

# Contents

## Справочник по выражениям анализа данных (DAX)

### Learn

[Общие сведения о DAX](#)

[Видео](#)

[Схема обучения "Использование DAX в Power BI Desktop"](#)

### Функции DAX

[Справочник по функциям DAX](#)

[Новые функции DAX](#)

### Функции даты и времени

[Общие сведения о функциях даты и времени](#)

[CALENDAR](#)

[CALENDARAUTO](#)

[DATE](#)

[DATEDIFF](#)

[DATEVALUE](#)

[DAY](#)

[EDATE](#)

[EOMONTH](#)

[HOUR](#)

[MINUTE](#)

[MONTH](#)

[NOW](#)

[QUARTER](#)

[SECOND](#)

[TIME](#)

[TIMEVALUE](#)

[TODAY](#)

[UTCNOW](#)

[UTCTODAY](#)

WEEKDAY

WEEKNUM

YEAR

YEARFRAC

## Функции фильтрации

[Общие сведения о функциях фильтрации](#)

ALL

ALLCROSSFILTERED

ALLEXCEPT

ALLNOBLANKROW

ALLSELECTED

CALCULATE

CALCULATETABLE

EARLIER

EARLIEST

FILTER

KEEPFILTERS

LOOKUPVALUE

REMOVEFILTERS

SELECTEDVALUE

## Финансовые функции

[Обзор финансовых функций](#)

ACCRINT

ACCRINTM

AMORDEGRC

AMORLINC

COUPDAYBS

COUPDAYS

COUPDAYSNC

COUPNCD

COUPNUM

COUPPCD

CUMIPMT  
CUMPRINC  
DB  
DDB  
DISC  
DOLLARDE  
DOLLARFR  
DURATION  
EFFECT  
FV  
INTRATE  
IPMT  
ISPMT  
MDURATION  
NOMINAL  
NPER  
ODDFPRICE  
ODDFYIELD  
ODDLPRICE  
ODDLYIELD  
PDURATION  
PMT  
PPMT  
PRICE  
PRICEDISC  
PRICEMAT  
PV  
RATE  
RECEIVED  
RRI  
SLN  
SYD

TBILLEQ

TBILLPRICE

TBILLYIELD

VDB

XIRR

XNPV

YIELD

YIELDDISC

YIELDMAT

## Информационные функции

Общие сведения об информационных функциях

CONTAINS

CONTAINSROW

CONTAINSSTRING

CONTAINSSTRINGEXACT

CUSTOMDATA

HASONEFILTER

HASONEVALUE

ISBLANK

ISCROSSFILTERED

ISEMPTY

ISERROR

ISEVEN

ISFILTERED

ISINSCOPE

ISLOGICAL

ISNONTEXT

ISNUMBER

ISODD

ISONORAFTER

ISSELECTEDMEASURE

ISSUBTOTAL

ISTEXT

NONVISUAL

SELECTEDMEASURE

SELECTEDMEASUREFORMATSTRING

SELECTEDMEASURENAME

USERNAME

USEROBJECTID

USERPRINCIPALNAME

## Логические функции

Общие сведения о логических функциях

AND

COALESCE

FALSE

IF

IFERROR

NOT

OR

SWITCH

true

## Математические и тригонометрические функции

Общие сведения о математических и тригонометрических функциях

ABS

ACOS

ACOSH

ACOT

ACOTH

ASIN

ASINH

ATAN

ATANH

CEILING

COMBIN

COMBINA

CONVERT

COS

COSH

ДЕНЕЖНАЯ ЕДИНИЦА

DEGREES

DIVIDE

EVEN

EXP

FACT

FLOOR

GCD

INT

ISO.CEILING

LCM

LN

LOG

LOG10

MOD

MROUND

ODD

PI

POWER

ПРОДУКТ

PRODUCTX

QUOTIENT

RADIANS

RAND

RANDBETWEEN

ROUND

ROUNDDOWN

ROUNDUP

[SIGN](#)

[SQRT](#)

[SUM](#)

[SUMX](#)

[TRUNC](#)

## [Другие функции](#)

[Общие сведения о других функциях](#)

[BLANK](#)

[ошибка](#)

## [Родительские и дочерние функции](#)

[Общие сведения о родительских и дочерних функциях](#)

[Основные сведения о функциях для иерархий "родители-потомки"](#)

[PATH](#)

[PATHCONTAINS](#)

[PATHITEM](#)

[PATHITEMREVERSE](#)

[PATHLENGTH](#)

## [Функции связей](#)

[Функции связей](#)

[CROSSFILTER](#)

[RELATED](#)

[RELATEDTABLE](#)

[USERELATIONSHIP](#)

## [Статистические функции](#)

[Общие сведения о статистических функциях](#)

[APPROXIMATEDISTINCTCOUNT](#)

[AVERAGE](#)

[AVERAGEA](#)

[AVERAGEX](#)

[BETA.DIST](#)

[BETA.INV](#)

[CHISQ.DIST](#)

CHISQ.DIST.RT  
CHISQ.INV  
CHISQ.INV.RT  
CONFIDENCE.NORM  
CONFIDENCE.T  
COT  
COTH  
COUNT  
COUNTA  
COUNTAX  
COUNTBLANK  
COUNTROWS  
COUNTX  
DISTINCTCOUNT  
EXPON.DIST  
GEOMEAN  
GEOMEANX  
MAX  
MAXA  
MAXX  
MEDIAN  
MEDIANX  
MIN  
MINA  
MINX  
NORM.DIST  
NORM.INV  
NORM.S.DIST  
NORM.S.INV  
PERCENTILE.EXC  
PERCENTILE.INC  
PERCENTILEX.EXC



PERCENTILEX.INC

PERMUT

POISSON.DIST

RANK.EQ

RANKX

SAMPLE

SIN

SINH

STDEV.S

STDEV.P

STDEVX.S

STDEVX.P

SQRTPI

T.DIST

T.DIST.2T

T.DIST.RT

T.INV

T.INV.2T

TAN

TANH

VAR.S

VAR.P

VARX.S

VARX.P

Функции обработки таблиц

Обзор функций обработки таблиц

ADDCOLUMNS

ADDMISSINGITEMS

CROSSJOIN

CURRENTGROUP

DATATABLE

DETAILROWS

DISTINCT (столбец)

DISTINCT (таблица)

EXCEPT

FILTERS

GENERATE

GENERATEALL

GENERATESERIES

GROUPBY

IGNORE

INTERSECT

NATURALINNERJOIN

NATURALLEFTOUTERJOIN

ROLLUP

ROLLUPADDISUBTOTAL

ROLLUPGROUP

ROLLUPISUBTOTAL

ROW

SELECTCOLUMNS

SUBSTITUTEWITHINDEX

SUMMARIZE

SUMMARIZECOLUMNS

Конструктор таблиц

TOPN

TREATAS

UNION

VALUES

Текстовые функции

Общие сведения о текстовых функциях

COMBINEVALUES

CONCATENATE

CONCATENATEX

EXACT

FIND  
FIXED  
FORMAT  
LEFT  
LEN  
LOWER  
MID  
REPLACE  
REPT  
RIGHT  
SEARCH  
SUBSTITUTE  
TRIM  
UNICHAR  
UNICODE  
UPPER

Значение

## Функции операций со временем

Обзор функций операций со временем

CLOSINGBALANCEMONTH  
CLOSINGBALANCEQUARTER  
CLOSINGBALANCEYEAR  
DATEADD  
DATESBETWEEN  
DATESINPERIOD  
DATESMTD  
DATESQTD  
DATESYTD  
ENDOFMONTH  
ENDOFQUARTER  
ENDOFYEAR  
FIRSTDATE

FIRSTNONBLANK  
FIRSTNONBLANKVALUE  
LASTDATE  
LASTNONBLANK  
LASTNONBLANKVALUE  
NEXTDAY  
NEXTMONTH  
NEXTQUARTER  
NEXTYEAR  
OPENINGBALANCEMONTH  
OPENINGBALANCEQUARTER  
OPENINGBALANCEYEAR  
PARALLELPERIOD  
PREVIOUSDAY  
PREVIOUSMONTH  
PREVIOUSQUARTER  
PREVIOUSYEAR  
SAMEPERIODLASTYEAR  
STARTOFMONTH  
STARTOFQUARTER  
STARTOFYEAR  
TOTALMTD  
TOTALQTD  
TOTALYTD

## Инструкции DAX

[Общие сведения об инструкциях](#)

[DEFINE](#)

[EVALUATE](#)

[ORDER BY](#)

[VAR](#)

## Глоссарий по DAX

## Операторы DAX

Запросы DAX

Именованние параметров DAX

Синтаксис DAX

# Общие сведения о DAX

15.12.2020 • 56 minutes to read

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.

```
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:

```
= [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):

```
= 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](http://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,

```
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 сверху 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
<code>= TODAY()</code>	Вставляет текущую дату в каждую строку вычисляемого столбца.
<code>= 3</code>	Вставляет значение 3 в каждую строку вычисляемого столбца.
<code>= [Column1] + [Column2]</code>	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.

#### NOTE

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):

```
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.

### 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 разделе [Функции связей](#).

## Статистические функции

DAX предоставляет статистические функции, выполняющие статистическую обработку данных. Помимо вычисления сумм и средних значений, нахождения минимального и максимального значений, DAX также позволяет отфильтровать столбец, прежде чем выполнять статистическую обработку или создавать статистические выражения на основе связанных таблиц. 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:

```

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 вывода результата. При создании формулы DAX термины, 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 <sup>1, 2</sup>	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 <sup>63</sup> ) и 9,223,372,036,854,775,807 (2 <sup>63</sup> -1).
Десятичное число	64-разрядное (8 байтовое) вещественное число <sup>1, 2</sup>	<p>Вещественные числа — это числа, которые могут иметь знаки после запятой. Вещественные числа включают широкий диапазон значений.</p> <p>Отрицательные числа от -1.79E +308 до -2.23E -308</p> <p>Ноль</p> <p>Положительные числа от 2.23E -308 до 1.79E + 308</p> <p>Однако количество значащих цифр ограничено 17 знаками после запятой.</p>
Логический	Логический	Значение True или False.

ТИП ДАННЫХ В МОДЕЛИ	ТИП ДАННЫХ В DAX	ОПИСАНИЕ
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.  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 непредвиденные 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.

```
= [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.

```
= 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 удастся 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 сослаться по имени. Например, следующая формула показывает, как сослаться на столбцы из двух таблиц по *полному* имени:

```
= 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 вернуть 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 [USERRELATIONSHIP](#) 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.

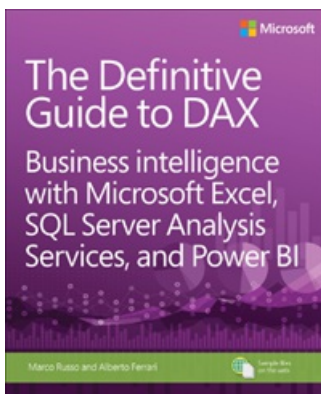
## Учебные материалы

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

[Использование DAX в Power BI Desktop](#) в Microsoft Learn.

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



# Видео

21.05.2020 • 2 minutes to read

Независимо от того, используете ли вы Power BI Desktop, Power Pivot в Excel или Analysis Services, для создания эффективных моделей данных необходимо освоить выражения анализа данных (DAX). Ниже приводится ряд видеороликов, которые помогут вам приступить к работе с этим эффективным языком выражений.

## Основы DAX

В этом видео партнер Майкрософт Альберто Феррари (Alberto Ferrari) рассказывает о важнейших понятиях DAX. На основе понятных практических примеров вы ознакомитесь с мерами, вычисляемыми столбцами и простейшими выражениями моделирования данных в DAX.

## DAX для опытных пользователей

В этом видео партнер Майкрософт Альберто Феррари (Alberto Ferrari) познакомит вас с теоретическими основами DAX, контекстом фильтра и строки, а также другими важными понятиями DAX.

# Справочник по функциям DAX

17.11.2020 • 4 minutes to read

В этом справочнике по функциям DAX представлены подробные сведения, включая синтаксис, параметры, возвращаемые значения и примеры, для каждой из более чем 250 функций, используемых в формулах DAX.

## IMPORTANT

Не все функции DAX поддерживаются или имеются в более ранних версиях Power BI Desktop, Analysis Services и Power Pivot в Excel.

## Содержимое раздела

**Новые функции DAX** — это новые или существенно переработанные функции.

**Функции даты и времени** — эти функции в DAX похожи на функции даты и времени в Microsoft Excel. Однако функции DAX основаны на типах данных `datetime`, используемых в Microsoft SQL Server.

**Функции фильтрации** — эти функции возвращают определенные типы данных, ищут значения в связанных таблицах и выполняют фильтрацию по связанным значениям. Функции уточняющих запросов работают с таблицами и связями между ними. Функции фильтрации позволяют управлять контекстом данных для создания динамических вычислений.

**Финансовые функции** — эти функции используются в формулах, которые выполняют финансовые вычисления, такие как чистая приведенная стоимость и норма прибыли.

**Информационные функции** — эти функции просматривают таблицу или столбец, предоставленные в качестве аргумента в другую функцию, и сообщают, соответствует ли значение ожидаемому типу. Например, функция `ISERROR` возвращает значение `TRUE`, если значение, на которое указывает ссылка, содержит ошибку.

**Логические функции** — эти функции возвращают сведения о значениях в выражении. Например, функция `TRUE` позволяет определить, возвращает ли выражение значение `TRUE`.

**Математические и тригонометрические функции** — математические функции в DAX похожи на математические и тригонометрические функции Excel. Однако числовые типы данных, используемые в функциях DAX, немного отличаются.

**Другие функции** — эти функции выполняют уникальные действия, и поэтому их нельзя отнести к другим категориям.

**Родительские и дочерние функции** — эти функции в DAX помогают пользователям управлять данными, представленными иерархией "родители-потомки", в своих моделях.

**Функции отношений** — эти функции предназначены для использования связей между таблицами и управления ими. Например, можно указать определенную связь для использования в вычислении.

**Статистические функции** — эти функции выполняют статистические вычисления. Помимо суммирования и определения среднего, минимального и максимального значений, в DAX можно фильтровать столбцы перед статистической обработкой или создавать статистические вычисления на основе связанных таблиц.

**Функции обработки таблиц** — эти функции возвращают таблицу или обрабатывают существующие таблицы.

[Функции работы с текстом](#) — эти функции позволяют возвращать часть строки, выполнять поиск текста в строке или сцеплять строковые значения. Есть также дополнительные функции для управления форматами дат, времени и чисел.

[Функции логики операций со временем](#) — эти функции позволяют создавать вычисления с использованием встроенных 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.

## См. также:

[Справочник по синтаксису DAX](#)

[Справочник по операторам DAX](#)

[Соглашения об именовании параметров DAX](#)

# Новые функции DAX

17.11.2020 • 2 minutes to read

DAX постоянно улучшается путем добавления новых функций и функциональных возможностей для поддержки новых возможностей. Новые функции и обновления включены в обновления служб, приложений и средств, которые в большинстве случаев публикуются каждый месяц.

Несмотря на постоянное обновление функций и функциональных возможностей, в документации описываются только те обновления, которые включают функциональные изменения, видимые для пользователей. Здесь показаны новые функции и обновления существующих функций за прошлый год.

## IMPORTANT

Не все функции поддерживаются во всех версиях Power BI Desktop, Analysis Services и Power Pivot в Excel. Новые и обновленные функции обычно сначала представляются в Power BI Desktop.

## Новые функции

КОМПОНЕНТ	МЕСЯЦ
<a href="#">ACCRINT</a>	Июль. 2020
<a href="#">ACCRINTM</a>	Июль. 2020
<a href="#">AMORDEGRC</a>	Июль. 2020
<a href="#">AMORLINC</a>	Июль. 2020
<a href="#">COUPDAYBS</a>	Июль. 2020
<a href="#">COUPDAYS</a>	Июль. 2020
<a href="#">COUPDAYSNC</a>	Июль. 2020
<a href="#">COUPNCD</a>	Июль. 2020
<a href="#">COUPNUM</a>	Июль. 2020
<a href="#">COUPPCD</a>	Июль. 2020
<a href="#">CUMIPMT</a>	Июль. 2020
<a href="#">CUMPRINC</a>	Июль. 2020
<a href="#">DB</a>	Июль. 2020
<a href="#">DDB</a>	Июль. 2020



КОМПОНЕНТ	МЕСЯЦ
DISC	Июль. 2020
DOLLARDE	Июль. 2020
DOLLARFR	Июль. 2020
DURATION	Июль. 2020
EFFECT	Июль. 2020
FV	Июль. 2020
INTRATE	Июль. 2020
IPMT	Июль. 2020
ISPMT	Июль. 2020
MDURATION	Июль. 2020
NOMINAL	Июль. 2020
NPER	Июль. 2020
ODDFPRICE	Июль. 2020
ODDFYIELD	Июль. 2020
ODDLPRICE	Июль. 2020
ODDLYIELD	Июль. 2020
PDURATION	Июль. 2020
PMT	Июль. 2020
PPMT	Июль. 2020
PRICE	Июль. 2020
PRICEDISC	Июль. 2020
PRICEMAT	Июль. 2020
PV	Июль. 2020
RATE	Июль. 2020
RECEIVED	Июль. 2020

КОМПОНЕНТ	МЕСЯЦ
RRI	Июль. 2020
SLN	Июль. 2020
SYD	Июль. 2020
TBILLEQ	Июль. 2020
TBILLPRICE	Июль. 2020
TBILLYIELD	Июль. 2020
VDB	Июль. 2020
YIELD	Июль. 2020
YIELDDISC	Июль. 2020
YIELDMAT	Июль. 2020
COALESCE	Март. 2020
FIRSTNONBLANKVALUE	Февраль 2020 г.
LASTNONBLANKVALUE	Февраль 2020 г.

## Обновленные функции

КОМПОНЕНТ	МЕСЯЦ
-----------	-------

# Функции даты и времени

02.10.2020 • 3 minutes to read

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

## В этой категории

КОМПОНЕНТ	DESCRIPTION
<a href="#">CALENDAR</a>	Возвращает таблицу с одним столбцом с именем Date, который содержит смежный набор дат.
<a href="#">CALENDARAUTO</a>	Возвращает таблицу с одним столбцом с именем Date, который содержит смежный набор дат.
<a href="#">DATE</a>	Возвращает указанную дату в формате datetime.
<a href="#">DATEDIFF</a>	Возвращает число границ интервала, пересекаемых между двумя датами.
<a href="#">DATEVALUE</a>	Преобразует дату в текстовом формате в дату в формате datetime.
<a href="#">DAY</a>	Возвращает день месяца: число от 1 до 31.
<a href="#">EDATE</a>	Возвращает дату, отстоящую на заданное количество месяцев от начальной даты.
<a href="#">EOMONTH</a>	Возвращает в формате datetime дату последнего дня месяца до или после указанного числа месяцев.
<a href="#">HOUR</a>	Возвращает час в виде числа от 0 до 23.
<a href="#">MINUTE</a>	Возвращает значение минуты в виде числа от 0 до 59 по заданному значению даты и времени.
<a href="#">MONTH</a>	Возвращает месяц в виде числа от 1 (январь) до 12 (декабрь).
<a href="#">NOW</a>	Возвращает текущую дату и время в формате datetime.
<a href="#">QUARTER</a>	Возвращает квартал в виде числа от 1 до 4.
<a href="#">SECOND</a>	Возвращает секунды значения времени в виде числа от 0 до 59.

КОМПОНЕНТ	DESCRIPTION
TIME	Преобразует заданные числовые значения часов, минут и секунд в значение времени в формате datetime.
TIMEVALUE	Преобразует время в текстовом формате во время в формате datetime.
TODAY	Возвращает текущую дату.
UTCNOW	Возвращает текущие дату и время в формате UTC
UTCTODAY	Возвращает текущую дату в формате UTC.
WEEKDAY	Возвращает число от 1 до 7, указывающее день недели для даты.
WEEKNUM	Возвращает номер недели для заданной даты и года в соответствии со значением return_type.
YEAR	Возвращает год для даты в виде четырехзначного целого числа в диапазоне 1900–9999.
YEARFRAC	Вычисляет часть года, представленную числом целых дней между двумя датами.

# CALENDAR

02.10.2020 • 2 minutes to read

Возвращает таблицу с одним столбцом с именем Date, который содержит смежный набор дат. Диапазон дат — от указанной даты начала до указанной конечной даты, включая эти две даты.

## Синтаксис

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

### Параметры

ТЕРМИН	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, который содержит смежный набор дат. Диапазон дат — от указанной даты начала до указанной конечной даты, включая эти две даты.

## Remarks

- Если start\_date больше end\_date, возвращается ошибка.
- 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 между 1 января 2005 г. и 31 декабря 2015 г.

```
= CALENDAR (DATE (2005, 1, 1), DATE (2015, 12, 31))
```

Для модели данных, которая включает фактические данные о продажах и прогнозы будущих продаж. Следующее выражение возвращает таблицу дат, охватывающую диапазон дат в этих двух таблицах.

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

# CALENDARAUTO

02.10.2020 • 2 minutes to read

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

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
fiscal_year_end_month	Любое выражение DAX, возвращающее целое число от 1 до 12. Если этот параметр опущен, по умолчанию используется значение, указанное в шаблоне таблицы календаря для текущего пользователя, если он есть. В противном случае значение по умолчанию — 12.

## Возвращаемое значение

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

## Remarks

- Диапазон дат вычисляется следующим образом.
  - Самая ранняя дата в модели, которая не находится в вычисляемом столбце или вычисляемой таблице, принимается как MinDate.
  - Самая поздняя 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 концом финансового года, связанного с MaxDate.
- Если модель не содержит значений datetime, не содержащихся в вычисляемых столбцах или вычисляемых таблицах, 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 этом примере в модели данных используются данные MinDate и MaxDate, равные 1 июля 2010 г. и 30 июня 2011 г. соответственно.

`CALENDARAUTO()` возвратит все даты между 1 января 2010 г. и 31 декабря 2011 г.

`CALENDARAUTO(3)` возвратит все даты между 1 марта 2010 г. и 28 февраля 2012 г.

# DATE

02.10.2020 • 6 minutes to read

Возвращает указанную дату в формате `datetime`.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
year	<p>Число, представляющее год.</p> <p>Значение аргумента <code>year</code> может содержать от одной до четырех цифр. Аргумент <code>year</code> интерпретируется в соответствии с системой дат, используемой компьютером.</p> <p>Поддерживаются даты, начинающиеся с 1 марта 1900 года.</p> <p>Если ввести число с десятичными знаками, число округляется.</p> <p>Для значений больше 9999 или меньше нуля (отрицательные значения) функция возвращает ошибку <code>#VALUE!</code>.</p> <p>Если значение <code>year</code> находится в диапазоне от 0 до 1899, то значение добавляется к 1900 для получения окончательного значения. См. следующие примеры.</p> <p><b>Примечание.</b> Для предотвращения нежелательных результатов следует использовать четыре цифры для аргумента <code>year</code>, если это возможно. Например, при использовании 07 в качестве значения года возвращается 1907.</p>
месяц	<p>Число, представляющее месяц или вычисление в соответствии со следующими правилами:</p> <p>Отрицательные целые числа не поддерживаются. Допустимые значения: 1–12.</p> <p>Если <code>month</code> является числом от 1 до 12, то оно представляет месяц года. 1 представляет январь, 2 — февраль и т. д. до 12, которое представляет декабрь.</p> <p>При вводе целого числа, превышающего 12, выполняются следующие вычисления: дата вычисляется путем сложения значения <code>month</code> со значением <code>year</code>. Например, если у вас есть дата DATE (2008, 18, 1), функция возвращает значение <code>datetime</code>, эквивалентное 1 июня 2009, поскольку 18 месяцев добавляются к началу 2008, и получается июнь 2009 года. Примеры см. ниже.</p>

ТЕРМИН	ОПРЕДЕЛЕНИЕ
day	<p>Число, представляющее день или вычисление в соответствии со следующими правилами:</p> <p>Отрицательные целые числа не поддерживаются. Допустимые значения: 1–31.</p> <p>Если <b>day</b> является числом от 1 до последнего дня в заданном месяце, то оно представляет день месяца.</p> <p>При вводе целого числа, превышающего последний день указанного месяца, выполняются следующие вычисления: дата вычисляется путем сложения значения <b>day</b> со значением <b>month</b>. Например, в формуле <code>DATE( 2008, 3, 32 )</code> функция DATE возвращает значение <b>datetime</b>, эквивалентное 1 апреля 2008 года, поскольку 32 дня добавляются к началу марта, что дает значение 1 апреля.</p> <p>Если <b>day</b> содержит десятичную часть, она округляется до ближайшего целого значения.</p>

## Возвращаемое значение

Возвращает указанную дату (**datetime**).

## Remarks

- Функция DATE принимает целые числа, которые являются входными аргументами, и формирует соответствующую дату. Функция DATE наиболее полезна в ситуациях, когда в формулах указаны год, месяц и день. Например, базовые данные могут содержать даты в формате, который не распознается как дата, например ГГГГММДД. Функцию DATE можно использовать вместе с другими функциями для преобразования дат в число, которое может быть распознано как дата.
- В отличие от Microsoft Excel, в котором даты хранятся в виде порядкового номера, функции даты DAX всегда возвращают тип данных **datetime**. Однако при необходимости можно использовать форматирование для вывода дат в виде порядковых номеров.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример: Возвращение простой даты

### Описание

Следующая формула возвращает дату 8 июля 2009 года:

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

## Пример: Годы до 1899 г.

Если значение, введенное для аргумента **year**, находится в диапазоне от 0 (ноль) до 1899 (включительно), это значение добавляется к 1900 для вычисления года. Следующая формула возвращает дату 2 января 1908 года: (1900 + 08).



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

## Пример: Годы после 1899 г.

Если **year** находится в диапазоне от 1900 до 9999 (включительно), это значение используется в качестве года. Следующая формула возвращает дату 2 января 2008 года:

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

## Пример: Работа с месяцами

Если **month** больше 12, **month** добавляет это число месяцев к первому месяцу в указанном году. Следующая формула возвращает дату 2 февраля 2009 года:

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

## Пример: Работа с днями

Если значение **day** больше, чем число дней в указанном месяце, **day** добавляет это число дней к первому дню месяца. Следующая формула возвращает дату 4 февраля 2008 года:

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

## См. также

[Функции даты и времени](#)

[DAY, функция](#)

[Функция TODAY](#)

# DATEDIFF

22.09.2020 • 2 minutes to read

Возвращает число границ интервала, пересекаемых между двумя датами.

## Синтаксис

```
DATEDIFF(<start_date>, <end_date>, <interval>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
start_date	Скалярное значение datetime.
end_date	Скалярное значение datetime Возвращаемое значение.
interval	Интервал, используемый при сравнении дат . Значение может быть одним из следующих: <ul style="list-style-type: none"><li>- SECOND</li><li>- MINUTE</li><li>- HOUR</li><li>- DAY</li><li>- WEEK</li><li>- MONTH</li><li>- QUARTER</li><li>- YEAR</li></ul>

## Возвращаемое значение

Число границ интервала, пересекаемых между двумя датами.

## Remarks

Если start\_date больше end\_date, возвращается ошибка.

## Пример

ДАТА
31.12.2012 23:59:59
01.01.2013 00:00:00

В следующих примерах возвращается значение 1:

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), SECOND )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), MINUTE )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), HOUR )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), DAY )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), WEEK )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), MONTH )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), QUARTER )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), YEAR )
```

# DATEVALUE

02.10.2020 • 2 minutes to read

Преобразует дату в текстовом формате в дату в формате datetime.

## Синтаксис

```
DATEVALUE(date_text)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
date_text	Текст, представляющий дату.

## Значение свойства или возвращаемое значение

Дата в формате datetime.

## Remarks

- При преобразовании функция DATEVALUE использует языковой стандарт, а также параметры даты и времени модели для определения значения даты. Если параметры даты и времени даты модели представляются в формате "День.Месяц.Год", то строка "08.01.2009" будет преобразована в значение типа datetime, эквивалентное 8 января 2009 года. Но если в соответствии с параметрами даты и времени даты представляются в формате "Месяц.День.Год", та же строка будет преобразована в значение типа datetime, эквивалентное 1 августа 2009 года.
- Если при преобразовании с использованием языкового стандарта, а также параметров даты и времени модели происходит сбой, функция DATEVALUE попытается использовать другие форматы даты. В таком случае некоторые строки, возможно, будут преобразованы с использованием одного формата, а другие строки — с использованием другого формата. Например, строка "05.04.2018", возможно, будет преобразована в "4 мая 2018 г.", а "20.04.2018" — в 20 апреля.
- Если компонент года в аргументе date\_text отсутствует, функция DATEVALUE использует текущий год из встроенных часов компьютера. Сведения о времени в аргументе date\_text не учитываются.
- При создании модели ее языковой стандарт, а также параметры даты и времени определяются на основе настроек приложения и компьютера.

## Пример

В приведенном ниже примере возвращаются разные значения datetime в зависимости от языкового стандарта и параметров модели, определяющих представление дат и времени.

- Если в соответствии с параметрами даты и времени день предшествует месяцу, в примере возвращается значение типа datetime, соответствующее 8 января 2009 г.
- Если в соответствии с параметрами даты и времени месяц предшествует дню, в примере возвращается значение типа datetime, соответствующее 1 августа 2009 г.

= DATEVALUE("8/1/2009")

См. также раздел

[Функции даты и времени](#)

# DAY

02.10.2020 • 3 minutes to read

Возвращает день месяца: число от 1 до 31.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
дата	Дата в формате <code>datetime</code> или текстовое представление даты.

## Возвращаемое значение

Целое число, указывающее день месяца.

## Примечания

- Функция DAY принимает в качестве аргумента дату дня, который необходимо найти. Даты можно предоставлять в функцию с помощью другой функции даты, с помощью выражения, возвращающего дату, или путем ввода даты в формате `datetime`. Можно также ввести дату в одном из допустимых строковых форматов дат.
- Значения, возвращаемые функциями YEAR, MONTH и DAY, имеют григорианский формат независимо от формата отображения предоставленных значений. Например, если дата предоставлена в формате отображения Хиджра, возвращаемые значения функций YEAR, MONTH и DAY будут связаны с соответствующей датой по григорианскому календарю.
- Если аргумент `date` является текстовым представлением даты, функция DAY использует языковой стандарт и параметры даты и времени клиентского компьютера, чтобы понять текстовое значение и выполнить преобразование. Если в соответствии с текущими параметрами даты и времени даты представляются в формате "День.Месяц.Год", то строка "08.01.2009" будет интерпретироваться как значение типа `datetime`, эквивалентное 8 января 2009 года, и функция возвращает значение 8. Однако если в соответствии с текущими параметрами даты и времени даты представляются в формате "Месяц.День.Год", та же строка будет интерпретироваться как значение типа `datetime`, эквивалентное 1 августа 2009 года, и функция возвращает значение 1.

## Пример: Получение дня из столбца дат

Приведенная ниже формула возвращает день из даты в столбце [Birthdate].

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

## Пример. Получение дня из даты в строковом формате

Приведенные ниже формулы возвращают день 4 из дат, которые были предоставлены в виде строк в допустимом текстовом формате.

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

## Пример. Получение значение дня в качестве условия

Приведенное ниже выражение возвращает день, в который был размещен каждый заказ на продажу, и помечает строку как акционный товар, если заказ был размещен в 10-й день месяца.

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

## См. также:

[Функции даты и времени](#)

[Функция TODAY](#)

[Функция DATE](#)

# EDATE

02.10.2020 • 3 minutes to read

Возвращает дату, отстоящую на заданное количество месяцев от начальной даты. Используйте EDATE для вычисления дат погашения или сроков выполнения, которые попадают в тот же день месяца, что и дата выпуска.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
start_date	Дата в формате <b>datetime</b> или <b>text</b> , представляющая дату начала.
months	Целое число месяцев до или после <b>start_date</b> .

## Возвращаемое значение

Дата (datetime).

## Примечания

- В отличие от Microsoft Excel, где даты хранятся в виде последовательных серийных номеров, DAX работает с датами в формате **datetime**. Даты, хранящиеся в других форматах, неявно преобразуются.
- Если **start\_date** не является допустимой датой, EDATE возвращает ошибку. Убедитесь, что ссылка на столбец или дата, указываемые в качестве первого аргумента, являются датой.
- Если значение **months** не целое, происходит усечение.
- Если аргумент **date** является текстовым представлением даты, функция EDATE использует языковой стандарт и параметры даты и времени клиентского компьютера, чтобы понять текстовое значение и выполнить преобразование. Если в соответствии с текущими параметрами даты и времени даты представляются в формате "День.Месяц.Год", то строка "08.01.2009" будет интерпретироваться как значение типа **datetime**, эквивалентное 8 января 2009 г. Однако если в соответствии с текущими параметрами даты и времени даты представляются в формате "Месяц.День.Год", та же строка будет преобразована в значение типа **datetime**, эквивалентное 1 августа 2009 года.
- Если запрошенная дата находится за последним днем соответствующего месяца, то возвращается последний день месяца. Например, следующие функции: EDATE ("2009-01-29", 1), EDATE ("2009-01-30", 1), EDATE ("2009-01-31", 1) возвращают 28 февраля 2009 г., что соответствует одному месяцу после начальной 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 возвращается дата через три месяца после даты заказа, которая хранится в столбце [TransactionDate].

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

## См. также

[Функция EOMONTH](#)

[Функции даты и времени](#)

# EOMONTH

02.10.2020 • 3 minutes to read

Возвращает в формате **datetime** дату последнего дня месяца до или после указанного числа месяцев. Используйте функцию EOMONTH для вычисления дат погашения или сроков выполнения, которые попадают на последний день месяца.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
start_date	Начальная дата в формате <b>datetime</b> или допустимое текстовое представление даты.
months	Число, представляющее количество месяцев до или после start_date. <b>Примечание.</b> Если введено число, которое не является целым, оно округляется в большую или меньшую сторону до ближайшего целого числа.

## Возвращаемое значение

Дата (datetime).

## Примечания

- В отличие от Microsoft Excel, где даты хранятся в виде последовательных серийных номеров, DAX работает с датами в формате **datetime**. Функция EOMONTH может принимать даты в других форматах с указанными ниже ограничениями.
- Если start\_date не является допустимой датой, EOMONTH возвращает ошибку.
- Если start\_date является числовым значением не в формате **datetime**, функция EOMONTH преобразует число в дату. Во избежание непредвиденных результатов преобразуйте число в формат **datetime** перед использованием функции EOMONTH.
- Если значение start\_date плюс months представляет собой недопустимую дату, функция EOMONTH возвращает ошибку. Даты до 1 марта 1900 г. и после 31 декабря 9999 г. являются недопустимыми.
- Если аргумент date является текстовым представлением даты, функция EDATE использует языковой стандарт и параметры даты и времени клиентского компьютера, чтобы распознать текстовое значение и выполнить преобразование. Если в соответствии с текущими параметрами даты и времени даты представляются в формате "День.Месяц.Год", то строка "08.01.2009" будет интерпретироваться как значение типа datetime, эквивалентное 8 января 2009 г. Однако если в соответствии с текущими параметрами даты и времени даты представляются в формате "Месяц.День.Год", та же строка будет преобразована в значение типа datetime, 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

Приведенное ниже выражение возвращает 31 мая 2008 г., так как аргумент `months` округляется до 2.

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

## См. также

[EDATE, функция](#)

[Функции даты и времени](#)

# HOURL

02.10.2020 • 2 minutes to read

Возвращает час в виде числа от 0 до 23.

## Синтаксис

```
HOURL(<datetime>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
DATETIME	Значение <b>datetime</b> , например 16:48:00 или 4:48 PM.

## Возвращаемое значение

Целое число от 0 до 23.

## Remarks

- Функция HOURL принимает в качестве аргумента время, содержащее час, который требуется найти. Время можно указать с помощью функции даты и времени, выражения, возвращающего значение **datetime**, либо путем ввода значения непосредственно в одном из поддерживаемых форматов времени. Время также можно указать в виде любого допустимого текстового представления времени.
- Если аргумент **datetime** является текстовым представлением даты и времени, функция использует языковой стандарт и параметры даты и времени клиентского компьютера, чтобы понять текстовое значение и выполнить преобразование. В большинстве языков в качестве разделителя времени используется двоеточие (:), и любой входной текст с двоеточиями в качестве разделителей времени будет анализироваться правильно. Проверьте региональные настройки, чтобы понять результаты.

## Пример

Следующий пример возвращает значение часа из столбца **TransactionTime** таблицы **Orders**.

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

## Пример

Следующий пример возвращает 15, то есть час, соответствующий 3 часам после полудня в 24-часовом формате. Текстовое значение автоматически анализируется и преобразуется в значение даты и времени.

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

См. также раздел

Функции даты и времени

Функция MINUTE

YEAR, функция

SECOND, функция

# MINUTE

02.10.2020 • 2 minutes to read

Возвращает значение минуты в виде числа от 0 до 59 по заданному значению даты и времени.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
DATETIME	Значение <b>datetime</b> или текст в приемлемом формате времени, например 16:48:00 или 4:48 PM.

## Возвращаемое значение

Целое число от 0 до 59.

## Remarks

- В отличие от Microsoft Excel, в котором даты и время хранятся в числовом формате, DAX использует тип данных **datetime** для дат и времени. Можно указать значение **datetime** для функции MINUTE, ссылаясь на столбец, в котором 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 языковой стандарт и параметры даты и времени клиентского компьютера, чтобы понять текстовое значение и выполнить преобразование. В большинстве языков в качестве разделителя времени используется двоеточие (:), и любой входной текст с двоеточиями в качестве разделителей времени будет анализироваться правильно. Проверьте региональные настройки, чтобы понять результаты.

## Пример 1

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

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

## Пример 2

В следующем примере возвращается значение 45, которое представляет собой количество минут во времени 1:45 PM.

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

См. также раздел

[Функции даты и времени](#)

[Функция HOUR](#)

[YEAR, функция](#)

[SECOND, функция](#)

# MONTH

02.10.2020 • 3 minutes to read

Возвращает месяц в виде числа от 1 (январь) до 12 (декабрь).

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Дата	Дата в <b>datetime</b> или текстовом формате.

## Возвращаемое значение

Целое число от 1 до 12.

## Remarks

- В отличие от Microsoft Excel, в котором даты хранятся в виде серийных номеров, в DAX при работе с датами используется формат **datetime**. Можно ввести дату, используемую в качестве аргумента функции MONTH, используя принятый для **datetime** формат, задав ссылку на столбец, содержащий даты, или выражение, возвращающее дату.
- Значения, возвращаемые функциями YEAR, MONTH и DAY, имеют григорианский формат независимо от формата отображения предоставленных значений. Например, если дата предоставлена в формате отображения Хиджра, возвращаемые значения функций YEAR, MONTH и DAY будут связаны с соответствующей датой по григорианскому календарю.
- Если аргумент **date** является текстовым представлением даты, функция использует языковой стандарт и параметры даты и времени клиентского компьютера, чтобы понять текстовое значение и выполнить преобразование. Если в соответствии с текущими параметрами даты и времени даты представляются в формате "День.Месяц.Год", то строка "08.01.2009" будет интерпретироваться как значение типа **datetime**, эквивалентное 8 января 2009 г, а функция вернет 1. Однако если в соответствии с текущими параметрами даты и времени дата представляется в формате "Месяц.День.Год", та же строка будет интерпретироваться как значение типа **datetime**, эквивалентное 1 августа 2009 года, и функция возвращает значение 8.
- Если текстовое представление даты не может быть правильно преобразовано в значение **datetime**, функция возвращает ошибку.

## Пример 1

Следующее выражение возвращает значение 3, которое представляет собой целое число, соответствующее марту в аргументе **date**.

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



## Пример 2

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

```
= MONTH(Orders[TransactionDate])
```

## См. также раздел

[Функции даты и времени](#)

[Функция HOUR](#)

[Функция MINUTE](#)

[YEAR, функция](#)

[SECOND, функция](#)

# NOW

02.10.2020 • 2 minutes to read

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 обновить это значение при каждом открытии листа.

## Синтаксис

```
NOW()
```

## Возвращаемое значение

Дата (**datetime**).

## Remarks

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

## Пример

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

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

## См. также

[Функция UTCNOW](#)

[Функция TODAY](#)

# QUARTER

22.09.2020 • 2 minutes to read

Возвращает квартал в виде числа от 1 (январь — март) до 4 (октябрь — декабрь).

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
дата	Дата.

## Возвращаемое значение

Целое число от 1 до 4.

## Примечания

Если входное значение BLANK, выходное значение также будет BLANK.

## Пример 1

Рассмотрим следующий запрос DAX:

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

Возвращает:

[ЗНАЧЕНИЕ]
1
4

## Пример 2

Рассмотрим следующий запрос DAX:

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

Возвращает:

FACTINTERNETSALES[ORDERDATE]	[QUARTER]
31.01.2008	1
01.04.2008	2

См. также

[YEAR](#)

[MONTH](#)

[DAY](#)

# SECOND

02.10.2020 • 3 minutes to read

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

## Синтаксис

```
SECOND(<time>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
time	Время в формате <b>datetime</b> , например 16:48:23 или 4:48:47 PM.

## Возвращаемое значение

Целое число от 0 до 59.

## Remarks

- В отличие от Microsoft Excel, в котором даты и время хранятся в виде серийных номеров, в DAX при работе с датами и временем используется формат **datetime**. Если исходные данные указаны в другом формате, DAX неявно преобразует данные. Можно использовать форматирование для вывода даты и времени в виде серийного номера.
- Значение даты и времени, указываемое в качестве аргумента второй функции, можно ввести в виде текстовой строки в кавычках (например, "6:45 PM"). Можно также указать значение времени в качестве результата другого выражения или как ссылку на столбец, содержащий время.
- Если указать числовое значение другого типа данных, например 13,60, то значение интерпретируется как серийный номер и представляется как тип данных **datetime**, прежде чем извлекать значение для секунд. Чтобы упростить понимание результатов, может потребоваться представить такие числа, как даты, прежде чем использовать их во второй функции. Например, при использовании функции **SECOND** со столбцом, содержащим числовое значение, например 25,56, формула возвращает значение 24. Это связано с тем, что при форматировании в виде даты значение 25,56 эквивалентно 25 января 1900 г., 1:26:24 PM.
- Если аргумент **time** является текстовым представлением даты и времени, функция использует языковой стандарт и параметры даты и времени клиентского компьютера, чтобы понять текстовое значение и выполнить преобразование. В большинстве языков в качестве разделителя времени используется двоеточие (:), и любой входной текст с двоеточиями в качестве разделителей времени будет анализироваться правильно. Проверьте региональные настройки, чтобы понять результаты.

## Пример

Следующая формула возвращает количество секунд во времени, которое содержится в столбце **TransactionTime** таблицы с именем **Orders**.

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

## Пример

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

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

## См. также раздел

[Функции даты и времени](#)

[HOUR](#)

[MINUTE](#)

[YEAR](#)

# TIME

02.10.2020 • 3 minutes to read

Преобразует заданные числовые значения часов, минут и секунд в значение времени в формате `datetime`.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
hour	Число от 0 до 23, представляющее часы.  Любое значение больше 23 будет делиться на 24, а остаток от деления рассматривается как значение часа.
minute	Число от 0 до 59, представляющее минуты.  Любое значение больше 59 преобразуется в часы и минуты.
second	Число от 0 до 59, представляющее секунды.  Любое значение больше 59 преобразуется в часы, минуты и секунды.

## Возвращаемое значение

Время (`datetime`).

## Remarks

- В отличие от Microsoft Excel, в котором значения даты и времени хранятся в виде порядковых чисел, в DAX при работе с датами и временем используется формат `datetime`. При использовании значения даты и времени в функции DAX числа в других форматах неявно преобразуются. Чтобы использовать порядковые числа, можно выполнить форматирование и изменить способ отображения чисел.
- Значения времени входят в состав значения даты и в формате порядкового числа представляются десятичным числом. Таким образом, значение `datetime`, определяющее время 12:00, будет эквивалентно значению 0,5, поскольку оно соответствует половине дня.
- Передавать аргументы в функцию `TIME` можно путем непосредственного ввода значений, в виде результата другого выражения, а также по ссылке на столбец, содержащий числовое значение. Применяются следующие ограничения:
  - Любое значение `hours` больше 23 будет делиться на 24, а остаток от деления рассматривается как значение часа.
  - Любое значение `minutes` больше 59 преобразуется в часы и минуты.
  - Любое значение `seconds` больше 59 преобразуется в часы, минуты и секунды.

- Если при преобразовании минут или секунд получается значение больше 24 часов, оно будет делиться на 24, а остаток от деления рассматривается как значение часа. При получении значения больше 24 часов определяющая дату часть не изменяется.
- Для большего удобства чтения значений времени, возвращаемых этой функцией, рекомендуется форматировать столбец или ячейку сводной таблицы, в которых содержится результат формулы, с использованием одного из форматов времени Microsoft Excel.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример 1

В обоих следующих примерах возвращается время 03:00:

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

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

## Пример 2

В обоих следующих примерах возвращается время 12:30:

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

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

## Пример 3

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

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

## См. также

[DATE](#)

[Функции даты и времени](#)



# TIMEVALUE

02.10.2020 • 2 minutes to read

Преобразует время в текстовом формате во время в формате datetime.

## Синтаксис

```
TIMEVALUE(time_text)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
time_text	Текстовая строка, которая представляет определенное время дня. Любая информация о дате, которую может содержать аргумент <code>time_text</code> , игнорируется.

## Возвращаемое значение

Дата (datetime).

## Remarks

- Значения времени входят в состав значения даты и представляются десятичным числом. Например, время 12:00 определяет середину дня и, соответственно, будет представлено значением 0,5.
- Если аргумент `time_text` является текстовым представлением даты и времени, функция использует языковой стандарт и параметры даты и времени модели, чтобы понять текстовое значение и выполнить преобразование. В большинстве языков в качестве разделителя времени используется двоеточие (:), и любой входной текст с двоеточиями в качестве разделителей времени будет анализироваться правильно. Проверьте региональные настройки, чтобы понять результаты.

## Пример

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

## См. также раздел

[Функции даты и времени](#)

# TODAY

02.10.2020 • 2 minutes to read

Возвращает текущую дату.

## Синтаксис

```
TODAY()
```

## Возвращаемое значение

Дата (datetime).

## Remarks

- Функция TODAY позволяет отображать на листе текущую дату независимо от того, когда была открыта книга. Кроме того, она применяется при вычислении интервалов.
- Если функция TODAY не обновляет дату в тех случаях, когда это ожидается, следует проверить параметры, определяющие порядок обновления столбца или книги.
- Функция NOW действует аналогично, однако возвращает точное время. Функция TODAY для всех дат возвращает время 12:00:00.

## Пример

Например, вы можете рассчитать возраст человека, родившегося в 1963 году, на момент его дня рождения в текущем году:

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

В этой формуле функция TODAY используется в качестве аргумента функции YEAR для получения текущего года, после чего из этого значения вычитается 1963, чтобы получить возраст человека.

## См. также раздел

[Функции даты и времени](#)

[NOW](#)

# UTCNOW

02.10.2020 • 2 minutes to read

Возвращает текущие дату и время в формате UTC.

## Синтаксис

```
UTCNOW()
```

## Возвращаемое значение

**Дата и время.**

## Remarks

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

## Пример

Формула:

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

Возвращает:

[ЗНАЧЕНИЕ]

2/2/2018 4:48:08 AM

## См. также раздел

[Функция NOW](#)

[Функция UTCTODAY](#)

# UTCTODAY

02.10.2020 • 2 minutes to read

Возвращает текущую дату в формате UTC.

## Синтаксис

```
UTCTODAY()
```

## Возвращаемое значение

Дата.

## Remarks

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

## Пример

Формула:

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

Возвращает:

[ЗНАЧЕНИЕ]

2/2/2018

## См. также раздел

[Функция NOW](#)

[Функция UTCNOW](#)

# WEEKDAY

02.10.2020 • 3 minutes to read

Возвращает число от 1 до 7, указывающее день недели для даты. По умолчанию дни отображаются в диапазоне от 1 (воскресенье) до 7 (суббота).

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
дата	Дата в формате <code>datetime</code> .  Даты должны вводиться с помощью функции <code>DATE</code> , с помощью выражений, которые приводят к дате, или в качестве результата других формул.
return_type	Число, определяющее возвращаемое значение:  Тип возвращаемого значения: 1, неделя начинается с воскресенья (1) и заканчивается в субботу (7). Нумерация с 1 до 7.  Тип возвращаемого значения: 2, неделя начинается с понедельника (1) и заканчивается в воскресенье (7).  Тип возвращаемого значения: 3, неделя начинается с понедельника (0) и заканчивается в субботу (6). Нумерация с 1 до 7.

## Возвращаемое значение

Целое число от 1 до 7.

## Примечания

- В отличие от Microsoft Excel, в котором значения даты хранятся в виде порядковых чисел, в DAX при работе со значениями даты и времени используется формат `datetime`. Если необходимо отобразить даты в виде серийных номеров, можно использовать параметры форматирования в Excel.
- Можно также ввести даты в виде допустимого текстового представления даты, но во избежание непредвиденных результатов лучше преобразовать текстовую дату в формат `datetime`.
- Если аргумент `date` является текстовым представлением даты, функция использует языковой стандарт и параметры даты и времени клиентского компьютера, чтобы понять текстовое значение и выполнить преобразование. Если в соответствии с текущими параметрами даты и времени даты представляются в формате "День.Месяц.Год", то строка "08.01.2009" будет интерпретироваться как значение типа `datetime`, эквивалентное 8 января 2009 года. Однако если в соответствии с текущими параметрами даты/времени даты представляются в формате "Месяц.День.Год", та же строка будет

преобразована в значение типа **datetime**, эквивалентное 1 августа 2009 года.

## Пример

В следующем примере показано получение даты из столбца [HireDate], добавление 1 и отображение дня недели, соответствующего этой дате. Поскольку аргумент **return\_type** опущен, используется формат по умолчанию, в котором 1 — воскресенье, а 7 — суббота. Если результат равен 4, то день будет иметь значение "Среда".

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

## См. также:

[Функции даты и времени](#)

[Функция WEEKNUM](#)

[YEARFRAC, функция](#)

# WEEKNUM

02.10.2020 • 2 minutes to read

Возвращает номер недели для заданной даты и года в соответствии со значением `return_type`. Номер недели указывает, что неделя попадает в число в пределах года.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
дата	Дата в формате <code>datetime</code> .
return_type	<p>Число, определяющее возвращаемое значение: если неделя начинается с воскресенья, используйте значение 1; если неделя начинается с понедельника, используйте значение 2. Значение по умолчанию — 1.</p> <p>Тип возвращаемого значения: 1, неделя начинается с воскресенья. Дни недели нумеруются с 1 до 7.</p> <p>Тип возвращаемого значения: 2, неделя начинается с понедельника. Дни недели нумеруются с 1 до 7.</p>

## Возвращаемое значение

Целое число.

## Примечания

- В отличие от Microsoft Excel, в котором даты хранятся в виде серийных номеров, в DAX при работе с датами и временем используется тип данных `datetime`. Если исходные данные имеют другой формат, DAX неявно преобразует данные в тип `datetime` для выполнения вычислений.
- По умолчанию функция `WEEKNUM` использует календарь, в котором неделя, содержащая 1 января, считается первой неделей года. Однако стандарт календаря ISO 8601, широко используемый в Европе, определяет первую неделю как содержащую большинство дней (четыре или более), которые выпадают в новый год. Это означает, что для года, в котором число дней в первой неделе января не превышает три дня, функция `WEEKNUM` возвращает номера недель, отличные от определения в стандарте ISO 8601.

## Пример

В следующем примере возвращается номер недели для даты 14 февраля 2010 г.

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

## Пример

В следующем примере возвращается номер недели для даты, хранящейся в столбце **HireDate** из таблицы **Employees**.

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

## См. также:

[Функции даты и времени](#)

[YEARFRAC, функция](#)

[Функция WEEKDAY](#)



# YEAR

02.10.2020 • 2 minutes to read

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

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Дата	Дата в <b>datetime</b> или в текстовом формате, содержащая год, который требуется найти.

## Возвращаемое значение

Целое число в диапазоне 1900–9999.

## Remarks

- В отличие от Microsoft Excel, в котором даты хранятся в виде серийных номеров, в DAX при работе с датами и временем используется тип данных **datetime**.
- Даты должны вводиться с помощью функции DATE или в качестве результатов других формул или функций. Можно также вводить даты через допустимые текстовые представления даты, например "March 3, 2007" или "Mar-3-2003".
- Значения, возвращаемые функциями YEAR, MONTH и DAY, имеют григорианский формат независимо от формата отображения предоставленных значений. Например, если дата предоставлена в формате отображения для календаря Хиджры, возвращаемые значения функций YEAR, MONTH и DAY будут связаны с соответствующей датой по григорианскому календарю.
- Если аргумент date является текстовым представлением даты, функция использует языковой стандарт и параметры даты и времени клиентского компьютера, чтобы понять текстовое значение и выполнить преобразование. Если формат строк несовместим с текущими параметрами языкового стандарта, могут возникнуть ошибки. Например, если языковой стандарт задает формат даты "месяц/день/год", а дата указывается как "день/месяц/год", то 25/1/2009 будет интерпретироваться не как 25 января 2009, а как недопустимая дата.

## Пример

В следующем примере будет возвращено число 2007.

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

## Пример — дата как результат выражения

## Description

В следующем примере возвращается год для сегодняшней даты.

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

## См. также раздел

[Функции даты и времени](#)

[Функция HOUR](#)

[Функция MINUTE](#)

[YEAR, функция](#)

[SECOND, функция](#)

# YEARFRAC

02.10.2020 • 2 minutes to read

Вычисляет часть года, представленную числом целых дней между двумя датами. Используйте функцию листа YEARFRAC, чтобы узнать долю преимуществ или обязательств по всему году для назначения определенного срока.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
start_date	Дата начала в формате <b>datetime</b> .
end_date	Дата окончания в формате <b>datetime</b> .
basis	<p>(необязательно) Используемый тип для подсчета дней. Все аргументы усекаются до целых значений.</p> <p>Основание — описание</p> <p>0 — US (NASD) 30/360</p> <p>1 — Факт/факт</p> <p>2 — Факт/360</p> <p>3 — Факт/365</p> <p>4 — European 30/360</p>

## Возвращаемое значение

Десятичное число. Внутренний тип данных — это 64-разрядное (8-байтовое) число с плавающей запятой двойной точности со знаком.

## Примечания

- В отличие от Microsoft Excel, в котором даты хранятся в виде серийных номеров, в DAX при работе с датами и временем используется формат **datetime**. Если необходимо просмотреть даты в виде серийных номеров, можно использовать параметры форматирования в Excel.
- Если **start\_date** или **end\_date** не являются допустимыми датами, YEARFRAC возвращает ошибку.
- Если **basis** < 0 или **basis** > 4, функция YEARFRAC возвращает ошибку.

## Пример 1

В следующем примере возвращается доля года, представленная разностью между датами в двух столбцах

TransactionDate и ShippingDate :

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

## Пример 2

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

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

Используйте четырехзначные значения года, когда это возможно, чтобы избежать получения непредвиденных результатов. При усечении года предполагается текущий год. Если дата указана или опущена, то предполагается первая дата месяца.

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

## См. также:

[Функции даты и времени](#)

[Функция WEEKNUM](#)

[YEARFRAC, функция](#)

[Функция WEEKDAY](#)

# Функции фильтров

02.10.2020 • 2 minutes to read

Функции фильтрации и значений в DAX являются самыми сложными и мощными и значительно отличаются от функций Excel. Функции поиска работают с использованием таблиц и связей, как в базе данных. Функции фильтрации дают возможность управлять контекстом данных для создания динамических вычислений.

## В этой категории

КОМПОНЕНТ	DESCRIPTION
<a href="#">ALL</a>	Возвращает все строки в таблице или все значения в столбце, не учитывая применяемые фильтры.
<a href="#">ALLCROSSFILTERED</a>	Очищает все фильтры, которые применяются к таблице.
<a href="#">ALLEXCEPT</a>	Удаляет все фильтры контекста в таблице, за исключением фильтров, примененных к указанным столбцам.
<a href="#">ALLNOBLANKROW</a>	Возвращает из родительской таблицы связи все строки, кроме пустых, или все уникальные значения столбца, кроме пустых, и не учитывает возможные фильтры контекста.
<a href="#">ALLSELECTED</a>	Удаляет фильтры контекста из столбцов и строк в текущем запросе, сохраняя все остальные фильтры контекста или явные фильтры.
<a href="#">CALCULATE</a>	Вычисляет выражение в измененном контексте фильтра.
<a href="#">CALCULATETABLE</a>	Вычисляет табличное выражение в измененном контексте фильтра.
<a href="#">EARLIER</a>	Возвращает текущее значение указанного столбца на проходе внешней оценки упомянутого столбца.
<a href="#">EARLIEST</a>	Возвращает текущее значение указанного столбца на проходе внешней оценки указанного столбца.
<a href="#">FILTER</a>	Возвращает таблицу, представляющую подмножество другой таблицы или выражения.
<a href="#">KEEPFILTERS</a>	Изменяет порядок применения фильтров при вычислении функции CALCULATE или CALCULATETABLE.
<a href="#">LOOKUPVALUE</a>	Возвращает значение для строки, удовлетворяющей всем критериям, указанным в условиях поиска. Функция может применять одно или несколько условий поиска.

КОМПОНЕНТ	DESCRIPTION
REMOVEFILTERS	Очищает фильтры в указанных таблицах или столбцах.
SELECTEDVALUE	Возвращает значение, если контекст для columnName был отфильтрован только по одному отдельному значению. В противном случае возвращает alternateResult.

# ALL

02.10.2020 • 13 minutes to read

Возвращает все строки в таблице или все значения в столбце, не учитывая применяемые фильтры. Эта функция удобна для очистки фильтров и создания вычислений по всем строкам в таблице.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
таблица	Таблица, в которой нужно очистить фильтры.
гистограмма	Столбец, в котором нужно очистить фильтры.

Аргумент функции ALL должен быть либо ссылкой на базовую таблицу, либо ссылкой на базовый столбец. Функция ALL не поддерживает табличные выражения или выражения столбцов.

## Возвращаемое значение

Таблица или столбец с удаленными фильтрами.

## Примечания

- Эта функция не используется сама по себе, а служит в качестве промежуточной функции, которую можно использовать для изменения набора результатов, по которым выполняется какое-либо другое вычисление.
- Нормальным поведением выражений DAX, содержащих функцию ALL(), является то, что все применяемые фильтры будут игнорироваться. Однако существуют ситуации, в которых это не так, поскольку существует технология DAX *Auto-Exist*, которая оптимизирует фильтрацию, чтобы уменьшить объем обработки, необходимый для определенных запросов DAX. Приведем пример, в котором Auto-Exist и ALL() предоставляют непредвиденные результаты: при фильтрации двух или более столбцов одной таблицы (например, в случае использования срезов); при этом существует мера в той же таблице, в которой используется ALL(). В этом случае Auto-Exist **выполнит слияние** нескольких фильтров в один и будет фильтровать только существующие сочетания значений. Из-за этого слияния мера будет рассчитываться на основе существующих сочетаний значений, а результат будет основываться на отфильтрованных значениях, а не на всех значениях, как ожидалось. Дополнительные сведения о технологии Auto-Exist и ее влиянии на вычисления см. в статье Microsoft MVP Альберто Феррари [Общие сведения о DAX Auto-Exist](https://sqlbi.com/articles/2017/05/23/dax-auto-exist/) по адресу [sql.bi.com](https://sqlbi.com).
- В следующей таблице описано, как можно применять функции ALL и ALLEXCEPT в различных сценариях.

ФУНКЦИЯ И НАЗНАЧЕНИЕ	ОПИСАНИЕ
ALL()	Удаляет все фильтры везде. ALL() можно использовать только для очистки фильтров, но не для возвращения таблицы.
ALL(Таблица)	Удаляет все фильтры из указанной таблицы. Фактически ALL(Таблица) возвращает все значения в таблице, удаляя все фильтры из контекста, которые могли быть применены. Эта функция полезна в случае, когда вы работаете со множеством уровней группирования и требуется создать вычисление, которое создает отношение агрегированного значения к общему значению. Этот сценарий демонстрируется в первом примере.
ALL (Column[, Column[, ...]])	Удаляет все фильтры из указанных столбцов в таблице. Все остальные фильтры в других столбцах таблицы по-прежнему применяются. Все аргументы столбца должны поступать из одной и той же таблицы. Вариант ALL(Column) полезен, если требуется удалить фильтры контекста для одного или нескольких конкретных столбцов и для сохранения всех остальных фильтров контекста. Этот сценарий демонстрируется во втором и третьем примерах.
ALLEXCEPT(Table, Column1 [,Column2]...)	Удаляет все фильтры контекста в таблице, за исключением фильтров, примененных к указанным столбцам. Это удобно в ситуациях, когда необходимо удалить фильтры для многих, но не всех столбцов 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 отношения продаж по категории к общему объему продаж

Предположим, что требуется определить объем продаж для текущей ячейки в сводной таблице, деленный на общий объем продаж для всех торговых посредников. Чтобы обеспечить неизменность знаменателя независимо от того, как пользователь сводной таблицы может фильтровать или группировать данные, вы определяете формулу, использующую ALL для создания правильного общего итога.

В следующей таблице показаны результаты создания меры **All Reseller Sales Ratio** (Соотношение продаж всех торговых посредников) с использованием формулы, приведенной в разделе кода. Чтобы увидеть, как это работает, добавьте поле CalendarYear в область **Метки строк** сводной таблицы и поле ProductCategoryName в область **Метки столбцов**. Затем перетащите меру **All Reseller Sales Ratio** (Соотношение продаж всех торговых посредников) в область **Значения** сводной таблицы. Для просмотра результатов в виде процентных отношений используйте функции форматирования Excel, чтобы применить форматирование значений в процентах к ячейкам, содержащим данную меру.



ТОВАРООБОРОТ ВСЕХ ПОСРЕДНИКОВ	МЕТКИ СТОЛБЦОВ				
Метки строк	Принадлежность и	Велосипеды	Экипировка	Запчасти	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 %
Общий итог	0,70 %	82,47 %	2,18 %	14,65 %	100,00 %

## Формула

```
= 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])` является суммой значений в `ResellerSales_USD[SalesAmount_USD]` для текущей ячейки сводной таблицы с фильтрами контекста, применяемыми к `CalendarYear` и `ProductCategoryName`.
2. Для знаменателя сначала нужно указать таблицу, `ResellerSales_USD`, и использовать функцию `ALL` для удаления всех фильтров контекста в таблице.
3. Затем можно использовать функцию `SUMX` для суммирования значений в столбце `ResellerSales_USD[SalesAmount_USD]`. Иными словами, вы получаете сумму `ResellerSales_USD[SalesAmount_USD]` для продаж всех торговых посредников.

## Пример 2

Вычисление отношения продаж продуктов к общему объему продаж за текущий год

Предположим, что нужно создать таблицу, показывающую процент продаж со сравнением по годам для каждой категории продуктов (`ProductCategoryName`). Чтобы получить процент для каждого года по каждому значению `ProductCategoryName`, нужно разделить сумму продаж для этого конкретного года и категории продуктов на сумму продаж для той же категории продуктов за все годы. Иными словами, нужно сохранить фильтр по `ProductCategoryName` и удалить фильтр по году при вычислении знаменателя процентного отношения.

В следующей таблице показаны результаты создания меры **Продажи торгового посредника за год** с использованием формулы, приведенной в разделе кода. Чтобы увидеть, как это работает, добавьте поле `CalendarYear` в область **Метки строк** сводной таблицы и поле `ProductCategoryName` в область **Метки столбцов**. Для просмотра результатов в виде процентных отношений используйте функции форматирования Excel, чтобы применить форматирование значений в процентах к ячейкам, содержащим данную меру **Продажи торгового посредника за год**.

ПРОДАЖИ ТОРГОВОГО ПОСРЕДНИКА ЗА ГОД	МЕТКИ СТОЛБЦОВ				
Метки строк	Принадлежност и	Велосипеды	Экипировка	Запчасти	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 %

## Формула

```
= 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])` является суммой значений в `ResellerSales_USD[SalesAmount_USD]` для текущей ячейки сводной таблицы с фильтрами контекста, применяемыми к столбцам `CalendarYear` и `ProductCategoryName`.
2. Для знаменателя вы удаляете существующий фильтр по `CalendarYear` с помощью функции `ALL(Column)`. При этом вычисляется сумма по оставшимся строкам в таблице `ResellerSales_USD` после применения существующих фильтров контекста из меток столбцов. Результатом является то, что сумма для знаменателя вычисляется по выбранной `ProductCategoryName` (неявный фильтр контекста) и для всех значений в году.

## Пример 3

Вычисление вклада категорий продуктов в общий объем продаж за год

Предположим, что нужно создать таблицу, показывающую процент продаж для каждой категории продуктов по годам. Чтобы получить процент для каждой категории продуктов за определенный год, нужно вычислить сумму продаж для конкретной категории продуктов (`ProductCategoryName`) за год *n*, а затем разделить полученное значение на сумму продаж за год *n* по всем категориям продуктов. Иными словами, нужно сохранить фильтр по году и удалить фильтр по `ProductCategoryName` при вычислении знаменателя процентного отношения.

В следующей таблице показаны результаты создания меры **Продажи торгового посредника по категории продуктов** с использованием формулы, приведенной в разделе кода. Чтобы увидеть, как это работает, добавьте поле `CalendarYear` в область **Метки строк** сводной таблицы и поле `ProductCategoryName` в область **Метки столбцов**. Затем добавьте новую меру в область **Значения** сводной таблицы. Для просмотра результатов в виде процентных отношений используйте функции форматирования Excel, чтобы применить форматирование значений в процентах к ячейкам, содержащим новую меру **Продажи торгового посредника по категории продуктов**.

ПРОДАЖИ ТОРГОВОГО ПОСРЕДНИКА ПО КАТЕГОРИИ ПРОДУКТОВ	МЕТКИ СТОЛБЦОВ				
Метки строк	Принадлежност и	Велосипеды	Экипировка	Запчасти	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 %

### Формула

```
= 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])` является суммой значений в `ResellerSales_USD[SalesAmount_USD]` для текущей ячейки сводной таблицы с фильтрами контекста, применяемыми к полям `CalendarYear` и `ProductCategoryName`.
2. Для знаменателя вы используете функцию `ALL(Column)`, чтобы удалить фильтр по `ProductCategoryName` и вычислить сумму по оставшимся строкам в таблице `ResellerSales_USD` после применения существующих фильтров контекста из меток строк. Результатом является то, что сумма для знаменателя вычисляется за выбранный год (неявный фильтр контекста) и для всех значений `ProductCategoryName`.

### См. также:

[Функции фильтрации](#)

[Функция ALL](#)

[Функция ALLEXCEPT](#)

[Функция FILTER](#)

# ALLCROSSFILTERED

22.09.2020 • 2 minutes to read

Очищает все фильтры, которые применяются к таблице.

## Синтаксис

```
ALLCROSSFILTERED(<table>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, в которой нужно очистить фильтры.

## Возвращаемое значение

Недоступно См. примечания.

## Remarks

- ALLCROSSFILTERED можно использовать только для очистки фильтров, но не для возвращения таблицы.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

```
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:

DIMSALESREASON[SALESREASONNAME]	[TOTALQUANTITY1]	[TOTALQUANTITY2]
Демонстрация	60398	
Реклама в журнале	60398	

DIMSALESREASON[SALESREASONNAME]	[TOTALQUANTITY1]	[TOTALQUANTITY2]
Производитель	60398	1818
Реклама	60398	7390
Другие	60398	3653
Price	60398	47733
Качество	60398	1551
Просмотр	60398	1640
Спонсорское предложение	60398	
Реклама по ТВ	60398	730

**NOTE**

Между таблицей FactInternetSales и таблицей DimSalesReason существует прямая или косвенная связь "многие ко многим".

# ALLEXCEPT

02.10.2020 • 4 minutes to read

Удаляет все фильтры контекста в таблице, за исключением фильтров, примененных к указанным столбцам.

## Синтаксис

```
ALLEXCEPT(<table>,<column>[,<column>[,...]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, в которой удаляются все фильтры контекста, за исключением фильтров для этих столбцов, указанных в последующих аргументах.
столбец	Столбец, для которого следует сохранить фильтры контекста.

Первым аргументом функции ALLEXCEPT должна быть ссылка на базовую таблицу; все последующие аргументы должны быть ссылками на базовые столбцы. Функция ALLEXCEPT не поддерживает табличные выражения или выражения столбцов.

## Возвращаемое значение

Таблица с удаленными фильтрами, за исключением фильтров по указанным столбцам.

## Remarks

- Эта функция не используется сама по себе, а служит в качестве промежуточной функции, которую можно использовать для изменения набора результатов, по которым выполняется какое-либо другое вычисление.
- Как описано в следующей таблице, функции ALL и ALLEXCEPT можно применять в различных сценариях.

ФУНКЦИЯ И НАЗНАЧЕНИЕ	DESCRIPTION
ALL(Таблица)	Удаляет все фильтры из указанной таблицы. Фактически ALL(Таблица) возвращает все значения в таблице, удаляя все фильтры из контекста, которые могли быть применены. Эта функция полезна в случае, когда вы работаете со множеством уровней группирования и требуется создать вычисление, которое создает отношение агрегированного значения к общему значению.

ФУНКЦИЯ И НАЗНАЧЕНИЕ	DESCRIPTION
ALL (Column[, Column[, ...]])	Удаляет все фильтры из указанных столбцов в таблице. Все остальные фильтры в других столбцах таблицы по-прежнему применяются. Все аргументы столбца должны поступать из одной и той же таблицы. Вариант ALL(Column) полезен, если требуется удалить фильтры контекста для одного или нескольких конкретных столбцов и для сохранения всех остальных фильтров контекста.
ALLEXCEPT(Table, Column1 [,Column2]...)	Удаляет все фильтры контекста в таблице, за исключением фильтров, примененных к указанным столбцам. Это удобно в ситуациях, когда необходимо удалить фильтры для многих, но не всех столбцов в таблице.

- 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 представлена формула, которую можно использовать в мере.

Формула суммирует SalesAmount\_USD и использует функцию ALLEXCEPT для удаления фильтров контекста в таблице DateTime, за исключением случаев, когда к столбцу CalendarYear применен фильтр.

```
= CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]), ALLEXCEPT(DateTime, DateTime[CalendarYear]))
```

Поскольку формула использует ALLEXCEPT, всякий раз, когда для среза сводной таблицы используется любой столбец из таблицы DateTime, кроме столбца CalendarYear, формула удалит фильтры среза, предоставляющие значение, равное сумме SalesAmount\_USD, для значения метки столбца, как показано в таблице 1.

Однако если для среза сводной таблицы используется столбец CalendarYear, результаты будут разными. Поскольку CalendarYear указывается в качестве аргумента для ALLEXCEPT, то при срезе данных по году фильтр будет применяться к годам на уровне строк, как показано в таблице 2. Пользователю рекомендуется сравнить эти таблицы, чтобы понять поведение ALLEXCEPT().

## См. также раздел

[Функции фильтрации](#)

[Функция ALL](#)

[Функция FILTER](#)

# ALLNOBLANKROW

02.10.2020 • 8 minutes to read

Возвращает из родительской таблицы связи все строки, кроме пустых, или все уникальные значения столбца, кроме пустых, и не учитывает возможные фильтры контекста.

## Синтаксис

```
ALLNOBLANKROW( {<table> | <column>[, <column>[, <column>[,...]]]} )
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
таблица	Таблица, в которой удаляются все фильтры контекста.
гистограмма	Столбец, в котором удаляются все фильтры контекста.

Необходимо передать только один параметр. Им может быть таблица или столбец.

## Возвращаемое значение

Таблица, если переданный параметр был таблицей, или столбец значений, если переданный параметр был столбцом.

