUMMARIZE	Возвращает сводную таблицу для запрошенных итогов по набору групп.
SUMMARIZECOLUMNS	Возвращает сводную таблицу по набору групп.
конструктора таблиц	Возвращает таблицу одного или нескольких столбцов.
TOPN	Возвращает верхние N-строки указанной таблицы.
TREATAS	Применяет результат табличного выражения в качестве фильтров к столбцам из не связанной таблицы.

Функция	Описание
UNION	Создает таблицу объединения (соединения) из пары таблиц.
VALUES	Возвращает таблицу с одним столбцом, содержащую различные значения из указанной таблицы или столбца.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ADDCOLUMNS

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Добавляет вычисляемые столбцы в заданное выражение таблицы или таблицы.

Синтаксис

DAX

```
ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]…)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Любое выражение DAX, возвращающее таблицу данных.
`name`	Имя столбца, заключенное в двойные кавычки.
`expression`	Любое выражение DAX, возвращающее скалярное выражение, вычисляемое для каждой строки `table`.

Возвращаемое значение

Таблица со всеми исходными столбцами и добавленными.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере возвращается расширенная версия таблицы "Категория продукта", которая включает общие значения продаж из канала торгового посредника и интернет-продаж.

DAX

```
ADDCOLUMNS(ProductCategory
                , "Internet Sales", SUMX(RELATEDTABLE(InternetSales_USD), InternetSales_USD[SalesAmount_USD])
                , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
```

В следующей таблице показана предварительная версия данных по мере получения любой функцией, ожидающей получения таблицы:

Развернуть таблицу

ProductCategory[ProductCategoryName]	ProductCategory[ProductCategoryAlternateKey]	ProductCategory[ProductCategoryKey]	[Интернет-продажи]	[Прс торг поср
Велосипеды	1	1	25107749.77	6308
Компоненты	2	2		1120
Одежда	3	3	306157.5829	1669
Принадлежности	4	4	640920.1338	5343

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ADDMISSINGITEMS

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Добавляет строки с пустыми значениями в таблицу, возвращаемую SUMMARIZECOLUMNS.

Синтаксис

DAX

```
ADDMISSINGITEMS ( [<showAll_columnName> [, <showAll_columnName> [, … ] ] ],
<table> [, <groupBy_columnName> [, [<filterTable>] [, <groupBy_columnName>
[, [<filterTable>] [, … ] ] ] ] ] ] )
```

Параметры

Развернуть таблицу

Срок	Определение
`showAll_columnName`	(Необязательно) Столбец, для которого возвращаются элементы без данных для используемых мер. Если он не указан, возвращаются все столбцы.
`table`	Таблица SUMMARIZECOLUMNS.
`groupBy_columnName`	(Необязательно) Столбец для группировки по указанному аргументу таблицы.
`filterTable`	(Необязательно) Табличное выражение, определяющее, какие строки возвращаются.

Возвращаемое значение

Таблица с одним или несколькими столбцами.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

C SUMMARIZECOLUMNS

Таблица, возвращаемая SUMMARIZECOLUMNS, будет содержать только строки со значениями. Упаковав выражение SUMMARIZECOLUMNS в выражение ADDMISSINGITEMS, строки, содержащие значения, также не возвращаются.

Пример

Без ADDMISSINGITEMSследующий запрос:

```
DAX
SUMMARIZECOLUMNS(
    'Sales'[CustomerId],
    "Total Qty", SUM ( Sales[TotalQty] )
)
```

Возвращает

Развернуть таблицу

CustomerId	TotalQty
A	5
B	3
C	3
E	2

При использовании ADDMISSINGITEMSследующий запрос:

```
DAX
EVALUATE
ADDMISSINGITEMS (
    'Sales'[CustomerId],
    SUMMARIZECOLUMNS(
        'Sales'[CustomerId],
        "Total Qty", SUM ( Sales[TotalQty] )
    ),
```

```
    'Sales'[CustomerId]
)
```

Возвращает

Развернуть таблицу

CustomerId	TotalQty
A	5
B	3
C	3
D	
E	2
F	

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CROSSJOIN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу, содержащую декартовую продукцию всех строк из всех таблиц в аргументах. Столбцы в новой таблице — это все столбцы во всех таблицах аргументов.

Синтаксис

DAX

```
CROSSJOIN(<table>, <table>[, <table>]…)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Любое выражение DAX, возвращающее таблицу данных

Возвращаемое значение

Таблица, содержащая декартовую продукцию всех строк из всех таблиц в аргументах.

Замечания

- Имена столбцов из `table` аргументов должны отличаться во всех таблицах или возвращается ошибка.
- Общее количество строк, возвращаемых CROSSJOIN() равно продукту числа строк из всех таблиц в аргументах; Кроме того, общее количество столбцов в результирующих таблицах — это сумма количества столбцов во всех таблицах. Например, если **TableA** имеет **строки rA** и столбцы **cA**, а **tableBстрок rB** и **столбцов cB**, **TableC** имеет **строки rC** и **столбцы cC**; Затем

результирующая таблица содержит **rA × rb × rC** строк **и столбцов cA + cB + cC**.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показаны результаты применения CROSSJOIN() к двум таблицам: **цвета** и **канцелярской**.

Таблица **цвета** содержит цвета и шаблоны:

Развернуть таблицу

Цвет	Рисунок
Красный	Горизонтальная полоса
Зеленый	Вертикальная полоса
Синий	Перекрестие

Таблица **Канцелярские** содержит шрифты и презентацию:

Развернуть таблицу

Шрифт	Представление
шрифт с засечками	рельефный
sans-serif	выгравированный

Выражение для создания перекрестного соединения представлено ниже:

```
DAX
CROSSJOIN( Colors, Stationery)
```

Если указанное выше выражение используется везде, где ожидается табличное выражение, результаты выражения будут следующими:

Развернуть таблицу

Цвет	Рисунок	Шрифт	Представление
Красный	Горизонтальная полоса	шрифт с засечками	рельефный
Зеленый	Вертикальная полоса	шрифт с засечками	рельефный
Синий	Перекрестие	шрифт с засечками	рельефный
Красный	Горизонтальная полоса	sans-serif	выгравированный
Зеленый	Вертикальная полоса	sans-serif	выгравированный
Синий	Перекрестие	sans-serif	выгравированный

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CURRENTGROUP

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает набор строк из аргумента таблицы выражения GROUPBY, который принадлежит текущей строке результата GROUPBY.

Синтаксис

```DAX
CURRENTGROUP ( )
```

Параметры

Никакой

Возвращаемое значение

Строки в аргументе таблицы функции GROUPBY, соответствующей одной группе значений аргументов groupBy_columnName.

Замечания

- Эту функцию можно использовать только в выражении GROUPBY.
- Эта функция не принимает аргументы и поддерживается только в качестве первого аргумента для одной из следующих функций агрегирования: AVERAGEX, COUNTAX, COUNTX, GEOMEANX, MAXX, MINX, PRODUCTX, STDEVX.S, STDEVX.P, SUMX, VARX.S, VARX.P.

Пример

См. GROUPBY.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DATATABLE

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Предоставляет механизм объявления встроенного набора значений данных.

Синтаксис

DAX

```
DATATABLE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1,
Value2...}, {ValueN, ValueN+1...}...})
```

Параметры

Развернуть таблицу

Срок	Определение
`ColumnName`	Любое выражение DAX, возвращающее таблицу.
`DataType`	Перечисление, включающее: INTEGER, DOUBLE, STRING, BOOLEAN, CURRENCY, DATETIME
`value`	Один аргумент, использующий синтаксис Excel для константы одномерного массива, вложенный для предоставления массива массивов. Этот аргумент представляет набор значений данных, которые будут находиться в таблице Например { {значения в строке1}, {значения в строке2}, {значения в строке3}и т. д. } Где {значения в строке1} — это разделенный запятыми набор констант, а именно сочетание констант, в сочетании с несколькими основными функциями, включая DATE, TIMEи BLANK, а также оператор плюса между DATE и TIME и унарным оператором минуса, чтобы можно было выразить отрицательные значения. Ниже приведены все допустимые значения: 3, -5, BLANK(), "2009-04-15 02:45:21". Значения могут не ссылаться ни на что за пределами немедленного выражения и не могут ссылаться на столбцы, таблицы, связи или что-либо другое.

Срок	Определение
	Отсутствующее значение будет обрабатываться идентично BLANK(). Например, такие же: {1,2,BLANK(),4} {1,2,,4}

Возвращаемое значение

Таблица, объявляющая встроенный набор значений.

Замечания

- В отличие от DATATABLE, конструктор таблиц разрешает любые скалярные выражения в качестве входных значений.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

DAX

```
= DataTable("Name", STRING,
               "Region", STRING
               ,{
                        {" User1","East"},
                        {" User2","East"},
                        {" User3","West"},
                        {" User4","West"},
                        {" User4","East"}
                }
          )
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DETAILROWS

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Вычисляет выражение подробных строк, определенное для меры, и возвращает данные.

Синтаксис

```DAX
DETAILROWS([Measure])
```

Параметры

Развернуть таблицу

Срок	Определение
`Measure`	Имя меры.

Возвращаемое значение

Таблица с данными, возвращаемыми выражением "Подробные строки". Если выражение "Подробные строки" не определено, возвращаются данные для таблицы, содержащей меру.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DISTINCT (column)

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу с одним столбцом, содержащую различные значения из указанного столбца. Другими словами, повторяющиеся значения удаляются и возвращаются только уникальные значения.

> **Примечание**
>
> Эту функцию нельзя использовать для возврата значений в ячейку или столбец на листе; Вместо этого вы вложены функцию DISTINCT в формулу, чтобы получить список различных значений, которые можно передать другой функции, а затем подсчитывать, суммировать или использовать для других операций.

Синтаксис

DAX

```
DISTINCT(<column>)
```

Параметры

Развернуть таблицу

Срок	Определение
`column`	Столбец, из которого возвращаются уникальные значения. Или выражение, возвращающее столбец.

Возвращаемое значение

Столбец уникальных значений.

Замечания

- Результаты DISTINCT влияют на текущий контекст фильтра. Например, если вы используете формулу в следующем примере для создания меры, результаты будут изменяться всякий раз, когда таблица фильтруется, чтобы отобразить только определенный регион или период времени.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанные функции

Существует другая версия функции DISTINCT, DISTINCT (table), которая возвращает таблицу, удаляя повторяющиеся строки из другой таблицы или выражения.

Функция VALUES аналогична ФУНКЦИИ DISTINCT; Его также можно использовать для возврата списка уникальных значений, и обычно возвращает точно те же результаты, что и DISTINCT. Однако в некоторых контекстах VALUES возвращает одно дополнительное специальное значение. Дополнительные сведения см. вфункции .

Пример

Следующая формула подсчитывает количество уникальных клиентов, которые создали заказы по интернет-каналу. В следующей таблице показаны возможные результаты при добавлении формулы в отчет.

DAX

```
= COUNTROWS(DISTINCT(InternetSales_USD[CustomerKey]))
```

Невозможно вставить список значений, которые DISTINCT возвращается непосредственно в столбец. Вместо этого вы передаете результаты функции DISTINCT другой функции, которая подсчитывает, фильтрует или агрегирует значения с помощью списка. Чтобы сделать пример как можно проще, здесь таблица различных значений была передана в функцию COUNTROWS.

Развернуть таблицу

Метки строк	Принадлежности	Велосипеды	Одежда	Общий итог
2005		1013		1013

Метки строк	Принадлежности	Велосипеды	Одежда	Общий итог
2006		2677		2677
2007	6792	4875	2867	9309
2008	9435	5451	4196	11377
Общий итог	15114	9132	6852	18484

Кроме того, обратите внимание, что результаты не являются добавками. То есть общее число уникальных клиентов в *2007* не является суммой уникальных клиентов *аксессуаров*, *велосипедов* и *одежды* в течение этого года. Причина заключается в том, что клиент может считаться в нескольких группах.

Связанное содержимое

функция фильтрация функцийFILTERRELATED функцииVALUES

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DISTINCT (table)

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу, удаляя повторяющиеся строки из другой таблицы или выражения.

Синтаксис

DAX

```
DISTINCT(<table>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, из которой возвращаются уникальные строки. Таблица также может быть выражением, которое приводит к таблице.

Возвращаемое значение

Таблица, содержащая только отдельные строки.

Связанные функции

Существует другая версия функции DISTINCT, DISTINCT (column), которая принимает имя столбца в качестве входного параметра.

Пример

Следующий запрос:

DAX

```
EVALUATE DISTINCT( { (1, "A"), (2, "B"), (1, "A") } )
```

Возвращает таблицу:

Развернуть таблицу

[Значение1]	[Значение2]
1	A
2	B

Связанное содержимое

ФункцииDISTINCT (column)FILTER функцииRELATED функцииVALUES

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EXCEPT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает строки первой таблицы в выражении, которое не отображается во второй таблице.

Синтаксис

DAX

```
EXCEPT(<table_expression1>, <table_expression2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Table_expression`	Любое выражение DAX, возвращающее таблицу.

Возвращаемое значение

Таблица, содержащая строки одной таблицы минус все строки другой таблицы.

Замечания

- Если строка отображается во всех таблицах, она и ее дубликаты отсутствуют в результирующем наборе. Если строка отображается только в table_expression1, она и ее дубликаты будут отображаться в результирующем наборе.
- Имена столбцов будут соответствовать именам столбцов в table_expression1.
- Возвращаемая таблица имеет происхождение на основе столбцов в table_expression1 независимо от происхождения столбцов во второй таблице. Например, если первый столбец первого table_expression имеет

происхождение в базовый столбец C1 в модели, то "Кроме" уменьшит строки на основе доступности значений в первом столбце второго table_expression и сохранят происхождение в базовом столбце C1 без изменений.

- Две таблицы должны иметь одинаковое количество столбцов.
- Столбцы сравниваются на основе позиционирования и сравнения данных без приведения типов.
- Возвращаемый набор строк зависит от порядка двух выражений.
- Возвращаемая таблица не содержит столбцы из таблиц, связанных с table_expression1.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Состояния1

Развернуть таблицу

Государство
A
B
B
B
C
D
D

Штаты2

Развернуть таблицу

Государство
B

Государство
C
D
D
D
E
E
E

Except(States1, States2)

Развернуть таблицу

Государство
A

Except(States2, States1)

Развернуть таблицу

Государство
E
E
E

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FILTERS

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает значения, которые непосредственно применяются в качестве фильтров для `columnName`.

Синтаксис

DAX

```
FILTERS(<columnName>)
```

Параметры

Развернуть таблицу

Срок	Описание
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX. Это не может быть выражением.

Возвращаемое значение

Значения, которые непосредственно применяются в качестве фильтров для `columnName`.

Замечания

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере показано, как определить количество прямых фильтров столбца.

```
DAX
= COUNTROWS(FILTERS(ResellerSales_USD[ProductKey]))
```

В этом примере вы узнаете, сколько прямых фильтров на ResellerSales_USD[ProductKey] применено к контексту, в котором вычисляется выражение.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

GENERATE

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает таблицу с декартовским продуктом между каждой строкой в *таблице1* и таблицей, которая приводит к оценке *таблицы 2* в контексте текущей строки из *таблицы 1*.

Синтаксис

```DAX
GENERATE(<table1>, <table2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table1`	Любое выражение DAX, возвращающее таблицу.
`table2`	Любое выражение DAX, возвращающее таблицу.

Возвращаемое значение

Таблица с декартовским продуктом между каждой строкой в *таблице1* и таблицей, которая приводит к оценке *таблицы 2* в контексте текущей строки из *таблицы 1*

Замечания

- Если оценка *таблицы 2* для текущей строки в *таблице1* возвращает пустую таблицу, то результирующая таблица не будет содержать текущую строку из *таблицы 1*. Это отличается от GENERATEALL(), где текущая строка из *таблицы 1* будет включена в результаты и столбцы, соответствующие *таблице 2* будут иметь значения NULL для этой строки.

- Все имена столбцов из *таблицы 1* и *таблицы* 2 должны отличаться или возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере пользователь хочет сводную таблицу продаж по регионам и категории продуктов для канала торговых посредников, например следующую таблицу:

Развернуть таблицу

SalesTerritory[SalesTerritoryGroup]	ProductCategory[ProductCategoryName]	[Продажи торговых посредников]
Европа	Принадлежности	$ 142,227,27
Европа	Велосипеды	$ 9,970,200,44
Европа	Одежда	$ 365,847,63
Европа	Компоненты	$ 2,214,440,19
Северная Америка	Принадлежности	$ 379,305,15
Северная Америка	Велосипеды	$ 52,403,796,85
Северная Америка	Одежда	$ 1,281,193,26
Северная Америка	Компоненты	$ 8,882,848,05
Тихоокеанский	Принадлежности	$ 12,769,57
Тихоокеанский	Велосипеды	$ 710,677,75
Тихоокеанский	Одежда	$ 22,902,38
Тихоокеанский	Компоненты	$ 108,549,71

Следующая формула создает приведенную выше таблицу:

```
DAX
```

```
GENERATE(
SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
,SUMMARIZE(ProductCategory
, [ProductCategoryName]
, "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD),
ResellerSales_USD[SalesAmount_USD])
)
)
```

1. Первый оператор SUMMARIZE, `SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])`, создает таблицу групп территорий, где каждая строка является группой территорий, как показано ниже:

Развернуть таблицу

SalesTerritory[SalesTerritoryGroup]
Северная Америка
Европа
Тихоокеанский
NA

2. Второй оператор SUMMARIZE, `SUMMARIZE(ProductCategory, [ProductCategoryName], "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))`, создает таблицу групп категорий продуктов с продажами торгового посредника для каждой группы, как показано ниже:

Развернуть таблицу

ProductCategory[ProductCategoryName]	[Продажи торговых посредников]
Велосипеды	$ 63,084,675,04
Компоненты	$ 11,205,837,96
Одежда	$ 1,669,943,27
Принадлежности	$ 534,301,99

3. Однако при получении приведенной выше таблицы и ее оценке в контексте каждой строки из таблицы групп территорий вы получаете разные результаты для каждой территории.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

GENERATEALL

Статья • 17.12.2024

применимо: ✅ вычисляемый столбец ✅ вычисляемой таблице ✅ Measure ✅ визуального вычисления

Возвращает таблицу с декартовой product между каждой строкой в *таблице1*and таблице, которая приводит к оценке *таблицы 2* в контексте текущей строки из *таблицы 1*.

Синтаксис

```
DAX

GENERATEALL(<table1>, <table2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`table1`	Любое выражение DAX, возвращающее таблицу.
`table2`	Любое выражение DAX, возвращающее таблицу.

Возврат value

Таблица с декартовой product между каждой строкой в *таблице1*and таблицей, которая приводит к оценке *таблицы 2* в контексте текущей строки из *таблицы 1*

Замечания

- If вычисление *таблицы 2* для текущей строки в *таблице1* возвращает пустую таблицу, а текущая строка из *таблицы 1* будет включена в результаты and столбцы, соответствующие столбцам *таблицы 2* будет иметь значение NULL values для этой строки. Это отличается от GENERATE(), где текущая строка из *таблицы 1***not** будет включена в результаты.

- All имена столбцов из таблицы *table1*and*table2* должны отличаться or возвращается error.
- Эта функция not поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах or правилах безопасности на уровне строк (RLS).

Пример

В следующем примере пользователь хочет сводную таблицу продаж по регионам andProduct категории для канала "Торговые посредники", например следующую таблицу:

Развернуть таблицу

SalesTerritory[SalesTerritoryGroup]	ProductCategory[ProductCategoryName]	[Продажи торговых посредников]
Европа	Принадлежности	$ 142,227,27
Европа	Велосипеды	$ 9,970,200,44
Европа	Одежда	$ 365,847,63
Европа	Компоненты	$ 2,214,440,19
NA	Принадлежности	
NA	Велосипеды	
NA	Одежда	
NA	Компоненты	
Северная Америка	Принадлежности	$ 379,305,15
Северная Америка	Велосипеды	$ 52,403,796,85
Северная Америка	Одежда	$ 1,281,193,26
Северная Америка	Компоненты	$ 8,882,848,05
Тихоокеанский	Принадлежности	$ 12,769,57
Тихоокеанский	Велосипеды	$ 710,677,75
Тихоокеанский	Одежда	$ 22,902,38

SalesTerritory[SalesTerritoryGroup]	ProductCategory[ProductCategoryName]	[Продажи торговых посредников]
Тихоокеанский	Компоненты	$ 108,549,71

Следующая формула создает приведенную выше таблицу:

DAX

```
GENERATEALL(
SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
,SUMMARIZE(ProductCategory
, [ProductCategoryName]
, "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD),
ResellerSales_USD[SalesAmount_USD])
)
)
```

1. first SUMMARIZE создает таблицу групп территорий, где каждая строка — это группа территорий, например перечисленные ниже:

   Развернуть таблицу

SalesTerritory[SalesTerritoryGroup]
Северная Америка
Европа
Тихоокеанский
NA

2. second SUMMARIZE создает таблицу групп категорий Product с продажами торгового посредника для каждой группы, как показано ниже:

   Развернуть таблицу

ProductCategory[ProductCategoryName]	[Продажи торговых посредников]
Велосипеды	$ 63,084,675,04
Компоненты	$ 11,205,837,96
Одежда	$ 1,669,943,27
Принадлежности	$ 534,301,99

3. Однако при получении приведенной выше таблицы andevaluate таблицу в контексте каждой строки из таблицы групп территорий, вы получите разные результаты для каждой территории.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

GENERATESERIES

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу одного столбца, содержащую значения арифметического ряда, то есть последовательность значений, в которой каждая из них отличается от предыдущего по количеству констант. Имя возвращаемого столбца — Значение.

Синтаксис

```
DAX

GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])
```

Параметры

Развернуть таблицу

Срок	Определение
`startValue`	Начальное значение, используемое для создания последовательности.
`endValue`	Конечное значение, используемое для создания последовательности.
`incrementValue`	(Необязательно) Добавочное значение последовательности. Если этот параметр не указан, значение по умолчанию равно 1.

Возвращаемое значение

Одна таблица столбцов, содержащая значения арифметического ряда. Имя столбца — Значение.

Замечания

- Если endValue меньше startValue, возвращается пустая таблица.
- добавочное значение должно быть положительным значением.

- Последовательность останавливается в последнем значении, которое меньше или равно endValue.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример 1

Следующий запрос DAX:

```
DAX

EVALUATE GENERATESERIES(1, 5)
```

Возвращает следующую таблицу с одним столбцом:

Развернуть таблицу

[значение]
1
2
3
4
5

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE GENERATESERIES(1.2, 2.4, 0.4)
```

Возвращает следующую таблицу с одним столбцом:

Развернуть таблицу

[значение]
1.2
1.6
2
2.4

Пример 3

Следующий запрос DAX:

```
DAX

EVALUATE GENERATESERIES(CURRENCY(10), CURRENCY(12.4), CURRENCY(0.5))
```

Возвращает следующую таблицу с одним столбцом:

Развернуть таблицу

[значение]
10
10.5
11
11.5
12

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

GROUPBY

Статья • 21.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ⚠️ визуального вычисления

> ⓘ **Примечание**
>
> Эта функция не рекомендуется использовать в **визуальных вычислениях**, так как, скорее всего, возвращает бессмысленные результаты.

Функция GROUPBY аналогична функции SUMMARIZE. Однако GROUPBY не выполняет неявную CALCULATE для столбцов расширений, которые он добавляет. GROUPBY позволяет использовать новую функцию CURRENTGROUPв функциях агрегирования в столбцах расширения, которые он добавляет. GROUPBY используется для выполнения нескольких агрегатов в одной таблице.

Синтаксис

DAX

```
GROUPBY (<table> [, <groupBy_columnName> [, <groupBy_columnName> [, …]]] [, <name>, <expression> [, <name>, <expression> [, …]]])
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Любое выражение DAX, возвращающее таблицу данных.
`groupBy_columnName`	Имя существующего столбца в таблице (или в связанной таблице), по которой данные должны быть сгруппированы. Этот параметр не может быть выражением.
`name`	Имя, заданное новому столбцу, добавляемого в список столбцов GroupBy, заключенное в двойные кавычки.
`expression`	Одна из функций агрегирования X с первым аргументом, CURRENTGROUP(). Полный список поддерживаемых функций

Срок	Определение
	агрегирования X см. в разделе с CURRENTGROUP ниже.

Возвращаемое значение

Таблица с выбранными столбцами для аргументов groupBy_columnName и столбцов расширения, назначенных аргументами имени.

Замечания

- Функция GROUPBY выполняет следующие действия:

    1. Начните с указанной таблицы (и всех связанных таблиц в направлении "к одному").

    2. Создайте группирование с помощью всех столбцов GroupBy (которые должны существовать в таблице на шаге 1.).

    3. Каждая группа является одной строкой в результате, но представляет набор строк в исходной таблице.

    4. Для каждой группы оцените добавляемые столбцы расширения. В отличие от функции SUMMARIZE, подразумеваемая CALCULATE не выполняется, и группа не помещается в контекст фильтра.

- Каждый столбец, для которого определяется имя, должно иметь соответствующее выражение; в противном случае возвращается ошибка. Первый аргумент, имя, определяет имя столбца в результатах. Второй аргумент, выражение, определяет вычисление, выполняемое для получения значения для каждой строки в этом столбце.

- `groupBy_columnName` должны находиться в таблице или в связанной таблице.

- Каждое имя должно быть заключено в двойные кавычки.

- Функция группирует выбранный набор строк в набор суммарных строк по значениям одного или нескольких столбцов groupBy_columnName. Для каждой группы возвращается одна строка.

- GROUPBY в основном используется для выполнения агрегирования по промежуточным результатам из DAX табличных выражений. Для эффективной

агрегирования по физическим таблицам в модели рекомендуется использовать функцию SUMMARIZECOLUMNS или SUMMARIZE.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

C CURRENTGROUP

CURRENTGROUP можно использовать только в выражении, определяющем столбец расширения в функции GROUPBY. В результате CURRENTGROUP возвращает набор строк из аргумента таблицы GROUPBY, принадлежащих текущей строке результата GROUPBY. Функция CURRENTGROUP не принимает аргументы и поддерживается только в качестве первого аргумента одной из следующих функций агрегирования: AVERAGEX, COUNTAX, COUNTX, GEOMEANX, MAXX, MINX, PRODUCTX, STDEVX.S, STDEVX.P, SUMX, VARX.S, VARX.P.

Пример

В следующем примере сначала вычисляется общий объем продаж по странам и категориям продуктов по физическим таблицам с помощью функции SUMMARIZECOLUMNS. Затем она использует функцию GROUPBY для сканирования промежуточного результата от первого шага, чтобы найти максимальное количество продаж в каждой стране по категориям продуктов.

DAX

```
DEFINE
VAR SalesByCountryAndCategory =
SUMMARIZECOLUMNS(
Geography[Country],
Product[Category],
"Total Sales", SUMX(Sales, Sales[Price] * Sales[Qty])
)

EVALUATE
GROUPBY(
SalesByCountryAndCategory,
Geography[Country],
"Max Sales", MAXX(CURRENTGROUP(), [Total Sales])
)
```

Связанное содержимое

функция SUMMARIZESUMMARIZECOLUMNS

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

IGNORE

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Изменяет поведение функции SUMMARIZECOLUMNS путем пропуска определенных выражений из вычисления BLANK/NULL. Строки, для которых все выражения, не использующие IGNORE возвращать BLANK/NULL, будут исключены независимо от того, будут ли выражения, использующие IGNORE оценивать BLANK/NULL или нет. Эту функцию можно использовать только в выражении SUMMARIZECOLUMNS.

Синтаксис

DAX

```
IGNORE(<expression>)
```

С SUMMARIZECOLUMNS,

DAX

```
SUMMARIZECOLUMNS(<groupBy_columnName>[, < groupBy_columnName >]…,
[<filterTable>]…[, <name>, IGNORE(…)]…)
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Любое выражение DAX, возвращающее одно значение (а не таблицу).

Возвращаемое значение

Функция не возвращает значение.

Замечания

IGNORE можно использовать только в качестве аргумента выражения для SUMMARIZECOLUMNS.

Пример

См. SUMMARIZECOLUMNS.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

INTERSECT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает пересечение строк двух таблиц, сохраняя дубликаты.