## Примечания

- Функция ALLNOBLANKROW фильтрует только пустую строку, которая отображается в родительской таблице связи, если в дочерней таблице есть одна или несколько строк со значениями, несовпадающими со значениями в родительском столбце. Подробное описание см. в приведенном ниже примере.
- В приведенной ниже таблице представлены все разновидности функции ALL, имеющиеся в DAX, и их отличия.

ФУНКЦИЯ И НАЗНАЧЕНИЕ	ОПИСАНИЕ
ALL(Столбец)	Удаляет все фильтры из указанного столбца в таблице. Все остальные фильтры в других столбцах таблицы по-прежнему применяются.
ALL(Таблица)	Удаляет все фильтры из указанной таблицы.
ALLEXCEPT(Таблица,Столбец1,Столбец2...)	Переопределяет все фильтры контекста в таблице, за исключением фильтров, примененных к указанным столбцам.



ФУНКЦИЯ И НАЗНАЧЕНИЕ	ОПИСАНИЕ
ALLNOBLANK(таблица столбец)	Возвращает из родительской таблицы связи все строки, кроме пустых, или все уникальные значения столбца, кроме пустых, и не учитывает возможные фильтры контекста.

Общее описание того, как работает функция ALL, а также пошаговые примеры использования функций ALL(Таблица) и ALL(Столбец), см. в статье [Функция ALL](#).

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

## Пример

В образце данных таблица ResellerSales\_USD содержит одну строку, которая не имеет значений и поэтому не может быть связана ни с одной из родительских таблиц связей в книге. Эта таблица будет использоваться в сводной таблице, чтобы продемонстрировать поведение пустых строк и выполнение подсчета для несвязанных данных.

### Шаг 1. Проверка несвязанных данных

Откройте **окно Power Pivot** и выберите таблицу ResellerSales\_USD. В столбце ProductKey отфильтруйте пустые значения. Останется одна строка. В ней должны быть пустыми значения во всех столбцах, кроме SalesOrderLineNumber.

### Шаг 2. Создание сводной таблицы

Создайте сводную таблицу и перетащите столбец datetime.[Calendar Year] в область "Метки строк". В приведенной ниже таблице показаны ожидаемые результаты.

МЕТКИ СТРОК
2005
2006
2007
2008
Общий итог

Обратите внимание на пустую метку между 2008 и Grand Total. Она представляет неизвестный элемент, то есть специальную группу, создаваемую для учета всех значений в дочерней таблице, у которых нет соответствующих значений в родительской таблице. В этом примере это столбец datetime.[Calendar Year].

Если в сводной таблице есть пустая метка, это означает, что в некоторых таблицах, связанных со столбцом datetime.[Calendar Year], есть пустые или несовпадающие значения. Пустая метка отображается в родительской таблице, но несовпадающие строки находятся в одной или нескольких дочерних таблицах.

Строки, которые добавляются в эту группу пустой метки, — это либо значения, которые не соответствуют ни одному из значений в родительской таблице (например, дата, которой нет в таблице datetime), либо значения NULL, то есть отсутствующие значения дат. В этом примере мы поместили пустое значение во все столбцы дочерней таблицы Sales. То, что в родительской таблице больше значений, чем в дочерних, не

вызывает проблем.

### Шаг 3. Подсчет строк с помощью функций ALL и ALLNOBLANK

Добавьте в таблицу `datetime` следующие две меры, чтобы подсчитать строки таблицы: **Countrows ALLNOBLANK of datetime**, **Countrows ALL of datetime**. Определить эти меры можно с помощью следующих формул:

```
// 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 столбец `datetime.[Calendar Year]` в область "Метки строк", а затем добавьте созданные меры. Результатом должна быть таблица наподобие следующей:

МЕТКИ СТРОК	COUNTROWS ALLNOBLANK OF DATETIME	COUNTROWS ALL OF DATETIME
2005	1280	1281
2006	1280	1281
2007	1280	1281
2008	1280	1281
	1280	1281
Общий итог	1280	1281

Количество строк в результатах различается на единицу. Однако если вы откроете **окно Power Pivot** и выберете таблицу `datetime`, то не найдете в ней пустые строки, так как указанная здесь специальная пустая строка — это неизвестный элемент.

### Шаг 4. Проверка точности подсчета

Чтобы убедиться в том, что функция `ALLNOBLANKROW` не подсчитывает действительно пустые строки и учитывает только специальную пустую строку в родительской таблице, добавьте следующие две меры в таблицу `ResellerSales_USD`: **Countrows ALLNOBLANKROW of ResellerSales\_USD**, **Countrows ALL of ResellerSales\_USD**.

Создайте сводную таблицу и перетащите столбец `datetime.[Calendar Year]` в область "Метки строк". Затем добавьте только что созданные меры. Результаты должны выглядеть следующим образом:

МЕТКИ СТРОК	COUNTROWS ALLNOBLANKROW OF RESELLERSALES_USD	COUNTROWS ALL OF RESELLERSALES_USD
2005	60856	60856

МЕТКИ СТРОК	COUNTROWS ALLNOBLANKROW OF RESELLERSALES_USD	COUNTROWS ALL OF RESELLERSALES_USD
2006	60856	60856
2007	60856	60856
2008	60856	60856
	60856	60856
Общий итог	60856	60856

Теперь меры дают одинаковые результаты. Связано это с тем, что функция ALLNOBLANKROW не подсчитывает действительно пустые строки в таблице, а учитывает только специальную пустую строку, которая создается в родительской таблице, если одна или несколько дочерних таблиц связи содержат несовпадающие или пустые значения.

## См. также:

[Функции фильтрации](#)

[Функция ALL](#)

[Функция FILTER](#)

# ALLSELECTED

22.09.2020 • 7 minutes to read

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. Ее можно использовать для получения визуальных итогов в запросах.

## Синтаксис

```
ALLSELECTED([<tableName> | <columnName>[, <columnName>[, <columnName>[,...]]]] )
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
tableName	Имя существующей таблицы с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением. Это необязательный параметр.
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX (обычно полное имя). Этот параметр не может быть выражением. Это необязательный параметр.

## Возвращаемое значение

Контекст запроса без фильтров столбцов и строк.

## Remarks

- Если имеется один аргумент, аргументом будет либо *tableName*, либо *columnName*. Если имеется несколько аргументов, они должны быть столбцами из одной и той же таблицы.
- Эта функция отличается от функции ALL(), поскольку она содержит все фильтры, явно заданные в запросе, и все фильтры контекста, кроме 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 создавать различные уровни визуальных итогов в табличном отчете с помощью выражений DAX. В отчете два (2) предыдущих фильтра были применены к данным товарооборота посредников; один по группе территории продаж — *Europe*, а другой — по типу рекламной акции (*Volume Discount*). После применения фильтров визуальные итоги можно вычислить для всего отчета, для всех лет или для всех категорий продуктов. Кроме того, для наглядности получен общий итог по всему товарообороту посредника с удалением всех фильтров в отчете. Вычисление следующего выражения DAX приводит к созданию таблицы со всеми сведениями, необходимыми для построения таблицы с визуальными

итогами.

```
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 получаете следующие результаты:

[ИМЯ КАТЕГОРИИ ПРОДУКТОВ]	[КАЛЕНДАРН ЫЙ ГОД]	[ОБЪЕМ ТОВАРОБОР ОТА ПОСРЕДНИКО В]	[ОБЩИЙ ИТОГ ПОСРЕДНИКА ]	[ВИЗУАЛЬНЫ Й ИТОГ ПОСРЕДНИКА ]	[ВИЗУАЛЬНЫ Й ИТОГ ПОСРЕДНИКА ЗА ВЕСЬ КАЛЕНДАРНЫ Й ГОД]	[ВИЗУАЛЬНЫ Й ИТОГ ПОСРЕДНИКА ДЛЯ ВСЕХ ИМЕН В КАТЕГОРИИ ПРОДУКТОВ]
Accessories	2000		80450596,98 23	877006,7987	38786,018	
Accessories	2001		80450596,98 23	877006,7987	38786,018	
Accessories	2002	625,7933	80450596,98 23	877006,7987	38786,018	91495,3104
Accessories	2003	26037,3132	80450596,98 23	877006,7987	38786,018	572927,0136
Accessories	2004	12122,9115	80450596,98 23	877006,7987	38786,018	212584,4747
Accessories	2005		80450596,98 23	877006,7987	38786,018	

[ИМЯ КАТЕГОРИИ ПРОДУКТОВ]	[КАЛЕНДАРН ЫЙ ГОД]	[ОБЪЕМ ТОВАРООБОР ОТА ПОСРЕДНИКО В]	[ОБЩИЙ ИТОГ ПОСРЕДНИКА ]	[ВИЗУАЛЬНЫ Й ИТОГ ПОСРЕДНИКА ]	[ВИЗУАЛЬНЫ Й ИТОГ ПОСРЕДНИКА ЗА ВЕСЬ КАЛЕНДАРНЫ Й ГОД]	[ВИЗУАЛЬНЫ Й ИТОГ ПОСРЕДНИКА ДЛЯ ВСЕХ ИМЕН В КАТЕГОРИИ ПРОДУКТОВ]
Accessories	2006		80450596,98 23	877006,7987	38786,018	
Bikes	2000		80450596,98 23	877006,7987	689287,7939	
Bikes	2001		80450596,98 23	877006,7987	689287,7939	
Bikes	2002	73778,938	80450596,98 23	877006,7987	689287,7939	91495,3104
Bikes	2003	439771,4136	80450596,98 23	877006,7987	689287,7939	572927,0136
Bikes	2004	175737,4423	80450596,98 23	877006,7987	689287,7939	212584,4747
Bikes	2005		80450596,98 23	877006,7987	689287,7939	
Bikes	2006		80450596,98 23	877006,7987	689287,7939	
Clothing	2000		80450596,98 23	877006,7987	95090,7757	
Clothing	2001		80450596,98 23	877006,7987	95090,7757	
Clothing	2002	12132,4334	80450596,98 23	877006,7987	95090,7757	91495,3104
Clothing	2003	58234,2214	80450596,98 23	877006,7987	95090,7757	572927,0136
Clothing	2004	24724,1209	80450596,98 23	877006,7987	95090,7757	212584,4747
Clothing	2005		80450596,98 23	877006,7987	95090,7757	
Clothing	2006		80450596,98 23	877006,7987	95090,7757	
Components	2000		80450596,98 23	877006,7987	53842,2111	

[ИМЯ КАТЕГОРИИ ПРОДУКТОВ]	[КАЛЕНДАРН ЫЙ ГОД]	[ОБЪЕМ ТОВАРООБОР ОТА ПОСРЕДНИКО В]	[ОБЩИЙ ИТОГ ПОСРЕДНИКА ]	[ВИЗУАЛЬНЫ Й ИТОГ ПОСРЕДНИКА ]	[ВИЗУАЛЬНЫ Й ИТОГ ПОСРЕДНИКА ЗА ВЕСЬ КАЛЕНДАРНЫ Й ГОД]	[ВИЗУАЛЬНЫ Й ИТОГ ПОСРЕДНИКА ДЛЯ ВСЕХ ИМЕН В КАТЕГОРИИ ПРОДУКТОВ]
Components	2001		80450596,98 23	877006,7987	53842,2111	
Components	2002	4958,1457	80450596,98 23	877006,7987	53842,2111	91495,3104
Components	2003	48884,0654	80450596,98 23	877006,7987	53842,2111	572927,0136
Components	2004		80450596,98 23	877006,7987	53842,2111	212584,4747
Components	2005		80450596,98 23	877006,7987	53842,2111	
Components	2006		80450596,98 23	877006,7987	53842,2111	

В отчете представлены следующие столбцы:

Reseller Sales Amount

Фактическое значение продаж через торгового посредника для года и категории продуктов. Это значение отображается в ячейке в центре отчета на пересечении года и категории.

Визуальный итог посредника за весь календарный год

Общий итог по категории продуктов за все годы. Это значение отображается в конце столбца или строки для определенной категории продукта и по всем годам в отчете.

Визуальный итог посредника для всех имен в категорий продуктов

Общий итог за год по всем категориям продуктов. Это значение отображается в конце столбца или строки для заданного года и для всех категорий продуктов в отчете.

Визуальный итог посредника

Общий итог по категории продуктов за все годы и для всех категорий продуктов. Это значение обычно отображается в нижнем правом углу таблицы.

Общий итог посредника

Это общий итог по всем продажам через торгового посредника до применения любого фильтра. Обратите внимание на разницу со столбцом [Визуальный итог посредника]. Не стоит забывать, что этот отчет содержит два (2) фильтра, один по группе категории продуктов, а другой — по типу рекламной акции.

#### NOTE

Если в выражении есть явные фильтры, эти фильтры также применяются к выражению.

# CALCULATE

02.10.2020 • 8 minutes to read

Вычисляет выражение в измененном контексте фильтра.

## NOTE

Также имеется функция [CALCULATETABLE](#). Она имеет точно такую же функциональность, за исключением того, что изменяет [контекст фильтра](#), примененный к выражению, возвращающему *объект таблицы*.

## Синтаксис

```
CALCULATE(<expression>[, <filter1> [, <filter2> [, ...]]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Вычисляемое выражение.
filter1, filter2,...	(Необязательно) Логические выражения или табличные выражения, определяющие фильтры или функции модификаторов фильтра.

Выражение, используемое в качестве первого параметра, по сути является тем же, что и мера.

Фильтры могут быть:

- выражениями логического фильтра;
- выражениями фильтра таблицы.
- Функции изменения фильтра

При наличии нескольких фильтров они оцениваются с помощью [логического оператора И](#). Это означает, что все условия должны быть ИСТИННЫМИ в одно и то же время.

#### Выражения логического фильтра

Фильтр логического выражения — это выражение, результатом которого является значение TRUE или FALSE. Существует несколько правил, которые они должны соблюдать:

- Они могут ссылаться только на один столбец.
- Они не могут ссылаться на меры.
- Выражение не может использовать вложенную функцию CALCULATE.
- Они не могут использовать функции, которые просматривают или возвращают таблицу, включая статистические функции.

#### Выражение фильтра таблицы

Фильтр табличных выражений применяет объект таблицы в качестве фильтра. Это может быть ссылка на таблицу модели, но, скорее всего, это функция, возвращающая объект таблицы.

Функцию [FILTER](#) можно использовать для применения сложных условий фильтра, включая те,



которые не могут быть определены логическим выражением фильтра.

#### Функции модификатора фильтра

Функции изменения фильтра позволяют не только добавлять фильтры. Они предоставляют дополнительный контроль при изменении контекста фильтра.

ФУНКЦИЯ	НАЗНАЧЕНИЕ
<a href="#">REMOVEFILTERS</a>	Удаление всех фильтров или фильтров из одного или нескольких столбцов таблицы или из всех столбцов одной таблицы.
<a href="#">ALL</a> <sup>1</sup> , <a href="#">ALLEXCEPT</a> , <a href="#">ALLNOBLANKROW</a>	Удаление фильтров из одного или нескольких столбцов или из всех столбцов одной таблицы.
<a href="#">KEEPFILTERS</a>	Добавление фильтра без удаления существующих фильтров для тех же столбцов.
<a href="#">USERELATIONSHIP</a>	Вызов неактивной связи между связанными столбцами. В этом случае активная связь автоматически становится неактивной.
<a href="#">CROSSFILTER</a>	Изменение направления фильтра (от обоих до одного и от одного до обоих) или отключение связи.

<sup>1</sup> Функция ALL и ее варианты ведут себя как модификаторы фильтров и как функции, возвращающие объекты таблицы. Если функция REMOVEFILTERS поддерживается вашим средством, лучше использовать ее для удаления фильтров.

## Возвращаемое значение

Значение, которое является результатом выражения.

## Remarks

- При указании выражений фильтра функция CALCULATE изменяет контекст фильтра для вычисления выражения. Для каждого критерия фильтра существует два возможных стандартных результата, если выражение фильтра не заключено в функцию KEEPFILTERS:
  - Если столбцы (или таблицы) не находятся в контексте фильтра, то новые фильтры будут добавлены в контекст фильтра для вычисления выражения.
  - Если столбцы (или таблицы) уже находятся в контексте фильтра, существующие фильтры будут перезаписаны новыми фильтрами для вычисления выражения CALCULATE.
- Функция CALCULATE, используемая *без фильтров*, позволяет добиться определенного требования. Она переводит контекст строки в контекст фильтра. Она необходима, если выражение (не мера модели), которое суммирует данные модели, необходимо вычислить в контексте строки. Этот сценарий может возникнуть в формуле вычисляемого столбца или при вычислении выражения в функции-итераторе. Обратите внимание, что если мера модели используется 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 таблицы **Sales** выдает результат дохода, но только для продуктов с синим цветом.

Примеры в этой статье можно добавить в образец модели *Power BI Desktop*. Чтобы получить модель, см. [пример модели DAX](#).

```
Blue Revenue =  
CALCULATE(  
    SUM(Sales[Sales Amount]),  
    'Product'[Color] = "Blue"  
)
```

КАТЕГОРИЯ	ОБЪЕМ ПРОДАЖ	СИНИЙ ДОХОД
Accessories	1 272 057,89 долл. США	165 406,62 долл. США
Bikes	94 620 526,21 долл. США	8 374 313,88 долл. США
Clothing	2 117 613,45 долл. США	259 488,37 долл. США
Компоненты	\$ 11 799 076,66	803 642,10 долл. США
<b>Всего</b>	<b>109 809 274,20 долл. США</b>	<b>9 602 850,97 долл. США</b>

Функция **CALCULATE** вычисляет сумму по столбцу **Sales Amount** таблицы **Sales** в измененном контексте фильтра. Новый фильтр добавляется в столбец **Color** таблицы **Product** — или фильтр перезаписывает любой фильтр, уже примененный к столбцу.

Следующее определение меры таблицы **Sales** дает отношение продаж по продажам для всех каналов продаж.

КАНАЛ	ОБЪЕМ ПРОДАЖ	% ДОХОДА ВСЕГО КАНАЛА
Интернет	29 358 677,22 \$	26,74 %
Reseller	\$ 80 450 596,98	73,26 %
<b>Всего</b>	<b>109 809 274,20 долл. США</b>	<b>100,00 %</b>

```
Revenue % Total Channel =  
DIVIDE(  
    SUM(Sales[Sales Amount]),  
    CALCULATE(  
        SUM(Sales[Sales Amount]),  
        REMOVEFILTERS('Sales Order'[Channel])  
    )  
)
```

Функция **DIVIDE** делит выражение, которое суммирует сумму значений столбца **Sales Amount** таблицы **Sales** (в контексте фильтра), на то же выражение в измененном контексте фильтра. Функция **CALCULATE** изменяет контекст фильтра с помощью функции **REMOVEFILTERS**, которая является функцией-модификатором фильтра. Она удаляет фильтры из столбца **Канал**

таблицы **Заказы**.

Следующее определение вычисленного столбца таблицы **Customer** классифицирует клиентов по классу лояльности. Это очень простой сценарий. Когда размер дохода, полученного от клиента, составляет менее 2500 долл. США, он классифицируется как *Low*; в противном случае — *High*.

```
Customer Segment =  
IF(  
    CALCULATE(SUM(Sales[Sales Amount]), ALLEXCEPT(Customer, Customer[CustomerKey])) < 2500,  
    "Low",  
    "High"  
)
```

В этом примере контекст строки преобразуется в контекст фильтра. Это называется *переходом контекста*. Функция **ALLEXCEPT** удаляет фильтры из всех столбцов таблицы **Customer**, за исключением столбца **CustomerKey**.

## См. также

[Контекст фильтра](#)

[Контекст строки](#)

[Функция CALCULATETABLE](#)

[Функции фильтрации](#)

# CALCULATETABLE

02.10.2020 • 5 minutes to read

Вычисляет табличное выражение в измененном контексте фильтра.

## NOTE

Также имеется функция [CALCULATE](#). Она имеет точно такую же функциональность, за исключением того, что изменяет [контекст фильтра](#), примененный к выражению, возвращающему *скалярное значение*.

## Синтаксис

```
CALCULATETABLE(<expression>[, <filter1> [, <filter2> [, ...]]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Вычисляемое табличное выражение.
filter1, filter2,...	(Необязательно) Логические выражения или табличные выражения, определяющие фильтры или функции модификаторов фильтра.

Выражение, используемое в качестве первого параметра, должно быть таблицей модели или функцией, возвращающей таблицу.

Фильтры могут быть:

- выражениями логического фильтра;
- выражениями фильтра таблицы.
- Функции изменения фильтра

При наличии нескольких фильтров они оцениваются с помощью [логического оператора И](#). Это означает, что все условия должны быть ИСТИННЫМИ в одно и то же время.

#### Выражения логического фильтра

Фильтр логического выражения — это выражение, результатом которого является значение TRUE или FALSE. Существует несколько правил, которые они должны соблюдать:

- Они могут ссылаться только на один столбец.
- Они не могут ссылаться на меры.
- Выражение не может использовать вложенную функцию CALCULATE.
- Они не могут использовать функции, которые просматривают или возвращают таблицу, включая статистические функции.

#### Выражение фильтра таблицы

Фильтр табличных выражений применяет объект таблицы в качестве фильтра. Это может быть ссылка на таблицу модели, но, скорее всего, это функция, возвращающая объект таблицы. Функцию [FILTER](#) можно использовать для применения сложных условий фильтра, включая те, которые не могут быть определены

логическим выражением фильтра.

#### Функции модификатора фильтра

Функции изменения фильтра позволяют не только добавлять фильтры. Они предоставляют дополнительный контроль при изменении контекста фильтра.

ФУНКЦИЯ	НАЗНАЧЕНИЕ
<a href="#">REMOVEFILTERS</a>	Удаление всех фильтров или фильтров из одного или нескольких столбцов таблицы или из всех столбцов одной таблицы.
<a href="#">ALL</a> <sup>1</sup> , <a href="#">ALLEXCEPT</a> , <a href="#">ALLNOBLANKROW</a>	Удаление фильтров из одного или нескольких столбцов или из всех столбцов одной таблицы.
<a href="#">KEEPFILTERS</a>	Добавление фильтра без удаления существующих фильтров для тех же столбцов.
<a href="#">USERELATIONSHIP</a>	Вызов неактивной связи между связанными столбцами. В этом случае активная связь автоматически становится неактивной.
<a href="#">CROSSFILTER</a>	Изменение направления фильтра (от обоих до одного и от одного до обоих) или отключение связи.

<sup>1</sup> Функция ALL и ее варианты ведут себя как модификаторы фильтров и как функции, возвращающие объекты таблицы. Если функция REMOVEFILTERS поддерживается вашим средством, лучше использовать ее для удаления фильтров.

## Возвращаемое значение

Таблица значений.

## Remarks

- При указании выражений фильтра функция CALCULATETABLE изменяет контекст фильтра для вычисления выражения. Для каждого критерия фильтра существует два возможных стандартных результата, если выражение фильтра не заключено в функцию KEEPFILTERS:
  - Если столбцы (или таблицы) не находятся в контексте фильтра, то новые фильтры будут добавлены в контекст фильтра для вычисления выражения.
  - Если столбцы (или таблицы) уже находятся в контексте фильтра, существующие фильтры будут перезаписаны новыми фильтрами для вычисления 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 продаж через Интернет за 2006 г. Это значение позже используется для вычисления коэффициента продаж через Интернет по сравнению с общим объемом продаж за 2006 г.

Следующая формула:

```

= SUMX(
    CALCULATETABLE(
        'InternetSales_USD',
        'DateTime'[CalendarYear] = 2006
    ),
    [SalesAmount_USD]
)

```

В результате создается следующая таблица:

МЕТКИ СТРОК	INTERNET SALESAMOUNT_USD	CALCULATETABLE 2006 INTERNET SALES	INTERNET SALES TO 2006 RATIO
2005	2 627 031,40 долл. США	5 681 440,58 долл. США	0,46
2006	5 681 440,58 долл. США	5 681 440,58 долл. США	1.00
2007 г.	8 705 066,67 долл. США	5 681 440,58 долл. США	1,53
2008	9 041 288,80 долл. США	5 681 440,58 долл. США	1,59
Grand Total	26 054 827,45 долл. США	5 681 440,58 долл. США	4.59

## См. также

- [Контекст фильтра](#)
- [Функция CALCULATE \(DAX\)](#)
- [Функции фильтрации \(DAX\)](#)

# EARLIER

02.10.2020 • 7 minutes to read

Возвращает текущее значение указанного столбца на проходе внешней оценки упомянутого столбца.

Функция EARLIER полезна для вложенных вычислений, где вы хотите использовать определенное значение в качестве входных данных и вести вычисления на основе этих данных. В Microsoft Excel такие вычисления можно выполнять только в контексте текущей строки. Однако в DAX можно сохранить значение входных данных, а затем выполнить вычисление, используя данные из всей таблицы.

Функция EARLIER в основном используется в контексте вычисляемых столбцов.

## Синтаксис

```
EARLIER(<column>, <number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец или выражение, которые разрешаются в столбец.
num	<p>(необязательно) Положительное число проходов внешней оценки.</p> <p>Оценка на один уровень выше представлена значением 1; на два уровня — 2 и т. д.</p> <p>Если значение опущено, по умолчанию используется 1.</p>

## Возвращаемое значение

Текущее значение строки на основе **столбца** по **числу** проходов оценки.

## Исключения

Описание ошибок

## Remarks

- Функция EARLIER успешно выполняется, если контекст строки есть до начала сканирования таблицы. В противном случае возвращается ошибка.
- Производительность EARLIER может быть низкой, поскольку теоретически она может потребовать выполнения числа операций, близкого к общему количеству строк (в столбце), умноженного на то же самое число (в зависимости от синтаксиса выражения). Например, если в столбце 10 строк, может потребоваться примерно 100 операций. Если у вас есть 100 строк, то число операций может приближаться к 10 000.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в

вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

**NOTE**

На практике подсистема аналитики VertiPaq в памяти выполняет оптимизацию для уменьшения фактического числа вычислений, но следует соблюдать осторожность при создании формул, использующих рекурсию.

Пример

Чтобы проиллюстрировать использование EARLIER, необходимо создать сценарий, который вычисляет значение ранга, а затем использует это значение в других вычислениях.

Следующий пример основан на простой таблице, **ProductSubcategory**, которая показывает общий объем продаж для каждого значения ProductSubcategory.

Итоговая таблица, включая столбец ранжирования, показана здесь.

PRODUCTSUBCATEGORYKEY	ENGLISHPRODUCTSUBCATEGORYNAME	TOTALSUBCATEGORYSALES	SUBCATEGORYRANKING
18	Полукомбинезоны	156 167,88 долл. США	18
26	Велосипедные стойки	220 720,70 долл. США	14
27	Парковка для велосипедов	35 628,69 долл. США	30
28	Фляги и кофры	59 342,43 долл. США	24
5	Велосипедные каретки	48 643,47 долл. США	27
6	Тормоза	62 113,16 долл. США	23
19	Ограничения	47 934,54 долл. США	28
7	Цепочки	8 847,08 долл. США	35
29	Чистящие средства	16 882,62 долл. США	32
8	Ведущие звезды	191 522,09 долл. США	15
9	Дерейлеры	64 965,33 долл. США	22
30	Бамперы	41 974,10 долл. США	29
10	Вилки	74 727,66 долл. США	21
20	Перчатки	228 353,58 долл. США	12
4	Рули	163 257,06 долл. США	17
11	Гарнитур	57 659,99 долл. США	25
31	Шлемы	451 192,31 долл. США	9



PRODUCTSUBCATEGORYKEY	ENGLISHPRODUCTSUBCATEGORYNAME	TOTALSUBCATEGORYSALES	SUBCATEGORYRANKING
32	Гидраторы	96 893,78 долл. США	20
21	Веломайки	699 429,78 долл. США	7
33	Освещение		36
34	Блокировки	15 059,47 долл. США	33
1	Горные велосипеды	34 305 864,29 долл.	2
12	Горные рамы	4 511 170,68 долл.	4
35	Кожухи		36
13	Педали	140 422,20 долл. США	19
36	Насосы	12 695,18 долл. США	34
2	Дорожные велосипеды	40 551 696,34 долл.	1
14	Дорожные рамы	3 636 398,71 долл.	5
15	Седла	52 526,47 долл. США	26
22	Шорты	385 707,80 долл. США	10
23	Носки	28 337,85 долл. США	31
24	Чулки	189 179,37 долл. США	16
37	Шины и камеры	224 832,81 долл. США	13
3	Туристические велосипеды	13 334 864,18 долл.	3
16	Туристические рамы	1 545 344,02 долл.	6
25	Жилеты	240 990,04 долл. США	11
17	Колеса	648 240,04 долл. США	8

## Создание значения ранга

Одним из способов получения значения ранга для заданного значения в строке является подсчет количества строк в одной и той же таблице, имеющих значение больше (или меньше), чем сравниваемое. Этот метод возвращает пустое или нулевое значение для самого высокого значения в таблице, тогда как равные значения будут иметь одинаковое значение ранга; следующее значение (после одинаковых значений) будет иметь другое значение ранга. См. следующий пример.

Новый вычисляемый столбец, `SubCategorySalesRanking`, создается с помощью следующей формулы.

```
= COUNTROWS(FILTER(ProductSubcategory, EARLIER(ProductSubcategory[TotalSubcategorySales])<ProductSubcategory[TotalSubcategorySales]))+1
```

Следующие шаги описывают метод вычисления более подробно.

1. Функция **EARLIER** возвращает значение *TotalSubcategorySales* для текущей строки в таблице. В этом случае, поскольку процесс запускается, это первая строка в таблице.
2. **EARLIER**([*TotalSubcategorySales*]) вычисляет значение 156 167,88 долл. США, текущую строку во внешнем цикле.
3. Функция **FILTER** теперь возвращает таблицу, где все строки имеют значение *TotalSubcategorySales* больше 156 167,88 долл. США (это текущее значение для **EARLIER**).
4. Функция **COUNTROWS** подсчитывает строки отфильтрованной таблицы и присваивает это значение новому вычисляемому столбцу в текущей строке, прибавив 1. Необходимо добавить 1, чтобы не допустить появления пустого значения верхнего ранга.
5. Формула вычисляемого столбца перемещается к следующей строке и повторяет шаги с 1 по 4. Эти действия повторяются до тех пор, пока не будет достигнут конец таблицы.

Функция **EARLIER** всегда будет получать значение столбца до текущей операции таблицы. Если необходимо получить значение из цикла перед этим, установите второй аргумент равным 2.

## См. также раздел

[Функция EARLIEST](#)

[Функции фильтрации](#)

# EARLIEST

22.09.2020 • 2 minutes to read

Возвращает текущее значение указанного столбца на проходе внешней оценки указанного столбца.

## Синтаксис

```
EARLIEST(<column>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Ссылка на столбец.

## Возвращаемое значение

Столбец с удаленными фильтрами.

## Remarks

- Функция EARLIEST аналогична функции EARLIER, за одним исключением — она 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 образец данных не поддерживает этот сценарий.

```
= EARLIEST(<column>)
```

## См. также раздел

[EARLIER, функция](#)

[Функции фильтрации](#)

# FILTER

02.10.2020 • 4 minutes to read

Возвращает таблицу, представляющую подмножество другой таблицы или выражения.

## Синтаксис

```
FILTER(<table>,<filter>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
таблица	Фильтруемая таблица. Это также может быть выражение, результатом которого является таблица.
filter	Логическое выражение, которое должно вычисляться для каждой строки таблицы. Например, <code>[Amount] &gt; 0</code> или <code>[Region] = "France"</code>

## Возвращаемое значение

Таблица, содержащая только отфильтрованные строки.

## Примечания

- Можно использовать фильтр для сокращения количества строк в таблице, с которыми вы работаете, и использовать только определенные данные в вычислениях. Функция FILTER не используется независимо, а в качестве функции, внедренной в другие функции, для которых требуется таблица в качестве аргумента.
- 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 через Интернет за пределами США с помощью меры, которая фильтрует продажи в США, а затем выполняет срез по календарному году и категориям продуктов. Чтобы создать эту меру, необходимо отфильтровать таблицу, Internet Sales USD, по территории продаж, а затем использовать отфильтрованную таблицу в функции SUMX.

В этом примере выражение

```
FILTER('InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States")
```

возвращает таблицу, которая является подмножеством продаж через Интернет за вычетом всех строк, относящихся к США. Функция RELATED связывает ключ территории в таблице продаж через Интернет с SalesTerritoryCountry в таблице SalesTerritory.

В следующей таблице показано подтверждение концепции для меры (продажи через Интернет ЗА

ПРЕДЕЛАМИ США), формула, которая приведена в разделе кода ниже. Таблица сравнивает все продажи через Интернет с продажами через Интернет за пределами США, чтобы показать, что выражение фильтра работает, исключая продажи в США из вычислений.

Чтобы повторно создать эту таблицу, добавьте поле SalesTerritoryCountry в области **Метки строк** отчета или сводной таблицы.

Таблица 1. Сравнение общего объема продаж в США и всех других регионах

МЕТКИ СТРОК	INTERNET SALES	ПРОДАЖИ ЧЕРЕЗ ИНТЕРНЕТ ЗА ПРЕДЕЛАМИ США
Австралия	4 999 021,84 долл. США	4 999 021,84 долл. США
Канада	1 343 109,10 долл. США	1 343 109,10 долл. США
Франция	2 490 944,57 долл. США	2 490 944,57 долл. США
Германия	2 775 195,60 долл. США	2 775 195,60 долл. США
Соединенное Королевство	5 057 076,55 долл. США	5 057 076,55 долл. США
США	9 389 479,79 долл. США	
Grand Total	26 054 827,45 долл. США	16 665 347,67 долл. США

В последней таблице отчета отображаются результаты при создании сводной таблицы с помощью меры (продажи через Интернет ЗА ПРЕДЕЛАМИ США). Добавьте поле CalendarYear в область Метки строк сводной таблицы и поле ProductCategoryName в область Метки столбцов.

Таблица 2. Сравнение продаж за пределами США по категориям продуктов

ПРОДАЖИ ЧЕРЕЗ ИНТЕРНЕТ ЗА ПРЕДЕЛАМИ США	МЕТКИ СТОЛБЦОВ			
Метки строк	Принадлежности	Велосипеды	Экипировка	Общий итог
2005		1 526 481,95 долл. США		1 526 481,95 долл. США
2006		3 554 744,04 долл. США		3 554 744,04 долл. США
2007 г.	156 480,18 долл. США	5 640 106,05 долл. США	70 142,77 долл. США	5 866 729,00 долл. США
2008	228 159,45 долл. США	5 386 558,19 долл. США	102 675,04 долл. США	5 717 392,68 долл. США
Grand Total	384 639,63 долл. США	16 107 890,23 долл. США	172 817,81 долл. США	16 665 347,67 долл. США

```
SUMX(FILTER('InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States"), 'InternetSales_USD'[SalesAmount_USD])
```

## См. также

[Функции фильтрации](#)

[Функция ALL](#)

[Функция ALLEXCEPT](#)

# KEEPFILTERS

02.10.2020 • 6 minutes to read

Изменяет порядок применения фильтров при вычислении функции CALCULATE или CALCULATETABLE.

## Синтаксис

```
KEEPFILTERS(<expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Любое выражение.

## Возвращаемое значение

Таблица значений.

## Remarks

- Функцию KEEPFILTERS можно использовать в контексте функций CALCULATE и CALCULATETABLE для переопределения стандартного поведения этих функций.
- По умолчанию аргументы фильтра в таких функциях, как CALCULATE, используются в качестве контекста для вычисления выражения, поэтому аргументы фильтра для CALCULATE заменяют все существующие фильтры в тех же столбцах. Новый контекст, на который влияет аргумент фильтра для CALCULATE, влияет только на существующие фильтры для столбцов, упомянутых в аргументе фильтра. Фильтры для столбцов, отличные от упомянутых в аргументах функции CALCULATE или других связанных функциях, остаются в силе и не изменяются.
- Функция KEEPFILTERS позволяет изменить это поведение. При использовании функции KEEPFILTERS все существующие фильтры в текущем контексте сравниваются со столбцами в аргументах фильтра, а пересечение этих аргументов используется в качестве контекста для вычисления выражения. В результате выполнения операции с любым из этих столбцов применяются оба набора аргументов: оба аргумента фильтра, используемые в CALCULATE, и фильтры в аргументах функции KEEPFILTER. Иными словами, фильтры CALCULATE заменяют текущий контекст, тогда как функция 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 рассматриваются некоторые распространенные сценарии, демонстрирующие использование функции KEEPFILTERS в формулах CALCULATE или CALCULATETABLE.

Первые три выражения получают простые данные, которые будут использоваться для сравнения:

- Продажи через Интернет в штате Вашингтон.

- Продажи через Интернет в штатах Вашингтон и Орегон (объединенные данные по обоим штатам).
- Продажи через Интернет в штате Вашингтон и провинции Британская Колумбия (объединенные данные по регионам).

Четвертое выражение вычисляет продажи через Интернет в штатах Вашингтон и Орегон с применением фильтра для штата Вашингтон и провинции Британская Колумбия.

Следующее выражение вычисляет продажи через Интернет в штатах Вашингтон и Орегон с использованием функции KEEPFILTERS и применением фильтра для штата Вашингтон и провинции Британская Колумбия, который является частью прежнего контекста.

```
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"
  )
)
```

При вычислении этого выражения с использованием образца базы данных AdventureWorks DW результаты будут следующими.

СТОЛБЕЦ	ЗНАЧЕНИЕ
[\$\$ 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



СТОЛБЕЦ	ЗНАЧЕНИЕ
[\$\$ in WA !!]	\$ 2,467,248.34

#### NOTE

В целях обучения приведенные выше результаты представлены в виде таблицы, а не одной строкой.

Сначала изучите выражение `[$$ in WA and OR ??]` . Может возникнуть вопрос, как эта формула может возвращать значение продаж в штатах Вашингтон и Орегон, когда внешнее выражение CALCULATE включает фильтр для штата Вашингтон и провинции Британская Колумбия. Ответ заключается в том, что поведение выражения CALCULATE по умолчанию переопределяет внешние фильтры в Geography'[State Province Code] и заменяет собственные аргументы фильтра, поскольку фильтры применяются к одному и тому же столбцу.

Затем изучите выражение `[$$ in WA !!]` . Может возникнуть вопрос, как эта формула может возвращать значение продаж только в штате Вашингтон, когда аргумент фильтра включает штат Орегон, а внешнее выражение CALCULATE включает фильтр для штата Вашингтон и провинции Британская Колумбия. Ответ заключается в том, что функция KEEPFILTERS изменяет стандартное поведение функции CALCULATE и добавляет дополнительный фильтр. Поскольку используется пересечение фильтров, в аргумент фильтра `'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="OR"` добавляется внешний фильтр `'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="BC"` . Поскольку оба фильтра применяются к одному и тому же столбцу, результирующий фильтр `'Geography'[State Province Code]="WA"` — это фильтр, применяемый при вычислении выражения.

## См. также раздел

[Функции фильтрации](#)

[CALCULATE, функция](#)

[Функция CALCULATETABLE](#)

# LOOKUPVALUE

21.10.2020 • 3 minutes to read

Возвращает значение для строки, удовлетворяющей всем критериям, указанным в одном или нескольких условиях поиска.

## Синтаксис

```
LOOKUPVALUE(  
    <result_columnName>,  
    <search_columnName>,  
    <search_value>  
    [, <search2_columnName>, <search2_value>]...  
    [, <alternateResult>]  
)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
result_columnName	Имя существующего столбца, содержащего возвращаемое значение. Этот параметр не может быть выражением.
search_columnName	Имя существующего столбца. Он может находиться в той же таблице, что и result_columnName или в связанной таблице. Этот параметр не может быть выражением.
search_value	Искомое значение в search_columnName.
alternateResult	(Необязательно) Значение, возвращаемое, если контекст для result_columnName был отфильтрован по нулю или более чем по одному отдельному значению. Если значение не указано, функция возвращает BLANK, если result_columnName фильтруется до нулевого значения, или ошибку, если имеется более одного уникального значения.

## Возвращаемое значение

Значение **result\_column** в строке, где все пары **search\_column** и **search\_value** в точности совпадают.

Если совпадений, удовлетворяющих всем значениям поиска, не найдено, возвращается значение BLANK или **alternateResult** (если оно задано). Иными словами, функция не будет возвращать значение поиска, если совпадают только некоторые критерии.

Если несколько строк соответствуют значениям поиска и во всех случаях значения **result\_column** идентичны, возвращается значение. Однако если **result\_column** возвращает разные значения, возвращается ошибка или **alternateResult**, если последний параметр задан.

## Remarks

- Если между таблицами результатов и поиска существует связь, в большинстве случаев использование функции [RELATED](#) вместо LOOKUPVALUE является более эффективным и обеспечивает лучшую производительность.
- Параметры `search_value` и `alternateResult` вычисляются до того, как функция выполняет итерацию по строкам таблицы поиска.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Примеры в этой статье можно добавить в образец модели Power BI Desktop. Чтобы получить модель, см. [пример модели DAX](#).

Следующий вычисляемый столбец, определенный в таблице `Sales`, использует функцию LOOKUPVALUE для возврата значений канала из таблицы `Sales Order`.

```
CHANNEL = LOOKUPVALUE('Sales Order'[Channel], 'Sales Order'[SalesOrderLineKey], [SalesOrderLineKey])
```

Однако, поскольку в этом случае между таблицами `Sales Order` и `Sales` существует связь, более эффективным будет использование функции [RELATED](#).

```
CHANNEL = RELATED('Sales Order'[Channel])
```

## См. также

[Функция RELATED \(DAX\)](#)

[Информационные функции](#)

# REMOVEFILTERS

22.09.2020 • 2 minutes to read

Очистка фильтров в указанных таблицах или столбцах.

## Синтаксис

```
REMOVEFILTERS([<table> | <column>[, <column>[, <column>[,...]]]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, в которой нужно очистить фильтры.
столбец	Столбец, в котором нужно очистить фильтры.

## Возвращаемое значение

Недоступно См. примечания.

## Remarks

- REMOVEFILTERS можно использовать только для очистки фильтров, но не для возвращения 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

```
DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales], CALCULATE([TotalSales],REMOVEFILTERS()))

EVALUATE
    SUMMARIZECOLUMNS(
        ROLLUPADDSUBTOTAL(DimProductCategory[EnglishProductCategoryName], "IsGrandTotal"),
        "TotalSales", [TotalSales],
        "%Sales", [%Sales]
    )
ORDER BY
    [IsGrandTotal] DESC, [TotalSales] DESC
```

Результаты

DIMPRODUCTCATEGORY[ENGLISHPRODUCTCATEGORYNAME]	[ISGRANDTOTAL]	[TOTALSALES]	[%SALES]
Строка1	True	29358677,2207	1
Bikes	False	28318144,6507	0,964557920570538
Accessories	False	700759,96	0,023868921434441
Clothing	False	339772,61	0,0115731579950215

## Пример 2

Запрос DAX

```
DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],
CALCULATE([TotalSales],REMOVEFILTERS(DimProductSubcategory[EnglishProductSubcategoryName])))

EVALUATE
SUMMARIZECOLUMNS(
    DimProductCategory[EnglishProductCategoryName],
    DimProductSubcategory[EnglishProductSubcategoryName],
    "TotalSales", [TotalSales],
    "%Sales", [%Sales]
)
ORDER BY
    DimProductCategory[EnglishProductCategoryName] ASC,
    DimProductSubcategory[EnglishProductSubcategoryName] ASC
```

Результаты

DIMPRODUCTCATEGORY[ENGLISHPRODUCTCATEGORYNAME]	DIMPRODUCTSUBCATEGORY[ENGLISHPRODUCTSUBCATEGORYNAME]	[TOTALSALES]	[%SALES]
Accessories	Велосипедные стойки	39360	0,05616759
Accessories	Парковка для велосипедов	39591	0,05649723
Accessories	Фляги и кофры	56798,19	0,08105228
Accessories	Чистящие средства	7218,6	0,0103011
Accessories	Бамперы	46619,58	0,06652717
Accessories	Шлемы	225335,6	0,3215589
Accessories	Гидраторы	40307,67	0,05751994
Accessories	Шины и камеры	245529,32	0,35037578
Bikes	Горные велосипеды	9952759,564	0,35146228

DIMPRODUCTCATEGORY [ENGLISHPRODUCTCATEGORYNAME]	DIMPRODUCTSUBCATEGORY [ENGLISHPRODUCTSUBCATEGORYNAME]	[TOTALSALES]	[%SALES]
Bikes	Дорожные велосипеды	14520584,04	0,51276608
Bikes	Туристические велосипеды	3844801,05	0,13577164
Clothing	Ограничения	19688,1	0,05794493
Clothing	Перчатки	35020,7	0,10307099
Clothing	Веломайки	172950,68	0,5090189
Clothing	Шорты	71319,81	0,20990453
Clothing	Носки	5106,32	0,01502864
Clothing	Жилеты	35687	0,10503201

# SELECTEDVALUE

22.09.2020 • 2 minutes to read

Возвращает значение, если контекст для columnName был отфильтрован только по одному отдельному значению. В противном случае возвращает alternateResult.

## Синтаксис

```
SELECTEDVALUE(<columnName>[, <alternateResult>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.
alternateResult	(Необязательно) Значение, возвращаемое, если контекст для columnName был отфильтрован по нулю или более чем по одному отдельному значению. Если не указано, значение по умолчанию является выражением BLANK().

## Возвращаемое значение

Значение, если контекст для columnName был отфильтрован только по одному отдельному значению. В противном случае — alternateResult.

## Remarks

- Эквивалентное выражение для `SELECTEDVALUE(<columnName>, <alternateResult>)` — `IF(HASONEVALUE(<columnName>), VALUES(<columnName>), <alternateResult>)`.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Рассмотрим следующий запрос DAX:

```
DEFINE
    MEASURE DimProduct[Selected Color] = SELECTEDVALUE(DimProduct[Color], "No Single Selection")
EVALUATE
    SUMMARIZECOLUMNS
        (ROLLUPADDSUBTOTAL(DimProduct[Color], "Is Total"),
          "Selected Color", [Selected Color])ORDER BY [Is Total] ASC,
          [Color] ASC
```

Он возвращает следующее:

DIMPRODUCT[COLOR]	[IS TOTAL]	[SELECTED COLOR]
Черный	FALSE	Черный
Синий	FALSE	Синий
Серый	FALSE	Серый
Несколько вариантов	FALSE	Несколько вариантов
Н/Д	FALSE	Н/Д
Красный	FALSE	Красный
Серебряная	FALSE	Серебряная
Серебряный/черный	FALSE	Серебряный/черный
White	FALSE	White
Желтый	FALSE	Желтый
	TRUE	Без выбора одного элемента



# Финансовые функции

02.10.2020 • 7 minutes to read

Финансовые функции в DAX 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.

## В этой категории

ФУНКЦИЯ	ОПИСАНИЕ
ACCRINT	Возвращает накопленный процент для ценной бумаги, по которой периодически выплачиваются проценты.
ACCRINTM	Возвращает накопленный процент для ценной бумаги, по которой выплачиваются проценты при погашении.
AMORDEGRC	Возвращает амортизацию для каждого учетного периода. Аналогично функции AMORLINC, за исключением того, что коэффициент амортизации применяется в зависимости от срока службы активов.
AMORLINC	Возвращает амортизацию для каждого учетного периода.
COUPDAYBS	Возвращает число дней с начала купонного периода до даты расчетов.
COUPDAYS	Возвращает число дней купонного периода, который содержит дату расчетов.
COUPDAYSNC	Возвращает число дней от даты расчетов до следующей даты выплаты купонного дохода.
COUPNCD	Возвращает следующую дату выплаты купонного дохода после даты расчетов.
COUPNUM	Возвращает количество купонов между датой сопоставления и датой погашения, округленное до ближайшего целого купона.
COUPPCD	Возвращает предыдущую купонную дату до даты расчетов.
CUMIPMT	Возвращает совокупный процент, уплаченный по кредиту между значениями start_period и end_period.
CUMPRINC	Возвращает совокупную основную сумму, уплаченную по кредиту между значениями start_period и end_period.

ФУНКЦИЯ	ОПИСАНИЕ
DB	Возвращает значение, указывающее амортизацию ресурса за указанный период методом фиксированного уменьшения остатка.
DDB	Возвращает значение, указывающее амортизацию ресурса за указанный период методом двойного уменьшения остатка (или другим методом, определенным пользователем).
DISC	Возвращает процентную ставку для ценной бумаги.
DOLLARDE	Преобразует цену в долларах, выраженную в виде целого числа и дробной части, например 1,02, в цену в долларах, выраженную в виде десятичного числа.
DOLLARFR	Преобразует цену в долларах, выраженную в виде целого числа и дробной части, например 1,02, в цену в долларах, выраженную в виде десятичного числа.
DURATION	Возвращает дюрацию Маколея для предполагаемой номинальной стоимости 100 долл. США.
EFFECT	Возвращает действительную годовую процентную ставку с учетом номинальной годовой процентной ставки и числа составляющих периодов в год.
FV	Вычисляет будущую стоимость инвестиции на основе постоянной процентной ставки.
INTRATE	Возвращает процентную ставку для полностью инвестированной ценной бумаги.
IPMT	Возвращает платеж по процентам за заданный 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 равными платежами по основной сумме.
MDURATION	Возвращает измененную дюрацию Маколея для ценной бумаги с предполагаемой номинальной стоимостью 100 долларов США.
NOMINAL	Возвращает номинальную годовую процентную ставку с учетом фактической ставки и числа составляющих периодов в год.
NPER	Возвращает число периодов для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.
ODDFPRICE	Возвращает цену за \$100 долл. США номинальной стоимости для ценной бумаги с нестандартным (коротким или длинным) первым периодом.

ФУНКЦИЯ	ОПИСАНИЕ
ODDFYIELD	Возвращает доходность ценной бумаги с нестандартным (коротким или длинным) первым периодом.
ODDLPRICE	Возвращает цену за 100 долл. США номинальной стоимости для ценной бумаги с нестандартным (коротким или длинным) последним купонным периодом.
ODDLYIELD	Возвращает доходность ценной бумаги с нестандартным (коротким или длинным) последним периодом.
PDURATION	Возвращает количество периодов, необходимых инвестиции для достижения заданной стоимости.
PMT	Вычисляет платеж по кредиту на основе постоянных платежей и постоянной процентной ставки.
PPMT	Возвращает платеж по основной сумме за заданный период для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.
PRICE	Возвращает цену за \$100 долл. США номинальной стоимости ценной бумаги с периодической выплатой процентов.
PRICEDISC	Возвращает цену за \$100 номинальной стоимости дисконтной ценной бумаги.
PRICEMAT	Возвращает цену за 100 долл. США номинальной стоимости ценной бумаги, по которой выплачивается процент при погашении.
PV	Вычисляет текущую стоимость кредита или инвестиции на основе постоянной процентной ставки.
RATE	Возвращает процентную ставку за период в рамках аннуитета.
RECEIVED	Возвращает сумму, полученную при погашении, для полностью инвестированной ценной бумаги.
RRI	Возвращает эквивалентную процентную ставку для роста инвестиций.
SLN	Возвращает линейную амортизацию актива за один период.
SYD	Возвращает значение, указывающее амортизацию актива за указанный период, рассчитанную методом суммы цифр лет.
TBILLEQ	Возвращает эквивалентную доходность облигации для казначейского векселя.

ФУНКЦИЯ	ОПИСАНИЕ
TBILLPRICE	Возвращает значение цены за 100 долл. США номинальной стоимости для казначейского векселя.
TBILLYIELD	Возвращает доход для казначейского векселя.
VDB	Возвращает значение, указывающее амортизацию актива за указанный период, включая частичные периоды, методом двойного уменьшения остатка (или другим методом, определенным пользователем).
XIRR	Возвращает внутреннюю норму прибыли для запланированных денежных потоков, которые могут не являться периодическими.
XNPV	Возвращает текущее значение для запланированных денежных потоков, которые могут не являться периодическими.
YIELD	Возвращает доход с ценной бумаги, по которой периодически выплачиваются проценты.
YIELDDISC	Возвращает годовой доход для дисконтной ценной бумаги.
YIELDMAT	Возвращает годовую доходность для ценной бумаги, по которой выплачиваются проценты при погашении.

# ACCRINT

02.10.2020 • 4 minutes to read

Возвращает накопленный процент для ценной бумаги, по которой периодически выплачиваются проценты.

## Синтаксис

```
ACCRINT(<issue>, <first_interest>, <settlement>, <rate>, <par>, <frequency>[, <basis>[, <calc_method>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
проблема	Дата выпуска ценной бумаги.
first_interest	Дата первой выплаты процентов по ценной бумаге.
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
rate	Годовая купонная ставка ценной бумаги.
par	Номинальная стоимость ценной бумаги.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.
calc_method	(Необязательно) Логическое значение, указывающее способ вычисления общего накопленного процента, когда дата расчетов превышает дату first_interest. Если значение calc_method не указано, предполагается, что оно равно TRUE. — Если calc_method принимает значение TRUE или опущен, функция ACCRINT возвращает общий накопленный процент с даты выпуска до даты расчетов. — Если calc_method принимает значение FALSE, функция ACCRINT возвращает общий накопленный процент с даты first_interest до даты расчетов.

Параметр basis принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Накопленный процент.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.

- ACCRINT рассчитывается следующим образом:

$$\text{ACCRINT} = \text{par} \times \frac{\text{rate}}{\text{frequency}} \times \sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i}$$

где:

- $\text{A}_i$  — общее число дней для купонного квазипериода  $i^{\text{th}}$  в нестандартном периоде.
- $\text{NC}$  — число купонных квазипериодов в нестандартном периоде. Если это число содержит дробную часть, округлите его до следующего целого числа.
- $\text{NL}_i$  — обычная длина в днях купонного квазипериода в нестандартном периоде.
- Значения issue, first\_interest и settlement усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - значения issue, first\_interest или settlement не являются допустимой датой;
  - $\text{issue} \geq \text{settlement}$ ;
  - $\text{rate} \leq 0$ ;
  - $\text{par} \leq 0$ ;
  - frequency является любым числом, кроме 1, 2 или 4;
  - $\text{basis} < 0$  или  $\text{basis} > 4$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Примеры

DATA	ОПИСАНИЕ
1.03.2007	Дата выпуска
31.08.2008	Дата первой выплаты процентов

DATA	ОПИСАНИЕ
1.05.2008	Дата расчета
10 %	Купонная ставка
1000	Номинальная стоимость
2	Периодичность — раз в полгода (см. выше)
0	База 30/360 (см. выше)

### Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0)
}
```

Возвращает накопленный процент от issue до settlement для ценной бумаги на основе указанных выше условий.

[ЗНАЧЕНИЕ]
116,9444444444444

### Пример 2

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    ACCRINT(DATE(2007,3,1), DATE(2008,8,31), DATE(2008,5,1), 0.1, 1000, 2, 0, FALSE)
}
```

Возвращает накопленный процент от first\_interest до settlement для ценной бумаги на основе указанных выше условий.

[ЗНАЧЕНИЕ]
66,94444444444445

# ACCRINTM

02.10.2020 • 2 minutes to read

Возвращает накопленный процент для ценной бумаги, по которой выплачиваются проценты при погашении.

## Синтаксис

```
ACCRINTM(<issue>, <maturity>, <rate>, <par>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
проблема	Дата выпуска ценной бумаги.
maturity	Дата погашения ценной бумаги.
rate	Годовая купонная ставка ценной бумаги.
par	Номинальная стоимость ценной бумаги.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр basis принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Накопленный процент.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.



- ACCRINTM рассчитывается следующим образом:

$$\text{\text{\text{ACCRINTM}}} = \text{\text{par}} \times \text{\text{rate}} \times \frac{\text{\text{A}}}{\text{\text{D}}}$$

где:

- $\text{\text{A}}$  — число накопленных дней, рассчитанных по месячной базе. Для подсчета процентов с выплатой в дату погашения используется число дней с даты выпуска до даты погашения.
- $\text{\text{D}}$  — годовая база.
- Значения issue и maturity усекаются до целых чисел.
- basis округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - issue или maturity не является допустимой датой;
  - $\text{issue} \geq \text{maturity}$ ;
  - $\text{rate} \leq 0$ ;
  - $\text{par} \leq 0$ ;
  - $\text{basis} < 0$  или  $\text{basis} > 4$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

DATA	ОПИСАНИЕ
01.04.2008	Дата выпуска
15.06.2008	Дата погашения
10 %	Процентный купон
1000	Номинальная стоимость
3	База по факту/365 (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    ACCRINTM(
        DATE(2008,4,1),
        DATE(2008,6,15),
        0.1,
        1000,
        3
    )
}
```

Возвращает накопленный процент для ценной бумаги на основе указанных выше условий.

[ЗНАЧЕНИЕ]
20,5479452054795

# AMORDEGRC

02.10.2020 • 4 minutes to read

Возвращает амортизацию для каждого учетного периода. Эта функция предоставляется для французской системы учета. Если ресурс приобретается в середине учетного периода, учитывается пропорциональная амортизация. Функция аналогична функции AMORLINC, за исключением того, что коэффициент амортизации применяется в вычислениях в зависимости от срока службы активов.

## Синтаксис

```
AMORDEGRC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>, <rate>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
cost	Стоимость актива.
date_purchased	Дата приобретения актива.
first_period	Дата окончания первого периода.
salvage	Остаточная стоимость в конце срока службы актива.
period	Период.
rate	Процент амортизации.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	СИСТЕМА ДАТ
0 или не указано	360 дней (метод NASD)
1	Actual
3	365 дней в году
4	360 дней в году (европейский метод)

## Возвращаемое значение

Амортизация для каждого учетного периода.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Эта функция возвращает амортизацию до последнего периода срока службы активов или до тех пор, пока накопленное значение амортизации не превысит стоимость ресурсов за вычетом остаточной стоимости.
- Коэффициенты амортизации:

СРОК СЛУЖБЫ АКТИВОВ (1/КОЭФФИЦИЕНТ)	КОЭФФИЦИЕНТ АМОРТИЗАЦИИ
От 3 до 4 лет	1.5
От 5 до 6 лет	2
Более 6 лет	2.5

- Коэффициент амортизации будет увеличен до 50 % за период, предшествующий последнему периоду, и до 100 % за последний период.
- Значения period и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - cost < 0;
  - значение first\_period или date\_purchased не является допустимой датой;
  - date\_purchased > first\_period;
  - salvage < 0 или salvage > cost;
  - period < 0;
  - rate ≤ 0.
  - Срок службы ресурсов составляет от 0 (ноль) до 1, от 1 до 2, от 2 до 3 и от 4 до 5.
  - basis является любым числом, кроме 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

DATA	ОПИСАНИЕ
2400	Cost
19.08.2008	Дата приобретения
31.12.2008	Конец первого периода
300	Остаточная стоимость
1	Период
15 %	Процент амортизации

DATA	ОПИСАНИЕ
1	База по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
  AMORDEGRC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)
}
```

Возвращает амортизацию за первый период на основе указанных выше условий.

[ЗНАЧЕНИЕ]
776

# AMORLINC

02.10.2020 • 2 minutes to read

Возвращает амортизацию для каждого учетного периода. Эта функция предоставляется для французской системы учета. Если ресурс приобретается в середине учетного периода, учитывается пропорциональная амортизация.

## Синтаксис

```
AMORLINC(<cost>, <date_purchased>, <first_period>, <salvage>, <period>, <rate>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
cost	Стоимость актива.
date_purchased	Дата приобретения актива.
first_period	Дата окончания первого периода.
salvage	Остаточная стоимость в конце срока службы актива.
period	Период.
rate	Процент амортизации.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	СИСТЕМА ДАТ
0 или не указано	360 дней (метод NASD)
1	Actual
3	365 дней в году
4	360 дней в году (европейский метод)

## Возвращаемое значение

Амортизация для каждого учетного периода.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Значения period и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - $cost < 0$ ;
  - значение first\_period или date\_purchased не является допустимой датой;
  - $date\_purchased > first\_period$ ;
  - $salvage < 0$  или  $salvage > cost$ ;
  - $period < 0$ ;
  - $rate \leq 0$ ;
  - basis является любым числом, кроме 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

DATA	ОПИСАНИЕ
2400	Cost
19.08.2008	Дата приобретения
31.12.2008	Конец первого периода
300	Остаточная стоимость
1	Период
15 %	Процент амортизации
1	База по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    AMORLINC(2400, DATE(2008,8,19), DATE(2008,12,31), 300, 1, 0.15, 1)
}
```

Возвращает амортизацию за первый период на основе указанных выше условий.

[ЗНАЧЕНИЕ]
360

# COUPDAYBS

02.10.2020 • 3 minutes to read

Возвращает число дней с начала купонного периода до даты расчетов.

## Синтаксис

```
COUPDAYBS(<settlement>, <maturity>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Число дней с начала купонного периода до даты расчетов.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как

прошло 39 448 дней после 30 декабря 1899 г.

- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Даты settlement и maturity усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ
25.01.2011	Дата расчета
15.11.2011	Дата погашения
2	Полугодовой купон (см. выше)
1	База по факту/по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    COUPDAYBS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Возвращает число дней с начала купонного периода до даты расчетов для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
71



# COUPDAYS

02.10.2020 • 3 minutes to read

Возвращает число дней купонного периода, который содержит дату расчетов.

## Синтаксис

```
COUPDAYS(<settlement>, <maturity>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Число дней купонного периода, который содержит дату расчетов.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как

прошло 39 448 дней после 30 декабря 1899 г.

- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Даты settlement и maturity усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ
25.01.2011	Дата расчета
15.11.2011	Дата погашения
2	Полугодовой купон (см. выше)
1	База по факту/по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    COUPDAYS(DATE(2011,1,25), DATE(2011,11,15), 2, 1)
}
```

Возвращает число дней купонного периода, который содержит дату расчетов, для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
181

# COUPDAYSNC

02.10.2020 • 3 minutes to read

Возвращает число дней от даты расчетов до следующей даты выплаты купонного дохода.

## Синтаксис

```
COUPDAYSNC(<settlement>, <maturity>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Число дней от даты расчетов до следующей даты выплаты купонного дохода.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как

прошло 39 448 дней после 30 декабря 1899 г.

- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Даты settlement и maturity усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ
25.01.2011	Дата расчета
15.11.2011	Дата погашения
2	Полугодовой купон (см. выше)
1	База по факту/по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    COUPDAYSNC(
        DATE(2011,1,25),
        DATE(2011,11,15),
        2,
        1
    )
}
```

Возвращает число дней от даты расчетов до следующей даты выплаты купонного дохода для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
110

# COUPNCD

02.10.2020 • 3 minutes to read

Возвращает следующую дату выплаты купонного дохода после даты расчетов.

## Синтаксис

```
COUPNCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Следующая дата выплаты купонного дохода после даты расчетов.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как

прошло 39 448 дней после 30 декабря 1899 г.

- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Даты settlement и maturity усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement ≥ maturity;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ
25.01.2011	Дата расчета
15.11.2011	Дата погашения
2	Полугодовой купон (см. выше)
1	База по факту/по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    COUPNCD(2011,1,25), 2011,11,15, 2, 1)
}
```

Возвращает дату следующей выплаты купонного дохода после даты расчетов для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
15.05.2011, 00:00:00

# COUPNUM

02.10.2020 • 3 minutes to read

Возвращает количество купонов между датой сопоставления и датой погашения, округленное до ближайшего целого купона.

## Синтаксис

```
COUPNUM(<settlement>, <maturity>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Количество купонов между датой расчета и датой погашения.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в

вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.

- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Даты settlement и maturity усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ
25 января 07 г.	Дата расчета
"15-Nov-08"	Дата погашения
2	Полугодовой купон (см. выше)
1	База по факту/по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    COUPNUM(DATE(2007,1,25), DATE(2008,11,15), 2, 1)
}
```

Возвращает число выплат купона по облигации с указанными выше терминами.

[ЗНАЧЕНИЕ]
4



# COUPPCD

02.10.2020 • 3 minutes to read

Возвращает предыдущую купонную дату до даты расчетов.

## Синтаксис

```
COUPPCD(<settlement>, <maturity>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Предыдущая дата выплаты купонного дохода до даты расчетов.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как

прошло 39 448 дней после 30 декабря 1899 г.

- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Даты settlement и maturity усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ
25.01.2011	Дата расчета
15.11.2011	Дата погашения
2	Полугодовой купон (см. выше)
1	База по факту/по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    COUPPCD(2011,1,25), 2011,11,15, 2, 1)
}
```

Возвращает предыдущую дату выплаты купонного дохода перед датой расчетов для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
15.11.2010, 00:00:00

# CUMIPMT

02.10.2020 • 3 minutes to read

Возвращает совокупный процент, уплаченный по кредиту между значениями start\_period и end\_period.

## Синтаксис

```
CUMIPMT(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка.
nper	Общее число периодов платежей.
pv	Текущее значение.
start_period	Первый период в вычислении. Значение должно быть от 1 до end_period (включительно).
end_period	Последний период в вычислении. Значение должно находиться между start_period и nper (включительно).
тип	Время платежа. Допустимые значения перечислены ниже в этой таблице.

Параметр type принимает следующие значения:

ТИП	ВРЕМЕННЫЕ СВОЙСТВА
0 (ноль)	Платеж в конце периода
1	Платеж в начале периода

## Возвращаемое значение

Совокупный процент, выплаченный за указанный период.

## Remarks

- Для указания rate и nper должны использоваться одинаковые единицы. При ежемесячных платежах по кредиту на четыре года с годовой процентной ставкой 10 % вычислите rate по формуле  $0,1/12$ , а nper — по формуле  $4*12$ . Если вы вносите ежегодные платежи по одному кредиту, rate будет составлять 0,1, а nper — 4.
- Значения start\_period, end\_period и type округляются до ближайшего целого числа.
- Возвращается ошибка, если:

- rate  $\leq$  0;
  - nper < 1;
  - pv  $\leq$  0;
  - start\_period < 1 или start\_period > end\_period;
  - end\_period < start\_period или end\_period > nper;
  - type является любым числом, кроме 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

DATA	ОПИСАНИЕ
9 %	Годовая процентная ставка
30	Лет кредита
125 000	Текущее значение

### Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    CUMIPMT(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

Возвращает общий процент, уплаченный за второй год выплат, период с 13 до 24, если платежи выполняются в начале каждого месяца.

[ЗНАЧЕНИЕ]
-11052,3395838718

### Пример 2

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    CUMIPMT(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

Возвращает проценты, уплаченные за один платеж в первый месяц, если платеж выполняется в конце месяца.

[ЗНАЧЕНИЕ]
-937,5

# CUMPRINC

02.10.2020 • 3 minutes to read

Возвращает совокупную основную сумму, уплаченную по кредиту между значениями start\_period и end\_period.

## Синтаксис

```
CUMPRINC(<rate>, <nper>, <pv>, <start_period>, <end_period>, <type>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка.
nper	Общее число периодов платежей.
pв	Текущее значение.
start_period	Первый период в вычислении. Значение должно быть от 1 до end_period (включительно).
end_period	Последний период в вычислении. Значение должно находиться между start_period и nper (включительно).
тип	Время платежа. Допустимые значения перечислены ниже в этой таблице.

Параметр type принимает следующие значения:

ТИП	ВРЕМЕННЫЕ СВОЙСТВА
0 (ноль)	Платеж в конце периода
1	Платеж в начале периода

## Возвращаемое значение

Совокупная основная сумма, выплаченная за указанный период.

## Remarks

- Для указания rate и nper должны использоваться одинаковые единицы. При ежемесячных платежах по кредиту на четыре года с годовой процентной ставкой 10 % вычислите rate по формуле  $0,1/12$ , а nper — по формуле  $4*12$ . Если вы вносите ежегодные платежи по одному кредиту, rate будет составлять 0,1, а nper — 4.
- Значения start\_period, end\_period и type округляются до ближайшего целого числа.

- Возвращается ошибка, если:
  - $\text{rate} \leq 0$ ;
  - $\text{nper} < 1$ ;
  - $\text{pv} \leq 0$ ;
  - $\text{start\_period} < 1$  или  $\text{start\_period} > \text{end\_period}$ ;
  - $\text{end\_period} < \text{start\_period}$  или  $\text{end\_period} > \text{nper}$ ;
  - $\text{type}$  является любым числом, кроме 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

DATA	ОПИСАНИЕ
9 %	Годовая процентная ставка
30	Срок в годах
125 000	Текущее значение

### Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    CUMPRINC(0.09/12, 30*12, 125000, 13, 24, 1)
}
```

Возвращает общую основную сумму, уплаченную за второй год выплат, период с 13 до 24, если платежи выполняются в начале каждого месяца.

[ЗНАЧЕНИЕ]
-927,153472378062

### Пример 2

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    CUMPRINC(0.09/12, 30*12, 125000, 1, 1, 0)
}
```

Возвращает основную сумму, уплаченную за один платеж в первый месяц, если платеж выполняется в конце месяца.

[ЗНАЧЕНИЕ]
-68,2782711809784

# DB

02.10.2020 • 3 minutes to read

Возвращает значение, указывающее амортизацию ресурса за указанный период методом фиксированного уменьшения остатка.

## Синтаксис

```
DB(<cost>, <salvage>, <life>, <period>[, <month>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
cost	Начальная стоимость актива.
salvage	Стоимость в конце периода амортизации (иногда называется остаточной стоимостью актива). Может иметь значение 0.
данные	Количество периодов, в течение которых производится амортизация актива (иногда называется полезным сроком службы актива).
period	Период, для которого вычисляется амортизация. Период должен использовать те же единицы, что и срок службы. Значение должно находиться между 1 и life (включительно).
month	(Необязательно) Число месяцев в первом году. Если тип не указан, предполагается, что он равен 12.

## Возвращаемое значение

Амортизация за указанный период.

## Remarks

- Метод фиксированного сокращения остатка рассчитывает амортизацию по фиксированной ставке. Для вычисления амортизации за период функция DB использует следующую формулу:

$$DB(\text{cost} - \text{общая амортизация за предыдущие периоды}) \times \text{rate}$$

где:

- $\text{rate} = 1 - (\frac{\text{salvage}}{\text{cost}})^{\frac{1}{\text{life}}}$ , округленное для трех десятичных знаков
- Для первого и последнего периода амортизация является особым случаем.
  - Для первого периода база данных использует следующую формулу:

$$\frac{\text{cost} \times \text{rate} \times \text{month}}{12}$$

- Для последнего периода база данных использует следующую формулу:

$$\frac{(\text{cost} - \text{общая амортизация за предыдущие периоды}) \times \text{rate}}{(12 - \text{month})^{12}}$$

- Значения period и month округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - $\text{cost} < 0$ ;
  - $\text{salvage} < 0$ ;
  - $\text{life} < 1$ ;
  - $\text{period} < 1$  или  $\text{period} > \text{life}$ ;
  - $\text{месяц} < 1$  или  $\text{месяц} > 12$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Примеры

### Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    DB(1000000, 0, 6, 1, 2)
}
```

Возвращает значение амортизации актива за последние два месяца года. Предполагается, что он будет стоить 1 000 000 долл. США через 6 лет.

**[ЗНАЧЕНИЕ]**

166666,666666667

### Пример 2

В следующем примере вычисляется общая сумма амортизации всех активов за разные годы на протяжении срока службы. В этом примере первый год включает 7 месяцев амортизации, а последний год включает только 5 месяцев.

```
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)
    )
)
```



# DDB

02.10.2020 • 3 minutes to read

Возвращает значение, указывающее амортизацию ресурса за указанный период методом двойного уменьшения остатка (или другим методом, определенным пользователем).