Синтаксис

DAX

```
INTERSECT(<table_expression1>, <table_expression2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Table_expression`	Любое выражение DAX, возвращающее таблицу.

Возвращаемое значение

Таблица, содержащая все строки в table_expression1, которые также находятся в table_expression2

Исключения

Замечания

- Интерсект не является коммутативным. Как правило, Intersect(T1, T2) будет иметь другой результирующий набор, чем Intersect(T2, T1).
- Повторяющиеся строки сохраняются. Если строка отображается в table_expression1 и table_expression2, она и все дубликаты в table_expression_1 включаются в результирующий набор.
- Имена столбцов будут соответствовать именам столбцов в table_expression1.

- Возвращаемая таблица имеет происхождение на основе столбцов в table_expression1 независимо от происхождения столбцов во второй таблице. Например, если первый столбец первого table_expression имеет происхождение в базовый столбец C1 в модели, интерсект уменьшит строки на основе пересечения первого столбца второго table_expression и сохраните происхождение в базовом столбце C1 без изменений.
- Столбцы сравниваются на основе позиционирования и сравнения данных без приведения типов.
- Возвращаемая таблица не содержит столбцы из таблиц, связанных с table_expression1.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Состояния1

Развернуть таблицу

Государство
A
A
B
B
B
C
D
D

Штаты2

Развернуть таблицу

Государство
B
C
D
D
D
E

Intersect(States1, States2)

Развернуть таблицу

Государство
B
B
B
C
D
D

Intersect(States2, States1)

Развернуть таблицу

Государство
B
C
D
D
D

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NATURALINNERJOIN

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Выполняет внутреннее соединение таблицы с другой таблицей.

Синтаксис

DAX

```
NATURALINNERJOIN(<LeftTable>, <RightTable>)
```

Параметры

Развернуть таблицу

Срок	Определение
`LeftTable`	Табличное выражение, определяющее таблицу слева от соединения.
`RightTable`	Выражение таблицы, определяющее таблицу справа от соединения.

Возвращаемое значение

Таблица, содержащая только строки, для которых значения в общих столбцах указаны в обеих таблицах. Возвращаемая таблица будет содержать общие столбцы из левой таблицы и других столбцов из обеих таблиц.

Замечания

- Таблицы объединяются по общим столбцам (по имени) в двух таблицах. Если в двух таблицах нет общих имен столбцов, возвращается ошибка.
- Для результатов не гарантируется порядок сортировки.
- Столбцы, присоединенные к ним, должны иметь одинаковый тип данных в обеих таблицах.

- Присоединяются только столбцы из одной исходной таблицы (с одинаковым происхождением). Например, Products[ProductID], WebSales[ProductdID], StoreSales[ProductdID] с отношениями "многие к одному" между WebSales и StoreSales и таблицей Products на основе столбца ProductID, WebSales и StoreSales таблицы объединяются в [ProductID].
- Строгая семантика сравнения используется во время соединения. Нет приведения типов; например, 1 не равен 1.0.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

NATURALLEFTOUTERJOIN

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NATURALLEFTOUTERJOIN

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Выполняет присоединение левойtable к rightTable с помощью семантики внешнего соединения слева.

Синтаксис

```
DAX

NATURALLEFTOUTERJOIN(<LeftTable>, <RightTable>)
```

Параметры

Развернуть таблицу

Срок	Определение
`LeftTable`	Табличное выражение, определяющее таблицу слева от соединения.
`RightTable`	Выражение таблицы, определяющее таблицу справа от соединения.

Возвращаемое значение

Таблица, содержащая только строки из RightTable, для которой значения в общих столбцах также присутствуют в LeftTable. Возвращаемая таблица содержит общие столбцы из левой таблицы и другие столбцы из обеих таблиц.

Замечания

- Таблицы объединяются по общим столбцам (по имени) в двух таблицах. Если в двух таблицах нет общих имен столбцов, возвращается ошибка.
- Для результатов не гарантируется порядок сортировки.
- Столбцы, присоединенные к ним, должны иметь одинаковый тип данных в обеих таблицах.

- Присоединяются только столбцы из одной исходной таблицы (с одинаковым происхождением). Например, Products[ProductID], WebSales[ProductdID], StoreSales[ProductdID] с отношениями "многие к одному" между WebSales и StoreSales и таблицей Products на основе столбца ProductID, WebSales и StoreSales таблицы объединяются в [ProductID].
- Строгая семантика сравнения используется во время соединения. Нет приведения типов; например, 1 не равен 1.0.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Связанное содержимое

NATURALINNERJOIN

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ROLLUP

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Изменяет поведение функции SUMMARIZE путем добавления строк свертки в результат для столбцов, определенных параметром groupBy_columnName. Эту функцию можно использовать только в выражении SUMMARIZE.

Синтаксис

DAX

```
ROLLUP ( <groupBy_columnName> [, <groupBy_columnName> [, … ] ] )
```

С SUMMARIZE,

DAX

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]…[,
ROLLUP(<groupBy_columnName>[,< groupBy_columnName>…])][, <name>,
<expression>]…)
```

Параметры

Развернуть таблицу

Термин	Определение
groupBy_columnName	Полное имя существующего столбца или функции ROLLUPGROUP, используемой для создания сводных групп на основе значений, найденных в нем. Этот параметр не может быть выражением.

Возвращаемое значение

Эта функция не возвращает значение. Он указывает только набор столбцов для промежуточных итогов.

Замечания

Эту функцию можно использовать только в выражении SUMMARIZE.

Пример

См. SUMMARIZE.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ROLLUPADDISSUBTOTAL

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Изменяет поведение функции SUMMARIZECOLUMNS путем добавления свертки или промежуточных строк в результат на основе столбцов groupBy_columnName. Эту функцию можно использовать только в выражении SUMMARIZECOLUMNS.

Синтаксис

DAX

```
ROLLUPADDISSUBTOTAL ( [<grandtotalFilter>], <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, … ] ] ] ] )
```

Параметры

Развернуть таблицу

Термин	Определение
`grandtotalFilter`	(Необязательно) Фильтр, применяемый к большому уровню.
`groupBy_columnName`	Имя существующего столбца, используемого для создания сводных групп на основе значений, найденных в нем. Не может быть выражением.
name	Имя столбца ISSUBTOTAL. Значения столбца вычисляются с помощью функции ISSUBTOTAL.
`groupLevelFilter`	(Необязательно) Фильтр для применения к текущему уровню.

Возвращаемое значение

Функция не возвращает значение.

Замечания

Никакой

Пример

См. SUMMARIZECOLUMNS.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ROLLUPGROUP

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Изменяет поведение функций SUMMARIZE и SUMMARIZECOLUMNS путем добавления строк свертки в результат для столбцов, определенных параметром groupBy_columnName. Эту функцию можно использовать только в выражении SUMMARIZE или SUMMARIZECOLUMNS.

Синтаксис

DAX

```
ROLLUPGROUP ( <groupBy_columnName> [, <groupBy_columnName> [, … ] ] )
```

Параметры

Развернуть таблицу

Термин	Определение
groupBy_columnName	Полное имя существующего столбца или функции ROLLUPGROUP, используемой для создания сводных групп на основе значений, найденных в нем. Этот параметр не может быть выражением.

Возвращаемое значение

Эта функция не возвращает значение. Он помечает набор столбцов, которые следует рассматривать как одну группу во время промежуточных итогов, ROLLUP или ROLLUPADDISSUBTOTAL.

Замечания

ROLLUPGROUP можно использовать только в качестве аргумента groupBy_columnName для ROLLUP, ROLLUPADDISSUBTOTALили ROLLUPISSUBTOTAL.

Пример

См. SUMMARIZE и SUMMARIZECOLUMNS.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ROLLUPISSUBTOTAL

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Объединяет группы свертки с столбцом, добавленным ROLLUPADDISSUBTOTAL. Эту функцию можно использовать только в выражении ADDMISSINGITEMS.

Синтаксис

DAX

```
ROLLUPISSUBTOTAL ( [<grandTotalFilter>], <groupBy_columnName>,
<isSubtotal_columnName> [, [<groupLevelFilter>] [, <groupBy_columnName>,
<isSubtotal_columnName> [, [<groupLevelFilter>] [, … ] ] ] ] )
```

Параметры

Развернуть таблицу

Термин	Определение
`grandTotalFilter`	(Необязательно) Фильтр, применяемый к большому уровню.
`groupBy_columnName`	Имя существующего столбца, используемого для создания сводных групп на основе значений, найденных в нем. Не может быть выражением.
isSubtotal_columnName	Имя столбца ISSUBTOTAL. Значения столбца вычисляются с помощью функции ISSUBTOTAL.
`groupLevelFilter`	(Необязательно) Фильтр для применения к текущему уровню.

Возвращаемое значение

Никакой

Замечания

Эту функцию можно использовать только в выражении ADDMISSINGITEMS.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Функция ROW

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу с одной строкой, содержащей значения, полученные из выражений, заданных каждому столбцу.

Синтаксис

DAX

```
ROW(<name>, <expression>[[,<name>, <expression>]…])
```

Параметры

Развернуть таблицу

Термин	Определение
`name`	Имя столбца, заключенное в двойные кавычки.
`expression`	Любое выражение DAX, возвращающее одно скалярное значение для заполнения. `name`.

Возвращаемое значение

Одна таблица строк

Замечания

- Аргументы всегда должны находиться в парах `name` и `expression`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере возвращается одна таблица строк с общим объемом продаж для каналов Интернета и торговых посредников.

DAX

```
ROW("Internet Total Sales (USD)", SUM(InternetSales_USD[SalesAmount_USD]),
         "Resellers Total Sales (USD)",
SUM(ResellerSales_USD[SalesAmount_USD]))
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SELECTCOLUMNS

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает таблицу с выбранными столбцами из таблицы и новыми столбцами, указанными DAX выражениями.

Синтаксис

DAX

```
SELECTCOLUMNS(<Table>, [<Name>], <Expression>, [<Name>], …)
```

Параметры

Развернуть таблицу

Термин	Определение
`Table`	Любое выражение DAX, возвращающее таблицу.
`Name`	Имя столбца, заключенное в двойные кавычки.
`Expression`	Любое выражение, возвращающее скалярное значение, например ссылку на столбец, целое число или строковое значение.

Возвращаемое значение

Таблица с таким же числом строк, что и таблица, указанная в качестве первого аргумента. Возвращаемая таблица содержит один столбец для каждой пары `Name`, `Expression` аргументов, и каждое выражение вычисляется в контексте строки из указанного `Table` аргумента.

Замечания

SELECTCOLUMNS имеет ту же сигнатуру, что и ADDCOLUMNS, и имеет то же поведение, что вместо указанного `Table`, SELECTCOLUMNS начинается с пустой таблицы перед добавлением столбцов.

Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Для следующей таблицы с именем **Customer**:

Развернуть таблицу

Страна/регион	Штат	Считать	Итог
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
USA	Целостности и доступности	5	500
USA	WA	10	900

DAX

```
SELECTCOLUMNS(Customer, "Country, State", [Country]&", "&[State])
```

Возвращает

Развернуть таблицу

Страна, штат
IND, JK
IND, MH
IND, WB
США, ЦС
США, WA

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SUBSTITUTEWITHINDEX

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает таблицу, представляющую левый полусоединен из двух таблиц, предоставленных в качестве аргументов. Semijoin выполняется с помощью общих столбцов, определяемых общими именами столбцов и общим типом данных. Столбцы, присоединенные к ним, заменяются одним столбцом в возвращаемой таблице, которая имеет целое число типа и содержит индекс. Индекс — это ссылка на нужную таблицу соединения с заданным порядком сортировки.

Столбцы в правой или второй таблице, указанной в левой или первой таблице, не включены в возвращаемую таблицу и не используются для соединения.

Индекс начинается с 0 (на основе 0) и увеличивается по одному для каждой дополнительной строки в предоставленной таблице справа/секунды соединения. Индекс основан на порядке сортировки, указанном для таблицы справа или второго соединения.

Синтаксис

DAX

```
SUBSTITUTEWITHINDEX(<table>, <indexColumnName>, <indexColumnsTable>,
[<orderBy_expression>, [<order>][, <orderBy_expression>, [<order>]]…])
```

Параметры

Развернуть таблицу

Термин	Определение
`table`	Таблица для фильтрации путем выполнения левого полусоединения с таблицей, указанной в качестве третьего аргумента (indexColumnsTable). Это таблица в левой части левого полусоединения, поэтому возвращаемая таблица содержит те же столбцы, что и эта таблица, за исключением того, что все общие столбцы двух таблиц будут заменены одним столбцом индекса в возвращаемой таблице.

Термин	Определение
`indexColumnName`	Строка, указывающая имя столбца индекса, который заменяет все общие столбцы в двух таблицах, предоставленных в качестве аргументов этой функции.
`indexColumnsTable`	Вторая таблица для левого полусоединяемого. Это таблица справа от левого полусоединения. Только значения, присутствующих в этой таблице, будут возвращены функцией. Кроме того, столбцы этой таблицы (на основе имен столбцов) будут заменены одним столбцом индекса в таблице, возвращаемой этой функцией.
`orderBy_expression`	Любое DAX выражение, в котором значение результата используется для указания требуемого порядка сортировки таблицы indexColumnsTable для создания правильных значений индекса. Порядок сортировки, указанный для таблицы indexColumnsTable, определяет индекс каждой строки в таблице, а этот индекс используется в таблице, возвращаемой для представления сочетаний значений в indexColumnsTable, так как они отображаются в таблице, предоставленной в качестве первого аргумента этой функции.
`order`	(Необязательно) Значение, указывающее, как сортировать orderBy_expression значения, возрастание или убывание: Значение: `Desc`. Альтернативное значение: `0`(ноль)/`FALSE`. Сортировка по убыванию значений orderBy_expression. Это значение по умолчанию при опущении параметра заказа. Значение: `ASC`. Альтернативное значение: `1`/`TRUE`. Ранжирует в порядке возрастания orderBy_expression.

Возвращаемое значение

Таблица, содержащая только те значения, которые присутствуют в таблице indexColumnsTable, и которая содержит столбец индекса вместо всех столбцов, присутствующих (по имени) в таблице indexColumnsTable.

Замечания

- Эта функция не гарантирует порядок сортировки результатов.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SUMMARIZE

Статья • 14.12.2023

Возвращает сводную таблицу для запрошенных итогов по набору групп.

Синтаксис

```
DAX

SUMMARIZE (<table>, <groupBy_columnName>[, <groupBy_columnName>]…[, <name>, <expression>]…)
```

Параметры

Развернуть таблицу

Термин	Определение
table	Любое выражение DAX, возвращающее таблицу данных.
groupBy_ColumnName	(Необязательно) Полное имя существующего столбца, используемого для создания сводных групп на основе значений, найденных в нем. Этот параметр не может быть выражением.
name	Имя, заданное общему столбцу или сводке, заключенному в двойные кавычки.
выражение	Любое выражение DAX, возвращающее одно скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста).

Возвращаемое значение

Таблица с выбранными столбцами для *аргументов groupBy_columnName* и суммированных столбцов, разработанных аргументами имени.

Замечания

- Каждый столбец, для которого определяется имя, должно иметь соответствующее выражение; в противном случае возвращается ошибка. Первый аргумент, имя, определяет имя столбца в результатах. Второй аргумент, выражение, определяет вычисление, выполняемое для получения значения для каждой строки в этом столбце.
- groupBy_columnName должны находиться в *таблице или в связанной таблице*.
- Каждое имя должно быть заключено в двойные кавычки.
- Функция группирует выбранный набор строк в набор суммарных строк по значениям одного или нескольких столбцов groupBy_columnName. Для каждой группы возвращается одна строка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере возвращается сводка по продажам торгового посредника, сгруппированного по календарю и имени категории продукта, эта таблица результатов позволяет выполнять анализ продаж торгового посредника по годам и категории продуктов.

```
DAX

SUMMARIZE(ResellerSales_USD
      , DateTime[CalendarYear]
      , ProductCategory[ProductCategoryName]
      , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
```

```
      , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
      )
```

В следующей таблице показана предварительная версия данных по мере получения любой функцией, ожидающей получения таблицы:

Развернуть таблицу

DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Сумма продаж (USD)]	[Сумма скидки (USD)]
2008	Велосипеды	12968255.42	36167.6592
2005	Велосипеды	6958251.043	4231.1621
2006	Велосипеды	18901351.08	178175.8399
2007	Велосипеды	24256817.5	276065.992
2008	Компоненты	2008052.706	39.9266
2005	Компоненты	574256.9865	0
2006	Компоненты	3428213.05	948.7674
2007	Компоненты	5195315.216	4226.0444
2008	Clothing	366507.844	4151.1235
2005	Clothing	31851.1628	90.9593
2006	Clothing	455730.9729	4233.039
2007	Clothing	815853.2868	12489.3835
2008	Аксессуары	153299.924	865.5945
2005	Аксессуары	18594.4782	4.293
2006	Аксессуары	86612.7463	1061.4872
2007	Аксессуары	275794.8403	4756.6546

С накопительным пакетом

Добавление синтаксиса ROLLUP изменяет поведение функции СУММКЕ путем добавления строк свертки в результат в столбцы groupBy_columnName. ROLLUP можно использовать только в выражении SUMMARIZE.

Пример

В следующем примере добавляются строки свертки в столбцы вызова функции СУММТЕ:

DAX

```
SUMMARIZE(ResellerSales_USD
      , ROLLUP( DateTime[CalendarYear], ProductCategory[ProductCategoryName])
      , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
      , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

Возвращает следующую таблицу:

Развернуть таблицу

DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Сумма продаж (USD)]	[Сумма скидки (USD)]
2008	Велосипеды	12968255.42	36167.6592
2005	Велосипеды	6958251.043	4231.1621

DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Сумма продаж (USD)]	[Сумма скидки (USD)]
2006	Велосипеды	18901351.08	178175.8399
2007	Велосипеды	24256817.5	276065.992
2008	Компоненты	2008052.706	39.9266
2005	Компоненты	574256.9865	0
2006	Компоненты	3428213.05	948.7674
2007	Компоненты	5195315.216	4226.0444
2008	Clothing	366507.844	4151.1235
2005	Clothing	31851.1628	90.9593
2006	Clothing	455730.9729	4233.039
2007	Clothing	815853.2868	12489.3835
2008	Аксессуары	153299.924	865.5945
2005	Аксессуары	18594.4782	4.293
2006	Аксессуары	86612.7463	1061.4872
2007	Аксессуары	275794.8403	4756.6546
2008		15496115.89	41224.3038
2005		7582953.67	4326.4144
2006		22871907.85	184419.1335
2007		30543780.84	297538.0745
		76494758.25	527507.9262

С помощью ROLLUPGROUP

Добавление ROLLUPGROUP в синтаксис ROLLUP можно использовать для предотвращения частичных промежуточных итогов в строках свертки. ROLLUPGROUP можно использовать только в выражении ROLLUP, ROLLUPADDISSUBTOTAL или ROLLUPISSUBTOTAL.

Пример

В следующем примере показан только общий итог всех лет и категорий без промежуточных итогов каждого года со всеми категориями:

DAX

```
SUMMARIZE(ResellerSales_USD
      , ROLLUP(ROLLUPGROUP( DateTime[CalendarYear], ProductCategory[ProductCategoryName]))
      , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
      , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

Возвращает следующую таблицу:

Развернуть таблицу

DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Сумма продаж (USD)]	[Сумма скидки (USD)]
2008	Велосипеды	12968255.42	36167.6592
2005	Велосипеды	6958251.043	4231.1621

DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Сумма продаж (USD)]	[Сумма скидки (USD)]
2006	Велосипеды	18901351.08	178175.8399
2007	Велосипеды	24256817.5	276065.992
2008	Компоненты	2008052.706	39.9266
2005	Компоненты	574256.9865	0
2006	Компоненты	3428213.05	948.7674
2007	Компоненты	5195315.216	4226.0444
2008	Clothing	366507.844	4151.1235
2005	Clothing	31851.1628	90.9593
2006	Clothing	455730.9729	4233.039
2007	Clothing	815853.2868	12489.3835
2008	Аксессуары	153299.924	865.5945
2005	Аксессуары	18594.4782	4.293
2006	Аксессуары	86612.7463	1061.4872
2007	Аксессуары	275794.8403	4756.6546
		76494758.25	527507.9262

С ISSUBTOTAL

С помощью ISSUBTOTAL можно создать другой столбец в выражении SUMMARIZE, возвращающий значение True, если строка содержит промежуточные значения столбца, заданного в качестве аргумента ISSUBTOTAL, в противном случае возвращает значение False. ISSUBTOTAL можно использовать только в выражении SUMMARIZE.

Пример

В следующем примере создается столбец ISSUBTOTAL для каждого столбца ROLLUP в заданном вызове функции SUMMARIZE:

DAX

```
SUMMARIZE(ResellerSales_USD
      , ROLLUP( DateTime[CalendarYear], ProductCategory[ProductCategoryName])
      , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
      , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
      , "Is Sub Total for DateTimeCalendarYear", ISSUBTOTAL(DateTime[CalendarYear])
      , "Is Sub Total for ProductCategoryName", ISSUBTOTAL(ProductCategory[ProductCategoryName])
)
```

Возвращает следующую таблицу:

Развернуть таблицу

[Общее значение для DateTimeCalendarYear]	[Вложено общее значение для ProductCategoryName]	DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Сумма продаж (USD)]	[Сумма скидки (USD)]
FALSE	FALSE				
FALSE	FALSE	2008	Велосипеды	12968255.42	36167.6592
FALSE	FALSE	2005	Велосипеды	6958251.043	4231.1621
FALSE	FALSE	2006	Велосипеды	18901351.08	178175.8399

[Общее значение для DateTimeCalendarYear]	[Вложено общее значение для ProductCategoryName]	DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Сумма продаж (USD)]	[Сумма скидки (USD)]
FALSE	FALSE	2007	Велосипеды	24256817.5	276065.992
FALSE	FALSE	2008	Компоненты	2008052.706	39.9266
FALSE	FALSE	2005	Компоненты	574256.9865	0
FALSE	FALSE	2006	Компоненты	3428213.05	948.7674
FALSE	FALSE	2007	Компоненты	5195315.216	4226.0444
FALSE	FALSE	2008	Clothing	366507.844	4151.1235
FALSE	FALSE	2005	Clothing	31851.1628	90.9593
FALSE	FALSE	2006	Clothing	455730.9729	4233.039
FALSE	FALSE	2007	Clothing	815853.2868	12489.3835
FALSE	FALSE	2008	Аксессуары	153299.924	865.5945
FALSE	FALSE	2005	Аксессуары	18594.4782	4.293
FALSE	FALSE	2006	Аксессуары	86612.7463	1061.4872
FALSE	FALSE	2007	Аксессуары	275794.8403	4756.6546
FALSE	TRUE				
FALSE	TRUE	2008		15496115.89	41224.3038
FALSE	TRUE	2005		7582953.67	4326.4144
FALSE	TRUE	2006		22871907.85	184419.1335
FALSE	TRUE	2007		30543780.84	297538.0745
TRUE	TRUE			76494758.25	527507.9262

Связанный контент

SUMMARIZECOLUMNS

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SUMMARIZECOLUMNS

Статья • 21.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ❌ визуального вычисления

Возвращает сводную таблицу по набору групп.

Синтаксис

DAX

```
SUMMARIZECOLUMNS( <groupBy_columnName> [, < groupBy_columnName >]…,
[<filterTable>]…[, <name>, <expression>]…)
```

Параметры

Развернуть таблицу

Термин	Определение
`groupBy_columnName`	Полный справочник по столбцам (Table[Column]) базовой таблице, для которой в возвращаемую таблицу включены различные значения. Каждый столбец groupBy_columnName объединяется (разные таблицы) или автоматически существует (одна и та же таблица) с последующими указанными столбцами.
`filterTable`	Табличное выражение, которое добавляется в контекст фильтра всех столбцов, указанных как groupBy_columnName аргументы. Значения, представленные в таблице фильтров, используются для фильтрации перед выполнением перекрестного соединения или автоматического существования.
`name`	Строка, представляющая имя столбца, используемое для последующего выражения.
`expression`	Любое выражение DAX, возвращающее одно значение (а не таблицу).

Возвращаемое значение

Таблица, содержащая сочетания значений из предоставленных столбцов на основе указанного группировки. В возвращаемую таблицу включены только строки, для

которых по крайней мере одно из предоставленных выражений возвращает непустое значение. Если все выражения оцениваются как BLANK/NULL для строки, эта строка не включена в возвращаемую таблицу.

Замечания

- Эта функция не гарантирует порядок сортировки результатов.
- Столбец нельзя указать несколько раз в параметре groupBy_columnName. Например, следующая формула недопустима.

  `SUMMARIZECOLUMNS( Sales[StoreId], Sales[StoreId] )`

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Контекст фильтра

Обратите внимание на следующий запрос:

DAX

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category],
    FILTER('Customer', 'Customer' [First Name] = "Alicia")
)
```

В этом запросе без меры столбцы groupBy не содержат столбцов из выражения FILTER (например, из таблицы Customer). Фильтр не применяется к столбцам groupBy. Таблицы "Территория продаж" и "Клиент" могут быть косвенно связаны с помощью таблицы фактов продаж торгового посредника. Так как они не связаны напрямую, выражение фильтра является no-op, а столбцы groupBy не влияют.

Однако с этим запросом:

DAX

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category], 'Customer' [Education],
    FILTER('Customer', 'Customer'[First Name] = "Alicia")
)
```

Столбцы groupBy содержат столбец, который влияет на фильтр, и этот фильтр применяется к результатам groupBy.

С IGNORE

Синтаксис IGNORE можно использовать для изменения поведения функции SUMMARIZECOLUMNS путем пропуска определенных выражений из вычисления BLANK/NULL. Строки, для которых все выражения, не использующие IGNORE возвращать BLANK/NULL, будут исключены независимо от того, будут ли выражения, использующие IGNORE оценивать BLANK/NULL или нет. IGNORE можно использовать только в выражении SUMMARIZECOLUMNS.

Пример

DAX

```
SUMMARIZECOLUMNS(
    Sales[CustomerId], "Total Qty",
    IGNORE( SUM( Sales[Qty] ) ),
    "BlankIfTotalQtyIsNot3", IF( SUM( Sales[Qty] )=3, 3 )
)
```

Это свертит столбец Sales[CustomerId], создав промежуточный итог для всех клиентов в заданной группе. Без IGNOREрезультат:

Развернуть таблицу

CustomerId	Всего Qty	BlankIfTotalQtyIsNot3
a	5	
Б	3	3
С	3	3

С IGNORE,

Развернуть таблицу

CustomerId	Всего Qty	BlankIfTotalQtyIsNot3
Б	3	3
С	3	3

Все выражения игнорируются,

```DAX
SUMMARIZECOLUMNS(
    Sales[CustomerId], "Blank",
    IGNORE( BLANK() ), "BlankIfTotalQtyIsNot5",
    IGNORE( IF( SUM( Sales[Qty] )=5, 5 ) )
)
```

Несмотря на то, что оба выражения возвращают пустое значение для некоторых строк, они включаются, так как нет неигнорированных выражений, возвращающих пустое.

Развернуть таблицу

CustomerId	Пустой	BlankIfTotalQtyIsNot5
a		5
Б		
C		

C NONVISUAL

Функция NONVISUAL помечает фильтр значений в SUMMARIZECOLUMNS функции, так как не влияет на значения мер, но применяется только к столбцам groupBy. NONVISUAL можно использовать только в выражении SUMMARIZECOLUMNS.

Пример

```DAX
DEFINE
MEASURE FactInternetSales[Sales] = SUM(FactInternetSales[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    NONVISUAL(TREATAS({2007, 2008}, DimDate[CalendarYear])),
    "Sales", [Sales],
    "Visual Total Sales", CALCULATE([Sales],
ALLSELECTED(DimDate[CalendarYear]))
)
ORDER BY [CalendarYear]
```

Возвращает результат, в котором [визуальный итог продаж] — это общее количество в течение всех лет:

Развернуть таблицу

DimDate[CalendarYear]	[Sales]	[Визуальный итог продаж]
2007	9,791,060.30	29,358,677.22
2008	9,770,899.74	29,358,677.22

В отличие от этого, один и тот же запрос *без* функции NONVISUAL:

DAX

```
DEFINE
MEASURE FactInternetSales[Sales] = SUM(FactInternetSales[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    TREATAS({2007, 2008}, DimDate[CalendarYear]),
    "Sales", [Sales],
    "Visual Total Sales", CALCULATE([Sales],
ALLSELECTED(DimDate[CalendarYear]))
)
ORDER BY [CalendarYear]
```

Возвращает результат, в котором [визуальный итог продаж] — это общее число в течение двух выбранных лет:

Развернуть таблицу

DimDate[CalendarYear]	[Sales]	[Визуальный итог продаж]
2007	9,791,060.30	19,561,960.04
2008	9,770,899.74	19,561,960.04

C ROLLUPADDISSUBTOTAL

Добавление синтаксиса ROLLUPADDISSUBTOTAL изменяет поведение функции SUMMARIZECOLUMNS путем добавления строк свертки или промежуточных итогов в результат на основе столбцов groupBy_columnName. ROLLUPADDISSUBTOTAL можно использовать только в выражении SUMMARIZECOLUMNS.

Пример с одним промежуточным итогом

DAX

```
DEFINE
VAR vCategoryFilter =
  TREATAS({"Accessories", "Clothing"}, Product[Category])
VAR vSubcategoryFilter =
  TREATAS({"Bike Racks", "Mountain Bikes"}, Product[Subcategory])
EVALUATE
  SUMMARIZECOLUMNS
  (
    ROLLUPADDISSUBTOTAL
    (
      Product[Category], "IsCategorySubtotal", vCategoryFilter,
      Product[Subcategory], "IsSubcategorySubtotal", vSubcategoryFilter
    ),
    "Total Qty", SUM(Sales[Qty])
  )
  ORDER BY
  [IsCategorySubtotal] DESC, [Category],
  [IsSubcategorySubtotal] DESC, [Subcategory]
```

Возвращает следующую таблицу:

Развернуть таблицу

Категория	Подкатегории	IsCategorySubtotal	IsSubcategorySubtotal	Всего Qty
		Истинный	Истинный	60398
Аксессуары		Ложный	Истинный	36092
Аксессуары	Багажники для велосипедов	Ложный	Ложный	328
Велосипеды	Горные велосипеды	Ложный	Ложный	4970
Clothing		Ложный	Истинный	9101

Пример с несколькими промежуточными итогами

DAX

```
SUMMARIZECOLUMNS (
    Regions[State], ROLLUPADDISSUBTOTAL ( Sales[CustomerId],
"IsCustomerSubtotal" ),
```

```
    ROLLUPADDISSUBTOTAL ( Sales[Date], "IsDateSubtotal"), "Total Qty", SUM(
Sales[Qty] )
)
```

Продажи группируются по состоянию, по клиенту, по дате, с промежуточными итогими для 1. Продажи по состоянию на 2. Продажи по состоянию, клиент 3. Свернутый как на клиенте, так и на сегодняшний день, ведущий к продажам по состоянию.

Возвращает следующую таблицу:

Развернуть таблицу

CustomerID	IsCustomerSubtotal	Штат	Всего Qty	Дата	IsDateSubtotal
a	FALSE	WA	5	7/10/2014	
Б	FALSE	WA	1	7/10/2014	
Б	FALSE	WA	2	7/11/2014	
C	FALSE	OR	2	7/10/2014	
C	FALSE	OR	1	7/11/2014	
	TRUE	WA	6	7/10/2014	
	TRUE	WA	2	7/11/2014	
	TRUE	OR	2	7/10/2014	
	TRUE	OR	1	7/11/2014	
a	FALSE	WA	5		TRUE
Б	FALSE	WA	3		TRUE
C	FALSE	OR	3		TRUE
	TRUE	WA	8		TRUE
	TRUE	OR	3		TRUE

С ROLLUPGROUP

Как и функция SUMMARIZE, ROLLUPGROUP можно использовать вместе с ROLLUPADDISSUBTOTAL, чтобы указать, какие суммарные группы или детализированные (промежуточные итоги) следует включить, уменьшая

количество возвращаемых промежуточных строк. ROLLUPGROUP можно использовать только в SUMMARIZECOLUMNS или SUMMARIZE выражении.

Пример с несколькими промежуточными итогами

DAX

```
SUMMARIZECOLUMNS(
    ROLLUPADDISSUBTOTAL( Sales[CustomerId], "IsCustomerSubtotal" ),
    ROLLUPADDISSUBTOTAL(ROLLUPGROUP(Regions[City], Regions[State]),
"IsCityStateSubtotal"),"Total Qty", SUM( Sales[Qty] )
)
```

По-прежнему группируются по городу и штату, но свернутые вместе при составлении промежуточных итогов возвращает следующую таблицу.

Развернуть таблицу

Штат	CustomerId	IsCustomerSubtotal	Всего Qty	Город	IsCityStateSubtotal
WA	a	FALSE	2	Bellevue	FALSE
WA	Б	FALSE	2	Bellevue	FALSE
WA	a	FALSE	3	Редмонд	FALSE
WA	Б	FALSE	1	Редмонд	FALSE
OR	C	FALSE	3	Портленд	FALSE
WA		TRUE	4	Bellevue	FALSE
WA		TRUE	4	Редмонд	FALSE
OR		TRUE	3	Портленд	FALSE
	a	FALSE	5		FALSE
	Б	FALSE	3		TRUE
	C	FALSE	3		TRUE
		TRUE	11		TRUE

Contextual SummarizeColumns

Общие сведения

До февраля 2023 года SUMMARIZECOLUMNS не поддерживали оценку в рамках перехода контекста вообще. В продуктах, выпущенных до этого месяца, это ограничение сделало SUMMARIZECOLUMNS не полезным в большинстве мер , не удалось вызвать меру SUMMARIZECOLUMNS в любом случае перехода контекста, включая другие SUMMARIZECOLUMNS инструкции.

С февраля 2023 года переход на контекст поддерживается в нескольких сценариях, но не во всех условиях. Поддерживаемые и ограниченные случаи:

Развернуть таблицу

Тип SummarizeColumns	Внешний фильтр с одним столбцом	Внешний фильтр с несколькими столбцами	Внешние столбцы GroupBy
СуммированиеColumns только с GroupBy	OK	OK	OK
Сводка посolumns с фильтрами и мерами	OK	ERROR	ERROR

Начиная с июня 2024 г. мы включаем контекстные суммыColumns, что позволяет суммироватьcolumns в любом переходе контекста, в мере SummarizeColumns теперь полностью поддерживается:

Развернуть таблицу

Тип SummarizeColumns	Внешний фильтр с одним столбцом	Внешний фильтр с несколькими столбцами	Внешние столбцы GroupBy
СуммированиеColumns только с GroupBy	OK	OK	OK
Сводка посolumns с фильтрами и мерами	OK	OK	OK

Однако это обновление также включает изменения в поведение СуммеColumns, что может изменить результаты существующих выражений:

Семантика SelfValue для внешних фильтров

Мы представляем семантическую концепцию с именем SelfValue, которая изменяет способ взаимодействия фильтров из внешних таблиц со столбцами GroupBy в SummarizeColumns. Это изменение запрещает фильтры из другой таблицы, чтобы повлиять на столбцы GroupBy, даже если таблицы связаны через связь по фильтрам. Пример, демонстрирующий влияние этого изменения, включает в себя следующее выражение:

```
CalculateTable(
  SummarizeColumns(
      'Reseller Sales'[ResellerKey],
      'Reseller Sales'[ProductKey]
  ),
  Treatas({(229)}, 'Product'[Product Key])
)
```

Перед этим обновлением фильтр TreatAs будет применяться к операции GroupBy в SummarizeColumns, используя связь между Product[Product"[Product Key] и "Sales" [ProductKey]. Следовательно, результаты запроса будут содержать только строки, в которых "Продажи торговых посредников" [ProductKey] равно 229. Однако после обновления столбцы GroupBy в СуммеColumns больше не будут фильтруться по столбцам из внешних таблиц, даже если между ними существует связь. Поэтому в приведенном выше примере столбец GroupBy "Sales"[ProductKey] не будет фильтроваться по столбцу Product[ProductKey]. В результате запрос будет содержать строки, в которых "Продажи торгового посредника"[ProductKey] не равно 229.

Если вы предпочитаете сохранить предыдущее поведение, вы можете перезаписать выражение с помощью суммы вместо суммарного кода, как показано ниже:

```
CalculateTable(
    SUMMARIZE(
        'Reseller Sales',
        [ResellerKey],
        [ProductKey]
    ),
    Treatas({(229)}, 'Product'[Product Key])
)
```

Это перезаписное выражение сохраняет исходную семантику, в которой операция GroupBy не влияет на ограничение SelfValue, введенное обновлением.

Проверка строк для столбцов группирования, полностью охваченных Treatas

До этого обновления в функции SummarizeColumns все столбцы GroupBy из определенной таблицы полностью охватываются одним фильтром Treatas из той же таблицы, как показано ниже:

```
SummarizeColumns(
  Geography[Country],
  Geography[State],
  Treatas(
      {("United States", "Alberta")},
      Geography[Country],
      Geography[State]
  )
)
```

Результат приведенного выше запроса будет включать все строки, указанные в фильтре Treatas, независимо от того, являются ли они допустимыми или нет. Например, результатом будет таблица с одной строкой ("Соединенные Штаты", "Альберта"), даже если такая строка с [Страна] = "Соединенные Штаты" и [Штат] = "Альберта" существовала в таблице Geography.

Эта проблема известна и устранена обновлением. После обновления такие недопустимые строки будут отфильтрованы, и будут возвращены только допустимые строки из таблицы GroupBy. Таким образом, результат для приведенного выше запроса будет пустым, так как в таблице Geography отсутствуют допустимые строки, соответствующие указанным значениям [Страна] и [State].

Запретить смешанные keepfilters/overriddefilters в одной таблице или кластере

Последнее обновление ввело временное ограничение, которое активирует сообщение об ошибке, указывающее:

```
"SummarizeColumns filters with keepfilters behavior and overridefilters
behavior are mixed within one cluster, which is not allowed. Consider adding
keepfilters() to all filters of summarizecolumns."
```

Эта ошибка возникает, когда обычные фильтры (которые переопределяют существующие фильтры) и фильтры с указанными KeepFilters присутствуют в одной таблице или кластере. Например:

```
Evaluate CalculateTable(
  SummarizeColumns(
      Product[Color],
      KeepFilters(
          TreatAs(
              {( "Washington")}
              , Geography[State]
          )
      ),
      TreatAs(
          {("United States"), ("Canada")}
          , Geography[Country]
      )
  )
  ,TreatAs({("Alberta")}, Geography[State])
  ,TreatAs({("Canada")}, Geography[Country])
)
```

В приведенном выше выражении существует два фильтра в таблице Geography: один с указанными KeepFilters и один без. Эти фильтры перекрываются внешними фильтрами в разных столбцах. В настоящее время эта конфигурация не допускается, так как во внутреннем режиме два фильтра кластеризованы в одну, и система не может определить правильное поведение переопределения фильтра для кластеризованного фильтра в целом в таких случаях.

Обратите внимание, что это ограничение является временным. Мы активно разрабатываем решения для удаления этого ограничения в будущих обновлениях. При возникновении этой ошибки рекомендуется настроить фильтры в SummarizeColumns, добавив или удалив KeepFilters при необходимости, чтобы обеспечить согласованное переопределение в каждой таблице.

Связанное содержимое

SUMMARIZE

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Конструктор таблиц

Статья • 23.01.2025

Возвращает таблицу одного или нескольких столбцов.

Синтаксис

DAX

```
{ <scalarExpr1>, <scalarExpr2>, … }
{ ( <scalarExpr1>, <scalarExpr2>, … ), ( <scalarExpr1>, <scalarExpr2>, … ),
… }
```

Параметры

Развернуть таблицу

Срок	Определение
`scalarExprN`	Любое выражение DAX, возвращающее скалярное значение.

Возвращаемое значение

Таблица одного или нескольких столбцов. Если существует только один столбец, имя столбца — Значение. Если есть N-столбцы, в которых N > 1, имена столбцов слева направо: Value1, Value2, ..., ValueN.

Замечания

- Первый синтаксис возвращает таблицу одного столбца. Второй синтаксис возвращает таблицу одного или нескольких столбцов.
- Число скалярных выражений должно быть одинаковым для всех строк.
- Если типы данных для столбца отличаются в разных строках, все значения преобразуются в общий тип данных.

Пример 1

Следующие запросы DAX:

```
DAX

EVALUATE { 1, 2, 3 }
```

и

```
DAX

EVALUATE { (1), (2), (3) }
```

Верните следующую таблицу одного столбца:

Развернуть таблицу

[значение]
1
2
3

Пример 2

Следующий запрос DAX:

```
DAX

EVALUATE
    {
        (1.5, DATE(2017, 1, 1), CURRENCY(199.99), "A"),
        (2.5, DATE(2017, 1, 2), CURRENCY(249.99), "B"),
        (3.5, DATE(2017, 1, 3), CURRENCY(299.99), "C")
    }
```

Возвращает

Развернуть таблицу

[Значение1]	[Значение2]	[Значение3]	[Значение4]
1.5	1/1/2017	199.99	A

[Значение1]	[Значение2]	[Значение3]	[Значение4]
2.5	1/2/2017	249.99	B
3.5	1/3/2017	299.99	C

Пример 3

Следующий запрос DAX:

```
DAX

EVALUATE { 1, DATE(2017, 1, 1), TRUE, "A" }
```

Возвращает следующую таблицу одного столбца типа данных String:

Развернуть таблицу

[значение]
1
1/1/2017
TRUE
A

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TOPN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает верхние N-строки указанной таблицы.

Синтаксис

DAX

```
TOPN(<N_Value>, <Table>, <OrderBy_Expression>, [<Order>[,
<OrderBy_Expression>, [<Order>]]…])
```

Параметры

Развернуть таблицу

Параметр	Определение
`N_Value`	Количество возвращаемых строк. Любое выражение DAX, возвращающее скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста). См. примечания, чтобы лучше понять, когда количество возвращаемых строк может превышать `n_value`.
`Table`	Любое выражение DAX, возвращающее таблицу данных, из которой извлекаются верхние строки n. Сведения о том, когда возвращается пустая таблица, см. в разделе "Примечания".
`OrderBy_Expression`	(Необязательно) Любое DAX выражение, в котором значение результата используется для сортировки таблицы и оценки для каждой строки `table`.
`Order`	(Необязательно) Значение, указывающее, как сортировать значения `OrderBy_Expression`: - `0` (ноль) или `FALSE`. Сортировка по убыванию значений `Order`. Значение по умолчанию, если параметр `Order` опущен. - `1` или `TRUE`. Ранжирует в порядке возрастания `OrderBy`.

Возвращаемое значение

Таблица с верхними строками N `Table` или пустой таблицей, если `N_Value` равно 0 (ноль) или меньше. Строки не сортируются в определенном порядке.

Замечания

- Если в `Order_By` значениях имеется галстук, в строке N-го таблицы возвращаются все привязанные строки. Затем при наличии связей в N-й строке функция может возвращать больше n строк.
- Если N_Value равно 0 (ноль) или меньше, TOPN возвращает пустую таблицу.
- TOPN не гарантирует порядок сортировки результатов.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула меры возвращает первые 10 проданных продуктов по сумме продаж.

DAX

```
= SUMX(
        TOPN(
            10,
            SUMMARIZE(
                    InternetSales,
                    InternetSales[ProductKey],
                    "TotalSales", SUM(InternetSales[SalesAmount])
            ),
            [TotalSales], DESC
        ),
        [TotalSales]
)
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TREATAS

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Применяет результат табличного выражения в качестве фильтров к столбцам из не связанной таблицы.

Синтаксис

DAX

```
TREATAS(table_expression, <column>[, <column>[, <column>[,…]]]} )
```

Параметры

Развернуть таблицу

Термин	Определение
`table_expression`	Выражение, которое приводит к таблице.
`column`	Один или несколько существующих столбцов. Это не может быть выражением.

Возвращаемое значение

Таблица, содержащая все строки в столбцах, которые также находятся в table_expression.

Замечания

- Число указанных столбцов должно соответствовать количеству столбцов в выражении таблицы и иметь одинаковый порядок.
- Если значение, возвращаемое в табличном выражении, не существует в столбце, оно игнорируется. Например, TREATAS({"Красный", "Зеленый", "Желтый"}, DimProduct[Color]) задает фильтр для столбца DimProduct[Color] с тремя значениями "Красный", "Зеленый" и "Желтый". Если значение "Желтый"

не существует в DimProduct[Color], то значения эффективного фильтра будут иметь значение "Красный" и "Зеленый".

- Лучше всего использовать, если связь не существует между таблицами. При наличии нескольких связей между таблицами рекомендуется использовать вместо этого USERELATIONSHIP.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере модель содержит две несвязанные таблицы продуктов. Если пользователь применяет фильтр к DimProduct1[ProductCategory], выбрав "Велосипеды", "Места", "Шины", "Велосипеды", "Места", "Шины" применяется к DimProduct2[ProductCategory].

DAX

```
CALCULATE(
    SUM(Sales[Amount]),
    TREATAS(
        VALUES(DimProduct1[ProductCategory]),
        DimProduct2[ProductCategory]
    )
)
```

Связанное содержимое

INTERSECT FILTER USERELATIONSHIP

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

UNION

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Создает таблицу объединения (соединения) из пары таблиц.

Синтаксис

DAX

```
UNION(<table_expression1>, <table_expression2> [,<table_expression>]…)
```

Параметры

Развернуть таблицу

Термин	Определение
`table_expression`	Любое выражение DAX, возвращающее таблицу.

Возвращаемое значение

Таблица, содержащая все строки из каждого из двух выражений таблицы.

Замечания

- Две таблицы должны иметь одинаковое количество столбцов.
- Столбцы объединяются по позиции в соответствующих таблицах.
- Имена столбцов в возвращаемой таблице будут соответствовать именам столбцов в table_expression1.
- Повторяющиеся строки сохраняются.
- Возвращаемая таблица имеет происхождение по возможности. Например, если первый столбец каждого table_expression имеет происхождение к одному базовому столбцу C1 в модели, первый столбец в результатах UNION

будет иметь происхождение до C1. Однако если объединенные столбцы имеют происхождение в разные базовые столбцы или есть столбец расширения, результирующий столбец в UNION не будет иметь происхождения.

- Если типы данных отличаются, результирующий тип данных определяется на основе правил приведения типов данных.
- Возвращаемая таблица не будет содержать столбцы из связанных таблиц.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующее выражение создает объединение путем объединения таблицы USAInventory и таблицы INDInventory в одну таблицу:

```
DAX

UNION(UsaInventory, IndInventory)
```

USAInventory

Развернуть таблицу

Страна или регион	Штат	Считать	Итог
USA	Целостности и доступности	5	500
USA	WA	10	900

INDInventory

Развернуть таблицу

Страна или регион	Штат	Считать	Итог
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

Возвращаемая таблица,

Развернуть таблицу

Страна или регион	Штат	Считать	Итог
USA	Целостности и доступности	5	500
USA	WA	10	900
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

VALUES

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Если входной параметр является именем столбца, возвращает таблицу с одним столбцом, содержащую различные значения из указанного столбца. Повторяющиеся значения удаляются и возвращаются только уникальные значения. Можно добавить BLANK значение. Если входной параметр является именем таблицы, возвращает строки из указанной таблицы. Повторяющиеся строки сохраняются. Можно добавить строку BLANK.

> **Примечание**
>
> Эту функцию нельзя использовать для возврата значений в ячейку или столбец на листе; скорее, вы используете его в качестве промежуточной функции, вложенной в формулу, чтобы получить список различных значений, которые можно считать или использовать для фильтрации или суммы других значений.

Синтаксис

DAX

```
VALUES(<TableNameOrColumnName>)
```

Параметры

Развернуть таблицу

Термин	Определение
`TableName` или `ColumnName`	Столбец, из которого возвращаются уникальные значения, или таблица, из которой должны быть возвращены строки.

Возвращаемое значение

Если входной параметр является именем столбца, одна таблица столбцов. Если входной параметр является именем таблицы, возвращается таблица с теми же столбцами.

Замечания

- При использовании функции VALUES в контексте, отфильтрованном, уникальные значения, возвращаемые VALUES, влияют на фильтр. Например, если вы фильтруете по регионам и возвращаете список значений для города, список будет содержать только те города в регионах, которые разрешены фильтром. Чтобы вернуть все города независимо от существующих фильтров, необходимо использовать функцию ALL для удаления фильтров из таблицы. Второй пример демонстрирует использование ALL с VALUES.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).
- Рекомендации по использованию VALUESсм. в разделе Использование SELECTEDVALUE вместо VALUES.

Связанные функции

В большинстве сценариев, когда аргумент является именем столбца, результаты функции VALUES идентичны результатам функции `DISTINCT`. Обе функции удаляют дубликаты и возвращают список возможных значений в указанном столбце. Однако функция VALUES также может возвращать пустое значение. Это пустое значение полезно в случаях, когда вы ищете различные значения из связанной таблицы, но значение, используемое в связи, отсутствует из одной таблицы. В терминологии базы данных это называется нарушением целостности ссылок. Такие несоответствия в данных могут возникать при обновлении одной таблицы, а связанная таблица не является.

Если аргумент является именем таблицы, результат функции VALUES возвращает все строки в указанной таблице плюс пустую строку, если имеется нарушение целостности ссылок. Функция DISTINCT удаляет повторяющиеся строки и возвращает уникальные строки в указанной таблице.

> ⓘ **Примечание**

Функция DISTINCT позволяет использовать имя столбца или любое допустимое табличное выражение, но функция VALUES принимает только имя столбца или имя таблицы в качестве аргумента.

В следующей таблице приводится сводка несоответствия между данными, которые могут возникать в двух связанных таблицах, когда целостность ссылок не сохраняется.

Развернуть таблицу

Таблица MyOrders	Таблица MySales
1 июня	1 июня продажи
2 июня	2 июня продажи
(даты заказа не были введены)	3 июня продажи

Если функция DISTINCT используется для возврата списка дат, будут возвращены только две даты. Однако если вы используете функцию VALUES, функция возвращает два даты плюс дополнительный пустой элемент. Кроме того, любая строка из таблицы MySales, которая не имеет даты сопоставления в таблице MyOrders, будет "сопоставлена" с этим неизвестным элементом.

Пример

Следующая формула подсчитывает количество уникальных счетов (заказов на продажу) и выдает следующие результаты при использовании в отчете, включающее имена категорий продуктов:

DAX

```
= COUNTROWS(VALUES('InternetSales_USD'[SalesOrderNumber]))
```

Возвраты

Развернуть таблицу

Метки строк	Подсчет счетов
Аксессуары	18,208
Велосипеды	15,205

Метки строк	Подсчет счетов
Clothing	7,461
Общий итог	27,659

Связанное содержимое

функция FILTERCOUNTROWS функциифильтрации

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Текстовые функции

Статья • 22.01.2025

Выражения анализа данных (DAX) включают набор текстовых функций на основе библиотеки строковых функций в Excel, но которые были изменены для работы с таблицами и столбцами в табличных моделях. В этом разделе описываются текстовые функции, доступные на языке DAX.

В этой категории

Развернуть таблицу

Функция	Описание
COMBINEVALUES	Объединяет две или несколько текстовых строк в одну текстовую строку.
CONCATENATE	Объединяет две текстовые строки в одну текстовую строку.
CONCATENATEX	Объединяет результат выражения, вычисляемого для каждой строки в таблице.
EXACT	Сравнивает две текстовые строки и возвращает `TRUE`, если они совпадают, `FALSE` в противном случае.
FIND	Возвращает начальную позицию одной текстовой строки в другой текстовой строке.
FIXED	Округляет число до указанного числа десятичных знаков и возвращает результат в виде текста.
FORMAT	Преобразует значение в текст в соответствии с указанным форматом.
LEFT	Возвращает указанное число символов из начала текстовой строки.
LEN	Возвращает количество символов в текстовой строке.
LOWER	Преобразует все буквы в текстовой строке в строчные буквы.
MID	Возвращает строку символов из середины текстовой строки, учитывая начальную позицию и длину.
REPLACE	REPLACE заменяет часть текстовой строки на основе числа указанных символов с другой текстовой строкой.
REPT	Повторяет текст с заданным числом раз.
RIGHT	RIGHT возвращает последний символ или символы в текстовой строке в

Функция	Описание
	зависимости от количества указанных символов.
SEARCH	Возвращает число символов, в которых сначала найдена определенная символьная или текстовая строка, считывающая слева направо.
SUBSTITUTE	Заменяет существующий текст новым текстом в текстовой строке.
TRIM	Удаляет все пробелы из текста, за исключением отдельных пробелов между словами.
UNICHAR	Возвращает символ Юникода, на который ссылается числовое значение.
UNICODE	Возвращает числовый код, соответствующий первому символу текстовой строки.
UPPER	Преобразует текстовую строку во все прописные буквы.
VALUE	Преобразует текстовую строку, представляющую число в число.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

COMBINEVALUES

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Объединяет две или несколько текстовых строк в одну текстовую строку. Основной целью этой функции является поддержка связей с несколькими столбцами в моделях DirectQuery. Дополнительные сведения см. в примечаниях.

Синтаксис

DAX

```
COMBINEVALUES(<delimiter>, <expression>, <expression>[, <expression>]…)
```

Параметры

Развернуть таблицу

Срок	Определение
`delimiter`	Разделитель, используемый во время объединения. Должно быть константным значением.
`expression`	Выражение DAX, значение которого будет присоединено к одной текстовой строке.

Возвращаемое значение

Сцепленная строка.

Замечания

- Функция COMBINEVALUES предполагает, но не проверяет, что если входные значения отличаются, выходные строки также отличаются. Исходя из этого предположения, когда COMBINEVALUES используется для создания вычисляемых столбцов для создания связи, которая объединяет несколько столбцов из двух таблиц DirectQuery, оптимизированное условие соединения создается во время запроса. Например, если пользователи хотят создать связь

между Table1(Column1, Column2) и Table2(Column1, Column2), они могут создать два вычисляемых столбца, по одному в каждой таблице, как:

```
DAX

Table1[CalcColumn] = COMBINEVALUES(",", Table1[Column1],
Table1[Column2])
```

и

```
DAX

Table2[CalcColumn] = COMBINEVALUES(",", Table2[Column1],
Table2[Column2])
```

А затем создайте связь между `Table1[CalcColumn]` и `Table2[CalcColumn]`. В отличие от других DAX функций и операторов, которые переводятся буквально в соответствующие операторы и функции SQL, приведенные выше отношения создают предикат соединения SQL следующим образом:

```
DAX

(Table1.Column1 = Table2.Column1 OR Table1.Column1 IS NULL AND
Table2.Column1 IS NULL)
```

и

```
DAX

(Table1.Column2 = Table2.Column2 OR Table1.Column2 IS NULL AND
Table2.Column2 IS NULL)
```

- Предикат соединения может обеспечить гораздо лучшую производительность запросов, чем один из них, включающий сложные операторы и функции SQL.
- Функция COMBINEVALUES зависит от пользователей, чтобы выбрать соответствующий разделитель, чтобы гарантировать, что уникальные сочетания входных значений создают отдельные выходные строки, но не проверяют, является ли предположение истинным. Например, если пользователи выбирают `"| "` в качестве разделителя, но одна строка в Таблице1 имеет `Table1[Column1] = "| "` и `Table2 [Column2] = " "`, а одна строка в Table2 имеет `Table2[Column1] = " "` и `Table2[Column2] = "| "`, два объединенных выходных данных будут одинаковыми `"|| "`, которые, как

представляется, указывают на то, что две строки совпадают в операции соединения. Две строки не объединяются, если обе таблицы находятся из одного источника DirectQuery, хотя они объединяются вместе, если обе таблицы импортируются.

Пример

Следующий запрос DAX:

DAX

```
EVALUATE
DISTINCT (
    SELECTCOLUMNS ( Date, "Month", COMBINEVALUES ( ", ", [MonthName],
[CalendarYear] ) )
)
```

Возвращает следующую таблицу одного столбца:

Развернуть таблицу

[месяц]
Январь 2020 г.
Февраль 2020 г.
Март 2020 г.
Апрель 2020 г.
Май 2020 г.
Июнь 2020 г.
Июль 2020 г.
Август 2020 г.
Сентябрь 2020 г.
Октябрь 2020 г.
Ноябрь 2020 г.
Декабрь 2020 г.
Январь 2021 г.

[месяц]
Январь 2021 г.
Февраль 2021 г.
Март 2021 г.
Апрель 2021 г.
Май 2021 г.
Июнь 2021 г.
Июль 2021 г.
Август 2021 г.
Сентябрь 2021 г.
Октябрь 2021 г.
Ноябрь 2021 г.
Декабрь 2021 г.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CONCATENATE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Объединяет две текстовые строки в одну текстовую строку.

Синтаксис

DAX

```
CONCATENATE(<text1>, <text2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`text1`	Первая текстовая строка, которая должна быть присоединена к одной текстовой строке. Строка может содержать текст или числа. Также можно использовать ссылки на столбцы.
`text2`	Вторая текстовая строка, которая должна быть присоединена к одной текстовой строке. Строка может содержать текст или числа. Также можно использовать ссылки на столбцы.

Возвращаемое значение

Сцепленная строка.

Замечания

- Функция CONCATENATE объединяет две текстовые строки в одну текстовую строку. Присоединенные элементы могут быть текстом, числами, логическими значениями, представленными как текст, или сочетанием этих элементов. Можно также использовать ссылку на столбец, если столбец содержит соответствующие значения.

- Функция CONCATENATE в DAX принимает только два аргумента, а функция Excel CONCATENATE принимает до 255 аргументов. Если необходимо объединить несколько столбцов, можно создать ряд вычислений или использовать оператор объединения (`&`) для объединения всех столбцов в более простом выражении.
- Если вы хотите напрямую использовать текстовые строки, а не использовать ссылку на столбец, необходимо заключить каждую строку в двойные кавычки.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример: объединение литералы

Пример формулы создает новое строковое значение путем объединения двух строковых значений, которые предоставляются в качестве аргументов.

DAX

```
= CONCATENATE("Hello ", "World")
```

Пример: объединение строк в столбцах

Пример формулы возвращает полное имя клиента, как указано в телефонной книге. Обратите внимание, как вложенная функция используется во втором аргументе. Это один из способов объединения нескольких строк при наличии более двух значений, которые необходимо использовать в качестве аргументов.

DAX

```
= CONCATENATE(Customer[LastName], CONCATENATE(", ", Customer[FirstName]))
```

Пример: условное объединение строк в столбцах

В примере формулы создается новый вычисляемый столбец в таблице Customer с полным именем клиента в виде сочетания имени, среднего и фамилии. Если по промежуточному имени нет, фамилия поступает непосредственно после первого

имени. Если имеется промежуточное имя, используется только первая буква среднего имени, а за начальной буквой следует период.

```
DAX
= CONCATENATE( [FirstName]&" ", CONCATENATE( IF( LEN([MiddleName])>1,
LEFT([MiddleName],1)&". ", ""), [LastName]))
```

Эта формула использует вложенные CONCATENATE и функции IF вместе с оператором амперсанда (`&`) для условного объединения трех строковых значений и добавления пробелов в качестве разделителей.

Пример. Объединение столбцов с различными типами данных

В следующем примере показано, как сцепить значения в столбцах с различными типами данных. Если значение, которое вы объединяете, числовое, значение будет неявно преобразовано в текст. Если оба значения являются числовыми, оба значения будут приведение к тексту и сцеплены, как если бы они были строками.

Развернуть таблицу

Описание продукта	Сокращение продукта (столбец 1 составного ключа)	Номер продукта (столбец 2 составного ключа)	Новый созданный ключевой столбец
Горный велосипед	MTN	40	MTN40
Горный велосипед	MTN	42	MTN42

```
DAX
= CONCATENATE('Products'[Product abbreviation],'Products'[Product number])
```

Функция CONCATENATE в DAX принимает только два аргумента, а функция Excel CONCATENATE принимает до 255 аргументов. Если вам нужно добавить дополнительные аргументы, можно использовать оператор ampersand (&). Например, следующая формула выдает результаты, MTN-40 и MTN-42.

```
DAX
```

```
= [Product abbreviation] & "-" & [Product number]
```

Связанное содержимое

CONCATENATEX

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CONCATENATEX

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Объединяет результат выражения, вычисляемого для каждой строки в таблице.

Синтаксис

DAX

```
CONCATENATEX(<table>, <expression>[, <delimiter> [, <orderBy_expression> [, <order>]]...])
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Таблица, содержащая строки, для которых будет оцениваться выражение.
`expression`	Выражение для каждой строки `table`.
`delimiter`	(Необязательно) Разделитель, используемый во время объединения.
`orderBy_expression`	(Необязательно) Любое DAX выражение, в котором значение результата используется для сортировки объединенных значений в выходной строке. Вычисляется для каждой строки `table`.
`order`	(Необязательно) Значение, указывающее, как сортировать `orderBy_expression` значения, возрастание или убывание.

Необязательный параметр `order` принимает следующие значения:

Развернуть таблицу

Ценность	Альтернативные значения	Описание
0 (ноль)	`FALSE`, `DESC`	Сортировка по убыванию значений `orderBy_expression`. Это значение по умолчанию, если параметр `order`

Ценность	Альтернативные значения	Описание
		опущен.
1	`TRUE`, `ASC`	Сортировка по возрастанию значений `orderBy_expression`.

Возвращаемое значение

Сцепленная строка.

Замечания

- Эта функция принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Второй аргумент — это столбец, содержащий значения, которые нужно объединить, или выражение, возвращающее значение.
- Сцепленные значения не обязательно сортируются в определенном порядке, если не указано `orderBy_expression`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Таблица "Сотрудники"

Развернуть таблицу

FirstName	LastName
Алан	Пивовар
Майкл	Блайт

Следующая формула:

```
DAX
= CONCATENATEX(Employees, [FirstName] & " " & [LastName], ",")
```

Возвращается: "Алан Пивовар, Майкл Блит"

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EXACT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Сравнивает две текстовые строки и возвращает `TRUE`, если они одинаковы, в противном случае возвращается `FALSE`. EXACT учитывает регистр, но игнорирует различия форматирования. EXACT учитывает регистр

Синтаксис

DAX

```
EXACT(<text1>,<text2>)
```

Параметры

Развернуть таблицу

Срок	Определение
`text1`	Первая текстовая строка или столбец, содержащий текст.
`text2`	Вторая текстовая строка или столбец, содержащий текст.

Возвращаемое значение

True или False. (Логический)

Пример

Следующая формула, используемая в вычисляемом столбце в таблице Product, проверяет значение Product для текущей строки по значению модели для текущей строки и возвращает значение True, если они одинаковы, и возвращает значение False, если они отличаются.

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе

DAX примера модели.

```
DAX
= EXACT([Product], [Model])
```

Связанное содержимое

текстовые функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FIND

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает начальную позицию одной текстовой строки в другой текстовой строке. FIND учитывает регистр.