## Синтаксис

```
DDB(<cost>, <salvage>, <life>, <period>[, <factor>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
cost	Начальная стоимость актива.
salvage	Стоимость в конце периода амортизации (иногда называется остаточной стоимостью актива). Может иметь значение 0.
данные	Количество периодов, в течение которых производится амортизация актива (иногда называется полезным сроком службы актива).
period	Период, для которого вычисляется амортизация. Период должен использовать те же единицы, что и срок службы. Значение должно находиться между 1 и life (включительно).
Показатель	(Необязательно) Коэффициент сокращения остатка. Если значение factor не указано, предполагается, что оно равно 2 (метод двойного сокращения остатка).

## Возвращаемое значение

Амортизация за указанный период.

## Remarks

- Метод двойного сокращения остатка рассчитывает амортизацию с повышенным коэффициентом. Самое большое значение амортизации приходится на первый период, а в последующих периодах скорость снижается. Для вычисления амортизации за период DDB использует следующую формулу:

$$\text{Min}((\text{cost} - \text{total depreciation from prior periods}) \times \frac{\text{factor}}{\text{life}}), (\text{cost} - \text{salvage} - \text{total depreciation from prior periods}))$$

- Если вы не хотите использовать метод двойного сокращения остатка, измените коэффициент.
- Используйте функцию VDB, если хотите перейти на метод равномерной амортизации, когда амортизация превышает вычисление снижающегося остатка.
- period округляется до ближайшего целого числа.

- Возвращается ошибка, если:
  - $\text{cost} < 0$ ;
  - $\text{salvage} < 0$ ;
  - $\text{life} < 1$ ;
  - $\text{period} < 1$  или  $\text{period} > \text{life}$ ;
  - $\text{factor} \leq 0$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Примеры

### Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    DDB(1000000, 0, 10, 5, 1.5)
}
```

Возвращает значение амортизации актива за 5<sup>th</sup> лет. Предполагается, что он будет стоить \0 долл. США через 10 лет. В этих расчетах используется коэффициент 1,5.

**[ЗНАЧЕНИЕ]**

78300,9375

### Пример 2

В следующем примере вычисляется общая сумма амортизации всех активов за разные годы на протяжении срока службы. В этом вычислении используется коэффициент по умолчанию 2 (метод двойного сокращения остатка).

```
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])
        )
    )
```

# DISC

02.10.2020 • 3 minutes to read

Возвращает процентную ставку для ценной бумаги.

## Синтаксис

```
DISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
pr	Цена ценной бумаги на \100 долл. США номинальной стоимости.
redemption	Стоимость выкупа ценной бумаги на \100 долл. США номинальной стоимости.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Скидка.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2018 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2018 года, датой расчетов — 1 июля 2018 г., а датой погашения — 1 января 2048 г., то есть через 30 лет после даты выпуска 1 января 2018 года.
- DISC рассчитывается следующим образом:  

$$\text{DISC} = \frac{\text{redemption} - \text{par}}{\text{redemption}} \times \frac{\text{B}}{\text{DSM}}$$

где:

  - $\text{B}$  — число дней в году в зависимости от базы года.
  - $\text{DSM}$  — число дней между settlement и maturity.
- Даты settlement и maturity усекаются до целых чисел.
- basis округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - pr  $\leq$  0;
  - redemption  $\leq$  0;
  - 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

DATA	ОПИСАНИЕ
01.07.2018	Дата расчета
01.01.2048	Дата погашения
97,975	Цена
100	Выкупная стоимость
1	База по факту/по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    DISC( DATE(2018,7,1), DATE(2048,1,1), 97.975, 100, 1)
}
```

Возвращает процентную ставку для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,000686384169121348

# DOLLARDE

02.10.2020 • 2 minutes to read

Преобразует цену в долларах, выраженную в виде целого числа и дробной части, например 1,02, в цену в долларах, выраженную в виде десятичного числа. Дробные числа в долларах иногда используются для цен на ценные бумаги.

## Синтаксис

```
DOLLARDE(<fractional_dollar>, <fraction>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
fractional_dollar	Число, выраженное в виде целой части и дробной части, разделенных десятичным символом.
дробь	Целое число, используемое в знаменателе дробной части.

## Возвращаемое значение

Десятичное значение *fractional\_dollar*.

## Remarks

- Дробная часть значения делится на указанное целое число. Например, если нужно, чтобы цена выражалась с точностью до 1/16 доллара, разделите дробную часть на 16. В этом случае 1,02 представляет \1,125 долл. США (\1 долл. США + 2/16 = \1,125 долл. США).
- Значение fraction округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - fraction < 1.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    DOLLARDE(1.02, 16)
}
```

Возвращает 1,125, десятичную цену исходной дробной цены, 1,02, которая читается как 1 и 2/16. Так как дробное значение равно 16, цена имеет точность в 1/16 доллара.



# DOLLARFR

02.10.2020 • 2 minutes to read

Преобразует цену в долларах, выраженную в виде десятичного числа, в цену в долларах, выраженную в виде целой части и дробной части, например 1,02. Дробные числа в долларах иногда используются для цен на ценные бумаги.

## Синтаксис

```
DOLLARFR(<decimal_dollar>, <fraction>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
decimal_dollar	Десятичное число.
дробь	Целое число, используемое в знаменателе дробной части.

## Возвращаемое значение

Дробное значение *decimal\_dollar*, выраженное в виде целой части и дробной части.

## Remarks

- Значение fraction округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - fraction < 1.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    DOLLARFR(1.125, 16)
}
```

Возвращает 1,02, которое читается как 1 и 2/16 и соответствует дробному варианту исходной десятичной цены — 1,125. Так как дробное значение равно 16, цена имеет точность в 1/16 доллара.



# DURATION

02.10.2020 • 3 minutes to read

Возвращает дюрацию Маколея для предполагаемой номинальной стоимости \100 долл. США. Дюрация определяется как взвешенное среднее от текущей стоимости денежных потоков и используется как мера реагирования цены облигаций на изменения доходности.

## Синтаксис

```
DURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
coupon	Годовая купонная ставка ценной бумаги.
yld	Годовая доходность ценной бумаги.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Дюрация Маколея.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Даты settlement и maturity усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - coupon  $< 0$ ;
  - yld  $< 0$ ;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ
01.07.2018	Дата расчета
01.01.2048	Дата погашения
8,0 %	Процентный купон
9,0 %	Процентный доход
2	Периодичность — раз в полгода (см. выше)
1	База по факту/по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    DURATION(
        DATE(2018,7,1),
        DATE(2048,1,1),
        0.08, 0.09, 2, 1
    )
}
```

Возвращает дюрацию Маколея для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
10,9191452815919

# EFFECT

02.10.2020 • 2 minutes to read

Возвращает действительную годовую процентную ставку с учетом номинальной годовой процентной ставки и числа составляющих периодов в год.

## Синтаксис

```
EFFECT(<nominal_rate>, <npery>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
nominal_rate	Номинальная процентная ставка.
npery	Число составляющих периодов в год.

## Возвращаемое значение

Фактическая годовая процентная ставка.

## Remarks

- EFFECT рассчитывается следующим образом:

$$\text{EFFECT} = \text{bigg}(1 + \frac{\text{nominal\_rate}}{\text{npery}} \text{bigg})^{\text{npery}} - 1$$

- npery округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - nominal\_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

DATA	ОПИСАНИЕ
5,25 %	Номинальная процентная ставка
4	Число составляющих периодов в год

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
  EFFECT(0.0525, 4)
}
```

Возвращает фактическую процентную ставку на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,0535426673707584

# FV

02.10.2020 • 3 minutes to read

Вычисляет будущую стоимость инвестиции на основе постоянной процентной ставки. Функцию FV можно использовать с периодическими, постоянными платежами и (или) одной общей суммой.

## Синтаксис

```
FV(<rate>, <nper>, <pmt>[, <pv>[, <type>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка за период.
nper	Общее число периодов платежей за аннуитет.
pmt	Платеж, производимый в каждый период, который не может измениться в течение аннуитета. Как правило, pmt включает в себя основную сумму и проценты, а не другие сборы или налоги.
pv	(Необязательно) Текущее значение, или общая сумма ряда будущих платежей на текущий момент. Если значение pv не указано, предполагается, что оно равно BLANK.
тип	(Необязательно) Номер 0 или 1, указывающий срок платежа. Если тип не указан, предполагается, что он равен 0. Допустимые значения перечислены ниже в этой таблице.

Параметр type принимает следующие значения:

ЗАДАТЬ ТИП РАВНЫМ	ЕСЛИ ПЛАТЕЖИ ДОЛЖНЫ ВНОСИТЬСЯ
0 или не указано	В конце периода
1	В начале периода

**Примечание.** Более полное описание аргументов в функции FV и дополнительные сведения о функциях аннуитета см. в описании функции PV.

## Возвращаемое значение

Будущее значение инвестиции.

## Remarks

- Для указания rate и nper должны использоваться одинаковые единицы. При ежемесячных платежах

по кредиту на четыре года с годовой процентной ставкой 12 % вычислите rate по формуле 0,12/12, а nper — по формуле 4\*12. Если вы вносите ежегодные платежи по одному кредиту, rate будет составлять 0,12, а nper — 4.

- Для всех аргументов вносимые вами платежи, например депозит для накопления, представлены отрицательным числом; полученные деньги, например дивиденды, представляются положительным числом.
- type округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - nper < 1;
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

DATA	ОПИСАНИЕ
6 %	Годовая процентная ставка
10	Число платежей
-200	Сумма платежа
-500	Текущее значение
1	Оплата происходит в начале периода (0 означает, что платеж производится в конце периода)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    FV(0.06/12, 10, -200, -500, 1)
}
```

Возвращает будущее значение инвестиции с использованием указанных выше терминов.

[ЗНАЧЕНИЕ]
2581,40337406012

# INTRATE

02.10.2020 • 3 minutes to read

Возвращает процентную ставку для полностью инвестированной ценной бумаги.

## Синтаксис

```
INTRATE(<settlement>, <maturity>, <investment>, <redemption>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
investment	Сумма, инвестированная в ценную бумагу.
redemption	Сумма, которая будет получена при погашении.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Процентная ставка.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как



прошло 39 448 дней после 30 декабря 1899 г.

- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.

- INTRATE рассчитывается следующим образом:

$$\text{INTRATE} = \frac{\text{redemption} - \text{investment}}{\text{investment}} \times \frac{\text{B}}{\text{DIM}}$$

где:

- $\text{B}$  — число дней в году в зависимости от базы года.
- $\text{DIM}$  — число дней от settlement до maturity.
- Даты settlement и maturity усекаются до целых чисел.
- basis округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - investment  $\leq 0$ ;
  - redemption  $\leq 0$ ;
  - 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

DATA	ОПИСАНИЕ
15.02.2008	Дата расчета
15.05.2008	Дата погашения
\1 000 000 долл. США	Investment
\1 014 420 долл. США	Выкупная стоимость
2	База по факту/360

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    INTRATE(DATE(2008,2,15), DATE(2008,5,15), 1000000, 1014420, 2)
}
```

Возвращает процентную ставку для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,05768

# IPMT

02.10.2020 • 3 minutes to read

Возвращает платеж по процентам за заданный период для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.

## Синтаксис

```
IPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка за период.
per	Период, для которого требуется найти процент. Значение должно находиться между 1 и nper (включительно).
nper	Общее число периодов платежей за аннуитет.
pв	Текущая стоимость или общая сумма ряда будущих платежей на текущий момент.
fv	(Необязательно) Будущая стоимость или денежный остаток, которого вы хотите достичь после последнего платежа. Если значение fv не указано, предполагается, что оно равно BLANK.
тип	(Необязательно) Номер 0 или 1, указывающий срок платежа. Если тип не указан, предполагается, что он равен 0. Допустимые значения перечислены ниже в этой таблице.

Параметр type принимает следующие значения:

ЗАДАТЬ ТИП РАВНЫМ	ЕСЛИ ПЛАТЕЖИ ДОЛЖНЫ ВНОСИТЬСЯ
0 или не указано	В конце периода
1	В начале периода

## Возвращаемое значение

Оплата по процентам за указанный период.

## Remarks

- Для указания rate и nper должны использоваться одинаковые единицы. При ежемесячных платежах

по кредиту на четыре года с годовой процентной ставкой 12 % вычислите rate по формуле 0,12/12, а nper — по формуле 4\*12. Если вы вносите ежегодные платежи по одному кредиту, rate будет составлять 0,12, а nper — 4.

- Для всех аргументов вносимые вами платежи, например депозит для накопления, представлены отрицательным числом; полученные деньги, например дивиденды, представляются положительным числом.
- type округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - 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

DATA	ОПИСАНИЕ
10,00 %	Годовой процент
3	Лет кредита
\8000 долл. США	Текущая стоимость кредита

### Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    IPMT(0.1/12, 1, 3*12, 8000)
}
```

Возвращает месячный процент, подлежащий уплате в первом месяце кредита, на основе указанных выше условий.

[ЗНАЧЕНИЕ]
-66,6666666666667

### Пример 2

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    IPMT(0.1, 3, 3, 8000)
}
```

Возвращает годовой процент, подлежащий уплате за последний год кредита, на основе указанных выше условий, если платежи вносятся ежегодно.

[ЗНАЧЕНИЕ]
–292,447129909366

# ISPMT

02.10.2020 • 4 minutes to read

Вычисляет проценты, выплаченные (или полученные) за указанный период кредита (или инвестиции) с равными платежами по основной сумме.

## Синтаксис

```
ISPMT(<rate>, <per>, <nper>, <pv>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка для инвестиций.
per	Период, для которого требуется найти процент. Значение должно находиться между 0 и nper-1 (включительно).
nper	Общее число периодов платежей для инвестиции.
pv	Текущая стоимость инвестиции. Для кредита pv — это сумма кредита.

## Возвращаемое значение

Выплаченный (или полученный) процент за указанный период.

## Remarks

- Для указания rate и nper должны использоваться одинаковые единицы. При ежемесячных платежах по кредиту на четыре года с годовой процентной ставкой 12 % вычислите rate по формуле 0,12/12, а nper — по формуле 4\*12. Если вы вносите ежегодные платежи по одному кредиту, rate будет составлять 0,12, а nper — 4.
- Для всех аргументов вносимые вами платежи, например депозит для накопления или снятие со счета, представлены отрицательным числом; полученные деньги, например дивиденды и другие депозиты, представляются положительным числом.
- ISPMT считает каждый период с нуля, а не с единицы.
- В большинстве кредитов используется график выплат с равными периодическими платежами. Функция IPMT возвращает платеж по процентам за указанный период для данного типа кредита.
- Для некоторых кредитов используется график выплат с равными платежами по основной сумме. Функция ISPMT возвращает платеж по процентам за указанный период для данного типа кредита.
- Возвращается ошибка, если:
  - nper = 0.

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

## Пример

DATA	ОПИСАНИЕ
\4000 долл. США	Текущая стоимость
4	Количество периодов
10 %	Тариф

Чтобы продемонстрировать, когда следует использовать ISPMT, в таблице амортизационных отчислений ниже используется график с равными платежами по основной сумме на основе указанных выше условий. Оплата процентов за каждый период равна произведению ставки и непоплаченного остатка за предыдущий период. А платеж за каждый период равен равному платежу по основной сумме плюс процент за этот период.

ПЕРИОД	ПЛАТЕЖ ПО ОСНОВНОЙ СУММЕ	ПЛАТЕЖ ПО ПРОЦЕНТАМ	ОБЩИЙ ПЛАТЕЖ	BALANCE
				4000,00
1	1000,00	400,00	1400,00	3000,00
2	1000,00	300,00	1300,00	2000,00
3	1000,00	200,00	1200,00	1000,00
4	1000,00	100,00	1100,00	0,00

Рассмотрим следующий запрос DAX:

```
DEFINE
VAR NumPaymentPeriods = 4
VAR PaymentPeriods = GENERATESERIES(0, NumPaymentPeriods-1)
EVALUATE
ADDCOLUMNS (
    PaymentPeriods,
    "Interest Payment",
    ISPMT(0.1, [Value], NumPaymentPeriods, 4000)
)
```

Возвращает процент, выплаченный за каждый период, используя график равных платежей по основной сумме и указанные выше условия. Отрицательные значения показывают, что процент выплачен, а не получен.

[ЗНАЧЕНИЕ]	[INTEREST PAYMENT]
0	-400
1	-300

[ЗНАЧЕНИЕ]	[INTEREST PAYMENT]
2	-200
3	-100



# MDURATION

02.10.2020 • 3 minutes to read

Возвращает измененную дюрацию Маколея для ценной бумаги с предполагаемой номинальной стоимостью \100 долларов США.

## Синтаксис

```
MDURATION(<settlement>, <maturity>, <coupon>, <yld>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
coupon	Годовая купонная ставка ценной бумаги.
yld	Годовая доходность ценной бумаги.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Измененная дюрация Маколея.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Измененная дюрация определяется следующим образом:
$$\text{MDURATION} = \frac{\text{DURATION}}{1 + (\frac{\text{Рыночная доходность}}{\text{Число купонных выплат в год}})}$$
- Даты settlement и maturity усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - coupon  $< 0$ ;
  - yld  $< 0$ ;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ
01.01.2008	Дата расчета
01.01.2016	Дата погашения
8 %	Процентный купон
9 %	Процентный доход
2	Периодичность — раз в полгода (см. выше)
1	База по факту/по факту (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    MDURATION(DATE(2008,1,1), DATE(2016,1,1), 0.08, 0.09, 2, 1)
}
```

Возвращает измененную дюрацию Маколея для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
5,73566981391884

# NOMINAL

02.10.2020 • 2 minutes to read

Возвращает номинальную годовую процентную ставку с учетом фактической ставки и числа составляющих периодов в год.

## Синтаксис

```
NOMINAL(<effect_rate>, <npery>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
effect_rate	Фактическая процентная ставка.
npery	Число составляющих периодов в год.

## Возвращаемое значение

Номинальная годовая процентная ставка.

## Remarks

- Связь между NOMINAL и EFFECT выражается следующим уравнением:

$$\text{EFFECT} = \text{Big}(1 + \frac{\text{nominal\_rate}}{\text{npery}} \text{Big})^{\text{npery}} - 1$$

- npery округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - 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

DATA	ОПИСАНИЕ
5,3543 %	Фактическая процентная ставка
4	Число составляющих периодов в год

Рассмотрим следующий запрос DAX:

```
EVALUATE
```

```
{  
  NOMINAL(0.053543, 4)  
}
```

Возвращает номинальную процентную ставку на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,052500319868356

# NPER

02.10.2020 • 2 minutes to read

Возвращает число периодов для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.

## Синтаксис

```
NPER(<rate>, <pmt>, <pv>[, <fv>[, <type>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка за период.
pmt	Платеж, производимый в каждый период, который не может измениться в течение аннуитета. Как правило, pmt включает в себя основную сумму и проценты, а не другие сборы или налоги.
pв	Текущая стоимость или общая сумма ряда будущих платежей на текущий момент.
fv	(Необязательно) Будущая стоимость или денежный остаток, которого вы хотите достичь после последнего платежа. Если значение fv не указано, предполагается, что оно равно BLANK.
тип	(Необязательно) Номер 0 или 1, указывающий срок платежа. Если тип не указан, предполагается, что он равен 0. Допустимые значения перечислены ниже в этой таблице.

Параметр type принимает следующие значения:

ЗАДАТЬ ТИП РАВНЫМ	ЕСЛИ ПЛАТЕЖИ ДОЛЖНЫ ВНОСИТЬСЯ
0 или не указано	В конце периода
1	В начале периода

## Возвращаемое значение

Число периодов для инвестиции.

## Remarks

- type 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

DATA	ОПИСАНИЕ
12 %	Годовая процентная ставка
–100	Платеж, производимый в каждый период
–1000	Текущая стоимость
10000	Будущая стоимость
1	Оплата происходит в начале периода (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    NPER(0.12/12, -100, -1000, 10000, 1)
}
```

Возвращает число периодов для инвестиции с указанными выше условиями.

[ЗНАЧЕНИЕ]
59,6738656742946

# ODDFPRICE

02.10.2020 • 7 minutes to read

Возвращает цену за \100 долл. США номинальной стоимости для ценной бумаги с нестандартным (коротким или длинным) первым периодом.

## Синтаксис

```
ODDFPRICE(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
проблема	Дата выпуска ценной бумаги.
first_coupon	Дата первой выплаты купонного дохода по ценной бумаге.
rate	Процентная ставка ценной бумаги.
yld	Годовая доходность ценной бумаги.
redemption	Стоимость выкупа ценной бумаги на \100 долл. США номинальной стоимости.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический



БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Цена за 100 долл. США номинальной стоимости.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- ODDFPRICE рассчитывается следующим образом:

### Первая выплата купонного дохода за нестандартный короткий период:

$$\text{ODDFPRICE} = \frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N - 1 + \frac{\text{DSC}}{\text{E}})}} + \frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{DFC}}{\text{E}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\frac{\text{DSC}}{\text{E}})}} + \frac{\sum_{k=2}^N \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k - 1 + \frac{\text{DSC}}{\text{E}})}}}{100 \times \frac{\text{rate}}{\text{frequency}}} \times \frac{\text{A}}{\text{E}}$$

где:

- $\text{A}$  — число дней с начала купонного периода до даты расчетов (дни начисления).
- $\text{DSC}$  — число дней от даты расчетов до следующей даты выплаты купонного дохода.
- $\text{DFC}$  — число дней от начала первого нестандартного купонного периода до первой даты выплаты купонного дохода.
- $\text{E}$  — число дней в купонном периоде.
- $\text{N}$  — количество купонов между датой расчета и датой выкупа. (Если это число содержит дробную часть, оно округляется до следующего целого числа.)

### Первая выплата купонного дохода за нестандартный длинный период:

$$\text{ODDFPRICE} = \frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\text{N} + \text{N}_q + \frac{\text{DSC}}{\text{E}})}} + \frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \text{Big}[\sum_{i=1}^{\text{NC}} \frac{\text{DC}_i}{\text{NL}_i}]}{(1 + \frac{\text{yld}}{\text{frequency}})^{(\text{N}_q + \frac{\text{DSC}}{\text{E}})}} + \frac{\sum_{k=1}^{\text{N}} \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(k - \text{N}_q + \frac{\text{DSC}}{\text{E}})}}}{100 \times \frac{\text{rate}}{\text{frequency}}} \times \frac{\sum_{i=1}^{\text{NC}} \frac{\text{A}_i}{\text{NL}_i}}{\text{Big}}$$

где:

- $\text{\text{A}}_{\text{\text{i}}}$  — число дней от начала  $i^{\text{th}}$  или последнего купонного квазипериода в нестандартном периоде.
- $\text{\text{DC}}_{\text{\text{i}}}$  — число дней от даты выпуска до первого купонного квазипериода ( $i = 1$ ) или число дней в купонном квазипериоде ( $i = 2, \dots, i = \text{\text{NC}}$ ).
- $\text{\text{DSC}}$  — число дней от даты расчетов до следующей даты выплаты купонного дохода.
- $\text{\text{E}}$  — число дней в купонном периоде.
- $\text{\text{N}}$  — количество купонов между первой реальной датой выплаты купонного дохода и датой выкупа. (Если это число содержит дробную часть, оно округляется до следующего целого числа.)
- $\text{\text{NC}}$  — число купонных квазипериодов в нестандартном периоде. (Если это число содержит дробную часть, оно округляется до следующего целого числа.)
- $\text{\text{NL}}_{\text{\text{i}}}$  — обычная длина в днях полного периода  $i^{\text{th}}$  или последнего купонного квазипериода в нестандартном периоде.
- $\text{\text{N}}_{\text{\text{q}}}$  — число целых купонных квазипериодов между датой расчетов и первой датой выплаты купонного дохода.
- Значения settlement, maturity, issue и first\_coupon усекаются до целых чисел.
- Значения basis и frequency округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - значение settlement, maturity, issue или first\_coupon не является допустимой датой;
  - условие  $\text{maturity} > \text{first\_coupon} > \text{settlement} > \text{issue}$  не удовлетворяется;
  - $\text{rate} < 0$ ;
  - $\text{yld} < 0$ ;
  - $\text{redemption} \leq 0$ ;
  - frequency является любым числом, кроме 1, 2 или 4;
  - $\text{basis} < 0$  или  $\text{basis} > 4$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

DATA	ОПИСАНИЕ АРГУМЕНТА
11.11.2008	Дата расчета
01.03.2021	Дата погашения
15.10.2008	Дата выпуска
01.03.2009	Дата первой выплаты купонного дохода
7,85 %	Процентный купон
6,25 %	Процентный доход
\100,00 долл. США	Выкупная стоимость
2	Периодичность — раз в полгода

DATA	ОПИСАНИЕ АРГУМЕНТА
1	База по факту/по факту

Рассмотрим следующий запрос 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 долл. США номинальной стоимости для ценной бумаги с нестандартным (коротким или длинным) первым периодом с использованием указанных выше условий.

[ЗНАЧЕНИЕ]
113,597717474079

# ODDFYIELD

02.10.2020 • 4 minutes to read

Возвращает доходность ценной бумаги с нестандартным (коротким или длинным) первым периодом.

## Синтаксис

```
ODDFYIELD(<settlement>, <maturity>, <issue>, <first_coupon>, <rate>, <pr>, <redemption>, <frequency>[,  
<basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
проблема	Дата выпуска ценной бумаги.
first_coupon	Дата первой выплаты купонного дохода по ценной бумаге.
rate	Процентная ставка ценной бумаги.
pr	Цена ценной бумаги.
redemption	Стоимость выкупа ценной бумаги на \100 долл. США номинальной стоимости.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Доходность ценной бумаги.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Функция ODDFYIELD вычисляется итеративно. Она использует метод Ньютона, основанный на формуле, применяемой для функции ODDFPRICE. Доходность изменяется в течение 100 итераций до тех пор, пока предполагаемая цена с учетом доходности не будет близка к цене. Формулу, используемую функцией ODDFYIELD, см. в описании функции ODDFPRICE.
- Значения settlement, maturity, issue и first\_coupon усекаются до целых чисел.
- Значения basis и frequency округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - значение settlement, maturity, issue или first\_coupon не является допустимой датой;
  - условие maturity > first\_coupon > settlement > issue не удовлетворяется;
  - rate < 0;
  - pr ≤ 0;
  - redemption ≤ 0;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ АРГУМЕНТА
11 ноября 2008 г.	Дата расчета
1 марта 2021 г.	Дата погашения
15 октября 2008 г.	Дата выпуска

DATA	ОПИСАНИЕ АРГУМЕНТА
1 марта 2009 г.	Дата первой выплаты купонного дохода
5,75 %	Процентный купон
84,50	Цена
100	Выкупная стоимость
2	Периодичность — раз в полгода
0	База 30/360

Рассмотрим следующий запрос 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 ценной бумаги с нестандартным (коротким или длинным) первым периодом с использованием указанных выше условий.

[ЗНАЧЕНИЕ]
0,0772455415972989

# ODDLPRICE

02.10.2020 • 4 minutes to read

Возвращает цену за \100 долл. США номинальной стоимости для ценной бумаги с нестандартным (коротким или длинным) последним купонным периодом.

## Синтаксис

```
ODDLPRICE(<settlement>, <maturity>, <last_interest>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
last_interest	Дата последней выплаты купонного дохода по ценной бумаге.
rate	Процентная ставка ценной бумаги.
yld	Годовая доходность ценной бумаги.
redemption	Стоимость выкупа ценной бумаги на \100 долл. США номинальной стоимости.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
4	Европейский стандарт, 30/360

# Возвращаемое значение

Цена за \100 долл. США номинальной стоимости.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Значения settlement, maturity и last\_interest усекаются до целых чисел.
- Значения basis и frequency округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - значение settlement, maturity или last\_interest не является допустимой датой;
  - условие maturity > settlement > last\_interest не удовлетворяется;
  - rate < 0;
  - yld < 0;
  - redemption ≤ 0;
  - frequency является любым числом, кроме 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

Рассмотрим следующий запрос DAX:

DATA	ОПИСАНИЕ АРГУМЕНТА
7 февраля 2008 г.	Дата расчета
15 июня 2008 г.	Дата погашения
15 октября 2007 г.	Дата последней выплаты процентов
3,75 %	Процентный купон
4,05 %	Процентный доход
\100 долл. США	Выкупная стоимость



DATA	ОПИСАНИЕ АРГУМЕНТА
2	Периодичность — раз в полгода
0	База 30/360

```
EVALUATE
{
  ODDLPRICE(
    DATE(2008,2,7),
    DATE(2008,6,15),
    DATE(2007,10,15),
    0.0375,
    0.0405,
    100,
    2,
    0
  )
}
```

Возвращает цену за \100 долл. США номинальной стоимости для ценной бумаги с нестандартным (коротким или длинным) последним купонным периодом с использованием указанных выше условий.

[ЗНАЧЕНИЕ]
99,8782860147213

# ODDLYIELD

02.10.2020 • 5 minutes to read

Возвращает доходность ценной бумаги с нестандартным (коротким или длинным) последним периодом.

## Синтаксис

```
ODDLYIELD(<settlement>, <maturity>, <last_interest>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
last_interest	Дата последней выплаты купонного дохода по ценной бумаге.
rate	Процентная ставка ценной бумаги.
pr	Цена ценной бумаги.
redemption	Стоимость выкупа ценной бумаги на \100 долл. США номинальной стоимости.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр basis принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
4	Европейский стандарт, 30/360

## Возвращаемое значение

Доходность ценной бумаги.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.

- ODDLYIELD рассчитывается следующим образом:

$$\begin{aligned} \text{ODDLYIELD} = & \text{bigg} \left[ \frac{(\text{redemption}) + ((\sum^{\text{NC}}_{i=1} \frac{\text{DC}_i}{\text{NL}_i}) \times \frac{100 \times \text{rate}}{\text{frequency}})}{(\text{par}) + ((\sum^{\text{NC}}_{i=1} \frac{\text{A}_i}{\text{NL}_i}) \times \frac{100 \times \text{rate}}{\text{frequency}})} \right] \times \text{bigg} \left[ \frac{\text{frequency}}{(\sum^{\text{NC}}_{i=1} \frac{\text{A}_i}{\text{NL}_i}) \times \frac{100 \times \text{rate}}{\text{frequency}}} \right] \\ & + ((\sum^{\text{NC}}_{i=1} \frac{\text{A}_i}{\text{NL}_i}) \times \frac{100 \times \text{rate}}{\text{frequency}}) \times \text{bigg} \left[ \frac{\text{frequency}}{(\sum^{\text{NC}}_{i=1} \frac{\text{DSC}_i}{\text{NL}_i})} \right] \end{aligned}$$

где:

- $\text{A}_i$  — общее число дней для  $i^{\text{th}}$  или последнего купонного квазипериода в нестандартном периоде начиная с даты последней выплаты процентов до выкупа.
- $\text{DC}_i$  — число рассчитанных дней в  $i^{\text{th}}$  или последнем купонном квазипериоде, ограниченное длиной фактического купонного периода.
- $\text{NC}$  — число купонных квазипериодов в нестандартном периоде. Если это число содержит дробную часть, оно округляется до следующего целого числа.
- $\text{NL}_i$  — обычная длина в днях периода  $i^{\text{th}}$  или последнего купонного квазипериода в нестандартном купонном периоде.
- Значения settlement, maturity и last\_interest усекаются до целых чисел.
- Значения basis и frequency округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - значение settlement, maturity или last\_interest не является допустимой датой;
  - условие maturity > settlement > last\_interest не удовлетворяется;
  - rate < 0;
  - pr ≤ 0;
  - redemption ≤ 0;
  - frequency является любым числом, кроме 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

Рассмотрим следующий запрос DAX:

DATA	ОПИСАНИЕ АРГУМЕНТА
20.04.2008	Дата расчета
15.06.2008	Дата погашения
24.12.2007	Дата последней выплаты процентов
3,75 %	Процентный купон
\99,875 долл. США	Цена
\100 долл. США	Выкупная стоимость
2	Периодичность — раз в полгода
0	База 30/360

```
EVALUATE
{
  ODDLYIELD(DATE(2008,4,20), DATE(2008,6,15), DATE(2007,12,24), 0.0375, 99.875, 100, 2, 0)
}
```

Возвращает доходность ценной бумаги с нестандартным (коротким или длинным) последним периодом с использованием указанных выше условий.

[ЗНАЧЕНИЕ]
0,0451922356291692

# PDURATION

02.10.2020 • 2 minutes to read

Возвращает количество периодов, необходимых инвестиции для достижения заданной стоимости.

## Синтаксис

```
PDURATION(<rate>, <pv>, <fv>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка за период.
pv	Текущая стоимость инвестиции.
fv	Требуемая будущая стоимость инвестиции.

## Возвращаемое значение

Количество периодов.

## Remarks

- Для расчета PDURATION используется следующее уравнение:

$$\text{PDURATION} = \frac{\log(\text{fv}) - \log(\text{pv})}{\log(1 + \text{rate})}$$

- Возвращается ошибка, если:
  - rate ≤ 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

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    PDURATION(0.025, 2000, 2200)
}
```

Возвращает количество лет, необходимых инвестиции размером 2000 долл. США с годовой доходностью 2,5 % для достижения стоимости 2200 долл. США.

<b>[ЗНАЧЕНИЕ]</b>
3,85986616262266

## Пример 2

Рассмотрим следующий запрос DAX:

<pre>EVALUATE {   PDURATION(0.025/12, 1000, 1200) }</pre>
---

Возвращает количество месяцев, необходимых инвестиции размером \1000 долл. США с годовой доходностью 2,5 % для достижения стоимости \1200 долл. США.

<b>[ЗНАЧЕНИЕ]</b>
87,6054764193714

# PMT

02.10.2020 • 4 minutes to read

Вычисляет платеж по кредиту на основе постоянных платежей и постоянной процентной ставки.

## Синтаксис

```
PMT(<rate>, <nper>, <pv>[, <fv>[, <type>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка по кредиту.
nper	Общее число платежей по кредиту.
pv	Текущая стоимость или общая сумма ряда будущих платежей. Также называется основной суммой.
fv	(Необязательно) Будущая стоимость или денежный остаток, которого вы хотите достичь после последнего платежа. Если значение fv не указано, предполагается, что оно равно BLANK.
тип	(Необязательно) Номер 0 или 1, указывающий срок платежа. Если тип не указан, предполагается, что он равен 0. Допустимые значения перечислены ниже в этой таблице.

Параметр type принимает следующие значения:

ЗАДАТЬ ТИП РАВНЫМ	ЕСЛИ ПЛАТЕЖИ ДОЛЖНЫ ВНОСИТЬСЯ
0 или не указано	В конце периода
1	В начале периода

**Примечание.** Более полное описание аргументов в PMT см. в описании функции PV.

## Возвращаемое значение

Сумма одного платежа по кредиту.

## Remarks

- Сумма платежа, возвращаемая функцией PMT, включает в себя основную сумму и проценты, но не налоги, условные платежи или сборы, иногда связанные с кредитами.
- Для указания rate и nper должны использоваться одинаковые единицы. При ежемесячных платежах по кредиту на четыре года с годовой процентной ставкой 12 % вычислите rate по формуле 0,12/12, а

per — по формуле  $4 \times 12$ . Если вы вносите ежегодные платежи по одному кредиту, rate будет составлять 0,12, а per — 4.

- type округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - $per < 1$ .

**Совет.** Чтобы найти общую сумму, выплачиваемую за весь срок кредита, умножьте возвращенное функцией PMT значение на per.

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

## Примеры

### Пример 1

DATA	ОПИСАНИЕ
8 %	Годовая процентная ставка
10	Число месяцев выплат
\10 000 долл. США	Сумма кредита

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    PMT(0.08/12, 10, 10000, 0, 1)
}
```

Возвращает сумму месячного платежа, подлежащую уплате в начале месяца, для кредита с указанными выше условиями.

[ЗНАЧЕНИЕ]
-1030,16432717797

**Примечание.** 1030,16432717797 — это сумма платежа за период. В результате общая сумма, выплачиваемая за весь срок кредита, примерно равна  $1030,16 \times 10 = \backslash 10\,301,60$  долл. США. Иными словами, выплачивается примерно \301,60 долл. США процентов.

### Пример 2

DATA	ОПИСАНИЕ
6 %	Годовая процентная ставка
18	Число лет выплат
\50 000 долл. США	Сумма кредита

Рассмотрим следующий запрос DAX:



EVALUATE

```
{  
  PMT(0.06/12, 18*12, 0, 50000)  
}
```

**[ЗНАЧЕНИЕ]**

-129,081160867991

Возвращает сумму, которую нужно экономить каждый месяц, чтобы получить \50 000 долл. США по окончании 18 лет согласно приведенным выше условиям.

# PPMT

02.10.2020 • 3 minutes to read

Возвращает платеж по основной сумме за заданный период для инвестиций на основе периодических, постоянных платежей и постоянной процентной ставки.

## Синтаксис

```
PPMT(<rate>, <per>, <nper>, <pv>[, <fv>[, <type>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка по кредиту.
per	Задаёт период. Значение должно находиться между 1 и nper (включительно).
nper	Общее число периодов платежей за аннуитет.
pв	Текущая стоимость — общая сумма ряда будущих платежей.
fv	(Необязательно) Будущая стоимость или денежный остаток, которого вы хотите достичь после последнего платежа. Если значение fv не указано, предполагается, что оно равно BLANK.
тип	(Необязательно) Номер 0 или 1, указывающий срок платежа. Если тип не указан, предполагается, что он равен 0. Допустимые значения перечислены ниже в этой таблице.

Параметр type принимает следующие значения:

ЗАДАТЬ ТИП РАВНЫМ	ЕСЛИ ПЛАТЕЖИ ДОЛЖНЫ ВНОСИТЬСЯ
0 или не указано	В конце периода
1	В начале периода

**Примечание.** Более полное описание аргументов в PPMT см. в описании функции PV.

## Возвращаемое значение

Платеж по основной сумме за заданный период.

## Remarks

- Для указания rate и nper должны использоваться одинаковые единицы. При ежемесячных платежах

по кредиту на четыре года с годовой процентной ставкой 12 % вычислите rate по формуле 0,12/12, а nper — по формуле 4\*12. Если вы вносите ежегодные платежи по одному кредиту, rate будет составлять 0,12, а nper — 4.

- type округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - $per < 1$  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

DATA	ОПИСАНИЕ АРГУМЕНТА
10 %	Годовая процентная ставка
2	Число лет кредита
\2 000,00 долл. США	Сумма кредита

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    PMT(0.1/12, 1, 2*12, 2000.00)
}
```

Возвращает платеж по основной сумме, сделанный в первом месяце кредита, на основе указанных выше условий.

[ЗНАЧЕНИЕ]
-75,6231860083663

## Пример 2

DATA	ОПИСАНИЕ АРГУМЕНТА
8 %	Годовая процентная ставка
10	Число лет кредита
\200 000,00 долл. США	Сумма кредита

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    PMT(0.08, 10, 10, 200000.00)
}
```

Возвращает платеж по основной сумме, сделанный на 10-й год кредита, на основе указанных выше условий.

[ЗНАЧЕНИЕ]
-27598,0534624214

# PRICE

02.10.2020 • 5 minutes to read

Возвращает цену за \100 долл. США номинальной стоимости ценной бумаги с периодической выплатой процентов.

## Синтаксис

```
PRICE(<settlement>, <maturity>, <rate>, <yld>, <redemption>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
rate	Годовая купонная ставка ценной бумаги.
yld	Годовая доходность ценной бумаги.
redemption	Стоимость выкупа ценной бумаги на \100 долл. США номинальной стоимости.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

# Возвращаемое значение

Цена за \100 долл. США номинальной стоимости.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Даты settlement и maturity усекаются до целых чисел.
- Значения basis и frequency округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - rate  $< 0$ ;
  - yld  $< 0$ ;
  - redemption  $\leq 0$ ;
  - frequency является любым числом, кроме 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).

## Внимание!

- Когда  $N > 1$  ( $N$  — количество купонов между датой расчета и датой выкупа), PRICE рассчитывается следующим образом:

$$\text{PRICE} = \frac{\text{redemption}}{(1 + \frac{\text{yld}}{\text{frequency}})^{(N - 1 + \frac{\text{DSC}}{\text{E}})}} + \frac{\sum_{k=1}^N \frac{100 \times \text{rate}}{\text{frequency}} (1 + \frac{\text{yld}}{\text{frequency}})^{(k - 1 + \frac{\text{DSC}}{\text{E}})}}{100 \times \frac{\text{rate}}{\text{frequency}}} \times \frac{\text{A}}{\text{E}}$$

- Когда  $N = 1$  ( $N$  — количество купонов между датой расчета и датой выкупа), PRICE рассчитывается следующим образом:

$$\text{DSR} = \text{E} - \text{A}$$

$$\text{T1} = 100 \times \frac{\text{rate}}{\text{frequency}} + \text{redemption}$$

$$\text{T2} = \frac{\text{yld}}{\text{frequency}} \times \frac{\text{DSR}}{\text{E}} + 1$$

$$\text{T3} = 100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{A}}{\text{E}}$$

$$\text{PRICE} = \frac{\text{T1}}{\text{T2}} - \text{T3}$$

где:

- $\text{DSC}$  — число дней от даты расчетов до следующей даты выплаты купонного дохода.

- $\text{\textit{E}}$  = число дней купонного периода, на который приходится дата расчетов.
- $\text{\textit{A}}$  — число дней с начала купонного периода до даты расчетов.

## Пример

DATA	ОПИСАНИЕ АРГУМЕНТА
15.02.2008	Дата расчета
15.11.2017	Дата погашения
5,75 %	Процент полугодового купона
6,50 %	Процентный доход
\100 долл. США	Выкупная стоимость
2	Периодичность — раз в полгода
0	База 30/360

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    PRICE(
        DATE(2008,2,15),
        DATE(2017,11,15),
        0.0575,
        0.065,
        100,
        2,
        0
    )
}
```

Возвращает цену облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
94,6343616213221

# PRICEDISC

02.10.2020 • 3 minutes to read

Возвращает цену за \100 долл. США номинальной стоимости дисконтной ценной бумаги.

## Синтаксис

```
PRICEDISC(<settlement>, <maturity>, <discount>, <redemption>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
discount	Дисконтная ставка ценной бумаги.
redemption	Стоимость выкупа ценной бумаги на \100 долл. США номинальной стоимости.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Цена за \100 долл. США номинальной стоимости.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в



вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.

- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2018 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2018 года, датой расчетов — 1 июля 2018 г., а датой погашения — 1 января 2048 г., то есть через 30 лет после даты выпуска 1 января 2018 года.

- PRICEDISC рассчитывается следующим образом:

$$\text{PRICEDISC} = \text{redemption} - \text{discount} \times \text{redemption} \times \frac{\text{DSM}}{\text{B}}$$

где:

- $\text{B}$  — число дней в году в зависимости от базы года.
- $\text{DSM}$  — число дней от settlement до maturity.
- Даты settlement и maturity усекаются до целых чисел.
- basis округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - discount  $\leq 0$ ;
  - redemption  $\leq 0$ ;
  - 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

DATA	ОПИСАНИЕ АРГУМЕНТА
16.02.2008	Дата расчета
01.03.2008	Дата погашения
5,25 %	Процентная ставка
\100 долл. США	Выкупная стоимость
2	База по факту/360

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    PRICEDISC(
        DATE(2008,2,16),
        DATE(2008,3,1),
        0.0525,
        100,
        2
    )
}
```

Возвращает цену облигации за \100 долл. США номинальной стоимости на основе указанных выше условий.

[ЗНАЧЕНИЕ]
99,7958333333333

# PRICEMAT

02.10.2020 • 4 minutes to read

Возвращает цену за \100 долл. США номинальной стоимости ценной бумаги, по которой выплачивается процент при погашении.

## Синтаксис

```
PRICEMAT(<settlement>, <maturity>, <issue>, <rate>, <yld>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
проблема	Дата выпуска ценной бумаги.
rate	Процентная ставка ценной бумаги на дату выпуска.
yld	Годовая доходность ценной бумаги.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Цена за \100 долл. США номинальной стоимости.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- PRICEMAT рассчитывается 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:

- $\text{B}$  — число дней в году в зависимости от базы года.
- $\text{DSM}$  — число дней от settlement до maturity.
- $\text{DIM}$  — число дней от issue до maturity.
- $\text{A}$  — число дней от issue до settlement.
- Значения settlement, maturity и issue усекаются до целых чисел.
- basis округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - значение settlement, maturity или issue не является допустимой датой;
  - условие maturity > settlement > issue не удовлетворяется;
  - rate < 0;
  - yld < 0;
  - 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

Рассмотрим следующий запрос DAX:

DATA	ОПИСАНИЕ
15.02.2008	Дата расчета
13.04.2008	Дата погашения
11.11.2007	Дата выпуска
6,10 %	Процент полугодового купона
6,10 %	Процентный доход
0	База 30/360

EVALUATE

```
{  
  PRICEMAT(DATE(2008,2,15), DATE(2008,4,13), DATE(2007,11,11), 0.061, 0.061, 0)  
}
```

Возвращает цену за \100 долл. США номинальной стоимости ценной бумаги на основе указанных выше условий.

**[ЗНАЧЕНИЕ]**

99,9844988755569

# PV

02.10.2020 • 5 minutes to read

Вычисляет текущую стоимость кредита или инвестиции на основе постоянной процентной ставки. PV можно использовать с периодическими постоянными платежами (например, по ипотеке или другому кредиту) и (или) для оценки будущей ценности, которая является целью инвестиции.

## Синтаксис

```
PV(<rate>, <nper>, <pmt>[, <fv>[, <type>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
rate	Процентная ставка за период. Например, если вы взяли кредит на автомобиль под 10 % годовых и вносите ежемесячные платежи, процентная ставка в месяц составляет 0,1/12, или 0,0083. Введите 0,1/12 или 0,0083 в формулу в качестве значения rate.
nper	Общее число периодов платежей за аннуитет. Например, если вы взяли кредит на автомобиль на четыре года и вносите ежемесячные платежи, срок кредита состоит из 4*12 (или 48) периодов. Введите 48 в формулу в качестве значения nper.
pmt	Платеж, производимый в каждый период, который не может измениться в течение аннуитета. Как правило, pmt включает в себя основную сумму и проценты, а не другие сборы или налоги. Например, ежемесячные платежи при кредите на автомобиль на сумму \10 000 долл. США на четыре года под 12 % составят \263,33 долл. США. Введите -263,33 в формулу в качестве значения pmt.
fv	(Необязательно) Будущая стоимость или денежный остаток, которого вы хотите достичь после последнего платежа. Если значение fv не указано, предполагается, что оно равно BLANK. Например, если вы хотите накопить \50 000 долл. США на какую-то цель за 18 лет, будущая стоимость будет составлять \50 000. Затем вы можете взять примерную процентную ставку и рассчитать, сколько необходимо откладывать каждый месяц.
тип	(Необязательно) Номер 0 или 1, указывающий срок платежа. Если тип не указан, предполагается, что он равен 0. Допустимые значения перечислены ниже в этой таблице.

Параметр type принимает следующие значения:

ЗАДАТЬ ТИП РАВНЫМ	ЕСЛИ ПЛАТЕЖИ ДОЛЖНЫ ВНОСИТЬСЯ
0 или не указано	В конце периода
1	В начале периода

## Возвращаемое значение

Текущая стоимость кредита или инвестиции.

## Remarks

- Для указания rate и nper должны использоваться одинаковые единицы. При ежемесячных платежах по кредиту на четыре года с годовой процентной ставкой 12 % вычислите rate по формуле 0,12/12, а nper — по формуле 4\*12. Если вы вносите ежегодные платежи по одному кредиту, rate будет составлять 0,12, а nper — 4.
- Следующие функции применяются к годовым выплатам:
  - CUMIPMT
  - CUMPRINC
  - FV
  - IPMT
  - PMT
  - PPMT
  - PV
  - RATE
  - XIRR
  - XNPV
- Аннуитет — это серия постоянных денежных выплат, производимых за непрерывный период. Например, кредит на автомобиль или ипотека — это аннуитет. Дополнительные сведения см. в описании каждой функции для аннуитета.
- В функциях аннуитета вносимые вами платежи, например депозит для накопления, представлены отрицательным числом; полученные деньги, например дивиденды, представляются положительным числом. Например, депозит в размере 1000 долл. США в банк будет представлен аргументом -1000, если вы вкладчик, и аргументом 1000, если вы представляете банк.
- Один финансовый аргумент разрешается в зависимости от других.
  - Если ставка не равна 0, то:
 
$$\text{pv} \times (1 + \text{rate})^{\text{nper}} + \text{pmt} \times (1 + \text{rate}) \times \text{type} \times \left( \frac{(1 + \text{rate})^{\text{nper}} - 1}{\text{rate}} \right) + \text{fv} = 0$$
  - Если ставка равна 0, то:
 
$$(\text{pmt} \times \text{nper}) + \text{pv} + \text{fv} = 0$$
- type округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - nper < 1;
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в

вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

DATA	ОПИСАНИЕ
\500,00 долл. США	Деньги, выплачиваемые из страхового аннуитета в конце каждого месяца.
8 %	Процентная ставка за выплаченные средства.
20	Количество лет, в течение которых будут выплачиваться средства.

Рассмотрим следующий запрос DAX:

<pre>EVALUATE {     PV(0.08/12, 12*20, 500.00, 0, 0) }</pre>
--

Возвращает текущее значение аннуитета на основе указанных выше условий.

[ЗНАЧЕНИЕ]
-59 777,1458511878



# RATE

02.10.2020 • 3 minutes to read

Возвращает процентную ставку за период в рамках аннуитета. RATE вычисляется путем итерации и может иметь ноль или более решений. Если последовательные результаты функции RATE не совпадают в рамках 0,0000001 после 20 итераций, возвращается ошибка.

## Синтаксис

```
RATE(<nper>, <pmt>, <pv>[, <fv>[, <type>[, <guess>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
nper	Общее число периодов платежей за аннуитет.
pmt	Платеж, производимый в каждый период, который не может измениться в течение аннуитета. Как правило, pmt включает в себя основную суммы и проценты, а не другие сборы или налоги.
pv	Текущее значение — общая сумма ряда будущих платежей.
fv	(Необязательно) Будущая стоимость или денежный остаток, которого вы хотите достичь после последнего платежа. Если значение fv опущено, предполагается, что оно равно 0 (будущая стоимость кредита, например, равна 0).
тип	(Необязательно) Номер 0 или 1, указывающий срок платежа. Если тип не указан, предполагается, что он равен 0. Допустимые значения перечислены ниже в этой таблице.
guess	(Необязательно) Ваше предположение о том, какой будет ставка. — Если не указано, предполагается, что это 10 %. — Если RATE не приходит к одному значению, попробуйте предположить другое значение. RATE обычно приходит к одному значению, если значение предположения находится в диапазоне от 0 до 1.

Параметр type принимает следующие значения:

ЗАДАТЬ ТИП РАВНЫМ	ЕСЛИ ПЛАТЕЖИ ДОЛЖНЫ ВНОСИТЬСЯ
0 или не указано	В конце периода
1	В начале периода

# Возвращаемое значение

Процентная ставка за период.

## Remarks

- Для указания guess и nper должны использоваться одинаковые единицы. При ежемесячных платежах по кредиту на четыре года с годовой процентной ставкой 12 % вычислите guess по формуле 0,12/12, а nper — по формуле 4\*12. Если вы вносите ежегодные платежи по одному кредиту, guess будет составлять 0,12, а nper — 4.
- type округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - nper  $\leq 0$ ;
  - RATE не приходит к значению 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

DATA	ОПИСАНИЕ
4	Лет кредита
-200	Ежемесячный платеж
8000	Сумма кредита

### Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    RATE(4*12, -200, 8000)
}
```

Возвращает месячную ставку кредита на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,00770147248820137

### Пример 2

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    RATE(4*12, -200, 8000) * 12
}
```

Возвращает годовую ставку кредита на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,0924176698584164

# RECEIVED

02.10.2020 • 3 minutes to read

Возвращает сумму, полученную при погашении, для полностью инвестированной ценной бумаги.

## Синтаксис

```
RECEIVED(<settlement>, <maturity>, <investment>, <discount>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
investment	Сумма, инвестированная в ценную бумагу.
discount	Дисконтная ставка ценной бумаги.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Сумма, полученная при погашении.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как

прошло 39 448 дней после 30 декабря 1899 г.

- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.

- RECEIVED рассчитывается следующим образом:

$$\text{RECEIVED} = \frac{\text{investment}}{1 - (\text{discount} \times \frac{\text{DIM}}{\text{B}})}$$

где:

- $\text{B}$  — число дней в году в зависимости от базы года.
- $\text{DIM}$  — число дней от issue до maturity.
- Даты settlement и maturity усекаются до целых чисел.
- basis округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - investment  $\leq 0$ ;
  - discount  $\leq 0$ ;
  - 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

Рассмотрим следующий запрос DAX:

ДАТА	ОПИСАНИЕ
15 февраля 2008 г.	Дата расчетов (выпуска)
15 мая 2008 г.	Дата погашения
\1 000 000,00 долл. США	Инвестиция
5,75 %	Процентная ставка
2	База по факту/360

```
EVALUATE
{
    RECEIVED(DATE(2008,2,15), DATE(2008,5,15), 1000000.00, 0.0575, 2)
}
```

Возвращает полную сумму, которая должна быть получена при погашении, для облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
1014584,6544071

# RRI

02.10.2020 • 2 minutes to read

Возвращает эквивалентную процентную ставку для роста инвестиций.

## Синтаксис

```
RRI(<nper>, <pv>, <fv>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
nper	Число периодов для инвестиции.
pv	Текущая стоимость инвестиции.
fv	Будущая стоимость инвестиции.

## Возвращаемое значение

Эквивалентная процентная ставка.

## Remarks

- RRI возвращает процентную ставку на основе значений  $\text{nper}$  (число периодов),  $\text{pv}$  (текущая стоимость) и  $\text{fv}$  (будущая стоимость), рассчитанную с помощью следующего уравнения:

$$\text{RRI} = \left( \frac{\text{fv}}{\text{pv}} \right)^{\frac{1}{\text{nper}}} - 1$$

- Возвращается ошибка, если:
  - $\text{nper} \leq 0$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

DATA	ОПИСАНИЕ
\10 000 долл. США	Текущая стоимость
\21 000 долл. США	Будущая стоимость
4	Лет инвестиции

Рассмотрим следующий запрос DAX:

```
EVALUATE
```

```
{  
  RRI(4*12, 10000, 21000)  
}
```

Возвращает эквивалентную процентную ставку для роста инвестиций на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,0155771057566627



# SLN

02.10.2020 • 2 minutes to read

Возвращает линейную амортизацию актива за один период.

## Синтаксис

```
SLN(<cost>, <salvage>, <life>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
cost	Начальная стоимость актива.
salvage	Стоимость в конце периода амортизации (иногда называется остаточной стоимостью актива).
данные	Количество периодов, в течение которых производится амортизация актива (иногда называется полезным сроком службы актива).

## Возвращаемое значение

Линейная амортизация за один период.

## Remarks

- Возвращается ошибка, 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

DATA	ОПИСАНИЕ
\30 000 долл. США	Cost
\7500 долл. США	Остаточная стоимость
10	Годы полезной жизни

Рассмотрим следующий запрос DAX:

EVALUATE

```
{  
  SLN(30000, 7500, 10)  
}
```

Возвращает скидку ежегодной амортизации с использованием указанных выше терминов.

**[ЗНАЧЕНИЕ]**

2250

# SYD

02.10.2020 • 2 minutes to read

Возвращает значение, указывающее амортизацию актива за указанный период, рассчитанную методом суммы цифр лет.

## Синтаксис

```
SYD(<cost>, <salvage>, <life>, <per>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
cost	Начальная стоимость актива.
salvage	Стоимость в конце периода амортизации (иногда называется остаточной стоимостью актива).
данные	Количество периодов, в течение которых производится амортизация актива (иногда называется полезным сроком службы актива).
per	Период. Должен быть указан в тех же единицах, что и срок службы. Значение должно находиться между 1 и life (включительно).

## Возвращаемое значение

Значение, указывающее амортизацию за указанный период, рассчитанную методом суммы цифр лет.

## Remarks

- SYD рассчитывается следующим образом:

$$\text{SYD} = \frac{(\text{cost} - \text{salvage}) \times (\text{life} - \text{per} + 1) \times 2}{(\text{life} + 1)}$$

- Возвращается ошибка, если:

- life < 1;
- per < 1 or per > life.

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

## Примеры

DATA	ОПИСАНИЕ
\30 000,00 долл. США	Начальная стоимость

DATA	ОПИСАНИЕ
\7500,00 долл. США	Остаточная стоимость
10	Срок службы в годах

### Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    SYD(30000.00, 7500.00, 10, 1)
}
```

Возвращает норму амортизации за первый год, рассчитанную методом суммы цифр лет, на основе указанных выше условий.

[ЗНАЧЕНИЕ]
4090,90909090909

### Пример 2

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    SYD(30000.00, 7500.00, 10, 10)
}
```

Возвращает норму амортизации за десятый (последний) год, рассчитанную методом суммы цифр лет, на основе указанных выше условий.

[ЗНАЧЕНИЕ]
409,090909090909

# TBILLEQ

02.10.2020 • 2 minutes to read

Возвращает эквивалентную доходность облигации для казначейского векселя.

## Синтаксис

```
TBILLEQ(<settlement>, <maturity>, <discount>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по казначейскому векселю. Дата расчетов должна быть после даты выпуска, когда казначейский вексель продан покупателю.
maturity	Дата погашения казначейского векселя. Дата погашения — это дата окончания срока действия казначейского векселя.
discount	Процентная ставка казначейского векселя.

## Возвращаемое значение

Эквивалентная доходность облигации для казначейского векселя.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.

- TBILLEQ вычисляется следующим образом:

$$\text{TBILLEQ} = \frac{365 \times \text{discount}}{360 - (\text{discount} \times \text{DSM})}$$

где:

- $\text{DSM}$  — число дней между значениями settlement и maturity, рассчитанное по базе 360 дней в году.
- Даты settlement и maturity усекаются до целых чисел.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity или maturity наступает более чем через один год после settlement;
  - discount  $\leq 0$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

# Пример

DATA	ОПИСАНИЕ
31.01.2008	Дата расчета
01.06.2008	Дата погашения
9,14 %	Процентная ставка

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    TBILLEQ(
        DATE(2008,3,31),
        DATE(2008,6,1),
        0.0914
    )
}
```

Возвращает эквивалентную доходность облигации для казначейского векселя на основе указанных выше значений.

[ЗНАЧЕНИЕ]
0,094151493565943

# TBILLPRICE

02.10.2020 • 2 minutes to read

Возвращает значение цены за \100 долл. США номинальной стоимости для казначейского векселя.

## Синтаксис

```
TBILLPRICE(<settlement>, <maturity>, <discount>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по казначейскому векселю. Дата расчетов должна быть после даты выпуска, когда казначейский вексель продан покупателю.
maturity	Дата погашения казначейского векселя. Дата погашения — это дата окончания срока действия казначейского векселя.
discount	Процентная ставка казначейского векселя.

## Возвращаемое значение

Цена казначейского векселя за \100 долл. США номинальной стоимости.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.

- TBILLPRICE рассчитывается следующим образом:

$$\text{\$}\text{TBILLPRICE} = 100 \times (1 - \frac{\text{\text{discount}} \times \text{\text{DSM}}}{360})\text{\$}$$

где:

- $\text{\$}\text{DSM}\text{\$}$  — это число дней от settlement до maturity, за исключением даты maturity, которая наступает более чем через один календарный год после даты settlement.
- Даты settlement и maturity усекаются до целых чисел.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity или maturity наступает более чем через один год после settlement;
  - discount  $\leq 0$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

# Пример

DATA	ОПИСАНИЕ
31.01.2008	Дата расчета
01.06.2008	Дата погашения
9,0 %	Процентная ставка

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    TBILLPRICE(
        DATE(2008,3,31),
        DATE(2008,6,1),
        0.09
    )
}
```

Возвращает цену казначейского векселя за \100 долл. США номинальной стоимости на основе указанных выше условий.

[ЗНАЧЕНИЕ]
98,45



# TBILLYIELD

02.10.2020 • 2 minutes to read

Возвращает доход для казначейского векселя.

## Синтаксис

```
TBILLYIELD(<settlement>, <maturity>, <pr>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по казначейскому векселю. Дата расчетов должна быть после даты выпуска, когда казначейский вексель продан покупателю.
maturity	Дата погашения казначейского векселя. Дата погашения — это дата окончания срока действия казначейского векселя.
pr	Цена казначейского векселя за \100 долл. США номинальной стоимости.

## Возвращаемое значение

Доход от казначейского векселя.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.

- TBILLYIELD рассчитывается следующим образом:

$$\text{TBILLYIELD} = \frac{100 - \text{pr}}{\text{pr}} \times \frac{360}{\text{DSM}}$$

где:

- $\text{DSM}$  — это число дней от settlement до maturity, за исключением даты maturity, которая наступает более чем через один календарный год после даты settlement.
- Даты settlement и maturity усекаются до целых чисел.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity или maturity наступает более чем через один год после settlement;
  - pr  $\leq$  0.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

# Пример

Рассмотрим следующий запрос DAX:

DATA	ОПИСАНИЕ
31.01.2008	Дата расчета
01.06.2008	Дата погашения
\98,45 долл. США	Цена за \100 долл. США номинальной стоимости

```
EVALUATE
{
    TBILLYIELD(
        DATE(2008,3,31),
        DATE(2008,6,1),
        98.45
    )
}
```

Возвращает доход от казначейского векселя на основе указанных выше значений.

[ЗНАЧЕНИЕ]
0,0914169629253426

# VDB

02.10.2020 • 4 minutes to read

Возвращает значение, указывающее амортизацию актива за указанный период, включая частичные периоды, методом двойного уменьшения остатка (или другим методом, определенным пользователем). VDB означает отклонение переменного баланса.

## Синтаксис

```
VDB(<cost>, <salvage>, <life>, <start_period>, <end_period>[, <factor>[, <no_switch>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
cost	Начальная стоимость актива.
salvage	Стоимость в конце периода амортизации (иногда называется остаточной стоимостью актива). Может иметь значение 0.
данные	Количество периодов, в течение которых производится амортизация актива (иногда называется полезным сроком службы актива).
start_period	Начальный период, для которого вычисляется амортизация. Начальный период должен использовать те же единицы, что и срок службы. Значение должно находиться между 1 и life (включительно).
end_period	Конечный период, для которого вычисляется амортизация. Конечный период должен использовать те же единицы, что и срок службы. Значение должно находиться между start_period и life (включительно).
Показатель	(Необязательно) Коэффициент сокращения остатка. Если значение factor не указано, предполагается, что оно равно 2 (метод двойного сокращения остатка). Если вы не хотите использовать метод двойного сокращения остатка, измените коэффициент. Описание метода двойного сокращения остатка см. в разделе DDB.

ТЕРМИН	ОПРЕДЕЛЕНИЕ
no_switch	(Необязательно) Логическое значение, указывающее, следует ли переключаться на линейную амортизацию, если амортизация превышает вычисление снижающегося остатка. Если не указано, предполагается, что это FALSE. — Если no_switch принимает значение TRUE, VDB не переключается на линейную амортизацию, даже если амортизация превышает вычисление снижающегося остатка. — Если no_switch принимает значение FALSE или опущено, VDB переключается на линейную амортизацию, если амортизация превышает вычисление снижающегося остатка.

## Возвращаемое значение

Амортизация за указанный период.

## Remarks

- Возвращается ошибка, если:
  - cost < 0;
  - salvage < 0;
  - life < 1;
  - start\_period < 1 или start\_period > end\_period;
  - end\_period < start\_period или end\_period > life;
  - factor < 0.
  - no\_switch не вычисляется как TRUE или FALSE.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Примеры

DATA	ОПИСАНИЕ
2400	Начальная стоимость
300	Остаточная стоимость
10	Время существования в годах

### Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    VDB(2400, 300, 10*365, 0, 1)
}
```

Возвращает амортизацию первого дня актива с коэффициентом 2.

[ЗНАЧЕНИЕ]

1,31506849315068

## Пример 2

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
  VDB(2400, 300, 10*12, 6, 18, 3)
}
```

Возвращает амортизацию актива между  $6\text{ \$}^{\{m\}}$  месяцем и  $18\text{ \$}^{\{m\}}$  месяцем. В этих расчетах используется коэффициент 3.

[ЗНАЧЕНИЕ]

540,185558199698

## Пример 3

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
  VDB(2400, 300, 10, 0, 0.875, 1.5)
}
```

Возвращает амортизацию актива в первом финансовом году владения активом, предполагая, что налоговые законы ограничивают вам 150 % амортизации снижающегося остатка. Актив приобретается в середине первого квартала финансового года.

[ЗНАЧЕНИЕ]

315

# XIRR

02.10.2020 • 2 minutes to read

Возвращает внутреннюю норму прибыли для запланированных денежных потоков, которые могут не являться периодическими.

## Синтаксис

```
XIRR(<table>, <values>, <dates>, [guess])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, для которой должны быть вычислены выражения значений и дат.
значения	Выражение, возвращающее значение денежного потока для каждой строки таблицы.
даты	Выражение, возвращающее дату денежного потока для каждой строки таблицы.
guess	(необязательно) Начальное предположение для внутренней нормы прибыли. Если этот параметр опущен, используется предположение 0,1 по умолчанию.

## Возвращаемое значение

Внутренняя норма прибыли для заданных входных данных. Если вычисление не возвращает допустимый результат, возвращается ошибка.

## Remarks

- Значение вычисляется как норма, удовлетворяющая следующей функции:

$$\sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

Где:

- $P_j$  —  $j^{\text{й}}$  платеж
  - $d_j$  — дата  $j^{\text{го}}$  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

В следующей формуле вычисляется внутренняя норма прибыли для таблицы CashFlows:

= XIRR( CashFlows, [Payment], [Date] )

ДАТА	ПЛАТЕЖ
01.01.2014	–10000
01.03.2014	2 750
30.10.2014	4250
15.02.2015	3 250
01.04.2015	2 750

Норма прибыли = 37,49 %

# XNPV

02.10.2020 • 2 minutes to read

Возвращает текущее значение для запланированных денежных потоков, которые могут не являться периодическими.

## Синтаксис

```
XNPV(<table>, <values>, <dates>, <rate>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, для которой должны быть вычислены выражения значений и дат.
значения	Выражение, возвращающее значение денежного потока для каждой строки таблицы.
даты	Выражение, возвращающее дату денежного потока для каждой строки таблицы.
rate	Коэффициент скидки, применяемый к денежному потоку для каждой строки таблицы.

## Возвращаемое значение

Чистое текущее значение.

## Remarks

- Значение вычисляется как следующая сумма:

$$\sum_{j=1}^N \frac{P_j}{(1 + \text{rate})^{\frac{d_j - d_1}{365}}}$$

Где:

- $P_j$  —  $j^{\text{й}}$  платеж
  - $d_j$  — дата  $j^{\text{го}}$  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

В следующем примере вычисляется текущее значение таблицы CashFlows:



= XNPV( CashFlows, [Payment], [Date], 0.09 )

ДАТА	ПЛАТЕЖ
01.01.2014	–10000
01.03.2014	2 750
30.10.2014	4250
15.02.2015	3 250
01.04.2015	2 750

Текущее значение = 2086.65

# YIELD

02.10.2020 • 5 minutes to read

Возвращает доход с ценной бумаги, по которой периодически выплачиваются проценты. Используйте YIELD для вычисления доходности облигаций.

## Синтаксис

```
YIELD(<settlement>, <maturity>, <rate>, <pr>, <redemption>, <frequency>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
rate	Годовая купонная ставка ценной бумаги.
pr	Цена ценной бумаги на \100 долл. США номинальной стоимости.
redemption	Стоимость выкупа ценной бумаги на \100 долл. США номинальной стоимости.
frequency	Количество купонных выплат в год. Для ежегодных выплат периодичность составляет 1; для выплат раз в полгода — 2; для ежеквартальных — 4.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

# Возвращаемое значение

Доходность ценной бумаги.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.

- Если до выкупа остается один или несколько купонных периодов, то YIELD вычисляется следующим образом:

$$\text{YIELD} = \frac{\left( \frac{\text{redemption}}{100} + \frac{\text{rate}}{\text{frequency}} \right) - \left( \frac{\text{par}}{100} + \left( \frac{\text{A}}{\text{E}} \times \frac{\text{rate}}{\text{frequency}} \right) \right) \frac{\text{par}}{100} + \left( \frac{\text{A}}{\text{E}} \times \frac{\text{rate}}{\text{frequency}} \right) \times \frac{\text{frequency}}{\text{E}} \times \text{DSR}}$$

где:

- $\text{A}$  — число дней с начала купонного периода до даты расчетов (дни начисления).
- $\text{DSR}$  — число дней от даты расчетов до даты выкупа.
- $\text{E}$  — число дней в купонном периоде.
- Если до выкупа остается более одного купонного периода, то YIELD вычисляется с помощью сотен итераций. Разрешение использует метод Ньютона, основанный на формуле, используемой для функции PRICE. Доходность изменяется до тех пор, пока предполагаемая цена с учетом доходности не будет близка к цене.
- Даты settlement и maturity усекаются до целых чисел.
- Значения frequency и basis округляются до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - rate  $< 0$ ;
  - pr  $\leq 0$ ;
  - redemption  $\leq 0$ ;
  - frequency является любым числом, кроме 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

DATA	ОПИСАНИЕ
15.02.2008	Дата расчета
15.11.2016	Дата погашения
5,75 %	Процентный купон
95,04287	Цена
\100 долл. США	Выкупная стоимость
2	Периодичность — раз в полгода (см. выше)
0	База 30/360 (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    YIELD(DATE(2008,2,15), DATE(2016,11,15), 0.0575, 95.04287, 100, 2,0)
}
```

Возвращает доходность облигации на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,0650000068807314

# YIELDDISC

02.10.2020 • 3 minutes to read

Возвращает годовой доход для дисконтной ценной бумаги.

## Синтаксис

```
YIELDDISC(<settlement>, <maturity>, <pr>, <redemption>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
pr	Цена ценной бумаги на \100 долл. США номинальной стоимости.
redemption	Стоимость выкупа ценной бумаги на \100 долл. США номинальной стоимости.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Годовой доход.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Даты settlement и maturity усекаются до целых чисел.
- basis округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - settlement или maturity не является допустимой датой;
  - settlement  $\geq$  maturity;
  - pr  $\leq$  0;
  - redemption  $\leq$  0;
  - 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

DATA	
16.02.2008	Дата расчета
1.03.2008	Дата погашения
99,795	Цена
\100 долл. США	Выкупная стоимость
2	База по факту/360

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    YIELDDISC(
        DATE(2008,2,16),
        DATE(2008,3,1),
        99.795,
        100,
        2
    )
}
```

Возвращает годовой доход по ценной бумаге на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,0528225719868583

# YIELDMAT

02.10.2020 • 3 minutes to read

Возвращает годовую доходность для ценной бумаги, по которой выплачиваются проценты при погашении.