Синтаксис

DAX

```
FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

Параметры

Развернуть таблицу

Срок	Определение
`find_text`	Текст, который нужно найти. Используйте двойные кавычки (пустой текст), чтобы соответствовать первому символу в `within_text`.
`within_text`	Текст, содержащий нужный текст.
`start_num`	(необязательно) Символ, с которого начинается поиск; Значение , если опущено, `start_num` = 1. Первый символ в `within_text` — номер 1.
`NotFoundValue`	(необязательно, но настоятельно рекомендуется) Значение, которое должно быть возвращено, если операция не находит соответствующую подстроку, обычно 0, -1 или BLANK(). Если не указано, возвращается ошибка.

Возвращаемое значение

Номер, показывающий начальную точку текстовой строки, которую вы хотите найти.

Замечания

- В то время как в Microsoft Excel есть несколько версий функции FIND для размещения однобайтового набора символов (SBCS) и двухбайтового набора символов (DBCS), DAX использует Юникод и подсчитывает каждый символ одинаково; Таким образом, не требуется использовать другую версию в зависимости от типа символа.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).
- FIND не поддерживает подстановочные знаки. Чтобы использовать подстановочные знаки, используйте SEARCH.

Пример

Следующий DAX запрос находит позицию первого буквы "Bike" в строке, содержащей имя торгового посредника. Если он не найден, возвращается пустое значение.

Помните, что FIND учитывает регистр. В этом примере, если в аргументе `find_text` использовался "велосипед", результаты не будут возвращены. Используйте SEARCH для учета регистра.

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

DAX

```
EVALUATE
CALCULATETABLE (
    ADDCOLUMNS (
        TOPN ( 10, SUMMARIZE('Reseller', [Reseller], [Business Type])),
        "Position of Bike", FIND ( "Bike", 'Reseller'[Reseller], 1, BLANK ()
)
    ),
    'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added
Reseller", "Warehouse"}
)
```

Возвращает

Развернуть таблицу

Перекупщик	Бизнес-тип	Положение велосипеда
Продавцы велосипедов томов	Склад	8
Массовый рынок велосипедов	Добавленный торговый посредник по значению	13
Циклы двойников	Добавленный торговый посредник по значению	
Богатый магазин универмага	Склад	
Коллекция прокатов	Специализированный магазин велосипедов	
Бюджетный магазин toy Store	Склад	
Глобальный спортивный выход	Склад	
Каталог велосипедов в Интернете	Склад	8
Шлемы и циклы	Добавленный торговый посредник по значению	
Jumbo Bikes	Специализированный магазин велосипедов	7

Связанное содержимое

функций текста

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FIXED

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Округляет число до указанного числа десятичных знаков и возвращает результат в виде текста. Можно указать, что результат возвращается с запятыми или без нее.

Синтаксис

DAX

```
FIXED(<number>, <decimals>, <no_commas>)
```

Параметры

Развернуть таблицу

Срок	Определение
`number`	Число, которое нужно округить и преобразовать в текст, или столбец, содержащий число.
`decimals`	(необязательно) Число цифр справа от десятичной запятой; Значение 2.
`no_commas`	(необязательно) Логическое значение: если 1, не отображайте запятые в возвращаемом тексте; Если значение 0 или опущено, отобразите запятые в возвращаемом тексте.

Возвращаемое значение

Число, представленное как текст.

Замечания

- Если значение, используемое для параметра `decimals`, отрицательное, `number` округляется слева от десятичной точки.
- Если не `decimals`, предполагается, что значение равно 2.

- Если значение `no_commas` равно 0 или опущено, возвращенный текст включает запятые как обычно.
- Основное различие между форматированием ячейки, содержащей число, с помощью команды и форматирования числа непосредственно с функцией FIXED заключается в том, что FIXED преобразует результат в текст. Число отформатировано с помощью команды из меню форматирования по-прежнему является числом.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула, используемая в вычисляемом столбце, получает числовое значение текущей строки в Product[List Price] и возвращает его в виде текста с 2 десятичными знаками и без запятых.

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

DAX

```
= FIXED([List Price],2,1)
```

Связанное содержимое

CEILING FLOOR ISO.CEILING MROUND ROUND ROUNDDOWN ROUNDUP

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FORMAT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует значение в текст в соответствии с указанным форматом.

Синтаксис

DAX

```
FORMAT(<value>, <format_string>[, <locale_name>])
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Значение или выражение, которое оценивается как одно значение.
`format_string`	Строка с шаблоном форматирования.
`locale_name`	(Необязательно) Имя языкового стандарта, используемого функцией. Возможные значения — это строки, принятые функцией API Windows LocaleNameToLCID().

Возвращаемое значение

Строка, содержащая `value` отформатированную `format_string`.

> **Примечание**
>
> Если `value` BLANK, функция возвращает пустую строку.
>
> Если `format_string` BLANK, значение форматируется с форматом "Общее число" или "Общая дата" (в соответствии с типом данных `value`).

Замечания

- Стандартные строки форматирования используют свойство языка и региональных параметров модели при форматировании результата. По умолчанию свойство языка и региональных параметров модели устанавливается в соответствии с языковым стандартом пользователя компьютера. Для новых моделей Power BI Desktop свойство языка и региональных параметров модели можно изменить в параметрах > региональных параметров > языке модели. Для служб Analysis Services язык и региональные параметры модели задаются в соответствии со свойством Language, изначально определенным экземпляром.

- Строки форматирования, поддерживаемые в качестве аргумента функции DAXFORMAT, основаны на строках формата, используемых Visual Basic (OLE Automation), а не на строках формата, используемых платформой .NET Framework. Таким образом, вы можете получить непредвиденные результаты или ошибку, если аргумент не соответствует определенным строкам формата. Например, "p" в качестве сокращенного значения "Процент" не поддерживается. Строки, предоставляемые в качестве аргумента функции FORMAT, не включенные в список предопределенных строк форматирования, обрабатываются как часть пользовательской строки форматирования или как строковый литерал.

- Дополнительные сведения об указании языкового стандарта с FORMATсм. в этом видео.

- Использование FORMAT изменяет результат меры на текстовый тип данных. Если результат меры изначально является числовым типом данных, то с FORMATмера не может использоваться в визуальных элементах, где для разделов значений требуется числовой тип данных, как и диаграммы. В Power BI можно также использовать строки динамического формата **для мер,** указать строку условного формата, которая поддерживает числовый тип данных меры.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Примеры

Форматирование строк

DAX

```
= FORMAT( 12345.67, "General Number")
= FORMAT( 12345.67, "Currency")
= FORMAT( 12345.67, "Fixed")
= FORMAT( 12345.67, "Standard")
= FORMAT( 12345.67, "Percent")
= FORMAT( 12345.67, "Scientific")
```

Возвращает:

12345.67 "Общее число" отображает число без форматирования.

$12,345.67 "Валюта" отображает номер с форматированием языкового стандарта валюты. В примере ниже показано форматирование валют США по умолчанию.

12345.67 "Исправлено" отображает по крайней мере одну цифру слева от десятичного разделителя и две цифры справа от десятичного разделителя.

12 345.67 "Стандартный" отображает по крайней мере одну цифру слева от десятичного разделителя и две цифры справа от десятичного разделителя и включает в себя разделители тысяч. В примере ниже показано форматирование номеров США по умолчанию.

1234 567,00 % "Процент" отображает число в процентах (умножено на 100) с форматированием и знаком процента справа от числа, разделенного одним пробелом.

1.23E+04 "Scientific" отображает число в научной нотации с двумя десятичными цифрами.

Datetime с необязательными locale_name

DAX

```
= FORMAT( dt"2020-12-15T12:30:59", BLANK(), "en-US" )
= FORMAT( dt"2020-12-15T12:30:59", BLANK(), "en-GB" )
= FORMAT( dt"2020-12-15T12:30:59", "mm/dd/yyyy", "en-GB" )
```

Возвращает:

12.15.2020 12:30:59 Где месяц предшествует дню и времени— 12-часовой формат.

15.12.2020 12:30:59 Где день предшествует месяцу и времени— 24-часовой формат.

12.15.2020 12:30:59 Где месяц предшествует дню и времени— 24-часовой формат. Так как указана строка, не зависящая от языкового стандарта, языковой стандарт не применяется и возвращается формат, отличный от языкового стандарта.

Предопределенные числовые форматы

В аргументе `format_string` можно указать следующие предопределенные числовые форматы:

Развернуть таблицу

`Format`	**Описание**
`"General Number"`	Отображает число без разделителей тысяч.
`"Currency"`	Отображает число с разделителями тысяч, при необходимости; отображает две цифры справа от десятичного разделителя. Выходные данные основаны на параметрах языкового стандарта системы.
`"Fixed"`	Отображает по крайней мере одну цифру слева и две цифры справа от десятичного разделителя.
`"Standard"`	Отображает число с разделителями тысяч, по крайней мере одна цифра слева и две цифры справа от десятичного разделителя.
`"Percent"`	Отображает число, умноженное на 100 с знаком процента (%), добавленным сразу справа; всегда отображает две цифры справа от десятичного разделителя.
`"Scientific"`	Использует стандартную научно-нотацию, предоставляя две значимые цифры.
`"Yes/No"`	Отображает значение No, если число равно 0; в противном случае отображается "Да".
`"True/False"`	Отображает значение False, если число равно 0; в противном случае отображается значение True.
`"On/Off"`	Отображает значение Off, если число равно 0; в противном случае отображается вкл.

Настраиваемые числовые форматы

Пользовательское выражение формата для чисел может содержать от одного до трех разделов, разделенных точкой с запятой. Если аргумент строки формата

содержит один из именованных числовых форматов, допускается только один раздел.

Развернуть таблицу

Если вы используете	Результатом является
Только один раздел	Выражение формата применяется ко всем значениям.
Два раздела	Первый раздел применяется к положительным значениям и нулям, второй — к отрицательным значениям.
Три раздела	Первый раздел относится к положительным значениям, второму к отрицательным значениям и третьему к нулю.

DAX

```
"$#,##0;($#,##0)"
```

Если между ними нет запятой, отсутствует раздел определяется с использованием формата положительного значения. Например, следующий формат отображает положительные и отрицательные значения, используя формат в первом разделе, и отображает "Ноль", если значение равно нулю.

DAX

```
"$#,##0"
```

Если между ними нет запятой, отсутствует раздел отображается с использованием формата положительного значения.

Пользовательские числовые символы форматирования

В аргументе `format_string` можно указать следующие пользовательские числовые символы:

Развернуть таблицу

Характер	Описание
Никакой	Отображение номера без форматирования.

Характер	Описание
(`0`)	Заполнитель цифр. Отображение цифры или нуля. Если выражение имеет цифру в позиции, где 0 отображается в строке форматирования, отобразите его; в противном случае отображается ноль в этой позиции. Если число имеет меньше цифр, чем нули (на любой стороне десятичного разряда) в выражении формата, отображает начальные или конечные нули. Если число имеет больше цифр справа от десятичного разделителя, чем справа от десятичного разделителя справа от десятичного разделителя в выражении форматирования, округите число до сколько десятичных разрядов, сколько имеет нули. Если число имеет больше цифр слева от десятичного разделителя, чем слева от десятичного разделителя слева от десятичного разделителя, отобразите дополнительные цифры без изменений.
(`#`)	Заполнитель цифр. Отображение цифры или ничего. Если выражение имеет цифру в позиции, где #отображается в строке форматирования, отобразите его; в противном случае не отображается ничего в этой позиции. Этот символ работает так же, как заполнитель цифр 0, за исключением того, что начальные и конечные нули не отображаются, если число имеет одинаковые или меньше цифр, чем символы # на обеих сторонах десятичного разделителя в выражении формата.
(`.`)	Заполнитель десятичных знаков. В некоторых языковых стандартах запятая используется в качестве десятичного разделителя. Заполнитель десятичного разряда определяет, сколько цифр отображается слева и справа от десятичного разделителя. Если выражение формата содержит только знаки числа слева от этого символа, числа меньше 1 начинаются с десятичного разделителя. Чтобы отобразить начальный ноль, отображаемый дробными числами, используйте 0 в качестве заполнителя первой цифры слева от десятичного разделителя. Фактический символ, используемый в качестве десятичного заполнителя в отформатированных выходных данных, зависит от формата числа, распознанного системой.
(`%`)	Заполнитель процента. Выражение умножается на 100. Символ процента (`%`) вставляется в положение, где оно отображается в строке формата.
(`,`)	Разделитель тысяч. В некоторых языковых стандартах период используется в качестве разделителя тысяч. Разделитель тысяч отделяет тысячи от сотен в пределах числа с четырьмя или более местами слева от десятичного разделителя. Стандартное использование разделителя тысяч указывается, если формат содержит разделитель тысяч, окруженный заполнителями цифр (`0` или `#`). Два смежных разделителя тысяч или разделитель тысяч сразу слева от десятичного разделителя (независимо от того, задано ли десятичное значение) означает "масштабировать число, разделив его на 1000, округляя по мере необходимости". Например, можно использовать строку формата "#0", чтобы представить 100 миллионов как 100. Числа меньше 1 миллиона отображаются как 0. Две смежные разделители тысяч в любой позиции, отличной от немедленного слева от десятичного разделителя, обрабатываются просто как указание использования разделителя тысяч. Фактический символ, используемый

Характер	Описание
	в качестве разделителя тысяч в отформатированных выходных данных, зависит от формата числа, распознанного системой.
(`:`)	Разделитель времени. В некоторых языковых стандартах другие символы могут использоваться для представления разделителя времени. Разделитель времени разделяет часы, минуты и секунды при форматировании значений времени. Фактический символ, используемый в качестве разделителя времени в отформатированных выходных данных, определяется параметрами системы.
(`/`)	Разделитель дат. В некоторых языковых стандартах другие символы могут использоваться для представления разделителя дат. Разделитель дат отделяет день, месяц и год, когда значения дат форматируются. Фактический символ, используемый в качестве разделителя дат в отформатированных выходных данных, определяется параметрами системы.
(`E- E+ e- e+`)	Научный формат. Если выражение формата содержит по крайней мере один заполнитель цифр (`0` или `#`) справа от E-, E+, e-или e+, число отображается в научном формате, и E или e вставляется между числом и его экспонентом. Число заполнителей цифр справа определяет количество цифр в экспоненте. Используйте E-или e-, чтобы поместить знак минуса рядом с отрицательными экспонентами. Используйте E+ или e+, чтобы поместить знак минуса рядом с отрицательными экспонентами и знак плюса рядом с положительными экспонентами.
`- + $` ( )	Отображение литерального символа. Чтобы отобразить символ, отличный от одного из перечисленных, предшествуйте ему с обратной косой чертой (`\`) или заключите его в двойные кавычки ("").
(`\\`)	Отображение следующего символа в строке формата. Чтобы отобразить символ с особым значением в качестве литерального символа, предшествуйте ему обратную косую черту (`\`). Обратная косая черта не отображается. Использование обратной косой черты совпадает с заключением следующего символа в двойные кавычки. Чтобы отобразить обратную косую черту, используйте две обратные косые черты (`\\`). Примеры символов, которые не могут отображаться в виде литеральных символов, являются символами форматирования даты и времени (a, c, d, h, m, n, p, q, s, t, w, y, /, и :), числовые символы форматирования (#, 0, %, E, e, запятая и период), а также символы форматирования строк (@, &, <, >и !).
("ABC")	Отображение строки внутри двойных кавычки (" ").

Стандартные форматы даты и времени

Следующие предопределенные форматы даты и времени можно указать в аргументе `format_string`. При использовании форматов, отличных от этих, они интерпретируются как настраиваемый формат даты и времени:

Развернуть таблицу

Формат	Описание
`"General Date"`	Отображает дату и (или) время. Например, 3.12.2008 11:07:31 AM. Отображение даты определяется текущим значением языка и региональных параметров приложения.
`"Long Date"` или `"Medium Date"`	Отображает дату в соответствии с форматом длинной даты текущего языка и региональных параметров. Например, среда, 12 марта 2008 г.
`"Short Date"`	Отображает дату с использованием короткого формата даты и региональных параметров. Например, 3.12.2008.
`"Long Time"` или	Отображает время с использованием длительного формата языка и региональных параметров; обычно включает в себя часы, минуты, секунды. Например, 11:07:31 AM.
`"Medium Time"`	Отображает время в формате 12 часов. Например, 11:07.
`"Short Time"`	Отображает время в формате 24 часа. Например, 11:07.

Настраиваемые форматы даты и времени

Следующие символы формата можно указать в `format_string` для создания настраиваемых форматов даты и времени:

Развернуть таблицу

Характер	Описание
(`:`)	Разделитель времени. В некоторых языковых стандартах другие символы могут использоваться для представления разделителя времени. Разделитель времени разделяет часы, минуты и секунды при форматировании значений времени. Фактический символ, используемый в качестве разделителя времени в отформатированных выходных данных, определяется параметрами системы.
(`/`)	Разделитель дат. В некоторых языковых стандартах другие символы могут использоваться для представления разделителя дат. Разделитель дат отделяет день, месяц и год, когда значения дат форматируются. Фактический символ, используемый в качестве разделителя дат в отформатированных выходных данных, определяется параметрами системы.
(`\`)	Обратная косая черта. Отображает следующий символ в виде литерального символа. Таким образом, он не интерпретируется как символ форматирования.
(`"`)	Двойная цитата. Отображается текст, заключенный в двойные кавычки. Таким образом, он не интерпретируется как символы форматирования.

Характер	Описание
`c`	Отображение даты как `ddddd` и отображение времени как `ttttt` в этом порядке. Отображение только сведений о дате, если до серийного номера даты отсутствует дробная часть; Отображение только сведений о времени, если целочисленная часть отсутствует.
`d`	Отображение дня в виде числа без начального нуля (1–31).
`dd`	Отображение дня в виде числа с начальным нулем (01-31).
`ddd`	Отображение дня в виде аббревиаций (Sun-Sat). Локализованный.
`dddd`	Отображение дня в виде полного имени (Sunday-Saturday). Локализованный.
`ddddd`	Отображение даты в виде полной даты (включая день, месяц и год), отформатированную в соответствии с параметром формата короткой даты системы. Формат короткой даты по умолчанию — `mm/dd/yyyy`.
`dddddd`	Отображайте серийный номер даты в виде полной даты (включая день, месяц и год), отформатированный в соответствии с параметром длинной даты, распознаваемой системой. Формат длинной даты по умолчанию — `dddd, mmmm d, yyyy`.
`w`	Отображение дня недели в виде числа (1 в воскресенье до 7 в субботу).
`ww`	Отображение недели года в виде числа (1-54).
`m`	Отображение месяца в виде числа без начального нуля (1–12). Если `m` немедленно следует `h` или `hh`, минуту, а не месяц отображается.
`mm`	Отображение месяца в виде числа с начальным нулем (01-12). Если `mm` немедленно следует `h` или `hh`, минуту, а не месяц отображается.
`mmm`	Отображение месяца в виде аббревиаций (Jan-Dec). Локализованный.
`mmmm`	Отображение месяца в виде полного имени месяца (January-December). Локализованный.
`q`	Отображение квартала года в виде числа (1–4).
`y`	Отображение дня года в виде числа (1–366).
`yy`	Отображение года в виде 2-цифрного числа (00-99).
`yyyy`	Отображение года в виде 4-цифрного номера (100-9999).
`h`	Отображение часа в виде числа без начального нуля (0–23).
`hh`	Отображение часа в виде числа с начальным нулем (00–23).
`n`	Отображение минуты в виде числа без начального нуля (0–59).

Характер	Описание
`nn`	Отображение минуты в виде числа с начальным нулем (00–59).
`s`	Отображение второго в виде числа без начального нуля (0–59).
`ss`	Отображение второго в виде числа с начальным нулем (00–59).
`ttttt`	Отображение времени в виде полного времени (включая час, минуту и секунду), отформатированный с помощью разделителя времени, определенного в формате времени, распознаваемом системой. Начальный нуль отображается, если выбран начальный нулевой параметр и время до 10:00 утра или вечера. Формат времени по умолчанию — `h:mm:ss`.
`AM/PM`	Используйте 12-часовые часы и отображайте верхний регистр am с любым часом до полудня; отображение верхнего регистра PM с любым часом между полуднем и 11:59 вечера.
`am/pm`	Используйте 12-часовые часы и отображайте строчные буквы am с любым часом до полудня; отображение нижнего регистра PM с любым часом от полудня до 11:59 вечера.
`A/P`	Используйте 12-часовые часы и отображайте прописные буквы A с любым часом до полудня; отображение верхнего регистра P с любым часом от полудня до 11:59 вечера.
`a/p`	Используйте 12-часовые часы и отображайте строчные буквы A с любым часом до полудня; Отображает строчные буквы P с любым часом между полденем и 11:59 вечера.
`AMPM`	Используйте 12-часовые часы и отображайте строковый литерал AM, определенный системой в любой час до полудня; Отображение строкового литерала PM в соответствии с системой с любым часом между полуднем и 11:59 P.M. AMPM может быть либо верхним или нижним регистром, но регистр строки, отображаемой строкой, соответствует строке, определенной параметрами системы. Формат по умолчанию — AM/PM. Если для системы задано значение 24-часового времени, строка обычно имеет пустую строку.

Форматирование даты и времени использует языковой стандарт текущего пользователя для форматирования строки. Например, рассмотрим дату 25 июня 2020 года. Если форматируется с помощью строки формата "m/d/гггг", она будет:

- Языковой стандарт пользователя — Соединенные Штаты Америки (en-US): "6.25.2020"
- Языковой стандарт пользователя — Германия (de-DE): "6.25.2020"

Примеры настраиваемого формата даты и времени

В следующих примерах используется дата и время четверга, 25 июня 2020 г. в 1:23:45. Германия (de-DE) использует 24-часовую систему. Нет эквивалента AM/PM.

Развернуть таблицу

Формат	Результат (en-US)	Результат (de-DE)
`"c"`	06/25/2020 13:23:45	25.06.2020 13:23:45
`"d"`	25	25
`"dd"`	25	25
`"ddd"`	Отпечаток	Делать
`"dddd"`	Четверг	Donnerstag
`"ddddd"`	06/25/2020	25.06.2020
`"dddddd"`	Четверг, 25 июня 2020 г.	Доннерстаг, 25. Juni 2020
`"w"`	5	5
`"ww"`	26	26
`"m"`	6	6
`"mm"`	06	06
`"mmm"`	Июн	Июн
`"mmmm"`	Июнь	Юни
`"q"`	2	2
`"y"`	177	177
`"yy"`	20	20
`"yyyy"`	2020	2020
`"""Year"" yyyy"`	Год 2020 г.	Год 2020 г.
`"yyyy \Qq"`	2020 Q2	2020 Q2
`"dd/mm/yyyy"`	25/06/2020	25.06.2020
`"mm/dd/yyyy"`	06/25/2020	06.25.2020
`"h:nn:ss"`	13:23:45	13:23:45
`"h:nn:ss AMPM"`	1:23:45 вечера	1:23:45

Формат	Результат (en-US)	Результат (de-DE)
`"hh:nn:ss"`	13:23:45	13:23:45
`"hh:nn:ss AMPM"`	01:23:45 вечера	01:23:45
`"ttttt"`	13:23:45	13:23:45
`"ttttt AMPM"`	13:23:45 вечера	13:23:45
`"mm/dd/yyyy hh:nn:ss AMPM"`	06.25.2020 01:23:45	6.25.2020 01:23:45

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LEFT

19.06.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает указанное число символов из начала текстовой строки.

Синтаксис

```
DAX
```

```
LEFT(<text>, <num_chars>)
```

Параметры

Развернуть таблицу

Срок	Определение
`text`	Текстовая строка, содержащая символы, которые требуется извлечь, или ссылка на столбец, содержащий текст.
`num_chars`	(необязательно) Количество символов, которые требуется извлечь LEFT; Значение 1, если опущено.

Возвращаемое значение

Текстовая строка.

Замечания

- В то время как Microsoft Excel содержит различные функции для работы с текстом в однобайтовых и двухбайтовых языках символов, DAX работает с Юникодом и сохраняет все символы одинаковой длины; поэтому достаточно одной функции.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере возвращаются первые пять символов города компании в столбце [Город] и первые пять букв ключа торгового посредника в столбце [ResellerKey] и объединяют их для создания идентификатора.

DAX

```
= CONCATENATE ( LEFT ( 'Reseller'[City], 5 ), LEFT ( 'Reseller'[ResellerKey], 5 )
)
```

Если аргумент `num_chars` является числом, превышающим число доступных символов, функция возвращает максимальные доступные символы и не вызывает ошибку. Например, столбец [ResellerKey] содержит такие числа, как 5, 24 и 312; поэтому результат также имеет переменную длину.

Связанное содержимое

текстовые функции

LEN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает количество символов в текстовой строке.

Синтаксис

```DAX
LEN(<text>)
```

Параметры

Развернуть таблицу

Срок	Определение
`text`	Текст, длина которого требуется найти, или столбец, содержащий текст. Пробелы считаются символами.

Возвращаемое значение

Целое число, указывающее количество символов в текстовой строке.

Замечания

- В то время как в Microsoft Excel имеются различные функции для работы с однобайтами и двухбайтами символьных языков, DAX использует Юникод и сохраняет все символы с одинаковой длиной.
- LEN всегда подсчитывает каждый символ как 1, независимо от того, какой параметр языка по умолчанию имеет значение.
- Если вы используете LEN со столбцом, который содержит нетекстовые значения, например даты или логические значения, функция неявно приводит значение к тексту с помощью текущего формата столбца.

Пример

Следующая формула суммирует длину адресов в столбцах , [AddressLine1] и [AddressLine2].

DAX

```
= LEN([AddressLine1])+LEN([AddressLin2])
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LOWER

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует все буквы в текстовой строке в строчные буквы.

Синтаксис

```
DAX
LOWER(<text>)
```

Параметры

Развернуть таблицу

Срок	Определение
`text`	Текст, который требуется преобразовать в нижний регистр, или ссылку на столбец, содержащий текст.

Возвращаемое значение

Текст в нижнем регистре.

Замечания

Символы, которые не являются буквами, не изменяются. Например, формула `=LOWER("123ABC")` возвращает `123abc`.

Пример

Следующая формула получает каждую строку в столбце [ProductCode], а значение преобразуется во все строчные регистры. Числа в столбце не затрагиваются.

```
DAX
```

```
= LOWER('New Products'[ProductCode])
```

Связанное содержимое

текстовые функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MID

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Возвращает строку символов из середины текстовой строки, учитывая начальную позицию и длину.

Синтаксис

```DAX
MID(<text>, <start_num>, <num_chars>)
```

Параметры

Развернуть таблицу

Срок	Определение
`text`	Текстовая строка, из которой требуется извлечь символы или столбец, содержащий текст.
`start_num`	Позиция первого символа, которую требуется извлечь. Позиции начинаются с 1.
`num_chars`	Количество возвращаемых символов.

Возвращаемое значение

Строка текста указанной длины.

Замечания

В то время как Microsoft Excel имеет различные функции для работы с однобайтами и двухбайтами символов, DAX использует Юникод и сохраняет все символы с одинаковой длиной.

Примеры

Следующее выражение:

```
DAX
MID("abcde",2,3))
```

Возвращает `"bcd"`.

Следующее выражение:

```
DAX
MID('Reseller'[ResellerName],1,5))
```

Возвращает тот же результат, что и `LEFT([ResellerName],5)`. Оба выражения возвращают первые 5 букв столбца, `[ResellerName]`.

Связанное содержимое

текстовые функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

REPLACE

Статья • 21.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

REPLACE заменяет часть текстовой строки на основе числа указанных символов с другой текстовой строкой.

Синтаксис

```DAX
REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)
```

Параметры

Развернуть таблицу

Термин	Определение
`old_text`	Строка текста, содержащего символы, которые нужно заменить, или ссылка на столбец, содержащий текст.
`start_num`	Позиция символа в `old_text`, которую вы хотите заменить `new_text`.
`num_chars`	Количество символов, которые требуется заменить. **Предупреждение:** Если аргумент, `num_chars`, является пустым или ссылается на столбец, который вычисляет пустое значение, строка для `new_text` вставляется в положение `start_num` без замены каких-либо символов. Это то же поведение, что и в Excel.
`new_text`	Замещающий текст для указанных символов в `old_text`.

Возвращаемое значение

Текстовая строка.

Замечания

- В то время как Microsoft Excel имеет различные функции для использования с однобайтами и двухбайтами символьных языков, DAX использует Юникод и

поэтому сохраняет все символы одинаковой длины.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующая формула создает новый вычисляемый столбец, который заменяет первые два символа кода продукта в столбце [ProductCode], на новый двухбуквенный код OB.

DAX

```
= REPLACE('New Products'[Product Code],1,2,"OB")
```

Связанное содержимое

функции текстовых функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

REPT

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Повторяет текст с заданным числом раз. Используйте REPT для заполнения ячейки несколькими экземплярами текстовой строки.

Синтаксис

DAX

```
REPT(<text>, <num_times>)
```

Параметры

Развернуть таблицу

Термин	Определение
`text`	Текст, который требуется повторить.
`num_times`	Положительное число, указывающее количество повтора текста.

Возвращаемое значение

Строка, содержащая изменения.

Замечания

- Если `num_times` равно 0 (ноль), REPT возвращает пустое значение.
- Если `num_times` не является целым числом, оно усечено.
- Результат функции REPT не может превышать 32 767 символов или REPT возвращает ошибку.

Пример: повторяющиеся литеральные строки

В следующем примере возвращается строка, 85, повторяющаяся три раза.

```
DAX
= REPT("85",3)
```

Пример: повторяющиеся значения столбцов

В следующем примере возвращается строка в столбце [MyText], повторяющаяся для количества раз в столбце [MyNumber]. Так как формула расширяется для всего столбца, результирующая строка зависит от текста и числа в каждой строке.

```
DAX
= REPT([MyText],[MyNumber])
```

Развернуть таблицу

MyText	MyNumber	CalculatedColumn1
Текст	2	TextText
Число	0	
85	3	858585

Связанное содержимое

Текстовые функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

RIGHT

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

RIGHT возвращает последний символ или символы в текстовой строке в зависимости от количества указанных символов.

Синтаксис

DAX

```
RIGHT(<text>, <num_chars>)
```

Параметры

Развернуть таблицу

Термин	Определение
`text`	Текстовая строка, содержащая символы, которые требуется извлечь, или ссылка на столбец, содержащий текст.
`num_chars`	(необязательно) Количество символов, которые требуется извлечь RIGHT; опущено, 1. Вы также можете использовать ссылку на столбец, содержащий числа.

Если ссылка на столбец не содержит текст, он неявно приведение в виде текста.

Возвращаемое значение

Текстовая строка, содержащая указанные правильные символы.

Замечания

- RIGHT всегда подсчитывает каждый символ, будь то один байт или двойной байт, как 1, независимо от того, какой параметр языка по умолчанию имеет значение.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример: возврат фиксированного числа символов

Следующая формула возвращает последние две цифры кода продукта в таблице New Products.

DAX

```
= RIGHT('New Products'[ProductCode],2)
```

Пример. Использование ссылки на столбец для указания числа символов

Следующая формула возвращает переменное число цифр из кода продукта в таблице New Products в зависимости от числа в столбце MyCount. Если в столбце нет значения, MyCount или значения является пустым, RIGHT также возвращает пустое значение.

DAX

```
= RIGHT('New Products'[ProductCode],[MyCount])
```

Связанное содержимое

текстовые функцииLEFTMID

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SEARCH

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает число символов, в которых сначала найдена определенная символьная или текстовая строка, считывающая слева направо. Поиск не учитывает регистр и учитывает акцент.

Синтаксис

DAX

```
SEARCH(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

Параметры

Развернуть таблицу

Термин	Определение
`find_text`	Текст, который требуется найти. Можно использовать подстановочные знаки — вопросительный знак (?) и звездочку (*) в `find_text`. Вопросительный знак соответствует любому одному символу; звездочка соответствует любой последовательности символов. Если вы хотите найти фактический вопросительный знак или звездочку, введите тильду (~) перед символом.
`within_text`	Текст, в котором требуется искать `find_text`, или столбец, содержащий текст.
`start_num`	(необязательно) Позиция символа в `within_text`, с которой требуется начать поиск. Если опущено, 1.
`NotFoundValue`	(необязательно, но настоятельно рекомендуется) Значение, которое должно быть возвращено, если операция не находит соответствующую подстроку, обычно 0, -1 или BLANK(). Если не указано, возвращается ошибка.