## Синтаксис

```
YIELDMAT(<settlement>, <maturity>, <issue>, <rate>, <pr>[, <basis>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
settlement	Дата расчетов по ценной бумаге. Дата расчетов по ценной бумаге должна быть после даты выпуска, когда ценная бумага продана покупателю.
maturity	Дата погашения ценной бумаги. Дата погашения — это дата окончания срока действия ценной бумаги.
проблема	Дата выпуска ценной бумаги.
rate	Процентная ставка ценной бумаги на дату выпуска.
pr	Цена ценной бумаги на \100 долл. США номинальной стоимости.
basis	(необязательно) Используемый тип для подсчета дней. Если база не указана, предполагается, что она равна 0. Допустимые значения перечислены ниже в этой таблице.

Параметр **basis** принимает следующие значения:

БАЗА	БАЗА РАСЧЕТА ПРОЦЕНТОВ
0 или не указано	Американский стандарт (NASD), 30/60
1	Фактический/фактический
2	Фактический/360
3	Фактический/365
4	Европейский стандарт, 30/360

## Возвращаемое значение

Годовой доход.

## Remarks

- Даты хранятся как последовательные серийные номера, чтобы их можно было использовать в вычислениях. В DAX 30 декабря 1899 г. — это день 0, а 1 января 2008 г. — день 39 448, так как прошло 39 448 дней после 30 декабря 1899 г.
- Дата расчетов — это дата приобретения покупателем купона, например облигации. Дата погашения — это дата окончания срока действия купона. Например, 30-летняя облигация выпущена 1 января 2008 г. и приобретена покупателем через шесть месяцев. Датой выпуска будет считаться 1 января 2008 года, датой расчетов — 1 июля 2008 г., а датой погашения — 1 января 2038 г., то есть через 30 лет после даты выпуска 1 января 2008 года.
- Значения settlement, maturity и issue усекаются до целых чисел.
- basis округляется до ближайшего целого числа.
- Возвращается ошибка, если:
  - значение settlement, maturity или issue не является допустимой датой;
  - условие maturity > settlement > issue не удовлетворяется;
  - rate < 0;
  - pr ≤ 0;
  - 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

DATA	ОПИСАНИЕ
15.03.2008	Дата расчета
3.11.2008	Дата погашения
8.11.2007	Дата выпуска
6,25 %	Процент полугодового купона
100,0123	Цена
0	База 30/360 (см. выше)

Рассмотрим следующий запрос DAX:

```
EVALUATE
{
    YIELDMAT(DATE(2008,3,15), DATE(2008,11,3), DATE(2007,11,8), 0.0625, 100.0123, 0)
}
```

Возвращает доходность ценной бумаги на основе указанных выше условий.

[ЗНАЧЕНИЕ]
0,0609543336915387





# Информационные функции

02.10.2020 • 4 minutes to read

Информационные функции DAX просматривают ячейку или строку, предоставленную в качестве аргумента, и сообщают, соответствует ли значение ожидаемому типу. Например, функция ISERROR возвращает значение TRUE, если упоминаемое значение содержит ошибку.

## В этой категории

КОМПОНЕНТ	DESCRIPTION
CONTAINS	Возвращает значение true, если значения во всех этих столбцах существуют или содержатся в них. В противном случае функция возвращает значение false.
CONTAINSROW	Возвращает TRUE, если строка значений присутствует или содержится в таблице. В остальных случаях возвращает FALSE.
CONTAINSSTRING	Возвращает значение TRUE или FALSE, указывающее, содержит ли одна строка другую строку.
CONTAINSSTRINGEXACT	Возвращает значение TRUE или FALSE, указывающее, содержит ли одна строка другую строку.
CUSTOMDATA	Возвращает содержимое свойства CustomData в строке подключения.
HASONEFILTER	Возвращает значение TRUE, если число непосредственно фильтруемых значений в <i>columnName</i> равно 1, в противном случае возвращает значение FALSE.
HASONEVALUE	Возвращает значение TRUE, если контекст для <i>columnName</i> был отфильтрован до одного конкретного значения. В противном случае — значение FALSE.
ISBLANK	Проверяет, является ли значение пустым, и возвращает TRUE или FALSE.
ISCROSSFILTERED	Возвращает значение TRUE, если фильтруется <i>columnName</i> или другой столбец в той же или связанной таблице.
ISEMPTY	Проверяет, пуста ли таблица.
ISERROR	Проверяет, является ли значение ошибкой, и возвращает значение TRUE или FALSE.
ISEVEN	Возвращает значение TRUE, если число четное, и значение FALSE, если нечетное.

КОМПОНЕНТ	DESCRIPTION
ISFILTERED	Возвращает значение TRUE, если <i>columnName</i> фильтруется напрямую.
ISINSCOPE	Возвращает значение true, когда указанный столбец является уровнем в иерархии уровней.
ISLOGICAL	Проверяет, является ли значение логическим (равным TRUE или FALSE), и возвращает значение TRUE или FALSE.
ISNONTEXT	Проверяет, является ли значение нетекстовым (пустые ячейки не считаются текстом), и возвращает значение TRUE или FALSE.
ISNUMBER	Проверяет, является ли значение числовым, и возвращает значение TRUE или FALSE.
ISODD	Возвращает значение TRUE, если число нечетное, и значение FALSE, если четное.
ISONORAFTER	Логическая функция, имитирующая поведение предложения Start At и возвращающая значение true для строки, удовлетворяющей всем параметрам условия.
ISSELECTEDMEASURE	Используется выражениями для элементов вычисления для определения меры, которая находится в контексте одной из указанных мер.
ISSUBTOTAL	Создает другой столбец в выражении SUMMARIZE, который содержит значение true, если строка содержит значения промежуточных итогов для столбца, указанного в качестве аргумента; в противном случае столбец содержит значение false.
ISTEXT	Проверяет, является ли значение текстовым, и возвращает TRUE или FALSE.
NONVISUAL	Помечает фильтр значений в выражении SUMMARIZECOLUMNS как не являющийся визуальным.
SELECTEDMEASURE	Используется выражениями для элементов вычисления для ссылки на меру, которая находится в контексте.
SELECTEDMEASUREFORMATSTRING	Используется выражениями для элементов вычисления, чтобы извлечь строку формата меры, которая находится в контексте.
SELECTEDMEASURENAME	Используется выражениями для элементов вычисления для определения меры, которая находится в контексте по имени.
USERNAME	Возвращает имя домена и имя пользователя из учетных данных, предоставленных системе во время подключения.

КОМПОНЕНТ	DESCRIPTION
USEROBJECTID	Возвращает идентификатор объекта или ИД безопасности текущего пользователя.
USERPRINCIPALNAME	Возвращает имя субъекта-пользователя.

# CONTAINS

02.10.2020 • 2 minutes to read

Возвращает значение true, если значения во всех этих столбцах существуют или содержатся в них. В противном случае функция возвращает значение false.

## Синтаксис

```
CONTAINS(<table>, <columnName>, <value>[, <columnName>, <value>]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Любое выражение DAX, возвращающее таблицу данных.
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.
value	Любое выражение DAX, возвращающее одиночное скалярное значение, искомое в <i>columnName</i> . Выражение должно быть вычислено только один раз до его передачи в список аргументов.

## Возвращаемое значение

Значение TRUE, если каждое заданное значение *value* может быть найдено в соответствующем столбце *columnName* или содержится в этих столбцах. В противном случае функция возвращает FALSE.

## Remarks

- Аргументы *columnName* и *value* должны задаваться парами. В противном случае возвращается ошибка.
- Столбец *columnName* должен принадлежать к заданной таблице *table* или к таблице, связанной с *table*.
- Если *columnName* ссылается на столбец в связанной таблице, имя должно быть полным; 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, которая сообщает, были ли интернет-продажи продукта 214 клиенту 11185.

```
= CONTAINS(InternetSales, [ProductKey], 214, [CustomerKey], 11185)
```



# Функция CONTAINSROW

02.10.2020 • 2 minutes to read

Возвращает TRUE, если строка значений присутствует или содержится в таблице. В остальных случаях возвращает FALSE.

## Синтаксис

```
CONTAINSROW(<tableExpr>, <scalarExpr>[, <scalarExpr>, ...])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
scalarExprN	Любое допустимое выражение DAX, которое возвращает скалярное значение.
tableExpr	Любое допустимое выражение DAX, которое возвращает таблицу данных.

## Возвращаемое значение

TRUE или FALSE.

## Remarks

- Оператор IN и функция CONTAINSROW функционально эквивалентны, за исключением синтаксиса.

```
<scalarExpr> IN <tableExpr>  
( <scalarExpr1>, <scalarExpr2>, ... ) IN <tableExpr>
```

- Количество выражений scalarExprN должно соответствовать количеству столбцов в tableExpr.
- Оператор NOT IN в DAX отсутствует. Чтобы выполнить логическое отрицание оператора IN, необходимо поставить оператор NOT перед всем выражением. Например, NOT [Color] IN { "Red", "Yellow", "Blue" }.
- В отличие от оператора =, оператор IN и функция CONTAINSROW выполняют строгое сравнение. Например, значение BLANK не соответствует 0.

## Пример 1

Приведенные ниже эквивалентные запросы DAX:

```
EVALUATE FILTER(ALL(DimProduct[Color]), [Color] IN { "Red", "Yellow", "Blue" })  
ORDER BY [Color]
```

```
EVALUATE FILTER(ALL(DimProduct[Color]), ([Color]) IN { "Red", "Yellow", "Blue" })
ORDER BY [Color]
```

И

```
EVALUATE FILTER(ALL(DimProduct[Color]), CONTAINSROW({ "Red", "Yellow", "Blue" }, [Color]))
ORDER BY [Color]
```

Возвращают следующую таблицу, содержащую один столбец:

DIMPRODUCT[COLOR]	
Синий	
Красный	
Желтый	

## Пример 2

Приведенные ниже эквивалентные запросы DAX:

```
EVALUATE FILTER(SUMMARIZE(DimProduct, [Color], [Size]), ([Color], [Size]) IN { ("Black", "L") })
```

И

```
EVALUATE FILTER(SUMMARIZE(DimProduct, [Color], [Size]), CONTAINSROW({ ("Black", "L") }, [Color], [Size]))
```

Выходные данные:

DIMPRODUCT[COLOR]	DIMPRODUCT[SIZE]
Черный	L

## Пример 3

Приведенные ниже эквивалентные запросы DAX:

```
EVALUATE FILTER(ALL(DimProduct[Color]), NOT [Color] IN { "Red", "Yellow", "Blue" })
ORDER BY [Color]
```

И

```
EVALUATE FILTER(ALL(DimProduct[Color]), NOT CONTAINSROW({ "Red", "Yellow", "Blue" }, [Color]))
ORDER BY [Color]
```

Возвращают следующую таблицу, содержащую один столбец:

DIMPRODUCT[COLOR]	
Черный	



DIMPRODUCT[COLOR]	
Серый	
Несколько вариантов	
Н/Д	
Серебряная	
Silver\Black	
Белый	

# CONTAINSSTRING

22.09.2020 • 2 minutes to read

Возвращает значение TRUE или FALSE, указывающее, содержит ли одна строка другую строку.

## Синтаксис

```
CONTAINSSTRING(<within_text>, <find_text>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
within_text	Текст, в котором выполняется поиск строки find_text.
find_text	Искомый текст.

## Возвращаемое значение

Значение TRUE, если find\_text является подстрокой within\_text; в противном случае — FALSE.

## Remarks

- CONTAINSSTRING не учитывает регистр.
- Можно использовать подстановочные знаки `?` и `*`. Используйте `~` для экранирования подстановочных знаков.

## Пример

Запрос DAX

```
EVALUATE
    ROW(
        "Case 1", CONTAINSSTRING("abcd", "bc"),
        "Case 2", CONTAINSSTRING("abcd", "BC"),
        "Case 3", CONTAINSSTRING("abcd", "a*d"),
        "Case 4", CONTAINSSTRING("abcd", "ef")
    )
```

Результаты

[ВАРИАНТ 1]	[ВАРИАНТ 2]	[ВАРИАНТ 3]	[ВАРИАНТ 4]
TRUE	TRUE	TRUE	FALSE

# CONTAINSSTRINGEXACT

21.09.2020 • 2 minutes to read

Возвращает значение TRUE или FALSE, указывающее, содержит ли одна строка другую строку.

## Синтаксис

```
CONTAINSSTRINGEXACT(<within_text>, <find_text>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
within_text	Текст, в котором выполняется поиск строки find_text.
find_text	Искомый текст.

## Возвращаемое значение

Значение TRUE, если find\_text является подстрокой within\_text; в противном случае — FALSE.

## Remarks

CONTAINSSTRINGEXACT учитывает регистр.

## Пример

Запрос DAX

```
EVALUATE
    ROW(
        "Case 1", CONTAINSSTRINGEXACT("abcd", "bc"),
        "Case 2", CONTAINSSTRINGEXACT("abcd", "BC"),
        "Case 3", CONTAINSSTRINGEXACT("abcd", "a*d"),
        "Case 4", CONTAINSSTRINGEXACT("abcd", "ef")
    )
```

### Результаты

[ВАРИАНТ 1]	[ВАРИАНТ 2]	[ВАРИАНТ 3]	[ВАРИАНТ 4]
TRUE	FALSE	FALSE	FALSE

# CUSTOMDATA

02.10.2020 • 2 minutes to read

Возвращает содержимое свойства CustomData в строке подключения.

## Синтаксис

```
CUSTOMDATA()
```

## Возвращаемое значение

Содержимое свойства CustomData в строке подключения.

Пусто, если свойство CustomData не было определено во время подключения.

## Remarks

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, имеет ли свойство CustomData значение **OK**.

```
= IF(CUSTOMDATA()="OK", "Correct Custom data in connection string", "No custom data in connection string property or unexpected value")
```

# HASONEFILTER

22.09.2020 • 2 minutes to read

Возвращает значение **TRUE**, если число непосредственно фильтруемых значений в *columnName* равно 1, в противном случае возвращает значение **FALSE**.

## Синтаксис

```
HASONEFILTER(<columnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.

## Возвращаемое значение

Значение **TRUE**, если число непосредственно фильтруемых значений в *columnName* равно 1, в противном случае возвращает значение **FALSE**.

## Remarks

- Эта функция похожа на **HASONEVALUE()** за тем исключением, что **HASONEVALUE()** работает на основе перекрестных фильтров, тогда как **HASONEFILTER()** — на основе прямого фильтра.
- 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()**, чтобы вернуть фильтр для **ResellerSales\_USD[ProductKey]**, если имеется один фильтр, или вернуть пустое значение **BLANK()**, если для **ResellerSales\_USD[ProductKey]** нет фильтров или указано более одного фильтра.

```
= IF(HASONEFILTER(ResellerSales_USD[ProductKey]),FILTERS(ResellerSales_USD[ProductKey]),BLANK())
```

# HASONEVALUE

02.10.2020 • 2 minutes to read

Возвращает значение **TRUE**, если контекст для *columnName* был отфильтрован до одного конкретного значения. В противном случае — значение **FALSE**.

## Синтаксис

```
HASONEVALUE(<columnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.

## Возвращаемое значение

Значение **TRUE**, если контекст для *columnName* был отфильтрован до одного конкретного значения. В противном случае — значение **FALSE**.

## Remarks

- 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, которая проверяет, выполняется ли срез контекста по одному значению, чтобы оценить процентное отношение по предопределенному сценарию. В данном случае требуется сравнить продажи торгового посредника с продажами за 2007 г., а затем определить, фильтруется ли контекст по отдельным годам. Кроме того, если сравнение не имеет смысла, требуется вернуть значение BLANK (Пусто).

```
=  
IF(HASONEVALUE(DateTime[CalendarYear]),SUM(ResellerSales_USD[SalesAmount_USD])/CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),DateTime[CalendarYear]=2007),BLANK())
```

# ISBLANK

02.10.2020 • 2 minutes to read

Проверяет, является ли значение пустым, и возвращает TRUE или FALSE.

## Синтаксис

```
ISBLANK(<value>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
value	Значение или выражение, которое необходимо проверить.

## Возвращаемое значение

Логическое значение TRUE, если значение является пустым; в противном случае — FALSE.

## Пример

Эта формула рассчитывает коэффициент увеличения или уменьшения продаж по сравнению с предыдущим годом. В примере используется функция IF для проверки значения продаж за предыдущий год во избежание ошибки деления на ноль.

```
//Sales to Previous Year Ratio  
  
= IF( ISBLANK('CalculatedMeasures'[PreviousYearTotalSales])  
    , BLANK()  
    , ( 'CalculatedMeasures'[Total Sales]-'CalculatedMeasures'[PreviousYearTotalSales] )  
      /'CalculatedMeasures'[PreviousYearTotalSales])
```

Результат:

МЕТКИ СТРОК	ОБЪЕМ ПРОДАЖ	ВСЕГО ПРОДАЖ ЗА ПРЕДЫДУЩИЙ ГОД	КОЭФФИЦИЕНТ ПРОДАЖ ПО СРАВНЕНИЮ С ПРЕДЫДУЩИМ ГОДОМ
2005	10 209 985,08 долл. США		
2006	28 553 348,43 долл. США	10 209 985,08 долл. США	179,66 %
2007 г.	39 248 847,52 долл. США	28 553 348,43 долл. США	37,46 %
2008	24 542 444,68 долл. США	39 248 847,52 долл. США	−37,47 %
Grand Total	102 554 625,71 долл. США		

См. также раздел

[Информационные функции](#)



# ISCROSSFILTERED

02.10.2020 • 2 minutes to read

Возвращает значение TRUE, если фильтруется *columnName* или другой столбец в той же или связанной таблице.

## Синтаксис

```
ISCROSSFILTERED(<columnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.

## Возвращаемое значение

Возвращает значение TRUE, если фильтруется *columnName* или другой столбец в той же или связанной таблице. В противном случае возвращает значение FALSE.

## Remarks

- Столбец считается фильтруемым перекрестно, когда фильтр, примененный к другому столбцу в той же таблице или связанной таблице, также влияет на *columnName* путем фильтрации. Столбец считается фильтруемым *напрямую*, когда фильтр или фильтры применяются к столбцу.
- Связанная функция [ISFILTERED](#) возвращает значение TRUE, если *columnName* фильтруется напрямую.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## См. также раздел

[Функция ISFILTERED](#)

[Функция FILTERS](#)

[Функция HASONEFILTER](#)

[HASONEVALUE](#), функция

# ISEMPTY

02.10.2020 • 2 minutes to read

Проверяет, пуста ли таблица.

## Синтаксис

```
ISEMPTY(<table_expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table_expression	Ссылка на таблицу или выражение DAX, возвращающее таблицу.

## Возвращаемое значение

Значение True, если таблица пуста (не содержит строк), в противном случае — значение False.

## Remarks

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":

СТРАНА ИЛИ РЕГИОН	СОСТОЯНИЕ	ОКРУГ	ИТОГ
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
США	Целостности и доступности	5	500
США	WA	10	900

```
EVALUATE  
ROW("Any countries with count > 25?", NOT(ISEMPTY(FILTER(Info, [Count]>25))))
```

Возвращаемое значение: FALSE

# ISERROR

22.09.2020 • 2 minutes to read

Проверяет, является ли значение ошибкой, и возвращает значение TRUE или FALSE.

## Синтаксис

```
ISERROR(<value>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
value	Проверяемое значение.

## Возвращаемое значение

Логическое значение TRUE, если значение является ошибкой; в противном случае — FALSE.

## Remarks

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 отношение общего объема продаж через Интернет к общему товарообороту посредников. Функция ISERROR используется для проверки ошибок, например деления на ноль. Если возникает ошибка, возвращается пустое значение, в противном случае возвращается отношение.

```
= IF( ISERROR(
    SUM('ResellerSales_USD'[SalesAmount_USD])
    /SUM('InternetSales_USD'[SalesAmount_USD])
), BLANK()
, SUM('ResellerSales_USD'[SalesAmount_USD])
  /SUM('InternetSales_USD'[SalesAmount_USD])
)
```

## См. также

[Информационные функции](#)

[IFERROR, функция](#)

[IF, функция](#)

# ISEVEN

22.09.2020 • 2 minutes to read

Возвращает значение TRUE, если число четное, и значение FALSE, если нечетное.

## Синтаксис

```
ISEVEN(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Значение для проверки. Если значение параметра number не является целым числом, оно усекается.

## Возвращаемое значение

Возвращает значение TRUE, если число четное, и значение FALSE, если нечетное.

## Remarks

- Если значение параметра number является нечисловым, ISEVEN возвращает #VALUE! (значение ошибки).
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

# ISFILTERED

02.10.2020 • 2 minutes to read

Возвращает значение TRUE, если *columnName* фильтруется напрямую. Если в столбце нет фильтра или если фильтрация происходит потому, что фильтруется другой столбец в той же таблице или связанной таблице, функция возвращает FALSE.

## Синтаксис

```
ISFILTERED(<columnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.

## Возвращаемое значение

Возвращает значение TRUE, если *columnName* фильтруется напрямую.

## Remarks

- *columnName* считается фильтруемым напрямую, когда фильтр или фильтры применяются к столбцу; столбец считается фильтруемым перекрестно, когда фильтр, примененный к другому столбцу в той же таблице или связанной таблице, также влияет на *columnName* путем фильтрации.
- Связанная функция [ISCROSSFILTERED](#) возвращает значение TRUE, если фильтруется *columnName* или другой столбец в той же или связанной таблице.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## См. также раздел

[Функция ISCROSSFILTERED](#)

[Функция FILTERS](#)

[Функция HASONEFILTER](#)

[HASONEVALUE](#), функция

# ISINSCOPE

22.09.2020 • 2 minutes to read

Возвращает значение true, когда указанный столбец является уровнем в иерархии уровней.

## Синтаксис

```
ISINSCOPE(<columnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.

## Возвращаемое значение

TRUE, когда указанный столбец является уровнем в иерархии уровней.

## Remarks

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

## Пример

```

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:

DIMPRODUCT[КАТЕГОРИЯ]	DIMPRODUCT[ПОДКАТЕГОРИЯ]	[ПОДЫТОГ КАТЕГОРИИ]	[ПОДЫТОГ ПОДКАТЕГОРИИ]	[ПРОДАЖИ]	[% РОДИТЕЛЬСКОЙ КОМПАНИИ]
		TRUE	TRUE	28 397 095,65	100,00
Accessories		FALSE	true	78 951,00	0,28
Accessories	Велосипедные стойки	FALSE	FALSE	39 360,00	49,85
Accessories	Парковка для велосипедов	FALSE	FALSE	39 591,00	50,15
Bikes		FALSE	true	28 318 144,65	99,72
Bikes	Горные велосипеды	FALSE	FALSE	9 952 759,56	35,15
Bikes	Дорожные велосипеды	FALSE	FALSE	14 520 584,04	51,28

DIMPRODUCT[КАТЕГОРИЯ]	DIMPRODUCT[ПОДКАТЕГОРИЯ]	[ПОДЫТОГ КАТЕГОРИИ]	[ПОДЫТОГ ПОДКАТЕГОРИИ]	[ПРОДАЖИ]	[% РОДИТЕЛЬСКОЙ КОМПАНИИ]
Bikes	Туристические велосипеды	FALSE	FALSE	3 844 801,05	13,58

См. также

[Функция SUMMARIZECOLUMNS](#)  
[CALCULATE](#), функция



# ISLOGICAL

22.09.2020 • 2 minutes to read

Проверяет, является ли значение логическим (равным TRUE или FALSE), и возвращает значение TRUE или FALSE.

## Синтаксис

```
ISLOGICAL(<value>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
value	Проверяемое значение.

## Возвращаемое значение

TRUE, если значение является логическим значением; FALSE, если это значение, отличное от TRUE или FALSE.

## Remarks

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 функции ISLOGICAL.

```
//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")
```

## См. также

[Информационные функции](#)

# ISNONTEXT

22.09.2020 • 2 minutes to read

Проверяет, является ли значение нетекстовым (пустые ячейки не считаются текстом), и возвращает значение TRUE или FALSE.

## Синтаксис

```
ISNONTEXT(<value>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
value	Проверяемое значение.

## Возвращаемое значение

Значение TRUE, если значение не является текстом или пустым; FALSE, если значение является текстом.

## Remarks

- 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.

```
//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")
```

## См. также раздел

[Информационные функции](#)

# ISNUMBER

22.09.2020 • 2 minutes to read

Проверяет, является ли значение числовым, и возвращает значение TRUE или FALSE.

## Синтаксис

```
ISNUMBER(<value>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
value	Проверяемое значение.

## Возвращаемое значение

TRUE, если значение является числовым; в противном случае — FALSE.

## Remarks

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 функции ISNUMBER.

```
//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")
```

## См. также

[Информационные функции](#)

# ISODD

22.09.2020 • 2 minutes to read

Возвращает значение TRUE, если число нечетное, и значение FALSE, если четное.

## Синтаксис

```
ISODD(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Значение для проверки. Если значение параметра number не является целым числом, оно усекается.

## Возвращаемое значение

Возвращает значение TRUE, если число нечетное, и значение FALSE, если четное.

## Remarks

- Если значение параметра number является нечисловым, ISODD возвращает #VALUE! (значение ошибки).
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

# ISONORAFTER

02.10.2020 • 2 minutes to read

Логическая функция, имитирующая поведение предложения "Start At" и возвращающая значение true для строки, удовлетворяющей всем параметрам условия.

Эта функция принимает переменное число трижды: первые два значения числе являются сравниваемыми выражениями, а третий параметр указывает порядок сортировки. Порядок сортировки может быть по возрастанию (по умолчанию) или по убыванию.

В зависимости от порядка сортировки первый параметр сравнивается со вторым параметром. Если порядок сортировки выполняется по возрастанию, то сравнивается, больше ли первый параметр второго параметра или равен ему. Если порядок сортировки выполняется по убыванию, то сравнивается, меньше ли второй параметр первого параметра или равен ему.

## Синтаксис

```
ISONORAFTER(<scalar_expression>, <scalar_expression>[, sort_order [, <scalar_expression>, <scalar_expression>[, sort_order]]...])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Скалярное выражение	Любое выражение, возвращающее скалярное значение, такое как ссылка на столбец, целое число или строковое значение. Как правило, первый параметр является ссылкой на столбец, а второй параметр — скалярным значением.
Порядок сортировки	(необязательно) Порядок сортировки столбца. Может быть по возрастанию (ASC) или по убыванию (DEC). По умолчанию используется порядок сортировки по возрастанию.

## Возвращаемое значение

True или False.

## Remarks

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

## Пример

Имя таблицы: "Информация"

СТРАНА ИЛИ РЕГИОН	СОСТОЯНИЕ	COUNT	ИТОГ
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
США	Целостности и доступности	5	500
США	WA	10	900

`FILTER(Info, ISONORAFTER(Info[Country], "IND", ASC, Info[State], "MH", ASC))`

# ISSELECTEDMEASURE

22.09.2020 • 2 minutes to read

Используется выражениями для элементов вычисления для определения меры, которая находится в контексте одной из указанных мер.

## Синтаксис

```
ISSELECTEDMEASURE( M1, M2, ... )
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
M1, M2, ...	Список мер.

## Возвращаемое значение

Логическое значение, указывающее, является ли мера, которая в данный момент находится в контексте, одной из указанных в списке параметров.

## Remarks

- Для элемента вычисления можно ссылаться только на выражение.
- 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 указанных в списке параметров. Если меры переименованы, то исправление формулы будет отражать изменения имени в выражении.

```
IF (
    ISSELECTEDMEASURE ( [Expense Ratio 1], [Expense Ratio 2] ),
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

## См. также

[SELECTEDMEASURE](#)

[SELECTEDMEASURENAME](#)

# ISSUBTOTAL

02.10.2020 • 2 minutes to read

Создает другой столбец в выражении [SUMMARIZE](#), который содержит значение true, если строка содержит значения промежуточных итогов для столбца, указанного в качестве аргумента; в противном случае столбец содержит значение false.

## Синтаксис

```
ISSUBTOTAL(<columnName>)
```

С функцией [SUMMARIZE](#)

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, ROLLUP(<groupBy_columnName>[, <groupBy_columnName>...])][, <name>, {<expression>|ISSUBTOTAL(<columnName>)}]...)
```

## Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя любого столбца в таблице функции SUMMARIZE или любого столбца связанной с таблицей таблицы.

## Возвращаемое значение

Значение True, если строка содержит значение промежуточного итога для столбца, указанного в качестве аргумента, в противном случае возвращает False.

## Remarks

- Эту функцию можно использовать только в выражении функции [SUMMARIZE](#).
- Этой функции должно предшествовать имя логического столбца.

## Пример

См. [SUMMARIZE](#).



# ISTEXT

02.10.2020 • 2 minutes to read

Проверяет, является ли значение текстовым, и возвращает TRUE или FALSE.

## Синтаксис

```
ISTEXT(<value>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
value	Проверяемое значение.

## Возвращаемое значение

TRUE, если значение является текстовым; в противном случае — FALSE.

## Remarks

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.

```
//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")
```

## См. также

[Информационные функции](#)

# NONVISUAL

02.10.2020 • 2 minutes to read

Помечает фильтр значений в выражении [SUMMARIZECOLUMNS](#) как не являющийся визуальным. Эту функцию можно использовать только в выражении [SUMMARIZECOLUMNS](#).

## Синтаксис

```
NONVISUAL(<expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Любое выражение DAX, возвращающее одиночное значение (не таблицу).

## Возвращаемое значение

Таблица значений.

## Remarks

- Помечает фильтр значений в функции [SUMMARIZECOLUMNS](#) как не влияющий на значения мер, но применяется только к столбцам группировки.
- Эту функцию можно использовать только в выражении [SUMMARIZECOLUMNS](#). Он используется как аргумент filterTable функции [SUMMARIZECOLUMNS](#) или аргумент groupLevelFilter функции [ROLLUPADDISSUBTOTAL](#) или [ROLLUPISSUBTOTAL](#).

## Пример

См. [SUMMARIZECOLUMNS](#).

# SELECTEDMEASURE

22.09.2020 • 2 minutes to read

Используется выражениями для элементов вычисления для ссылки на меру, которая находится в контексте.

## Синтаксис

```
SELECTEDMEASURE()
```

### Параметры

Отсутствуют

## Возвращаемое значение

Ссылка на меру, которая в данный момент находится в контексте, когда вычисляется элемент вычисления.

## Remarks

- Для элемента вычисления можно ссылаться только на выражение.
- 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 с начала года для всех элементов, находящихся в контексте меры.

```
CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))
```

## См. также

[SELECTEDMEASURENAME](#)

[ISSELECTEDMEASURE](#)

# SELECTEDMEASUREFORMATSTRING

22.09.2020 • 2 minutes to read

Используется выражениями для элементов вычисления, чтобы извлечь строку формата меры, которая находится в контексте.

## Синтаксис

```
SELECTEDMEASUREFORMATSTRING()
```

### Параметры

Отсутствуют

## Возвращаемое значение

Строка, содержащая строку формата меры, которая в данный момент находится в контексте, когда вычисляется элемент вычисления.

## Remarks

- На эту функцию можно ссылаться только в выражениях для элементов вычисления в группах вычислений. Она предназначена для использования **выражения строки формата** свойства элементов 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 вычисляется свойством выражения строки формата для элемента вычисления. Если в контексте фильтра имеется одна валюта, строка формата извлекается из столбца DimCurrency [FormatString]. В противном случае используется строка формата меры в контексте.

```
SELECTEDVALUE( DimCurrency[FormatString], SELECTEDMEASUREFORMATSTRING() )
```

## См. также

[SELECTEDMEASURE](#)  
[ISSELECTEDMEASURE](#)

# SELECTEDMEASURENAME

22.09.2020 • 2 minutes to read

Используется выражениями для элементов вычисления для определения меры, которая находится в контексте по имени.

## Синтаксис

```
SELECTEDMEASURENAME()
```

### Параметры

Отсутствуют

## Возвращаемое значение

Строковое значение, содержащее имя меры, которая в данный момент находится в контексте, когда вычисляется элемент вычисления.

## Remarks

- Для элемента вычисления можно ссылаться только на выражение.
- Эта функция часто используется в целях отладки при создании групп вычислений.

## Пример

Следующее выражение элемента вычисления проверяет, является ли текущая мера коэффициентом расходов, и условно применяет логику вычисления. Поскольку проверка основана на сравнении строк, она не использует возможность исправлению формулы и не будет автоматически отражать переименование объектов. Для аналогичного сравнения, в котором используется исправление формулы, обратитесь к функции ISSLECTEDMEASURE.

```
IF (
    SELECTEDMEASURENAME = "Expense Ratio",
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

## См. также

[SELECTEDMEASURE](#)

[ISSELECTEDMEASURE](#)

# USERNAME

02.10.2020 • 2 minutes to read

Возвращает имя домена и имя пользователя из учетных данных, предоставленных системе во время подключения.

## Синтаксис

```
USERNAME()
```

### Параметры

У этого выражения нет параметров.

## Возвращаемое значение

Имя пользователя из учетных данных, предоставленных системе во время подключения.

## Пример

Следующая формула проверяет, является ли имя входа пользователя частью UsersTable.

```
= IF(CONTAINS(UsersTable,UsersTable[login], USERNAME()), "Allowed", BLANK())
```

# USEROBJECTID

22.09.2020 • 2 minutes to read

Возвращает идентификатор объекта текущего пользователя из Azure AD или ИД безопасности (SID).

## Синтаксис

```
USEROBJECTID()
```

### Параметры

У этого выражения нет параметров.

## Возвращаемое значение

Идентификатор объекта текущего пользователя из Azure AD для моделей Power BI или Azure Analysis Services либо SID для моделей SQL Server Analysis Services.

# USERPRINCIPALNAME

22.09.2020 • 2 minutes to read

Возвращает имя субъекта-пользователя.

## Синтаксис

```
USERPRINCIPALNAME()
```

### Параметры

У этого выражения нет параметров.

## Возвращаемое значение

Имя userprincipalname во время подключения.



# Логические функции

02.10.2020 • 2 minutes to read

Логические функции работают с выражениями и возвращают сведения об их значениях или наборах. Например, можно использовать функцию IF для проверки результата выражения и создания условных результатов.

## В этой категории

КОМПОНЕНТ	DESCRIPTION
AND	Проверяет, имеют ли оба аргумента значение TRUE, и возвращает значение TRUE, если оба аргумента имеют значение TRUE.
COALESCE	Возвращает первое выражение, которое не дает значение BLANK.
FALSE	Возвращает логическое значение FALSE.
IF	Проверяет условие и возвращает одно значение, если оно равно TRUE, в противном случае возвращает второе значение.
IFERROR	Вычисляет выражение и возвращает указанное значение, если выражение возвращает ошибку
NOT	Изменяет FALSE на TRUE или TRUE на FALSE.
OR	Проверяет, имеет ли один из аргументов значение TRUE, чтобы вернуть значение TRUE.
SWITCH	Вычисляет выражение по списку значений и возвращает одно из нескольких возможных результирующих выражений.
TRUE	Возвращает логическое значение TRUE.

# AND

02.10.2020 • 2 minutes to read

Проверяет, имеют ли оба аргумента значение TRUE, и возвращает значение TRUE, если оба аргумента имеют значение TRUE. В противном случае возвращает значение FALSE.

## Синтаксис

```
AND(<logical1>,<logical2>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
logical_1, logical_2	Проверяемые логические значения.

## Возвращаемое значение

Возвращает значение true или false в зависимости от сочетания проверяемых значений.

## Remarks

Функция **AND** в DAX принимает только два аргумента. Если необходимо выполнить операцию AND над несколькими выражениями, можно создать ряд вычислений или лучше использовать оператор AND ( && ), чтобы объединить их в более простое выражение.

## Пример 1

В следующей формуле показан синтаксис функции AND.

```
= IF(AND(10 > 9, -10 < -1), "All true", "One or more false")
```

Поскольку оба условия, переданные в качестве аргументов функции 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 продукта формула определяет, больше ли продажи через Интернет в текущем и предыдущим годам, чем продажи через торговых посредников за те же периоды. Если оба условия истинны, то для каждой категории формула возвращает значение "Через Интернет больше".

```

= 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"
, ""
)

```

## Результаты

ФУНКЦИЯ AND	МЕТКИ СТОЛБЦОВ					
Метки строк	2005	2006	2007 г.	2008		Grand Total
Полукомбине зоны						
Велосипедны е стойки						
Парковка для велосипедов				Через Интернет больше		
Фляги и кофры				Через Интернет больше		
Велосипедны е каретки						
Тормоза						
Ограничения						
Цепочки						
Чистящие средства						
Ведущие звезды						
Дерейлеры						
Бамперы				Через Интернет больше		
Вилки						
Перчатки						
Рули						

ФУНКЦИЯ AND	МЕТКИ СТОЛБЦОВ					
Гарнитуры						
Шлемы						
Гидраторы						
Веломайки						
Освещение						
Блокировки						
Горные велосипеды						
Горные рамы						
Кожухи						
Педали						
Насосы						
Дорожные велосипеды						
Дорожные рамы						
Седла						
Шорты						
Носки						
Чулки						
Шины и камеры				Через Интернет больше		
Туристическое велосипеды						
Туристическое рамы						
Жилеты						
Колеса						

ФУНКЦИЯ AND	МЕТКИ СТОЛБЦОВ					
Grand Total						

См. также раздел

[Логические функции](#)

# COALESCE

22.09.2020 • 2 minutes to read

Возвращает первое выражение, которое не дает значение BLANK. Если все выражения имеют значение BLANK, возвращается значение BLANK.

## Синтаксис

```
COALESCE(<expression>, <expression>[, <expression>]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Любое выражение DAX, которое возвращает скалярное выражение.

## Возвращаемое значение

Скалярное значение, возвращаемое одним из выражений, или значение BLANK, если все выражения имеют значение BLANK.

## Примечания

Входные выражения могут иметь разные типы данных.

## Пример 1

Рассмотрим следующий запрос DAX:

```
EVALUATE { COALESCE(BLANK(), 10, DATE(2008, 3, 3)) }
```

Возвращает 10 — первое выражение, которое не возвращает значение BLANK.

## Пример 2

Следующее выражение 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.

# FALSE

02.10.2020 • 2 minutes to read

Возвращает логическое значение FALSE.

## Синтаксис

```
FALSE()
```

## Возвращаемое значение

Всегда значение FALSE.

## Remarks

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.

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

В следующей таблице показаны результаты использования формулы в качестве примера с 'ProductCategory'[ProductCategoryName] в метках строк и 'DateTime'[CalendarYear] в метках столбцов.

TRUE-FALSE	МЕТКИ СТОЛБЦОВ					
Метки строк	2005	2006	2007 г.	2008		Grand Total
Accessories	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
Bikes	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
Clothing	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Components	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Grand Total	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE

## См. также раздел

[Функция TRUE](#)

[Функция NOT](#)





# IF

02.10.2020 • 3 minutes to read

Проверяет условие и возвращает одно значение, если оно равно TRUE, в противном случае возвращает второе значение.

## Синтаксис

```
IF(<logical_test>, <value_if_true>[, <value_if_false>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
logical_test	Любое значение или выражение, в результате вычисления которого можно получить значение TRUE или FALSE.
value_if_true	Значение, которое возвращается, если проверка логического выражения дает результат TRUE.
value_if_false	(Необязательно) Значение, которое возвращается, если проверка логического выражения дает результат FALSE. Если этот параметр опущен, возвращается значение BLANK.

## Возвращаемое значение

Либо **значение\_если\_true**, **значение\_если\_false**, либо значение BLANK.

## Remarks

- Функция IF может вернуть тип данных variant, если значения value\_if\_true и value\_if\_false относятся к разным типам данных. При этом функция попытается вернуть типа данных single, если значения value\_if\_true и value\_if\_false относятся к числовым типам данных. В последнем случае функция IF будет неявно преобразовывать типы данных для использования обоих значений.
- Например, формула `IF(<condition>, TRUE(), 0)` TRUE или 0. При этом формула `IF(<condition>, 1.0, 0)` возвращает только десятичные значения, даже если значение value\_if\_false относится к целочисленному типу данных. См. сведения о [неявном преобразовании типов данных](#).

## Примеры

В следующих определениях вычисляемых столбцов таблицы Product используется функция IF различными способами классификации каждого продукта по его прейскуранту.

В первом примере проверяется, является ли значение в столбце List Price меньше 500. Если это условие истинно, возвращается значение Low. Поскольку нет **значения\_если\_false**, возвращается значение BLANK.

*Примеры в этой статье можно добавить в образец модели Power BI Desktop. Чтобы получить модель, см.*

### пример модели DAX.

```
Price Group =  
IF(  
    'Product'[List Price] < 500,  
    "Low"  
)
```

Во втором примере используется тот же тест, но в этот раз включается **значение\_если\_false**. Таким образом, формула классифицирует каждый продукт как Low или High.

```
Price Group =  
IF(  
    'Product'[List Price] < 500,  
    "Low",  
    "High"  
)
```

В третьем примере используется 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.

```
Price Group =  
IF(  
    'Product'[List Price] < 500,  
    "Low",  
    IF(  
        'Product'[List Price] < 1500,  
        "Medium",  
        "High"  
    )  
)
```

#### ТИП

Если необходимо вложить несколько функций IF, лучше использовать функцию [SWITCH](#). Эта функция предоставляет более элегантный способ написания выражения, возвращающего более двух возможных значений.

## См. также

[Функция SWITCH \(DAX\)](#)

[Логические функции](#)

# IFERROR

02.10.2020 • 2 minutes to read

Вычисляет выражение и возвращает указанное значение, если выражение возвращает ошибку. В противном случае возвращается значение самого выражения.

## Синтаксис

```
IFERROR(value, value_if_error)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
value	Любое значение или выражение.
value_if_error	Любое значение или выражение.

## Возвращаемое значение

Скаляр того же типа, что и **значение**

## Remarks

- С помощью функции IFERROR можно перехватывать и обрабатывать ошибки в выражении.
- Если **value** или **value\_if\_error** является пустой ячейкой, функция IFERROR обрабатывает его как пустое строковое значение ("").
- Функция IFERROR основывается на функции IF и использует те же сообщения об ошибках, однако имеет меньше аргументов. Ниже описывается связь между функциями IFERROR и IF:

```
IFERROR(A,B) := IF(ISERROR(A), B, A)
```

Возвращаемые значения A и B должны иметь одинаковый тип данных. Это значит, что столбец или выражение, используемые для **value**, и возвращаемое для **value\_if\_error** значение должны иметь одинаковый 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, если при вычислении выражения 25/0 происходит ошибка, возвращается значение 9999. Если это выражение возвращает любое значение, кроме ошибки, в вызывающее выражение передается само это значение.

```
= IFERROR(25/0,9999)
```

См. также раздел

[Логические функции](#)

# NOT

02.10.2020 • 2 minutes to read

Изменяет FALSE на TRUE или TRUE на FALSE.

## Синтаксис

```
NOT(<logical>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
логические	Значение или выражение, в результате вычисления которого можно получить значение TRUE или FALSE.

## Возвращаемое значение

TRUE или FALSE.

## Пример

В следующем примере извлекаются значения из вычисляемого столбца, который был создан для иллюстрации функции IF. В этом примере вычисляемому столбцу присваивается имя по умолчанию, **Calculated Column1**, и он содержит следующую формулу: `= IF([Orders]<300,"true","false")`

Формула проверяет значение в столбце [Orders] и возвращает "true", если число заказов не превышает 300.

Теперь создайте новый вычисляемый столбец, **Calculated Column2**, и введите следующую формулу.

```
= NOT([CalculatedColumn1])
```

Для каждой строки **Calculated Column1** значения "true" и "false" интерпретируются как логические значения TRUE или FALSE, а функция NOT возвращает логическое противоположное значение.

## См. также раздел

[Функция TRUE](#)

[Функция FALSE](#)

[IF, функция](#)

# OR

22.09.2020 • 3 minutes to read

Проверяет, имеет ли один из аргументов значение TRUE, чтобы вернуть значение TRUE. Функция возвращает значение FALSE, если оба аргумента имеют значение FALSE.

## Синтаксис

```
OR(<logical1>,<logical2>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
logical_1, logical_2	Проверяемые логические значения.

## Возвращаемое значение

Значение типа Boolean. Значение равно TRUE, если любой из двух аргументов имеет значение TRUE; значение равно FALSE, если оба аргумента имеют значение FALSE.

## Remarks

- Функция OR в DAX принимает только два аргумента. Если необходимо выполнить операцию OR над несколькими выражениями, можно создать ряд вычислений или лучше использовать оператор OR ( || ), чтобы объединить их в более простое выражение.
- Функция вычисляет аргументы до первого аргумента TRUE, а затем возвращает значение TRUE.

## Пример

В следующем примере показано, как использовать функцию OR для получения продавцов, относящихся к категории Circle of Excellence. Это те, кто выручил больше миллиона долларов США на продаже туристических велосипедов или более двух с половиной миллионов долларов за весь 2007 г.

```
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"
, ""
)
```

### Результаты

SALESPERSON FLAG	TRUE					

SALESPERSON FLAG	TRUE					
Функция OR	Метки столбцов					
Метки строк	2005	2006	2007 г.	2008		Grand Total
Abbas, Syed E						
Alberts, Amy E						
Ansman- Wolfe, Pamela O						
Blythe, Michael G	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Campbell, David R						
Carson, Jillian	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Ito, Shu K						
Jiang, Stephen Y						
Mensa- Annan, Tete A						
Mitchell, Linda C	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Pak, Jae B	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Reiter, Tsvi Michael						
Saraiva, José Edvaldo	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence
Tsofilias, Lynn N						
Valdez, Rachel B						
Vargas, Garrett R						

SALESPERSON FLAG	TRUE					
Varkey Chudukatil, Ranjit R						Circle of Excellence
Grand Total	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence	Circle of Excellence

См. также раздел

[Логические функции](#)



# SWITCH

22.09.2020 • 2 minutes to read

Вычисляет выражение по списку значений и возвращает одно из нескольких возможных результирующих выражений.

## Синтаксис

```
SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).
value	Постоянное значение, которому должны соответствовать результаты <i>expression</i> .
набор по	Любое скалярное выражение, которое вычисляется, если результаты <i>expression</i> совпадают с соответствующим <i>value</i> .
else	Любое скалярное выражение, которое вычисляется, если результаты <i>expression</i> не совпадают ни с одним из аргументов <i>value</i> .

## Возвращаемое значение

Скалярное значение, поступающее от одного из выражений *result*, если обнаружено совпадение с *value*, или из выражения *else*, если нет совпадения с *value*.

## Remarks

Все выражения *result* и выражение *else* должны иметь одинаковый тип данных.

## Пример

В следующем примере создается вычисляемый столбец названий месяцев.

```
= SWITCH([Month], 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" )
```

# TRUE

02.10.2020 • 2 minutes to read

Возвращает логическое значение TRUE.

## Синтаксис

```
TRUE()
```

## Возвращаемое значение

Всегда TRUE.

## Remarks

Слово TRUE также интерпретируется как логическое значение TRUE.

## Пример

Формула возвращает логическое значение TRUE, если значение в столбце 'InternetSales\_USD'[SalesAmount\_USD] больше 200 000.

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

В следующей таблице показаны результаты использования формулы в качестве примера в сводной таблице или визуализации с 'ProductCategory'[ProductCategoryName] в метках строк и 'DateTime'[CalendarYear] в метках столбцов.

TRUE-FALSE	МЕТКИ СТОЛБЦОВ					
Метки строк	2005	2006	2007 г.	2008		Grand Total
Accessories	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
Bikes	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
Clothing	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Components	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Grand Total	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE

## См. также раздел

[FALSE](#)

NOT  
IF

# Математические и тригонометрические функции

21.09.2020 • 4 minutes to read

Математические функции в выражениях анализа данных (DAX) очень похожи на математические и тригонометрические функции Excel. В этом разделе перечислены математические функции, представленные в DAX.

## В этой категории

КОМПОНЕНТ	DESCRIPTION
<a href="#">ABS</a>	Возвращает абсолютное значение числа.
<a href="#">ACOS</a>	Возвращает арккосинус или обратный косинус числа.
<a href="#">ACOSH</a>	Возвращает обратный гиперболический косинус числа.
<a href="#">ACOT</a>	Возвращает арккотангенс (обратный котангенс) угла.
<a href="#">ACOTH</a>	Возвращает обратный гиперболический котангенс угла.
<a href="#">ASIN</a>	Возвращает арксинус или обратный синус числа.
<a href="#">ASINH</a>	Возвращает обратный гиперболический синус числа.
<a href="#">ATAN</a>	Возвращает арктангенс или обратный тангенс числа.
<a href="#">ATANH</a>	Возвращает обратный гиперболический тангенс числа.
<a href="#">CEILING</a>	Округляет число до ближайшего большего целого или ближайшего большего числа, кратного заданной значимости.
<a href="#">COMBIN</a>	Возвращает число комбинаций для заданного числа элементов.
<a href="#">COMBINA</a>	Возвращает число комбинаций (с повторениями) для заданного числа элементов.
<a href="#">CONVERT</a>	Преобразует выражение одного типа данных в другой.
<a href="#">COS</a>	Возвращает косинус заданного угла.
<a href="#">COSH</a>	Возвращает гиперболический косинус числа.
<a href="#">CURRENCY</a>	Вычисляет аргумент и возвращает результат в виде денежного типа данных.
<a href="#">DEGREES</a>	Преобразует радианы в градусы.

КОМПОНЕНТ	DESCRIPTION
DIVIDE	Выполняет деление и возвращает альтернативный результат или выражение BLANK() при делении на 0.
EVEN	Возвращает число, округленное до ближайшего четного целого.
EXP	Возвращает число $e$ , возведенное в заданную степень.
FACT	Возвращает факториал числа, равный ряду $1 * 2 * 3 * \dots$ , заканчивая заданным числом.
FLOOR	Округляет число к нулю до ближайшего числа, кратного заданной значимости.
GCD	Возвращает наибольший общий делитель для двух или более целых чисел.
INT	Округляет число в меньшую сторону до ближайшего целого.
ISO.CEILING	Округляет число до ближайшего большего целого или ближайшего большего числа, кратного заданной значимости.
LCM	Возвращает наименьшее общее кратное целых чисел.
LN	Возвращает натуральный логарифм числа.
LOG	Возвращает логарифм числа по заданному основанию.
LOG10	Возвращает десятичный логарифм числа.
MROUND	Возвращает число, округленное до нужного кратного.
ODD	Возвращает число, округленное до ближайшего нечетного целого.
PI	Возвращает число Пи (3,14159265358979) с точностью до 15 знаков.
POWER	Возвращает результат возведения числа в степень.
PRODUCT	Возвращает произведение чисел в столбце.
PRODUCTX	Возвращает произведение значений выражения, вычисляемого для каждой строки в таблице.
QUOTIENT	Выполняет деление и возвращает только целую часть результата деления.
RADIANS	Преобразует градусы в радианы.

КОМПОНЕНТ	DESCRIPTION
RAND	Возвращает случайное число не меньше 0 и меньше 1 с равномерным распределением.
RANDBETWEEN	Возвращает случайное число в диапазоне между двумя указанными числами.
ROUND	Округляет число до указанного количества десятичных разрядов.
ROUNDDOWN	Округляет число в меньшую сторону, к нулю.
ROUNDUP	Округляет число в большую сторону, от нуля.
SIGN	Определяет знак числа, результат вычисления или значение в столбце.
SQRT	Возвращает квадратный корень числа.
SUM	Добавляет все числа в столбце.
SUMX	Возвращает сумму значений выражения, вычисляемого для каждой строки в таблице.
TRUNC	Усекает число до целого путем удаления десятичной или дробной части.

# ABS

02.10.2020 • 2 minutes to read

Возвращает абсолютное значение числа.

## Синтаксис

```
ABS(<number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Число, абсолютное значение которого требуется определить.

## Возвращаемое значение

Десятичное число.

## Remarks

Абсолютное значение числа — это целое или десятичное число без знака. Функцию ABS можно использовать, чтобы гарантировать, что только неотрицательные числа возвращаются из выражений при вложении в функции, требующие положительного числа.

## Пример

В следующем примере возвращается абсолютное значение разницы между ценой по прейскуранту и ценой у дилера, которую можно использовать в новом вычисляемом столбце **DealerMarkup**.

```
= ABS([DealerPrice]-[ListPrice])
```

## См. также раздел

[Математические и тригонометрические функции](#)  
[SIGN, функция](#)

# ACOS

02.10.2020 • 2 minutes to read

Возвращает арккосинус или обратный косинус числа. Арккосинус — это угол, косинус которого является *числом*. Возвращаемый угол указывается в радианах в диапазоне от 0 (нуля) до числа  $\pi$ .

## Синтаксис

ACOS(number)

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Number	Косинус нужного угла должен быть задан в пределах от $-1$ до $1$ .

## Возвращаемое значение

Возвращает арккосинус или обратный косинус числа.

## Remarks

Если требуется преобразовать результат из радиан в градусы, умножьте его на  $180/\pi()$  или используйте функцию DEGREES.

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= ACOS(-0,5)	Арккосинус числа $-0,5$ в радианах, $2*\pi/3$ .	2,094395102
= ACOS(-0,5)*180/PI()	Арккосинус числа $-0,5$ в градусах.	120



# ACOSH

02.10.2020 • 2 minutes to read

Возвращает обратный гиперболический косинус числа. Число должно не меньше 1. Обратный гиперболический косинус — это значение, гиперболический косинус которого равен заданному *числу*, поэтому выражение ACOSH(COSH(число)) равно изначальному числу.

## Синтаксис

ACOSH(number)

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Любое вещественное число не меньше 1.

## Возвращаемое значение

Возвращает обратный гиперболический косинус числа.

## Remarks

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

## Пример

ФОРМУЛА	ОПИСАНИЕ	РЕЗУЛЬТАТ
= ACOSH(1)	Обратный гиперболический косинус 1.	0
= ACOSH(10)	Обратный гиперболический косинус 10.	2,993228

# ACOT

22.09.2020 • 2 minutes to read

Возвращает главное значение арккотангенса (обратного котангенса) числа.

## Синтаксис

```
ACOT(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Number	Косинус нужного угла. Должен быть вещественным числом.

## Возвращаемое значение

Одно десятичное значение.

# ACOT

22.09.2020 • 2 minutes to read

Возвращает обратный гиперболический котангенс угла.

## Синтаксис

```
ACOTH(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Number	Абсолютное значение числа должно быть больше 1.

## Возвращаемое значение

Одно десятичное значение.

## Комментарии

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

# ASIN

02.10.2020 • 2 minutes to read

Возвращает арксинус или обратный синус числа. Арксинус — это угол, синус которого является *number*. Возвращаемый угол указывается в радианах в диапазоне от  $-\pi/2$  до  $\pi/2$ .

## Синтаксис

ASIN(*number*)

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Синус нужного угла должен быть задан в пределах от $-1$ до $1$ .

## Возвращаемое значение

Возвращает арксинус или обратный синус числа.

## Remarks

Чтобы выразить арксинус в градусах, умножьте результат на  $180/\pi()$  или используйте функцию DEGREES.

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= ASIN(-0,5)	Арксинус числа $-0,5$ в радианах ( $-\pi/6$ )	$-0,523598776$
= ASIN(-0,5)*180/PI()	Арксинус числа $-0,5$ в градусах	$-30$
= DEGREES(ASIN(-0,5))	Арксинус числа $-0,5$ в градусах	$-30$

# ASINH

02.10.2020 • 2 minutes to read

Возвращает обратный гиперболический синус числа. Обратный гиперболический синус — это значение, гиперболический синус которого равен заданному *числу*, поэтому выражение ASINH(SINH(число)) равно *изначальному числу*.

## Синтаксис

```
ASINH(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Любое вещественное число.

## Возвращаемое значение

Возвращает обратный гиперболический синус числа.

## Remarks

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

## Пример

ФОРМУЛА	ОПИСАНИЕ	РЕЗУЛЬТАТ
= ASINH(-2,5)	Обратный гиперболический синус – 2,5	-1,647231146
= ASINH(10)	Обратный гиперболический синус 10	2,99822295

# ATAN

02.10.2020 • 2 minutes to read

Возвращает арктангенс или обратный тангенс числа. Арктангенс — это угол, тангенс которого является *number*. Возвращаемый угол указывается в радианах в диапазоне от  $-\pi/2$  до  $\pi/2$ .

## Синтаксис

```
ATAN(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Тангенс нужного угла.

## Возвращаемое значение

Возвращает обратный гиперболический тангенс числа.

## Remarks

Чтобы выразить арктангенс в градусах, умножьте результат на  $180/\pi()$  или используйте функцию DEGREES.

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= ATAN(1)	Арктангенс числа 1 в радианах, $\pi/4$	0,785398163
= ATAN(1)*180/PI()	Арктангенс числа 1 в градусах	45

# ATANH

02.10.2020 • 2 minutes to read

Возвращает обратный гиперболический тангенс числа. Число должно находиться в диапазоне от  $-1$  до  $1$  (исключая  $-1$  и  $1$ ). Обратный гиперболический тангенс — это значение, гиперболический тангенс которого равен заданному *числу*, поэтому выражение  $\text{ATANH}(\text{TANH}(\text{number}))$  равно *изначальному числу*.

## Синтаксис

$\text{ATANH}(\text{number})$

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Любое вещественное число от $-1$ до $1$ .

## Возвращаемое значение

Возвращает обратный гиперболический тангенс числа.

## Remarks

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

## Пример

ФОРМУЛА	ОПИСАНИЕ	РЕЗУЛЬТАТ
$= \text{ATANH}(0,76159416)$	Обратный гиперболический тангенс 0,76159416	1,00000001
$= \text{ATANH}(-0,1)$		$-0,100335348$

## См. также

[Функция ATAN](#)

# CEILING

02.10.2020 • 2 minutes to read

Округляет число до ближайшего большего целого или ближайшего большего числа, кратного заданной значимости.

## Синтаксис

```
CEILING(<number>, <significance>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Число, которое требуется округлить, или ссылка на столбец, содержащий числа.
significance	Кратность, до которой требуется округление. Например, чтобы округлить до ближайшего целого числа, введите 1.

## Возвращаемое значение

Число, округленное согласно указанию.

## Remarks

- В DAX есть две функции CEILING со следующими отличиями.
  - Функция CEILING эмулирует поведение функции CEILING в Excel.
  - Функция ISO. CEILING соответствует поведению, определяемому стандартом ISO, для определения значения "потолка".
- Эти две функции возвращают одно и то же значение для положительных чисел, но разные значения для отрицательных чисел. При указании положительного значения significance CEILING и ISO.CEILING округляют отрицательные числа вверх (в сторону положительной бесконечности). При использовании отрицательного значения significance функция CEILING округляет отрицательные числа вниз (в сторону отрицательной бесконечности), а ISO.CEILING округляет отрицательные числа вверх (в сторону положительной бесконечности).
- Возвращаемое значение обычно имеет тот же тип, что и значащий аргумент, за исключением следующих случаев:
  - Если числовой аргумент имеет тип Currency, то возвращаемым типом будет Currency.
  - Если тип аргумента significance логический, то возвращаемым типом будет целое число.
  - Если тип аргумента significance нечисловой, то возвращаемым типом будет действительное число.

## Пример 1

Следующая формула возвращает 4,45. Это может быть полезно, если вы хотите избежать использования более мелких единиц в ценах. Если имеющийся продукт стоит 4,42 долл. США, можно использовать CEILING



для округления цен до ближайшей единицы в пять центов.

```
= CEILING(4.42,0.05)
```

## Пример 2

Следующая формула возвращает результаты, аналогичные предыдущему примеру, но использует числовые значения, хранящиеся в столбце **ProductPrice**.

```
= CEILING([ProductPrice],0.05)
```

## См. также раздел

[Математические и тригонометрические функции](#)

[Функция FLOOR](#)

[Функция ISO.CEILING](#)

[ROUNDUP, функция](#)

# COMBIN

02.10.2020 • 2 minutes to read

Возвращает число комбинаций для заданного числа элементов. Используйте функцию COMBINABLE, чтобы определить общее возможное количество групп для заданного числа элементов.

## Синтаксис

```
COMBIN(number, number_chosen)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Число элементов.
number_chosen	число элементов в каждой комбинации.

## Возвращаемое значение

Возвращает число комбинаций для заданного числа элементов.

## Remarks

- Числовые аргументы усекаются до целых значений.
- Если любой из аргументов является нечисловым, функция COMBIN возвращает #VALUE! (значение ошибки).
- Если  $\text{number} < 0$ ,  $\text{number\_chosen} < 0$  или  $\text{number} < \text{number\_chosen}$ , функция COMBIN возвращает #NUM! (значение ошибки).
- Комбинация — это любой набор или подмножество элементов, независимо от их внутреннего порядка. Комбинации отличаются от перестановок, для которых внутренний порядок важен.
- Число комбинаций выглядит следующим образом, где  $\text{number} = n$  и  $\text{number\_chosen} = k$ :

$${}^n\text{C}_k = \frac{P_{k,n}}{k!} = \frac{n!}{k!(n-k)!}$$

Where

$$P_{k,n} = \frac{n!}{(n-k)!}$$

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

## Пример

ФОРМУЛА	ОПИСАНИЕ	РЕЗУЛЬТАТ
---------	----------	-----------

ФОРМУЛА	ОПИСАНИЕ	РЕЗУЛЬТАТ
= COMBIN(8,2)	Возможные команды из двух человек, которые можно формировать из восьми кандидатов.	28

# COMBINA

02.10.2020 • 2 minutes to read

Возвращает число комбинаций (с повторениями) для заданного числа элементов.

## Синтаксис

```
COMBINA(number, number_chosen)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Должно быть не меньше 0 и не меньше чем number_chosen. Нецелочисленные значения усекаются.
number_chosen	Должно быть больше или равно 0. Нецелочисленные значения усекаются.

## Возвращаемое значение

Возвращает число комбинаций (с повторениями) для заданного числа элементов.

## Remarks

- Если значение любого из аргументов выходит за пределы их ограничений, функция COMBINABLE возвращает #NUM! (значение ошибки).
- Если любой из аргументов не является числовым, то функция COMBINA возвращает #VALUE! (значение ошибки).
- Используется следующее уравнение, где \$N\$ — это number, а \$M\$ — number\_chosen:

$$N! / (N - M)! \times M!$$

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

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= COMBINA(4,3)	Возвращает число комбинаций (с повторениями) для 4 и 3.	20
= COMBINA(10,3)	Возвращает число комбинаций (с повторениями) для 10 и 3.	220

# CONVERT

02.10.2020 • 2 minutes to read

Преобразует выражение из одного типа данных в другой.

## Синтаксис

```
CONVERT(<Expression>, <Datatype>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Выражение	Любое допустимое выражение.
Datatype	Перечисление, включающее: INTEGER (целое число), DOUBLE (десятичное число), STRING (текст), BOOLEAN (true или false), CURRENCY (число с фиксированной десятичной запятой), DATETIME (дата, время и т. д.).

## Возвращаемое значение

Возвращает значение <Expression>, переведенное в <Datatype>.

## Примечания

- Функция возвращает ошибку, если значение не может быть преобразовано в указанный тип данных.
- Вычисляемые столбцы DAX должны иметь один тип данных. Так как функции MEDIAN и MEDIANX над целочисленным столбцом возвращают смешанные типы данных (Integer или Double), следующее выражение вычисляемого столбца в результате вернет ошибку:

```
MedianNumberCarsOwned = MEDIAN(DimCustomer[NumberCarsOwned]) .
```

- Чтобы избежать смешанных типов данных, измените выражение так, чтобы оно всегда возвращало тип данных Double, 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

```
EVALUATE { CONVERT(DATE(1900, 1, 1), INTEGER) }
```

Возвращает

[ЗНАЧЕНИЕ]
2

# COS

02.10.2020 • 2 minutes to read

Возвращает косинус заданного угла.

## Синтаксис

```
COS(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Обязательный элемент. Угол в радианах, косинус которого нужно найти.

## Возвращаемое значение

Возвращает косинус заданного угла.

## Remarks

Если угол задан в градусах, то необходимо умножить угол на  $\text{PI}()/180$  или использовать функцию RADIANS для преобразования угла в радианы.

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= COS(1,047)	Косинус 1,047 радианы	0,5001711
= COS(60*PI()/180)	Косинус 60 градусов	0,5
= COS(RADIANS(60))	Косинус 60 градусов	0,5

# COSH

02.10.2020 • 2 minutes to read

Возвращает гиперболический косинус числа.

## Синтаксис

`COSH(number)`

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Обязательный элемент. Любое вещественное число, гиперболический косинус которого нужно найти.

## Возвращаемое значение

Гиперболический косинус числа.

## Remarks

- Формула гиперболического косинуса:

$$\text{COSH}(z) = \frac{e^z + e^{-z}}{2}$$

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

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= COSH(4)	Гиперболический косинус 4.	27,308233
= COSH(EXP(1))	Гиперболический косинус основания натурального логарифма.	7,6101251



# CURRENCY

22.09.2020 • 2 minutes to read

Вычисляет аргумент и возвращает результат в виде денежного типа данных.

## Синтаксис

```
CURRENCY(<value>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
value	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется только один раз перед выполнением остальных операций.

## Возвращаемое значение

Значение выражения вычисляется и возвращается как значение типа валюты.

## Remarks

- Функция CURRENCY округляет пятый значащий десятичный знак в значении для возврата четвертой десятичной цифры; округление выполняется, если пятый значащий десятичный знак равен или больше 5. Например, если значение равно 3,66666666666666, то преобразование в валюту возвращает \$3,6667; однако если значение равно 3,0123456789, то преобразование в валюту возвращает \$3,0123.
- Если тип данных выражения — TrueFalse, то CURRENCY(<TrueFalse>) возвратит \$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 выполнено, то число будет преобразовано в денежное значение, в противном случае возвращается ошибка.
- Если тип данных выражения — DateTime, то CURRENCY(<DateTime>) преобразует значение DateTime в число, а это число — в денежное. Значения DateTime имеют целую часть, представляющую количество дней между заданной датой и 1900-03-01, и дробную часть дня (где 12 часов или полдень составляет 0,5 дня). Если значение выражения не является правильным значением DateTime, возвращается ошибка.

## Пример

Преобразование числа 1234,56 в денежный тип данных.

```
= CURRENCY(1234.56)
```

Возвращает значение \$1234,5600.

# DEGREES

02.10.2020 • 2 minutes to read

Преобразует радианы в градусы.

## Синтаксис

```
DEGREES(angle)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
angle	Обязательный элемент. Угол в радианах, который нужно преобразовать.

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= DEGREES(PI())	Градусы Пи радиан	180

# DIVIDE

02.10.2020 • 2 minutes to read

Выполняет деление и возвращает альтернативный результат или выражение BLANK() при делении на 0.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
numerator	Делимое число.
denominator	Делитель.
alternateresult	(Необязательно) Значение, возвращаемое когда деление на ноль приводит к ошибке. Если не указано, значение по умолчанию является выражением BLANK().

## Возвращаемое значение

Десятичное число.

## Remarks

Альтернативный результат при делении на 0 должен быть константой.

## Пример

В следующем примере будет возвращено число 2,5.

```
= DIVIDE(5,2)
```

## Пример 1

В следующем примере будет возвращено значение BLANK.

```
= DIVIDE(5,0)
```

## Пример 2

В следующем примере будет возвращено число 1.

```
= DIVIDE(5,0,1)
```

См. также раздел

[QUOTIENT, функция](#)

[Математические и тригонометрические функции](#)

# EVEN

02.10.2020 • 2 minutes to read

Возвращает число, округленное до ближайшего четного целого. Вы можете использовать эту функцию для обработки парных элементов. Например, упаковочный ящик принимает элементы по одному или по двум. Ящик считается полным, когда количество элементов, округленное до ближайшей пары, совпадает с емкостью ящика.

## Синтаксис

EVEN(number)

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Округляемое значение.

## Возвращаемое значение

Возвращает число, округленное до ближайшего четного целого.

## Remarks

- Если значение параметра number является нечисловым, функция EVEN возвращает #VALUE! (значение ошибки).
- Независимо от знака числа, значение округляется в большую сторону, если оно корректируется от нуля. Если number является четным целым числом, округление не выполняется.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= EVEN(1,5)	Округляет число 1,5 до ближайшего четного целого	2
= EVEN(3)	Округляет число 3 до ближайшего четного целого	4
= EVEN(2)	Округляет число 2 до ближайшего четного целого	2
= EVEN(-1)	Округляет число -1 до ближайшего четного целого	-2

# EXP

02.10.2020 • 2 minutes to read

Возвращает число  $e$ , возведенное в заданную степень. Константа  $e$  примерно равна 2,71828182845904 и служит основанием натуральных логарифмов.

## Синтаксис

EXP(<number>)

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Показатель степени, в которую возводится число $e$ . Константа $e$ примерно равна 2,71828182845904 и служит основанием натуральных логарифмов.

## Возвращаемое значение

Десятичное число.

## Исключения

## Remarks

- Функция EXP является обратной для функции LN, которая возвращает натуральный логарифм заданного числа.
- Для вычисления степеней по основанию, отличному от  $e$ , используйте оператор возведения в степень (^). Дополнительные сведения см. в [справочнике оператора DAX](#).

## Пример

С помощью следующей формулы число  $e$  возводится в степень, задаваемую числом из столбца [Power] .

= EXP([Power])

## См. также раздел

[Математические и тригонометрические функции](#)

[LN, функция](#)

[EXP, функция](#)

[LOG, функция](#)

[LOG, функция](#)

# FACT

02.10.2020 • 2 minutes to read

Возвращает факториал числа, равный ряду  $1 * 2 * 3 * \dots *$ , заканчивая заданным числом.

## Синтаксис

```
FACT(<number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Неотрицательное число, для которого нужно вычислить факториал.

## Возвращаемое значение

Десятичное число.

## Remarks

- Если число не является целым, оно усекается 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] .

```
= FACT([Values])
```

В следующей таблице показаны ожидаемые результаты.

ЗНАЧЕНИЯ	РЕЗУЛЬТАТЫ
0	1
1	1
2	2
3	6
4	24

ЗНАЧЕНИЯ	РЕЗУЛЬТАТЫ
5	120
170	7.257415615308E+306

См. также раздел

[Математические и тригонометрические функции](#)  
[TRUNC, функция](#)



# FLOOR

02.10.2020 • 2 minutes to read

Округляет число к нулю до ближайшего числа, кратного заданной значимости.

## Синтаксис

```
FLOOR(<number>, <significance>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Округляемое числовое значение.
significance	Кратное, до которого требуется округление. Аргументы <b>number</b> и <b>significance</b> должны иметь один и тот же знак.

## Возвращаемое значение

Десятичное число.

## Remarks

- Если любой из аргументов является нечисловым, функция FLOOR возвращает значение ошибки #VALUE! .
- Если аргументы number и significance имеют разные знаки, функция FLOOR возвращает значение ошибки #NUM! .
- Независимо от знака числа, значение округляется в меньшую сторону, если оно корректируется от нуля. Если аргумент number является точным кратным аргумента significance, округление не выполняется.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Следующая формула получает значение из столбца [Total Product Cost] таблицы InternetSales и округляет его к нулю до ближайшего значения, кратного 0,1.

```
= FLOOR(InternetSales[Total Product Cost],.5)
```

В следующей таблице показаны ожидаемые результаты для некоторых примеров.

ЗНАЧЕНИЯ	ОЖИДАЕМЫЙ РЕЗУЛЬТАТ
10,8423	10,8
8,0373	8
2,9733	2,9

См. также раздел

[Математические и тригонометрические функции](#)

# GCD

02.10.2020 • 2 minutes to read

Возвращает наибольший общий делитель для двух или более целых чисел. Наибольший общий делитель — это максимальное целое число, на которое без остатка делятся число 1 и число 2.

## Синтаксис

```
GCD(number1, [number2], ...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number1, number2, ...	Первый аргумент является обязательным, последующие числа являются необязательными. Поддерживаются значения от 1 до 255. Если любое из значений не является целым, происходит усечение.

## Возвращаемое значение

Наибольший общий делитель для двух или более целых чисел.

## Remarks

- Если любой из аргументов является нечисловым, функция GCD возвращает #VALUE! (значение ошибки).
- Если любой из аргументов является отрицательным, функция GCD возвращает #NUM! (значение ошибки).
- Единица является делителем для любого числа.
- Простое число делится без остатка только на единицу и на само себя.
- Если в функцию GCD передан параметр, значение которого  $\geq 2^{53}$ , функция GCD возвращает #NUM! (значение ошибки).
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= GCD(5, 2)	Наибольший общий делитель для чисел 5 и 2.	1
= GCD(24, 36)	Наибольший общий делитель для чисел 24 и 36.	12

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= GCD(7, 1)	Наибольший общий делитель для чисел 7и 1.	1

# INT

02.10.2020 • 2 minutes to read

Округляет число в меньшую сторону до ближайшего целого.

## Синтаксис

```
INT(<number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Число, округляемое в меньшую сторону до целого.

## Возвращаемое значение

Целое число.

## Remarks

Функции TRUNC и INT похожи в том, что обе возвращают целые числа. TRUNC удаляет дробную часть числа. INT округляет число до ближайшего меньшего целого, исходя из значения дробной части числа. Функции INT и TRUNC различаются только при использовании для отрицательных чисел: `TRUNC(-4.3)` возвращает значение `-4`, но `INT(-4.3)` возвращает `-5`, так как `-5` — это меньшее число.

## Пример

В следующем выражении значение 1 округляется до 1: При использовании функции ROUND результат будет равен 2.

```
= INT(1.5)
```

## См. также раздел

[Математические и тригонометрические функции](#)

[Функция ROUND](#)

[ROUNDUP, функция](#)

[Функция ROUNDDOWN](#)

[MROUND, функция](#)

# ISO.CEILING

02.10.2020 • 3 minutes to read

Округляет число до ближайшего большего целого или ближайшего большего числа, кратного заданной значимости.

## Синтаксис

```
ISO.CEILING(<number>[, <significance>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
число	Число, которое требуется округлить, или ссылка на столбец, содержащий числа.
significance	(необязательно) Кратность, до которой требуется округление. Например, чтобы округлить до ближайшего целого числа, введите 1. Если значение significance не указано, число округляется до ближайшего целого.

## Возвращаемое значение

Число с типом, совпадающим с аргументом *number*, округленное как указано.

## Примечания

В DAX есть две функции CEILING со следующими отличиями.

- Функция CEILING эмулирует поведение функции CEILING в Excel.
- Функция ISO.CEILING соответствует поведению, определяемому стандартом ISO, для определения значения "потолка".

Эти две функции возвращают одно и то же значение для положительных чисел, но разные значения для отрицательных чисел. При указании положительного значения significance CEILING и ISO.CEILING округляют отрицательные числа вверх (в сторону положительной бесконечности). При использовании отрицательного значения significance функция CEILING округляет отрицательные числа вниз (в сторону отрицательной бесконечности), а ISO.CEILING округляет отрицательные числа вверх (в сторону положительной бесконечности).

Возвращаемое значение обычно имеет тот же тип, что и аргумент significance, за исключением следующих случаев:

- Если первый аргумент имеет тип Currency, результат будет иметь тип Currency.
- Если необязательный аргумент не включен, результат имеет целочисленный тип.
- Если аргумент significance имеет логический тип, результат имеет целочисленный тип.
- Если аргумент significance имеет нечисловой тип, результат имеет вещественный тип.

## Пример: Положительные числа

Следующая формула возвращает 4,45. Это может быть полезно, если вы хотите избежать использования более мелких единиц в ценах. Если имеющийся продукт стоит 4,42 долл. США, можно использовать ISO.CEILING для округления цен до ближайшей единицы в пять центов.

```
= ISO.CEILING(4.42,0.05)
```

## Пример: Отрицательные числа

Следующая формула возвращает по ISO значение −4,40.

```
= ISO.CEILING(-4.42,0.05)
```

## См. также

[Математические и тригонометрические функции](#)

[Функция FLOOR](#)

[CEILING, функция](#)

[ROUNDUP, функция](#)

# LCM

02.10.2020 • 2 minutes to read

Возвращает наименьшее общее кратное целых чисел. Наименьшее общее кратное является наименьшим положительным целым числом, которое кратно всем целочисленным аргументам (number1, number2 и т. д.). Используйте LCM для добавления дробей с разными знаменателями.

## Синтаксис

```
LCM(number1, [number2], ...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number1, number2,...	Первый аргумент является обязательным, последующие числа являются необязательными. От 1 до 255 значений, для которых нужно найти наименьшее общее кратное. Если значение не целое, происходит усечение.

## Возвращаемое значение

Возвращает наименьшее общее кратное целых чисел.

## Remarks

- Если аргумент является нечисловым, функция LCM возвращает #VALUE! (значение ошибки).
- Если аргумент отрицателен, функция LCM возвращает #VALUE! (значение ошибки).
- Если  $LCM(a,b) > 2^{53}$ , LCM возвращает #NUM! (значение ошибки).
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= LCM(5, 2)	Наименьшее общее кратное 5 и 2.	10
= LCM(24, 36)	Наименьшее общее кратное 24 и 36.	72



# LN

02.10.2020 • 2 minutes to read

Возвращает натуральный логарифм числа. Натуральный логарифм основан на константе  $e$  (2,71828182845904).

## Синтаксис

```
LN(<number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Положительное число, для которого вычисляется натуральный логарифм.

## Возвращаемое значение

Десятичное число.

## Remarks

LN является инверсией функции EXP.

## Пример

В следующем примере возвращается натуральный логарифм числа в столбце [Values] .

```
= LN([Values])
```

## См. также раздел

[Математические и тригонометрические функции](#)

[EXP, функция](#)

# LOG

02.10.2020 • 2 minutes to read

Возвращает логарифм числа по заданному основанию.

## Синтаксис

```
LOG(<number>,<base>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Положительное число, для которого вычисляется логарифм.
base;	Основание логарифма. По умолчанию основание равно 10.

## Возвращаемое значение

Десятичное число.

## Remarks

Если значение слишком велико для отображения, может появиться сообщение об ошибке.

Функция LOG10 действует похоже, но всегда возвращает общий логарифм, то есть логарифм по основанию 10.

## Пример

Следующие формулы возвращают один и тот же результат — 2.

```
= LOG(100,10)  
= LOG(100)  
= LOG10(100)
```

## См. также раздел

[Математические и тригонометрические функции](#)

[EXP, функция](#)

[LOG, функция](#)

[LOG, функция](#)

# LOG10

02.10.2020 • 2 minutes to read

Возвращает десятичный логарифм числа.

## Синтаксис

```
LOG10(<number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Положительное число, для которого вычисляется десятичный логарифм.

## Возвращаемое значение

Десятичное число.

## Remarks

Функция LOG позволяет изменить основание логарифма вместо использования основания 10.

## Пример

Следующие формулы возвращают один и тот же результат — 2:

```
= LOG(100,10)  
= LOG(100)  
= LOG10(100)
```

## См. также раздел

[Математические и тригонометрические функции](#)

[EXP, функция](#)

[LOG, функция](#)

[LOG, функция](#)

# MOD

02.10.2020 • 2 minutes to read

Возвращает остаток от деления числа на делитель. Результат всегда имеет тот же знак, что и делитель.

## Синтаксис

```
MOD(<number>, <divisor>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
число	Число, остаток от деления которого вычисляется.
делитель	Число, на которое делится делимое.

## Возвращаемое значение

Целое число.

## Примечания

- Если делитель равен 0 (нулю), MOD возвращает ошибку. Нельзя делить на 0.
- Функция MOD может быть выражена в терминах функции INT:  $MOD(n, d) = n - d * INT(n/d)$

## Пример 1

Следующая формула возвращает 1, остаток от деления 3 на 2.

```
= MOD(3,2)
```

## Пример 2

Следующая формула возвращает -1, остаток от деления 3 на 2. Обратите внимание, что знак всегда совпадает с знаком делителя.

```
= MOD(-3,-2)
```

## См. также:

[Математические и тригонометрические функции](#)

[Функция ROUND](#)

[ROUNDUP, функция](#)

[Функция ROUNDDOWN](#)

[MROUND, функция](#)



# MROUND

02.10.2020 • 2 minutes to read

Возвращает число, округленное до нужного кратного.

## Синтаксис

```
MROUND(<number>, <multiple>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
число	Число для округления.
multiple	Кратность, до которой требуется округление числа.

## Возвращаемое значение

Десятичное число.

## Примечания

MROUND округляет в большую сторону от нуля, если остаток от деления **number** на указанный аргумент **multiple** больше или равен половине значения **multiple**.

## Пример: Число десятичных знаков

Следующее выражение округляет число 1,3 до ближайшего числа, кратного 0,2. Ожидаемый результат: 1,4.

```
= MROUND(1.3,0.2)
```

## Пример: Отрицательные числа

Следующее выражение округляет число -10 до ближайшего числа, кратного -3. Ожидаемый результат: -9.

```
= MROUND(-10,-3)
```

## Пример: Ошибка

Следующее выражение возвращает ошибку, так как числа имеют разные знаки.

```
= MROUND(5,-2)
```

## См. также

Математические и тригонометрические функции

Функция ROUND

ROUNDUP, функция

Функция ROUNDDOWN

MROUND, функция

Функция INT

# ODD

02.10.2020 • 2 minutes to read

Возвращает число, округленное до ближайшего нечетного целого.

## Синтаксис

```
ODD(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Обязательный элемент. Округляемое значение.

## Возвращаемое значение

Возвращает число, округленное до ближайшего нечетного целого.

## Remarks

- Если значение параметра number является нечисловым, ODD возвращает #VALUE! (значение ошибки).
- Независимо от знака числа, значение округляется в большую сторону, если оно корректируется от нуля. Если number является нечетным целым числом, округление не выполняется.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= ODD(1,5)	Округляет число 1,5 до ближайшего нечетного целого числа.	3
= ODD(3)	Округляет число 3 до ближайшего нечетного целого числа.	3
= ODD(2)	Округляет число 2 до ближайшего нечетного целого числа.	3
= ODD(-1)	Округляет число -1 до ближайшего нечетного целого числа.	-1
= ODD(-2)	Округляет число -2 (от нуля ) до ближайшего нечетного целого числа.	-3



# PI

02.10.2020 • 2 minutes to read

Возвращает число Пи (3,14159265358979) с точностью до 15 знаков.

## Синтаксис

```
PI()
```

## Возвращаемое значение

Десятичное число Пи (3,14159265358979) с точностью до 15 знаков.

## Remarks

Пи — это математическая константа. В DAX число Пи представляется как вещественное число, которое точно соответствует 15 цифрам, так же, как и в Excel.

## Пример

Следующая формула вычисляет площадь круга с учетом радиуса в столбце `[Radius]`.

```
= PI()*([Radius]^2)
```

## См. также раздел

[Математические и тригонометрические функции](#)

# POWER

02.10.2020 • 2 minutes to read

Возвращает результат возведения числа в степень.

## Синтаксис

```
POWER(<number>, <power>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Базовое число, которое может быть любым вещественным числом.
power	Показатель степени, в которую нужно возвести число.

## Возвращаемое значение

Десятичное число.

## Пример

Следующий пример вернет 25.

```
= POWER(5,2)
```

## См. также раздел

[Математические и тригонометрические функции](#)

# ПРОДУКТ

02.10.2020 • 2 minutes to read

Возвращает произведение чисел в столбце.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий числа, для которых вычисляется произведение.

## Возвращаемое значение

Десятичное число.

## Remarks

- Чтобы вернуть произведение выражения, вычисленного для каждой строки в таблице, используйте функцию [PRODUCTX](#).
- Учитываются только числа в столбце. Пустые значения, логические значения и текст игнорируются. Например, примененная к объекту директива

```
PRODUCT( Table[Column] ) равно 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

В следующем примере вычисляется произведение столбца AdjustedRates в таблице Annuity:

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

## См. также

[PRODUCTX](#)

# PRODUCTX

02.10.2020 • 2 minutes to read

Возвращает произведение значений выражения, вычисляемого для каждой строки в таблице.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
таблица	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.

## Возвращаемое значение

Десятичное число.

## Remarks

- Чтобы вернуть произведение чисел в столбце, используйте функцию [PRODUCT](#).
- Функция PRODUCTX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Вторым аргументом является столбец, содержащий числа, для которых вычисляется произведение, или выражение, результатом которого является столбец.
- Учитываются только числа в столбце. Пустые значения, 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 следующем примере вычисляется будущее значение инвестиции:

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

## См. также раздел

[PRODUCT](#)

# QUOTIENT

02.10.2020 • 2 minutes to read

Выполняет деление и возвращает только целую часть результата деления. Используйте эту функцию, если требуется удалить оставшуюся часть деления.

## Синтаксис

```
QUOTIENT(<numerator>, <denominator>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
numerator	Делимое или число, которое нужно разделить.
denominator	Делитель или число, на которое нужно разделить исходное число.

## Возвращаемое значение

Целое число.

## Remarks

- Если любой из аргументов не является числовым, то функция QUOTIENT возвращает значение #VALUE! (значение ошибки).
- Вместо литерального значения для любого из аргументов можно использовать ссылку на столбец. Однако если столбец, на который указывает ссылка, содержит 0 (ноль), то для всего столбца значений возвращается ошибка.

## Пример

Следующие формулы возвращают один и тот же результат — 2.

```
= QUOTIENT(5,2)
```

```
= QUOTIENT(10/2,2)
```

## См. также раздел

[Математические и тригонометрические функции](#)

# RADIANS

02.10.2020 • 2 minutes to read

Преобразует градусы в радианы.

## Синтаксис

```
RADIANS(angle)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
angle	Обязательный элемент. Угол в градусах, который нужно преобразовать.

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= RADIANS(270)	270 градусов в радианах (4,712389 или $3\pi/2$ радиан)	4,712389

# RAND

02.10.2020 • 2 minutes to read

Возвращает случайное число не меньше 0 и меньше 1 с равномерным распределением. Возвращаемое число изменяется каждый раз при повторном вычислении ячейки, содержащей эту функцию.

## Синтаксис

```
RAND()
```

## Возвращаемое значение

Десятичное число.

## Remarks

- Повторное вычисление зависит от различных факторов, в том числе от того, настроен ли для модели **ручной** или **автоматический** режим повторного вычисления и были ли обновлены данные.
- Функция RAND и другие переменные функции, не имеющие фиксированных значений, не всегда пересчитываются. Например, выполнение запроса или фильтрация обычно не приведет к повторной оценке таких функций. Однако результаты этих функций будут пересчитаны при повторном вычислении всего столбца. Эти ситуации включают обновление из внешнего источника данных или ручное редактирование данных, что приводит к повторному вычислению формул, содержащих эти функции.
- RAND всегда пересчитывается, если функция используется в определении меры.
- Функция RAND не может возвращать нулевой результат, чтобы предотвратить ошибку деления на ноль.

## Примеры

Чтобы сгенерировать случайное число между двумя другими числами, используйте следующее:

```
= RAND()*(b-a)+a
```

Создание случайного числа больше 0 и меньше 1:

```
= RAND()
```

Создание случайного числа больше 0 и меньше 100

```
= RAND()*100
```

Создание случайного целого числа больше 0 и меньше 100

```
INT(RAND()*100)
```

См. также раздел

[Математические и тригонометрические функции](#)

[Статистические функции](#)



# RANDBETWEEN

02.10.2020 • 2 minutes to read

Возвращает случайное число в диапазоне между двумя указанными числами.

## Синтаксис

```
RANDBETWEEN(<bottom>,<top>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Нижнее	Наименьшее целое число, возвращаемое функцией.
TOP	Наибольшее целое число, возвращаемое функцией.

## Возвращаемое значение

Целое число.

## Remarks

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.

```
= RANDBETWEEN(1,10)
```

## См. также

[Математические и тригонометрические функции](#)

[Статистические функции](#)

# ROUND

02.10.2020 • 2 minutes to read

Округляет число до указанного количества десятичных разрядов.

## Синтаксис

```
ROUND(<number>, <num_digits>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Число, которое требуется округлить.
num_digits	Число знаков, до которого требуется округлить число. Отрицательное значение округляет цифры слева от десятичной запятой; нулевое значение округляет до ближайшего целого числа.

## Возвращаемое значение

Десятичное число.

## Remarks

- Если значение параметра num\_digits больше 0, то число округляется до указанного числа десятичных разрядов.
- Если значение параметра num\_digits равно 0, то число округляется до ближайшего целого числа.
- Если значение параметра num\_digits меньше 0, то число округляется слева от десятичной запятой.
- Связанные функции
  - Чтобы всегда округлять в большую сторону (от нуля), используйте функцию ROUNDUP.
  - Чтобы всегда округлять в меньшую сторону (в сторону нуля), используйте функцию ROUNDDOWN.
  - Чтобы округлить число до определенного кратного (например, для округления до ближайшего числа, кратного 0,5), используйте функцию MROUND.
  - Для получения целой части числа можно использовать функции TRUNC и INT.

## Пример 1

Следующая формула округляет число 2,15 до одного десятичного знака. Ожидаемый результат — 2,2.

```
= ROUND(2.15,1)
```

## Пример 2

Следующая формула округляет число 21,5 до одного десятичного знака слева от десятичной запятой.  
Ожидаемый результат: 20.

```
= ROUND(21.5, -1)
```

## См. также раздел

[Математические и тригонометрические функции](#)

[ROUND](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

[TRUNC](#)

# ROUNDDOWN

02.10.2020 • 2 minutes to read

Округляет число в меньшую сторону, к нулю.

## Синтаксис

```
ROUNDDOWN(<number>, <num_digits>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Округляемое в меньшую сторону вещественное число.
num_digits	Число знаков, до которого требуется округлить число. Отрицательное значение округляет число слева от десятичной запятой; ноль — до ближайшего целого числа.

## Возвращаемое значение

Десятичное число.

## Remarks

- Если **num\_digits** больше 0, то значение в параметре **number** округляется в меньшую сторону до указанного числа десятичных разрядов.
- Если **num\_digits** равен 0, то значение в параметре **number** округляется в меньшую сторону до ближайшего целого числа.
- Если **num\_digits** меньше 0, то значение в параметре **number** округляется в меньшую сторону слева от десятичной запятой.
- Поведение ROUNDDOWN аналогично функции ROUND, за исключением того, что она всегда округляет число в меньшую сторону. Функция INT также округляет в меньшую сторону, но при использовании INT результат всегда является целым числом, тогда как с помощью функции ROUNDDOWN можно управлять точностью результата.

## Пример 1

Следующий пример округляет 3,14159 в меньшую сторону до трех десятичных разрядов. Ожидаемый результат: 3,141.

```
= ROUNDDOWN(3.14159,3)
```

## Пример 2

В следующем примере значение 31415,92654 округляется в меньшую сторону до 2 десятичных знаков слева от запятой. Ожидаемый результат: 31400.

```
= ROUNDDOWN(31415.92654, -2)
```

## См. также

[Математические и тригонометрические функции](#)

[ROUND](#)

[ROUNDUP](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

# ROUNDUP

02.10.2020 • 2 minutes to read

Округляет число в большую сторону, от нуля.

## Синтаксис

```
ROUNDUP(<number>, <num_digits>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
число	Округляемое в большую сторону вещественное число.
num_digits	Число знаков, до которого требуется округлить число. При отрицательном значении <b>num_digits</b> округление выполняется слева от десятичной запятой; если значение <b>num_digits</b> равно нулю или опущено, <b>number</b> округляется до ближайшего целого числа.

## Возвращаемое значение

Десятичное число.

## Примечания

- Если **num\_digits** больше 0, то число округляется до указанного числа десятичных разрядов.
- Если **num\_digits** равен 0, то число округляется до ближайшего целого числа.
- Если **num\_digits** меньше 0, то число округляется слева от десятичной запятой.
- Поведение ROUNDUP аналогично функции ROUND, за исключением того, что она всегда округляет число в большую сторону.

## Пример

Следующая формула округляет число Пи до четырех десятичных знаков. Ожидаемый результат: 3,1416.

```
= ROUNDUP(PI(),4)
```

## Пример: Число десятичных знаков в качестве второго аргумента

Следующая формула округляет число 1,3 до ближайшего числа, кратного 0,2. Ожидаемый результат: 2.

```
= ROUNDUP(1.3,0.2)
```

## Пример: Отрицательное число в качестве второго аргумента

Следующая формула округляет значение в столбце FreightCosts ожидаемыми результатами, приведенными в следующей таблице:

```
= ROUNDUP([Values],-1)
```

Если num\_digits меньше нуля, число знаков слева от десятичного знака увеличивается на указанное значение.

FREIGHTCOST	ОЖИДАЕМЫЙ РЕЗУЛЬТАТ
13,25	20
2.45	10
25,56	30
1,34	10
345,01	350

## См. также

[Математические и тригонометрические функции](#)

[ROUND](#)

[ROUNDDOWN](#)

[MROUND](#)

[INT](#)

# SIGN

02.10.2020 • 2 minutes to read

Определяет знак числа, результат вычисления или значение в столбце. Функция возвращает 1, если число положительное, 0, если число равно нулю, и -1, если число отрицательное.

## Синтаксис

```
SIGN(<number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Любое вещественное число, столбец, содержащий числа, или выражение, результатом которого является число.

## Возвращаемое значение

Целое число. Возможные возвращаемые значения: 1, 0 и -1.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ	DESCRIPTION
1	Число положительное
0	Число равно нулю
-1	Число отрицательно

## Пример

Следующая формула возвращает знак результата выражения, который вычисляет цену продажи за минусом стоимости.

```
= SIGN( ([Sale Price] - [Cost]) )
```

## См. также раздел

[Математические и тригонометрические функции](#)



# SQRT

02.10.2020 • 2 minutes to read

Возвращает квадратный корень числа.

## Синтаксис

```
SQRT(<number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Число, для которого вычисляется квадратный корень, столбец, содержащий числа, или выражение, результатом которого является число.

## Возвращаемое значение

Десятичное число.

## Remarks

Если число отрицательное, функция SQRT возвращает ошибку.

## Пример

Следующая формула:

```
= SQRT(25)
```

## См. также раздел

[Математические и тригонометрические функции](#)

# SUM

22.09.2020 • 2 minutes to read

Добавляет все числа в столбце.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий суммируемые числа.

## Возвращаемое значение

Десятичное число.

## Remarks

Если необходимо отфильтровать суммированные значения, можно использовать функцию SUMX и указать выражение для суммирования.

## Пример

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

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

## См. также раздел

[SUMX](#)

# SUMX

02.10.2020 • 2 minutes to read

Возвращает сумму значений выражения, вычисляемого для каждой строки в таблице.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.

## Возвращаемое значение

Десятичное число.

## Remarks

- Функция SUMX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Вторым аргументом является столбец, содержащий числа, которые необходимо суммировать, или выражение, результатом которого является столбец.
- Учитываются только числа в столбце. Пустые значения, логические значения и текст игнорируются.
- Более сложные примеры SUMX в формулах см. в разделах [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 выполняется фильтрация таблицы, InternetSales, в выражении `ShippingTerritoryID = 5`, а затем возвращается сумма всех значений в столбце Freight. Иными словами, выражение возвращает сумму расходов на транспортировку только для указанной области продаж.

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

Если фильтрация столбца не требуется, используйте функцию SUM. Функция SUM аналогична функции Excel с тем же именем, за исключением того, что в качестве ссылки принимается столбец.

## См. также раздел

[SUM](#)



# TRUNC

02.10.2020 • 2 minutes to read

Усекает число до целого путем удаления десятичной или дробной части.

## Синтаксис

```
TRUNC(<number>,<num_digits>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Число для усечения.
num_digits	Число, указывающее точность усечения; 0 по умолчанию

## Возвращаемое значение

Целое число.

## Remarks

Функции TRUNC и INT похожи в том, что обе возвращают целые числа. TRUNC удаляет дробную часть числа. INT округляет число до ближайшего меньшего целого, исходя из значения дробной части числа. Функции INT и TRUNC различаются только при использовании для отрицательных чисел: `TRUNC(-4.3)` возвращает значение `-4`, но `INT(-4.3)` возвращает `-5`, так как `-5` — это меньшее число.

## Пример 1

Следующая формула возвращает значение 3, целую часть числа Пи.

```
= TRUNC(PI())
```

## Пример 2

Следующая формула возвращает значение `-8`, целую часть числа `-8,9`.

```
= TRUNC(-8.9)
```

## См. также раздел

[Математические и тригонометрические функции](#)

[ROUND](#)

[ROUNDUP](#)

[ROUNDDOWN](#)

[MROUND](#)



# Другие функции

22.09.2020 • 2 minutes to read

Эти функции выполняют уникальные действия, которые нельзя отнести к другим категориям.

## В этой категории

КОМПОНЕНТ	DESCRIPTION
BLANK	Возвращает пустое значение.
ошибка	Вызывает ошибку с сообщением об ошибке.

# BLANK

02.10.2020 • 2 minutes to read

Возвращает пустое значение.

## Синтаксис

```
BLANK()
```

## Возвращаемое значение

Пустое значение.

## Remarks

- Значения BLANK не эквивалентны значениям NULL. DAX использует пустые значения как для значений NULL базы данных, так и для пустых ячеек в Excel.
- Некоторые функции DAX по-разному обрабатывают пустые ячейки в Microsoft Excel. Пустые значения и пустые строки ("" ) не всегда эквивалентны, но некоторые операции могут считать их подобными.

## Пример

В следующем примере показано, как можно работать с пустыми значениями в формулах. Формула вычисляет соотношение продаж через торговых посредников и продаж через Интернет. Однако прежде чем пытаться вычислить соотношение, знаменатель следует проверить на нулевые значения. Если знаменатель равен нулю, должно возвращаться пустое значение. В противном случае вычисляется соотношение.

```
= IF( SUM(InternetSales_USD[SalesAmount_USD])= 0 , BLANK() ,  
SUM(ResellerSales_USD[SalesAmount_USD])/SUM(InternetSales_USD[SalesAmount_USD]) )
```

В таблице показаны ожидаемые результаты, когда эта формула используется для создания сводной таблицы.

СООТНОШЕНИЕ ПРОДАЖ ЧЕРЕЗ ТОРГОВЫХ ПОСРЕДНИКОВ И ПРОДАЖ ЧЕРЕЗ ИНТЕРНЕТ	МЕТКИ СТОЛБЦОВ			
Метки строк	Accessories	Bikes	Clothing	Grand Total
2005		2,65		2,89
2006		3,33		4,03
2007 г.	1,04	2,92	6,63	3.51



СООТНОШЕНИЕ ПРОДАЖ ЧЕРЕЗ ТОРГОВЫХ ПОСРЕДНИКОВ И ПРОДАЖ ЧЕРЕЗ ИНТЕРНЕТ	МЕТКИ СТОЛБЦОВ			
2008	0,41	1,53	2,00	1,71
Grand Total	0,83	2,51	5,45	2,94

В исходном источнике данных столбец, вычисленный с помощью функции BLANK, может содержать текст, пустые строки или значения NULL. Если исходный источник данных был базой данных SQL Server, значения NULL и пустые строки являются разными типами данных. Однако для этой операции выполняется неявное приведение типов, а DAX обрабатывает их как одно и то же.

## См. также раздел

[Текстовые функции](#)

[Функция ISBLANK](#)

# ошибка

02.10.2020 • 2 minutes to read

Вызывает ошибку с сообщением об ошибке.

## Синтаксис

```
ERROR(<text>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text	Текстовая строка, содержащая сообщение об ошибке.

## Возвращаемое значение

None

## Remarks

- Функцию ERROR можно поместить в выражение DAX в любом месте, где ожидается скалярное значение.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример 1

Рассмотрим следующий запрос 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]
```

Он завершается ошибкой и выдает сообщение "red color encountered" (обнаружен красный цвет).

## Пример 2

Рассмотрим следующий запрос 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]

```

Он возвращает следующую таблицу:

DIMPRODUCT[COLOR]	[MEASURE]
Черный	Черный
Синий	Синий
Серый	Серый
Несколько вариантов	Несколько вариантов
Н/Д	Н/Д
Красный	Красный
Серебряная	Серебряная
Silver\Black	Silver\Black
Белый	White
Желтый	Желтый

Так как пурпурный цвет не является одним из цветов продукта, функция ERROR не выполняется.

# Родительские и дочерние функции

22.09.2020 • 2 minutes to read

Эти функции управляют данными, представленными в виде иерархий "родители-потомки".

Дополнительные сведения см. в разделе [Основные сведения о функциях для иерархий "родители-потомки" в DAX](#).

## В этой категории

КОМПОНЕНТ	DESCRIPTION
<a href="#">PATH</a>	Возвращает текстовую строку с разделителями, в которой находятся идентификаторы всех родительских элементов текущего идентификатора.
<a href="#">PATHCONTAINS</a>	Возвращает значение TRUE, если указанный <i>элемент</i> имеется в заданном <i>пути</i> .
<a href="#">PATHITEM</a>	Возвращает элемент в заданном параметре <i>position</i> из строки, полученной при вычислении функции PATH.
<a href="#">PATHITEMREVERSE</a>	Возвращает элемент в заданном параметре <i>position</i> из строки, полученной при вычислении функции PATH.
<a href="#">PATHLENGTH</a>	Возвращает число родительских элементов для указанного элемента в заданном результате PATH, включая сам элемент.

# Основные сведения о функциях для иерархий "родители-потомки" в DAX

02.10.2020 • 6 minutes to read

В DAX есть пять функций, помогающих пользователям управлять данными, представленными иерархией "родители-потомки", в своих моделях. С помощью этих функций пользователь может получить весь список происхождения родителей строки, узнать, сколько уровней содержит цепочка происхождения, идущая к верхнему родителю, понять, какой элемент является родительским на N уровней выше текущей строки или потомком N-ного уровня от текущей строки, либо же определить, является ли заданный родительский элемент иерархии родителем для текущей строки.

## Функции иерархии "родители-потомки" в DAX

Следующая таблица содержит иерархию типа "родители-потомки" для столбцов. Столбцы **EmployeeKey** и **ParentEmployeeKey** используются во всех примерах функций.

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 для сотрудника 112 не определен родительский элемент, у сотрудника 14 сотрудник 112 указан как менеджер (ParentEmployeeKey), у сотрудника 3 в качестве менеджера также указан сотрудник 14, а у сотрудников 11, 13 и 162 в качестве менеджера указан сотрудник 3. Описание выше помогает понять, что у сотрудника 112 нет руководителя, то есть он является руководителем верхнего уровня для всех показанных здесь сотрудников; кроме того, сотрудник 3 подчиняется сотруднику 14 и руководит сотрудниками 11, 13 и 162.

В следующей таблице представлены доступные функции, дано краткое описание каждой функции и приведен пример функции с данными, которые показаны выше.

**Функция PATH** — возвращает текст с разделителями, в котором находятся идентификаторы всех родительских элементов текущей строки, начиная с самого старого или верхнего и заканчивая текущей.

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

**Функция PATHLENGTH** — возвращает количество уровней в заданном пути PATH(), начиная с текущего уровня до самого старого или верхнего родительского уровня. В следующем примере столбец PathLength определен как "`= PATHLENGTH([Path])`"; пример включает все данные из примера Path(), помогающие понять, как работает эта функция.

EMPLOYEEKEY	PARENTEMPLOYEEKEY	ПУТЬ	PATHLENGTH
112		112	1
14	112	112 14	2
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

**Функция PATHITEM DAX** — возвращает элемент в указанной позиции из аналогичного PATH() результата, считая слева направо. В следующем примере столбец PathItem (четвертый слева) определен как "`= PATHITEM([Path], 4)`"; этот пример возвращает EmployeeKey в четвертой позиции в строке Path слева, если использовать примеры данных из примера Path().

EMPLOYEEKEY	PARENTEMPLOYEEKEY	ПУТЬ	PATHITEM — ЧЕТВЕРТЫЙ 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

Функция **PATHITEMREVERSE** — возвращает элемент в позиции *position* из аналогичного PATH() результата, считая в обратном направлении — справа налево.

В следующем примере столбец PathItemReverse (третий справа) определен как "`= PATHITEMREVERSE([Path], 3)`"; этот пример возвращает EmployeeKey в третьей позиции в строке Path справа, если использовать примеры данных из примера Path().

EMPLOYEEKEY	PARENTEMPLOYEEKEY	ПУТЬ	PATHITEMREVERSE — ТРЕТИЙ 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 значение TRUE, если указанный *item* есть в заданном *path*. В следующем примере столбец PathContains — сотрудник 162 определен как "`= PATHCONTAINS([Path], "162")`"; этот пример возвращает TRUE, если данный путь содержит сотрудника 162. В этом примере используются результаты из приведенного выше примера Path().

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



# PATH

22.09.2020 • 4 minutes to read

Возвращает текстовую строку с разделителями, в которой находятся идентификаторы всех родительских элементов текущего идентификатора, начиная с самого старого и заканчивая текущей.

## Синтаксис

```
PATH(<ID_columnName>, <parent_columnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
ID_columnName	Имя существующего столбца, содержащего уникальный идентификатор для строк в таблице. Этот параметр не может быть выражением. Тип данных значения в <i>ID_columnName</i> должен быть текстовым или целым числом, а также иметь тот же тип данных, что и столбец, указанный в параметре <i>parent_columnName</i> .
parent_columnName	Имя существующего столбца, содержащего уникальный идентификатор родительского элемента текущей строки. Этот параметр не может быть выражением. Тип данных значения в <i>parent_columnName</i> должен быть текстовым или целым числом, а также иметь тот же тип данных, что и значение, указанное в параметре <i>ID_columnName</i> .

## Возвращаемое значение

Текстовая строка с разделителями, в которой находятся идентификаторы всех родительских элементов текущего идентификатора.

## Remarks

- Эта функция используется в таблицах, имеющих некоторый тип внутренней иерархии, для возврата элементов, связанных с текущим значением строки. Например, в таблице Employees, которая содержит сотрудников, руководителей сотрудников и руководителей менеджеров, можно вернуть путь, который подключает сотрудника к своему менеджеру.
- Путь не ограничивается одним уровнем связей типа "родители-потомки"; он может возвращать связанные строки, которые находятся на нескольких уровнях выше указанной начальной строки.
  - Для разделения предков используется вертикальная черта "|".
  - Значения в *ID\_columnName* и *parent\_columnName* должны иметь один и тот же тип данных: текст или целое число.
  - Значения в *parent\_columnName* должны присутствовать в *ID\_columnName*. Это значит, что нельзя выполнить поиск родительского элемента, если на дочернем уровне нет значения.
  - Если значение параметра *parent\_columnName* — BLANK, то PATH() возвращает значение *ID\_columnName*. Иными словами, если требуется найти руководителя сотрудника, но в столбце

*parent\_columnName* нет данных, функция PATH возвращает только идентификатор сотрудника.

- Если параметр *ID\_columnName* содержит дубликаты и *parent\_columnName*, одинаковый для этих дубликатов, PATH() возвращает общее значение *parent\_columnName*. Однако если значение *parent\_columnName* отличается для этих дубликатов, то PATH() возвращает ошибку. Иными словами, если у вас есть два вхождения для одного и того же идентификатора сотрудника и они имеют один и тот же идентификатор руководителя, функция PATH возвращает идентификатор такого руководителя. Однако при наличии двух идентичных идентификаторов сотрудников, имеющих разные идентификаторы менеджеров, функция PATH возвращает ошибку.
- Если значение параметра *ID\_columnName* — BLANK, то PATH() возвращает значение BLANK.
- Если *ID\_columnName* содержит вертикальную черту "|", 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, в котором перечисляются все менеджеры для каждого сотрудника.

```
= PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey])
```

# PATHCONTAINS

22.09.2020 • 2 minutes to read

Возвращает значение TRUE, если указанный *item* имеется в заданном *path*.

## Синтаксис

```
PATHCONTAINS(<path>, <item>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
path	Строка, созданная в результате вычисления функции PATH.
item	Искомое текстовое выражение для результата вычисления PATH.

## Возвращаемое значение

Значение TRUE, если *item* существует в *path*; в противном случае возвращается значение FALSE.

## Remarks

- Если *item* является целым числом, оно преобразуется в текст, а затем вычисляется функция. Если преобразование завершается сбоем, 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 указан в списке руководителей, возвращенных функцией PATH, функция PATHCONTAINS возвращает значение TRUE; в противном случае возвращается значение FALSE.

```
= PATHCONTAINS(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), "23")
```

# PATHITEM

02.10.2020 • 2 minutes to read

Возвращает элемент в заданном параметре *position* из строки, полученной при вычислении функции PATH. Позиции подсчитываются слева направо.

## Синтаксис

```
PATHITEM(<path>, <position>[, <type>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
path	Текстовая строка в виде результатов вычисления функции PATH.
position	Целочисленное выражение с позицией возвращаемого элемента.
тип	(необязательно) Перечисление, определяющее тип данных результата:

#### Перечисление типов

ПЕРЕЧИСЛЕНИЕ	АЛЬТЕРНАТИВНОЕ ПЕРЕЧИСЛЕНИЕ	ОПИСАНИЕ
TEXT	0	Результаты возвращаются с текстовым типом данных. (по умолчанию).
ЦЕЛОЕ ЧИСЛО	1	Результаты возвращаются в виде целых чисел.

## Возвращаемое значение

Идентификатор, возвращаемый функцией PATH в указанной позиции в списке идентификаторов. Элементы, возвращаемые функцией PATH, упорядочиваются от наиболее удаленных до текущего.

## Remarks

- Эта функция может использоваться для возврата определенного уровня из иерархии, возвращаемой функцией PATH. Например, можно вернуть только менеджеров для всех сотрудников, которым они не подчинены непосредственно.
- Если указать для *position* число, которое меньше единицы (1) или больше, чем число элементов в параметре *path*, функция PATHITEM возвращает значение BLANK
- Если значение параметра *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 менеджер третьего уровня по отношению к текущему сотруднику. В качестве входных данных для функции PATH принимаются идентификаторы сотрудника и менеджера, которые возвращают строку с иерархией родительских элементов для текущего сотрудника. Из этой строки PATHITEM возвращает третью запись в виде целого числа.

```
= PATHITEM(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

# PATHITEMREVERSE

02.10.2020 • 2 minutes to read

Возвращает элемент в заданном параметре *position* из строки, полученной при вычислении функции PATH. Позиции учитываются в обратном направлении справа налево.

## Синтаксис

```
PATHITEMREVERSE(<path>, <position>[, <type>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
path	Текстовая строка, полученная в результате вычисления функции PATH.
position	Целочисленное выражение с позицией возвращаемого элемента. Позиции учитываются в обратном направлении справа налево.
тип	(необязательно) Перечисление, определяющее тип данных результата:

### Перечисление типов

ПЕРЕЧИСЛЕНИЕ	АЛЬТЕРНАТИВНОЕ ПЕРЕЧИСЛЕНИЕ	ОПИСАНИЕ
TEXT	0	Результаты возвращаются с текстовым типом данных. (по умолчанию).
ЦЕЛОЕ ЧИСЛО	1	Результаты возвращаются в виде целых чисел.

## Возвращаемое значение

N-ное восходящее производное по заданному пути, считая от текущего до самого старого.

## Remarks

- Эта функция может использоваться для получения отдельного элемента иерархии, полученной из функции PATH.
- Эта функция меняет порядок в иерархии на обратный стандарту, чтобы ближайшие элементы были перечислены первыми. Например, если функция PATH возвращает список руководителей, стоящих выше сотрудника по иерархии, функция PATHITEMREVERSE возвращает идентификатор непосредственного руководителя сотрудника в положении 2, так как в положении 1 содержится идентификатор сотрудника.
- Если указать для *position* число меньше единицы (1) или больше числа элементов в параметре *path*,

функция PATHITEM возвращает значение BLANK.

- Если значение параметра *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 столбец идентификатора сотрудника принимается в качестве входных данных для функции PATH и обращается возвращаемый список элементов-предков. Указанная позиция имеет значение 3, а тип возвращаемого значения — 1; таким образом, функция PATHITEMREVERSE возвращает целое число, представляющее руководителя на два уровня выше сотрудника.

```
= PATHITEMREVERSE(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

# PATHLENGTH

22.09.2020 • 2 minutes to read

Возвращает число родительских элементов для указанного элемента в заданном результате PATH, включая сам элемент.

## Синтаксис

```
PATHLENGTH(<path>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
path	Текстовое выражение, полученное в результате вычисления функции PATH.

## Возвращаемое значение

Число родительских элементов для указанного элемента в заданном результате PATH, включая указанный элемент.

## Remarks

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 идентификатор сотрудника принимается в качестве входных данных для функции PATH и возвращает список руководителей выше этого сотрудника в иерархии. Функция PATHLENGTH принимает этот результат и подсчитывает различные уровни сотрудников и руководителей, включая сотрудника, с которого вы начали.

```
= PATHLENGTH(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]))
```



# Функции связей

02.10.2020 • 2 minutes to read

Функции в этой категории предназначены для использования связей между таблицами и управления ими.

## В этой категории

ФУНКЦИЯ	ОПИСАНИЕ
<a href="#">CROSSFILTER</a>	Задаёт направление перекрестной фильтрации, используемое в вычислении для связи между двумя столбцами.
<a href="#">RELATED</a>	Возвращает связанное значение из другой таблицы.
<a href="#">RELATEDTABLE</a>	Вычисляет табличное выражение в контексте, изменённом указанными фильтрами.
<a href="#">USERELATIONSHIP</a>	Указывает используемую в конкретном вычислении связь в соответствии с той, которая существует между columnName1 и columnName2.

# CROSSFILTER

22.09.2020 • 4 minutes to read

Задаёт направление перекрестной фильтрации, используемое в вычислении для связи между двумя столбцами.

## Синтаксис

```
CROSSFILTER(<columnName1>, <columnName2>, <direction>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName1	Полное имя существующего столбца, использующего стандартный синтаксис DAX, которое обычно представляет часть отношения для использования. Если аргументы заданы в обратном порядке, функция будет менять их перед использованием. Этот аргумент не может быть выражением.
columnName2	Полное имя существующего столбца, использующего стандартный синтаксис DAX, которое обычно представляет одну сторону или сторону подстановки используемой связи. Если аргументы заданы в обратном порядке, функция будет менять их перед использованием. Этот аргумент не может быть выражением.
Direction	<p>Направление перекрестной фильтрации, которое будет использоваться. Используйте один из следующих вариантов:</p> <p>Лицензия не требуется</p> <p>С этой связью перекрестная фильтрация не выполняется</p> <p><b>one</b> — фильтры на одной стороне или на стороне подстановки фильтра связей.</p> <p><b>both</b> - Фильтры на обеих сторонах фильтра</p> <p><b>none</b> — с этой связью перекрестная фильтрация не выполняется</p>

## Возвращаемое значение

Функция не возвращает значение; функция задаёт направление перекрестной фильтрации для указанной связи на время выполнения запроса.

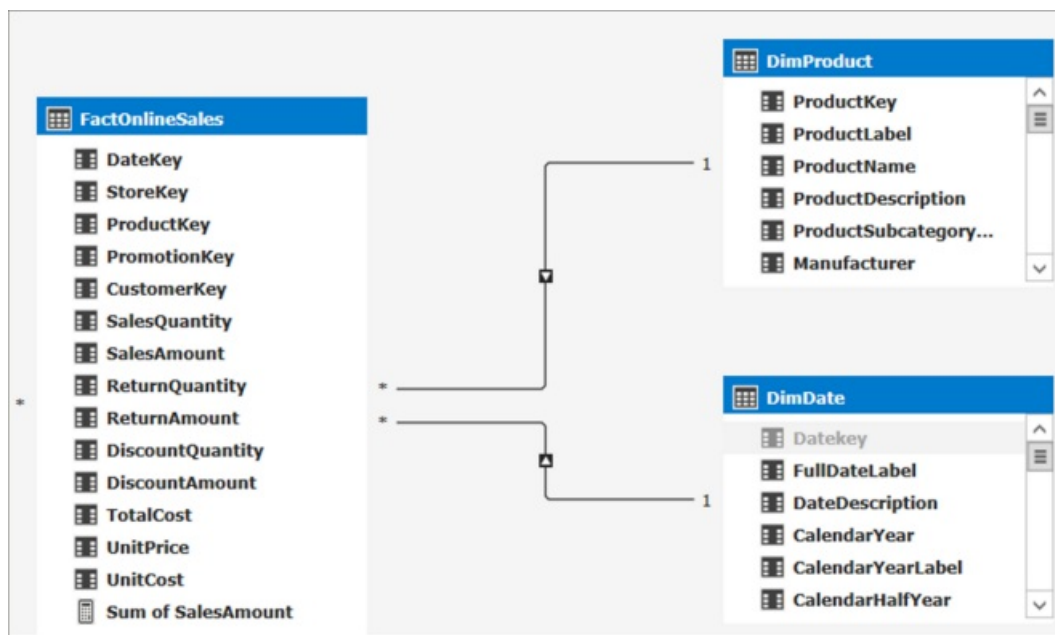
## Примечания

- В случае со связью 1:1 не существует разницы между направлением one и both.

- CROSSFILTER можно использовать только в функциях, принимающих фильтр в качестве аргумента, например: CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD и TOTALYTD.
- CROSSFILTER использует существующие связи в модели, определяя связи по столбцам конечной точки.
- В CROSSFILTER параметр перекрестной фильтрации связи неважен. То есть независимо от того, настроена ли связь для фильтрации в одном или обоих направлениях в модели, это не влияет на использование функции. CROSSFILTER переопределит все существующие настройки перекрестной фильтрации.
- Если какой-либо из столбцов с именем в качестве аргумента не является частью связи или если аргументы принадлежат разным связям, возвращается ошибка.
- Если выражения CALCULATE являются вложенными и несколько выражений CALCULATE содержат функцию CROSSFILTER, то функцией CROSSFILTER с самым глубоким уровнем вложения является та, которая имеет приоритет 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 рисунке модели ниже DimProduct и DimDate имеют единое направление связи с FactOnlineSales.



По умолчанию получить число продуктов, проданных по годам, не удастся:

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
<b>Grand Total</b>	<b>\$29,358,677.22</b>	<b>606</b>

Количество продуктов по годам можно получить двумя способами:

- Включите двустороннюю перекрестную фильтрацию в связях. Это изменит поведение фильтров для всех данных между этими двумя таблицами.
- Используйте функцию CROSSFILTER, чтобы изменить способ работы связей только для этой меры.

При использовании DAX можно использовать функцию CROSSFILTER, чтобы изменить принцип работы направления перекрестной фильтрации между двумя столбцами, определенными связью. В этом случае выражение DAX выглядит следующим образом:

```
BiDi:= CALCULATE([Distinct Count of ProductKey], CROSSFILTER(FactInternetSales[ProductKey],
DimProduct[ProductKey] , Both))
```

С помощью функции CROSSFILTER в этом 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	
<b>Grand Total</b>	<b>\$29,358,677.22</b>	<b>606</b>	<b>606</b>

# RELATED

15.12.2020 • 5 minutes to read

Возвращает связанное значение из другой таблицы.

## Синтаксис

```
RELATED(<column>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
гистограмма	Столбец, содержащий значения, которые необходимо получить.

## Возвращаемое значение

Одиночное значение, связанное с текущей строкой.

## Примечания

- Функция требует RELATED наличия связи между текущей таблицей и таблицей со связанными данными. Необходимо указать столбец, содержащий нужные данные, после чего функция будет использовать существующую связь "многие к одному" для получения значений из указанного столбца в связанной таблице. Если связь не существует, ее необходимо создать.
- Когда функция RELATED выполняет поиск, она проверяет все значения в указанной таблице, независимо от примененных фильтров.
- Функции RELATED требуется контекст строки; поэтому ее можно использовать только в выражении вычисляемого столбца, где текущий контекст строки является однозначным, или как вложенную функцию в выражении, использующем функцию сканирования таблицы. Функция сканирования таблицы, такая как SUMX, возвращает значение текущей строки, а затем сканирует другую таблицу на предмет наличия в ней экземпляров этого значения.
- Функцию RELATED невозможно использовать для выборки столбца в [ограниченной связи](#).

## Пример

В следующем примере для создания отчета о продажах, в котором исключены данные о продажах в США, создается мера "Продажи через Интернет за пределами США". Чтобы создать меру, необходимо отфильтровать таблицу InternetSales\_USD, чтобы исключить все продажи, совершенные в США, в таблице SalesTerritory. США как страна упоминаются в таблице SalesTerritory 5 раз, по одному разу для каждого из следующих регионов: Северо-Запад, Северо-Восток, Центр, Юго-Запад и Юго-Восток.

Первый способ фильтрации продаж через Интернет для создания меры заключается в добавлении выражения фильтра, как указано ниже:

```

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 нелогичен, подвержен ошибкам ввода и может не работать, если в будущем любой из существующих регионов будет разделен.

В этом случае лучше всего использовать существующую связь между InternetSales\_USD и SalesTerritory, а также явно указать, что страна должна отличаться от США. Для этого создайте выражение фильтра, аналогичное следующему:

```

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 поиска используется функцией фильтра, чтобы определить, отфильтрована ли строка InternetSales\_USD.

#### NOTE

Если этот пример не работает, может потребоваться создать связь между таблицами.

```

= 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 мере ("Продажи через Интернет за пределами США") работает должным образом.

МЕТКИ СТРОК	INTERNET SALES	ПРОДАЖИ ЧЕРЕЗ ИНТЕРНЕТ ЗА ПРЕДЕЛАМИ США
Австралия	4 999 021,84 долл. США	4 999 021,84 долл. США
Канада	1 343 109,10 долл. США	1 343 109,10 долл. США
Франция	2 490 944,57 долл. США	2 490 944,57 долл. США
Германия	2 775 195,60 долл. США	2 775 195,60 долл. США
Соединенное Королевство	5 057 076,55 долл. США	5 057 076,55 долл. США
США	9 389 479,79 долл. США	
Grand Total	26 054 827,45 долл. США	16 665 347,67 долл. США

В следующей таблице показан окончательный отчет, который может быть получен при использовании этой меры в сводной таблице:

ПРОДАЖИ ЧЕРЕЗ ИНТЕРНЕТ ЗА ПРЕДЕЛАМИ США	МЕТКИ СТОЛБЦОВ			
Метки строк	Принадлежности	Велосипеды	Экипировка	Общий итог
2005		1 526 481,95 долл. США		1 526 481,95 долл. США
2006		3 554 744,04 долл. США		3 554 744,04 долл. США
2007 г.	156 480,18 долл. США	5 640 106,05 долл. США	70 142,77 долл. США	5 866 729,00 долл. США
2008	228 159,45 долл. США	5 386 558,19 долл. США	102 675,04 долл. США	5 717 392,68 долл. США
Grand Total	384 639,63 долл. США	16 107 890,23 долл. США	172 817,81 долл. США	16 665 347,67 долл. США

См. также

[RELATEDTABLE](#)

[Функции фильтрации](#)

# RELATEDTABLE

02.10.2020 • 2 minutes to read

Вычисляет табличное выражение в контексте, измененном указанными фильтрами.

## Синтаксис

```
RELATEDTABLE(<tableName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
tableName	Имя существующей таблицы с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.

## Возвращаемое значение

Таблица значений.

## Remarks

- Функция RELATEDTETABLE изменяет контекст, в котором выполняется фильтрация данных, и вычисляет выражение в новом контексте, который вы задаете.
- 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 продаж через Интернет в таблице категорий продуктов.

Результаты показаны в следующей таблице.

КЛЮЧ КАТЕГОРИИ ПРОДУКТА	АЛЬТЕРНАТИВНЫЙ КЛЮЧ КАТЕГОРИИ ПРОДУКТОВ	ИМЯ КАТЕГОРИИ ПРОДУКТА	INTERNET SALES
1	1	Bikes	28 318 144,65 долл. США
2	2	Components	
3	3	Clothing	339 772,61 долл. США
4	4	Accessories	700 759,96 долл. США



```
= SUMX( RELATEDTABLE('InternetSales_USD')  
      , [SalesAmount_USD])
```

См. также раздел

[CALCULATETABLE](#)

[Функции фильтрации](#)

# USERELATIONSHIP

02.10.2020 • 4 minutes to read

Указывает используемую в конкретном вычислении связь в соответствии с той, которая существует между columnName1 и columnName2.

## Синтаксис

```
USERELATIONSHIP(<columnName1>,<columnName2>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName1	Полное имя существующего столбца, использующего стандартный синтаксис DAX, которое обычно представляет часть отношения для использования. Если аргументы заданы в обратном порядке, функция будет менять их перед использованием. Этот аргумент не может быть выражением.
columnName2	Полное имя существующего столбца, использующего стандартный синтаксис DAX, которое обычно представляет одну сторону или сторону подстановки используемой связи. Если аргументы заданы в обратном порядке, функция будет менять их перед использованием. Этот аргумент не может быть выражением.

## Возвращаемое значение

Функция не возвращает значение; функция лишь задает указанную связь на время выполнения запроса.

## Примечания

- USERELATIONSHIP можно использовать только в функциях, принимающих фильтр в качестве аргумента, например: CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD и TOTALYTD.
- USERELATIONSHIP нельзя использовать, если для таблицы, в которую включена мера, задана безопасность на уровне строк. Например,

```
CALCULATE(SUM([SalesAmount]), USERELATIONSHIP(FactInternetSales[CustomerKey], DimCustomer[CustomerKey]))
```

будет возвращать ошибку, если для DimCustomer задана безопасность на уровне строк.
- USERELATIONSHIP использует существующие связи в модели, определяя связи по столбцам конечной точки.
- В USERELATIONSHIP состояние связи (является ли связь активной или нет) не имеет значения, то есть не влияет на использование функции. Даже если связь неактивна, она будет использоваться и переопределит любые другие активные связи, которые могут присутствовать в модели, но не

упоминались в аргументах функции.

- Если какой-либо из столбцов с именем в качестве аргумента не является частью связи или если аргументы принадлежат разным связям, возвращается ошибка.
- Если для соединения таблицы A с таблицей B в вычислении требуется несколько связей, каждая связь должна быть указана в другой функции USERELATIONSHIP.
- Если выражения CALCULATE являются вложенными и несколько выражений CALCULATE содержат функцию USERELATIONSHIP, то функцией USERELATIONSHIP с самым глубоким уровнем вложения является та, которая имеет приоритет в случае конфликта или неоднозначности.
- До 10 функций USERELATIONSHIP могут быть вложенными; однако выражение может иметь и более глубокий уровень вложенности. Так, приведенный ниже образец выражения имеет три уровня вложенности, но только два уровня с точки зрения USERELATIONSHIP:

```
=CALCULATE(CALCULATE( CALCULATE( &lt;anyExpression&gt;;, USERELATIONSHIP( t1[colA], t2[colB])), t99[colZ]=999), USERELATIONSHIP( t1[colA], t2[colA]))
```

## Пример

В следующем примере показано, как переопределить активную связь по умолчанию между таблицами InternetSales и DateTime. Связь по умолчанию существует между столбцом OrderDate в таблице InternetSales и столбцом Date в таблице DateTime.

Чтобы вычислить сумму продаж через Интернет и сделать возможным создание срезов с помощью ShippingDate вместо традиционной OrderDate, создайте меру [InternetSales by ShippingDate], используя следующее выражение:

```
= CALCULATE(SUM(InternetSales[SalesAmount]), USERELATIONSHIP(InternetSales[ShippingDate], DateTime[Date]))
```

Необходимо, чтобы между InternetSales[ShipmentDate] и DateTime[Date] существовали связи и эти связи не были активными. Кроме того, должна существовать активная связь между InternetSales[OrderDate] и DateTime[Date].

# Статистические функции

21.09.2020 • 6 minutes to read

В языке выражений анализа данных (DAX) имеется множество функций для создания агрегатов, таких как суммы, количества и средние значения. Эти функции очень похожи на статистические функции, используемые в Microsoft Excel. В данном разделе перечислены статистические функции, представленные в DAX.

## В этой категории

ФУНКЦИЯ	ОПИСАНИЕ
<a href="#">APPROXIMATEDISTINCTCOUNT</a>	Возвращает <i>приблизительное</i> число строк, содержащих различные значения в столбце.
<a href="#">AVERAGE</a>	Возвращает среднее арифметическое всех чисел в столбце.
<a href="#">AVERAGEA</a>	Возвращает среднее арифметическое всех значений в столбце.
<a href="#">AVERAGEX</a>	Возвращает среднее арифметическое набора выражений, вычисленных для таблицы.
<a href="#">BETA.DIST</a>	Возвращает бета-распределение.
<a href="#">BETA.INV</a>	Возвращает обратное значение бета-версии интегральной функции плотности вероятности (BETA.DIST).
<a href="#">CHISQ.DIST</a>	Возвращает распределение хи-квадрат.
<a href="#">CHISQ.DISTRT</a>	Возвращает правостороннюю вероятность распределения хи-квадрат.
<a href="#">CHISQ.INV</a>	Возвращает обратную левостороннюю вероятность распределения хи-квадрат.
<a href="#">CHISQ.INV.RT</a>	Возвращает обратную правостороннюю вероятность распределения хи-квадрат.
<a href="#">CONFIDENCE.NORM</a>	Доверительный интервал — это диапазон значений.
<a href="#">CONFIDENCE.T</a>	Возвращает доверительный интервал для среднего значения совокупности с использованием распределения Стьюдента.
<a href="#">COT</a>	Возвращает котангенс угла, указанного в радианах.
<a href="#">COTH</a>	Возвращает гиперболический котангенс гиперболического угла.

ФУНКЦИЯ	ОПИСАНИЕ
COUNT	Подсчитывает количество ячеек в столбце, которые содержат числа.
COUNTA	Подсчитывает количество непустых ячеек в столбце.
COUNTAX	Подсчитывает непустые результаты при вычислении результата выражения для таблицы.
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, результатом вычисления которого является число, при вычислении выражения для таблицы.
DATATABLE	Предоставляет механизм для объявления встроенного набора значений данных.
DISTINCTCOUNT	Подсчитывает количество уникальных значений в столбце.
DISTINCTCOUNTNOBLANK	Подсчитывает количество уникальных значений в столбце.
EXPON.DIST	Возвращает экспоненциальное распределение.
GEOMEAN	Возвращает среднее геометрическое чисел в столбце.
GEOMEANX	Возвращает среднее геометрическое выражения, вычисляемого для каждой строки в таблице.
MAX	Возвращает максимальное числовое значение в столбце или между двумя скалярными выражениями.
MAXA	Возвращает максимальное значение в столбце.
MAXX	Вычисляет выражение для каждой строки таблицы и возвращает максимальное числовое значение.
MEDIAN	Возвращает медиану чисел в столбце.
MEDIANX	Возвращает медианное значение выражения, вычисляемого для каждой строки в таблице.
MIN	Возвращает минимальное 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.

ФУНКЦИЯ	ОПИСАНИЕ
MINX	Возвращает наименьшее числовое значение, полученное в результате вычисления выражения для каждой строки таблицы.
NORM.DIST	Возвращает нормальное распределение для указанного среднего значения и стандартного отклонения.
NORM.INV	Обратная функция нормального интегрального распределения для указанного среднего значения и стандартного отклонения.
NORM.S.DIST	Возвращает стандартное нормальное распределение (среднее значение равно нулю, а стандартное отклонение — единице).
NORM.S.INV	Возвращает обратное значение стандартного нормального интегрального распределения.
PERCENTILE.EXC	Возвращает k-й процентиль значений в диапазоне, где k находится в диапазоне 0...1, исключая крайние значения.
PERCENTILE.INC	Возвращает k-й процентиль значений в диапазоне, где k находится в диапазоне 0..1, включая крайние значения.
PERCENTILEX.EXC	Возвращает число для процентиля выражения, вычисляемого для каждой строки в таблице.
PERCENTILEX.INC	Возвращает число для процентиля выражения, вычисляемого для каждой строки в таблице.
POISSON.DIST	Возвращает распределение Пуассона.
RANK.EQ	Возвращает ранжирование числа в списке чисел.
RANKX	Возвращает ранжирование числа в списке чисел для каждой строки в аргументе <i>table</i> .
SAMPLE	Возвращает выборку из N строк из указанной таблицы.
SIN	Возвращает синус заданного угла.
SINH	Возвращает гиперболический синус числа.
STDEV.P	Возвращает среднеквадратичное отклонение всей совокупности.
STDEV.S	Возвращает среднеквадратичное отклонение выборки из совокупности.
STDEVX.P	Возвращает среднеквадратичное отклонение всей совокупности.

ФУНКЦИЯ	ОПИСАНИЕ
STDEVX.S	Возвращает среднеквадратичное отклонение выборки из совокупности.
SQRTPI	Возвращает квадратный корень (число * Пи).
T.DIST	Возвращает левостороннее t-распределение Стьюдента.
T.DIST.2T	Возвращает двустороннее t-распределение Стьюдента.
T.DIST.RT	Возвращает правостороннее t-распределение Стьюдента.
T.INV	Возвращает левостороннее обратное t-распределение Стьюдента.
T.INV.2t	Возвращает двустороннее обратное t-распределение Стьюдента.
TAN	Возвращает тангенс заданного угла.
TANH	Возвращает гиперболический тангенс числа.
VAR.P	Возвращает дисперсию всей совокупности.
VAR.S	Возвращает дисперсию выборки из совокупности.
VARX.P	Возвращает дисперсию всей совокупности.
VARX.S	Возвращает дисперсию выборки из совокупности.

# APPROXIMATEDISTINCTCOUNT

22.09.2020 • 2 minutes to read

Возвращает *приблизительное* число строк, содержащих различные значения в столбце. Эта функция может запрашивать большие объемы данных с большей производительностью, чем DISTINCTCOUNT, с небольшими отклонениями от точных результатов.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПИСАНИЕ
гистограмма	Столбец, содержащий значения для подсчета. Этот параметр не может быть выражением.

## Возвращаемое значение

Приблизительное количество уникальных значений в *column*.

## Примечания

Единственным аргументом для этой функции является *column*. Можно использовать столбцы, содержащие данные любого типа. Если функция не находит строки для подсчета, она возвращает значение BLANK, в противном случае возвращается число уникальных значений.



# "AVERAGE" (Среднее)

22.09.2020 • 3 minutes to read

Возвращает среднее арифметическое всех чисел в столбце.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий числа, для которых вычисляется среднее значение.

## Возвращаемое значение

Возвращает десятичное число, представляющее среднее арифметическое чисел в столбце.

## Remarks

- Эта функция принимает указанный столбец в качестве аргумента и находит среднее значение в этом столбце. Если необходимо найти среднее арифметическое выражения, результатом которого является набор чисел, используйте функцию AVERAGEX.
- Нечисловые значения в столбце обрабатываются следующим образом.
  - Если столбец содержит текст, статистическая обработка невозможна, а функции возвращают пустые значения.
  - Если столбец содержит логические значения или пустые ячейки, эти значения игнорируются.
  - Ячейки с нулевым значением учитываются.
- При среднем значении ячеек необходимо помнить о разнице между пустой ячейкой и ячейкой, содержащей значение 0. Если ячейка содержит значение 0, она добавляется к сумме чисел, а строка подсчитывается вместе с числом строк, используемым в качестве делителя. Однако если ячейка содержит пустое значение, строка не учитывается.
- Если нет строк для статистической обработки, функция возвращает пустое значение. Однако если имеются строки, но ни одна из них не соответствует указанным критериям, функция возвращает 0. Excel также возвращает ноль, если не найдены строки, 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.

```
= 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 арифметическое вычисленных значений.

Функция AVERAGEA принимает в качестве аргумента столбец, однако в противном случае поведение похоже на поведение функции СРЗНАЧА в Excel. С помощью функции AVERAGEA можно вычислить среднее значение для столбца, содержащего пустые значения.

# AVERAGEA

02.10.2020 • 2 minutes to read

Возвращает среднее арифметическое всех значений в столбце. Обрабатывает текстовые и нечисловые значения.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий значения, для которых вычисляется среднее.

## Возвращаемое значение

Десятичное число.

## Remarks

- Функция AVERAGEA принимает столбец и усредняет числа в нем, но также обрабатывает нечисловые типы данных в соответствии со следующими правилами:
  - Значения, результат вычисления которых TRUE, учитываются как 1.
  - Значения, результат вычисления которых FALSE, учитываются как 0 (ноль).
  - Значения, которые содержат нечисловое количество текста, учитываются как 0 (ноль).
  - Пустой текст ("") учитывается как 0 (ноль).
- Если вы не хотите включать в вычисление логические значения и текстовые представления чисел в качестве части вычисления, используйте функцию AVERAGE.
- Если нет строк для статистической обработки, функция возвращает пустое значение. Однако если имеются строки, но ни одна из них не соответствует указанным критериям, функция возвращает 0. Microsoft Excel также возвращает ноль, если не найдены строки, соответствующие условиям.
- 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 ссылке столбце, учитывая следующую таблицу. Если вы использовали функцию AVERAGE, то среднее значение будет 21/2; при использовании функции AVERAGEA результат равен 22/5.

TRANSACTION ID	СУММА	РЕЗУЛЬТАТ
0000123	1	Учитывается как 1

TRANSACTION ID	СУММА	РЕЗУЛЬТАТ
0000124	20	Учитывается как 20
0000125	Недоступно	Учитывается как 0
0000126		Учитывается как 0
0000126	TRUE	Учитывается как 1

= AVERAGEA([Amount])

См. также раздел

- Функция AVERAGE
- Функция AVERAGEX
- Статистические функции

# AVERAGEX

02.10.2020 • 2 minutes to read

Возвращает среднее арифметическое набора выражений, вычисленных для таблицы.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Имя таблицы или выражение, указывающее таблицу, для которой можно выполнить агрегацию.
expression	Выражение со скалярным результатом, которое будет вычисляться для каждой строки таблицы в первом аргументе.

## Возвращаемое значение

Десятичное число.

## Remarks

- Функция AVERAGEX позволяет вычислить выражения для каждой строки таблицы, а затем взять результирующий набор значений и вычислить его арифметическое среднее. Таким образом, функция принимает в качестве первого аргумента таблицу и выражение в качестве второго аргумента.
- Во всех остальных отношениях AVERAGEX следует тем же правилам, что и AVERAGE. Нельзя включать нечисловые или пустые ячейки. Аргументы table и expression обязательны.
- Если нет строк для агрегации, функция возвращает пустое значение. Если же строки есть, но ни одна из них не соответствует указанным условиям, функция возвращает 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 среднее значение стоимости доставки и налог для каждого заказа в таблице InternetSales; сначала суммируется стоимость доставки плюс TaxAmt в каждой строке, а затем производится усреднение этих сумм.

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

Если в выражении, используемом в качестве второго аргумента, используется несколько операций, то для управления порядком вычислений необходимо использовать круглые скобки. Дополнительные сведения

см. в [справочнике по синтаксису DAX](#).

## См. также раздел

[Функция AVERAGE](#)

[AVERAGEA, функция](#)

[Статистические функции](#)

# BETA.DIST

22.09.2020 • 2 minutes to read

Возвращает бета-распределение. Бета-распределение обычно используется для изучения вариативности в процентном отношении чего-либо в образцах, например часть дня, которую люди проводят перед телевизором.

## Синтаксис

```
BETA.DIST(x,alpha,beta,cumulative,[A],[B])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
x	Значение в интервале между A и B, для которого вычисляется функция
Коэффициент альфа	Параметр распределения.
Бета-версия	Параметр распределения.
Объект	Необязательный параметр. Нижняя граница интервала X.
B	Необязательный параметр. Верхняя граница интервала X.

## Возвращаемое значение

Возвращает бета-распределение.

## Remarks

- Если аргумент является нечисловым, функция BETA.DIST возвращает #VALUE! (значение ошибки).
- Если какой-то из аргументов не является целым числом, значение округляется.
- Если  $\alpha \leq 0$  или  $\beta \leq 0$ , BETA.DIST возвращает #NUM! (значение ошибки).
- Если  $x < A$ ,  $x > B$  или  $A = B$ , BETA.DIST возвращает #NUM! (значение ошибки).
- Если пропущены значения для A и B, BETA.DIST использует стандартное интегральное бета-распределение, в котором  $A = 0$  и  $B = 1$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

# BETA.INV

22.09.2020 • 2 minutes to read

Возвращает обратное значение бета-версии интегральной функции плотности вероятности (BETA.DIST).

Если вероятность = BETA.DIST(x,...TRUE), то бета-версия. BETA.INV(вероятность,...) = x. Бета-распределение можно использовать в планировании проекта для моделирования вероятного времени выполнения, учитывающего ожидаемое время завершения и вариативность.

## Синтаксис

```
BETA.INV(probability,alpha,beta,[A],[B])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Вероятность	Вероятность, связанная с бета-распределением.
Коэффициент альфа	Параметр распределения.
Бета-версия	Параметр распределения.
Объект	Необязательный параметр. Нижняя граница интервала X.
B	Необязательный параметр. Верхняя граница интервала X.

## Возвращаемое значение

Возвращает обратное значение бета-версии интегральной функции плотности вероятности (BETA.DIST).

## Remarks

- Если аргумент является нечисловым, функция BETA.INV возвращает #VALUE! (значение ошибки).
- Если какой-то из аргументов не является целым числом, значение округляется.
- Если  $\alpha \leq 0$  или  $\beta \leq 0$ , BETA.INV возвращает #NUM! (значение ошибки).
- Если  $\text{probability} \leq 0$  или  $\text{probability} > 1$ , функция BETA.INV возвращает #NUM! (значение ошибки).
- Если пропущены значения для A и B, BETA.INV использует стандартное интегральное бета-распределение, в котором  $A = 0$  и  $B = 1$ .
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).



# CHISQ.DIST

22.09.2020 • 2 minutes to read

Возвращает распределение хи-квадрат.

Распределение хи-квадрат обычно используется для изучения вариативности в процентном отношении чего-либо в образцах, например часть дня, которую люди проводят перед телевизором.

## Синтаксис

```
CHISQ.DIST(<x>, <deg_freedom>, <cumulative>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
x	Значение, по которому вы хотите оценить распределение.
Deg_freedom	Число градусов свободы.
cumulative	Логическое значение, определяющее форму функции. Если cumulative имеет значение TRUE, функция CHISQ.DIST возвращает накопительную функцию распределения, а если FALSE — функцию плотности вероятности.

## Возвращаемое значение

Возвращает распределение хи-квадрат.

## Примечания

- Если x или deg\_freedom имеют нечисловое значение, возвращается ошибка.
- If deg\_freedom не является целым числом, значение округляется.
- Если  $x < 0$ , возвращается ошибка.
- Если  $\text{deg\_freedom} < 1$  или  $\text{deg\_freedom} > 10^{10}$ , возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Рассмотрим следующий запрос DAX:

```
EVALUATE { CHISQ.DIST(2, 2, TRUE) }
```

Возвращает

[ЗНАЧЕНИЕ]
0,632120558828558

# CHISQ.DIST.RT

22.09.2020 • 2 minutes to read

Возвращает правостороннюю вероятность распределения хи-квадрат.

Распределение хи-квадрат связано с критерием хи-квадрат. Используйте критерий хи-квадрат для сравнения наблюдаемых и ожидаемых значений. Например, в ходе генетического эксперимента может проверяться гипотеза о том, что новое поколение растений будет иметь определенный цвет. Сравнивая наблюдаемые результаты с ожидаемыми, можно выяснить, является ли исходная гипотеза верной.

## Синтаксис

```
CHISQ.DIST.RT(<x>, <deg_freedom>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
x	Значение, по которому вы хотите оценить распределение.
Deg_freedom	Число градусов свободы.

## Возвращаемое значение

Правосторонняя вероятность распределения хи-квадрат.

## Примечания

- Если x или deg\_freedom имеют нечисловое значение, возвращается ошибка.
- If deg\_freedom не является целым числом, значение округляется.
- Если  $x < 0$ , возвращается ошибка.
- Если  $\text{deg\_freedom} < 1$  или  $\text{deg\_freedom} > 10^{10}$ , возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Рассмотрим следующий запрос DAX:

```
EVALUATE { CHISQ.DIST.RT(2, 5) }
```

Возвращает

[ЗНАЧЕНИЕ]
0,84914503608461

# CHISQ.INV

02.10.2020 • 2 minutes to read

Возвращает обратную левостороннюю вероятность распределения хи-квадрат.

Распределение хи-квадрат обычно используется для изучения вариативности в процентном отношении чего-либо в образцах, например часть дня, которую люди проводят перед телевизором.

## Синтаксис

```
CHISQ.INV(probability,deg_freedom)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Вероятность	Вероятность, связанная с распределением хи-квадрат.
Deg_freedom	Число градусов свободы.

## Возвращаемое значение

Возвращает обратную левостороннюю вероятность распределения хи-квадрат.

## Remarks

- Если аргумент является нечисловым, функция CHISQ.INV возвращает #VALUE! (значение ошибки).
- Если вероятность < 0 or probability > 1, функция CHISQ.INV возвращает #NUM! (значение ошибки).
- If deg\_freedom не является целым числом, значение округляется.
- Если deg\_freedom < 0 or deg\_freedom > 10^10, функция CHISQ.INV. RT возвращает #NUM! (значение ошибки).
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= CHISQ.INV(0,93.1)	Обратная левостороннюю вероятность распределения хи-квадрат для 0,93 с использованием значения свободы в 1 градус.	5,318520074
= CHISQ.INV(0,6.2)	Обратная левостороннюю вероятность распределения хи-квадрат для 0,6 с использованием значения свободы в 2 градуса.	1,832581464



# CHISQ.INV.RT

22.09.2020 • 2 minutes to read

Возвращает обратную правостороннюю вероятность распределения хи-квадрат.

Если вероятность = CHISQ.DIST.RT( $x, \dots$ ), то CHISQ.INV.RT(probability, ...) =  $x$ . Используйте эту функцию для сравнения наблюдаемых результатов с ожидаемыми, чтобы решить, является ли исходная гипотеза допустимой.

## Синтаксис

```
CHISQ.INV.RT(probability, deg_freedom)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Вероятность	Вероятность, связанная с распределением хи-квадрат.
Deg_freedom	Число градусов свободы.