Возвращаемое значение

Число начальной позиции первой текстовой строки из первого символа второй текстовой строки.

Замечания

- Функция поиска не учитывает регистр. Поиск "N" находит первое вхождение "N" или "n".
- Функция поиска учитывает акцент. Поиск "á" найдет первое вхождение "á", но никаких вхождения "a", "à" или прописных версий "A", "Á".
- Функцию SEARCH можно использовать для определения расположения символа или текстовой строки в другой текстовой строке, а затем использовать функцию MID для возврата текста или использовать функцию REPLACE для изменения текста.
- Если `find_text` не удается найти в `within_text`, формула возвращает ошибку. Это поведение похоже на Excel, которое возвращает #VALUE, если подстрока не найдена. Значения NULL в `within_text` будут интерпретированы как пустая строка в этом контексте.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

Следующий DAX запрос находит позицию первой буквы цикла в строке, содержащей имя торгового посредника. Если он не найден, возвращается пустое значение.

SEARCH не учитывает регистр. В этом примере, если "cycle" или "Cycle" используется в аргументе `find_text`, результаты возвращаются в любом случае. Используйте FIND для учета регистра.

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

DAX

```
EVALUATE
CALCULATETABLE (
```

```
    ADDCOLUMNS (
        TOPN ( 10, SUMMARIZE('Reseller', [Reseller], [Business Type])),
        "Position of cycle", SEARCH ( "cycle", 'Reseller'[Reseller], 1,
BLANK () )
    ),
    'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added
Reseller", "Warehouse"}
)
```

Возвращает

Развернуть таблицу

Reseller	Business Type	Position of cycle
Volume Bike Sellers	Склад	
Mass Market Bikes	Добавленный торговый посредник по значению	
Twin Cycles	Добавленный торговый посредник по значению	6
Rich Department Store	Склад	
Rental Gallery	Специализированный магазин велосипедов	
Budget Toy Store	Склад	
Global Sports Outlet	Склад	
Online Bike Catalog	Склад	
Шлемы и циклы	Добавленный торговый посредник по значению	13
Jumbo Bikes	Специализированный магазин велосипедов	

Связанное содержимое

FIND REPLACE функции текста

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SUBSTITUTE

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Заменяет существующий текст новым текстом в текстовой строке.

Синтаксис

```
DAX
SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)
```

Параметры

Развернуть таблицу

Термин	Определение
`text`	Текст, в котором требуется заменить символы, или ссылку на столбец, содержащий текст.
`old_text`	Существующий текст, который требуется заменить.
`new_text`	Текст, на который нужно заменить `old_text`.
`instance_num`	(необязательно) Вхождение `old_text`, которое требуется заменить. Если опущено, все экземпляры `old_text` заменяются

Возвращаемое значение

Строка текста.

Замечания

- Используйте функцию SUBSTITUTE, если вы хотите заменить определенный текст в текстовой строке; используйте функцию REPLACE, если вы хотите заменить любой текст переменной длины, которая возникает в определенном расположении в текстовой строке.

- Функция SUBSTITUTE учитывает регистр. Если регистр не соответствует `text` и `old_text`, SUBSTITUTE не заменит текст.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример: подстановка в строке

Следующая формула создает копию столбца [Product Code], заменяющего новый код продукта `NW` для старого кода продукта `PA` где бы он ни был в столбце.

DAX

```
= SUBSTITUTE([Product Code], "NW", "PA")
```

Связанное содержимое

текстовые функцииREPLACE

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TRIM

Статья • 23.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ✅ визуального вычисления

Удаляет все пробелы из текста, за исключением отдельных пробелов между словами.

Синтаксис

```DAX
TRIM(<text>)
```

Параметры

Развернуть таблицу

Термин	Определение
`text`	Текст, из которого требуется удалить пробелы, или столбец, содержащий текст.

Возвращаемое значение

Строка с удаленными пробелами.

Замечания

- Используйте TRIM в тексте, полученном от другого приложения, которое может иметь неправильный интервал.
- Функция TRIM изначально была разработана для обрезки 7-разрядного пробела ASCII (значение 32) из текста. В наборе символов Юникода существует дополнительный пробел, который называется неразрывным пробелом, который имеет десятичное значение 160. Этот символ обычно используется на веб-страницах в качестве сущности HTML и >. Сама по себе функция TRIM не удаляет этот неразрывный пробел. Пример того, как

обрезать оба пробела из текста, см. в разделе "Удаление пробелов" и непечатаемых символов из текста.

Пример

В следующей формуле создается новая строка, которая не содержит пробела в пути.

```
DAX
= TRIM("A column with trailing spaces.   ")
```

При создании формулы формула распространяется по строке так же, как и при вводе, чтобы вы видели исходную строку в каждой формуле, а результаты не очевидны. Однако при вычислении формулы строка обрезается.

Вы можете убедиться, что формула выдает правильный результат, проверив длину вычисляемого столбца, созданного предыдущей формулой, следующим образом:

```
DAX
= LEN([Calculated Column 1])
```

Связанное содержимое

Текстовые функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

UNICHAR

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает символ Юникода, на который ссылается числовое значение.

Синтаксис

```
DAX

UNICHAR(number)
```

Параметры

Развернуть таблицу

Термин	Определение
`number`	Номер Юникода, представляющий символ.

Возвращаемое значение

Символ, представленный номером Юникода.

Замечания

- Если символы XML не являются недопустимыми, UNICHAR возвращает ошибку.
- Если номера Юникода являются частичными суррогатами и типы данных недопустимы, UNICHAR возвращает ошибку.
- Если числа являются числовыми значениями, которые выходят за пределы допустимого диапазона, UNICHAR возвращает ошибку.
- Если число равно нулю (0), UNICHAR возвращает ошибку.

- Возвращаемый символ Юникода может быть строкой символов, например в кодах UTF-8 или UTF-16.

Пример

В следующем примере возвращается символ, представленный номером Юникода 66 (верхний регистр A).

```
DAX

= UNICHAR(65)
```

В следующем примере возвращается символ, представленный номером Юникода 32 (пробел).

```
DAX

= UNICHAR(32)
```

В следующем примере возвращается символ, представленный номером Юникода 9733 (★ символ).

```
DAX

= UNICHAR(9733)
```

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

UNICODE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Возвращает число (кодовую точку), соответствующее первому символу текста.

Синтаксис

```DAX
UNICODE( <Text> )
```

Параметры

Развернуть таблицу

Термин	Определение
`Text`	Текст — это символ, для которого требуется значение Юникода.

Возвращаемое значение

Числовый код для первого символа в текстовой строке.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

UPPER

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует текстовую строку во все прописные буквы.

Синтаксис

```DAX
UPPER (<text>)
```

Параметры

Развернуть таблицу

Термин	Определение
`text`	Текст, преобразованный в верхний регистр, или ссылка на столбец, содержащий текст.

Возвращаемое значение

Тот же текст в верхнем регистре.

Пример

Следующая формула преобразует строку в столбце [ProductCode], во все верхние регистры. Не алфавитные символы не затрагиваются.

```DAX
= UPPER(['New Products'[Product Code])
```

Связанное содержимое

функции текстовых функций

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

VALUE

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

Преобразует текстовую строку, представляющую число в число.

Синтаксис

DAX

```
VALUE(<text>)
```

Параметры

Развернуть таблицу

Термин	Определение
`text`	Преобразованный текст.

Возвращаемое значение

Преобразованное число в десятичном типе данных.

Замечания

- Значение, переданное в качестве параметра `text`, может находиться в любом из постоянных, числовых, дат или форматов времени, распознаваемых приложением или службами, которые вы используете. Если `text` не входит в один из этих форматов, возвращается ошибка.
- Обычно не требуется использовать функцию VALUE в формуле, так как подсистема неявно преобразует текст в числа при необходимости.
- Также можно использовать ссылки на столбцы. Например, если у вас есть столбец, содержащий смешанные типы чисел, VALUE можно использовать для преобразования всех значений в один числовый тип данных. Однако если вы

используете функцию VALUE с столбцом, содержащим смешанные числа и текст, весь столбец помечается ошибкой, так как не все значения во всех строках можно преобразовать в числа.

Пример

Следующая формула преобразует типизированные строки "3" в числовое значение 3.

```
DAX
= VALUE("3")
```

Связанное содержимое

Текстовые функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Функции аналитики времени

Статья • 23.01.2025

Выражения анализа данных (DAX) включают функции аналитики времени, позволяющие управлять данными с помощью периодов времени, включая дни, месяцы, кварталы и годы, а затем создавать и сравнивать вычисления в течение этих периодов. Прежде чем использовать функции аналитики времени, обязательно пометьте одну из таблиц, содержащих столбец даты в виде таблицы даты.

В этой категории

Развернуть таблицу

Function	Description
CLOSINGBALANCEMONTH	Вычисляет выражение по последней дате месяца в текущем контексте.
CLOSINGBALANCEQUARTER	Вычисляет выражение по последней дате квартала в текущем контексте.
CLOSINGBALANCEYEAR	Вычисляет выражение по последней дате года в текущем контексте.
DATEADD	Возвращает таблицу, содержащую столбец дат, сдвигается либо вперед, либо назад во времени по указанному количеству интервалов от дат в текущем контексте.
DATESBETWEEN	Возвращает таблицу, содержащую столбец дат, начинающихся с указанной даты начала и продолжающейся до указанной даты окончания.
DATESINPERIOD	Возвращает таблицу, содержащую столбец дат, начинающихся с указанной даты начала, и продолжается для указанного числа и типа интервалов даты.
DATESMTD	Возвращает таблицу, содержащую столбец дат месяца на дату в текущем контексте.
DATESQTD	Возвращает таблицу, содержащую столбец дат для даты квартала в текущем контексте.
DATESYTD	Возвращает таблицу, содержащую столбец дат года к дате в текущем контексте.

Function	Description
ENDOFMONTH	Возвращает последнюю дату месяца в текущем контексте для указанного столбца дат.
ENDOFQUARTER	Возвращает последнюю дату квартала в текущем контексте для указанного столбца дат.
ENDOFYEAR	Возвращает последнюю дату года в текущем контексте для указанного столбца дат.
FIRSTDATE	Возвращает первую дату в текущем контексте для указанного столбца дат.
LASTDATE	Возвращает последнюю дату в текущем контексте для указанного столбца дат.
NEXTDAY	Возвращает таблицу, содержащую столбец всех дат со следующего дня, на основе первой даты, указанной в столбце дат в текущем контексте.
NEXTMONTH	Возвращает таблицу, содержащую столбец всех дат из следующего месяца, на основе первой даты в столбце дат в текущем контексте.
NEXTQUARTER	Возвращает таблицу, содержащую столбец всех дат в следующем квартале на основе первой даты, указанной в столбце дат в текущем контексте.
NEXTYEAR	Возвращает таблицу, содержащую столбец всех дат в следующем году, на основе первой даты в столбце дат в текущем контексте.
OPENINGBALANCEMONTH	Вычисляет выражение по первой дате месяца в текущем контексте.
OPENINGBALANCEQUARTER	Вычисляет выражение по первой дате квартала в текущем контексте.
OPENINGBALANCEYEAR	Вычисляет выражение по первой дате года в текущем контексте.
PARALLELPERIOD	Возвращает таблицу, содержащую столбец дат, который представляет период параллельно с датами в указанном столбце дат в текущем контексте, при этом даты сдвигали число интервалов в течение времени или назад.
PREVIOUSDAY	Возвращает таблицу, содержащую столбец всех дат, представляющих день, предшествующий первой дате в столбце дат в текущем контексте.

Function	Description
PREVIOUSMONTH	Возвращает таблицу, содержащую столбец всех дат из предыдущего месяца, на основе первой даты в столбце дат в текущем контексте.
PREVIOUSQUARTER	Возвращает таблицу, содержащую столбец всех дат из предыдущего квартала на основе первой даты в столбце дат в текущем контексте.
PREVIOUSYEAR	Возвращает таблицу, содержащую столбец всех дат предыдущего года, учитывая последнюю дату в столбце дат в текущем контексте.
SAMEPERIODLASTYEAR	Возвращает таблицу, содержащую столбец дат, сменяемых в год назад с дат в столбце указанных дат в текущем контексте.
STARTOFMONTH	Возвращает первую дату месяца в текущем контексте для указанного столбца дат.
STARTOFQUARTER	Возвращает первую дату квартала в текущем контексте для указанного столбца дат.
STARTOFYEAR	Возвращает первую дату года в текущем контексте для указанного столбца дат.
TOTALMTD	Вычисляет значение выражения для месяца на дату в текущем контексте.
TOTALQTD	Вычисляет значение выражения для дат в квартале к дате в текущем контексте.
TOTALYTD	Вычисляет текущее значение выражения в текущем контексте.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CLOSINGBALANCEMONTH

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Вычисляет `expression` по последней дате месяца в текущем контексте.

Синтаксис

DAX

```
CLOSINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

Параметры

Развернуть таблицу

Параметр	Определение
`expression`	Выражение, возвращающее скалярное значение.
`dates`	Столбец, содержащий даты.
`filter`	(необязательно) Выражение, указывающее фильтр для применения к текущему контексту.

Возвращаемое значение

Скалярное значение, представляющее `expression`, вычисляемое по последней дате месяца в текущем контексте.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

    > ⓘ **Примечание**
    >
    > Ограничения логических выражений описаны в CALCULATE функции.

    > ⓘ **Примечание**
    >
    > Выражение `filter` имеет ограничения, описанные в CALCULATE функции.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет значение инвентаризации по окончании месяца для инвентаризации продукта.

DAX

```
=
CLOSINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*Product
Inventory[UnitsBalance]),DateTime[DateKey])
```

Связанное содержимое

функции аналитики времениCLOSINGBALANCEYEAR
функцииCLOSINGBALANCEQUARTER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CLOSINGBALANCEQUARTER

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Оценивает `expression` по последней дате квартала в текущем контексте.

Синтаксис

DAX

```
CLOSINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Выражение, возвращающее скалярное значение.
`dates`	Столбец, содержащий даты.
`filter`	(необязательно) Выражение, указывающее фильтр для применения к текущему контексту.

Возвращаемое значение

Скалярное значение, представляющее `expression`, вычисляемое по последней дате квартала в текущем контексте.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

    > ⓘ **Примечание**
    >
    > Ограничения логических выражений описаны в CALCULATE функции.

    > ⓘ **Примечание**
    >
    > Выражение `filter` имеет ограничения, описанные в CALCULATE функции.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет "Стоимость инвентаризации за квартал" инвентаризации продукта.

DAX

```
=
CLOSINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*Produ
ctInventory[UnitsBalance]),DateTime[DateKey])
```

Связанное содержимое

функции аналитики времениCLOSINGBALANCEYEAR
функцииCLOSINGBALANCEMONTH

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

CLOSINGBALANCEYEAR

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Оценивает `expression` по последней дате года в текущем контексте.

Синтаксис

DAX

```
CLOSINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Выражение, возвращающее скалярное значение.
`dates`	Столбец, содержащий даты.
`filter`	(необязательно) Выражение, указывающее фильтр для применения к текущему контексту.
`year_end_date`	(необязательно) Литеральная строка с датой, определяющей дату окончания года. Значение по умолчанию — 31 декабря.

Возвращаемое значение

Скалярное значение, представляющее `expression`, вычисляемое по последней дате года в текущем контексте.

Замечания

- Параметр `year_end_date` — строковый литерал даты в том же языковом стандарте, что и языковой стандарт клиента, в котором была создана книга. Часть года даты игнорируется.
- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

    > ⓘ **Примечание**
    >
    > Ограничения логических выражений описаны в CALCULATE функции.

    > ⓘ **Примечание**
    >
    > Выражение `filter` имеет ограничения, описанные в CALCULATE функции.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже примере формулы создается мера, которая вычисляет "Стоимость инвентаризации конца года" инвентаризации продукта.

DAX

```
=
CLOSINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductI
nventory[UnitsBalance]),DateTime[DateKey])
```

Связанное содержимое

функции аналитики времениCLOSINGBALANCEYEAR функцииCLOSINGBALANCEQUARTERCLOSINGBALANCEMONTH функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DATEADD

Статья • 21.03.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец дат, сдвигается либо вперед, либо назад во времени по указанному количеству интервалов от дат в текущем контексте.

Синтаксис

```
DAX

DATEADD(<dates>,<number_of_intervals>,<interval>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.
`number_of_intervals`	Целое число, указывающее количество интервалов для добавления или вычитания из дат.
`interval`	Интервал, с помощью которого необходимо переместить даты. Значение интервала может быть одним из следующих: `year`, `quarter`, `month`, `day`

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

Аргумент `dates` может быть одним из следующих:

- Ссылка на столбец даты и времени,
- Табличное выражение, возвращающее один столбец значений даты и времени,
- Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

  > ⓘ **Примечание**
  >
  > Ограничения логических выражений описаны в разделе CALCULATE функции.

- Если число, указанное для `number_of_intervals` положительно, даты в `dates` перемещаются вперед в течение времени; Если число отрицательное, даты в `dates` сдвигаются назад.
- Параметр `interval` — это перечисление, а не набор строк; поэтому значения не должны быть заключены в кавычки. Кроме того, значения: `year`, `quarter`, `month`, `day` должны быть полны при их использовании.
- В таблице результатов содержатся только даты, существующие в столбце `dates`.
- Если даты в текущем контексте не образуют непрерывный интервал, функция возвращает ошибку.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример. Смена набора дат

Следующая формула вычисляет даты, которые за год до дат в текущем контексте.

DAX

```
= DATEADD(DateTime[DateKey],-1,year)
```

Специальное поведение

Если выбор включает последние два дня месяца, DATEADD будет использовать семантику расширения и будет включать дни до конца месяца. Например, когда 27 и 28 февраля 2013 года включены в выборку, а месяц добавляется, DATEADD возвращается 27 марта до 31 марта.

Это поведение происходит только в том случае, если последние два дня месяца включены в выборку. Если выбрано только 27 февраля, он перейдет к 27 марта.

```
DAX
```

```
= DATEADD(DateTime[DateKey], 1, month)
```

Связанное содержимое

функции аналитики временифункции даты и времени

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DATESBETWEEN

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец дат, начинающихся с указанной даты начала и продолжающейся до указанной даты окончания.

Эта функция подходит для передачи в качестве фильтра в функцию CALCULATE. Используйте его для фильтрации выражения по пользовательскому диапазону дат.

> **Примечание**
>
> Если вы работаете со стандартными интервалами дат, такими как дни, месяцы, кварталы или годы, рекомендуется использовать более подходящую функцию DATESINPERIOD.

Синтаксис

DAX

```
DATESBETWEEN(<Dates>, <StartDate>, <EndDate>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Dates`	Столбец даты.
`StartDate`	Выражение даты.
`EndDate`	Выражение даты.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- В наиболее распространенном случае использования `dates` является ссылкой на столбец даты помеченной таблицы дат.
- Если `StartDate` BLANK, `StartDate` будет самым ранним значением в столбце `dates`.
- Если `EndDate` BLANK, `EndDate` будет последним значением в столбце `dates`.
- Даты, используемые в качестве `StartDate` и `EndDate`, включены включительно. Например, если значение `StartDate` равно 1 июля 2019 г., то эта дата будет включена в возвращаемую таблицу (если дата существует в столбце `dates`).
- Возвращаемая таблица может содержать только даты, хранящиеся в столбце `Dates`. Например, если столбец `Dates` начинается с 1 июля 2017 года, а значение `StartDate` — 1 июля 2016 года, возвращаемая таблица начнется с 1 июля 2017 года.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем **определении таблицы sales** используется функция DATESBETWEEN для создания вычисления *актуального* (LTD). Срок жизни представляет собой накопление меры с течением времени с самого начала времени.

Обратите внимание, что формула использует функцию MAX. Эта функция возвращает последнюю дату, которая находится в контексте фильтра. Таким образом, функция DATESBETWEEN возвращает таблицу дат, начиная с самой ранней даты до последней даты.

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

DAX

```
Customers LTD =
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerKey]),
    DATESBETWEEN(
        'Date'[Date],
        BLANK(),
        MAX('Date'[Date])
    )
)
```

Рассмотрим, что самая ранняя дата, хранящуюся в таблице даты 1 июля 2017 года. Таким образом, когда отчет фильтрует меру на месяц 2020 года, функция DATESBETWEEN возвращает диапазон дат с 1 июля 2017 г. до 30 июня 2020 г.

Связанное содержимое

- функции аналитики времени (DAX)
- функции даты и времени (DAX)
- функция DATESINPERIOD (DAX)

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DATESINPERIOD

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец дат, начинающихся с указанной даты начала, и продолжается для указанного числа и типа интервалов даты.

Эта функция подходит для передачи в качестве фильтра в функцию CALCULATE. Используйте его для фильтрации выражения по стандартным интервалам даты, таким как дни, месяцы, кварталы или годы.

Синтаксис

DAX

```
DATESINPERIOD(<dates>, <start_date>, <number_of_intervals>, <interval>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец даты.
`start_date`	Выражение даты.
`number_of_intervals`	Целое число, указывающее количество интервалов для добавления или вычитания из дат.
`interval`	Интервал, с помощью которого необходимо переместить даты. Значение интервала может быть одним из следующих: `DAY`, `MONTH`, `QUARTER` и `YEAR`

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- В наиболее распространенном случае использования `dates` является ссылкой на столбец даты помеченной таблицы дат.
- Если число, указанное для `number_of_intervals` положительно, даты перемещаются вперед в течение времени; Если число отрицательное, даты сдвигаются назад во времени.
- Параметр `interval` является перечислением. Допустимые значения: `DAY`, `MONTH`, `QUARTER` и `YEAR`. Так как это перечисление, значения не передаются в виде строк. Поэтому не заключайте их в кавычки.
- Возвращаемая таблица может содержать только даты, хранящиеся в столбце `dates`. Например, если столбец `dates` начинается с 1 июля 2017 года, а значение `start_date` — 1 июля 2016 года, возвращаемая таблица начнется с 1 июля 2017 года.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже **определении меры продаж** таблица использует функцию DATESINPERIOD для вычисления выручки за предыдущий год (PY).

Обратите внимание, что формула использует функцию MAX. Эта функция возвращает последнюю дату, которая находится в контексте фильтра. Таким образом, функция DATESINPERIOD возвращает таблицу дат, начиная с последней даты в прошлом году.

Примеры в этой статье можно использовать с примером модели Adventure Works DW 2020 Power BI Desktop. Сведения о получении модели см. в разделе DAX примера модели.

```
DAX
```

```
Revenue PY =
CALCULATE(
    SUM(Sales[Sales Amount]),
    DATESINPERIOD(
        'Date'[Date],
        MAX('Date'[Date]),
        -1,
        YEAR
    )
)
```

Рассмотрим, что отчет фильтруется по месяцам июня 2020 года. Функция MAX возвращается 30 июня 2020 г. Функция DATESINPERIOD возвращает диапазон дат от 1 июля 2019 г. до 30 июня 2020 г. Это год значений дат, начиная с 30 июня 2020 года за последний год.

Связанное содержимое

функции аналитики времени (DAX)функции даты и времени (DAX)DATESBETWEEN (DAX)

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DATESMTD

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец дат месяца на дату в текущем контексте.

Синтаксис

```
DAX

DATESMTD(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

Аргумент `dates` может быть одним из следующих:

- Ссылка на столбец даты и времени.
- Табличное выражение, возвращающее один столбец значений даты и времени.

- Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

  > ⓘ **Примечание**
  >
  > Ограничения логических выражений описаны в разделе CALCULATE функции.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже примере формулы создается мера, которая вычисляет "Итог даты месяца" для продаж в Интернете.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
DATESMTD(DateTime[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениDATESYTD функцииDATESQTD функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DATESQTD

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец дат для даты квартала в текущем контексте.

Синтаксис

DAX

```
DATESQTD(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

Аргумент `dates` может быть одним из следующих:

- Ссылка на столбец даты и времени.

- Табличное выражение, возвращающее один столбец значений даты и времени.
- Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

  > ⓘ **Примечание**
  >
  > Ограничения логических выражений описаны в разделе CALCULATE функции.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже примере формулы создается мера, которая вычисляет "ежеквартально выполняющийся итог" интернет-продаж.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
DATESQTD(DateTime[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениDATESYTD функцииDATESMTD функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DATESYTD

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец дат года к дате в текущем контексте.

Синтаксис

DAX

```
DATESYTD(<dates> [,<year_end_date>])
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.
`year_end_date`	(необязательно) Литеральная строка с датой, определяющей дату окончания года. Значение по умолчанию — 31 декабря.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

Аргумент `dates` может быть одним из следующих:

- Ссылка на столбец даты и времени,

- Табличное выражение, возвращающее один столбец значений даты и времени,
- Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

> ⓘ **Примечание**
>
> Ограничения логических выражений описаны в разделе CALCULATE функции.

- Параметр `year_end_date` — строковый литерал даты в том же языковом стандарте, что и языковой стандарт клиента, в котором была создана книга. Часть года даты игнорируется. В зависимости от языкового стандарта формат может быть примерно таким, как "m-dd" или "dd-m".
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет "Выполнение итогов" для продаж в Интернете.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
DATESYTD(DateTime[DateKey]))
```

В приведенном ниже примере формулы создается мера, которая вычисляет "Итог по финансовому году" для продаж в Интернете с помощью языкового стандарта США для формата даты.

DAX

```
= CALCULATE(
    SUM(InternetSales_USD[SalesAmount_USD]),
    DATESYTD(DateTime[DateKey],
        "6-30"
        )
)
```

Связанное содержимое

функции аналитики временифункции даты и времениDATESMTD функцииDATESQTD функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ENDOFMONTH

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает последнюю дату месяца в текущем контексте для указанного столбца дат.

Синтаксис

DAX

```
ENDOFMONTH(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.

    - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, возвращающая конец месяца для текущего контекста.

DAX

```
= ENDOFMONTH(DateTime[DateKey])
```

Связанное содержимое

функции функции даты и временифункции аналитики времени функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ENDOFQUARTER

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает последнюю дату квартала в текущем контексте для указанного столбца дат.

Синтаксис

DAX

```
ENDOFQUARTER(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени,
  - Табличное выражение, возвращающее один столбец значений даты и времени,

    - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, возвращающая конец квартала для текущего контекста.

```
DAX

= ENDOFQUARTER(DateTime[DateKey])
```

Связанное содержимое

функции функции даты и временифункции аналитики времени функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ENDOFYEAR

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает последнюю дату года в текущем контексте для указанного столбца дат.

Синтаксис

DAX

```
ENDOFYEAR(<dates> [,<year_end_date>])
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.
`year_end_date`	(необязательно) Литеральная строка с датой, определяющей дату окончания года. Значение по умолчанию — 31 декабря.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени,

    - Табличное выражение, возвращающее один столбец значений даты и времени,
    - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Параметр `year_end_date` — строковый литерал даты в том же языковом стандарте, что и языковой стандарт клиента, в котором была создана книга. Часть года даты игнорируется.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая возвращает конец финансового года, который заканчивается 30 июня для текущего контекста.

DAX

```
= ENDOFYEAR(DateTime[DateKey],"06/30/2004")
```

Связанное содержимое

функции функции даты и временифункции аналитики времени функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

FIRSTDATE

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает первую дату в текущем контексте для указанного столбца дат.

Синтаксис

```
DAX

FIRSTDATE(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.

    - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Если текущий контекст является одной датой, дата, возвращаемая FIRSTDATE и LASTDATE функциями, будет равна.
- Возвращаемое значение — это таблица, содержащая один столбец и одно значение. Поэтому эту функцию можно использовать в качестве аргумента для любой функции, требующей таблицы в его аргументах. Кроме того, возвращаемое значение можно использовать всякий раз, когда требуется значение даты.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая получает первую дату, когда продажа была выполнена в канале продаж Интернета для текущего контекста.

DAX

```
= FIRSTDATE('InternetSales_USD'[SaleDateKey])
```

Связанное содержимое

функции функции даты и временифункции аналитики времени функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

LASTDATE

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает последнюю дату в текущем контексте для указанного столбца дат.

Синтаксис

```
DAX

LASTDATE(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени,
  - Табличное выражение, возвращающее один столбец значений даты и времени,

    - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Если текущий контекст является одной датой, дата, возвращаемая FIRSTDATE и LASTDATE функциями, будет равна.
- Технически возвращаемое значение — это таблица, содержащая один столбец и одно значение. Поэтому эту функцию можно использовать в качестве аргумента для любой функции, требующей таблицы в его аргументах. Кроме того, возвращаемое значение можно использовать всякий раз, когда требуется значение даты.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая получает последнюю дату для текущего контекста, когда продажа была выполнена в канале продаж в Интернете.

DAX

```
= LASTDATE('InternetSales_USD'[SaleDateKey])
```

Связанное содержимое

функции функции даты и временифункции аналитики времени функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NEXTDAY

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ⚠️ визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец всех дат со следующего дня, на основе первой даты, указанной в столбце `dates` в текущем контексте.

Синтаксис

DAX

```
NEXTDAY(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- Эта функция возвращает все даты от следующего дня до первой даты в входном параметре. Например, если первая дата в аргументе `dates` относится к 10 июня 2009 г.; затем эта функция возвращает все даты, равные 11 июня 2009 года.

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет "продажи в следующий день" интернет-продаж.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
NEXTDAY('DateTime'[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениNEXTQUARTER функцииNEXTMONTH функцииNEXTYEAR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NEXTMONTH

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец всех дат из следующего месяца, на основе первой даты в столбце `dates` в текущем контексте.