## Возвращаемое значение

Возвращает обратную правостороннюю вероятность распределения хи-квадрат.

## Remarks

- Если любой из аргументов является нечисловым, функция CHISQ.INV.RT #VALUE! (значение ошибки).
- Если probability < 0 или probability > 1, функция CHISQ.INV.RT возвращает #NUM! (значение ошибки).
- If deg\_freedom не является целым числом, значение округляется.
- Если deg\_freedom < 1, функция CHISQ.INV.RT возвращает #NUM! (значение ошибки).
- Учитывая значение вероятности, функция CHISQ.INV.RT ищет такое значение  $x$ , чтобы CHISQ.DIST.RT( $x$ , deg\_freedom) = probability. Таким образом, точность CHISQ.INV.RT зависит от точности функции CHISQ.DIST.RT. Функция CHISQ.INV.RT использует метод итеративного поиска. Если поиск не сходится после 64 итераций, 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).

# CONFIDENCE.NORM

02.10.2020 • 3 minutes to read

Доверительный интервал — это диапазон значений. Среднее значение выборки,  $\bar{x}$ , находится в центре этого диапазона, а сам диапазон — это  $\bar{x} \pm \text{CONFIDENCE.NORM}$ . Например, если  $\bar{x}$  — среднее время доставки для заказов по почте в выборке,  $\bar{x} \pm \text{CONFIDENCE.NORM}$  является диапазоном средних значений генеральной совокупности. Для любого среднего значения генеральной совокупности,  $\mu_0$ , в этом диапазоне вероятность получения среднего значения выборки тем дальше от  $\mu_0$ , чем  $\bar{x}$  больше  $\alpha$ ; для любого среднего значения генеральной совокупности,  $\mu_0$ , вне этого диапазона вероятность получения среднего значения выборки тем дальше от  $\mu_0$ , чем  $\bar{x}$  меньше  $\alpha$ . Другими словами, предположим, что мы используем  $\bar{x}$ ,  $\text{standard\_dev}$  и  $\text{size}$  для создания двустороннего теста на уровне значимости  $\alpha$  с гипотезой, что среднее значение генеральной совокупности равно  $\mu_0$ . Тогда мы не будем отклонять эту гипотезу, если  $\mu_0$  находится в пределах доверительного интервала, и отклоним эту гипотезу, если  $\mu_0$  не находится в доверительном интервале. Доверительный интервал не позволяет нам сделать вывод, что существует вероятность  $1 - \alpha$  того, что доставка очередной посылки займет время в пределах доверительного интервала.

## Синтаксис

```
CONFIDENCE.NORM(alpha, standard_dev, size)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
alpha	Уровень значимости, используемый для вычисления уровня достоверности. Уровень достоверности равен $100 \cdot (1 - \alpha) \%$ ; другими словами, значение $\alpha$ , равное 0,05, соответствует уровню достоверности в 95 процентов.
standard_dev	Стандартное отклонение популяции для диапазона данных, полагаемое известным.
standard_dev,size	Размер выборки.

## Возвращаемое значение

Диапазон значений

## Remarks

- Если аргумент является нечисловым, функция CONFIDENCE.NORM возвращает #VALUE! (значение ошибки).
- Если  $\alpha \leq 0$  или  $\alpha \geq 1$ , то CONFIDENCE.NORM возвращает #NUM! (значение ошибки).
- Если  $\text{standard\_dev} \leq 0$ , то CONFIDENCE.NORM возвращает #NUM! (значение ошибки).
- Если  $\text{size}$  не является целым числом, значение округляется.



- Если  $size < 1$ , то CONFIDENCE.NORM возвращает #NUM! (значение ошибки).
- Если предполагается, что  $\alpha$  равняется 0,05, необходимо вычислить область под кривой стандартного нормального распределения, которая равна  $(1 - \alpha)$ , или 95 процентам. Это значение составляет  $\pm 1,96$ . Таким образом, доверительный интервал составит:

$$\overline{x} \pm 1.96 \cdot \frac{\sigma}{\sqrt{n}}$$

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

# CONFIDENCE.T

02.10.2020 • 2 minutes to read

Возвращает доверительный интервал для среднего значения совокупности с использованием распределения Стьюдента.

## Синтаксис

```
CONFIDENCE.T(alpha, standard_dev, size)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
alpha	Уровень значимости, используемый для вычисления уровня достоверности. Уровень достоверности равен $100 \cdot (1 - \alpha) \%$ ; другими словами, значение alpha, равное 0,05, соответствует уровню достоверности в 95 процентов.
standard_dev	Стандартное отклонение популяции для диапазона данных, полагаемое известным.
size	Размер выборки.

## Возвращаемое значение

Возвращает доверительный интервал для среднего значения совокупности с использованием распределения Стьюдента.

## Remarks

- Если аргумент является нечисловым, функция CONFIDENCE.T возвращает #VALUE! (значение ошибки).
- Если  $\alpha \leq 0$  или  $\alpha \geq 1$ , то CONFIDENCE.T возвращает #NUM! (значение ошибки).
- Если  $\text{standard\_dev} \leq 0$ , то CONFIDENCE.T возвращает #NUM! (значение ошибки).
- Если size не является целым числом, значение округляется.
- Если size равен 1, то 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

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
---------	-------------	-----------

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= CONFIDENCE.T(0,05,1,50)	Доверительный интервал для среднего значения генеральной совокупности, основанный на размере выборки 50, с уровнем значимости 5 % и стандартным отклонением 1. Это основано на распределении Стьюдента.	0.284196855

# COT

22.09.2020 • 2 minutes to read

Возвращает котангенс угла, указанного в радианах.

## Синтаксис

```
COT (<number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
число	Угол в радианах, котангенс которого нужно найти.

## Возвращаемое значение

Котангенс заданного угла.

## Примечания

- Абсолютное значение числа должно быть меньше  $2^{27}$  и не равно 0.
- Если число находится за пределами этого диапазона, возвращается ошибка.
- Если оно представлено нечисловым значением, возвращается ошибка.

## Пример

Рассмотрим следующий запрос DAX:

```
EVALUATE { COT(30) }
```

Возвращает

[ЗНАЧЕНИЕ]
-0,156119952161659

# COTH

02.10.2020 • 2 minutes to read

Возвращает гиперболический котангенс гиперболического угла.

## Синтаксис

```
COTH (<number>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
число	Гиперболический угол в радианах, для которого нужно получить гиперболический котангенс.

## Возвращаемое значение

Гиперболический котангенс заданного угла.

## Примечания

- Гиперболический котангенс является аналогом обычного котангенса (окружности).
- Абсолютное значение числа должно быть меньше  $2^{27}$  и не равно 0.
- Если число находится за пределами этого диапазона, возвращается ошибка.
- Если оно представлено нечисловым значением, возвращается ошибка.
- Используется следующее уравнение:

$$\text{COTH}(N) = \frac{1}{\text{TANH}(N)} = \frac{\text{COSH}(N)}{\text{SINH}(N)} = \frac{e^N + e^{-N}}{e^N - e^{-N}}$$

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

## Пример

Рассмотрим следующий запрос DAX:

```
EVALUATE { COTH(2) }
```

Возвращает

[ЗНАЧЕНИЕ]

1,03731472072755

# COUNT

02.10.2020 • 2 minutes to read

Функция COUNT подсчитывает количество ячеек в столбце, которые содержат непустые значения.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий значения для подсчета.

## Возвращаемое значение

Целое число.

## Remarks

- Единственным аргументом, допустимым для этой функции, является столбец. Функция COUNT подсчитывает количество строк, содержащих следующие виды значений:
  - Числа
  - даты.
  - строк
- Если функция не находит строк для подсчета, она возвращает пустое значение.
- Пустые значения пропускаются. Значения TRUE и FALSE не поддерживаются.
- Если требуется оценить столбец значений TRUE или FALSE, используйте функцию COUNT.
- 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 значений в столбце ShipDate.

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

Если необходимо подсчитать логические или текстовые значения, используйте функцию COUNTA или COUNTAX.

## См. также раздел

[Функция COUNTA](#)  
[COUNTAX, функция](#)



# COUNTA

02.10.2020 • 2 minutes to read

Функция COUNTA подсчитывает количество непустых ячеек в столбце.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий значения для подсчета

## Возвращаемое значение

Целое число.

## Remarks

- Если функция не находит строк для подсчета, возвращается 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 возвращаются все строки в таблице `Reseller`, которые имеют любое значение в столбце, где хранятся номера телефонов. Поскольку имя таблицы не содержит пробелов, кавычки необязательны.

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

## См. также раздел

[Функция COUNT](#)

[COUNTAX, функция](#)

[COUNTX, функция](#)

[Статистические функции](#)



# COUNTAX

02.10.2020 • 2 minutes to read

Функция COUNTAX подсчитывает непустые результаты при вычислении результата выражения для таблицы. То есть она работает так же, как функция COUNTA, только используется для итерации по строкам в таблице и подсчета количества строк, в которых указанные выражения приводят к непустому результату.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.

## Возвращаемое значение

Целое число.

## Remarks

- Как и функция COUNTA, функция COUNTAX подсчитывает ячейки, содержащие данные любого типа, включая другие выражения. Например, если столбец содержит выражение, результатом которого является пустая строка, функция COUNTAX считает этот результат непустым. Обычно функция COUNTAX не учитывает пустые ячейки, но в данном случае ячейка содержит формулу, поэтому она учитывается.
- Если нет функции 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 подсчитывается количество непустых строк в столбце Phone с использованием таблицы, полученной в результате фильтрации таблицы Reseller по столбцу [Status] = Active.

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

## См. также раздел

[Функция COUNT](#)

[Функция COUNTA](#)

COUNTX, функция

Статистические функции

# COUNTBLANK

02.10.2020 • 2 minutes to read

Подсчитывает количество пустых ячеек в столбце.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий пустые ячейки для подсчета.

## Возвращаемое значение

Целое число. Если не найдены строки, соответствующие условиям, возвращаются пустые значения.

## Remarks

- Единственным аргументом, допустимым для этой функции, является столбец. Можно использовать столбцы, содержащие данные любого типа, но учитываются только пустые ячейки. Ячейки, имеющие нулевое значение (0), не учитываются, так как ноль считается числовым значением, а не пустым.
- Если нет строк для статистической обработки, функция возвращает пустое значение. Однако если имеются строки, но ни одна из них не соответствует указанным критериям, функция возвращает 0. Microsoft Excel также возвращает ноль, если не найдены строки, соответствующие условиям.
- Иными словами, если функция COUNTBLANK не находит пустые значения, 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 таблице Reseller, в которых имеются пустые значения для BankName.

```
= COUNTBLANK(Reseller[BankName])
```

Если необходимо подсчитать логические или текстовые значения, используйте функцию COUNTA или COUNTAX.

## См. также

[Функция COUNT](#)

[Функция COUNTA](#)

COUNTAX, функция

COUNTX, функция

Статистические функции

# COUNTROWS

02.10.2020 • 2 minutes to read

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

## Синтаксис

```
COUNTROWS(<table>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Имя таблицы, содержащей строки для подсчета, или выражение, возвращающее таблицу.

## Возвращаемое значение

Целое число.

## Remarks

- Эта функция может использоваться для подсчета количества строк в базовой таблице, но чаще используется для подсчета количества строк, которые были результатом фильтрации таблицы, или применения контекста к таблице.
- Если нет строк для статистической обработки, функция возвращает пустое значение. Однако если имеются строки, но ни одна из них не соответствует указанным критериям, функция возвращает 0. Microsoft Excel также возвращает ноль, если не найдены строки, соответствующие условиям.
- 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.

```
= COUNTROWS('Orders')
```

## Пример 2

В следующем примере показано, как использовать COUNTROWS с контекстом строки. В этом сценарии существует два набора данных, которые связаны по номеру заказа. В таблице Reseller содержится по одной строке для каждого торгового посредника; в таблице ResellerSales содержится несколько строк для каждого заказа, где каждая строка содержит один заказ для конкретного торгового посредника. Таблицы соединяются связью по столбцу ResellerKey.

Формула возвращает значение ResellerKey, а затем подсчитывает количество строк в связанной таблице, имеющих одинаковый идентификатор торгового посредника. Результат выводится в столбце Calculatedcolumn1.

```
= COUNTROWS(RELATEDTABLE(ResellerSales))
```

В следующей таблице показана часть ожидаемых результатов:

RESELLERKEY	CALCULATEDCOLUMN1
1	73
2	70
3	394

См. также раздел

[Функция COUNT](#)

[Функция COUNTA](#)

[COUNTAX, функция](#)

[COUNTX, функция](#)

[Статистические функции](#)

# COUNTX

02.10.2020 • 2 minutes to read

Подсчитывает количество строк, содержащих непустое значение, или выражение, результатом вычисления которого является значение, отличное от пустого, при вычислении выражения для таблицы.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки для подсчета.
expression	Выражение, возвращающее набор значений, содержащих значения, которые необходимо подсчитать.

## Возвращаемое значение

Целое число.

## Remarks

- Функция COUNTX принимает два аргумента. Первый аргумент всегда должен быть таблицей или любым выражением, возвращающим таблицу. Вторым аргументом является столбец или выражение, поиск которого осуществляется по COUNTX.
- Функция COUNTX подсчитывает только значения, даты или строки. Если функция не находит строк для подсчета, она возвращает пустое значение.
- Если необходимо 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 продуктов, имеющих цену по преysкуранту.

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

## Пример 2

В следующей формуле показано, как передать отфильтрованную таблицу в COUNTX для первого аргумента. Формула использует критерий фильтра для получения только тех строк таблицы продуктов, которые соответствуют условию, ProductSubCategory = "Caps", а затем подсчитывает строки в

результатирующей таблице со стоимостью по преЙскуранту. Критерий фильтра применяется к таблице продуктов, но использует значение, которое вы ищете в связанной таблице ProductSubCategory.

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

## См. также раздел

[Функция COUNT](#)

[Функция COUNTA](#)

[COUNTAX, функция](#)

[Статистические функции](#)



# DISTINCTCOUNT

02.10.2020 • 2 minutes to read

Подсчитывает количество уникальных значений в столбце.

## Синтаксис

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

### Параметры

ТЕРМИН	DESCRIPTION
столбец	Столбец, содержащий значения для подсчета

## Возвращаемое значение

Количество уникальных значений в *column*.

## Remarks

- Единственным аргументом, допустимым для этой функции, является столбец. Можно использовать столбцы, содержащие данные любого типа. Если функция не находит строки для подсчета, она возвращает значение BLANK, в противном случае возвращается число уникальных значений.
- Функция DISTINCTCOUNT включает 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 приведенном ниже примере показано, как подсчитать количество уникальных заказов на продажу в столбце ResellerSales\_USD[SalesOrderNumber].

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

Использование приведенной выше меры в таблице с календарным годом в боковой и верхней категориях продукции дает следующие результаты.

ПОДСЧЕТ РАЗЛИЧНЫХ ЗАКАЗОВ ЧЕРЕЗ ТОРГОВОГО ПОСРЕДНИКА	МЕТКИ СТОЛБЦОВ					
Метки строк	Accessories	Bikes	Clothing	Components		Общий итог

ПОДСЧЕТ РАЗЛИЧНЫХ ЗАКАЗОВ ЧЕРЕЗ ТОРГОВОГО ПОСРЕДНИКА	МЕТКИ СТОЛБЦОВ					
2005	135	345	242	205		366
2006	356	850	644	702		1015
2007 г.	531	1 234	963	1138		1 521
2008	293	724	561	601		894
					1	1
Grand Total	1315	3 153	2410	2 646	1	3797

В приведенном выше примере видно несоответствие чисел общего итога строк. Это происходит потому, что один порядок может содержать линейные элементы в том же порядке из разных категорий продуктов.

## См. также

[Функция COUNT](#)

[Функция COUNTA](#)

[COUNTAX, функция](#)

[COUNTX, функция](#)

[Статистические функции](#)

# EXPON.DIST

02.10.2020 • 2 minutes to read

Возвращает экспоненциальное распределение. Используйте EXPON.DIST для моделирования времени между событиями, например тем, как долго банкомат будет выдавать деньги. Так, можно использовать EXPON.DIST, чтобы определить вероятность того, что процесс займет не более минуты.

## Синтаксис

```
EXPON.DIST(x,lambda,cumulative)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
x	Обязательный элемент. Значение функции.
lambda	Обязательный элемент. Значение параметра.
cumulative	Обязательный элемент. Логическое значение, указывающее, какая форма экспоненциальной функции должна быть предоставлена. Если аргумент имеет значение TRUE, функция EXPON.DIST возвращает интегральную функцию распределения, а если FALSE — функцию плотности вероятности.

## Возвращаемое значение

Возвращает экспоненциальное распределение.

## Remarks

- Если x или lambda не является числом, EXPON.DIST возвращает #VALUE! (значение ошибки).
- Если x или lambda не является целым числом, значение округляется.
- Если  $x < 0$ , EXPON.DIST возвращает #NUM! (значение ошибки).
- Если  $\lambda \leq 0$ , EXPON.DIST возвращает #NUM! (значение ошибки).

- Уравнение для функции плотности вероятности имеет следующий вид:

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

- Уравнение для функции интегрального распределения имеет следующий вид:

$$F(x; \lambda) = 1 - e^{-\lambda x}$$

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

# GEOMEAN

02.10.2020 • 2 minutes to read

Возвращает среднее геометрическое чисел в столбце.

Чтобы вернуть среднее геометрическое выражения, вычисленного для каждой строки в таблице, используйте [функцию GEOMEANX](#).

## Синтаксис

```
GEOMEAN(<column>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий числа, для которых вычисляется геометрическое среднее.

## Возвращаемое значение

Десятичное число.

## Remarks

- Учитываются только числа в столбце. Пустые значения, логические значения и текст игнорируются.
- `GEOMEAN( Table[Column] )` — эквивалент `GEOMEANX( 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

В следующем примере вычисляется геометрическое среднее значений в столбце Return таблицы Investment:

```
= GEOMEAN( Investment[Return] )
```

## См. также раздел

[Функция GEOMEANX](#)

# GEOMEANX

02.10.2020 • 2 minutes to read

Возвращает среднее геометрическое выражения, вычисляемого для каждой строки в таблице.

Чтобы вернуть среднее геометрическое чисел в столбце, используйте [функцию GEOMEAN](#).

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.

## Возвращаемое значение

Десятичное число.

## Remarks

- Функция GEOMEANX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Вторым аргументом является столбец, содержащий числа, для которых вычисляется среднее геометрическое, или выражение, результатом которого является столбец.
- Учитываются только числа в столбце. Пустые значения, 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 следующем примере вычисляется геометрическое среднее значений в столбце ReturnPct таблицы Investments:

```
= GEOMEANX( Investments, Investments[ReturnPct] + 1 )
```

## См. также раздел

[Функция GEOMEAN](#)

# MAX

02.10.2020 • 2 minutes to read

Возвращает максимальное значение из столбца или из двух скалярных выражений.

## Синтаксис

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

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, в котором определяется наибольшее значение.
expression	Любое выражение DAX, возвращающее одиночное значение.

## Возвращаемое значение

Самое высокое значение.

## Remarks

- При сравнении двух выражений пустое значение обрабатывается как 0. Это значит, что `Max(1, Blank())` возвращает 1, а `Max(-1, Blank())` возвращает значение 0. Если оба аргумента пусты, функция MAX возвращает пустое значение. Если какое-то из выражений возвращает недопустимое значение, функция MAX возвращает ошибку.
- Значения TRUE и FALSE не поддерживаются. Если требуется оценить столбец значений TRUE или FALSE, используйте функцию MAXA.

## Пример 1

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

```
= MAX(InternetSales[ExtendedAmount])
```

## Пример 2

В следующем примере возвращается наибольшее значение из двух выражений.

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

См. также раздел

[Функция МАХА](#)

[Функция МАХХ](#)

[Статистические функции](#)

# MAXA

02.10.2020 • 2 minutes to read

Возвращает максимальное значение в столбце.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, в котором определяется наибольшее значение.

## Возвращаемое значение

Самое высокое значение.

## Remarks

- Функция MAXA принимает в качестве аргумента столбец и ищет максимальное значение из следующих типов значений:
  - Числа
  - даты.
- Логические значения, такие как TRUE и FALSE. Строки, вычисляемые как TRUE, равны 1; строки, вычисляемые как FALSE, равны 0 (нулю).
- Пустые ячейки не учитываются. Если столбец не содержит значений, которые можно использовать, функция MAXA возвращает 0 (ноль).
- Если необходимо 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 означает разницу между ценой по прейскуранту и ценой торгового посредника.

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

## Пример 2

В следующем примере возвращается наибольшее значение из столбца, содержащего даты и время. Таким образом, эта формула возвращает дату последней транзакции.



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

См. также раздел

[Функция MAX](#)

[Функция MAXX](#)

[Статистические функции](#)

# MAXX

02.10.2020 • 2 minutes to read

Вычисляет выражение для каждой строки таблицы и возвращает максимальное значение.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.

## Возвращаемое значение

Самое высокое значение.

## Remarks

- Аргумент table функции MAXX может быть именем таблицы или выражением, результатом вычисления которого является таблица. Второй аргумент задает выражение, вычисляемое для каждой строки таблицы.
- Из значений для вычисления учитывается только следующее:
  - Числа
  - Тексты
  - даты.
- Пустые значения пропускаются. Значения TRUE и FALSE не поддерживаются.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример 1

В следующей формуле в качестве второго аргумента используется выражение для вычисления общего объема налогов и цены доставки для каждого заказа в таблице InternetSales. Ожидаемый результат: 375,7184.

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

## Пример 2

Следующая формула сначала фильтрует таблицу InternetSales с помощью критерия фильтра, чтобы

вернуть подмножество 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 максимальную сумму налогов и цены доставки только для этих заказов. Ожидаемый результат: 250,3724.

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

## См. также раздел

[Функция MAX](#)

[Функция MAXA](#)

[Статистические функции](#)

# MEDIAN

02.10.2020 • 2 minutes to read

Возвращает медиану чисел в столбце.

Чтобы вернуть медиану выражения, вычисленного для каждой строки в таблице, используйте функцию [MEDIANX](#).

## Синтаксис

```
MEDIAN(<column>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий числа, для которых вычисляется медиана.

## Возвращаемое значение

Десятичное число.

## Remarks

- Учитываются только числа в столбце. Пустые значения, логические значения и текст игнорируются.
- `MEDIAN( Table[Column] )` — эквивалент `MEDIANX( 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

В следующем примере вычисляется медиана столбца с именем Age в таблице Customers:

```
= MEDIAN( Customers[Age] )
```

## См. также раздел

[Функция MEDIANX](#)

# MEDIANX

02.10.2020 • 2 minutes to read

Возвращает медианное значение выражения, вычисляемого для каждой строки в таблице.

Чтобы вернуть медиану чисел в столбце, используйте функцию [MEDIAN](#).

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.

## Возвращаемое значение

Десятичное число.

## Remarks

- Функция MEDIANX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Вторым аргументом является столбец, содержащий числа, для которых вычисляется медиана, или выражение, результатом которого является столбец.
- Учитываются только числа в столбце.
- Логические значения и текст игнорируются.
- MEDIANX не пропускает пустые значения, в отличие от MEDIAN.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Ниже приводится вычисление медианного возраста клиентов, проживающих в США.

```
= MEDIANX( FILTER(Customers, RELATED( Geography[Country]="USA" ) ), Customers[Age] )
```

## См. также раздел

[Функция MEDIAN](#)

# MIN

02.10.2020 • 2 minutes to read

Возвращает минимальное значение из столбца или из двух скалярных выражений.

## Синтаксис

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

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

## Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, в котором определяется наименьшее значение.
expression	Любое выражение DAX, возвращающее одиночное значение.

## Возвращаемое значение

Наименьшее значение.

## Remarks

- Функция MIN принимает в качестве аргумента столбец или два выражения и возвращает наименьшее значение. Учитываются следующие типы значений в столбцах:
  - Числа
  - Тексты
  - даты.
  - Пустые значения
- При сравнении выражений пустое значение обрабатывается как 0. Это значит, что `Min(1, Blank())` возвращает 0, а `Min(-1, Blank())` возвращает значение -1. Если оба аргумента пусты, функция MIN возвращает пустое значение. Если какое-то из выражений возвращает недопустимое значение, функция MIN возвращает ошибку.
- Значения TRUE и FALSE не поддерживаются. Если требуется оценить столбец значений TRUE или FALSE, используйте функцию MINA.

## Пример 1

В следующем примере возвращается наименьшее значение из вычисляемого столбца ResellerMargin.

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

## Пример 2

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

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

## Пример 3

В следующем примере возвращается наименьшее значение из двух скалярных выражений.

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

## См. также раздел

[Функция MINA](#)

[Функция MINX](#)

[Статистические функции](#)

# MINA

02.10.2020 • 2 minutes to read

Возвращает минимальное значение в столбце.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, для которого определяется наименьшее значение.

## Возвращаемое значение

Наименьшее значение.

## Remarks

- Функция MINA принимает в качестве аргумента столбец, который содержит числа, и определяет наименьшее значение следующим образом:
  - Если столбец не содержит значений, функция MINA возвращает 0 (ноль).
  - Строки в столбце, результатом вычисления которых являются логические значения, такие как TRUE и FALSE, обрабатываются как в случае TRUE и 0 (ноль) в случае FALSE.
  - Пустые ячейки не учитываются.
- Если необходимо 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 минимальный тариф на фрахт из таблицы InternetSales.

```
= MINA(InternetSales[Freight])
```

## Пример 2

Следующее выражение возвращает минимальное значение в столбце PostalCode. Поскольку тип данных столбца — текстовый, функция не находит никаких значений, а формула возвращает ноль (0).

```
= MINA([PostalCode])
```



См. также раздел

[Функция MIN](#)

[Функция MINX](#)

[Статистические функции](#)

# MINX

02.10.2020 • 2 minutes to read

Возвращает наименьшее значение, полученное в результате вычисления выражения для каждой строки таблицы.

## Синтаксис

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

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.

## Возвращаемое значение

Наименьшее значение.

## Remarks

- Функция MINX принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Второй аргумент содержит выражение, вычисляемое для каждой строки таблицы.
- Пустые значения пропускаются. Значения TRUE и FALSE не поддерживаются.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример 1

Следующий пример фильтрует таблицу InternetSales и возвращает только строки для определенной территории продаж. Затем формула находит минимальное значение в столбце Freight.

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

## Пример 2

В следующем примере используется та же отфильтрованная 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 в результате вычисления.

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

В первом примере имена столбцов являются неполными. Во втором примере имена столбцов являются полными.

## См. также раздел

[Функция MIN](#)

[Функция MINA](#)

[Статистические функции](#)

# NORM.DIST

22.09.2020 • 2 minutes to read

Возвращает нормальное распределение для указанного среднего значения и стандартного отклонения.

## Синтаксис

```
NORM.DIST(X, Mean, Standard_dev, Cumulative)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
X	Значение, для которого требуется распределение.
Среднее значение	Среднее арифметическое распределения.
Standard_dev	Стандартное отклонение распределения.
Cumulative*	Логическое значение, определяющее форму функции. Если аргумент имеет значение TRUE, функция NORM.DIST возвращает интегральную функцию распределения, а если FALSE — функцию плотности вероятности.

## Возвращаемое значение

Нормальное распределение для указанного среднего значения и стандартного отклонения.

## Remarks

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

## Пример

```
EVALUATE { NORM.DIST(42, 40, 1.5, TRUE) }
```

Возвращает

[ЗНАЧЕНИЕ]
0,908788780274132

## См. также

[Функция NORM.S.DIST](#)

[Функция NORM.INV](#)

[NORM.S.INV](#)



# NORM.INV

22.09.2020 • 2 minutes to read

Обратная функция нормального интегрального распределения для указанного среднего значения и стандартного отклонения.

## Синтаксис

```
NORM.INV(Probability, Mean, Standard_dev)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Вероятность	Вероятность, соответствующая нормальному распределению.
Среднее значение	Среднее арифметическое распределения.
Standard_dev	Стандартное отклонение распределения.

## Возвращаемое значение

Возвращает обратную функцию нормального интегрального распределения для указанного среднего значения и стандартного отклонения.

## Remarks

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

## Пример

```
EVALUATE { NORM.INV(0.908789, 40, 1.5) }
```

### Возвращает

[ЗНАЧЕНИЕ]
42.00000200956628780274132

## См. также

[NORM.S.INV](#)

[Функция NORM.S.DIST](#)

[Функция NORM.DIST](#)

# NORM.S.DIST

22.09.2020 • 2 minutes to read

Возвращает стандартное нормальное распределение (среднее значение равно нулю, а стандартное отклонение — единице).

## Синтаксис

```
NORM.S.DIST(Z, Cumulative)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Z	Значение, для которого требуется распределение.
Cumulative	Логическое значение, определяющее форму функции. Если аргумент имеет значение TRUE, функция NORM.S.DIST возвращает интегральную функцию распределения, а если FALSE — функцию плотности вероятности.

## Возвращаемое значение

Стандартное нормальное распределение (среднее значение равно нулю, а стандартное отклонение — единице).

## Remarks

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

## Пример

```
EVALUATE { NORM.S.DIST(1.333333, TRUE) }
```

Возвращает

[ЗНАЧЕНИЕ]
0,908788725604095

## См. также

[Функция NORM.INV](#)  
[Функция NORM.DIST](#)  
[NORM.S.INV](#)

# NORM.S.INV

22.09.2020 • 2 minutes to read

Возвращает обратное значение стандартного нормального интегрального распределения. Распределение (среднее значение равно нулю, а стандартное отклонение — единице).

## Синтаксис

```
NORM.S.INV(Probability)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Вероятность	Вероятность, соответствующая нормальному распределению.

## Возвращаемое значение

Обратное значение стандартного нормального интегрального распределения. Распределение (среднее значение равно нулю, а стандартное отклонение — единице).

## Remarks

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

## Пример

```
EVALUATE { NORM.S.INV(0.908789) }
```

Возвращает

[ЗНАЧЕНИЕ]

1,33333467304411

## См. также

[NORM.INV](#)

[Функция NORM.S.DIST](#)

[Функция NORM.DIST](#)



# PERCENTILE.EXC

02.10.2020 • 2 minutes to read

Возвращает k-й процентиль значений в диапазоне, где k находится в диапазоне 0...1, исключая крайние значения.

Чтобы вернуть число для процентиля выражения, вычисляемого для каждой строки в таблице, используйте функцию [PERCENTILE.EXC](#).

## Синтаксис

```
PERCENTILE.EXC(<column>, <k>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий значения, определяющие относительное место.
k	Значение процентиля в диапазоне 0..1, не включительно.

## Возвращаемое значение

k-й процентиль значений в диапазоне, где k находится в диапазоне 0...1, исключая крайние значения.

## Remarks

- Если столбец пуст, возвращается BLANK().
- Если k равно нулю или является пустым, то процентиль  $1/(n + 1)$  возвращает наименьшее значение. Если значение равно нулю, оно выходит за пределы допустимого диапазона, и возвращается ошибка.
- Если k не является числом или находится вне диапазона от 0 до 1, возвращается ошибка.
- Если k не кратно  $1/(n + 1)$ , PERCENTILE.EXC выполняет интерполяцию, чтобы определить значение на процентиле k.
- PERCENTILE.EXC выполняет интерполяцию, если значение указанного процентиля находится между двумя значениями в массиве. Если невозможно выполнить интерполяцию для указанного процентиля k, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## См. также раздел

[PERCENTILEX.EXC](#)

# PERCENTILE.INC

02.10.2020 • 2 minutes to read

Возвращает k-й процентиль значений в диапазоне, где k находится в диапазоне 0..1, включая крайние значения.

Чтобы вернуть число для процентиля выражения, вычисляемого для каждой строки в таблице, используйте функцию [PERCENTILE.INC](#).

## Синтаксис

```
PERCENTILE.INC(<column>, <k>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, содержащий значения, определяющие относительное место.
k	Значение процентиля в диапазоне 0..1, включительно.

## Возвращаемое значение

k-й процентиль значений в диапазоне, где k находится в диапазоне 0..1, включая крайние значения.

## Remarks

- Если столбец пуст, возвращается BLANK().
- Если k равно нулю или является пустым, то процентиль  $1/(n + 1)$  возвращает наименьшее значение. Если значение равно нулю, оно выходит за пределы допустимого диапазона, и возвращается ошибка.
- Если k не является числом или находится вне диапазона от 0 до 1, возвращается ошибка.
- Если k не кратно  $1/(n + 1)$ , PERCENTILE.INC выполняет интерполяцию, чтобы определить значение на процентиле k.
- PERCENTILE.INC выполняет интерполяцию, если значение указанного процентиля находится между двумя значениями в массиве. Если невозможно выполнить интерполяцию для указанного процентиля k, возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## См. также раздел

[PERCENTILEX.INC](#)

# PERCENTILEX.EXC

02.10.2020 • 2 minutes to read

Возвращает число для процентиля выражения, вычисляемого для каждой строки в таблице.

Чтобы получить процентиль чисел в столбце, используйте [функцию PERCENTILEX.EXC](#).

## Синтаксис

```
PERCENTILEX.EXC(<table>, <expression>, k)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.
k	Желаемое значение процентиля в диапазоне 0–1, не включительно.

## Возвращаемое значение

Число для процентиля выражения, вычисляемого для каждой строки в таблице.

## Remarks

- Если  $k$  равно нулю или является пустым, то процентиль  $1/(n + 1)$  возвращает наименьшее значение. Если значение равно нулю, оно выходит за пределы допустимого диапазона, и возвращается ошибка.
- Если  $k$  не является числом или находится вне диапазона от 0 до 1, возвращается ошибка.
- Если  $k$  не кратно  $1/(n + 1)$ , PERCENTILEX.EXC выполняет интерполяцию, чтобы определить значение на процентиле  $k$ .
- PERCENTILEX.EXC выполняет интерполяцию, если значение указанного процентиля находится между двумя значениями в массиве. Если невозможно выполнить интерполяцию для указанного процентиля  $k$ , возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## См. также раздел

[PERCENTILEX.EXC](#)

# PERCENTILEX.INC

02.10.2020 • 2 minutes to read

Возвращает число для процентиля выражения, вычисляемого для каждой строки в таблице.

Чтобы получить процентиль чисел в столбце, используйте [функцию PERCENTILEX.INC](#).

## Синтаксис

```
PERCENTILEX.INC(<table>, <expression>;, k)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.
k	Желаемое значение процентиля в диапазоне 0–1 включительно.

## Возвращаемое значение

Число для процентиля выражения, вычисляемого для каждой строки в таблице.

## Remarks

- Если  $k$  равно нулю или является пустым, то процентиль  $1/(n - 1)$  возвращает наименьшее значение. Если значение равно нулю, оно выходит за пределы допустимого диапазона, и возвращается ошибка.
- Если  $k$  не является числом или находится вне диапазона от 0 до 1, возвращается ошибка.
- Если  $k$  не кратно  $1/(n - 1)$ , PERCENTILEX.EXC выполняет интерполяцию, чтобы определить значение на процентиле  $k$ .
- PERCENTILEX.INC выполняет интерполяцию, если значение указанного процентиля находится между двумя значениями в массиве. Если невозможно выполнить интерполяцию для указанного процентиля  $k$ , возвращается ошибка.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## См. также раздел

[PERCENTILEX.INC](#)

# PERMUT

02.10.2020 • 2 minutes to read

Возвращает число перестановок для заданного числа объектов, которые можно выбрать из числовых объектов. Перестановка — это любой набор или подмножество объектов или событий, где важен внутренний порядок. Перестановки отличаются от комбинаций, для которых внутренний порядок не важен. Используйте эту функцию для вычислений вероятности в стиле лотереи.

## Синтаксис

```
PERMUT(number, number_chosen)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Обязательный элемент. Целое число, которое описывает количество объектов.
number_chosen	Обязательный элемент. Целое число, которое описывает количество объектов при каждой перестановке.

## Возвращаемое значение

Возвращает число перестановок для заданного числа объектов, которые можно выбрать из числовых объектов.

## Remarks

- Оба аргумента усекаются до целых значений.
- Если number или number\_chosen не является числом, то функция PERMUT возвращает #VALUE! (значение ошибки).
- Если  $\text{number} \leq 0$  или  $\text{number\_chosen} < 0$ , то функция PERMUT возвращает #NUM! (значение ошибки).
- Если  $\text{number} < \text{number\_chosen}$ , то функция PERMUT возвращает #NUM! (значение ошибки).
- Уравнение для числа перестановок имеет следующие значения:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

- 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 возможны перестановки для группы из трех объектов, из которой выбираются два.

```
= PERMUT(3,2)
```

Результат:

# POISSON.DIST

02.10.2020 • 2 minutes to read

Возвращает распределение Пуассона. Оно обычно используется для прогнозирования числа событий за определенное время, например, сколько машин приедет на парковку за минуту.

## Синтаксис

```
POISSON.DIST(x,mean,cumulative)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
x	Обязательный элемент. Число событий.
mean	Обязательный элемент. Ожидаемое числовое значение.
cumulative	Обязательный элемент. Логическое значение, определяющее форму возвращаемого распределения вероятностей. Если накопительное значение — TRUE, то функция POISSON.DIST возвращает интегральную вероятность Пуассона, относящуюся к числу случайных событий между 0 и X включительно; в противном случае — вероятностную меру Пуассона, соответствующую тому, что число произошедших событий в точности равно X.

## Возвращаемое значение

Возвращает распределение Пуассона.

## Remarks

- Если x не является целым числом, значение округляется.
- Если x или mean не является числом, POISSON.DIST возвращает #VALUE! (значение ошибки).
- Если  $x < 0$ , POISSON.DIST возвращает #NUM! (значение ошибки).
- Если  $mean < 0$ , POISSON.DIST возвращает #NUM! (значение ошибки).
- Функция POISSON.DIST вычисляется следующим образом.

- Для cumulative = FALSE:

$$\text{POISSON} = \frac{e^{-\lambda} \lambda^x}{x!}$$

- Для cumulative = TRUE:

$$\text{CUMPOISSON} = \sum_{k=0}^x \frac{e^{-\lambda} \lambda^k}{k!}$$

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





# RANK.EQ

02.10.2020 • 3 minutes to read

Возвращает ранжирование числа в списке чисел.

## Синтаксис

```
RANK.EQ(<value>, <columnName>[, <order>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
значение	Любое выражение DAX, возвращающее одиночное скалярное значение, ранг которого должен быть найден. Выражение вычисляется только один раз, перед вычислением функции, и полученное значение передается в список аргументов.
columnName	Имя существующего столбца, по которому будут определяться ранги. Оно не может быть выражением или столбцом, созданным с помощью этих функций: ADDCOLUMNS, ROW или SUMMARIZE.
порядок	(необязательно) Значение, указывающее способ ранжирования <i>number</i> — от наименьшего до наибольшего или от наибольшего до наименьшего:

### порядковые значения

ЗНАЧЕНИЕ	АЛЬТЕРНАТИВНОЕ ЗНАЧЕНИЕ	ОПИСАНИЕ
0 (ноль)	FALSE	Ранги в порядке убывания <i>columnName</i> . Если значение <i>value</i> равно наибольшему числу в <i>columnName</i> , RANK.EQ возвращает 1.
1	true	Ранги в порядке возрастания <i>columnName</i> . Если значение <i>value</i> равно наименьшему числу в <i>columnName</i> , RANK.EQ возвращает 1.

## Возвращаемое значение

Число, указывающее ранг значения *value* среди чисел в *columnName*.

## Remarks

- Столбец *columnName* не может ссылаться ни на один столбец, созданный с помощью этих функций: ADDCOLUMNS, ROW или SUMMARIZE.I
- Если *value* отсутствует в *columnName* или является пустым, RANK.EQ возвращает пустое значение.

- Дублирующиеся значения *value* получают одно и то же значение ранга; значение следующего по порядку ранга будет равно этому значению ранга, к которому прибавлено число повторяющихся значений. Например, если пять (5) значений связаны с рангом 11, то следующему значению будет присвоен 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* для всех чисел в одном столбце.

```
= RANK.EQ(InternetSales_USD[SalesAmount_USD], InternetSales_USD[SalesAmount_USD])
```

## Пример 2

В следующем примере производится ранжирование подмножества значений по заданному примеру.

Предположим, у вас есть таблица учащихся с их результатами за определенную национальную проверочную работу, а также полный набор оценок в этой работе. В следующем вычисляемом столбце вы получите общенациональные ранги для каждого из учащихся.

```
= RANK.EQ(Students[Test_Score], NationalScores[Test_Score])
```

# RANKX

02.10.2020 • 4 minutes to read

Возвращает ранжирование числа в списке чисел для каждой строки в аргументе *table*.

## Синтаксис

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

### Параметры

#### **table**

Любое выражение DAX, возвращающее таблицу данных, в которой вычисляется выражение.

#### **expression**

Любое выражение DAX, возвращающее одиночное скалярное значение. Выражение вычисляется для каждой строки *table*, чтобы создать все возможные значения для ранжирования. См. раздел "Примечания", чтобы понять поведение функции, когда результатом вычисления *expression* является значение BLANK.

#### **value**

(необязательно) Любое выражение DAX, возвращающее одиночное скалярное значение, ранг которого должен быть найден. См. раздел "Примечания", чтобы понять поведение функции, когда параметр *value* не найден в выражении.

Если значение параметра *value* опущено, вместо него используется значение выражения в текущей строке.

#### **order**

(необязательно) Значение, указывающее способ ранжирования *value* — от наименьшего до наибольшего или от наибольшего до наименьшего:

VALUE	АЛЬТЕРНАТИВНОЕ ЗНАЧЕНИЕ	DESCRIPTION
0 (ноль)	FALSE	Ранжирование значений выражения в порядке убывания. Если значение равно наибольшему числу в выражении, RANKX возвращает 1.  Это значение по умолчанию, если параметр <i>order</i> опущен.
1	TRUE	Ранжирование значений выражения в порядке возрастания. Если значение равно наименьшему числу в выражении, RANKX возвращает 1.

#### **ties**

(необязательно) Перечисление, определяющее порядок ранжирования при наличии связей.

ПЕРЕЧИСЛЕНИЕ	ОПИСАНИЕ
--------------	----------

ПЕРЕЧИСЛЕНИЕ	ОПИСАНИЕ
Skip	<p>Следующее значение ранжирования — это значение ранга для связи, а также число привязанных значений. Например, если пять (5) значений связаны с рангом 11, то следующему значению будет присвоен 16 (11 + 5).</p> <p>Это значение по умолчанию, если параметр <i>ties</i> опущен.</p>
Dense	<p>Следующее значение ранга — это следующий ранг. Например, если пять (5) значений связаны с рангом 11, то следующее значение получит ранг 12.</p>

## Возвращаемое значение

Номер ранга параметра *value* среди всех возможных значений *expression*, вычисляемого для всех строк таблицы чисел *table*.

## Remarks

- Если результатом вычисления *expression* или *value* является значение BLANK, оно обрабатывается как 0 (ноль) для всех выражений, результатом которых является число, или как пустой текст для всех текстовых выражений.
- Если *value* отсутствует среди всех возможных значений *expression*, функция RANKX временно добавляет *value* к значениям из *expression* и повторно вычисляет RANKX для определения соответствующего ранга *value*.
- Необязательные аргументы можно пропустить путем помещения в список аргументов пустой запятой (,), т. е. RANKX(Inventory, [InventoryCost],,, "Dense")
- 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 Products вычисляет рейтинг продаж для каждого продукта в канале продаж через Интернет.

```
= RANKX(ALL(Products), SUMX(RELATEDTABLE(InternetSales), [SalesAmount]))
```

# SAMPLE

02.10.2020 • 2 minutes to read

Возвращает выборку из N строк из указанной таблицы.

## Синтаксис

```
SAMPLE(<n_value>, <table>, <orderBy_expression>, [<order>[, <orderBy_expression>, [<order>]]...])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
n_value	Количество возвращаемых строк. Это любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста). Если указано нецелочисленное значение (или выражение), результат приводится как целое число.
table	Любое выражение DAX, возвращающее таблицу данных, из которой нужно извлечь выборку строк "n".
orderBy_expression	(необязательно) Любое скалярное выражение DAX, в котором значение результата вычисляется для каждой строки <i>table</i> .
порядок	(необязательно) Значение, указывающее, как сортировать значения <i>orderBy_expression</i> , — по возрастанию или по убыванию: 0 (ноль), сортировка значений <i>order_by</i> в порядке убывания. 1, ранжирование <i>order_by</i> в порядке возрастания.

## Возвращаемое значение

Таблица, состоящая из выборки из N строк таблицы *table* или пустой таблицы, если *n\_value* имеет значение 0 (ноль) или меньше. Если указаны аргументы OrderBy, выборка будет стабильной и детерминированной, возвращая первую строку, последнюю строку и равномерно распределенные строки между ними. Если порядок не указан, выборка будет случайной, нестабильной и недетерминированной.

## Remarks

- Если значение *n\_value* равно 0 (нулю) или меньше, то SAMPLE возвращает пустую таблицу.
- Чтобы избежать дублирования значений в выборке, таблица, указанная в качестве второго аргумента, должна быть сгруппирована 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).

# SIN

02.10.2020 • 2 minutes to read

Возвращает синус заданного угла.

## Синтаксис

```
SIN(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Обязательный элемент. Угол в радианах, синус которого нужно найти.

## Возвращаемое значение

Возвращает синус заданного угла.

## Remarks

Если аргумент указан в градусах, умножьте его на  $\text{PI}()/180$  или используйте функцию **RADIANS**, чтобы преобразовать его в радианы.

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= SIN(PI())	Синус Пи радиан (0, приблизительно).	0,0
= SIN(PI()/2)	Синус Пи/2 радиан.	1.0
= SIN(30*PI()/180)	Синус 30 градусов.	0,5
= SIN(RADIANS(30))	Синус 30 градусов.	0,5

# SINH

02.10.2020 • 2 minutes to read

Возвращает гиперболический синус числа.

## Синтаксис

```
SINH(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Обязательный элемент. Любое вещественное число.

## Возвращаемое значение

Возвращает гиперболический синус числа.

## Remarks

- 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.

```
= 2.868*SINH(0.0342\*1.03)
```

Возвращает 0,1010491.

# STDEV.S

02.10.2020 • 2 minutes to read

Возвращает среднеквадратичное отклонение выборки из совокупности.

## Синтаксис

```
STDEV.S(<ColumnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX (обычно полное имя). Этот параметр не может быть выражением.

## Возвращаемое значение

Среднеквадратичное отклонение выборки из совокупности.

## Исключения

### Remarks

- STDEV.S предполагает, что столбец ссылается на выборку из совокупности. Если данные представляют собой полную совокупность, вычислите стандартное отклонение с помощью функции STDEV.P.
- Функция STDEV.S использует следующую формулу:

$$\sqrt{\frac{\sum (x - \tilde{x})^2}{(n-1)}}$$

где  $\tilde{x}$  — среднее значение  $x$  для образца заполнения, а  $n$  — размер генеральной совокупности.

- Пустые строки отфильтровываются из *columnName* и не учитываются в вычислениях.
- Если *columnName* содержит менее двух непустых строк, 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 показана формула для меры, которая вычисляет стандартное отклонение столбца SalesAmount\_USD, когда таблица InternetSales\_USD представляет собой выборку из совокупности.

```
= STDEV.S(InternetSales_USD[SalesAmount_USD])
```



# STDEV.P

02.10.2020 • 2 minutes to read

Возвращает среднеквадратичное отклонение всей совокупности.

## Синтаксис

```
STDEV.P(<ColumnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX (обычно полное имя). Этот параметр не может быть выражением.

## Возвращаемое значение

Число, представляющее стандартное отклонение всей совокупности.

## Remarks

- STDEV.P предполагает, что столбец ссылается на всю совокупность. Если данные представляют собой выборку из совокупности, вычислите стандартное отклонение с помощью функции STDEV.S.
- STDEV.P использует следующую формулу:

$$\sqrt{\sum (x - \tilde{x})^2 / n}$$

где  $\tilde{x}$  — среднее значение  $x$  для всей совокупности, а  $n$  — размер генеральной совокупности.

- Пустые строки отфильтровываются из *columnName* и не учитываются в вычислениях.
- Если *columnName* содержит менее двух непустых строк, 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 показана формула для меры, которая вычисляет стандартное отклонение столбца SalesAmount\_USD, когда таблица InternetSales\_USD представляет собой всю совокупность.

```
= STDEV.P(InternetSales_USD[SalesAmount_USD])
```

# STDEVX.S

02.10.2020 • 2 minutes to read

Возвращает среднее квадратичное отклонение выборки из совокупности.

## Синтаксис

```
STDEVX.S(<table>, <expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).
expression	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).

## Возвращаемое значение

Число со стандартным отклонением выборки из совокупности.

## Исключения

### Remarks

- Функция STDEVX.S вычисляет *expression* для каждой строки таблицы *table* и возвращает стандартное отклонение *expression*, предполагая, что эта таблица *table* ссылается на выборку из совокупности. Если *table* представляет собой полную совокупность, вычислите стандартное отклонение с помощью функции STDEVX.P.
- STDEVX.S использует следующую формулу:
$$\sqrt{\frac{\sum (x - \bar{x})^2}{(n-1)}}$$
где  $\bar{x}$  — среднее значение *x* для всей совокупности, а *n* — размер генеральной совокупности.
- Пустые строки отфильтровываются из *columnName* и не учитываются в вычислениях.
- Если *columnName* содержит менее двух непустых строк, 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 показана формула для вычисляемого столбца, который оценивает стандартное отклонение цены за единицу для каждого продукта в выборке, когда формула используется в таблице

продуктов.

```
= STDEVX.S(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

# STDEVX.P

02.10.2020 • 2 minutes to read

Возвращает среднеквадратичное отклонение всей совокупности.

## Синтаксис

```
STDEVX.P(<table>, <expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).
expression	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).

## Возвращаемое значение

Среднеквадратичное отклонение всей совокупности.

## Remarks

- STDEVX.P вычисляет *expression* для каждой строки таблицы *table* и возвращает стандартное отклонение выражения, предполагая, что эта таблица ссылается на всю совокупность. Если данные в *table* представляют собой выборку из совокупности, для вычисления стандартного отклонения следует использовать функцию STDEV.S.

- STDEVX.P использует следующую формулу:

$$\sqrt{\sum (x - \tilde{x})^2 / n}$$

где  $\tilde{x}$  — среднее значение  $x$  для всей совокупности, а  $n$  — размер генеральной совокупности.

- Пустые строки отфильтровываются из *columnName* и не учитываются в вычислениях.
- Если *columnName* содержит менее двух непустых строк, 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 показана формула для вычисляемого столбца, которая вычисляет стандартное отклонение цены за единицу товара при использовании формулы в таблице *Product*.

```
= STDEVX.P(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

# SQRTPI

02.10.2020 • 2 minutes to read

Возвращает квадратный корень (число \* Пи).

## Синтаксис

```
SQRTPI(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Обязательный элемент. Число, на которое умножается Пи.

## Возвращаемое значение

Возвращает квадратный корень (число \* Пи).

## Пример

ФОРМУЛА	ОПИСАНИЕ	РЕЗУЛЬТАТ
= SQRTPI(1)	Квадратный корень числа Пи.	1,772454
= SQRTPI(2)	Квадратный корень из 2 * Пи.	2,506628

# T.DIST

22.09.2020 • 2 minutes to read

Возвращает левостороннее t-распределение Стьюдента.

## Синтаксис

```
T.DIST(X,Deg_freedom,Cumulative)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
X	Числовое значение, по которому вычисляется распределение.
Deg_freedom	Целое число, указывающее число степеней свободы.
Cumulative	Логическое значение, определяющее форму функции. Если cumulative имеет значение TRUE, функция T.DIST возвращает накопительную функцию распределения, а если FALSE — функцию плотности вероятности.

## Возвращаемое значение

Левостороннее t-распределение Стьюдента.

## Remarks

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

## Пример

```
EVALUATE { T.DIST(60, 1, TRUE) }
```

Возвращает:

[ЗНАЧЕНИЕ]
0,994695326367377

## См. также

[T.DIST.2T](#)  
[T.DIST.2T](#)  
[T.INV](#)  
[T.INV.2t](#)

# T.DIST.2T

22.09.2020 • 2 minutes to read

Возвращает двустороннее t-распределение Стьюдента.

## Синтаксис

```
T.DIST.2T(X,Deg_freedom)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
X	Числовое значение, по которому вычисляется распределение.
Deg_freedom	Целое число, указывающее число степеней свободы.

## Возвращаемое значение

Двустороннее t-распределение Стьюдента.

## Remarks

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

## Пример

```
EVALUATE { T.DIST.2T(1.959999998, 60) }
```

Возвращает

[ЗНАЧЕНИЕ]

0,054644929975921

## См. также

[T.DIST](#)

[T.DIST. RT](#)

[T.INV](#)

[T.INV.2t](#)



# T.DIST.RT

22.09.2020 • 2 minutes to read

Возвращает правостороннее t-распределение Стьюдента.

## Синтаксис

```
T.DIST.RT(X, Deg_freedom)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
X	Числовое значение, по которому вычисляется распределение.
Deg_freedom	Целое число, указывающее число степеней свободы.

## Возвращаемое значение

Правостороннее t-распределение Стьюдента.

## Remarks

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

## Пример

```
EVALUATE { T.DIST.RT(1.959999998, 60) }
```

Возвращает

[ЗНАЧЕНИЕ]

0,0273224649879605

## См. также

[T.DIST](#)

[T.DIST.2T](#)

[T.INV](#)

[T.INV.2t](#)

# T.INV

02.10.2020 • 2 minutes to read

Возвращает левостороннее обратное t-распределение Стьюдента.

## Синтаксис

```
T.INV(Probability,Deg_freedom)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Вероятность	Вероятность, связанная с t-распределением Стьюдента.
Deg_freedom	Число степеней свободы, характеризующих распределение.

## Возвращаемое значение

Левостороннее обратное t-распределение Стьюдента.

## Remarks

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

## Пример

```
EVALUATE { T.INV(0.75, 2) }
```

Возвращает

[ЗНАЧЕНИЕ]
0,816496580927726

## См. также

[T.INV.2T](#)

[T.DIST](#)

[T.DIST.2T](#)

[T.DIST.RT](#)

# T.INV.2T

22.09.2020 • 2 minutes to read

Возвращает двустороннее обратное t-распределение Стьюдента.

## Синтаксис

```
T.INV.2T(Probability,Deg_freedom)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Вероятность	Вероятность, связанная с t-распределением Стьюдента.
Deg_freedom	Число степеней свободы, характеризующих распределение.

## Возвращаемое значение

Двустороннее обратное t-распределение Стьюдента.

## Remarks

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

## Пример

```
EVALUATE { T.INV.2T(0.546449, 60) }
```

Возвращает

[ЗНАЧЕНИЕ]
0,606533075825759

## См. также

[T.INV](#)

[T.DIST](#)

[T.DIST.2T](#)

[T.DIST.RT](#)

# TAN

02.10.2020 • 2 minutes to read

Возвращает тангенс заданного угла.

## Синтаксис

```
TAN(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Обязательный элемент. Угол в радианах, тангенс которого нужно найти.

## Возвращаемое значение

Возвращает тангенс заданного угла.

## Remarks

Если аргумент указан в градусах, умножьте его на  $\text{PI}()/180$  или используйте функцию **RADIANS**, чтобы преобразовать его в радианы.

## Пример

ФОРМУЛА	DESCRIPTION	РЕЗУЛЬТАТ
= TAN(0,785)	Тангенс 0,785 радиан (0,99920)	0,99920
= TAN(45*PI()/180)	Тангенс 45 градусов (1)	1
= TAN(RADIANS(45))	Тангенс 45 градусов (1)	1

# TANH

02.10.2020 • 2 minutes to read

Возвращает гиперболический тангенс числа.

## Синтаксис

TANH(*number*)

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Обязательный элемент. Любое вещественное число.

## Возвращаемое значение

Возвращает гиперболический тангенс числа.

## Remarks

- Формула гиперболического тангенса:

$$\text{TANH}(z) = \frac{\text{SINH}(z)}{\text{COSH}(z)}$$

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

## Пример

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

# VAR.S

02.10.2020 • 2 minutes to read

Возвращает дисперсию выборки из совокупности.

## Синтаксис

```
VAR.S(<columnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX (обычно полное имя). Этот параметр не может быть выражением.

## Возвращаемое значение

Число с дисперсией выборки из совокупности.

## Remarks

- VAR.S предполагает, что столбец ссылается на выборку из совокупности. Если данные представляют собой всю совокупность, вычислите дисперсию с помощью функции VAR.P.
- Функция VAR.S использует следующую формулу:
$$\sum (x - \bar{x})^2 / (n - 1)$$
где  $x$  — среднее значение  $x$  для всей выборки,  
а  $n$  — это размер совокупности.
- Пустые строки отфильтровываются из *columnName* и не учитываются в вычислениях.
- Если *columnName* содержит менее двух непустых строк, 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 показана формула для меры, которая вычисляет дисперсию столбца SalesAmount\_USD, из таблицы InternetSales\_USD по выборке из совокупности.

```
= VAR.S(InternetSales_USD[SalesAmount_USD])
```

# VAR.P

02.10.2020 • 2 minutes to read

Возвращает дисперсию всей совокупности.

## Синтаксис

```
VAR.P(<columnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX (обычно полное имя). Этот параметр не может быть выражением.

## Возвращаемое значение

Число с дисперсией выборки из совокупности.

## Remarks

- VAR.P предполагает, что столбец ссылается на всю совокупность. Если данные представляют собой выборку из совокупности, вычислите дисперсию с помощью функции VAR.S.

- VAR.P использует следующую формулу:

$$\sum (x - \bar{x})^2 / n$$

где  $\bar{x}$  — среднее значение  $x$  для всей совокупности,

а  $n$  — это размер совокупности.

- Пустые строки отфильтровываются из *columnName* и не учитываются в вычислениях.
- Если *columnName* содержит менее двух непустых строк, 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 показана формула для меры, которая оценивает дисперсию столбца SalesAmount\_USD, из таблицы InternetSales\_USD по всей совокупности.

```
= VAR.P(InternetSales_USD[SalesAmount_USD])
```

# VARX.S

02.10.2020 • 2 minutes to read

Возвращает дисперсию выборки из совокупности.

## Синтаксис

```
VARX.S(<table>, <expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Любое выражение DAX, возвращающее таблицу данных.
expression	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).

## Возвращаемое значение

Дисперсия выборки из совокупности.

## Remarks

- Функция VARX.S вычисляет *expression* для каждой строки *table* и возвращает дисперсию *expression*; предполагается, что *table* ссылается на выборку совокупности. Если *table* представляет всю совокупность, для вычисления дисперсии следует использовать функцию VARX.P.

- Функция VAR.S использует следующую формулу:

$$\sum (x - \bar{x})^2 / (n - 1)$$

где  $\bar{x}$  — среднее значение  $x$  для всей выборки,

а  $n$  — это размер совокупности.

- Пустые строки отфильтровываются из *columnName* и не учитываются в вычислениях.
- Если *columnName* содержит менее двух непустых строк, 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 показана формула для вычисляемого столбца, который оценивает дисперсию цены за единицу для каждого продукта в выборке, когда формула используется в таблице продуктов.

```
= VARX.S(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```





# VARX.P

02.10.2020 • 2 minutes to read

Возвращает дисперсию всей совокупности.

## Синтаксис

```
VARX.P(<table>, <expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Любое выражение DAX, возвращающее таблицу данных.
expression	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).

## Возвращаемое значение

Число с дисперсией выборки из совокупности.

## Remarks

- VARX.P вычисляет <expression> для каждой строки таблицы <table> и возвращает дисперсию выражения <expression>, предполагая, что эта таблица <table> ссылается на всю совокупность. Если <table> представляет собой выборку из совокупности, вычислите дисперсию с помощью функции VARX.S.

- VARX.P использует следующую формулу:

$$\sum (x - \tilde{x})^2 / n$$

где  $\tilde{x}$  — среднее значение  $x$  для всей совокупности,

а  $n$  — это размер совокупности.

- Пустые строки отфильтровываются из *columnName* и не учитываются в вычислениях.
- Если *columnName* содержит менее двух непустых строк, 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 показана формула для вычисляемого столбца, которая вычисляет дисперсию цены за единицу товара при использовании формулы в таблице Product

```
= VARX.P(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -  
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

# Функции обработки таблиц

02.10.2020 • 4 minutes to read

Эти функции возвращают таблицу или обрабатывают существующие таблицы.

## В этой категории

ФУНКЦИЯ	ОПИСАНИЕ
<a href="#">ADDCOLUMNS</a>	Добавляет вычисляемые столбцы в заданную таблицу или табличное выражение.
<a href="#">ADDMISSINGITEMS</a>	Добавляет сочетание элементов из нескольких столбцов в таблицу, если они еще не существуют.
<a href="#">CROSSJOIN</a>	Возвращает таблицу, содержащую декартово произведение всех строк из всех таблиц в аргументах.
<a href="#">CURRENTGROUP</a>	Возвращает набор строк из табличного аргумента выражения GROUPBY.
<a href="#">DATATABLE</a>	Предоставляет механизм для объявления встроенного набора значений данных.
<a href="#">DETAILROWS</a>	Вычисляет выражение строк детализации, определенное для меры, и возвращает данные.
<a href="#">DISTINCT (столбец)</a>	Возвращает таблицу с одним столбцом, которая содержит уникальные значения из указанного столбца.
<a href="#">DISTINCT (таблица)</a>	Возвращает таблицу, удаляя повторяющиеся строки из другой таблицы или выражения.
<a href="#">EXCEPT</a>	Возвращает строки одной таблицы, которых нет в другой таблице.
<a href="#">FILTERS</a>	Возвращает таблицу значений, которые напрямую применяются к <i>columnName</i> в качестве фильтров.
<a href="#">GENERATE</a>	Возвращает таблицу с декартовым произведением между каждой строкой в <i>table1</i> и таблицей, полученной в результате вычисления <i>table2</i> в контексте текущей строки из <i>table1</i> .
<a href="#">GENERATEALL</a>	Возвращает таблицу с декартовым произведением между каждой строкой в <i>table1</i> и таблицей, полученной в результате вычисления <i>table2</i> в контексте текущей строки из <i>table1</i> .
<a href="#">GENERATESERIES</a>	Возвращает таблицу с одним столбцом, которая содержит значения арифметического ряда.

ФУНКЦИЯ	ОПИСАНИЕ
GROUPBY	Аналогично SUMMARIZE, GROUPBY не выполняет неявную операцию CALCULATE ни для каких добавляемых ею столбцов расширения.
IGNORE	Изменяет SUMMARIZECOLUMNS, пропуская определенные выражения при вычислении BLANK/NULL.
INTERSECT	Возвращает пересечение строк двух таблиц, сохраняющее дубликаты.
NATURALINNERJOIN	Выполняет внутреннее соединение таблицы с другой таблицей.
NATURALLEFTOUTERJOIN	Выполняет внутреннее соединение таблицы с другой таблицей.
ROLLUP	Изменяет поведение функций SUMMARIZE путем добавления строк свертки к результату для столбцов, определенных параметром groupBy_columnName.
ROLLUPADDISSUBTOTAL	Изменяет поведение SUMMARIZECOLUMNS, добавляя строки свертки или промежуточных итогов к результату на основе столбцов groupBy_columnName.
ROLLUPISSUBTOTAL	Объединяет группы свертки со столбцом, добавленным функцией ROLLUPADDISSUBTOTAL в выражении ADDMISSINGITEMS.
ROLLUPGROUP	Изменяет поведение функций SUMMARIZE и SUMMARIZECOLUMNS путем добавления строк свертки к результату для столбцов, определенных параметром groupBy_columnName.
ROW	Возвращает таблицу с одной строкой, содержащей значения, которые являются результатом выражений, заданных для каждого столбца.
SELECTCOLUMNS	Добавляет вычисляемые столбцы в заданную таблицу или табличное выражение.
SUBSTITUTEWITHINDEX	Возвращает таблицу, которая представляет левое полусоединение двух таблиц, представленных в качестве аргументов.
SUMMARIZE	Возвращает сводную таблицу для запрошенных итоговых значений по набору групп.
SUMMARIZECOLUMNS	Возвращает сводную таблицу для набора групп.
Конструктор таблиц	Возвращает таблицу из одного или нескольких столбцов.
TOPN	Возвращает верхние N строк указанной таблицы.

ФУНКЦИЯ	ОПИСАНИЕ
TREATAS	Применяет результат табличного выражения в качестве фильтров к столбцам несвязанной таблицы.
UNION	Создает объединенную таблицу из пары таблиц.
VALUES	Возвращает таблицу с одним столбцом, содержащую уникальные значения из указанных таблицы или столбца.

# ADDCOLUMNS

22.09.2020 • 2 minutes to read

Добавляет вычисляемые столбцы в заданную таблицу или табличное выражение.

## Синтаксис

```
ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Любое выражение DAX, возвращающее таблицу данных.
name	Имя, присвоенное столбцу, заключенное в двойные кавычки.
expression	Любое выражение DAX, возвращающее скалярное выражение, вычисляемое для каждой строки параметра <i>table</i> .

## Возвращаемое значение

Таблица со всеми ее исходными столбцами и добавленными элементами.

## Remarks

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 категорий продуктов, включающая общие значения продаж через торговых посредников и продажи через Интернет.

```
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, которая будет получена любой функцией, ожидающей получения таблицы:

PRODUCTCATEGORY[ PRODUCTCATEGORY NAME]	PRODUCTCATEGORY[ PRODUCTCATEGORYA LTERNATEKEY]	PRODUCTCATEGORY[ PRODUCTCATEGORYK EY]	[INTERNET SALES]	[ТОВАРООБОРОТ ПОСРЕДНИКОВ]
Bikes	1	1	25107749,77	63084675,04

PRODUCTCATEGORY[ PRODUCTCATEGORY NAME]	PRODUCTCATEGORY[ PRODUCTCATEGORYA LTERNATEKEY]	PRODUCTCATEGORY[ PRODUCTCATEGORYK EY]	[INTERNET SALES]	[ТОВАРООБОРОТ ПОСРЕДНИКОВ]
Components	2	2		11205837,96
Clothing	3	3	306157,5829	1669943,267
Accessories	4	4	640920,1338	534301,9888



# ADDMISSINGITEMS

02.10.2020 • 2 minutes to read

Добавляет строки с пустыми значениями в таблицу, возвращенную выражением [SUMMARIZECOLUMNS](#).

## Синтаксис

```
ADDMISSINGITEMS ( [ <showAll_columnName> [ , <showAll_columnName> [ , ... ] ] ], <table> [ , <groupBy_columnName> [ ,  
[ <filterTable> ] [ , <groupBy_columnName> [ , [ <filterTable> ] [ , ... ] ] ] ] ] )
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
showAll_columnName	(Необязательно) Столбец, для которого возвращаются элементы без данных для используемых мер. Если не указан, возвращаются все столбцы.
table	Таблица SUMMARIZECOLUMNS.
groupBy_columnName	(Необязательно) Столбец, используемый для группировки в указанном табличном аргументе.
filterTable	(Необязательно) Табличное выражение, определяющее возвращаемые строки.

## Возвращаемое значение

Таблица из одного или нескольких столбцов.

## Remarks

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

## С функцией SUMMARIZECOLUMNS

Таблица, возвращаемая выражением [SUMMARIZECOLUMNS](#), будет содержать только строки со значениями. При переносе выражения [SUMMARIZECOLUMNS](#) в выражение ADDMISSINGITEMS также возвращаются строки, содержащие значения без значений.

### Пример

Без выражения ADDMISSINGITEMS выполните следующий запрос:

```
SUMMARIZECOLUMNS(  
    'Sales'[CustomerId],  
    "Total Qty", SUM ( Sales[TotalQty] )  
)
```

Возвращает:

CUSTOMERID	TOTALQTY
Объект	5
B	3
C	3
E	2

С выражением ADDMISSINGITEMS — следующий запрос:

```
EVALUATE
ADDMISSINGITEMS (
  'Sales'[CustomerId],
  SUMMARIZECOLUMNS(
    'Sales'[CustomerId],
    "Total Qty", SUM ( Sales[TotalQty] )
  ),
  'Sales'[CustomerId]
)
```

Возвращает:

CUSTOMERID	TOTALQTY
Объект	5
B	3
C	3
D	
E	2
F	

# CROSSJOIN

22.09.2020 • 2 minutes to read

Возвращает таблицу, содержащую декартово произведение всех строк из всех таблиц в аргументах. Столбцы в новой таблице являются всеми столбцами во всех таблицах аргументов.

## Синтаксис

```
CROSSJOIN(<table>, <table>[, <table>]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
таблица	Любое выражение DAX, возвращающее таблицу данных

## Возвращаемое значение

Таблица, содержащая декартово произведение всех строк из всех таблиц в аргументах.

## Примечания

- Имена столбцов из аргументов *table* должны быть разными во всех таблицах, иначе возвращается ошибка.
- Общее число строк, возвращаемых функцией CROSSJOIN(), равно произведению числа строк из всех таблиц в аргументах; кроме того, общее число столбцов в таблице результатов является суммой количества столбцов во всех таблицах. Например, если TableA содержит rA строк и cA столбцов, TableB содержит rB строк и cB столбцов, а TableC содержит rC строк и cC столбцов, то результирующая таблица содержит  $rA \times rB \times rC$  строк и  $cA + cB + cC$  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 показаны результаты применения функции CROSSJOIN() к двум таблицам: **Цвета** и **Бланк**.

Таблица **Цвета** содержит цвета и узоры:

ЦВЕТ	УЗОР
Красный	Горизонтальные полосы
Зеленый	Вертикальные полосы
Синий	Штриховка

В таблице **Бланк** содержатся шрифты и их стиль отображения:

ШРИФТ	ОТОБРАЖЕНИЕ
Serif	Приподнятый
Sans-Serif	Утопленный

Ниже представлено выражение для создания перекрестного объединения.

```
CROSSJOIN( Colors, Stationery)
```

Если выражение выше используется везде, где ожидается табличное выражение, результаты выражения будут выглядеть следующим образом:

ЦВЕТ	УЗОР	ШРИФТ	ОТОБРАЖЕНИЕ
Красный	Горизонтальные полосы	Serif	Приподнятый
Зеленый	Вертикальные полосы	Serif	Приподнятый
Синий	Штриховка	Serif	Приподнятый
Красный	Горизонтальные полосы	Sans-Serif	Утопленный
Зеленый	Вертикальные полосы	Sans-Serif	Утопленный
Синий	Штриховка	Sans-Serif	Утопленный

# CURRENTGROUP

02.10.2020 • 2 minutes to read

Возвращает набор строк из аргумента "table" выражения [GROUPBY](#), относящихся к текущей строке результата [GROUPBY](#).

## Синтаксис

```
CURRENTGROUP ( )
```

### Параметры

Отсутствуют

## Возвращаемое значение

Строки в табличном аргументе функции [GROUPBY](#), соответствующие одной группе значений аргументов groupBy\_columnName.

## Remarks

- Эту функцию можно использовать только в выражении [GROUPBY](#).
- Функция не принимает никакие аргументы и поддерживается только в качестве первого аргумента для одной из следующих статистических функций: [AVERAGEX](#), [COUNTAX](#), [COUNTX](#), [GEOMEANX](#), [MAXX](#), [MINX](#), [PRODUCTX](#), [STDEVX.S](#), [STDEVX.P](#), [SUMX](#), [VARX.S](#), [VARX.P](#).

## Пример

См. раздел [GROUPBY](#).

# DATATABLE

22.09.2020 • 2 minutes to read

Предоставляет механизм для объявления встроенного набора значений данных.

## Синтаксис

```
DATATABLE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}, {ValueN, ValueN+1...}}...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
ColumnName	Любое выражение DAX, возвращающее таблицу.
DataType	Перечисление, включающее: INTEGER, DOUBLE, STRING, BOOLEAN, CURRENCY, DATETIME
Значение	<p>Один аргумент, использующий синтаксис Excel для одномерной константы массива, вложенный для задания массива массивов. Этот аргумент представляет набор значений данных, которые будут находиться в таблице</p> <p>Например, { {значения в строке 1}, {значения в строке 2}, {значения в строке 3}, и т.д. }</p> <p>Здесь {значения в строке 1} — это набор константных выражений с разделителями-запятыми, а именно сочетание констант с несколькими базовыми функциями, включая DATE, TIME и BLANK, а также оператор сложения между DATE и TIME и унарный минус, чтобы выразить отрицательные значения.</p> <p>Вот ряд допустимых значений: 3, -5, BLANK(), "2009-04-15 02:45:21". Значения не могут ссылаться на что-либо за пределами самого выражения: ни на столбцы, ни на таблицы, связи или что-то еще.</p> <p>Отсутствующее значение будет обрабатываться как BLANK(). Например, следующие записи идентичны: {1,2,BLANK(),4} {1,2,,4}</p>

## Возвращаемое значение

Таблица, объявляющая встроенный набор значений.