Синтаксис

```
DAX

NEXTMONTH(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- Эта функция возвращает все даты от следующего дня до первой даты в входном параметре. Например, если первая дата в аргументе `dates` относится к 10 июня 2009 г.; затем эта функция возвращает все даты в течение месяца 2009 года.

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет "продажи в следующем месяце" для продаж в Интернете.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
NEXTMONTH('DateTime'[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениNEXTDAY функцииNEXTQUARTER функцииNEXTYEAR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NEXTQUARTER

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец всех дат в следующем квартале на основе первой даты, указанной в столбце `dates` в текущем контексте.

Синтаксис

```
DAX

NEXTQUARTER(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- Эта функция возвращает все даты в следующем квартале на основе первой даты в входном параметре. Например, если первая дата в столбце `dates` относится к 10 июня 2009 года, эта функция возвращает все даты для квартала по сентябрь 2009 года.

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет "продажи в следующем квартале" для интернет-продаж.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
NEXTQUARTER('DateTime'[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениNEXTDAY функцииNEXTMONTH функцииNEXTYEAR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

NEXTYEAR

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец всех дат в следующем году, на основе первой даты в столбце `dates` в текущем контексте.

Синтаксис

```DAX
NEXTYEAR(<dates>[,<year_end_date>])
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.
`year_end_date`	(необязательно) Литеральная строка с датой, определяющей дату окончания года. Значение по умолчанию — 31 декабря.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- Эта функция возвращает все даты в следующем году на основе первой даты в входном столбце. Например, если первая дата в столбце `dates` относится к

2007 году, эта функция возвращает все даты 2008 года.

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Параметр `year_end_date` — строковый литерал даты в том же языковом стандарте, что и языковой стандарт клиента, в котором была создана книга. Часть года даты игнорируется.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет "продажи в следующем году" для продаж в Интернете.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
NEXTYEAR('DateTime'[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениNEXTDAY функцииNEXTQUARTER функцииNEXTMONTH

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

OPENINGBALANCEMONTH

Статья • 21.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Вычисляет `expression` по дате, соответствующей концу предыдущего месяца в текущем контексте.

Синтаксис

DAX

```
OPENINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Выражение, возвращающее скалярное значение.
`dates`	Столбец, содержащий даты.
`filter`	(необязательно) Выражение, указывающее фильтр для применения к текущему контексту.

Возвращаемое значение

Скалярное значение, представляющее `expression`, вычисляемое по первой дате месяца в текущем контексте.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Выражение `filter` имеет ограничения, описанные в разделе, CALCULATE функции.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже примере формулы создается мера, которая вычисляет "Начальное значение инвентаризации месяца" инвентаризации продукта.

DAX

```
=
OPENINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*Product
Inventory[UnitsBalance]),DateTime[DateKey])
```

Связанное содержимое

функция OPENINGBALANCEYEAROPENINGBALANCEQUARTER функциифункции аналитики времениCLOSINGBALANCEMONTH

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

OPENINGBALANCEQUARTER

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ⚠️ визуального вычисления

> ⓘ **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Вычисляет `expression` по дате, соответствующей концу предыдущего квартала в текущем контексте.

Синтаксис

DAX

```
OPENINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Выражение, возвращающее скалярное значение.
`dates`	Столбец, содержащий даты.
`filter`	(необязательно) Выражение, указывающее фильтр для применения к текущему контексту.

Возвращаемое значение

Скалярное значение, представляющее `expression`, вычисляемое по первой дате квартала в текущем контексте.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Выражение `filter` имеет ограничения, описанные в разделе, CALCULATE функции.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже примере формулы создается мера, которая вычисляет значение "Четверть начала инвентаризации" инвентаризации продукта.

DAX

```
=
OPENINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*Produ
ctInventory[UnitsBalance]),DateTime[DateKey])
```

Связанное содержимое

функция OPENINGBALANCEYEAROPENINGBALANCEMONTH функцииФункции аналитики времениCLOSINGBALANCEQUARTER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

OPENINGBALANCEYEAR

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Оценивает `expression` по дате, соответствующей концу предыдущего года в текущем контексте.

Синтаксис

DAX

```
OPENINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Выражение, возвращающее скалярное значение.
`dates`	Столбец, содержащий даты.
`filter`	(необязательно) Выражение, указывающее фильтр для применения к текущему контексту.
`year_end_date`	(необязательно) Литеральная строка с датой, определяющей дату окончания года. Значение по умолчанию — 31 декабря.

Возвращаемое значение

Скалярное значение, представляющее `expression`, вычисляемое по первой дате года в текущем контексте.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Выражение `filter` имеет ограничения, описанные в разделе, CALCULATE функции.
- Параметр `year_end_date` — строковый литерал даты в том же языковом стандарте, что и языковой стандарт клиента, в котором была создана книга. Часть года даты игнорируется.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже примере формулы создается мера, которая вычисляет значение "Дата начала инвентаризации" инвентаризации продукта.

DAX

```
=
OPENINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductI
nventory[UnitsBalance]),DateTime[DateKey])
```

Связанное содержимое

функция OPENINGBALANCEQUARTEROPENINGBALANCEMONTH функциифункции аналитики времениCLOSINGBALANCEYEAR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PARALLELPERIOD

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ⚠️ визуального вычисления

> ⓘ **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец дат, представляющий период параллельно дат в указанном столбце `dates` столбца в текущем контексте, при этом даты сдвигали число интервалов в течение времени или назад.

Синтаксис

DAX

```
PARALLELPERIOD(<dates>,<number_of_intervals>,<interval>)
```

Параметры

⛶ Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.
`number_of_intervals`	Целое число, указывающее количество интервалов для добавления или вычитания из дат.
`interval`	Интервал, с помощью которого необходимо переместить даты. Значение интервала может быть одним из следующих: `year`, `quarter`, `month`.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- Эта функция принимает текущий набор дат в столбце, заданном `dates`, сдвигает первую дату и последнюю дату указанного числа интервалов, а затем возвращает все смежные даты между двумя смещенными датами. Если интервал является частичным диапазоном месяца, квартала или года, то любые частичные месяцы в результатах также заполняются для завершения всего интервала.

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени,
  - Табличное выражение, возвращающее один столбец значений даты и времени,
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE функции.

- Если число, указанное для `number_of_intervals` положительно, даты в `dates` перемещаются вперед в течение времени; Если число отрицательное, даты в `dates` сдвигаются назад.

- Параметр `interval` — это перечисление, а не набор строк; поэтому значения не должны быть заключены в кавычки. Кроме того, значения: `year`, `quarter`, `month` должны быть полны при использовании.

- Таблица результатов содержит только даты, которые отображаются в значениях базового столбца таблицы.

- Функция PARALLELPERIOD аналогична функции DATEADD, за исключением того, что PARALLELPERIOD всегда возвращает полные периоды на заданном уровне детализации вместо частичных периодов, которые DATEADD возвращаются. Например, если у вас есть выбор дат, начинающихся с 10 июня, и заканчивается 21 июня того же года, и вы хотите переместить этот выбор вперед на один месяц, функция PARALLELPERIOD вернет все даты с следующего месяца (1 июля по 31 июля); однако если вместо этого используется DATEADD, результат будет включать только даты с 10 июля по 21 июля.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет продажи за предыдущий год для продаж через Интернет.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
PARALLELPERIOD(DateTime[DateKey],-1,year))
```

Связанное содержимое

Функции аналитики времени функции даты и времениDATEADD функции

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PREVIOUSDAY

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец всех дат, представляющих день, предшествующий первой дате в столбце `dates` в текущем контексте.

Синтаксис

DAX

```
PREVIOUSDAY(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- Эта функция определяет первую дату в входном параметре, а затем возвращает все даты, соответствующие дню предыдущей первой даты. Например, если первая дата в аргументе `dates` относится к 10 июня 2009 г.; эта функция возвращает все даты, равные 9 июня 2009 года.

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE функции.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет "продажи за предыдущий день" для продаж в Интернете.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
PREVIOUSDAY('DateTime'[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениPREVIOUSMONTH функцииPREVIOUSQUARTER функцииPREVIOUSYEAR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PREVIOUSMONTH

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец всех дат из предыдущего месяца, на основе первой даты в столбце `<Dates>` в текущем контексте.

Синтаксис

```
DAX

PREVIOUSMONTH(<Dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`Dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- Эта функция возвращает все даты из предыдущего месяца, используя первую дату в столбце, используемом в качестве входных данных. Например, если первая дата в аргументе `Dates` относится к 10 июня 2009 года, эта функция возвращает все даты в течение месяца 2009 года.

- Аргумент `Dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет "продажи за предыдущий месяц" для продаж в Интернете.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
PREVIOUSMONTH('DateTime'[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениPREVIOUSDAYPREVIOUSQUARTERPREVIOUSYEAR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PREVIOUSQUARTER

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец всех дат предыдущего квартала на основе первой даты в столбце `dates` в текущем контексте.

Синтаксис

DAX

```
PREVIOUSQUARTER(<dates>)
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- Эта функция возвращает все даты из предыдущего квартала, используя первую дату в входном столбце. Например, если первая дата в аргументе `dates` относится к 10 июня 2009 года, эта функция возвращает все даты для квартала по март 2009 года.

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет "предыдущий квартал продаж" для продаж в Интернете.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
PREVIOUSQUARTER('DateTime'[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениPREVIOUSMONTHPREVIOUSDAYPREVIOUSYEAR

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

PREVIOUSYEAR

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях, так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец всех дат предыдущего года, учитывая последнюю дату в столбце `dates` в текущем контексте.

Синтаксис

DAX

```
PREVIOUSYEAR(<dates>[,<year_end_date>])
```

Параметры

Развернуть таблицу

Срок	Определение
`dates`	Столбец, содержащий даты.
`year_end_date`	(необязательно) Литеральная строка с датой, определяющей дату окончания года. Значение по умолчанию — 31 декабря.

Возвращаемое значение

Таблица, содержащая один столбец значений даты.

Замечания

- Эта функция возвращает все даты из предыдущего года, учитывая последнюю дату в входном параметре. Например, если последняя дата в аргументе `dates`

относится к 2009 году, эта функция возвращает все даты в течение 2008 года до указанного `year_end_date`.

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE.
- Параметр `year_end_date` — строковый литерал даты в том же языковом стандарте, что и языковой стандарт клиента, в котором была создана книга. Часть года даты игнорируется.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет продажи за предыдущий год для продаж через Интернет.

DAX

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),
PREVIOUSYEAR('DateTime'[DateKey]))
```

Связанное содержимое

функции аналитики временифункции даты и времениPREVIOUSMONTHPREVIOUSDAYPREVIOUSQUARTER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

SAMEPERIODLASTYEAR

Статья • 24.03.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ⚠️ визуального вычисления

> ⓘ **Примечание**
>
> Эта функция не рекомендуется использовать в **визуальных вычислениях** , так как, скорее всего, возвращает бессмысленные результаты.

Возвращает таблицу, содержащую столбец дат, сменяемых в год назад с дат в указанном столбце `dates` в текущем контексте.

Синтаксис

DAX

```
SAMEPERIODLASTYEAR(<dates>)
```

Параметры

Развернуть таблицу

Термин	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица значений дат с одним столбцом.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени,
  - Табличное выражение, возвращающее один столбец значений даты и времени,

    - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE.
- Возвращаемые даты совпадают с датами, возвращаемыми этой эквивалентной формулой: `DATEADD(dates, -1, year)`
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая вычисляет продажи торговых посредников за предыдущий год.

DAX

```
= CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),
SAMEPERIODLASTYEAR(DateTime[DateKey]))
```

Специальное поведение

Если выбор включает последние два дня месяца, SAMEPERIODLASTYEAR будет использовать семантику расширения и будет включать дни до конца месяца. Например, когда 27 и 28 февраля 2009 года включены в выборку, SAMEPERIODLASTYEAR возвращается 27 февраля до 29 февраля 2008 года.

Это поведение происходит только в том случае, если последние два дня месяца включены в выборку. Если выбрано только 27 февраля, он перейдет к 27 февраля.

DAX

```
= SAMEPERIODLASTYEAR(DateTime[DateKey])
```

Связанное содержимое

функции аналитики временифункции даты и времени
PREVIOUSYEARPARALLELPERIOD

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

STARTOFMONTH

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях , так как, скорее всего, возвращает бессмысленные результаты.

Возвращает первую дату месяца в текущем контексте для указанного столбца дат.

Синтаксис

```
DAX

STARTOFMONTH(<dates>)
```

Параметры

Развернуть таблицу

Термин	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.

- Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая возвращает начало месяца для текущего контекста.

DAX

```
= STARTOFMONTH(DateTime[DateKey])
```

Связанное содержимое

функции даты и временифункции аналитики времениSTARTOFYEARSTARTOFQUARTER

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

STARTOFQUARTER

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях , так как, скорее всего, возвращает бессмысленные результаты.

Возвращает первую дату квартала в текущем контексте для указанного столбца дат.

Синтаксис

DAX

```
STARTOFQUARTER(<dates>)
```

Параметры

Развернуть таблицу

Термин	Определение
`dates`	Столбец, содержащий даты.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.

    - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая возвращает начало квартала для текущего контекста.

```
DAX
= STARTOFQUARTER(DateTime[DateKey])
```

Связанное содержимое

функции даты и временифункции аналитики времениSTARTOFYEARSTARTOFMONTH

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

STARTOFYEAR

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях , так как, скорее всего, возвращает бессмысленные результаты.

Возвращает первую дату года в текущем контексте для указанного столбца дат.

Синтаксис

```DAX
STARTOFYEAR(<dates>)
```

Параметры

Развернуть таблицу

Термин	Определение
`dates`	Столбец, содержащий даты.
`YearEndDate`	(Необязательно) Значение даты окончания года.

Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.

    - Табличное выражение, возвращающее один столбец значений даты и времени.
    - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.

- Ограничения логических выражений описаны в разделе CALCULATE.

- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В следующем примере формулы создается мера, которая возвращает начало года для текущего контекста.

DAX

```
= STARTOFYEAR(DateTime[DateKey])
```

Связанное содержимое

функции даты и временифункции аналитики времениSTARTOFQUARTERSTARTOFMONTH

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TOTALMTD

Статья • 23.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях , так как, скорее всего, возвращает бессмысленные результаты.

Вычисляет значение `expression` для текущего месяца в текущем контексте.

Синтаксис

DAX

```
TOTALMTD(<expression>,<dates>[,<filter>])
```

Параметры

Развернуть таблицу

Параметр	Определение
`expression`	Выражение, возвращающее скалярное значение.
`dates`	Столбец, содержащий даты.
`filter`	(необязательно) Выражение, указывающее фильтр для применения к текущему контексту.

Возвращаемое значение

Скалярное значение, представляющее `expression`, вычисляемое для дат текущего месяца, учитывая даты в `dates`.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE.
- Выражение `filter` имеет ограничения, описанные в разделе, CALCULATE.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже примере формулы создается мера, которая вычисляет сумму "месяц выполнения" или "сумма выполнения месяца" для продаж в Интернете.

DAX

```
= TOTALMTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

Связанное содержимое

ALL CALCULATE TOTALYTD TOTALQTD

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TOTALQTD

Статья • 22.01.2025

применяется: вычисляемый столбец вычисляемой таблицы измерение визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях , так как, скорее всего, возвращает бессмысленные результаты.

Вычисляет значение `expression` для дат в квартале до даты в текущем контексте.

Синтаксис

DAX

```
TOTALQTD(<expression>,<dates>[,<filter>])
```

Параметры

Развернуть таблицу

Параметр	Определение
`expression`	Выражение, возвращающее скалярное значение.
`dates`	Столбец, содержащий даты.
`filter`	(необязательно) Выражение, указывающее фильтр для применения к текущему контексту.

Возвращаемое значение

Скалярное значение, представляющее `expression`, вычисляемое для всех дат в текущем квартале до даты, учитывая даты в `dates`.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE.
- Выражение `filter` имеет ограничения, описанные в разделе, CALCULATE.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже примере формулы создается мера, которая вычисляет "четверть запущенного итога" или "четверть беговой суммы" для продаж в Интернете.

DAX

```
= TOTALQTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

Связанное содержимое

ALL CALCULATE TOTALYTD TOTALMTD

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

TOTALYTD

Статья • 22.01.2025

применяется: ✅ вычисляемый столбец ✅ вычисляемой таблицы ✅ измерение ⚠️ визуального вычисления

> **Примечание**
>
> Эта функция не рекомендуется использовать в визуальных вычислениях , так как, скорее всего, возвращает бессмысленные результаты.

Оценивает текущее значение `expression` в текущем контексте.

Синтаксис

DAX

```
TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])
```

Параметры

Развернуть таблицу

Параметр	Определение
`expression`	Выражение, возвращающее скалярное значение.
`dates`	Столбец, содержащий даты.
`filter`	(необязательно) Выражение, указывающее фильтр для применения к текущему контексту.
`year_end_date`	(необязательно) Литеральная строка с датой, определяющей дату окончания года. Значение по умолчанию — 31 декабря.

Возвращаемое значение

Скалярное значение, представляющее `expression`, оцененное для текущего года `dates`.

Замечания

- Аргумент `dates` может быть одним из следующих:
  - Ссылка на столбец даты и времени.
  - Табличное выражение, возвращающее один столбец значений даты и времени.
  - Логическое выражение, определяющее таблицу значений даты и времени с одним столбцом.
- Ограничения логических выражений описаны в разделе CALCULATE.
- Выражение `filter` имеет ограничения, описанные в разделе, CALCULATE.
- Параметр `year_end_date` — строковый литерал даты в том же языковом стандарте, что и языковой стандарт клиента, в котором была создана книга. Часть года даты не требуется и игнорируется. Например, следующая формула указывает (фискальный) year_end_date 6/30 в книге языкового стандарта EN-US.

  DAX

  ```
  = TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey],
  ALL('DateTime'), "6/30")
  ```

  В этом примере year_end_date можно указать как "6/30", "30 июня", "30 июня" или любую строку, которая разрешается в месяц или день. Однако рекомендуется указать year_end_date с помощью "month/day" (как показано), чтобы убедиться, что строка разрешается на дату.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

Пример

В приведенном ниже примере формулы создается мера, которая вычисляет "год выполнения суммы" или "год выполнения суммы" для продаж в Интернете.

DAX

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

Связанное содержимое

ALL CALCULATE DATESYTD TOTALMTD TOTALQTD

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

Statements

Статья • 22.01.2025

В этой категории

Развернуть таблицу

Утверждение	Описание
DEFINE	(ключевое слово) В этой статье представлено одно или несколько определений сущностей, которые можно применить к одному или нескольким операторам EVALUATE.
EVALUATE	(ключевое слово) Представляет инструкцию, содержащую табличное выражение, необходимое для выполнения запроса DAX.
MEASURE	(ключевое слово) Представляет определение меры, которое можно использовать в одном или нескольких операторах EVALUATE в запросе.
ORDER BY	(ключевое слово) Представляет инструкцию, которая определяет порядок сортировки результатов запроса, возвращаемых оператором EVALUATE.
START AT	(ключевое слово) Представляет инструкцию, которая определяет начальное значение, в котором возвращаются результаты запроса оператора ORDER BY.
VAR	(ключевое слово) Сохраняет результат выражения в виде именованной переменной, которая затем может быть передана в качестве аргумента другим выражениям мер.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

DEFINE

Статья • 23.01.2025

Представляет инструкцию с одним или несколькими определениями сущностей, которые можно применить к одному или нескольким операторам запроса.

Синтаксис

DAX

```
[DEFINE
    (
     (MEASURE <table name>[<measure name>] = <scalar expression>) |
     (VAR <var name> = <table or scalar expression>) |
     (TABLE <table name> = <virtual table definition>) |
     (COLUMN <table name>[<column name>] = <scalar expression>) |
    ) +
]

(EVALUATE <table expression>) +
```

Параметры

Развернуть таблицу

Срок	Определение
`Entity`	MEASURE, VAR, TABLE[1] или COLUMN[1].
`name`	Имя определения меры, var, таблицы или столбца. Это не может быть выражением. Имя не должно быть уникальным. Имя существует только для длительности запроса.
`expression`	Любое DAX выражение, возвращающее табличное или скалярное значение. Выражение может использовать любую из определенных сущностей. Если требуется преобразовать скалярное выражение в табличное выражение, обтекайте выражение внутри конструктора таблицы с фигурными скобками `{}` или используйте функцию `ROW()` для возврата одной таблицы строк.

[1]Осторожность: определения таблицы и столбца, области запросов предназначены только для внутреннего использования. Хотя вы можете определить выражения TABLE и COLUMN для запроса без синтаксической ошибки, они могут создавать ошибки среды выполнения и не рекомендуется.

Замечания

- Запрос DAX может содержать несколько операторов EVALUATE, но может иметь только одну инструкцию DEFINE. Определения в инструкции DEFINE могут применяться к любым EVALUATE операторам в запросе.
- В инструкции DEFINE требуется по крайней мере одно определение.
- Определения мер для мер модели переопределения запроса с тем же именем.
- VAR имена имеют уникальные ограничения. Дополнительные сведения см. в разделе VAR — параметры.
- Дополнительные сведения об использовании инструкции DEFINE см. в DAX запросах.
- Дополнительные сведения о виртуальном столбце см. в виртуального столбца
- Дополнительные сведения о виртуальной таблице см. в

Связанное содержимое

EVALUATE
VAR
MEASURE
виртуальных таблицDAX запросов

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

EVALUATE

Статья • 21.01.2025

Представляет инструкцию, содержащую табличное выражение, необходимое в запросе DAX.

Синтаксис

```DAX
EVALUATE <table>
```

Параметры

Развернуть таблицу

Срок	Определение
`table`	Табличное выражение

Возвращаемое значение

Результат табличного выражения.

Замечания

- Запрос DAX может содержать несколько операторов EVALUATE.
- Дополнительные сведения об использовании инструкций EVALUATE см. в DAX запросах.

Пример

```DAX
EVALUATE
    'Internet Sales'
```

Возвращает все строки и столбцы из таблицы "Продажи в Интернете" в виде таблицы.

Связанное содержимое

запросов

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

MEASURE

Статья • 22.01.2025

Представляет определение меры в инструкции DEFINE запроса DAX.

Синтаксис

DAX

```
[DEFINE
    (
      MEASURE <table name>[<measure name>] = <scalar expression>
    ) +
]

(EVALUATE <table expression>) +
```

Параметры

Развернуть таблицу

Срок	Определение
`table name`	Имя таблицы, содержащей меру.
`measure name`	Имя меры. Это не может быть выражением. Имя не должно быть уникальным. Имя существует только для длительности запроса.
`scalar expression`	Выражение DAX, возвращающее скалярное значение.

Возвращаемое значение

Вычисляемый результат выражения меры.

Замечания

- Определения мер для мер модели переопределения запроса с одинаковым именем в течение длительности запроса. Они не повлияют на меру модели.
- Выражение меры можно использовать с любым другим выражением в том же запросе.

- Дополнительные сведения об использовании инструкций MEASURE см. в DAX запросах.

Связанное содержимое

DEFINE EVALUATE VAR DAX запросов

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

ORDER BY

Статья • 23.01.2025

Представляет инструкцию, которая определяет порядок сортировки результатов запроса, возвращаемого инструкцией EVALUATE в DAX запроса.

Синтаксис

DAX

```
[ORDER BY {<expression> [{ASC | DESC}]}[, …]]
```

Параметры

Развернуть таблицу

Срок	Определение
`expression`	Любое выражение DAX, возвращающее одно скалярное значение.
`ASC`	(по умолчанию) Порядок сортировки по возрастанию.
`DESC`	Порядок сортировки по убыванию.

Возвращаемое значение

Результат инструкции EVALUATE в порядке возрастания (ASC) или убывания (DESC) .

Замечания

Дополнительные сведения об использовании инструкций ORDER BY см. в DAX запросах.

Связанное содержимое

запросов

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

START AT

Статья • 21.01.2025

Представляет инструкцию, которая определяет начальное значение, в котором возвращаются результаты запроса предложения ORDER BY в инструкции EVALUATE в DAX запроса.

Синтаксис

DAX

```
[START AT {<value>|<parameter>} [, …]]
```

Параметры

Развернуть таблицу

Срок	Определение
`value`	Константное значение. Не может быть выражением.
`parameter`	Имя параметра в инструкции XMLA с префиксом `@` символа.

Замечания

- START AT аргументы имеют соответствие "один к одному" столбцам в инструкции ORDER BY. В инструкции START AT может быть столько аргументов, сколько в инструкции ORDER BY, но не больше. Первый аргумент в инструкции START AT определяет начальное значение столбца 1 столбца ORDER BY. Второй аргумент в инструкции START AT определяет начальное значение столбца 2 столбца ORDER BY в строках, которые соответствуют первому значению столбца 1.
- Дополнительные сведения об использовании инструкций START AT см. в DAX запросах.

Связанное содержимое

запросов

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

VAR

Статья • 22.01.2025

Сохраняет результат выражения в виде именованной переменной, которая затем может быть передана в качестве аргумента другим выражениям мер. После вычисления результирующих значений для выражения переменной эти значения не изменяются, даже если переменная ссылается в другом выражении.

Синтаксис

DAX

```
VAR <name> = <expression>
```

Параметры

Развернуть таблицу

Срок	Определение
`name`	Имя переменной (идентификатор). Разделители не поддерживаются. Например, "varName" или [varName] приведет к ошибке. Поддерживаемый набор символов: a-z, A-Z, 0-9. Значение 0-9 недопустимо в качестве первого символа. __ (двойной подчеркивание) допускается в качестве префикса имени идентификатора. Другие специальные символы не поддерживаются. Зарезервированные ключевые слова не разрешены. Имена существующих таблиц не допускаются. Пустые пробелы не допускаются.
`expression`	Выражение DAX, которое возвращает скалярное или табличное значение.

Возвращаемое значение

Именованной переменной, содержащей результат аргумента выражения.

Замечания

- Выражение, переданное в качестве аргумента для VAR, может содержать другое объявление VAR.

- При ссылке на переменную:
  - Меры не могут ссылаться на переменные, определенные вне выражения меры, но могут ссылаться на переменные функциональной области, определенные в выражении.
  - Переменные могут ссылаться на меры.
  - Переменные могут ссылаться на ранее определенные переменные.
  - Столбцы в табличных переменных нельзя ссылаться с помощью синтаксиса TableName[ColumnName].

- Рекомендации по использованию VARсм. в статье Использование переменных для улучшения DAX формул.

- Дополнительные сведения о том, как используется в запросе, см. в запросов.

Пример

Чтобы вычислить процент роста по сравнению с годом без использования переменной, можно создать три отдельных меры. Первая мера вычисляет сумму продаж:

DAX

```
Sum of SalesAmount = SUM(SalesTable[SalesAmount])
```

Вторая мера вычисляет сумму продаж за предыдущий год:

DAX

```
SalesAmount PreviousYear =
    CALCULATE([Sum of SalesAmount],
    SAMEPERIODLASTYEAR(Calendar[Date])
    )
```

Затем можно создать третью меру, которая объединяет другие две меры, чтобы вычислить процент роста. Обратите внимание, что мера Sum of SalesAmount используется в двух местах; сначала, чтобы определить, есть ли продажа, снова вычислить процент.

DAX

```
Sum of SalesAmount YoY%: =
    IF([Sum of SalesAmount] ,
        DIVIDE(([Sum of SalesAmount] - [SalesAmount PreviousYear]), [Sum of
SalesAmount])
    )
```

С помощью переменной можно создать одну меру, которая вычисляет тот же результат:

DAX