## Примечания

- В отличие от DATATABLE, [конструктор таблиц](#) допускает 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

```
= DataTable("Name", STRING,  
            "Region", STRING  
            ,{  
                {" User1", "East"},  
                {" User2", "East"},  
                {" User3", "West"},  
                {" User4", "West"},  
                {" User4", "East"}  
            }  
            )
```

# DETAILROWS

02.10.2020 • 2 minutes to read

Вычисляет выражение строк детализации, определенное для меры, и возвращает данные.

## Синтаксис

```
DETAILROWS([Measure])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Measure	Имя меры.

## Возвращаемое значение

Таблица с данными, возвращенными выражением строк детализации. Если выражение строк детализации не определено, возвращаются данные для таблицы, содержащей меру.



# DISTINCT (столбец)

02.10.2020 • 3 minutes to read

Возвращает таблицу с одним столбцом, которая содержит уникальные значения из указанного столбца. Иными словами, повторяющиеся значения удаляются, и возвращаются только уникальные значения.

## NOTE

Эту функцию нельзя использовать для возвращения значений в ячейку или столбец на листе. Вместо этого функцию DISTINCT можно вложить в формулу, чтобы получить список уникальных значений, которые можно передать в другую функцию для подсчета их количества, суммирования или других операций.

## Синтаксис

```
DISTINCT(<column>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец, из которого должны быть возвращены уникальные значения. Это может быть также выражение, возвращающее столбец.

## Возвращаемое значение

Столбец уникальных значений.

## Remarks

- На результаты функции DISTINCT влияет текущий контекст фильтра. Например, в приведенном ниже примере результаты формулы для создания меры будут меняться при фильтрации таблицы для отображения только определенного региона или периода 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 \(таблица\)](#), которая возвращает таблицу, удаляя повторяющиеся строки из другой таблицы или выражения.

Функция VALUES похожа на функцию DISTINCT. Ее также можно использовать для получения списка уникальных значений, и, как правило, она возвращает те же результаты, что и DISTINCT. Однако в некоторых контекстах функция VALUES возвращает еще одно особое значение. Дополнительные сведения см. в разделе [Функция VALUES](#).

## Пример

Приведенная ниже формула подсчитывает количество уникальных клиентов, разместивших заказы через Интернет. В таблице далее показаны возможные результаты при добавлении формулы в сводную таблицу.

```
= COUNTROWS(DISTINCT(InternetSales_USD[CustomerKey]))
```

Список значений, которые возвращает функция `DISTINCT`, нельзя вставить непосредственно в столбец. Вместо этого результаты функции `DISTINCT` необходимо передать в другую функцию, которая подсчитывает, фильтрует или суммирует значения из списка. Чтобы максимально упростить пример, в этом случае таблица с уникальными значениями была передана в функцию `COUNTROWS`.

УНИКАЛЬНЫЕ ИНТЕРНЕТ- ЗАКАЗЧИКИ	МЕТКИ СТОЛБЦОВ			
Метки строк	Accessories	Bikes	Clothing	Grand Total
2005		1013		1013
2006		2677		2677
2007 г.	6792	4 875	2867	9 309
2008	9 435	5451	4196	11377
Grand Total	15 114	9132	6 852	18484

Кроме того, обратите внимание, что результаты не являются аддитивными. То есть общее число уникальных клиентов в 2007 году не является суммой уникальных клиентов по категориям *Accessories*, *Bikes* и *Clothing* в этом году. Причина в том, что один и тот же клиент может быть учтен в нескольких категориях.

## См. также раздел

[Функции фильтрации](#)

[Функция FILTER](#)

[Функция RELATED](#)

[VALUES, функция](#)

# DISTINCT (таблица)

02.10.2020 • 2 minutes to read

Возвращает таблицу, удаляя повторяющиеся строки из другой таблицы или выражения.

## Синтаксис

```
DISTINCT(<table>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, из которой должны быть возвращены уникальные строки. Это также может быть выражение, результатом которого является таблица.

## Возвращаемое значение

Таблица, содержащая только уникальные строки.

## Связанные функции

Существует другая версия функции DISTINCT, [DISTINCT \(столбец\)](#), которая принимает имя столбца в качестве входного параметра.

## Пример

Рассмотрим следующий запрос:

```
EVALUATE DISTINCT( { (1, "A"), (2, "B"), (1, "A") } )
```

Он возвращает следующую таблицу:

[VALUE1]	[VALUE2]
1	Объект
2	B

## См. также раздел

[Функции фильтрации](#)

[DISTINCT \(столбец\)](#)

[Функция FILTER](#)

[Функция RELATED](#)

[VALUES, функция](#)

# EXCEPT

22.09.2020 • 2 minutes to read

Возвращает строки одной таблицы, которых нет в другой таблице.

## Синтаксис

```
EXCEPT(<table_expression1>, <table_expression2>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Table_expression	Любое выражение DAX, возвращающее таблицу.

## Возвращаемое значение

Таблица, которая содержит строки одной таблицы за вычетом всех строк из другой таблицы.

## Remarks

- Если строка присутствует в обеих таблицах, ни она, ни ее дубликаты не включаются в результирующий набор. Если строка встречается только в table\_expression1, и она, и ее дубликаты будут включены в результирующий набор.
- Имена столбцов в возвращаемой таблице будут соответствовать именам столбцов в table\_expression1.
- Возвращаемая таблица содержит происхождение данных, основанное на столбцах в table\_expression1, независимо от происхождения данных столбцов во второй таблице. Например, если первый столбец первого table\_expression содержит происхождение данных для базового столбца C1 в модели, то функция Except будет уменьшать строки на основе доступности значений в первом столбце второго table\_expression и сохранит происхождение данных в базовом столбце C1 без изменений.
- Две таблицы должны иметь одинаковое число столбцов.
- Столбцы сравниваются в зависимости от позиционирования, а сравнение данных выполняется без приведения типов.
- Возвращаемый набор строк зависит от порядка следования этих двух выражений.
- Возвращаемая таблица не включает 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

States1

ШТАТ
Объект
B
B
B
C
D
D

States2

ШТАТ
B
C
D
D
D
E
E
E

Except(States1, States2)

ШТАТ
Объект

Except(States2, States1)

ШТАТ
E
E
E

# FILTERS

02.10.2020 • 2 minutes to read

Возвращает значения, которые напрямую применяются к *columnName* в качестве фильтров.

## Синтаксис

```
FILTERS(<columnName>)
```

### Параметры

ТЕРМИН	DESCRIPTION
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.

## Возвращаемое значение

Значения, которые напрямую применяются к *columnName* в качестве фильтров.

## Remarks

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 фильтров, примененных к столбцу напрямую.

```
= COUNTROWS(FILTERS(ResellerSales_USD[ProductKey]))
```

В нем определяется, сколько фильтров применено напрямую к столбцу ResellerSales\_USD[ProductKey] в контексте, в котором вычисляется выражение.

# GENERATE

02.10.2020 • 3 minutes to read

Возвращает таблицу с декартовым произведением между каждой строкой в *table1* и таблицей, полученной в результате вычисления *table2* в контексте текущей строки из *table1*.

## Синтаксис

```
GENERATE(<table1>, <table2>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table1	Любое выражение DAX, возвращающее таблицу.
table2	Любое выражение DAX, возвращающее таблицу.

## Возвращаемое значение

Таблица с декартовым произведением между каждой строкой в *table1* и таблицей, полученной в результате вычисления *table2* в контексте текущей строки из *table1*.

## Remarks

- Если вычисление *table2* для текущей строки в *table1* возвращает пустую таблицу, то таблица результатов не будет содержать текущую строку из *table1*. В этом состоит отличие от `GENERATEALL()`, где текущая строка из таблицы *table1* будет включена в результаты, а столбцы, соответствующие *table2*, будут иметь значения NULL для этой строки.
- Все имена столбцов из *table1* и *table2* должны быть разными, иначе 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 пользователю нужна сводная таблица продаж по региону и категории продуктов для посредников, как показано в следующей таблице:

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[ТОВАРООБОРОТ ПОСРЕДНИКОВ]
Европа	Accessories	142 227,27 долл. США
Европа	Bikes	9 970 200,44 долл. США
Европа	Clothing	365 847,63 долл. США

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[ТОВАРООБОРОТ ПОСРЕДНИКОВ]
Европа	Components	2 214 440,19 долл. США
Северная Америка	Accessories	379 305,15 долл. США
Северная Америка	Bikes	52 403 796,85 долл. США
Северная Америка	Clothing	1 281 193,26 долл. США
Северная Америка	Components	8 882 848,05 долл. США
Тихоокеанский регион	Accessories	12 769,57 долл. США
Тихоокеанский регион	Bikes	710 677,75 долл. США
Тихоокеанский регион	Clothing	22 902,38 долл. США
Тихоокеанский регион	Components	108 549,71 долл. США

Ниже показана формула, создающая приведенную ранее таблицу:

```
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, в которой каждая строка соответствует группе территорий, как показано ниже:

SALESTERRITORY[SALESTERRITORYGROUP]
Северная Америка
Европа
Тихоокеанский регион
Н/Д

2. Второй оператор SUMMARIZE

```
SUMMARIZE(ProductCategory, [ProductCategoryName], "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
```

создает таблицу групп категорий с товарооборотом посредника по каждой группе, как показано ниже:

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[ТОВАРООБОРОТ ПОСРЕДНИКОВ]
Bikes	63 084 675,04 долл. США



PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[ТОВАРООБОРОТ ПОСРЕДНИКОВ]
Components	11 205 837,96 долл. США
Clothing	1 669 943,27 долл. США
Accessories	534 301,99 долл. США

3. Тем не менее при вычислении приведенной выше таблицы в контексте каждой строки из таблицы групп территорий для каждой территории будут получены разные результаты.

# GENERATEALL

15.12.2020 • 3 minutes to read

Возвращает таблицу с декартовым произведением между каждой строкой в *table1* и таблицей, полученной в результате вычисления *table2* в контексте текущей строки из *table1*.

## Синтаксис

```
GENERATEALL(<table1>, <table2>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table1	Любое выражение DAX, возвращающее таблицу.
table2	Любое выражение DAX, возвращающее таблицу.

## Возвращаемое значение

Таблица с декартовым произведением между каждой строкой в *table1* и таблицей, полученной в результате вычисления *table2* в контексте текущей строки из *table1*.

## Remarks

- Если вычисление *table2* для текущей строки в *table1* возвращает пустую таблицу, то текущая строка из таблицы *table1* будет включена в результаты, а столбцы, соответствующие *table2*, будут иметь значения NULL для этой строки. В этом заключается отличие от функции GENERATE(), для которой текущая строка из *table1* \*\*\*не\* будет включаться в результаты.
- Все имена столбцов из *table1* и *table2* должны быть разными, иначе 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 пользователю нужна сводная таблица продаж по региону и категории продуктов для посредников, как показано в следующей таблице:

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[ТОВАРООБОРОТ ПОСРЕДНИКОВ]
Европа	Accessories	\$ 142 227,27
Европа	Bikes	\$ 9 970 200,44
Европа	Clothing	\$ 365 847,63

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[ТОВАРООБОРОТ ПОСРЕДНИКОВ]
Европа	Components	\$ 2 214 440,19
Н/Д	Accessories	
Н/Д	Bikes	
Н/Д	Clothing	
Н/Д	Components	
Северная Америка	Accessories	\$ 379 305,15
Северная Америка	Bikes	\$ 52 403 796,85
Северная Америка	Clothing	\$ 1 281 193,26
Северная Америка	Components	\$ 8 882 848,05
Тихоокеанский регион	Accessories	\$ 12 769,57
Тихоокеанский регион	Bikes	\$ 710 677,75
Тихоокеанский регион	Clothing	\$ 22 902,38
Тихоокеанский регион	Components	\$ 108 549,71

Ниже показана формула, создающая приведенную ранее таблицу:

```

GENERATEALL(
SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
,SUMMARIZE(ProductCategory
, [ProductCategoryName]
, "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD])
)
)

```

1. Первая операция SUMMARIZE создает таблицу групп территорий, в которой каждая строка соответствует группе территорий, как показано ниже:

SALESTERRITORY[SALESTERRITORYGROUP]
Северная Америка
Европа
Тихоокеанский регион
Н/Д

2. Вторая операция SUMMARIZE создает таблицу групп категорий с товарооборотом посредника по каждой группе, как показано ниже:

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[ТОВАРООБОРОТ ПОСРЕДНИКОВ]
Bikes	63 084 675,04 долл. США
Components	11 205 837,96 долл. США
Clothing	1 669 943,27 долл. США
Accessories	534 301,99 долл. США

3. Тем не менее при вычислении приведенной выше таблицы в контексте каждой строки из таблицы групп территорий для каждой территории будут получены разные результаты.

# GENERATESERIES

22.09.2020 • 2 minutes to read

Возвращает таблицу с одним столбцом, содержащую значения арифметического ряда, т. е. последовательность значений, в которой каждое следующее отличается от предыдущего на постоянную величину. Возвращаемый столбец имеет имя Value.

## Синтаксис

```
GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
startValue	Начальное значение, используемое при создании последовательности.
endValue	Конечное значение, используемое при создании последовательности.
incrementValue	(Необязательное значение) Значение приращения для последовательности. Если не указано, по умолчанию используется значение 1.

## Возвращаемое значение

Таблица с одним столбцом, которая содержит значения арифметического ряда. Столбец имеет имя Value.

## Remarks

- Если startValue меньше endValue, возвращается пустая таблица.
- Аргумент incrementValue должен иметь положительное значение.
- Последовательность заканчивается значением, 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

Рассмотрим следующий запрос DAX:

```
EVALUATE GENERATESERIES(1, 5)
```

Возвращает следующую таблицу, содержащую один столбец:

[ЗНАЧЕНИЕ]	
1	
2	
3	
4	
5	

## Пример 2

Рассмотрим следующий запрос DAX:

```
EVALUATE GENERATESERIES(1.2, 2.4, 0.4)
```

Возвращает следующую таблицу, содержащую один столбец:

[ЗНАЧЕНИЕ]	
1.2	
1.6	
2	
2.4	

## Пример 3

Рассмотрим следующий запрос DAX:

```
EVALUATE GENERATESERIES(CURRENCY(10), CURRENCY(12.4), CURRENCY(0.5))
```

Возвращает следующую таблицу, содержащую один столбец:

[ЗНАЧЕНИЕ]	
10	
10.5	
11	
11,5	
12	

# GROUPBY

02.10.2020 • 5 minutes to read

Функция GROUPBY аналогична функции [SUMMARIZE](#). Однако GROUPBY не выполняет неявную операцию [CALCULATE](#) ни для каких добавляемых ею столбцов расширения. GROUPBY позволяет использовать новую функцию [CURRENTGROUP](#) внутри агрегатных функций в добавляемых столбцах расширения. GROUPBY используется для выполнения нескольких агрегатов в одном сканировании таблицы.

## Синтаксис

```
GROUPBY (<table> [, <groupBy_columnName> [, <groupBy_columnName> [, ...]] [, <name>, <expression> [, <name>, <expression> [, ...]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
таблица	Любое выражение DAX, возвращающее таблицу данных.
groupBy_columnName	Имя существующего столбца в таблице (или связанной таблице), по которому группируются данные. Этот параметр не может быть выражением.
name	Имя, присвоенное новому столбцу, который добавляется в список столбцов GroupBy, заключенное в двойные кавычки.
expression	Одна из статистических функций X с первым аргументом — CURRENTGROUP(). Полный список поддерживаемых статистических функций X см. в разделе "С функцией CURRENTGROUP" ниже.

## Возвращаемое значение

Таблица с выбранными столбцами для аргументов groupBy\_columnName и столбцов расширения, назначенных с помощью аргументов имени.

## Remarks

- Функция GROUPBY выполняет следующие действия.
  - Начинает с указанной таблицы (и всех связанных таблиц в направлении "к одному").
  - Создает группирование, используя все столбцы GroupBy (которые должны существовать в таблице из шага 1).
  - Каждая группа является одной строкой в результате, однако представляет набор строк в исходной таблице.
  - Для каждой группы оценивает добавляемые столбцы расширения. В отличие от функции SUMMARIZE, подразумеваемая операция CALCULATE не выполняется, а группа не помещается в контекст фильтра.

- Каждый столбец, для которого определяется имя, должен иметь соответствующее выражение; в противном случае возвращается ошибка. Первый аргумент name определяет имя столбца в результатах. Второй аргумент expression определяет вычисление, выполняемое для получения значения каждой строки в этом столбце.
- Значение groupBy\_columnName должно находиться в таблице или связанной таблице.
- Каждое имя должно быть заключено в двойные кавычки.
- Функция группирует выбранный набор строк в набор сводных строк по значениям одного столбца groupBy\_columnName или нескольких. Для каждой группы возвращается одна строка.
- GROUPBY в основном используется для выполнения агрегатов по промежуточным результатам из выражений таблиц DAX. Для эффективных агрегатов по физическим таблицам в модели рассмотрите возможность использования функции [SUMMARIZECOLUMNS](#) или [SUMMARIZE](#).
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## С функцией CURRENTGROUP

[CURRENTGROUP](#) можно использовать только в выражении, определяющем столбец расширения внутри функции GROUPBY. В результате [CURRENTGROUP](#) возвращает набор строк из аргумента "table" функции GROUPBY, относящихся к текущей строке результата GROUPBY. Функция [CURRENTGROUP](#) не принимает никакие аргументы и поддерживается только в качестве первого аргумента для одной из следующих статистических функций: [AVERAGEX](#), [COUNTX](#), [COUNTX](#), [GEOMEANX](#), [MAXX](#), [MINX](#), [PRODUCTX](#), [STDEVX.S](#), [STDEVX.P](#), [SUMX](#), [VARX.S](#), [VARX.P](#).

### Пример

В следующем примере сначала вычисляется общий объем продаж, сгруппированный по странам и категориям продуктов, по физическим таблицам с помощью функции [SUMMARIZECOLUMNS](#). Затем функция GROUPBY используется для проверки промежуточного результата первого шага, чтобы найти максимальный объем продаж в каждой стране по категориям продуктов.

```
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])
)
```

## См. также

[Функция SUMMARIZE](#)

[Функция SUMMARIZECOLUMNS](#)



# IGNORE

02.10.2020 • 2 minutes to read

Служит для изменения поведения функции [SUMMARIZECOLUMNS](#) — она будет пропускать определенные выражения из вычисления BLANK/NULL. Строки, для которых все выражения, не использующие IGNORE, возвращают BLANK/NULL, будут исключены, независимо от того, является ли результатом вычисления выражений, использующих IGNORE, значение в BLANK/NULL. Эту функцию можно использовать только в выражении [SUMMARIZECOLUMNS](#).

## Синтаксис

```
IGNORE(<expression>)
```

С функцией SUMMARIZECOLUMNS

```
SUMMARIZECOLUMNS(<groupBy_columnName>[, <groupBy_columnName >]..., [<filterTable>]...[, <name>, IGNORE(...)]...)
```

## Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Любое выражение DAX, возвращающее одиночное значение (не таблицу).

## Возвращаемое значение

Функция не возвращает значение.

## Remarks

IGNORE можно использовать в качестве аргумента выражения только для функции [SUMMARIZECOLUMNS](#).

## Пример

См. [SUMMARIZECOLUMNS](#).

# INTERSECT

22.09.2020 • 2 minutes to read

Возвращает пересечение строк двух таблиц, сохраняющее дубликаты.

## Синтаксис

```
INTERSECT(<table_expression1>, <table_expression2>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Table_expression	Любое выражение DAX, возвращающее таблицу.

## Возвращаемое значение

Таблица, содержащая все строки в table\_expression1, которые также находятся в table\_expression2

## Исключения

## Remarks

- Функция INTERSECT не является коммутативной. В целом Intersect (T1, T2) будет иметь результирующий набор, отличный от Intersect (T2, T1).
- Дублирующиеся 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 в возвращаемой таблице будут соответствовать именам столбцов в table\_expression1.
- Возвращаемая таблица содержит происхождение данных, основанное на столбцах в table\_expression1, независимо от происхождения данных столбцов во второй таблице. Например, если первый столбец первого table\_expression содержит происхождение данных для базового столбца C1 в модели, то INTERSECT будет уменьшать строки на основе пересечения по первому столбцу второго table\_expression и сохранит происхождение данных в базовом столбце C1 без изменений.
- Столбцы сравниваются в зависимости от позиционирования, а сравнение данных выполняется без приведения типов.
- Возвращаемая таблица не включает 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

States1

ШТАТ
Объект
Объект
B
B
B
C
D
D

States2

ШТАТ
B
C
D
D
D
E

Intersect(States1, States2)

ШТАТ
B
B
B
C
D
D

Intersect(States2, States1)

<b>ШТАТ</b>
B
C
D
D
D

# NATURALINNERJOIN

22.09.2020 • 2 minutes to read

Выполняет внутреннее соединение таблицы с другой таблицей. Таблицы соединяются по общим столбцам (по имени) в двух таблицах. Если две таблицы не имеют общих имен столбцов, возвращается ошибка.

## Синтаксис

```
NATURALINNERJOIN(<leftJoinTable>, <rightJoinTable>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
leftJoinTable	Табличное выражение, определяющее таблицу в левой части объединения.
rightJoinTable	Табличное выражение, определяющее таблицу в правой части объединения.

## Возвращаемое значение

Таблица, содержащая только строки, значения которых в указанных столбцах общего типа присутствуют в обеих таблицах. Возвращаемая таблица будет содержать общие столбцы из левой таблицы и другие столбцы из обеих таблиц.

## Remarks

- Порядок сортировки результатов не гарантируется.
- Объединяемые столбцы должны иметь один и тот же тип данных в обеих таблицах.
- Объединяются только столбцы из одной исходной таблицы (с одинаковым происхождением данных). Например, Products[ProductID], WebSales[ProductID], StoreSales[ProductID] со связями "многие к одному" между таблицами WebSales и StoreSales и таблицей Products, основанной на столбце ProductID в таблицах WebSales и StoreSales, объединяются в [ProductID].
- При объединении используются более строгие семантики сравнения. Приведение типов отсутствует; например, значение 1 не равно 1,0.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

# NATURALLEFTOUTERJOIN

22.09.2020 • 2 minutes to read

Выполняет внутреннее соединение таблицы с другой таблицей. Таблицы соединяются по общим столбцам (по имени) в двух таблицах. Если две таблицы не имеют общих имен столбцов, возвращается ошибка.

## Синтаксис

```
NATURALLEFTOUTERJOIN(<leftJoinTable>, <rightJoinTable>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
leftJoinTable	Табличное выражение, определяющее таблицу в левой части объединения.
rightJoinTable	Табличное выражение, определяющее таблицу в правой части объединения.

## Возвращаемое значение

Таблица, содержащая только строки из rightJoinTable, для которых значения в указанных столбцах общего типа также присутствуют в leftJoinTable. Возвращаемая таблица будет содержать общие столбцы из левой таблицы и другие столбцы из обеих таблиц.

## Remarks

- Порядок сортировки результатов не гарантируется.
- Объединяемые столбцы должны иметь один и тот же тип данных в обеих таблицах.
- Объединяются только столбцы из одной исходной таблицы (с одинаковым происхождением данных). Например, Products[ProductID], WebSales[ProductID], StoreSales[ProductID] со связями "многие к одному" между таблицами WebSales и StoreSales и таблицей Products, основанной на столбце ProductID в таблицах WebSales и StoreSales, объединяются в [ProductID].
- При объединении используются более строгие семантики сравнения. Приведение типов отсутствует; например, значение 1 не равно 1,0.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

# ROLLUP

02.10.2020 • 2 minutes to read

Изменяет поведение функции [SUMMARIZE](#) путем добавления строк свертки к результату для столбцов, определенных параметром `groupBy_columnName`. Эту функцию можно использовать только в выражении [SUMMARIZE](#).

## Синтаксис

```
ROLLUP ( <groupBy_columnName> [, <groupBy_columnName> [, ... ] ] )
```

С функцией [SUMMARIZE](#)

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, ROLLUP(<groupBy_columnName>[, <groupBy_columnName>...])][, <name>, <expression>]...)
```

## Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
<code>groupBy_columnName</code>	Полное имя существующего столбца или функция <code>ROLLUPGROUP</code> , которые используются для создания сводных групп на основе найденных значений. Этот параметр не может быть выражением.

## Возвращаемое значение

Эта функция не возвращает значение. Она указывает только набор столбцов для промежуточного итога.

## Remarks

Эту функцию можно использовать только в выражении [SUMMARIZE](#).

## Пример

См. [SUMMARIZE](#).

# ROLLUPADDISSUBTOTAL

02.10.2020 • 2 minutes to read

Изменяет поведение функции [SUMMARIZECOLUMNS](#), добавляя строки свертки или промежуточных итогов к результату на основе столбцов `groupBy_columnName`. Эту функцию можно использовать только в выражении [SUMMARIZECOLUMNS](#).

## Синтаксис

```
ROLLUPADDISSUBTOTAL ( [<grandtotalFilter>], <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, <groupBy_columnName>, <name> [, [<groupLevelFilter>] [, ... ] ] ] ] )
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
<code>grandtotalFilter</code>	(Необязательно) Фильтр, применяемый к общему итоговому уровню.
<code>groupBy_columnName</code>	Имя существующего столбца, используемого для создания сводных групп на основе значений, найденных в нем. Не может быть выражением.
<code>name</code>	Имя столбца <code>ISSUBTOTAL</code> . Значения столбца вычисляются с помощью функции <code>ISSUBTOTAL</code> .
<code>groupLevelFilter</code>	(Необязательно) Фильтр, применяемый к текущему уровню.

## Возвращаемое значение

Функция не возвращает значение.

## Remarks

None

## Пример

См. [SUMMARIZECOLUMNS](#).



# ROLLUPGROUP

02.10.2020 • 2 minutes to read

Изменяет поведение функций [SUMMARIZE](#) и [SUMMARIZECOLUMNS](#) путем добавления строк свертки к результату для столбцов, определенных параметром `groupBy_columnName`. Эту функцию можно использовать только в выражении [SUMMARIZE](#) или [SUMMARIZECOLUMNS](#).

## Синтаксис

```
ROLLUPGROUP ( <groupBy_columnName> [ , <groupBy_columnName> [ , ... ] ] )
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
<code>groupBy_columnName</code>	Полное имя существующего столбца или функция ROLLUPGROUP, которые используются для создания сводных групп на основе найденных значений. Этот параметр не может быть выражением.

## Возвращаемое значение

Эта функция не возвращает значение. Она помечает набор столбцов как единую группу во время вычисления промежуточного итога с помощью функции [ROLLUP](#) или [ROLLUPADDISSUBTOTAL](#).

## Remarks

Функцию ROLLUPGROUP можно использовать только в качестве аргумента `groupBy_columnName` для функций [ROLLUP](#), [ROLLUPADDISSUBTOTAL](#) или [ROLLUPISSUBTOTAL](#).

## Пример

См. [SUMMARIZE](#) и [SUMMARIZECOLUMNS](#).

# ROLLUPISSUBTOTAL

02.10.2020 • 2 minutes to read

Объединяет группы свертки со столбцом, добавленным функцией [ROLLUPADDISSUBTOTAL](#). Эту функцию можно использовать только в выражении [ADDMISSINGITEMS](#).

## Синтаксис

```
ROLLUPISSUBTOTAL ( [<grandTotalFilter>], <groupBy_columnName>, <isSubtotal_columnName> [,  
[<groupLevelFilter>] [, <groupBy_columnName>, <isSubtotal_columnName> [, [<groupLevelFilter>] [, ... ] ] ] ] )
```

## Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
grandTotalFilter	(Необязательно) Фильтр, применяемый к общему итоговому 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, применяемый к текущему уровню.

## Возвращаемое значение

None

## Remarks

Эту функцию можно использовать только в выражении [ADDMISSINGITEMS](#).

# Функция ROW

22.09.2020 • 2 minutes to read

Возвращает таблицу с одной строкой, содержащей значения, которые являются результатом выражений, заданных для каждого столбца.

## Синтаксис

```
ROW(<name>, <expression>[[,<name>, <expression>]...])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
name	Имя, присвоенное столбцу, заключенное в двойные кавычки.
expression	Любое выражение DAX, возвращающее одиночное скалярное значение для заполнения. <i>name</i> .

## Возвращаемое значение

Таблица с одной строкой

## Remarks

- Аргументы должны всегда поступать парами *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 таблица, состоящая из одной строки, с общими данными о продажах через Интернет и через посредников.

```
ROW("Internet Total Sales (USD)", SUM(InternetSales_USD[SalesAmount_USD]),  
    "Resellers Total Sales (USD)", SUM(ResellerSales_USD[SalesAmount_USD]))
```

# SELECTCOLUMNS

02.10.2020 • 2 minutes to read

Добавляет вычисляемые столбцы в заданную таблицу или табличное выражение.

## Синтаксис

```
SELECTCOLUMNS(<table>, <name>, <scalar_expression> [, <name>, <scalar_expression>]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Любое выражение DAX, возвращающее таблицу.
name	Имя, присвоенное столбцу, заключенное в двойные кавычки.
expression	Любое выражение, возвращающее скалярное значение, такое как ссылка на столбец, целое число или строковое значение.

## Возвращаемое значение

Таблица с тем же количеством строк, что и у таблицы, указанной в качестве первого аргумента. Возвращаемая таблица содержит по одному столбцу для каждой пары аргументов <name>, <scalar\_expression>, и каждое выражение вычисляется в контексте строки из заданного аргумента <table>.

## Remarks

SELECTCOLUMNS имеет ту же сигнатуру, что и ADDCOLUMNS, и то же поведение, но начинается не с указанного аргумента <table>, а с пустой таблицы до добавления 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 Info:

COUNTRY	ШТАТ	COUNT	ИТОГ
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
США	CA	5	500

COUNTRY	ШТАТ	COUNT	ИТОГ
США	WA	10	900

```
SELECTCOLUMNS(Info, "StateCountry", [State]&"", "&[Country])
```

Возвращает:

STATECOUNTRY
IND, JK
IND, MH
IND, WB
USA, CA
USA, WA

# SUBSTITUTEWITHINDEX

22.09.2020 • 4 minutes to read

Возвращает таблицу, которая представляет левое полусоединение двух таблиц, представленных в качестве аргументов. Полусоединение выполняется с помощью общих столбцов, определяемых общими именами столбцов и типами данных. Соединяемые столбцы заменяются одним столбцом в возвращаемой таблице, который имеет целочисленный тип и содержит индекс. Индекс является ссылкой на правую таблицу соединения в соответствии с заданным порядком сортировки.

Столбцы в правой (второй) таблице, которые не существуют в левой (первой) указанной таблице, не включаются в возвращаемую таблицу и не используются для объединения.

Индекс начинается с 0 и увеличивается на единицу для каждой дополнительной строки в правой (второй) таблице соединения. Индекс основан на порядке сортировки, указанном для правой (второй) таблицы соединения.

## Синтаксис

```
SUBSTITUTEWITHINDEX(<table>, <indexColumnName>, <indexColumnsTable>, [<orderBy_expression>, [<order>]][, <orderBy_expression>, [<order>]]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
таблица	Таблица, подлежащая фильтрации путем выполнения левого полусоединения с таблицей, указанной в качестве третьего аргумента (indexColumnsTable). Это таблица с левой стороны левого полусоединения, поэтому возвращаемая таблица содержит те же столбцы, что и эта таблица, за исключением того, что все общие столбцы этих двух таблиц будут заменены в возвращаемой таблице одним индексным столбцом.
indexColumnName	Строка, указывающая имя столбца индекса, который заменяет все общие столбцы в двух таблицах, указанных в качестве аргументов этой функции.
indexColumnsTable	Вторая таблица для левого полусоединения. Это таблица с правой стороны левого полусоединения. Функция будет возвращать только значения, представленные в этой таблице. Кроме того, столбцы этой таблицы (отбираемые на основе имен столбцов) будут заменены одним столбцом индекса в таблице, возвращаемой этой функцией.

ТЕРМИН	ОПРЕДЕЛЕНИЕ
orderBy_expression	Любое выражение DAX, в котором значение результата используется для указания требуемого порядка сортировки таблицы indexColumnsTable для формирования правильных значений индекса. Порядок сортировки, заданный для таблицы indexColumnsTable, определяет индекс каждой строки в таблице; этот индекс используется в таблице, возвращаемой для представления сочетаний значений в indexColumnsTable так, как они появляются в таблице, указанной в качестве первого аргумента этой функции.
order	<p>(необязательно) Значение, указывающее, как сортировать значения orderBy_expression, — по возрастанию или по убыванию:</p> <p>Значение: <b>DESC</b>. Альтернативное значение: 0 (ноль) или <b>FALSE</b>. Сортировка значений в порядке убывания orderBy_expression. Это значение по умолчанию, если параметр order опущен.</p> <p>Значение: <b>ASC</b>. Альтернативное значение: 1/<b>TRUE</b>. Ранжирование значений в порядке возрастания orderBy_expression.</p>

## Возвращаемое значение

Таблица, которая включает только те значения, которые присутствуют в таблице indexColumnsTable, и имеющая столбец индекса вместо всех имеющихся столбцов (по имени) в таблице indexColumnsTable.

## Примечания

- 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).

# SUMMARIZE

02.10.2020 • 8 minutes to read

Возвращает сводную таблицу для запрошенных итоговых значений по набору групп.

## Синтаксис

```
SUMMARIZE (<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, <name>, <expression>]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
таблица	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, присвоенное общему или сводному столбцу, заключенное в двойные кавычки.
expression	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).

## Возвращаемое значение

Таблица с выбранными столбцами для аргументов *groupBy\_columnName* и сводных столбцов, созданных с помощью аргументов имени.

## Примечания

- Каждый столбец, для которого определяется имя, должен иметь соответствующее выражение; в противном случае возвращается ошибка. Первый аргумент *name* определяет имя столбца в результатах. Второй аргумент *expression* определяет вычисление, выполняемое для получения значения каждой строки в этом столбце.
- Значение *groupBy\_columnName* должно находиться в *таблице* или в связанной таблице для *таблицы*.
- Каждое имя должно быть заключено в двойные кавычки.
- Функция группирует выбранный набор строк в набор сводных строк по значениям одного столбца *groupBy\_columnName* или нескольких. 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 продаж торгового посредника, сгруппированных по календарному году и имени категории продукта. Эта таблица результатов позволяет вести анализ продаж через торгового посредника по годам и по категориям продуктов.

```
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, которая будет получена любой функцией, ожидающей получения таблицы:

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (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	Запчасти	2008052,706	39,9266
2005	Запчасти	574256,9865	0
2006	Запчасти	3428213,05	948,7674
2007	Запчасти	5195315,216	4226,0444
2008	Экипировка	366507,844	4151,1235
2005	Экипировка	31851,1628	90,9593
2006	Экипировка	455730,9729	4233,039
2007	Экипировка	815853,2868	12489,3835
2008	Принадлежности	153299,924	865,5945
2005	Принадлежности	18594,4782	4,293
2006	Принадлежности	86612,7463	1061,4872
2007	Принадлежности	275794,8403	4756,6546

# С функцией ROLLUP

Добавление синтаксиса **ROLLUP** изменяет поведение функции SUMMARIZE — к результату добавляются строки сводки на основе столбцов groupBy\_columnName. Функцию **ROLLUP** можно использовать только в выражении SUMMARIZE.

## Пример

В следующем примере строки свертки добавляются в столбцы Group-By в вызове функции SUMMARIZE:

```
SUMMARIZE(ResellerSales_USD
    , ROLLUP( DateTime[CalendarYear], ProductCategory[ProductCategoryName])
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

Он возвращает следующую таблицу:

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
2008	Bikes	12968255,42	36167,6592
2005	Bikes	6958251,043	4231,1621
2006	Bikes	18901351,08	178175,8399
2007 г.	Bikes	24256817,5	276065,992
2008	Компоненты	2008052,706	39,9266
2005	Компоненты	574256,9865	0
2006	Компоненты	3428213,05	948,7674
2007 г.	Компоненты	5195315,216	4226,0444
2008	Clothing	366507,844	4151,1235
2005	Clothing	31851,1628	90,9593
2006	Clothing	455730,9729	4233,039
2007 г.	Clothing	815853,2868	12489,3835
2008	Accessories	153299,924	865,5945
2005	Accessories	18594,4782	4,293
2006	Accessories	86612,7463	1061,4872
2007 г.	Accessories	275794,8403	4756,6546
2008		15496115,89	41224,3038

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
2005		7582953,67	4326,4144
2006		22871907,85	184419,1335
2007 г.		30543780,84	297538,0745
		76494758,25	527507,9262

## С функцией ROLLUPGROUP

Добавление **ROLLUPGROUP** внутри синтаксиса **ROLLUP** позволяет предотвратить появление частичного промежуточного итога в строках свертки. Функцию **ROLLUPGROUP** можно использовать только в выражениях **ROLLUP**, **ROLLUPADDISSUBTOTAL** или **ROLLUPISSUBTOTAL**.

### Пример

В следующем примере показан только общий итог по всем годам и категориям без промежуточного итога каждого года со всеми категориями:

```
SUMMARIZE(ResellerSales_USD
    , ROLLUP(ROLLUPGROUP( DateTime[CalendarYear], ProductCategory[ProductCategoryName]))
    , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
    , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

Он возвращает следующую таблицу:

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
2008	Bikes	12968255,42	36167,6592
2005	Bikes	6958251,043	4231,1621
2006	Bikes	18901351,08	178175,8399
2007 г.	Bikes	24256817,5	276065,992
2008	Компоненты	2008052,706	39,9266
2005	Компоненты	574256,9865	0
2006	Компоненты	3428213,05	948,7674
2007 г.	Компоненты	5195315,216	4226,0444
2008	Clothing	366507,844	4151,1235
2005	Clothing	31851,1628	90,9593

DATETIME[CALENDAR YEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
2006	Clothing	455730,9729	4233,039
2007 г.	Clothing	815853,2868	12489,3835
2008	Accessories	153299,924	865,5945
2005	Accessories	18594,4782	4,293
2006	Accessories	86612,7463	1061,4872
2007 г.	Accessories	275794,8403	4756,6546
		76494758,25	527507,9262

## С функцией ISSUBTOTAL

С помощью [ISSUBTOTAL](#) можно создать другой столбец в выражении SUMMARIZE, который содержит значение true, если строка содержит значения промежуточных итогов для столбца, указанного в качестве аргумента [ISSUBTOTAL](#); в противном случае столбец содержит значение false. Функцию [ISSUBTOTAL](#) можно использовать только в выражении SUMMARIZE.

### Пример

В следующем примере создается столбец [ISSUBTOTAL](#) для каждого из столбцов [ROLLUP](#) в заданном вызове функции SUMMARIZE:

```
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])
)
```

Он возвращает следующую таблицу:

[IS SUB TOTAL FOR DATETIMECALENDAR YEAR]	[IS SUB TOTAL FOR PRODUCTCATEGORYNAME]	DATETIME[CALENDAR YEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
FALSE	FALSE				
FALSE	FALSE	2008	Bikes	12968255,42	36167,6592
FALSE	FALSE	2005	Bikes	6958251,043	4231,1621
FALSE	FALSE	2006	Bikes	18901351,08	178175,8399
FALSE	FALSE	2007 г.	Bikes	24256817,5	276065,992
FALSE	FALSE	2008	Компоненты	2008052,706	39,9266

[IS SUB TOTAL FOR DATETIME[CALENDAR]YEAR]	[IS SUB TOTAL FOR PRODUCTCATEGORYNAME]	DATETIME[CALENDAR]YEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[SALES AMOUNT (USD)]	[DISCOUNT AMOUNT (USD)]
FALSE	FALSE	2005	Компоненты	574256,9865	0
FALSE	FALSE	2006	Компоненты	3428213,05	948,7674
FALSE	FALSE	2007 г.	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 г.	Clothing	815853,2868	12489,3835
FALSE	FALSE	2008	Accessories	153299,924	865,5945
FALSE	FALSE	2005	Accessories	18594,4782	4,293
FALSE	FALSE	2006	Accessories	86612,7463	1061,4872
FALSE	FALSE	2007 г.	Accessories	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 г.		30543780,84	297538,0745
TRUE	TRUE			76494758,25	527507,9262

См. также

[SUMMARIZECOLUMNS](#)

# SUMMARIZECOLUMNS

02.10.2020 • 9 minutes to read

Возвращает сводную таблицу для набора групп.

## Синтаксис

```
SUMMARIZECOLUMNS( <groupBy_columnName> [, <groupBy_columnName >]..., [<filterTable>]...[, <name>,  
<expression>]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
groupBy_columnName	Полная ссылка на столбец (таблица [столбец]) в базовой таблице, для которой различающиеся значения включаются в возвращаемую таблицу. Каждый столбец groupBy_columnName перекрестно соединен (в различных таблицах) или существует автоматически (в той же таблице) с последующими указанными столбцами.
filterTable	Табличное выражение, которое добавляется в контекст фильтра всех столбцов, указанных как аргументы groupBy_columnName. Значения в таблице фильтров используются для фильтрации перед выполнением перекрестного соединения или автоматического существования.
name	Строка, представляющая имя столбца, который используется для последующего указанного выражения.
expression	Любое выражение DAX, возвращающее одиночное значение (не таблицу).

## Возвращаемое значение

Таблица, содержащая сочетания значений из указанных столбцов, в зависимости от указанных параметров группирования. В возвращаемую таблицу включаются только строки, для которых по крайней мере одно из представленных выражений возвращает непустое значение. Если все выражения оценивают строку как BLANK/NULL, такая строка не включается в возвращаемую таблицу.

## Remarks

- Эта функция не гарантирует порядок сортировки результатов.
- Столбец можно указать в параметре groupBy\_columnName только один раз. Например, следующая формула недействительна.

```
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:

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category],
    FILTER('Customer', 'Customer' [First Name] = "Alicia")
)
```

В этом запросе столбцы groupBy без меры не содержат столбцов из выражения FILTER (например, из таблицы Customer). Фильтр не применяется к столбцам groupBy. Таблицы Sales Territory и Customer могут быть косвенно связаны с таблицей фактов Reseller sales. Поскольку они не связаны напрямую, выражение фильтра является холостой командой и столбцы groupBy не затрагиваются.

Однако с этим запросом:

```
SUMMARIZECOLUMNS (
    'Sales Territory'[Category], 'Customer' [Education],
    FILTER('Customer', 'Customer'[First Name] = "Alicia")
)
```

Столбцы groupBy содержат столбец, который затронут фильтром, и этот фильтр применяется к результатам groupBy.

## С параметром IGNORE

Синтаксис **IGNORE** можно использовать для изменения поведения функции SUMMARIZECOLUMNS — она будет пропускать определенные выражения из вычисления BLANK/NULL. Строки, для которых все выражения, не использующие **IGNORE**, возвращают BLANK/NULL, будут исключены, независимо от того, является ли результатом вычисления выражений, использующих **IGNORE**, значение в BLANK/NULL. Функцию **IGNORE** можно использовать только в выражении SUMMARIZECOLUMNS.

### Пример

```
SUMMARIZECOLUMNS(
    Sales[CustomerId], "Total Qty",
    IGNORE( SUM( Sales[Qty] ) ),
    "BlankIfTotalQtyIsNot3", IF( SUM( Sales[Qty] )=3, 3 )
)
```

В результате получается сводный столбец Sales[CustomerId], в котором создается промежуточный итог для всех клиентов в заданной группе. Без параметра **IGNORE** результат будет следующим:

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
Объект	5	

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
B	3	3
C	3	3

С параметром **IGNORE**

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
B	3	3
C	3	3

Все выражения пропускаются.

```
SUMMARIZECOLUMNS(
    Sales[CustomerId], "Blank",
    IGNORE( Blank() ), "BlankIfTotalQtyIsNot5",
    IGNORE( IF( SUM( Sales[Qty] )=5, 5 ) )
)
```

Несмотря на то, что оба выражения возвращают пустые значения для некоторых строк, они включаются, поскольку нет игнорируемых выражений, возвращающих пустое значение.

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
Объект		5
B		
C		

## С функцией NONVISUAL

Функция **NONVISUAL** помечает фильтр значений в функции SUMMARIZECOLUMNS как не влияющий на значения мер, но применяется только к столбцам группировки. Функцию **NONVISUAL** можно использовать только в выражении SUMMARIZECOLUMNS.

### Пример

```
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, где [Visual Total Sales] — это сумма за все годы:



DIMDATE[CALENDARYEAR]	[SALES]	[VISUAL TOTAL SALES]
2007 г.	9 791 060,30	29 358 677,22
2008	9 770 899,74	29 358 677,22

Один и тот же запрос *без* функции NONVISUAL для сравнения:

```
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, где [Visual Total Sales] — это сумма за два выбранных года:

DIMDATE[CALENDARYEAR]	[SALES]	[VISUAL TOTAL SALES]
2007 г.	9 791 060,30	19 561 960,04
2008	9 770 899,74	19 561 960,04

## С функцией ROLLUPADDISSUBTOTAL

Добавление синтаксиса [ROLLUPADDISSUBTOTAL](#) изменяет поведение функции SUMMARIZECOLUMNS — к результату добавляются строки сводки или промежуточных итогов на основе столбцов groupBy\_columnName. Функцию [ROLLUPADDISSUBTOTAL](#) можно использовать только в выражении SUMMARIZECOLUMNS.

### Пример с одним промежуточным итогом

```
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]
```

Он возвращает следующую таблицу:

КАТЕГОРИЯ	ПОДКАТЕГОРИЯ	ISCATEGORYSUBTOTAL	ISSUBCATEGORYSUBTOTAL	TOTAL QTY
		Да	Да	60398
Accessories		False	Да	36092
Accessories	Велосипедные стойки	False	False	328
Bikes	Горные велосипеды	False	False	4970
Clothing		False	Да	9101

### Пример с несколькими промежуточными итогами

```
SUMMARIZECOLUMNS (
    Regions[State], ROLLUPADDISSUBTOTAL ( Sales[CustomerId], "IsCustomerSubtotal" ),
    ROLLUPADDISSUBTOTAL ( Sales[Date], "IsDateSubtotal"), "Total Qty", SUM( Sales[Qty] )
)
```

Продажи группируются по штату, по клиентам, по датам, с промежуточными итогами для 1. Продажи по штату, по датам 2. Продажи по штату, по клиентам 3. Сводка как по клиентам, так и по датам, ведущим к продажам по штату.

Он возвращает следующую таблицу:

CUSTOMERID	ISCUSTOMERSUBTOTAL	СОСТОЯНИЕ	TOTAL QTY	ДАТА	ISDATESUBTOTAL
Объект	FALSE	WA	5	7/10/2014	
B	FALSE	WA	1	7/10/2014	
B	FALSE	WA	2	7/11/2014	
C	FALSE	ИЛИ	2	7/10/2014	
C	FALSE	ИЛИ	1	7/11/2014	
	true	WA	6	7/10/2014	
	true	WA	2	7/11/2014	
	true	ИЛИ	2	7/10/2014	
	true	ИЛИ	1	7/11/2014	
Объект	FALSE	WA	5		true
B	FALSE	WA	3		true
C	FALSE	OR	3		TRUE

CUSTOMERID	ISCUSTOMERS UBTOTAL	СОСТОЯНИЕ	TOTAL QTY	ДАТА	ISDATESUBTOT AL
	TRUE	WA	8		TRUE
	TRUE	ИЛИ	3		true

## С функцией ROLLUPGROUP

Как и в случае с функцией [SUMMARIZE](#), функцию [ROLLUPGROUP](#) можно использовать вместе с функцией [ROLLUPADDISSUBTOTAL](#) для указания того, какие сводные группы и детализации (промежуточные итоги) следует включить путем уменьшения количества возвращаемых строк промежуточных итогов. Функцию [ROLLUPGROUP](#) можно использовать только в выражении [SUMMARIZECOLUMNS](#) или [SUMMARIZE](#).

### Пример с несколькими промежуточными итогами

```
SUMMARIZECOLUMNS(
    ROLLUPADDISSUBTOTAL( Sales[CustomerId], "IsCustomerSubtotal" ),
    ROLLUPADDISSUBTOTAL(ROLLUPGROUP(Regions[City], Regions[State]), "IsCityStateSubtotal"), "Total
Qty", SUM( Sales[Qty] )
)
```

Группировка по городу и штату, однако со сведением при вычислении промежуточных итогов возвращается следующая таблица.

СОСТОЯНИЕ	CUSTOMERID	ISCUSTOMERS UBTOTAL	TOTAL QTY	CITY	ISCITYSTATESU BTOTAL
WA	Объект	FALSE	2	Бельвью	FALSE
WA	B	FALSE	2	Бельвью	FALSE
WA	Объект	FALSE	3	Redmond	FALSE
WA	B	FALSE	1	Redmond	FALSE
ИЛИ	C	FALSE	3	Портленд	FALSE
WA		true	4	Бельвью	FALSE
WA		true	4	Redmond	FALSE
ИЛИ		true	3	Портленд	FALSE
	Объект	FALSE	5		FALSE
	B	FALSE	3		true
	C	FALSE	3		TRUE
		TRUE	11		true

См. также

[SUMMARIZE](#)

# Конструктор таблиц

02.10.2020 • 2 minutes to read

Возвращает таблицу из одного или нескольких столбцов.

## Синтаксис

```
{ <scalarExpr1>, <scalarExpr2>, ... }  
{ ( <scalarExpr1>, <scalarExpr2>, ... ), ( <scalarExpr1>, <scalarExpr2>, ... ), ... }
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
scalarExprN	Любое выражение DAX, которое возвращает скалярное значение.

## Возвращаемое значение

Таблица из одного или нескольких столбцов. Если имеется только один столбец, то имя столбца имеет значение. При наличии N столбцов, где  $N > 1$ , имена столбцов слева направо равны Value1, Value2, ..., ValueN.

## Remarks

- Первый синтаксис возвращает таблицу из одного столбца. Второй синтаксис возвращает таблицу из одного или нескольких столбцов.
- Число скалярных выражений должно быть одинаковым для всех строк.
- Если типы данных значений для столбца различаются в разных строках, все значения преобразуются в общий тип данных.

## Пример 1

Следующие запросы DAX:

```
EVALUATE { 1, 2, 3 }
```

и

```
EVALUATE { (1), (2), (3) }
```

Возвращают следующую таблицу, содержащую один столбец:

[ЗНАЧЕНИЕ]
1

[ЗНАЧЕНИЕ]
2
3

## Пример 2

Рассмотрим следующий запрос 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:

[VALUE1]	[VALUE2]	[VALUE3]	[VALUE4]
1.5	01.01.2017	199,99	Объект
2.5	02.01.2017	249,99	B
3,5	03.01.2017	299,99	C

## Пример 3

Рассмотрим следующий запрос DAX:

```
EVALUATE { 1, DATE(2017, 1, 1), TRUE, "A" }
```

Возвращает следующую таблицу из одного столбца строкового типа данных:

[ЗНАЧЕНИЕ]
1
01.01.2017
TRUE
Объект

# TOPN

02.10.2020 • 2 minutes to read

Возвращает верхние N строк указанной таблицы.

## Синтаксис

```
TOPN(<n_value>, <table>, <orderBy_expression>, [<order>[, <orderBy_expression>, [<order>]]...])
```

### Параметры

**Количество возвращаемых строк.** Это любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).

Сведения о том, когда количество возвращаемых строк может быть больше *n\_value*, см. в разделе "Примечания".

Сведения о том, когда возвращается пустая таблица, см. в разделе "Примечания".

**table** Любое выражение DAX, возвращающее таблицу данных, из которой нужно извлечь первые "n" строк.

### orderBy\_expression

Любое выражение DAX, в котором значение результата используется для сортировки таблицы и вычисляется для каждой строки *table*.

**order** (необязательно) Значение, указывающее, как сортировать значения *orderBy\_expression*, — по возрастанию или по убыванию:

VALUE	АЛЬТЕРНАТИВНОЕ ЗНАЧЕНИЕ	ОПИСАНИЕ
0 (ноль)	FALSE	Сортировка значений <i>order_by</i> в порядке убывания.  Это значение по умолчанию, если параметр <i>order</i> опущен.
1	true	Ранжирование <i>order_by</i> в порядке возрастания.

## Возвращаемое значение

Таблица, содержащая первые N строк таблицы *table*, или пустая таблица, если *n\_value* имеет значение 0 (ноль) или меньше. Строки не обязательно сортируются в определенном порядке.

## Remarks

- Если в N-ной строке таблицы имеется связь значений *order\_by*, то возвращаются все связанные строки. Затем при наличии связей в N-ной строке функция может вернуть более N строк.
- Если значение *n\_value* равно 0 (нулю) или меньше, то TOPN возвращает пустую таблицу.
- Функция 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

В следующем примере создается мера с 10 товарами с наибольшим объемом продаж.

```
= SUMX(TOPN(10, SUMMARIZE(Product, [ProductKey], "TotalSales",  
SUMX(RELATED(InternetSales_USD[SalesAmount_USD]), InternetSales_USD[SalesAmount_USD]) +  
SUMX(RELATED(ResellerSales_USD[SalesAmount_USD]), ResellerSales_USD[SalesAmount_USD]))
```



# TREATAS

22.09.2020 • 2 minutes to read

Применяет результат табличного выражения в качестве фильтров к столбцам несвязанной таблицы.

## Синтаксис

```
TREATAS(table_expression, <column>[, <column>[, <column>[,...]]] )
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table_expression	Выражение, результатом которого является таблица.
гистограмма	Один существующий столбец или несколько. Этот параметр не может быть выражением.

## Возвращаемое значение

Таблица, содержащая все строки в столбцах, которые также находятся в table\_expression.

## Примечания

- Указанное число столбцов должно соответствовать числу столбцов в табличном выражении; столбцы должны стоять в том же порядке.
- Если значение, возвращаемое в табличном выражении, не существует в столбце, оно игнорируется. Например, TREATAS({"Red", "Green", "Yellow"}, DimProduct[Color]) задает фильтр для столбца DimProduct[Color] с тремя значениями: "Red", "Green" и "Yellow". Если "Yellow" не существует в DimProduct[Color], то действующие фильтры будут иметь значения "Red" и "Green".
- Лучше всего использовать эту функцию, если между таблицами не существует связи. При наличии между таблицами множества связей попробуйте вместо этого использовать [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 по "Bikes", "Seats", "Tires" к DimProduct1[ProductCategory], то для DimProduct2[ProductCategory] применяется тот же фильтр: "Bikes", "Seats", "Tires".

```
CALCULATE(  
    SUM(Sales[Amount]),  
    TREATAS(VALUES(DimProduct1[ProductCategory]), DimProduct2[ProductCategory])  
)
```

См. также:

[INTERSECT](#)

[FILTER](#)

[USERRELATIONSHIP](#)

# UNION

02.10.2020 • 2 minutes to read

Создает объединенную таблицу из пары таблиц.

## Синтаксис

```
UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table_expression	Любое выражение DAX, возвращающее таблицу.

## Возвращаемое значение

Таблица, содержащая все строки из каждого из двух табличных выражений.

## Remarks

- Две таблицы должны иметь одинаковое число столбцов.
- Столбцы объединяются по положению в соответствующих таблицах.
- Имена столбцов в возвращаемой таблице будут соответствовать именам столбцов в table\_expression1.
- Дублирующиеся строки сохраняются.
- Возвращаемая таблица включает происхождение данных, где это возможно. Например, если первый столбец каждого параметра table\_expression имеет происхождение данных для того же базового столбца C1 в модели, первый столбец в результате выполнения функции UNION будет иметь происхождение данных для C1. Однако если объединенные столбцы имеют происхождение данных для разных базовых столбцов или имеется столбец расширения, то результирующий столбец в UNION не будет содержать происхождение данных.
- Если типы данных различаются, 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 таблиц USAInventory и INDInventory в одну таблицу:

```
UNION(UsaInventory, IndInventory)
```

USAINVENTORY

COUNTRY	ШТАТ	COUNT	ИТОГ
США	CA	5	500
США	WA	10	900

INDINVENTORY

COUNTRY	ШТАТ	COUNT	ИТОГ
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

Возвращаемая таблица

COUNTRY	ШТАТ	COUNT	ИТОГ
США	CA	5	500
США	WA	10	900
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

# VALUES

02.10.2020 • 5 minutes to read

Если входной параметр является именем столбца, функция возвращает таблицу с одним столбцом, которая содержит различные значения из указанного столбца. Повторяющиеся значения удаляются; возвращаются только уникальные значения. Можно добавить значение BLANK. Если входной параметр является именем таблицы, функция возвращает строки из указанной таблицы. Дублирующиеся строки сохраняются. Можно добавить строку BLANK.

## NOTE

Данную функцию не удастся использовать для возвращения значений в ячейку или столбец на листе. Вместо этого ее можно использовать как промежуточную функцию, вложенную в формулу, чтобы получить список уникальных значений, которые можно подсчитать или использовать для фильтрации или суммирования других значений.

## Синтаксис

```
VALUES(<TableNameOrColumnName>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
TableName или ColumnName	Столбец, из которого должны возвращаться уникальные значения, или таблица, из которой должны возвращаться строки.

## Возвращаемое значение

Если входной параметр является именем столбца, то возвращается таблица с одним столбцом. Если входной параметр является именем таблицы, то возвращается таблица с теми же столбцами.

## Remarks

- При использовании функции VALUES в контексте, который был отфильтрован, уникальные значения, возвращаемые функцией VALUES, зависят от фильтра. Например, если выполняется фильтрация по региону и возвращается список значений для City, список будет включать только города в регионах, разрешенных фильтром. Чтобы вернуть все города независимо от существующих фильтров, необходимо использовать функцию ALL для удаления фильтров из таблицы. Во втором примере демонстрируется использование функции ALL совместно с функцией VALUES.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Связанные функции

В большинстве случаев, когда аргумент является именем столбца, результаты функции VALUES идентичны значениям функции DISTINCT. Обе функции удаляют повторяющиеся значения и возвращают список

возможных значений в указанном столбце. Однако функция VALUES также может возвращать пустое значение. Это пустое значение полезно в тех случаях, когда выполняется поиск уникальных значений из связанной таблицы, но значение, используемое в связи, отсутствует в одной таблице. В терминологии баз данных это называется нарушением целостности данных. Такие несоответствия в данных могут возникать, когда одна таблица обновляется, а связанная таблица — нет.

Если аргумент является именем таблицы, результат функции VALUES возвращает все строки в указанной таблице плюс пустую строку, если нарушена целостность данных. Функция DISTINCT удаляет повторяющиеся строки и возвращает уникальные строки в указанной таблице.

**NOTE**

Функция DISTINCT допускает, чтобы имя столбца или любое допустимое табличное выражение было аргументом, тогда как функция VALUES принимает в качестве аргумента только имя столбца или имя таблицы.

В следующей таблице приведены несоответствия между данными, которые могут возникнуть в двух связанных таблицах, если нарушена целостность данных.

ТАБЛИЦА MYORDERS	ТАБЛИЦА MYSALES
June 1	June 1 sales
June 2	June 2 sales
(даты заказов не указаны)	June 3 sales

Если для возврата списка дат из сводной таблицы, содержащей эти таблицы, была использована функция DISTINCT, возвращаются только две даты. Однако при использовании функции VALUES возвращаются две даты плюс дополнительный пустой элемент. Кроме того, любая строка из таблицы MySales, которая не имеет соответствующей даты в таблице MyOrders, будет сопоставлена с этим неизвестным элементом.

## Пример

Следующая формула подсчитывает количество уникальных счетов (заказов на продажу) и возвращает следующие результаты при использовании в отчете, который содержит имена категорий продуктов:

```
= COUNTROWS(VALUES('InternetSales_USD'[SalesOrderNumber]))
```

**Результаты**

МЕТКИ СТРОК	КОЛИЧЕСТВО СЧЕТОВ
Accessories	18 208
Bikes	15 205
Clothing	7461
Grand Total	27 659

См. также раздел

Функция FILTER  
Функция COUNTROWS  
Функции фильтрации

# Текстовые функции

21.09.2020 • 3 minutes to read

Выражения анализа данных (DAX) включают набор текстовых функций, основанных на библиотеке строковых функций в Excel, но которые были изменены для работы с таблицами и столбцами в табличных моделях. В этом разделе описываются текстовые функции, доступные в языке DAX.

## В этой категории

КОМПОНЕНТ	DESCRIPTION
<a href="#">COMBINEVALUES</a>	объединяет две текстовые строки в одну.
<a href="#">CONCATENATE</a>	Объединяет две текстовые строки в одну.
<a href="#">CONCATENATEX</a>	Объединяет результат выражения, вычисляемого для каждой строки в таблице.
<a href="#">EXACT</a>	Сравнивает две текстовые строки и возвращает значение TRUE, если они идентичны, и FALSE в противном случае.
<a href="#">FIND</a>	Возвращает начальную позицию одной текстовой строки в другой текстовой строке.
<a href="#">FIXED</a>	Округляет число до указанного десятичного разряда и возвращает результат в виде текста.
<a href="#">FORMAT</a>	Преобразует значение в текст в соответствии с указанным форматом.
<a href="#">LEFT</a>	Возвращает указанное количество символов с начала текстовой строки.
<a href="#">LEN</a>	Возвращает число символов в текстовой строке.
<a href="#">LOWER</a>	Преобразует все буквы в текстовой строке в нижний регистр.
<a href="#">MID</a>	Возвращает строку символов из середины текстовой строки по заданной начальной позиции и длине.
<a href="#">REPLACE</a>	Функция REPLACE заменяет часть текстовой строки на основе указанного числа символов другой текстовой строкой.
<a href="#">REPT</a>	Повторяет текст заданное число раз.
<a href="#">RIGHT</a>	Возвращает последний символ или символы в текстовой строке на основе указанного количества символов.



КОМПОНЕНТ	DESCRIPTION
SEARCH	Возвращает номер символа, на котором были в первый раз найдены определенный символ или текстовая строка; читается слева направо.
SUBSTITUTE	Заменяет существующий текст новым текстом в текстовой строке.
TRIM	Удаляет все пробелы из текста, за исключением одиночных пробелов между словами.
UNICHAR	Возвращает символ Юникода, на который ссылается числовое значение.
UNICODE	Возвращает число (код), соответствующее первому знаку в текстовой строке.
UPPER	Преобразует все буквы в текстовой строке в прописные.
VALUE	Преобразует текстовую строку, представляющую число, в числовой формат.

# COMBINEVALUES

21.09.2020 • 3 minutes to read

объединяет две текстовые строки в одну. Эта функция предназначена в первую очередь для обеспечения поддержки связей между несколькими столбцами в моделях DirectQuery. Подробные сведения см. в разделе **Примечания**.

## Синтаксис

```
COMBINEVALUES(<delimiter>, <expression>, <expression>[, <expression>]...)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
разделитель	Разделитель, используемый при сцепке. Значение должно быть константой.
expression	Выражение DAX, значение которого будет присоединено к текстовой строке.

## Возвращаемое значение

Объединенная строка.

## Remarks

- Функция COMBINEVALUES предполагает, но не проверяет, что если входные значения различаются, выходные строки также отличаются. Исходя из этого предположения, когда COMBINEVALUES используется для создания вычисляемых столбцов, чтобы создать связь, соединяющую несколько столбцов из двух таблиц DirectQuery, при выполнении запроса создается оптимизированное условие соединения. 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:

```
Table1[CalcColumn] = COMBINEVALUES(",", Table1[Column1], Table1[Column2])
```

и

```
Table2[CalcColumn] = COMBINEVALUES(",", Table2[Column1], Table2[Column2]),
```

а затем создать связь между сущностями Table1[CalcColumn] и Table2[CalcColumn]. В отличие от других функций и операторов DAX, которые буквально переводятся в соответствующие операторы и функции SQL, приведенная выше связь создает предикат соединения SQL следующим образом:

```
(Table1.Column1 = Table2.Column1 OR Table1.Column1 IS NULL AND Table2.Column1 IS NULL)
```

и

```
(Table1.Column2 = Table2.Column2 OR Table1.Column2 IS NULL AND Table2.Column2 IS NULL) .
```

- Предикат соединения может обеспечить более высокую производительность запросов, чем в случае использования сложных операторов и функций SQL.

- Функция COMBINEVALUES полагается на пользователей в выборе соответствующего разделителя, чтобы уникальные сочетания входных значений создавали разные выходные строки, но не проверяет, истинно ли это допущение. Например, если пользователь выбирает " | " в качестве разделителя, но одна строка в Table1 содержит Table1[Column1] = " | " и Table2 [Column2] = " " , тогда как другая строка в Table2 содержит Table2[Column1] = " " и Table2[Column2] = " | " , два сцепленных выхода будут одинаковыми " | | " , и кажется, что эти две строки будут соответствовать друг другу в ходе операции соединения. Две строки не соединяются друг с другом, если обе таблицы находятся в одном источнике DirectQuery, хотя они соединяются друг с другом, если обе таблицы импортируются.

## Пример

Рассмотрим следующий запрос DAX:

```
EVALUATE DISTINCT(SELECTCOLUMNS(DimDate, "Month", COMBINEVALUES(", ", [MonthName], [CalendarYear])))
```

Он возвращает следующую таблицу, содержащую один столбец:

[MONTH]
January,2007
February,2007
March,2007
April,2007
May,2007
June,2007
July,2007
August,2007
September,2007
October,2007
November,2007
December,2007
January,2008
January,2008
February,2008
March,2008
April,2008

[MONTH]
May,2008
June,2008
July,2008
August,2008
September,2008
October,2008
November,2008
December,2008

# CONCATENATE

02.10.2020 • 4 minutes to read

Объединяет две текстовые строки в одну.

## Синтаксис

```
CONCATENATE(<text1>, <text2>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text1, text2	Текстовые строки, сцепляемые в одну текстовую строку. Строки могут содержать текст или числа.  Также можно использовать ссылки на столбцы.

## Возвращаемое значение

Объединенная строка.

## Примечания

- Функция CONCATENATE сцепляет две текстовые строки в одну. Соединяемые элементы могут быть текстовыми, численными или логическими значениями, представленными в виде текста, или сочетанием этих элементов. Можно также использовать ссылку на столбец, если столбец содержит соответствующие значения.
- Функция CONCATENATE в DAX принимает только два аргумента, тогда как функция CONCATENATE в Excel принимает до 255 аргументов. Если необходимо выполнить сцепку над несколькими выражениями, можно создать ряд вычислений или лучше использовать оператор сцепки ( & ), чтобы объединить их в более простое выражение.
- Если вы хотите использовать текстовые строки напрямую, а не указывать ссылку на столбец, каждую строку необходимо заключить 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: Сцепка литералов

В примере формулы новое строковое значение создается путем сцепки двух строковых значений, указанных в качестве аргументов.

```
= CONCATENATE("Hello ", "World")
```

## Пример: Сцепка строк в столбцах

Пример формулы возвращает полное имя клиента, указанное в телефонной книге. Обратите внимание, что в качестве второго аргумента используется вложенная функция. Это один из способов сцепки нескольких строк, если есть более двух значений, которые необходимо использовать в качестве аргументов.

```
= CONCATENATE(Customer[LastName], CONCATENATE(", ", Customer[FirstName]))
```

## Пример: Условная сцепка строк в столбцах

Пример формулы создает в таблице Customer новый вычисляемый столбец с полным именем клиента в виде сочетания имени, среднего инициала и фамилии. Если среднее имя отсутствует, фамилия стоит сразу после имени. Если среднее имя есть, то используется только его инициал.

```
= CONCATENATE( [FirstName]&" ", CONCATENATE( IF( LEN([MiddleName])>1, LEFT([MiddleName],1)&" ", ""), [LastName]))
```

В этой формуле используются вложенные функции CONCATENATE и IF вместе с оператором-амперсандом ( & ) для условного сцепления трех строковых значений и добавления пробелов в качестве разделителей.

## Пример: Сцепка столбцов с различными типами данных

В следующем примере показано, как сцеплять значения в столбцах, имеющих различные типы данных. Если сцепляемое значение является числовым, оно будет неявно преобразовано в текст. Если оба значения являются числовыми, то оба значения будут приведены к тексту и сцеплены, как если бы они были строками.

ОПИСАНИЕ ПРОДУКТА	СОКРАЩЕННОЕ НАЗВАНИЕ ПРОДУКТА (ПЕРВЫЙ СТОЛБЕЦ СОСТАВНОГО КЛЮЧА)	АРТИКУЛ ПРОДУКТА (ВТОРОЙ СТОЛБЕЦ СОСТАВНОГО КЛЮЧА)	НОВЫЙ СОЗДАННЫЙ КЛЮЧЕВОЙ СТОЛБЕЦ
Горные велосипеды	MTN	40	MTN40
Горные велосипеды	MTN	42	MTN42

```
= CONCATENATE('Products'[Product abbreviation], 'Products'[Product number])
```

Функция CONCATENATE в DAX принимает только два аргумента, тогда как функция CONCATENATE в Excel принимает до 255 аргументов. Если необходимо добавить дополнительные аргументы, можно использовать оператор-амперсанд (&). Например, следующая формула выдает результаты, MTN-40 и MTN-42.

```
= [Product abbreviation] & "-" & [Product number]
```

## См. также

[Текстовые функции](#)

# CONCATENATEX

02.10.2020 • 2 minutes to read

Объединяет результат выражения, вычисляемого для каждой строки в таблице.

## Синтаксис

```
CONCATENATEX(<table>, <expression>, [delimiter])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Таблица, содержащая строки, для которых будет вычисляться выражение.
expression	Вычисляемое выражение для каждой строки таблицы.
разделитель	(необязательно) Разделитель, используемый при объединении.

## Возвращаемое значение

Текстовая строка.

## Remarks

- Функция принимает в качестве первого аргумента таблицу или выражение, возвращающее таблицу. Вторым аргументом является столбец, содержащий значения, которые необходимо сцепить, или выражение, возвращающее значение.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Таблица Employees

FIRSTNAME	LASTNAME
Алан	Брюэр
Michael	Blythe

Следующая формула:

```
CONCATENATEX(Employees, [FirstName] & " " & [LastName], ",")
```

Возвращает:

"Алан Брюэр, Майкл Блайт"



# EXACT

21.09.2020 • 2 minutes to read

Сравнивает две текстовые строки и возвращает значение TRUE, если они идентичны, и FALSE в противном случае. Функция EXACT учитывает регистр символов, но при этом игнорирует различия в форматировании. С помощью функции EXACT можно проверять текст, вводимый в документ.