```
YoY% =
  VAR Sales =
      SUM(SalesTable[SalesAmount])
  VAR SalesLastYear =
      CALCULATE ( SUM ( SalesTable[SalesAmount] ), SAMEPERIODLASTYEAR (
'Calendar'[Date] ) )

  return if(Sales, DIVIDE(Sales - SalesLastYear, Sales))
```

Используя переменную, вы можете получить тот же результат, но более читаемым способом. И поскольку результат выражения хранится в переменной, производительность меры может быть значительно улучшена, так как она не должна пересчитываться при каждом использовании.

Связанное содержимое

Использование переменных для улучшения формул DAXDAX запросов

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

глоссарий DAX

Статья • 10.04.2025

Аналитический запрос

Визуальные элементы Power BI запрашивают модель данных с помощью *аналитического запроса*. Аналитический запрос стремится уменьшить потенциально большие объемы данных и сложности модели с помощью трех отдельных этапов: фильтрация, группа и сводка. Аналитический запрос создается автоматически при назначении полей визуальным элементам отчета. Авторы отчетов могут управлять поведением назначений полей путем переименования полей, изменения метода суммирования или отключения суммирования для достижения группирования. Во время разработки отчета фильтры можно добавлять в отчет, страницу отчета или визуальный элемент. В режиме чтения фильтры можно изменить в области **Фильтры** или с помощью взаимодействия с срезами и другими визуальными элементами, применяя перекрестную фильтрацию.

BLANK

DAX означает отсутствие значения как BLANK. Это эквивалент SQL NULL, но он не ведет себя точно так же. Он более тесно связано с Excel и тем, как он определяет пустую ячейку. BLANK вычисляется как нулевая или пустая строка при сочетании с другими операциями. Например, BLANK + 20 = 20. Всегда используйте прописные буквы; множественное число — BLANKs с маленькой буквой 's'.

Вычисляемый столбец

Вычисление модели, используемое для добавления столбца в табличную модель, путем написания формулы DAX. Формула должна возвращать скалярное значение и вычисляется для каждой строки в таблице. Вычисляемый столбец можно добавить в таблицу режима импорта или режима хранения DirectQuery.

Вычисляемая мера

В табличном моделировании не существует такого понятия, как вычисляемая мера . Используйте меру *вместо*. Слово *вычисленное* используется для описания вычисленных таблиц и столбцов. Он отличает их от таблиц и столбцов, поступающих из Power Query. Power Query не имеет концепции меры.

Вычисляемая таблица

Вычисление модели, используемое для добавления таблицы в табличную модель, путем написания формулы DAX. Формула должна возвращать объект таблицы. В результате таблица использует режим хранения импорта.

Вычисление

Преднамеренный процесс, который преобразует один или несколько входных данных в один или несколько результатов. В табличной модели данных вычисление может быть объектом модели; вычисляемая таблица, вычисляемый столбец или мера.

Контекст

Описывает среду, в которой вычисляется формула DAX. Существует два типа контекста: *контекст строк* и *контекст фильтра*. Контекст строки представляет "текущую строку" и используется для вычисления формул и выражений вычисляемых столбцов, используемых итераторами таблиц. Контекст фильтра используется для оценки мер, и он представляет фильтры, применяемые непосредственно к столбцам модели и фильтрам, распространяемым связями модели.

DAX

Язык выражений анализа данных (DAX) — это язык формул для Power Pivot в Excel, Power BI, Azure Analysis Services и табличного моделирования в службах SQL Server Analysis Services. Можно также использовать DAX для добавления вычислений модели данных и определения правил безопасности на уровне строк (RLS).

Динамическая безопасность

Если правила безопасности на уровне строк (RLS) применяются с использованием учетной записи пользователя отчета. Правила фильтруют таблицы модели с использованием имени учетной записи пользователя, что может быть сделано с помощью функций USERNAME или USERPRINCIPALNAME. См. безопасность на уровне строк.

Выражение

Единица логики DAX , которая вычисляется и возвращает результат. Выражения могут объявлять переменные, в этом случае им назначается вложенное выражение, и они должны включать инструкцию RETURN, которая выводит окончательное выражение. Выражения создаются с помощью объектов модели (таблиц, столбцов или мер), функций, операторов или констант.

Поле

Ресурс модели данных, представленный в области полей . Поля используются для настройки фильтров отчетов и визуальных элементов. Поля состоят из столбцов модели, уровней иерархии и мер.

Формула

Одно или несколько DAX выражений, используемых для определения вычисления модели. Внутренние выражения называются вложенными выражениями. Plural — это *формулы*.

Функция

DAX функции имеют аргументы, позволяющие передавать параметры. Формулы могут использовать множество вызовов функций, возможно вложение функций в другие функции. В формуле имена функций должны следовать скобкам. Параметры передаются внутри скобок.

Неявная мера

Автоматическое вычисление, достигнутое путем настройки визуального элемента Power BI для суммирования значений столбцов. **числовые** столбцы поддерживают наибольшее разнообразие методов суммирования, включая: сумма, среднее, минимум, максимум, число (уникальные значения), количество, стандартное отклонение, дисперсия или медиана. Столбцы других типов данных также можно суммировать. **текстовые** столбцы можно суммировать с помощью: First (в алфавитном порядке), Last (в алфавитном порядке), Count (Distinct) или Count. **столбцы даты** можно обобщить, используя следующие методы: "Ранний", "Последний", "Подсчет (уникальных значений)" или "Подсчет". **логические столбцы** можно суммировать с помощью: Count (Distinct) или Count.

Функция итератора

Функция DAX , которая перечисляет все строки заданной таблицы и вычисляет заданное выражение для каждой строки. Она обеспечивает гибкость и контроль над тем, как вычисления модели суммируют данные.

MDX

Язык многомерных выражений (MDX) — это формульный язык для многомерных моделей служб SQL Server Analysis Services (также известных как *кубы*). MDX можно использовать для запроса табличных моделей, однако оно не может определять неявные меры. Он может запрашивать только меры, которые уже определены в модели.

Мера

Вычисление, приводящее к результатам суммирования. Меры — это *неявные* или *явные*. Явная мера — это вычисление, добавленное в табличную модель данных путем написания DAX формулы. Формула меры должна возвращать скалярное значение. В панели **Поля** явные показатели обозначены значком калькулятора. Явные меры требуются, если модель запрашивается с помощью многомерных выражений (MDX), как в случае использования функции "Анализ в Excel". Явная мера обычно называется мерой.

Группа мер

Таблица моделей, содержащая по крайней мере одну меру, и не имеет иерархий или видимых столбцов. В области **полей** каждая группа мер помечена значком многокалькулятора. Группы мер перечислены в верхней части панели **полей** и отсортированы в алфавитном порядке по имени.

Вычисление модели

Именованной формулы, которая используется для добавления вычисляемой таблицы, вычисляемого столбца или измерения в табличную модель данных. Его структура `NAME` = `FORMULA`. Большинство вычислений добавляются модельировщиками данных в Power BI Desktop, но меры также можно добавить в отчет о динамическом подключении. См. меры доклада .

Быстрые меры

Функция в Power BI Desktop, которая устраняет необходимость писать DAX формулы для часто используемых мер. Быстрые меры включают среднее значение для каждой категории, рейтинг и изменение относительно базового значения.

Меры по отчету

Также называется мерами уровня отчета . Они добавляются в отчет о активном подключении в Power BI Desktop путем написания формулы DAX, но только для подключений к моделям Power BI или табличным моделям Служб Analysis Services.

Безопасность на уровне строк

Также известен как `RLS`. Метод проектирования для ограничения доступа к подмножествам данных для конкретных пользователей. В табличной модели она достигается путем создания ролей модели. Роли имеют правила, которые являются DAX выражениями для фильтрации строк таблицы.

Скалярный

В DAX скаляре содержится одно значение. Скаляр может иметь любой тип данных: Decimal, Integer, DateTime, String, Currency, Boolean. Скалярное значение может быть результатом выражения, вычисляемого из нескольких значений. Например, функция агрегирования, например MAX(), возвращает одно максимальное значение из набора значений, из которого требуется оценить.

Суммаризация

Операция, применяемая к значениям столбца. См. измерение.

Аналитика времени

Анализ временных данных относится к вычислениям во времени, например, с начала года (YTD).

Функция аналитики времени

DAX включает многие функции временной аналитики. Каждый раз функция аналитики достигает своего результата путем изменения контекста фильтра для фильтров дат.

Примеры функций: TOTALYTD и SAMEPERIODLASTYEAR.

Значение, значения

Данные для визуализации.

Параметр What-if

Функция Power BI Desktop, которая позволяет принимать пользовательский ввод через фильтры. Каждый параметр создает вычисляемую таблицу с одним столбцом и меру, которая возвращает одно выбранное значение. Меру можно использовать в вычислениях модели для реагирования на входные данные пользователя.

операторы DAX

Статья • 22.04.2025

Язык выражения анализа данных используетDAX операторы для создания выражений, которые сравнивают значения, выполняют арифметические вычисления или работают со строками.

Типы операторов

Существует четыре различных типа операторов вычисления: арифметическое, сравнение, объединение текста и логические.

Арифметические операторы

Чтобы выполнять основные математические операции, такие как сложение, вычитание или умножение, объединяйте числа и получайте числовые результаты, используйте следующие арифметические операторы.

Развернуть таблицу

Арифметический оператор	Значение	Пример
`+` (знак плюса)	Сложение	`3+3`
`-` (знак минуса)	Вычитание или знак	`3-1-1`
`*` (звездочка)	Умножение	`3*3`
`/` (косая черта вперед)	Деление	`3/3`
`^` (карет)	Экспоненциация	`16^4`

> **Примечание**
>
> Знак "плюс" может функционировать как как *двоичный оператор*, так и как *унарный оператор*. Двоичный оператор требует наличия чисел по обе стороны оператора и выполняет сложение. При использовании значений в DAX формуле на обеих сторонах двоичного оператора пытается привести значения к числовым типам данных, DAX если они еще не являются числами. Напротив, унарный оператор может применяться к любому типу аргумента. Символ плюса не влияет на тип или значение и просто игнорируется, в то время как оператор минус создает отрицательное значение, если применяется к числового значения.

Операторы сравнения

Можно сравнить два значения со следующими операторами. При сравнении двух значений с помощью этих операторов результат является логическим значением либо `TRUE`, либо `FALSE`.

Развернуть таблицу

Оператор сравнения	Значение	Пример
`=`	Равно	[Регион] = "США"
`==`	Строгое равенство	[Регион] == "США"
`>`	Больше чем	[Дата продаж] > "Январь 2009"
`<`	Менее	[Дата продаж] < "1 января 2009 года"
`>=`	Больше или равно	[Сумма] >= 20000
`<=`	Меньше или равно	[Сумма] <= 100
`<>`	Не равно	[Регион] <> "США"

Все операторы сравнения, кроме == обрабатываются BLANK как равные числу 0, пустой строке "", (1899, DATE12, 30) или `FALSE`. В результате [столбец] = 0 будет true, если значение [столбец] равно 0 или BLANK. В отличие от этого, [столбец] == 0 истинно только тогда, когда значение [столбец] равно 0.

Оператор объединения текста

Используйте символ амперсанда (`&`) для конкатенации двух или более текстовых строк, чтобы получить один фрагмент текста.

Развернуть таблицу

Текстовый оператор	Значение	Пример
`&` (амперсанд)	Подключает или сцепляет два значения для создания одного непрерывного текстового значения	`[Region] & ", " & [City]`

Логические операторы

Используйте логические операторы (`&&`) и (`||`) для объединения выражений для создания одного результата.

Развернуть таблицу

Текстовый оператор	Значение	Примеры
(двойной знак &)	Создает условие AND между двумя выражениями, каждое из которых имеет логический результат. Если оба выражения возвращают `TRUE`, сочетание выражений также возвращает `TRUE`; в противном случае сочетание возвращает `FALSE`.	`([Region] = "France") && ([BikeBuyer] = "yes"))`
`||` (символ двойного канала)	Создает условие OR между двумя логическими выражениями. Если одно из выражений возвращает `TRUE`, результат `TRUE`; только если оба выражения `FALSE`, результат является `FALSE`.	`(([Region] = "France") || ([BikeBuyer] = "yes"))`
`IN`	Создает логическое OR условие для каждой строки при сравнении с таблицей. Примечание. Синтаксис конструктора таблиц использует фигурные скобки.	`'Product'[Color] IN { "Red", "Blue", "Black" }`

Операторы и порядок приоритета

В некоторых случаях порядок выполнения вычисления может повлиять на возвращаемое значение; Поэтому важно понимать, как определяется порядок и как можно изменить порядок получения нужных результатов.

Порядок вычислений

Выражение вычисляет операторы и значения в заданном порядке. Все выражения всегда начинаются с знака равенства (=). Знак равенства указывает, что следующие символы представляют собой выражение.

После знака равенства вычисляются элементы (операнды), разделенные операторами вычислений. Выражения всегда считываются слева направо, но порядок группирования элементов можно контролировать в некоторой степени с помощью круглых скобок.

Приоритет оператора

При объединении нескольких операторов в одной формуле операции упорядочены в соответствии со следующей таблицей. Если операторы имеют равное значение

приоритета, они упорядочены слева направо. Например, если выражение содержит оператор умножения и деления, они вычисляются в том порядке, в котором они отображаются в выражении слева направо.

Развернуть таблицу

Оператор	Описание
`^`	Экспоненциация
`–`	Знак (например, –1)
`*` и `/`.	Умножение и деление
`+` и `–`.	Добавление и вычитание
`&`	Подключает две строки текста (объединение)
`=,==,<,>,<=,>=,<>,IN`	Сравнение
`NOT`	`NOT` (унарный оператор)

Использование круглых скобок для управления порядком вычисления

Чтобы изменить порядок оценки, необходимо заключить в скобки эту часть формулы, которую необходимо вычислить первым. Например, следующая формула создает 11, так как умножение вычисляется перед добавлением. Формула умножается на 2 на 3, а затем добавляет 5 к результату.

```
DAX
=5+2*3
```

В отличие от этого, если вы используете скобки для изменения синтаксиса, порядок изменяется таким образом, чтобы 5 и 2 были добавлены вместе, а результат умножен на 3, чтобы создать 21.

```
DAX
=(5+2)*3
```

В следующем примере скобки вокруг первой части формулы требуют сначала вычислить выражение `(3 + 0.25)`, а затем разделить его результат на результат выражения `(3 -`

`0.25)`.

```
DAX

=(3 + 0.25)/(3 - 0.25)
```

В следующем примере сначала используется оператор возведения в степень в соответствии с правилами приоритета, а затем применяется оператор знака. Результатом этого выражения является -4.

```
DAX

=-2^2
```

Чтобы убедиться, что знак операции применяется к числовому значению первым, можно использовать скобки, чтобы управлять операторами, как показано в следующем примере. Результатом этого выражения является 4.

```
DAX

= (-2)^2
```

Совместимость

DAX легко обрабатывает и сравнивает различные типы данных, такие как Microsoft Excel. Однако базовый механизм вычислений основан на службах SQL Server Analysis Services и предоставляет дополнительные расширенные функции реляционного хранилища данных, включая расширенную поддержку типов дат и времени. Поэтому в некоторых случаях результаты вычислений или поведения функций могут не совпадать с результатами в Excel. Кроме того, DAX поддерживает больше типов данных, чем Excel. В этом разделе описаны основные различия.

Принудительное выполнение типов данных операндов

Как правило, два операнда в левой и правой части любого оператора должны быть одинаковыми типами данных. Однако если типы данных отличаются, DAX они преобразуются в общий тип данных для применения оператора в некоторых случаях:

1. Оба операнда преобразуются в самый большой распространенный тип данных.
2. Оператор применяется, если это возможно.

Например, предположим, что у вас есть два числа, которые нужно объединить. Одно число получается в результате применения формулы, например `= [Price] * .20`, и результат может содержать много десятичных знаков. Другое число — целое число, предоставленное в виде строкового значения.

В этом случае DAX оба числа преобразуются в реальные числа в числовом формате, используя самый большой числовой формат, который может хранить оба типа чисел. Затем DAX будет применять умножение.

В зависимости от сочетания типов данных приведение типов может не применяться для операций сравнения. Полный список типов данных, поддерживаемых данными, см. в разделе "Типы данных", поддерживаемые DAXтабличными моделями и типами данных в Power BI Desktop.

Целочисленное число, реальное число, валюта, дата и время и пустое считаются числовыми для сравнения. Пустое значение равно нулю при выполнении сравнения. Для операций сравнения поддерживаются следующие сочетания типов данных.

Развернуть таблицу

Тип данных в левой части	Тип данных на правой стороне
Числовой	Числовой
булев	булев
Струна	Струна

Другие сравнения смешанных типов данных возвращают ошибку. Например, формула, например ="1" > 0, возвращает ошибку, указывающую, что *DAX операции сравнения не поддерживают сравнение значений типа Text со значениями целочисленного типа*.

Развернуть таблицу

Типы данных, используемые в DAX	Типы данных, используемые в Excel
Числа (I8, R8)	Числа (R8)
Струна	Струна
булев	булев
ДатаВремя	Вариант
Валюта	Валюта

Различия в порядке приоритета

Порядок приоритета операций в DAX формулах в основном совпадает с тем, что используется Microsoft Excel, но некоторые операторы Excel не поддерживаются, например процент. Кроме того, диапазоны не поддерживаются.

Поэтому всякий раз, когда вы копируете и вставляете формулы из Excel, внимательно просматривайте формулу, так как некоторые операторы или элементы в формулах могут быть недопустимыми. Если есть сомнения в том порядке, в котором выполняются операции, рекомендуется использовать скобки для управления порядком операций и удаления неоднозначности результатов.

Связанный контент

DAX именование синтаксическогоDAX параметра

запросы DAX

Статья • 17.12.2024

Программное обеспечение для построения отчетов, такое как Power BI and Excel, запускает запросы DAX при отображении визуальных элементов в отчете, or добавлении поля в таблицу, and и DAX запросы корректируются при применении фильтра filter. Анализатор производительности в Power BI Desktop может показать эти DAX запросы andeven запускать их в представлении запросов DAX.

С помощью представления запросов в desktop BI Desktop запросов в службе бизнес-аналитики можно создать выполнять собственные запросы . С помощью Microsoft Fabricвы можете повысить производительность с помощью Copilot для записи DAX запросов в DAX представлении запросов классического or web. Помимо средств бизнес-аналитики PowerDAX запросы можно выполнять в записных книжках Fabric с помощью семантической связи для чтения данных из семантических моделей с помощью Python, and с REST API выполнения запросов, также доступных в Power Automate. Другие средства, такие как SQL Server Management Studio (SSMS), Power построителя отчетов бизнес-аналитики, and средства с открытым кодом, такие как DAX Studio, также позволяют создавать and выполнять запросы DAX.

DAX запросы возвращают результаты в виде таблицы right в средстве, что позволяет быстро создавать and тестировать производительность формул DAX в мерах, or просто просматривать данные в семантической модели. INFO and INFO. Функции VIEW DAX также могут получать сведения о семантической модели, например список таблиц, столбцов, метрик, and и многое другое.

Прежде чем изучать запросы, важно иметь четкое понимание основ DAX. If если вы еще этого не сделали, обязательно посмотрите обзор DAX.

Ключевые слова

DAX запросы имеют простой синтаксис, состоящий только из одного обязательного ключевого слова, EVALUATE. за EVALUATE следует табличное выражение, например функция DAXor имени таблицы, которая при запуске выводит таблицу результатов. К табличным выражениям, выводящим таблицу результатов, относятся:

1. Общие функции DAX, которые выводит таблицу, такие как SUMMARIZE, SUMMARIZECOLUMNS, SELECTCOLUMNS, FILTER, UNION, TOPN, ADDCOLUMNS, DATATABLE, and многие другие, работают с EVALUATE для вывода таблицы результатов.

2. Таблицы в модели, на которые ссылается имя, работают с EVALUATE для вывода таблицы результатов с данными в таблице. Например, **EVALUATE "Имя таблицы" можно выполнить как запрос**DAX.
3. Меры в модели or любой формулы DAX, возвращающей скалярную value, работают с EVALUATE, чтобы отобразить value в формате таблицы результатов, когда заключены в фигурные скобки. Например, **EVALUATE {[Total Sales]}**or**EVALUATE {COUNTROWS('Sales')}** можно запустить как запрос DAX. Они называются конструкторами таблиц .

Существует несколько необязательных ключевых слов, относящихся к запросам DAX: ORDER BY, START AT, DEFINE, MEASURE, VAR, TABLE, and COLUMN.

EVALUATE (обязательно)

На самом базовом уровне запрос DAX — это оператор `EVALUATE`, содержащий табличное выражение. Требуется хотя бы одна запись EVALUATE, но запрос может содержать любое количество EVALUATEstatements.

Синтаксис EVALUATE

DAX

```
EVALUATE <table>
```

параметры EVALUATE

Развернуть таблицу

Срок	Определение
`table`	Табличное выражение.

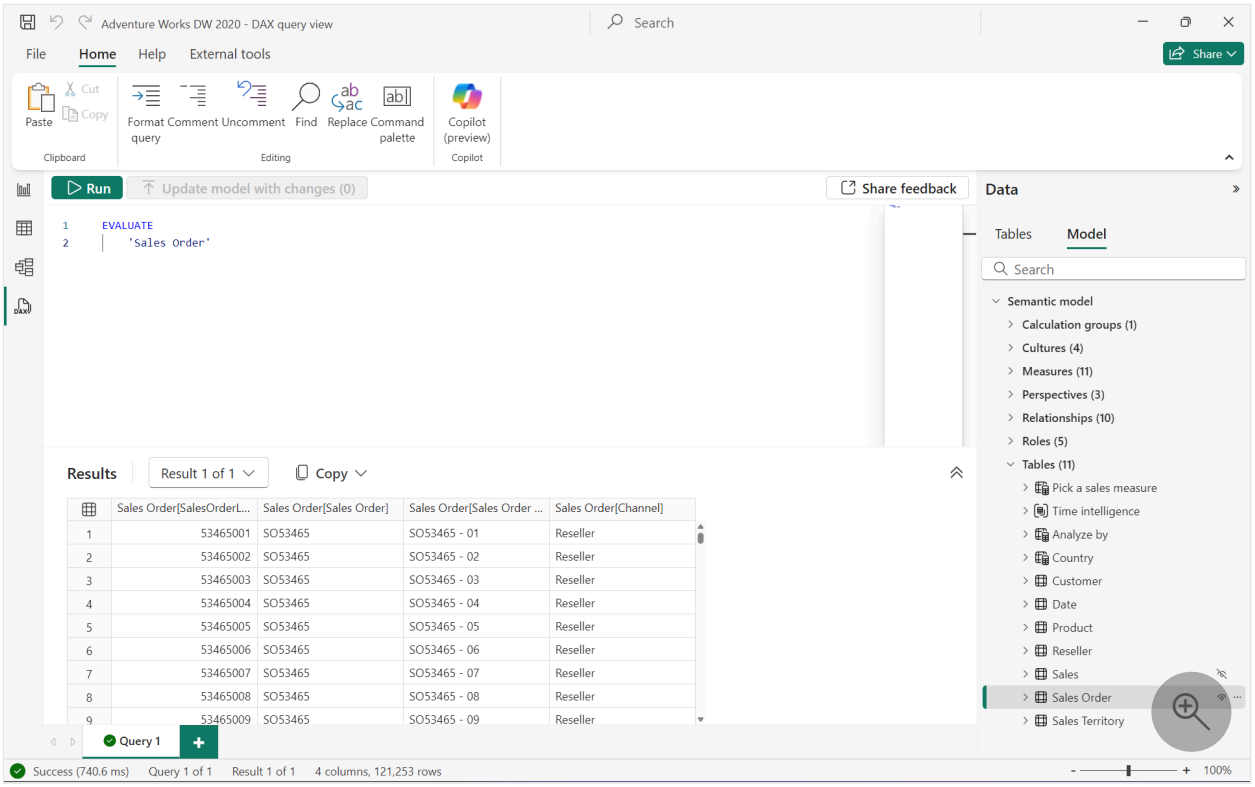
Пример EVALUATE

DAX

```
EVALUATE
    'Sales Order'
```

Возвращает all строку and столбец из таблицы "Заказ на продажу" в виде результирующей таблицы. Это может быть ограничено посредством

использования TOPNorFILTERи упорядочивания and с помощью ORDER BY.



ORDER BY (необязательно)

Необязательное ключевое слово `ORDER BY` определяет один or больше столбцов в выражениях запроса or, используемых для сортировки результатов запроса. Любое выражение, которое можно оценить для каждой строки результата, допустимо. Любой столбец в самом запросе также является допустимым.

Сортировка по свойству столбца в семантических моделях not применяется к результатам запроса DAX. If столбец должен быть отсортирован по другому столбцу в модели, например в случае Month имени, сортировка по столбцу также должна быть включена в запрос DAX, который будет использоваться в ORDER BY.

Синтаксис ORDER BY

DAX

```
EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, …]]
```

параметры ORDER BY

Развернуть таблицу

Срок	Определение
`expression`	Любое выражение DAX, возвращающее один скалярный value, or столбец, включенный в запрос DAX.
`ASC`	(по умолчанию) Порядок сортировки по возрастанию.
`DESC`	Порядок сортировки по убыванию.

Пример ORDER BY

DAX

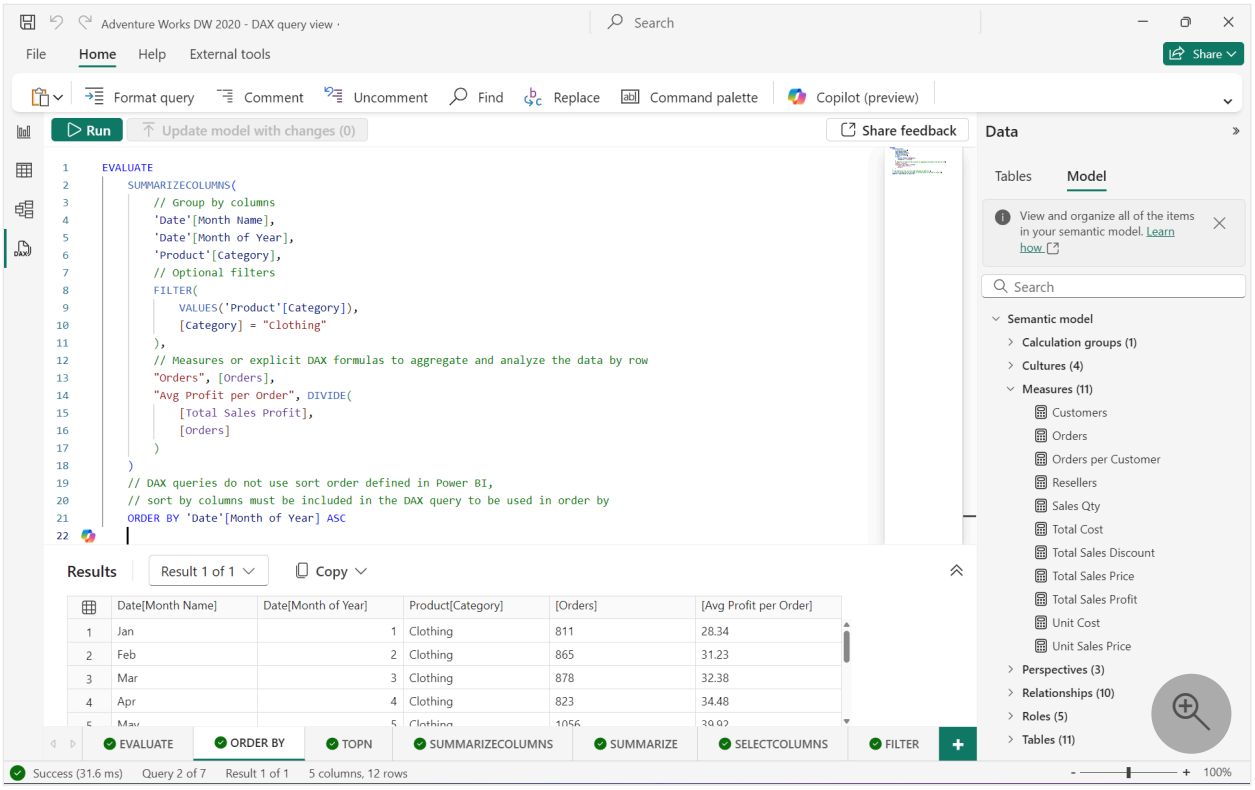
```
EVALUATE
    SUMMARIZECOLUMNS(
        // Group by columns
        'Date'[Month Name],
        'Date'[Month of Year],
        'Product'[Category],

        // Optional filters
        FILTER(
            VALUES('Product'[Category]),
            [Category] = "Clothing"
        ),

        // Measures or explicit DAX formulas to aggregate and analyze the
data by row
        "Orders", [Orders],
        "Avg Profit per Order", DIVIDE(
            [Total Sales Profit],
            [Orders]
        )
    )

    // DAX queries do not use sort order defined in Power BI,
    // sort by columns must be included in the DAX query to be used in order
by
    ORDER BY 'Date'[Month of Year] ASC
```

Возвращает заказы одежды andaverage прибыль на заказ по month, в порядке возрастания по month, в виде таблицы результатов.

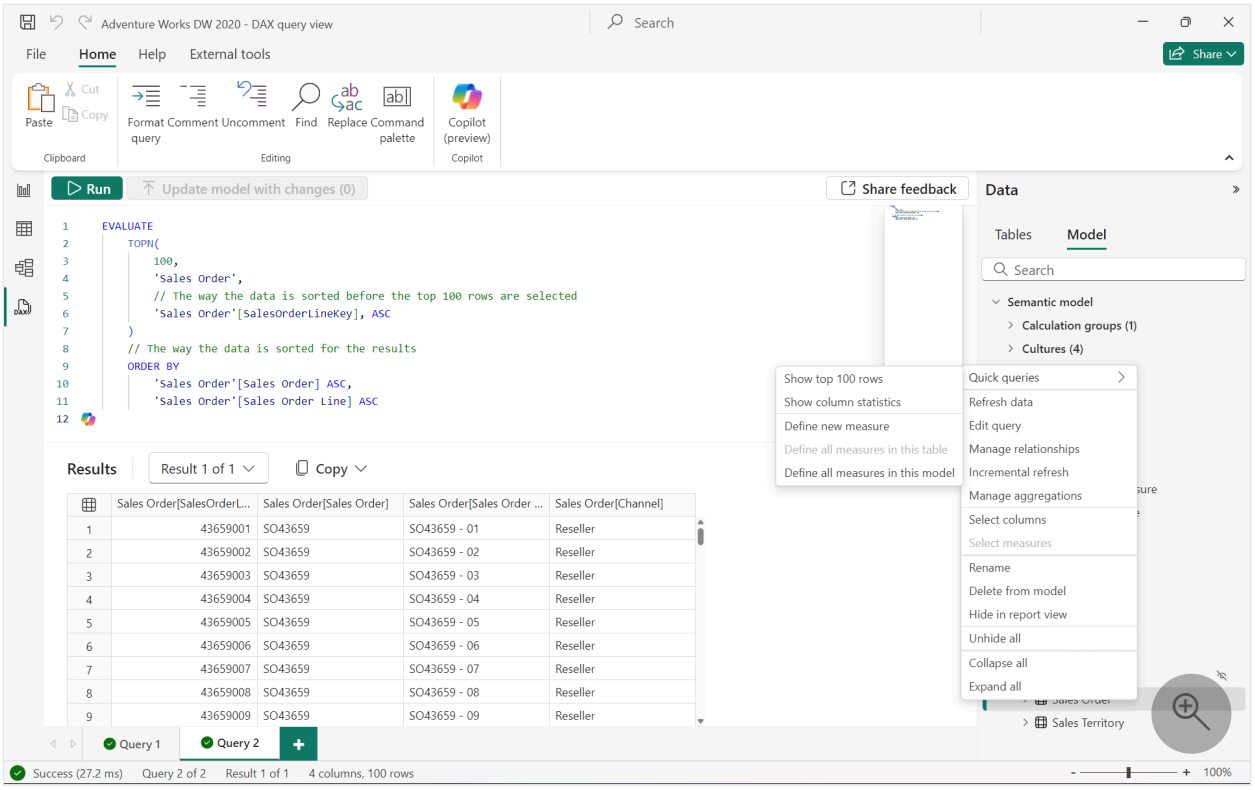


TOPN выполняет ли not выбор указанного количества строк в соответствии с порядком сортировки, указанным в ORDER BY? Вместо этого TOPN имеет собственный синтаксис, чтобы дополнительно указать сортировку до возврата первых 100 строк. ORDER BY сортирует только таблицу результатов, возвращаемую TOPN.

DAX

```
EVALUATE
    TOPN(
        100,
        'Sales Order',
        // The way the data is sorted before the top 100 rows are selected
        'Sales Order'[SalesOrderLineKey], ASC
    )
    // The way the data is sorted for the results
    ORDER BY
        'Sales Order'[Sales Order] ASC,
        'Sales Order'[Sales Order Line] ASC
```

Возвращает первые 100 заказов по продажам, отсортированных по возрастанию SalesOrderLienKey, а затем сортирует результаты first по заказу на продажу, а затем по строке заказов на продажу.



START AT (необязательно)

Необязательное ключевое слово `START AT` используется в предложении `ORDER BY`. Он определяет value, с которого начинаются результаты запроса.

Синтаксис START AT

DAX

```
EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, …]
[START AT {<value>|<parameter>} [, …]]]
```

параметры START AT

Развернуть таблицу

Срок	Определение
`value`	Константа value. Не может быть выражением.
`parameter`	Имя параметра в инструкции XMLA, которому предшествует символ `@`.

START AT примечания

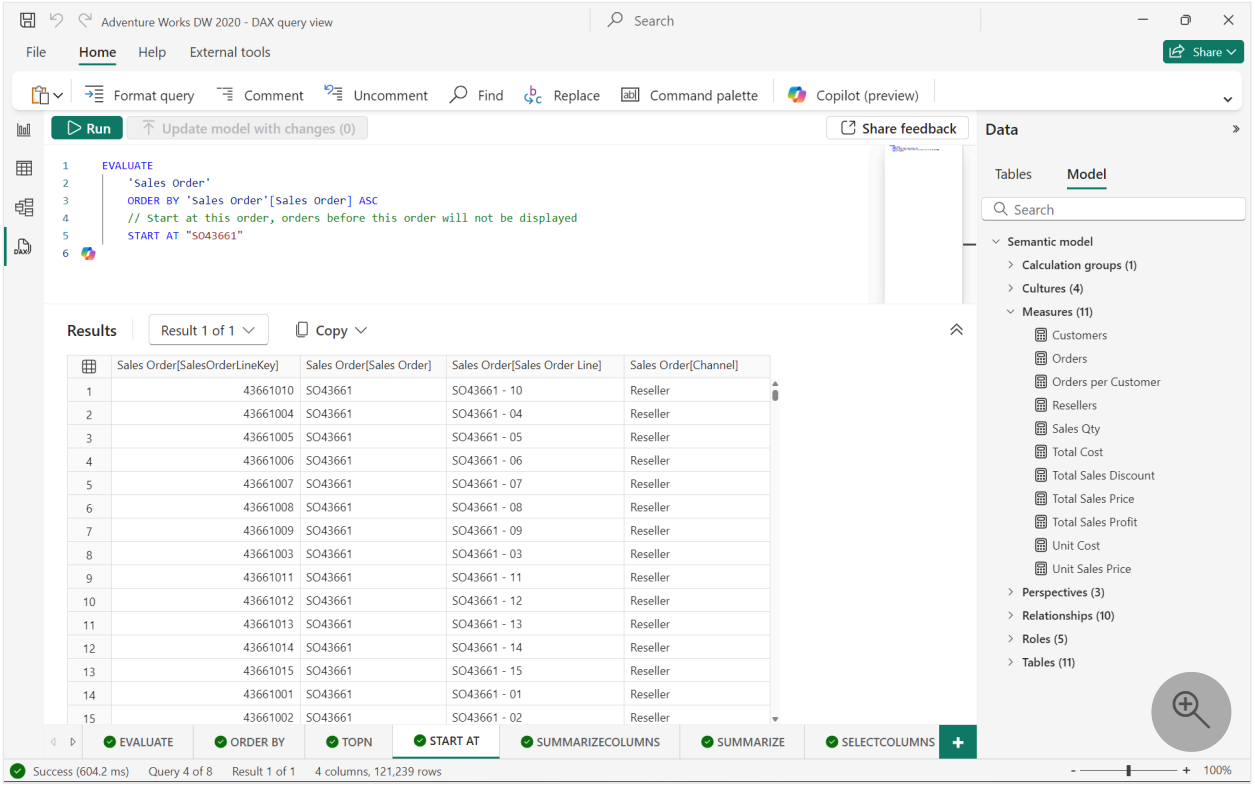
Аргументы START AT соответствуют столбцам в конструкции ORDER BY "один к одному". В предложении START AT может быть столько аргументов, сколько в предложении ORDER BY, но not больше. Аргумент first в START AT определяет начальный value в первом столбце из столбцов ORDER BY. Аргумент second в START AT определяет начальный value во втором столбце среди столбцов ORDER BY в строках, которые соответствуют firstvalue для столбца 1.

Пример START AT

DAX

```
EVALUATE
    'Sales Order'
    ORDER BY 'Sales Order'[Sales Order] ASC
    // Start at this order, orders before this order will not be displayed
    START AT "SO43661"
```

Возвращает all столбцы из таблицы "Заказ продаж" в порядке возрастания по заказу на продажу, начиная с SO43661. Строки перед этим заказом на продажу not включены в таблицу результатов.



	Sales Order[SalesOrderLineKey]	Sales Order[Sales Order]	Sales Order[Sales Order Line]	Sales Order[Channel]
1	43661010	SO43661	SO43661 - 10	Reseller
2	43661004	SO43661	SO43661 - 04	Reseller
3	43661005	SO43661	SO43661 - 05	Reseller
4	43661006	SO43661	SO43661 - 06	Reseller
5	43661007	SO43661	SO43661 - 07	Reseller
6	43661008	SO43661	SO43661 - 08	Reseller
7	43661009	SO43661	SO43661 - 09	Reseller
8	43661003	SO43661	SO43661 - 03	Reseller
9	43661011	SO43661	SO43661 - 11	Reseller
10	43661012	SO43661	SO43661 - 12	Reseller
11	43661013	SO43661	SO43661 - 13	Reseller
12	43661014	SO43661	SO43661 - 14	Reseller
13	43661015	SO43661	SO43661 - 15	Reseller
14	43661001	SO43661	SO43661 - 01	Reseller
15	43661002	SO43661	SO43661 - 02	Reseller

DEFINE (необязательно)

Необязательное ключевое слово `DEFINE` содержит одно or больше определений вычисляемых сущностей, которые существуют только для duration запроса. В

отличие от `EVALUATE`, в запросе DAX может быть только один блок `DEFINE` с более чем одним or определением. `DEFINE` должны предшествовать and инструкции first`EVALUATE` допустимы для allEVALUATEstatements в запросе. Определения могут быть переменными, мерами, таблицами[1], and столбцами[1]. Определения могут ссылаться на другие определения, которые появляются перед or, предшествуя текущему определению. Хотя бы одно определение является обязательным, if ключевое слово `DEFINE` включается в запрос.

`DEFINE MEASURE` — это распространенный сценарий для создания новых мер, or редактирования существующих мер в семантической модели. Если measure уже существует в модели, запрос DAX будет использовать формулу measureDAX, определенную в запросе. Это полезно для тестирования мер с помощью запроса DAX перед обновлением модели.

`DEFINE MEASURE` также полезно создать дополнительный анализ с формулами DAX для конкретного запроса DAX, где вы можете not иметь разрешение на добавление модели measureor необходимо not его в модели.

Синтаксис DEFINE

DAX

```
[DEFINE
    (
     (MEASURE <table name>[<measure name>] = <scalar expression>) |
     (VAR <var name> = <table or scalar expression>) |
     (TABLE <table name> = <virtual table definition>) |
     (COLUMN <table name>[<column name>] = <scalar expression>) |
    ) +
]

(EVALUATE <table expression>) +
```

параметры DEFINE

Развернуть таблицу

Срок	Определение
`Entity`	MEASURE, VAR, TABLE[1], or COLUMN[1].
`name`	Имя для measure, var, определения таблицы, колонки or. Это не может быть выражением. Имя not должно быть уникальным. Имя существует только для duration запроса.

Срок	Определение
`expression`	Любое выражение DAX, которое возвращает таблицу or или скалярное значение value. Выражение может использовать любую из определенных сущностей. If необходимо convert преобразовать скалярное выражение в табличное выражение, обернуть выражение внутри конструктора таблицы с фигурными скобками `{}`, or использовать функцию `ROW()` для возврата таблицы с одной строкой.

[1] Предупреждение: **определения** запроса and таблицы столбцов предназначены исключительно для внутреннего использования. Хотя вы можете использовать выражения define TABLE and COLUMN для запроса без синтаксической ошибки error, они могут вызвать ошибки времени выполнения и поэтому не рекомендуются and и not.

DEFINE примечания

- Запрос DAX может иметь несколько EVALUATEstatements, но может иметь только одну инструкцию DEFINE. Определения в инструкции DEFINE могут применяться к любым EVALUATEstatements в запросе.
- В инструкции DEFINE требуется по крайней мере одно определение.
- Measure определения для модели запроса переопределяют меры с тем же именем, но используются только в рамках самого запроса. Они not повлияют на measureмодель.
- VAR имена имеют уникальные ограничения. Дополнительные сведения см. в разделе VAR — параметры.

Пример DEFINE

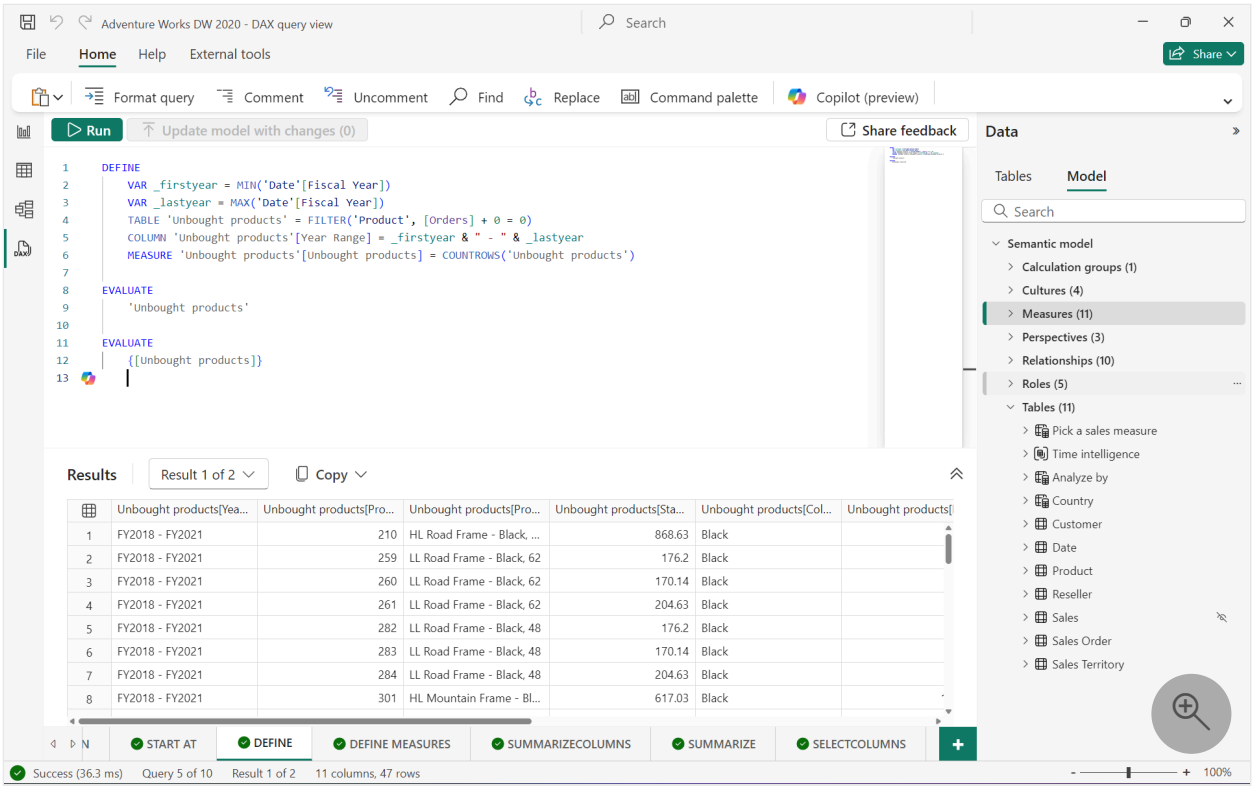
DAX

```
DEFINE
    VAR _firstyear = MIN('Date'[Fiscal Year])
    VAR _lastyear = MAX('Date'[Fiscal Year])
    TABLE 'Unbought products' = FILTER('Product', [Orders] + 0 = 0)
    COLUMN 'Unbought products'[Year Range] = _firstyear & " - " & _lastyear
    MEASURE 'Unbought products'[Unbought products] = COUNTROWS('Unbought
products')

EVALUATE
    'Unbought products'
```

```
EVALUATE
    {[Unbought products]}
```

Возвращает таблицу, определенную в запросе DAX, чтобы отобразить некупленные продукты с дополнительным столбцом, который ссылается на заданные переменные. Кроме того, measure определяется and как count строки непроданных продуктов.



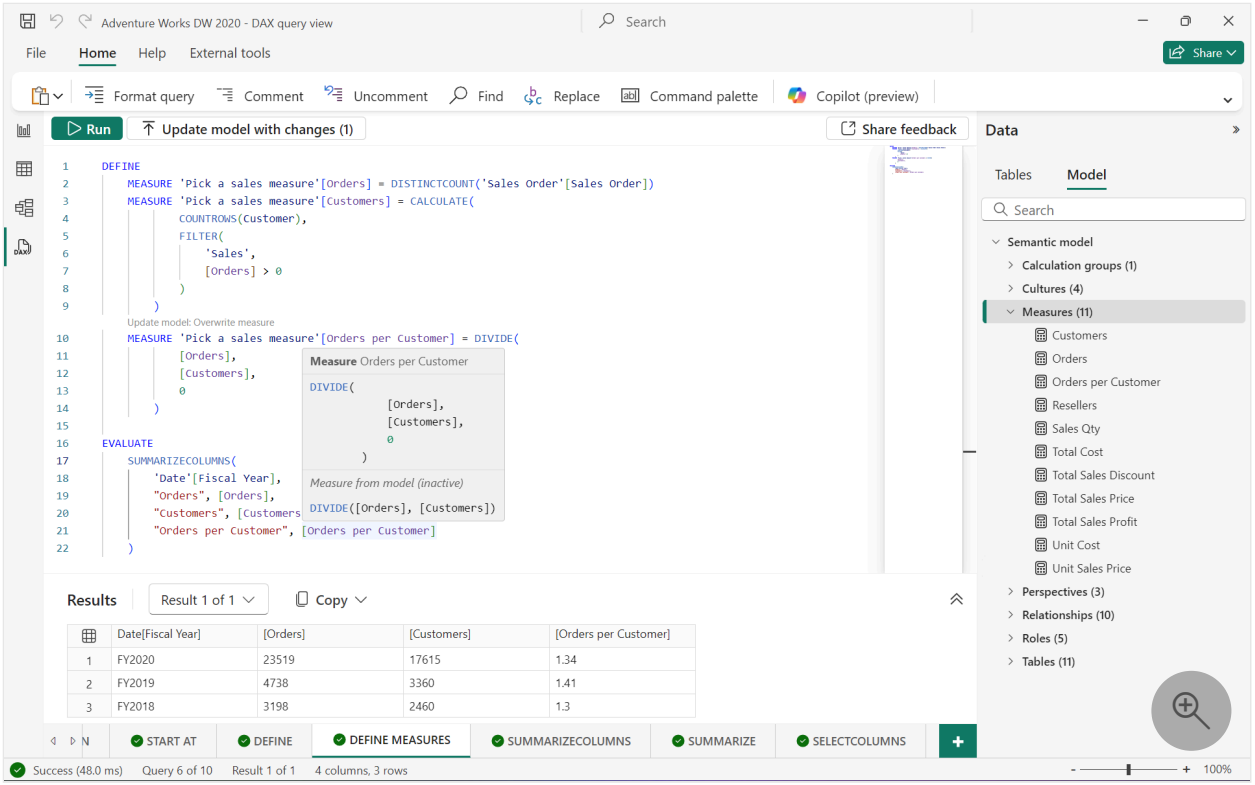

```
DAX
```

```dax
DEFINE
    MEASURE 'Pick a sales measure'[Orders] = DISTINCTCOUNT('Sales
Order'[Sales Order])
    MEASURE 'Pick a sales measure'[Customers] = CALCULATE(
            COUNTROWS(Customer),
            FILTER(
                'Sales',
                [Orders] > 0
            )
        )
    MEASURE 'Pick a sales measure'[Orders per Customer] = DIVIDE(
            [Orders],
            [Customers],
            0
        )

EVALUATE
    SUMMARIZECOLUMNS(
        'Date'[Fiscal Year],
        "Orders", [Orders],
```

```
        "Customers", [Customers],
        "Orders per Customer", [Orders per Customer]
    )
```

Возвращает таблицу, оценивающую три заданные меры, чтобы продемонстрировать результаты по финансовым year. All измерения также существуют в модели, and Заказы на Клиента модифицированы в запросе DAX.



Параметры в запросах DAX

Четко заданное заявление запроса DAX можно параметризовать and, а затем использовать снова and, изменяя только параметр values.

Метод Execute Method (XMLA) содержит элемент коллекции Parameters Element (XMLA), который позволяет определять параметры назначенным . Внутри коллекции каждый элемент параметра (XMLA) определяет имя параметра and и атрибут value к нему.

Ссылайтесь на параметры XMLA, добавляя префикс `@` к имени параметра. Любое место в синтаксисе, где разрешено value, value можно заменить вызовом параметра. All параметры XMLA имеют текстовый тип.

> **ⓘ Важно!**

Параметры, определённые в разделе параметров andnot, используемые в элементе `<STATEMENT>` generate, в ответе error в XMLA. Параметры, используемые andnot, определенные в элементе `<Parameters>` generate ответа error в XMLA.

содержимое Related

DAX statements SUMMARIZECOLUMNS

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

соглашения об именовании параметров DAX

Статья • 22.01.2025

Имена параметров стандартизированы в справочном документе DAX для облегчения использования и понимания функций.

Имена параметров

Развернуть таблицу

Срок	Определение
`expression`	Любое выражение DAX, возвращающее одно скалярное значение, где выражение вычисляется несколько раз (для каждой строки или контекста).
`value`	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение должно оцениваться ровно один раз перед всеми остальными операциями.
`table`	Любое выражение DAX, возвращающее таблицу данных.
`tableName`	Имя существующей таблицы с использованием стандартного синтаксиса DAX. Это не может быть выражением.
`columnName`	Имя существующего столбца с использованием стандартного синтаксиса DAX, обычно полностью квалифицированное. Это не может быть выражением.
`name`	Строка константы, которая будет использоваться для предоставления имени нового объекта.
`order`	Перечисление, используемое для определения порядка сортировки.
`ties`	Перечисление, используемое для определения обработки равных значений.
`type`	Перечисление, используемое для определения типа данных для PathItem и PathItemReverse.

Использование префиксов для имени параметров или только префикса.

Развернуть таблицу

Срок	Определение
`prefixing`	Имена параметров могут быть более квалифицированы с префиксом, который описывает использование аргумента и чтобы избежать неоднозначного чтения параметров. Например: Result_ColumnName — ссылается на существующий столбец, используемый для получения значений результатов в функции LOOKUPVALUE(). Search_ColumnName — ссылается на существующий столбец, используемый для поиска значения в функции LOOKUPVALUE().
`omitting`	Имена параметров будут опущены, если префикс достаточно ясен для описания параметра. Например, вместо использования следующего синтаксиса DATE (Year_Value, Month_Value, Day_Value) пользователь может читать DATE (год, месяц, день); повторяющееся три раза значение суффикса не добавляет ничего в лучшее понимание функции, и она загромождает чтение ненужным образом. Однако, если параметр содержит префикс Year_columnName, то имя параметра и префикс сохраняются, чтобы пользователь понял, что параметр требует ссылку на существующий столбец с годами.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет

синтаксис DAX

Статья • 23.01.2025

В этой статье описаны синтаксис и требования к языку выражений формул DAX.

Требования к синтаксису

Формула DAX всегда начинается с знака равенства (`=`). После знака равенства можно указать любое выражение, которое вычисляется скалярным или выражение, которое можно преобразовать в скаляр. К ним относятся следующие:

- Скалярная константа или выражение, использующее скалярный оператор (`+,-,*,/,>=,...,&&, ...`)
- Ссылки на столбцы или таблицы. Язык DAX всегда использует таблицы и столбцы в качестве входных данных для функций, никогда не массив или произвольный набор значений.
- Операторы, константы и значения, предоставляемые в рамках выражения.
- Результат функции и её обязательные аргументы. Некоторые функции DAX возвращают таблицу вместо скалярного значения и должны быть обёрнуты в функцию, которая вычисляет таблицу и возвращает скаляр; если таблица является таблицей с одним столбцом и одной строкой, то она рассматривается как скалярное значение.

  Большинство функций DAX требуют одного или нескольких аргументов, которые могут включать таблицы, столбцы, выражения и значения. Однако некоторые функции, такие как PI, не требуют никаких аргументов, но всегда требуют скобки для указания аргумента NULL. Например, необходимо всегда вводить PI(), а не PI. Вы также можете вложить функции в другие функции.
- Выражения. Выражение может содержать любой или все из следующих: операторы, константы или ссылки на столбцы.

Например, ниже приведены все допустимые формулы.

Развернуть таблицу

Формула	Результат
`= 3`	3

Формула	Результат
`= "Sales"`	**Продажи**
`= 'Sales'[Amount]`	Если вы используете эту формулу в таблице Sales, вы получите значение столбца "Сумма" в таблице Sales для текущей строки.

`= (0.03 * [Amount])`
 	
`=0.03 * [Amount]`
	Три процента значения в столбце "Сумма" текущей таблицы.
	
	Хотя эту формулу можно использовать для вычисления процента, результат не отображается в процентах, если в таблице не применяется форматирование.

`= PI()`	Значение константы pi.

Формулы могут вести себя по-разному в зависимости от того, как они используются. Необходимо всегда учитывать контекст и то, как данные, используемые в формуле, связаны с другими данными, которые могут использоваться в вычислении.

Требования к именованию

Модель данных часто содержит несколько таблиц. Вместе таблицы и их столбцы составляют базу данных, хранящуюся в подсистеме аналитики в памяти (VertiPaq). В этой базе данных все таблицы должны иметь уникальные имена. Имена столбцов также должны быть уникальными в каждой таблице. Все имена объектов *без учета регистра*; Например, имена **SALES** и **Sales** будут представлять ту же таблицу.

Каждый столбец и мера, добавляемая в существующую модель данных, должны принадлежать определенной таблице. Вы указываете таблицу, содержащую столбец неявно, при создании вычисляемого столбца в таблице или явным образом при создании меры и указании имени таблицы, в которой должно храниться определение меры.

При использовании таблицы или столбца в качестве входных данных для функции обычно необходимо *указывать* имя столбца. *полное* имя столбца — это имя таблицы, за которым следует имя столбца в квадратных скобках: например, "Продажи в США"[Продукты]. Полностью квалифицированное имя всегда требуется при обращении к столбцу в следующих контекстах:

- В качестве аргумента функции `VALUES`
- В качестве аргумента функции `ALL` или `ALLEXCEPT`

- В аргументе фильтра для функций `CALCULATE` или `CALCULATETABLE`
- В качестве аргумента функции `RELATEDTABLE`
- В качестве аргумента для любой функции анализа времени

Имя столбца *без параметров* — это только имя столбца, заключенного в квадратные скобки: например, [Сумма продаж]. Например, при ссылке на скалярное значение из той же строки текущей таблицы можно использовать неквалифицированное имя столбца.

Если имя таблицы содержит пробелы, зарезервированные ключевые слова или запрещенные символы, необходимо заключить имя таблицы в одинарные кавычки. Кроме того, необходимо заключить имена таблиц в кавычки, если имя содержит любые символы за пределами буквенно-цифрового диапазона ANSI, независимо от того, поддерживает ли языковой стандарт набор символов. Например, если открыть книгу, содержащую имена таблиц, написанные в кириллических символах, например "Таблица", имя таблицы должно быть заключено в кавычки, даже если оно не содержит пробелов.

> ⓘ **Примечание**
>
> Чтобы упростить ввод полных имен столбцов, используйте функцию автозаполнения в редакторе формул.

Таблицы

- Имена таблиц требуются всякий раз, когда столбец находится в таблице, отличной от текущей таблицы. Имена таблиц должны быть уникальными в базе данных.
- Имена таблиц должны быть заключены в одинарные кавычки, если они содержат пробелы, другие специальные символы или любые буквенно-цифровые символы, отличные от английского.

Меры

- Имена мер всегда должны находиться в квадратных скобках.
- Имена мер могут содержать пробелы.

- Каждое имя меры должно быть уникальным в модели. Таким образом, имя таблицы является необязательным перед именем меры при ссылке на существующую меру. Однако при создании меры необходимо всегда указывать таблицу, в которой будет храниться определение меры.

Столбцы

Имена столбцов должны быть уникальными в контексте таблицы; однако несколько таблиц могут иметь столбцы с одинаковыми именами (диамбигуация поставляется с именем таблицы).

Как правило, на столбцы можно ссылаться без указания на базовую таблицу, к которой они относятся, за исключением случаев, когда необходимо разрешить конфликт имен или при использовании функций, требующих полных имен столбцов.

Зарезервированные ключевые слова

Если имя, используемое для таблицы, совпадает с зарезервированным ключевым словом Analysis Services, возникает ошибка и необходимо переименовать таблицу. Однако ключевые слова можно использовать в именах объектов, если имя объекта заключено в квадратные скобки (для столбцов) или кавычки (для таблиц).

> ⓘ **Примечание**
>
> Кавычки могут быть представлены несколькими различными символами в зависимости от приложения. Если вы вставляете формулы из внешнего документа или веб-страницы, убедитесь, что код ASCII символов, используемых для открытия и закрытия кавычек, совпадает. В противном случае DAX не удается распознать символы как кавычки, что делает ссылку недопустимой.

Специальные символы

В именах таблиц, столбцов или мер недопустимы следующие символы и типы символов:

- Начальные или конечные пробелы, если только они не заключены в разделители имен, скобки или одиночные апострофы.
- Управляющие символы

- Следующие символы, недопустимые в именах объектов:

  `.,;':/\\*|?&%$!+=()[]{}<>`

Примеры имен объектов

В следующей таблице показаны примеры некоторых имен объектов:

Развернуть таблицу

Типы объектов	Примеры	Комментарий
Имя таблицы	**Продажи**	Если имя таблицы не содержит пробелы или другие специальные символы, имя не должно быть заключено в кавычки.
Имя таблицы	**"Продажи в Канаде"**	Если имя содержит пробелы, вкладки или другие специальные символы, заключите имя в одинарные кавычки.
Полностью квалифицированное имя столбца	**продажи[сумма]**	Имя таблицы предшествует имени столбца, а имя столбца заключено в квадратные скобки.
Полное квалифицированное имя метрики	**Продажи/ Прибыль**	Имя таблицы предшествует имени меры, а имя меры заключено в квадратные скобки. В определенных контекстах всегда требуется полностью квалифицированное имя.
Неуказанное имя столбца	**[Сумма]**	Неквалифицированное имя — это имя столбца, заключенное в квадратные скобки. Контексты, в которых можно использовать некавалифицированное имя, включают формулы в вычисляемый столбец в той же таблице или в функцию агрегирования, которая сканирует ту же таблицу.
Полный столбец в таблице с пробелами	**"Продажи в Канаде"[Кол-во]**	Имя таблицы содержит пробелы, поэтому он должен быть окружен одними кавычками.

Другие ограничения

Синтаксис, необходимый для каждой функции, и тип операции, который он может выполнять, сильно зависит от функции. Однако в целом следующие правила применяются ко всем формулам и выражениям:

- DAX формулы и выражения не могут изменять или вставлять отдельные значения в таблицы.
- Вы не можете создавать вычисляемые строки с помощью DAX. Вы можете создавать только вычисляемые столбцы и меры.
- При определении вычисляемых столбцов можно вложить функции на любой уровень.
- DAX имеет несколько функций, возвращающих таблицу. Как правило, значения, возвращаемые этими функциями, используются в качестве входных данных для других функций, для которых требуется таблица в качестве входных данных.

операторы и константы DAX

В следующей таблице перечислены операторы, поддерживаемые DAX. Дополнительные сведения о синтаксисе отдельных операторов см. в DAX операторах.

Развернуть таблицу

Тип оператора	Символ и использование
Оператор круглых скобок	`()` порядке приоритета и группировке аргументов
Арифметические операторы	`+` (добавление) `-` (вычитание) `*` (умножение) `/` (деление) `^` (экспонентация)
Операторы сравнения	`=` (равно) `>` (больше) `<` (меньше) `>=` (больше или равно) `<=` (меньше или равно)

Тип оператора	Символ и использование
	`<>` (не равно)
Оператор объединения текста	`&` (объединение)
Операторы логики	`&&` (и) `||` (или)

Типы данных

Не требуется приведение, преобразование или другое указание типа данных столбца или значения, используемого в формуле DAX. При использовании данных в формуле DAXDAX автоматически определяет типы данных в ссылочных столбцах и значениях, которые вы вводите, и выполняет неявные преобразования при необходимости для выполнения указанной операции.

Например, если вы пытаетесь добавить число в значение даты, подсистема интерпретирует операцию в контексте функции и преобразует числа в общий тип данных, а затем представляет результат в предполагаемом формате, дате.

Однако существуют некоторые ограничения на значения, которые можно успешно преобразовать. Если значение или столбец имеет тип данных, несовместимый с текущей операцией, DAX возвращает ошибку. Кроме того, DAX не предоставляет функций, которые позволяют явно изменять, преобразовывать или приводить тип данных уже импортированных в модель данных.

> ⓘ **Важно!**
>
> DAX не поддерживает использование вариантного типа данных. Поэтому при загрузке или импорте данных в модель данных ожидается, что данные в каждом столбце обычно совпадают.

Некоторые функции возвращают скалярные значения, включая строки, в то время как другие функции работают с числами, целыми числами и реальными числами, либо датами и временем. Тип данных, необходимый для каждой функции, описан в разделе DAX функций.

Таблицы, содержащие несколько столбцов и несколько строк данных, можно использовать в качестве аргумента функции. Некоторые функции также

возвращают таблицы, которые хранятся в памяти и могут использоваться в качестве аргументов для других функций.

Дата и время

DAX сохраняет значения даты и времени с помощью типа данных datetime, используемого Microsoft SQL Server. Формат datetime использует число с плавающей запятой, где значения даты соответствуют целочисленной части, представляющей количество дней с 30 декабря 1899 года. Значения времени соответствуют десятичной части значения даты, в которой часы, минуты и секунды представлены десятичными дробями дня. DAX функции даты и времени неявно преобразуют аргументы в тип данных datetime.

> ⓘ **Примечание**
>
> Точное максимальное значение DateTime, поддерживаемое DAX, — 31 декабря 9999 г. 00:00:00.

Литерал даты и времени

Начиная с версии Power BI Desktop за август 2021 г. значения даты и времени DAX можно указать в виде литерала в формате `dt"YYYY-MM-DD"`, `dt"YYYY-MM-DDThh:mm:ss"` или `dt"YYYY-MM-DD hh:mm:ss"`. При указании в качестве литерала использование функций DATE, TIME, DATEVALUE, TIMEVALUE в выражении не требуется.

Например, следующее выражение использует функции DATE и TIME для фильтрации по OrderDate:

DAX

```
EVALUATE
FILTER (
        FactInternetSales,
        [OrderDate] > (DATE(2015,1,9) + TIME(2,30,0)) &&[OrderDate] <
(DATE(2015,12,31) + TIME(11,59,59))
)
```

Указанное выражение фильтра можно задать как литерал:

DAX

```
EVALUATE
FILTER (
        FactInternetSales,
        [OrderDate] > dt"2015-1-9T02:30:00" && [OrderDate] < dt"2015-12-
31T11:59:59"
)
```

> **Примечание**
>
> Формат литерала типа DAX даты и времени не поддерживается во всех версиях Power BI Desktop, Analysis Services и Power Pivot в Excel. Новые и обновленные функции DAX обычно впервые появились в Power BI Desktop, а затем включены в службы Analysis Services и Power Pivot в Excel.

Обратная связь

Были ли сведения на этой странице полезными? Да Нет