## Синтаксис

```
EXACT(<text1>,<text2>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text1	Первая текстовая строка или столбец, содержащий текст.
text2	Вторая текстовая строка или столбец, содержащий текст.

## Возвращаемое значение

Значение true или false. (логическое значение)

## Пример

Следующая формула сравнивает значения Column1 и Column2 для текущей строки, возвращая TRUE в случае их равенства и FALSE, если они различаются.

```
= EXACT([Column1],[Column2])
```

## См. также раздел

[Текстовые функции](#)

# FIND

21.09.2020 • 2 minutes to read

Возвращает начальную позицию одной текстовой строки в другой текстовой строке. Функция FIND учитывает регистр символов.

## Синтаксис

```
FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
find_text	Искомый текст. Для совпадения с первым символом в <b>within_text</b> (тексте с искомым текстом) используйте двойные кавычки (пустой текст).
within_text	Текст, содержащий искомый текст.
start_num	(Необязательный аргумент) Символ, с которого начинается поиск; если не указан, <b>start_num</b> = 1. Первый символ в <b>within_text</b> — это символ номер 1.
NotFoundValue	(необязательно) Значение, которое возвращается в случае, если операции не удалось найти соответствующую подстроку: как правило, это значение 0, -1 или BLANK().

## Возвращаемое значение

Число, указывающее на начальную позицию искомой текстовой строки.

## Remarks

- В Microsoft Excel предусмотрено несколько версий функции FIND для работы с однобайтовыми и двухбайтовыми кодировками. Тем не менее в DAX используется кодировка Юникод, и каждый символ обрабатывается одинаково. В связи с этим вам не нужно выбирать версию функции, соответствующую типу символа.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).
- Функция FIND не поддерживает подстановочные знаки. Если вам нужно использовать подстановочные знаки, выбирайте функцию [SEARCH](#).

## Пример

Следующая формула находит позицию первой буквы модели продукта (BMX) в строке, содержащей его описание.

```
= FIND("BMX","line of BMX racing goods")
```

См. также

[Текстовые функции](#)

# FIXED

02.10.2020 • 2 minutes to read

Округляет число до указанного десятичного разряда и возвращает результат в виде текста. Вы можете указать, будет ли возвращаемый результат содержать запятые.

## Синтаксис

```
FIXED(<number>, <decimals>, <no_commas>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Число, которое требуется округлить и преобразовать в текст, или столбец, содержащий число.
decimals	(Необязательный аргумент) Количество разрядов справа от десятичного разделителя; по умолчанию 2.
no_commas	(Необязательный аргумент) Логическое значение: значение 1 показывает, что в возвращаемом тексте не выводятся запятые; значение 0 (по умолчанию) показывает, что в возвращаемом тексте выводятся запятые.

## Возвращаемое значение

Число в текстовом представлении.

## Remarks

- Если параметр **decimals** имеет отрицательное значение, **number** округляется слева от десятичного разделителя.
- Если параметр **decimals** не указан, по умолчанию ему присваивается значение 2.
- Если параметр **no\_commas** равен 0 или не указан, в возвращаемый текст включаются запятые.
- Основное различие между форматированием ячейки с числом с использованием команды и напрямую с использованием функции **FIXED** заключается в том, что функция **FIXED** преобразует результат в текстовый вид. Если используется команда из меню форматирования, число сохраняется в числовом виде.
- 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 получается числовое значение из текущей строки столбца **PctCost** и возвращается его текстовое представление с 4 десятичными разрядами без запятых.

```
= FIXED([PctCost],3,1)
```

Числовые значения могут иметь не более 15 значащих цифр. Десятичные значения могут иметь размер до 127.

## См. также

[Текстовые функции](#)

[Математические и тригонометрические функции](#)

# FORMAT

15.12.2020 • 24 minutes to read

Преобразует значение в текст в соответствии с указанным форматом.

## Синтаксис

```
FORMAT(<value>, <format_string>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
value	Значение или выражение, при вычислении которого возвращается одиночное значение.
format_string	Строка, содержащая шаблон форматирования.

## Возвращаемое значение

Строка, содержащая value в формате, определенном с помощью format\_string.

### NOTE

Если value имеет значение BLANK, функция возвращает пустую строку.

Если format\_string — BLANK, значение форматируется с использованием формата "Общий числовой формат" или "Общий формат даты" (в зависимости от типа данных value).

## Remarks

- Во всех стандартных строках формата при форматировании результата используется текущий языковой стандарт пользователя.
- Функция FORMAT (DAX) поддерживает в качестве аргумента строки формата, основанные на строках формата Visual Basic (OLE-автоматизация), а не на строках формата, используемых в .NET Framework. В связи с этим, если аргумент не соответствует какой-либо определенной строке, может возвращаться непредвиденный результат или ошибка. Например, не поддерживается применение сокращения "p" для обозначения процентов. Строки, которые указываются в качестве аргумента функции FORMAT и при этом не входят в список стандартных строк формата, обрабатываются как часть пользовательской строки формата или как строковый литерал.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Примеры

```
= 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")
```

Он возвращает следующее:

12345,67. "Общее число" отображает число без форматирования.

\$12345,67. "Валюта" отображает число в формате валюты в соответствии с вашим языковым стандартом. В примере ниже показан формат валют по умолчанию для США.

12345,67. "Фиксированный" отображает не менее одной цифры слева и двух цифр справа от десятичного разделителя.

12 345,67. "Стандартный" отображает не менее одной цифры слева и двух цифр справа от десятичного разделителя и включает разделитель групп разрядов. В примере ниже показан формат числа по умолчанию для США.

1234567,00 % . "Процент" отображает число в процентах (умноженное на 100) с форматированием и знаком процента справа от числа, разделенного одним пробелом.

1.23E+04. "Научный" отображает число в экспоненциальном представлении с двумя десятичными цифрами.

## Предопределенные числовые форматы

В аргументе `format_string` можно указать следующие предопределенные числовые форматы:

ФОРМАТ	ОПИСАНИЕ
"General Number"	Отображает число без разделителей групп разрядов.
"Currency"	Отображает число с разделителями групп разрядов при необходимости. Отображает две цифры справа от десятичного разделителя. Вывод основан на параметрах локали системы.
"Fixed"	Отображает не менее одной цифры слева и две цифры справа от десятичного разделителя.
"Standard"	Отображает число с разделителями групп разрядов. Не менее одной цифры слева и двух цифр справа от десятичного разделителя.
"Percent"	Отображает число, умноженное на 100, со знаком процента (%), добавленным справа. Всегда отображает две цифры справа от десятичного разделителя.
"Scientific"	Использует стандартное научное представление, предоставляя две значащие цифры.
"Yes/No"	Отображает No, если число равно 0; в противном случае отображает Yes.

ФОРМАТ	ОПИСАНИЕ
"True/False"	Отображает False, если число равно 0; в противном случае отображает True.
"On/Off"	Отображает Off, если число равно 0; в противном случае отображает On.

## Настраиваемые числовые форматы

Настраиваемое выражение формата для чисел может содержать от одного до трех разделов, разделенных точками с запятой. Если аргумент строки формата содержит один из именованных числовых форматов, допускается использование только одного раздела.

ПРИ ИСПОЛЬЗОВАНИИ	В РЕЗУЛЬТАТЕ
Только один раздел	Формат распространяется на все значения.
Две секции	Первый раздел распространяется на неотрицательные значения, а второй — на отрицательные.
Три секции	Первый раздел распространяется на положительные значения, второй — на отрицательные, третий — на нулевые.

"\$#,##0;(\$#,##0)"
---------------------

Если вы добавите символы точки с запятой, между которыми ничего нет, то отсутствующий раздел будет определен с использованием формата положительного значения. Например, в следующем выражении формата положительные и отрицательные значения форматируются в первом разделе, а для нулевых значений выводится "Ноль".

"\$#,##0"
-----------

Если вы добавите символы точки с запятой, между которыми ничего нет, то отсутствующий раздел будет отображаться с использованием формата положительного значения.

### Символы настраиваемых числовых форматов

В аргументе `format_string` можно указать следующие символы настраиваемых числовых форматов:

СИМВОЛ	ОПИСАНИЕ
Нет	Отображает число без форматирования.



СИМВОЛ	ОПИСАНИЕ
(0)	<p>Заполнитель цифр. Отображает цифру или ноль. Если выражение имеет цифру в позиции, где в строке форматирования стоит 0, она отобразится. В противном случае в этой позиции отобразится ноль. Если число имеет меньше цифр, чем нулей (по обе стороны от десятичного разделителя) в выражении формата, отобразятся ноли в начале или конце. Если число имеет больше цифр справа от десятичного разделителя, чем нулей справа от десятичного разделителя в выражении формата, округлите число до такого количества знаков после запятой, которое равно количеству нулей. Если число имеет больше цифр слева от десятичного разделителя, чем нулей слева от десятичного разделителя в выражении формата, отобразите дополнительные цифры без изменений.</p>
#	<p>Заполнитель цифр. Отображает цифру или ничего. Если выражение имеет цифру в позиции, где знак # указан в строке форматирования, она отобразится. В противном случае в этой позиции ничего не отобразится. Этот символ аналогичен заполнителю цифры 0, за исключением того, что ноли в начале и конце не отображаются, если число имеет такое же или меньшее количество цифр, чем число символов # по обе стороны от десятичного разделителя в выражении формата.</p>
.	<p>Заполнитель десятичных чисел. В некоторых языковых стандартах в качестве десятичного разделителя используется запятая. Десятичный заполнитель определяет, сколько цифр отображается слева и справа от десятичного разделителя. Если выражение формата содержит только символы решетки слева от этого символа, числа меньше 1 начинаются с десятичного разделителя. Чтобы в начале дробных чисел отобразить ноль, используйте 0 в качестве заполнителя первой цифры слева от десятичного разделителя. Фактический символ, используемый в качестве десятичного заполнителя в форматированных выходных данных, зависит от числового формата, распознаваемого системой.</p>
%	<p>Заполнитель процентов. Выражение умножается на 100. Символ процента ( % ) вставляется в позицию, где он отображается в строке форматирования.</p>

СИМВОЛ	ОПИСАНИЕ
(,)	Разделитель групп разрядов. В некоторых языковых стандартах в качестве разделителя групп разрядов используется запятая. Разделитель групп разрядов отделяет тысячи от сотен в числе, состоящем из четырех или более знаков слева от десятичного разделителя. Стандартное использование разделителя групп разрядов указывается, если формат содержит разделитель групп разрядов, окруженный заполнителями цифр (0 или #). Два смежных разделителя групп разрядов или один разделитель групп разрядов непосредственно слева от десятичного разделителя (независимо от того, указан ли десятичный знак) означают, что "число необходимо масштабировать, разделив его на 1000, а затем при необходимости округлив". Например, вы можете использовать строку форматирования "#0,," , чтобы представить 100 миллионов как 100. Числа меньше одного миллиона отображаются как 0. Два смежных разделителя групп разрядов в любой позиции, кроме непосредственно места слева от десятичного разделителя, рассматриваются просто как указание на использование разделителя групп разрядов. Фактический символ, используемый в качестве разделителя групп разрядов в форматированных выходных данных, зависит от числового формата, распознаваемого системой.
(:)	Разделитель времени. В некоторых языковых стандартах для представления разделителя времени могут использоваться другие символы. Разделитель времени разделяет часы, минуты и секунды при форматировании значений времени. Фактический символ, используемый в качестве разделителя времени в форматированных выходных данных, определяется установленными параметрами системы.
(/)	Разделитель даты. В некоторых языковых стандартах для представления разделителя даты могут использоваться другие символы. Разделитель даты разделяет день, месяц и год при форматировании значений времени. Фактический символ, используемый в качестве разделителя даты в форматированных выходных данных, определяется установленными параметрами системы.
(E- E+ e- e+ )	Экспоненциальный формат. Если выражение формата содержит хотя бы один заполнитель цифр (0 или #) справа от E-, E+, e- или e+, число отображается в экспоненциальном формате и E или e вставляется между числом и его показателем степени. Число заполнителей цифр справа определяет количество цифр в показателе степени. Используйте E- или e-, чтобы поместить знак "минус" рядом с отрицательными показателями степени. Используйте E+ или e+, чтобы поместить знак "минус" рядом с отрицательными показателями степени, а знак "плюс" рядом с положительными показателями.
- + \$ ( )	Отображает литеральный символ. Чтобы отобразить символ, который отличается от указанного в списке, поставьте перед ним обратную косую черту (\) или заключите его в двойные кавычки (" ").

СИМВОЛ	ОПИСАНИЕ
(\)	Отображает следующий символ в строке форматирования. Чтобы отобразить символ, который имеет специальное значение в виде литерального символа, поставьте перед ним обратную косую черту ( \ ). Сама по себе обратная косая черта не отображается. Использование обратной косой черты аналогично заключению выводимого символа в двойные кавычки. Чтобы отобразить обратную косую черту, используйте две обратные косые черты ( \ \ ). Примерами символов, которые нельзя отобразить в виде литеральных символов, являются символы форматирования даты и времени (a, c, d, h, m, n, p, q, s, t, w, y, / и :), символы форматирования чисел (#, 0, %, E, e, запятая и точка) и символы форматирования строк (@, &, <, > и !).
("ABC")	Отображает строку, заключенную в двойные кавычки (" ").

## Предопределенные форматы даты и времени

В аргументе `format_string` можно указать следующие предопределенные форматы даты и времени. При использовании форматов, отличных от этих, они обрабатываются как настраиваемые форматы даты и времени:

ФОРМАТ	ОПИСАНИЕ
"General Date"	Отображает дату и/или время. Например, 12.03.2008 11:07:31. Отображение даты определяется текущим значением языка и региональных параметров приложения.
"Long Date" либо "Medium Date"	Отображает дату в соответствии с длинным форматом даты текущего языка и региональных параметров. Например: среда, 12 марта 2008.
"Short Date"	Отображает дату в соответствии с коротким форматом даты текущего языка и региональных параметров. Например, 12.03.2008.
"Long Time" или	Отображает время с использованием длинного формата времени текущего языка и региональных параметров; обычно включает часы, минуты, секунды. Например, 11:07:31.
"Medium Time"	Отображает время в 12-часовом формате. Например, 11:07 AM.
"Short Time"	Отображает время в 24-часовом формате. Например, 11:07.

## Настраиваемые форматы даты и времени

В аргументе `format_string` можно указать следующие символы формата для создания настраиваемых форматов даты и времени:

СИМВОЛ	DESCRIPTION
(:)	Разделитель времени. В некоторых языковых стандартах для представления разделителя времени могут использоваться другие символы. Разделитель времени разделяет часы, минуты и секунды при форматировании значений времени. Фактический символ, используемый в качестве разделителя времени в форматированных выходных данных, определяется установленными параметрами системы.
(/)	Разделитель даты. В некоторых языковых стандартах для представления разделителя даты могут использоваться другие символы. Разделитель даты разделяет день, месяц и год при форматировании значений времени. Фактический символ, используемый в качестве разделителя даты в форматированных выходных данных, определяется установленными параметрами системы.
(\ )	Обратная косая черта. Отображает следующий символ в виде литерального символа. Поэтому он не интерпретируется как символ форматирования.
(")	Двойная кавычка. Отображается текст, заключенный в двойные кавычки. Поэтому он не интерпретируется как символы форматирования.
c	Отображает дату в виде <code>dddd</code> и время в виде <code>tttt</code> в указанном порядке. Если в числе даты нет дробной части, выводятся только сведения о дате. Если целочисленная часть отсутствует, отображаются только сведения о времени.
d	Отображает день месяца в виде числа без нуля в начале (1–31).
дд	Отображает день месяца в виде числа с нулем в начале (01–31).
ddd	Отображает сокращенное название дня недели (вс–сб). Локализовано.
dddd	Отображает полное название дня недели (воскресенье–суббота). Локализовано.
ддддд	Отображает полную дату (включая день, месяц и год), отформатированную в соответствии с системным сокращенным форматом даты. Краткий формат даты по умолчанию — <code>mm/dd/yyyy</code> .
dddddd	Отображает число даты в виде полной даты (включая день, месяц и год), отформатированной в соответствии с форматом полной даты, распознаваемым вашей системой. Полный формат даты по умолчанию — <code>dddd, mmmm d, yyyy</code> .

СИМВОЛ	DESCRIPTION
w	Отображает номер дня недели (от 1 — воскресенье, до 7 — суббота).
ww	Отображает номер недели года (1–54).
m	Отображает месяц в виде числа без нуля в начале (1–12). Если <code>m</code> следует сразу после <code>h</code> или <code>hh</code> , отображаются минуты, а не месяц.
MM	Отображает месяц в виде числа с нулем в начале (01–12). Если <code>mm</code> следует сразу после <code>h</code> или <code>hh</code> , отображаются минуты, а не месяц.
MMM	Отображает сокращенное название месяца (янв–дек). Локализовано.
MMMM	Отображает полное название месяца (январь–декабрь). Локализовано.
q	Отображает номер квартала в году (1–4).
да	Отображает номер дня в году (1–366).
yy	Отображает 2-значное число года (00–99).
ggg	Отображает 4-значное число года (100–9999).
h	Отображает время в виде числа без нуля в начале (0–23).
hh	Отображает время в виде числа с нулем в начале (00–23).
n	Отображает минуты в виде числа без нуля в начале (0–59).
nn	Отображает минуты в виде числа с нулем в начале (00–59).
s	Отображает секунды в виде числа без нуля в начале (0–59).
ss	Отображает секунды в виде числа с нулем в начале (00–59).
tttt	Отображает полное время (включая час, минуту и секунду), отформатированное с помощью разделителя времени, который определен форматом времени, распознаваемым системой. Если выбран соответствующий параметр, отображаются нули в начале при выводе времени до 10:00 до полудня или после полудня. Формат времени по умолчанию — <code>h:mm:ss</code> .

СИМВОЛ	DESCRIPTION
AM/PM	Используется 12-часовой формат времени. С любым значением времени до полудня отображаются прописные буквы AM, а с любым значением времени между полуднем и 23:59 (11:59) — прописные буквы PM.
AM/PM	Используется 12-часовой формат времени. С любым значением времени до полудня отображаются строчные буквы am, а с любым значением времени между полуднем и 23:59 (11:59) — строчные буквы pm.
A/P	Используется 12-часовой формат времени. С любым значением времени до полудня отображается прописная буква A, а с любым значением времени между полуднем и 23:59 (11:59) — прописная буква P.
A/P	Используется 12-часовой формат времени. С любым значением времени до полудня отображается строчная буква a, а с любым значением времени между полуднем и 23:59 (11:59) — строчная буква p.
AMPM	Используется 12-часовой формат времени. С любым значением времени до полудня отображается строковый литерал AM в соответствии с настройками системы, а с любым значением времени между полуднем и 23:59 (11:59) — строковый литерал PM в соответствии с настройками системы. Литералы времени AMPM могут быть либо прописными, либо строчными буквами, однако регистр отображаемой строки соответствует строке, которая определена в настройках системы. Формат по умолчанию — AM/PM. Если в системе задан 24-часовой формат времени, для этой строки, как правило, задается пустая строка.

При форматировании даты-времени используются текущие региональные настройки. Например, рассмотрим дату 25 июня 2020 г. При форматировании с использованием строки форматирования "М/д/гггг" она будет:

- Региональные настройки — США (en-US): "6/25/2020"
- Региональные настройки — Германия (de-DE): "6.25.2020"

#### Примеры настраиваемых форматов даты и времени

В следующих примерах используется четверг, 25 июня 2020 г, 1:23:45 PM (после полудня). В Германии (de-DE) используется 24-часовая система. Эквивалента AM/PM не существует.

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"	Thu	Рекомендуется

ФОРМАТ	РЕЗУЛЬТАТ (EN-US)	РЕЗУЛЬТАТ (DE-DE)
"dddd"	Четверг	Donnerstag
"dddddd"	06/25/2020	25.06.2020
"dddddd"	Четверг, 25 июня, 2020 г.	Donnerstag, 25. Juni 2020
"w"	5	5
"ww"	26	26
"m"	6	6
"mm"	06	06
"mmm"	Июнь	Июнь
"mmm"	Июнь	Juni
"q"	2	2
"y"	177	177
"yy"	20	20
"yyyy"	2020	2020
" ""Year"" yyyy"	2020 год	2020 год
"yyyy \Qq"	2-й квартал 2020 г.	2-й квартал 2020 г.
"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 PM	1:23:45
"hh:nn:ss"	13:23:45	13:23:45
"hh:nn:ss AMPM"	01:23:45 PM	01:23:45
"ttttt"	13:23:45	13:23:45
"ttttt AMPM"	13:23:45 PM	13:23:45
"mm/dd/yyyy hh:nn:ss AMPM"	06/25/2020 01:23:45 PM	6.25.2020 01:23:45

# LEFT

21.09.2020 • 2 minutes to read

Возвращает указанное количество символов с начала текстовой строки.

## Синтаксис

```
LEFT(<text>, <num_chars>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text	Текстовая строка, содержащая символы, которые необходимо извлечь, или ссылка на столбец, содержащий текст.
num_chars	(необязательно) Количество символов, извлекаемых функцией LEFT; по умолчанию 1.

## Возвращаемое значение

Текстовая строка.

## Remarks

- Microsoft Excel содержит различные функции для работы с текстом в однобайтовых и двухбайтовых символьных языках, а DAX работает с Юникодом и сохраняет все символы с одинаковой длиной. 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

Следующий пример возвращает первые пять символов названия компании в столбце [ResellerName] и первые пять букв географического кода в столбце [GeographyKey] и объединяет их для создания идентификатора.

```
= CONCATENATE(LEFT('Reseller'[ResellerName],LEFT(GeographyKey,3))
```

Если аргумент num\_chars является числом, превышающим число доступных символов, функция возвращает максимальное количество доступных символов и не выдает ошибку. Например, столбец [GeographyKey] содержит числа, такие как 1, 12 и 311, поэтому результат также имеет переменную длину.

## См. также

[Текстовые функции](#)



# LEN

02.10.2020 • 2 minutes to read

Возвращает число символов в текстовой строке.

## Синтаксис

```
LEN(<text>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text	Текст, длину которого необходимо найти, или столбец, содержащий текст. Пробелы считаются символами.

## Возвращаемое значение

Целое число, указывающее количество символов в текстовой строке.

## Remarks

- Microsoft Excel содержит различные функции для работы с текстом в однобайтовых и двухбайтовых символьных языках, а DAX работает с Юникодом и сохраняет все символы с одинаковой длиной.
- Функция LEN всегда считает каждый символ как 1, независимо от значения параметра языка по умолчанию.
- Если вы используете LEN со столбцом, содержащим нетекстовые значения, такие как даты или логические значения, функция неявно приводит значение к тексту, используя текущий формат столбца.

## Пример

Следующая формула суммирует длину адресов в столбцах [AddressLine1] и [AddressLine2].

```
= LEN([AddressLine1])+LEN([AddressLin2])
```

# LOWER

21.09.2020 • 2 minutes to read

Преобразует все буквы в текстовой строке в нижний регистр.

## Синтаксис

```
LOWER(<text>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text	Текст для приведения в нижний регистр или ссылка на столбец, содержащий текстовую строку.

## Возвращаемое значение

Текст в нижнем регистре.

## Remarks

Символы, не являющиеся буквами, не изменяются. Так, формула `= LOWER("123ABC")` возвращает результат 123abc.

## Пример

Следующая формула возвращает каждую строку в столбце [ProductCode] и преобразует значение в символы нижнего регистра. Числа в столбце не затрагиваются.

```
= LOWER('New Products'[ProductCode])
```

## См. также раздел

[Текстовые функции](#)

# MID

22.09.2020 • 2 minutes to read

Возвращает строку символов из середины текстовой строки по заданной начальной позиции и длине.

## Синтаксис

```
MID(<text>, <start_num>, <num_chars>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text	Текстовая строка или столбец с текстом, откуда извлекаются символы.
start_num	Положение первого извлекаемого символа. Отсчет начинается с 1.
num_chars	Число возвращаемых символов.

## Возвращаемое значение

Строка текста указанной длины.

## Remarks

Microsoft Excel содержит различные функции для работы с текстом в однобайтовых и двухбайтовых символьных языках, а DAX работает с Юникодом и сохраняет все символы с одинаковой длиной.

## Примеры

Следующее выражение

```
MID("abcde",2,3))
```

Возвращает bcd.

Следующее выражение

```
MID('Reseller'[ResellerName],1,5))
```

Возвращает тот же результат, что и `LEFT([ResellerName],5)`. Оба выражения возвращают первые 5 букв столбца: `[ResellerName]`.

## См. также раздел

[Текстовые функции](#)

# REPLACE

22.09.2020 • 2 minutes to read

Функция REPLACE заменяет часть текстовой строки на основе указанного числа символов другой текстовой строкой.

## Синтаксис

```
REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
old_text	Текстовая строка, содержащая символы, которые необходимо заменить, или ссылка на столбец, содержащий текст.
start_num	Положение в <b>old_text</b> символа, который заменяется на <b>new_text</b> .
num_chars	Количество замещаемых символов. <b>Внимание!</b> Если аргумент <i>num_chars</i> пуст или ссылается на столбец, результатом которого является пустое значение, строка <i>new_text</i> вставляется в позиции <i>start_num</i> без замены каких-либо символов. Это то же поведение, что и в Excel.
new_text	Замещающий текст для указанных символов в <b>old_text</b> .

## Возвращаемое значение

Текстовая строка.

## Примечания

- Microsoft Excel содержит различные функции для работы с текстом в однобайтовых и двухбайтовых символьных языках, а DAX работает с Юникодом и сохраняет все символы с одинаковой длиной.
- 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, заменяющий первые два символа кода продукта в столбце [ProductCode] на новый двухбуквенный код, OB.

```
= REPLACE('New Products'[Product Code],1,2,"OB")
```

## См. также



# REPT

22.09.2020 • 2 minutes to read

Повторяет текст заданное число раз. Функция REPT позволяет заполнить ячейку нужным количеством экземпляров текстовой строки.

## Синтаксис

```
REPT(<text>, <num_times>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
текст	Повторяемый текст.
num_times	Положительное число, указывающее количество повторов текста.

## Возвращаемое значение

Строка, содержащая изменения.

## Примечания

- Если значение параметра `number_times` равно 0 (нулю), функция REPT возвращает пустое значение.
- Если значение параметра `number_times` не является целым числом, оно усекается.
- Длина результата вычисления функции REPT не может превышать 32 767 символов, в противном случае REPT возвращает ошибку.

## Пример: Повторяющиеся строковые литералы

В следующем примере возвращается строка 85, повторяемая три раза.

```
= REPT("85",3)
```

## Пример: Повторяющиеся значения столбцов

В следующем примере возвращается строка в столбце [MyText], повторяемая несколько раз в столбце [MyNumber]. Поскольку формула распространяется на весь столбец, результирующая строка зависит от текста и числового значения в каждой строке.

```
= REPT([MyText],[MyNumber])
```

MYTEXT	MYNUMBER	CALCULATEDCOLUMN1
Текст	2	TextText
Номер	0	
85	3	858585

## См. также

[Текстовые функции](#)

# RIGHT

22.09.2020 • 2 minutes to read

Возвращает последний символ или символы в текстовой строке на основе указанного количества символов.

## Синтаксис

```
RIGHT(<text>, <num_chars>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
текст	Текстовая строка, содержащая символы, которые необходимо извлечь, или ссылка на столбец, содержащий текст.
num_chars	(необязательно) Количество символов, извлекаемых функцией RIGHT; по умолчанию 1. Можно также использовать ссылку на столбец, содержащий числа.

Если ссылка на столбец не содержит текст, она неявно приводится к тексту.

## Возвращаемое значение

Текстовая строка, содержащая крайние правые символы.

## Примечания

- Функция RIGHT всегда считает каждый символ, будь то однобайтный или двухбайтный символ, как 1, независимо от значения параметра языка по умолчанию.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример: Возврат фиксированного количества символов

Следующая формула возвращает две последние цифры кода продукта в таблице новых продуктов.

```
= 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 новых продуктов, в зависимости от числа в столбце MyCount. Если в столбце нет значения, MyCount или значение не является пустым, то функция RIGHT также возвращает пустое значение.



```
= RIGHT('New Products'[ProductCode],[MyCount])
```

## См. также

[Текстовые функции](#)

[LEFT](#)

[MID](#)

# SEARCH

02.10.2020 • 5 minutes to read

Возвращает номер символа, на котором были в первый раз найдены определенный символ или текстовая строка; читается слева направо. Поиск ведется без учета регистра и диакритических знаков.

## Синтаксис

```
SEARCH(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
find_text	Искомый текст.  В <b>find_text</b> можно использовать подстановочные знаки — вопросительный знак (?) и звездочку (*). Вопросительный знак соответствует любому отдельному символу; звездочка соответствует любой последовательности символов. Если нужно найти сами знак вопроса или звездочку, введите тильду (~) перед символом.
within_text	Текст, в котором выполняется поиск строки <b>find_text</b> , или столбец с текстом.
start_num	(необязательно) Положение в <b>within_text</b> символа, с которого начинается поиск. Если этот параметр опущен, используется значение 1.
NotFoundValue	(необязательно) Значение, которое возвращается в случае, если операции не удалось найти соответствующую подстроку: как правило, это значение 0, -1 или BLANK().

## Возвращаемое значение

Номер начальной позиции первой текстовой строки относительно первого символа второй текстовой строки.

## Примечания

- Функция поиска не учитывает регистр. При поиске "N" будет найдено первое вхождение "N" или "n".
- Функция поиска не учитывает диакритические знаки. При поиске "á" будет найдено первое вхождение "á", но не вхождения "a", "à" или прописные "A", "Á".
- С помощью этой функции можно найти одну текстовую строку во второй текстовой строке и получить позицию, с которой начинается первая строка.
- Функцию SEARCH можно использовать для определения расположения символа или текстовой

строки в другой текстовой строке, а затем использовать функцию MID для возврата текста или изменить текст с помощью функции REPLACE.

- Если find\_text не удастся найти в within\_text, формула возвращает ошибку. Такое поведение аналогично Excel, где возвращается значение #VALUE, если подстрока не найдена. Значения NULL в within\_text будут интерпретироваться 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: поиск в строке

Следующая формула находит расположение буквы "n" в слове "printer".

```
= SEARCH("n","printer")
```

Формула возвращает 4, поскольку "n" — четвертый символ в слове "printer".

## Пример: поиск в столбце

В качестве аргумента SEARCH можно использовать ссылку на столбец. Следующая формула находит позицию символа "-" (дефис) в столбце [PostalCode].

```
= SEARCH("-",[PostalCode])
```

Возвращаемый результат представляет собой столбец чисел, указывающих индексы дефиса.

## Пример: обработка ошибок в функции SEARCH

Формула в предыдущем примере завершится ошибкой, если строка поиска не будет найдена в каждой строке исходного столбца. Поэтому в следующем примере показано, как использовать IFERROR с функцией SEARCH, чтобы гарантировать, что для каждой строки возвращается допустимый результат.

Следующая формула находит позиции символа "-" в столбце и возвращает значение –1, если строка не найдена.

```
= IFERROR(SEARCH("-",[PostalCode]),-1)
```

Тип данных значения, используемого в качестве выходных данных при ошибке, должен соответствовать типу выходных данных в случае, когда ошибки нет. В этом случае вы указываете числовое значение для вывода в случае ошибки, так как SEARCH возвращает целочисленное значение. Однако можно также вернуть пустое значение (пустую строку), используя `BLANK()` в качестве второго аргумента IFERROR.

## См. также

[MID](#)

[REPLACE](#)

[Текстовые функции](#)

# SUBSTITUTE

22.09.2020 • 2 minutes to read

Заменяет существующий текст новым текстом в текстовой строке.

## Синтаксис

```
SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
текст	Текст, в котором нужно заменить символы, или ссылка на столбец, содержащий текст.
old_text	Существующий текст, который требуется заменить.
new_text	Текст, на который заменяется old_text.
instance_num	(необязательно) Вхождение old_text, которое требуется заменить. Если этот параметр опущен, заменяются все экземпляры old_text

## Возвращаемое значение

Строка текста.

## Примечания

- Используйте функцию SUBSTITUTE, если требуется заменить конкретный текст в текстовой строке. Используйте функцию REPLACE, если нужно заменить любой текст переменной длины, который находится в определенном месте в текстовой строке.
- Функция SUBSTITUTE учитывает регистр. Если регистр не совпадает с text и old\_text, то функция SUBSTITUTE не заменяет 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], который заменяет новый код продукта NW для старого кода продукта PA везде, где он встречается в столбце.

```
= SUBSTITUTE([Product Code], "NW", "PA")
```

## См. также

Текстовые функции

REPLACE

# TRIM

22.09.2020 • 2 minutes to read

Удаляет все пробелы из текста, за исключением одиночных пробелов между словами.

## Синтаксис

```
TRIM(<text>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text	Текст, из которого должны быть удалены пробелы, или столбец, содержащий текст.

## Возвращаемое значение

Строка с удаленными пробелами.

## Remarks

- Используйте TRIM в тексте, полученном из другого приложения с неравномерными пробелами.
- Функция TRIM изначально была разработана для удаления 7-разрядного символа пробела ASCII (значение 32) из текста. В кодировке Юникод имеется дополнительный символ пробела, называемый символом неразрывного пробела, который имеет десятичное значение 160. Этот символ обычно используется на веб-страницах в качестве объекта HTML &nbsp; . Сама по себе функция TRIM не удаляет этот неразрывный символ пробела. Пример того, как удалить символы пробела из текста, см. в разделе "Удаление пробелов и непечатаемых символов из текста".

## Пример

Следующая формула создает новую строку, которая не содержит конечных пробелов.

```
= TRIM("A column with trailing spaces.  ")
```

При создании формулы она передается по строке при введении, так что вы видите исходную строку в каждой формуле, но не результаты. Однако при вычислении формулы строка усекается.

Чтобы убедиться, что формула дает правильный результат, проверьте длину вычисляемого столбца, созданного предыдущей формулой, следующим образом:

```
= LEN([Calculated Column 1])
```

## См. также раздел

[Текстовые функции](#)

# UNICHAR

02.10.2020 • 2 minutes to read

Возвращает символ Юникода, на который ссылается числовое значение.

## Синтаксис

```
UNICHAR(number)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
number	Номер Юникода, представляющий символ.

## Возвращаемое значение

Символ, представленный номером Юникода.

## Remarks

- Если XML-символы не являются допустимыми, UNICHAR возвращает ошибку.
- Если числа 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 коде UTF-8 или UTF-16.

## Пример

В следующем примере возвращается символ, представленный номером Юникода 66 (А в верхнем регистре).

```
= UNICHAR(65)
```

В следующем примере возвращается символ, представленный номером Юникода 32 (пробел).

```
= UNICHAR(32)
```

В следующем примере возвращается символ, представленный номером Юникода 9733 (символ ★).

```
= UNICHAR(9733)
```

# UNICODE

22.09.2020 • 2 minutes to read

Возвращает число (код знака), соответствующее первому знаку в тексте.

## Синтаксис

```
UNICODE( <Text> )
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
текст	Текст — это символ, для которого требуется получить значение Юникода.

## Возвращаемое значение

Числовой код для первого символа в текстовой строке.



# UPPER

22.09.2020 • 2 minutes to read

Преобразует все буквы в текстовой строке в прописные.

## Синтаксис

```
UPPER (<text>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text	Текст для приведения в верхний регистр или ссылка на столбец, содержащий текстовую строку.

## Возвращаемое значение

Тот же текст в верхнем регистре.

## Пример

Следующая формула преобразует строку в столбце [ProductCode] в верхний регистр. На небуквенные символы это не влияет.

```
= UPPER(['New Products'[Product Code])
```

## См. также раздел

[Текстовые функции](#)

[LOWER, функция](#)

# Значение

22.09.2020 • 2 minutes to read

Преобразует текстовую строку, представляющую число, в числовой формат.

## Синтаксис

```
VALUE(<text>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
text	Текст для преобразования.

## Возвращаемое значение

Преобразованное число в десятичном типе данных.

## Remarks

- Значение, передаваемое в качестве параметра text, может быть в любом формате константы, числа, даты или времени, распознаваемом используемым приложением или службами. Если значение параметра text указано в формате, отличном от указанных выше форматов, возвращается ошибка.
- Обычно не требуется использовать функцию VALUE в формуле, поскольку подсистема неявно преобразует текст в числа по мере необходимости.
- Также можно использовать ссылки на столбцы. Например, если имеется столбец, содержащий смешанные типы чисел, можно использовать значение для преобразования всех значений в один числовой тип данных. Однако при использовании функции VALUE со столбцом, содержащим смешанные числа и текст, весь столбец помечается как ошибка, поскольку не все значения во всех строках могут быть преобразованы в числа.

## Пример

Следующая формула преобразует типизированную строку "3" в числовое значение 3.

```
= VALUE("3")
```

## См. также раздел

[Текстовые функции](#)

# Функции операций со временем

02.10.2020 • 6 minutes to read

Язык выражений для анализа данных (DAX) включает функции логики операций со временем, которые позволяют манипулировать данными с использованием временных периодов, таких как дни, месяцы, кварталы и годы, а затем создавать и сравнивать вычисления на основе этих периодов.

## В этой категории

ФУНКЦИЯ	ОПИСАНИЕ
<a href="#">CLOSINGBALANCEMONTH</a>	Вычисляет выражение в последний день месяца в текущем контексте.
<a href="#">CLOSINGBALANCEQUARTER</a>	Вычисляет выражение в последний день квартала в текущем контексте.
<a href="#">CLOSINGBALANCEYEAR</a>	Вычисляет выражение в последний день года в текущем контексте.
<a href="#">DATEADD</a>	Возвращает таблицу, содержащую столбец дат со сдвигом вперед или назад на указанное число интервалов от дат в текущем контексте.
<a href="#">DATESBETWEEN</a>	Возвращает таблицу, содержащую столбец дат, который начинается с указанной даты начала и продолжается до указанной даты окончания.
<a href="#">DATESINPERIOD</a>	Возвращает таблицу, содержащую столбец дат, который начинается с даты начала и продолжается заданные количество и тип интервалов даты.
<a href="#">DATESMTD</a>	Возвращает таблицу, содержащую столбец дат месяца до текущей даты, в текущем контексте.
<a href="#">DATESQTD</a>	Возвращает таблицу, содержащую столбец дат квартала до текущей даты, в текущем контексте.
<a href="#">DATESYTD</a>	Возвращает таблицу, содержащую столбец дат года до текущей даты, в текущем контексте.
<a href="#">ENDOFMONTH</a>	Возвращает последнюю дату месяца в текущем контексте для указанного столбца дат.
<a href="#">ENDOFQUARTER</a>	Возвращает последнюю дату квартала в текущем контексте для указанного столбца дат.
<a href="#">ENDOFYEAR</a>	Возвращает последнюю дату года в текущем контексте для указанного столбца дат.

ФУНКЦИЯ	ОПИСАНИЕ
FIRSTDATE	Возвращает первую дату в текущем контексте для указанного столбца дат.
FIRSTNONBLANK	Возвращает первое значение столбца column, отфильтрованного по текущему контексту, где выражение не является пустым.
LASTDATE	Возвращает последнюю дату в текущем контексте для указанного столбца дат.
LASTNONBLANK	Возвращает последнее значение столбца column, отфильтрованного по текущему контексту, где выражение не является пустым.
NEXTDAY	Возвращает таблицу, содержащую столбец со всеми датами следующего дня на основе первой даты, указанной в столбце dates в текущем контексте.
NEXTMONTH	Возвращает таблицу, содержащую столбец со всеми датами следующего месяца на основе первой даты в столбце dates в текущем контексте.
NEXTQUARTER	Возвращает таблицу, содержащую столбец со всеми датами следующего квартала на основе первой даты в столбце dates в текущем контексте.
NEXTYEAR	Возвращает таблицу, содержащую столбец со всеми датами следующего года на основе первой даты в столбце dates в текущем контексте.
OPENINGBALANCEMONTH	Вычисляет выражение в первый день месяца в текущем контексте.
OPENINGBALANCEQUARTER	Вычисляет выражение в первый день квартала в текущем контексте.
OPENINGBALANCEYEAR	Вычисляет выражение expression в первый день года в текущем контексте.
PARALLELPERIOD	Возвращает таблицу, содержащую столбец дат, который представляет период, параллельный датам в указанном столбце dates, в текущем контексте, с датами, сдвинутыми на указанное число интервалов вперед или назад во времени.
PREVIOUSDAY	Возвращает таблицу, содержащую столбец со всеми датами, представляющими день, предшествующий первой дате в столбце dates, в текущем контексте.
PREVIOUSMONTH	Возвращает таблицу, содержащую столбец со всеми датами предыдущего месяца на основе первой даты в столбце dates в текущем контексте.

ФУНКЦИЯ	ОПИСАНИЕ
PREVIOUSQUARTER	Возвращает таблицу, содержащую столбец со всеми датами предыдущего квартала на основе первой даты в столбце dates в текущем контексте.
PREVIOUSYEAR	Возвращает таблицу, содержащую столбец со всеми датами предыдущего года на основе последней даты в столбце dates в текущем контексте.
SAMEPERIODLASTYEAR	Возвращает таблицу, содержащую столбец дат со сдвигом на год назад от дат в столбце dates 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 дат в месяце до текущей даты в текущем контексте.
TOTALQTD	Вычисляет значение выражения для дат в квартале до текущей даты в текущем контексте.
TOTALYTD	Вычисляет значение с начала года для выражения в текущем контексте.

# CLOSINGBALANCEMONTH

02.10.2020 • 2 minutes to read

Вычисляет **выражение** в последний день месяца в текущем контексте.

## Синтаксис

```
CLOSINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

### Параметры

ПАРАМЕТР	ОПРЕДЕЛЕНИЕ
expression	Выражение, возвращающее скалярное значение.
даты	Столбец, содержащий даты.
фильтр	(необязательно) Выражение, задающее фильтр, применяемый к текущему контексту.

## Возвращаемое значение

Скалярное значение, представляющее **выражение**, которое вычисляется в последний день месяца в текущем контексте.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.

### NOTE

Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).

### NOTE

Выражение **filter** имеет ограничения, описанные в разделе [Функция 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 значение стоимости складских запасов

по состоянию на конец месяца.

```
=  
CLOSINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[  
DateKey])
```

## См. также

[Функции операций со временем](#)

[CLOSINGBALANCEYEAR](#), функция

[Функция CLOSINGBALANCEQUARTER](#)

# CLOSINGBALANCEQUARTER

02.10.2020 • 2 minutes to read

Вычисляет **выражение** в последний день квартала в текущем контексте.

## Синтаксис

```
CLOSINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Выражение, возвращающее скалярное значение.
даты	Столбец, содержащий даты.
фильтр	(необязательно) Выражение, задающее фильтр, применяемый к текущему контексту.

## Возвращаемое значение

Скалярное значение, представляющее **выражение**, которое вычисляется в последний день квартала в текущем контексте.

## Remarks

- Аргумент `dates` может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.

### NOTE

Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).

### NOTE

Выражение `filter` имеет ограничения, описанные в разделе [Функция 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 значение стоимости складских запасов



по состоянию на конец квартала.

```
=  
CLOSINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTim  
e[DateKey])
```

## См. также

[Функции операций со временем](#)

[CLOSINGBALANCEYEAR, функция](#)

[CLOSINGBALANCEMONTH, функция](#)

# CLOSINGBALANCEYEAR

02.10.2020 • 2 minutes to read

Вычисляет **выражение** в последний день года в текущем контексте.

## Синтаксис

```
CLOSINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Выражение, возвращающее скалярное значение.
даты	Столбец, содержащий даты.
фильтр	(необязательно) Выражение, задающее фильтр, применяемый к текущему контексту.
year_end_date	(необязательно) Литеральная строка с датой, которая определяет дату окончания года. Значение по умолчанию — 31 декабря.

## Возвращаемое значение

Скалярное значение, представляющее **выражение**, которое вычисляется в последний день года в текущем контексте.

## Remarks

- Параметр `year_end_date` — это строковый литерал даты в языковом стандарте клиента, в котором была создана книга. Компонент года в дате игнорируется.
- Аргумент `dates` может принимать любое из следующих значений:
  - Ссылка на столбец даты и времени.
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.

### NOTE

Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).

### NOTE

Выражение `filter` имеет ограничения, описанные в разделе [Функция 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 значение стоимости складских запасов по состоянию на конец года.

```
=  
CLOSINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

## См. также

[Функции операций со временем](#)

[CLOSINGBALANCEYEAR, функция](#)

[Функция CLOSINGBALANCEQUARTER](#)

[CLOSINGBALANCEMONTH, функция](#)

# DATEADD

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец дат со сдвигом вперед или назад на указанное число интервалов от дат в текущем контексте.

## Синтаксис

```
DATEADD(<dates>,<number_of_intervals>,<interval>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.
number_of_intervals	Целое число, указывающее количество интервалов, добавляемых в даты или вычитаемых из них.
interval	Интервал для сдвига дат. Значение параметра interval может быть одним из следующих: <code>year</code> , <code>quarter</code> , <code>month</code> , <code>day</code> .

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

Аргумент `dates` может принимать любое из следующих значений:

- ссылка на столбец даты и времени;
- табличное выражение, возвращающее один столбец значений даты и времени;
- логическое выражение, определяющее таблицу значений даты и времени из одного столбца.

### NOTE

Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).

- Если число, указанное для `number_of_intervals`, положительное, даты в `dates` перемещаются вперед во времени. Если число отрицательное, даты в `dates` сдвигаются назад во времени.
- Параметр `interval` является перечислением, а не набором строк. Поэтому значения не должны заключаться в кавычки. Кроме того, значения `year`, `quarter`, `month`, `day` должны быть записаны полностью при использовании.
- Таблица результатов содержит только даты, которые существуют в столбце `dates`.
- Если даты в текущем контексте не формируют 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. Сдвиг набора дат

Следующая формула вычисляет даты на год раньше дат в текущем контексте.

```
= DATEADD(DateTime[DateKey], -1, year)
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

# DATESBETWEEN

02.10.2020 • 3 minutes to read

Возвращает таблицу, содержащую столбец дат, который начинается с указанной даты начала и продолжается до указанной даты окончания.

Эта функция подходит для передачи в качестве фильтра функции [CALCULATE](#). Используйте его для фильтрации выражения по настраиваемому диапазону дат.

## NOTE

Если вы работаете со стандартными интервалами дат, такими как дни, месяцы, кварталы или годы, рекомендуется использовать более подходящую функцию [DATESINPERIOD](#).

## Синтаксис

```
DATESBETWEEN(<dates>, <start_date>, <end_date>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
dates	Столбец даты.
start_date	Выражение даты.
end_date	Выражение даты.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- В наиболее распространенном варианте использования **dates** является ссылкой на столбец дат помеченной таблицы дат.
- Если значение **start\_date** равно BLANK, **start\_date** будет самым ранним значением в столбце **dates**.
- Если значение **end\_date** равно BLANK, **end\_date** будет самым поздним значением в столбце **dates**.
- Даты, используемые в параметрах **start\_date** и **end\_date**, являются включающими. Например, если значение **start\_date** равно 1 июля 2019 г., то эта дата будет включена в возвращаемую таблицу (предоставляя дату в столбце **dates**).
- Возвращаемая таблица может содержать только даты, хранящиеся в столбце **dates**. Например, если столбец **dates** начинается с 1 июля 2017 г., а значение **start\_date** равно 1 июля 2016 г., то возвращаемая таблица будет начинаться с 1 июля 2017 г.
- 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 меры таблицы **Sales** используется функция **DATESBETWEEN** для создания вычисления *срока действия* (LTD). Срок действия представляет накопление меры с течением времени с момента самого начала времени.

Обратите внимание, что в формуле используется функция **MAX**. Эта функция возвращает последнюю дату в контексте фильтра. Таким образом, функция **DATESBETWEEN** возвращает таблицу дат, начиная с самой ранней даты до момента, когда сообщается о последней дате.

*Примеры в этой статье можно добавить в образец модели Power BI Desktop. Чтобы получить модель, см. [пример модели DAX](#).*

```
Customers LTD =  
CALCULATE(  
    DISTINCTCOUNT(Sales[CustomerKey]),  
    DATESBETWEEN(  
        'Date'[Date],  
        BLANK(),  
        MAX('Date'[Date])  
    )  
)
```

Предположим, что самая ранняя дата, хранящаяся в таблице **Date**, — 1 июля 2017 г. Поэтому, когда отчет фильтрует меру по июню 2020 г., функция **DATESBETWEEN** возвращает диапазон дат с 1 июля 2017 г. по 30 июня 2020 г.

## См. также

- [Функции логики операций со временем \(DAX\)](#)
- [Функции даты и времени \(DAX\)](#)
- [Функция DATESINPERIOD \(DAX\)](#)

# DATESINPERIOD

02.10.2020 • 3 minutes to read

Возвращает таблицу, содержащую столбец дат, который начинается с даты начала и продолжается заданные количество и тип интервалов даты.

Эта функция подходит для передачи в качестве фильтра функции [CALCULATE](#). Используйте его для фильтрации выражения по стандартным интервалам дат, например дням, месяцам, кварталам или годам.

## Синтаксис

```
DATESINPERIOD(<dates>, <start_date>, <number_of_intervals>, <interval>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
dates	Столбец даты.
start_date	Выражение даты.
number_of_intervals	Целое число, указывающее количество интервалов, добавляемых в даты или вычитаемых из них.
interval	Интервал для сдвига дат. Значение параметра interval может быть одним из следующих: <code>DAY</code> , <code>MONTH</code> , <code>QUARTER</code> , <code>YEAR</code> .

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- В наиболее распространенном варианте использования `dates` является ссылкой на столбец дат помеченной таблицы дат.
- Если число, указанное для `number_of_intervals`, положительное, даты перемещаются вперед во 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 г., а значение `start_date` равно 1 июля 2016 г., то возвращаемая таблица будет начинаться с 1 июля 2017 г.
- 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 меры таблицы Sales используется функция DATESINPERIOD для вычисления дохода за прошлый год (PY).

Обратите внимание, что в формуле используется функция [MAX](#). Эта функция возвращает последнюю дату в контексте фильтра. Таким образом, функция DATESINPERIOD возвращает таблицу дат, начиная с последней даты последнего года.

*Примеры в этой статье можно добавить в образец модели Power BI Desktop. Чтобы получить модель, см. [пример модели DAX](#).*

```
Revenue PY =  
CALCULATE(  
    SUM(Sales[Sales Amount]),  
    DATESINPERIOD(  
        'Date'[Date],  
        MAX('Date'[Date]),  
        -1,  
        YEAR  
    )  
)
```

Учтите, что отчет фильтруется по месяцам июня 2020 г. Функция MAX возвращает 30 июня 2020 г. Затем функция DATESINPERIOD возвращает диапазон дат с 1 июля 2019 г. по 30 июня 2020 г. Это год со значениями, начиная с 30 июня 2020 г. за последний год.

## См. также

[Функции логики операций со временем \(DAX\)](#)

[Функции даты и времени \(DAX\)](#)

[Функция DATESBETWEEN \(DAX\)](#)

# DATESMTD

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец дат месяца до текущей даты, в текущем контексте.

## Синтаксис

```
DATESMTD(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

Аргумент **dates** может принимать любое из следующих значений:

- ссылка на столбец даты и времени;
- табличное выражение, возвращающее один столбец значений даты и времени;
- логическое выражение, определяющее таблицу значений даты и времени из одного столбца.

### NOTE

Ограничения логических выражений описаны в разделе [Функция 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 образце формулы создается мера, которая вычисляет общий объем продаж за месяц до текущей даты.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESMTD(DateTime[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[DATESYTD, функция](#)

[Функция DATESQTD](#)



# DATESQTD

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец дат квартала до текущей даты, в текущем контексте.

## Синтаксис

```
DATESQTD(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

Аргумент **dates** может принимать любое из следующих значений:

- ссылка на столбец даты и времени;
- табличное выражение, возвращающее один столбец значений даты и времени;
- логическое выражение, определяющее таблицу значений даты и времени из одного столбца.

### NOTE

Ограничения логических выражений описаны в разделе [Функция 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 образце формулы создается мера, которая вычисляет продажи через Интернет за квартал нарастающим итогом.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESQTD(DateTime[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[DATESYTD, функция](#)

[DATESMTD, функция](#)



# DATESYTD

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец дат года до текущей даты, в текущем контексте.

## Синтаксис

```
DATESYTD(<dates> [, <year_end_date>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.
year_end_date	(необязательно) Литеральная строка с датой, которая определяет дату окончания года. Значение по умолчанию — 31 декабря.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

Аргумент **dates** может принимать любое из следующих значений:

- ссылка на столбец даты и времени;
- табличное выражение, возвращающее один столбец значений даты и времени;
- логическое выражение, определяющее таблицу значений даты и времени из одного столбца.

### NOTE

Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).

- Параметр **year\_end\_date** — это строковый литерал даты в языковом стандарте клиента, в котором была создана книга. Компонент года в дате 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 образце формулы создается мера, которая вычисляет объем продаж нарастающим итогом.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESYTD(DateTime[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[DATESMTD, функция](#)

[Функция DATESQTD](#)

# ENDOFMONTH

02.10.2020 • 2 minutes to read

Возвращает последнюю дату месяца в текущем контексте для указанного столбца дат.

## Синтаксис

```
ENDOFMONTH(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция 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 образце формулы создается мера, которая возвращает конец месяца для текущего контекста.

```
= ENDOFMONTH(DateTime[DateKey])
```

## См. также

[Функции даты и времени](#)

[Функции операций со временем](#)

[Функция ENDOFYEAR](#)

[Функция ENDOFQUARTER](#)



# ENDOFQUARTER

02.10.2020 • 2 minutes to read

Возвращает последнюю дату квартала в текущем контексте для указанного столбца дат.

## Синтаксис

```
ENDOFQUARTER(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция 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 образце формулы создается мера, которая возвращает конец квартала для текущего контекста.

```
= ENDOFQUARTER(DateTime[DateKey])
```

## См. также

[Функции даты и времени](#)

[Функции операций со временем](#)

[Функция ENDOFYEAR](#)

[Функция ENDOFMONTH](#)

# ENDOFYEAR

02.10.2020 • 2 minutes to read

Возвращает последнюю дату года в текущем контексте для указанного столбца дат.

## Синтаксис

```
ENDOFYEAR(<dates> [, <year_end_date>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.
year_end_date	(необязательно) Литеральная строка с датой, которая определяет дату окончания года. Значение по умолчанию — 31 декабря.

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

## Remarks

- Аргумент `dates` может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Параметр `year_end_date` — это строковый литерал даты в языковом стандарте клиента, в котором была создана книга. Компонент года в дате 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 образце формулы создается мера, которая возвращает конец финансового года, заканчивающегося 30 июня, для текущего контекста.

```
= ENDOFYEAR(DateTime[DateKey], "06/30/2004")
```

## См. также

[Функции даты и времени](#)

[Функции операций со временем](#)

Функция ENDOFMONTH  
Функция ENDOFQUARTER

# FIRSTDATE

02.10.2020 • 2 minutes to read

Возвращает первую дату в текущем контексте для указанного столбца дат.

## Синтаксис

```
FIRSTDATE(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Если текущий контекст определяет одну дату, функции FIRSTDATE и LASTDATE возвращают одинаковую дату.
- Возвращаемое значение представляет собой таблицу, содержащую один столбец и одно значение. Таким образом, эту функцию можно использовать в качестве аргумента для любой функции, принимающей в качестве аргумента таблицу. Кроме того, возвращаемое значение можно использовать везде, где требуется дата.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

В следующем образце формулы создается мера, которая получает первую дату продажи через канал в Интернете для текущего контекста.

```
= FIRSTDATE('InternetSales_USD'[SaleDateKey])
```

## См. также

[Функции даты и времени](#)

Функции операций со временем

LASTDATE, функция

Функция FIRSTNONBLANK

# FIRSTNONBLANK

02.10.2020 • 2 minutes to read

Возвращает первое значение столбца `column`, отфильтрованного по текущему контексту, где выражение не является пустым.

## Синтаксис

```
FIRSTNONBLANK(<column>, <expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Выражение столбца.
expression	Выражение, вычисленное для пустых значений для каждого значения столбца <code>column</code> .

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку с вычисленным первым значением.

## Remarks

- Аргумент `column` может принимать любое из следующих значений:
  - ссылка на любой столбец;
  - таблица из одного столбца;
- логическое выражение, определяющее таблицу из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Эта функция обычно используется для возврата первого значения столбца, для которого выражение не является пустым. Например, можно получить последнее значение, для которого имеются продажи продукта.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## См. также

[LASTNONBLANK](#), функция  
[Статистические функции](#)

# FIRSTNONBLANKVALUE

02.10.2020 • 2 minutes to read

Вычисляет выражение, отфильтрованное по отсортированным значениям столбца, а также возвращает первое значение выражения, которое не является пустым.

## Синтаксис

```
FIRSTNONBLANKVALUE(<column>, <expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец или выражение, которое возвращает таблицу с одним столбцом.
expression	Выражение, вычисленное для каждого значения столбца <column>.

## Возвращаемое значение

Первое непустое значение выражения <expression>, соответствующее отсортированным значениям столбца <column>.

## Remarks

- Аргумент column может принимать любое из следующих значений:
  - ссылка на любой столбец;
  - таблица из одного столбца;
- Эта функция отличается от FIRSTNONBLANK тем, что столбец <column> добавляется в контекст фильтра для вычисления выражения <expression>.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Рассмотрим следующий запрос DAX:

```
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[Class],
    "FNBV",
    FIRSTNONBLANKVALUE(
        DimDate[Date],
        SUM(FactInternetSales[SalesAmount])
    )
)
```

Возвращает:

DIMPRODUCT[CLASS]	[FNBV]
L	699,0982
H	13778,24
M	1000,4375
	533,83



# LASTDATE

02.10.2020 • 2 minutes to read

Возвращает последнюю дату в текущем контексте для указанного столбца дат.

## Синтаксис

```
LASTDATE(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Если текущий контекст определяет одну дату, функции **FIRSTDATE** и **LASTDATE** возвращают одинаковую дату.
- Технически возвращаемое значение представляет собой таблицу, содержащую один столбец и одно значение. Таким образом, эту функцию можно использовать в качестве аргумента для любой функции, принимающей в качестве аргумента таблицу. Кроме того, возвращаемое значение можно использовать везде, где требуется дата.
- Эта функция не поддерживается для использования в режиме **DirectQuery** при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

В следующем образце формулы создается мера, которая получает последнюю для текущего контекста дату продажи через канал в Интернете.

Чтобы увидеть, как это работает, создайте сводную таблицу и добавьте поле **CalendarYear** в область **Метки строк** сводной таблицы. Затем добавьте меру с именем **LastSaleDate**, используя формулу, определенную в разделе кода, в области **Значения** сводной таблицы.

```
= LASTDATE('InternetSales_USD'[SaleDateKey])
```

## См. также

[Функции даты и времени](#)

[Функции операций со временем](#)

[FIRSTDATE, функция](#)

[LASTNONBLANK, функция](#)

# LASTNONBLANK

02.10.2020 • 2 minutes to read

Возвращает последнее значение столбца `column`, отфильтрованного по текущему контексту, где выражение не является пустым.

## Синтаксис

```
LASTNONBLANK(<column>,<expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Выражение столбца.
expression	Выражение, вычисленное для пустых значений для каждого значения столбца <code>column</code> .

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку с вычисленным последним значением.

## Remarks

- Аргумент `column` может принимать любое из следующих значений:
  - ссылка на любой столбец;
  - таблица из одного столбца;
  - логическое выражение, определяющее таблицу из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Эта функция обычно используется для возврата последнего значения столбца, для которого выражение не является пустым. Например, можно получить последнее значение, для которого имеются продажи продукта.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## См. также

[Функция FIRSTNONBLANK](#)

[Статистические функции](#)

# LASTNONBLANKVALUE

02.10.2020 • 2 minutes to read

Вычисляет выражение, отфильтрованное по отсортированным значениям столбца, а также возвращает последнее значение выражения, которое не является пустым.

## Синтаксис

```
LASTNONBLANKVALUE(<column>, <expression>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
столбец	Столбец или выражение, которое возвращает таблицу с одним столбцом.
expression	Выражение, вычисленное для каждого значения столбца <column>.

## Возвращаемое значение

Последнее непустое значение выражения <expression>, соответствующее отсортированным значениям столбца <column>.

## Remarks

- Аргумент column может принимать любое из следующих значений:
  - ссылка на любой столбец;
  - таблица из одного столбца;
- Эта функция отличается от LASTNONBLANK тем, что столбец <column> добавляется в контекст фильтра для вычисления выражения <expression>.
- Эта функция не поддерживается для использования в режиме DirectQuery при использовании в вычисляемых столбцах или правилах безопасности на уровне строк (RLS).

## Пример

Рассмотрим следующий запрос DAX:

```
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[Class],
    "LNBV",
    LASTNONBLANKVALUE(
        DimDate[Date],
        SUM(FactInternetSales[SalesAmount])
    )
)
```

Возвращает:

DIMPRODUCT[CLASS]	[LNBV]
L	132,44
H	137,6
M	84,97
	2288,6

# NEXTDAY

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец со всеми датами следующего дня на основе первой даты, указанной в столбце **dates** в текущем контексте.

## Синтаксис

```
NEXTDAY(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- Эта функция возвращает все даты со следующего дня до первой даты во входном параметре. Например, если первой датой в аргументе **dates** является 10 июня 2009 г, то функция возвращает все даты, равные 11 июня 2009 г.
- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция 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 через Интернет за следующий год.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTDAY('DateTime'[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[Функция NEXTQUARTER](#)

NEXTMONTH, функция

Функция NEXTYEAR

# NEXTMONTH

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец со всеми датами следующего месяца на основе первой даты в столбце **dates** в текущем контексте.

## Синтаксис

```
NEXTMONTH(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- Эта функция возвращает все даты со следующего дня до первой даты во входном параметре. Например, если первой датой в аргументе **dates** является 10 июня 2009 г, эта функция возвращает все даты июня 2009 г.
- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция 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 образце формулы создается мера, которая вычисляет продажи через Интернет за следующий месяц.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTMONTH('DateTime'[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[Функция NEXTDAY](#)



Функция NEXTQUARTER

Функция NEXTYEAR

# NEXTQUARTER

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец со всеми датами следующего квартала на основе первой даты в столбце **dates** в текущем контексте.

## Синтаксис

```
NEXTQUARTER(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- Эта функция возвращает все даты следующего квартала на основе первой даты во входном параметре. Например, если первой датой в столбце **dates** является 10 июня 2009 г, эта функция возвращает все даты с июля по сентябрь 2009 г.
- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция 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 через Интернет за следующий квартал.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTQUARTER('DateTime'[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[Функция NEXTDAY](#)

NEXTMONTH, функция

Функция NEXTYEAR

# NEXTYEAR

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец со всеми датами следующего года на основе первой даты в столбце `dates` в текущем контексте.

## Синтаксис

```
NEXTYEAR(<dates>[,<year_end_date>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.
year_end_date	(необязательно) Литеральная строка с датой, которая определяет дату окончания года. Значение по умолчанию — 31 декабря.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- Эта функция возвращает все даты следующего года на основе первой даты во входном столбце. Например, если первая дата в столбце `dates` указывает 2007 г, эта функция возвращает все даты 2008 г.
- Аргумент `dates` может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Параметр `year_end_date` — это строковый литерал даты в языковом стандарте клиента, в котором была создана книга. Компонент года в дате 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 образце формулы создается мера, которая вычисляет продажи через Интернет за следующий год.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTYEAR('DateTime'[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[Функция NEXTDAY](#)

[Функция NEXTQUARTER](#)

[NEXTMONTH, функция](#)

# OPENINGBALANCEMONTH

02.10.2020 • 2 minutes to read

Вычисляет **выражение** в первый день месяца в текущем контексте.

## Синтаксис

```
OPENINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Выражение, возвращающее скалярное значение.
даты	Столбец, содержащий даты.
фильтр	(необязательно) Выражение, задающее фильтр, применяемый к текущему контексту.

## Возвращаемое значение

Скалярное значение, представляющее **выражение**, которое вычисляется в первый день месяца в текущем контексте.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Выражение **filter** имеет ограничения, описанные в разделе [Функция 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 значение стоимости складских запасов по состоянию на начало месяца.

```
=  
OPENINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[  
DateKey])
```

## См. также

[Функция OPENINGBALANCEYEAR](#)

[Функция OPENINGBALANCEQUARTER](#)

[Функции операций со временем](#)

[CLOSINGBALANCEMONTH](#), функция

# OPENINGBALANCEQUARTER

02.10.2020 • 2 minutes to read

Вычисляет **выражение** в первый день квартала в текущем контексте.

## Синтаксис

```
OPENINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Выражение, возвращающее скалярное значение.
даты	Столбец, содержащий даты.
filter*	(необязательно) Выражение, задающее фильтр, применяемый к текущему контексту.

## Возвращаемое значение

Скалярное значение, представляющее **выражение**, которое вычисляется в первый день квартала в текущем контексте.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Выражение **filter** имеет ограничения, описанные в разделе [Функция 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 значение стоимости складских запасов по состоянию на начало квартала.

```
=  
OPENINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTim  
e[DateKey])
```



## См. также

[Функция OPENINGBALANCEYEAR](#)

[OPENINGBALANCEMONTH](#), функция

[Функции операций со временем](#)

[Функция CLOSINGBALANCEQUARTER](#)

# OPENINGBALANCEYEAR

02.10.2020 • 2 minutes to read

Вычисляет выражение `expression` в первый день года в текущем контексте.

## Синтаксис

```
OPENINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Выражение, возвращающее скалярное значение.
даты	Столбец, содержащий даты.
фильтр	(необязательно) Выражение, задающее фильтр, применяемый к текущему контексту.
year_end_date	(необязательно) Литеральная строка с датой, которая определяет дату окончания года. Значение по умолчанию — 31 декабря.

## Возвращаемое значение

Скалярное значение, представляющее **выражение**, которое вычисляется в первый день года в текущем контексте.

## Remarks

- Аргумент `dates` может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Выражение `filter` имеет ограничения, описанные в разделе [Функция CALCULATE](#).
- Параметр `year_end_date` — это строковый литерал даты в языковом стандарте клиента, в котором была создана книга. Компонент года в дате 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 значение стоимости складских запасов по состоянию на начало года.

=

```
OPENINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

## См. также

[Функция OPENINGBALANCEQUARTER](#)

[OPENINGBALANCEMONTH, функция](#)

[Функции операций со временем](#)

[CLOSINGBALANCEYEAR, функция](#)

# PARALLELPERIOD

02.10.2020 • 4 minutes to read

Возвращает таблицу, содержащую столбец дат, который представляет период, параллельный датам в указанном столбце **dates**, в текущем контексте, с датами, сдвинутыми на указанное число интервалов вперед или назад во времени.

## Синтаксис

```
PARALLELPERIOD(<dates>,<number_of_intervals>,<interval>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.
number_of_intervals	Целое число, указывающее количество интервалов, добавляемых в даты или вычитаемых из них.
interval	Интервал для сдвига дат. Значение параметра interval может быть одним из следующих: <code>year</code> , <code>quarter</code> , <code>month</code> .

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- Эта функция принимает текущий набор дат в столбце, указанном параметром **dates**, сдвигает первую и последнюю дату на указанное число интервалов, а затем возвращает все смежные даты между двумя сдвинутыми датами. Если интервал является частичным диапазоном месяца, квартала или года, то все неполные месяцы в результатах также заполняются для завершения всего интервала.
- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция CALCULATE](#).
- Если число, указанное для **number\_of\_intervals**, положительное, даты в **dates** перемещаются вперед во времени. Если число отрицательное, даты в **dates** сдвигаются назад во времени.
- Параметр **interval** является перечислением, а не набором строк. Поэтому значения не должны заключаться в кавычки. Кроме того, значения `year`, `quarter`, `month` должны быть записаны полностью при использовании.
- Таблица результатов содержит только даты, которые отображаются в значениях столбца базовой

таблицы.

- Функция PARALLELPERIOD похожа на функцию DATEADD, за исключением того, что PARALLELPERIOD всегда возвращает полные периоды на заданном уровне детализации вместо частичных периодов, возвращаемых функцией DATEADD. Например, если у вас есть набор дат, которые начинаются 10 июня и заканчиваются 21 июня того же года, и вы хотите сместить этот интервал вперед на один месяц, функция PARALLELPERIOD возвратит все даты следующего месяца (с 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 через Интернет за предыдущий год.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PARALLELPERIOD(DateTime[DateKey],-1,year))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[DATEADD, функция](#)

# PREVIOUSDAY

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец со всеми датами, представляющими день, предшествующий первой дате в столбце `dates`, в текущем контексте.

## Синтаксис

```
PREVIOUSDAY(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- Эта функция определяет первую дату во входном параметре, а затем возвращает все даты, соответствующие дню, предшествующему первой дате. Например, если первой датой в аргументе `dates` является 10 июня 2009 г, эта функция возвращает все даты, равные 9 июня 2009 г.
- Аргумент `dates` может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения логических выражений описаны в разделе [Функция 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 через Интернет за предыдущий день.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSDAY('DateTime'[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[PREVIOUSMONTH, функция](#)

Функция PREVIOUSQUARTER

Функция PREVIOUSYEAR

# PREVIOUSMONTH

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец со всеми датами предыдущего месяца на основе первой даты в столбце **dates** в текущем контексте.

## Синтаксис

```
PREVIOUSMONTH(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты.	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- Эта функция возвращает все даты из предыдущего месяца, используя первую дату в столбце, заданном в качестве входных данных. Например, если первой датой в аргументе **dates** является 10 июня 2009 г, эта функция возвращает все даты мая 2009 г.
- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция 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 через Интернет за предыдущий месяц.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSMONTH('DateTime'[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[PREVIOUSDAY](#)



PREVIOUSQUARTER  
PREVIOUSYEAR

# PREVIOUSQUARTER

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец со всеми датами предыдущего квартала на основе первой даты в столбце **dates** в текущем контексте.

## Синтаксис

```
PREVIOUSQUARTER(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- Эта функция возвращает все даты из предыдущего квартала, используя первую дату в столбце входных данных. Например, если первой датой в аргументе **dates** является 10 июня 2009 г, эта функция возвращает все даты за квартал с января по март 2009 г.
- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция 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 через Интернет за предыдущий квартал.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSQUARTER('DateTime'[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[PREVIOUSMONTH](#)

PREVIOUSDAY  
PREVIOUSYEAR

# PREVIOUSYEAR

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец со всеми датами предыдущего года на основе последней даты в столбце `dates` в текущем контексте.

## Синтаксис

```
PREVIOUSYEAR(<dates>[, <year_end_date>])
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.
year_end_date	(необязательно) Литеральная строка с датой, которая определяет дату окончания года. Значение по умолчанию — 31 декабря.

## Возвращаемое значение

Таблица, содержащая один столбец значений даты.

## Remarks

- Эта функция возвращает все даты с предыдущего указанного года до последней даты во входном параметре. Например, если последняя дата в аргументе `dates` ссылается на 2009 год, эта функция возвращает все даты для 2008 года, вплоть до указанного `year_end_date`.
- Аргумент `dates` может принимать любое из следующих значений:
  - Ссылка на столбец даты и времени.
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция CALCULATE](#).
- Параметр `year_end_date` — это строковый литерал даты в языковом стандарте клиента, в котором была создана книга. Компонент года в дате 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 через Интернет за предыдущий год.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSYEAR('DateTime'[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[PREVIOUSMONTH](#)

[PREVIOUSDAY](#)

[PREVIOUSQUARTER](#)

# SAMEPERIODLASTYEAR

02.10.2020 • 2 minutes to read

Возвращает таблицу, содержащую столбец дат со сдвигом на год назад от дат в столбце `dates` в текущем контексте.

## Синтаксис

```
SAMEPERIODLASTYEAR(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
dates	Столбец, содержащий даты.

## Возвращаемое значение

Таблица значений дат с одним столбцом.

## Remarks

- Аргумент `dates` может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция CALCULATE](#).
- Возвращаемые даты совпадают с датами, возвращаемыми следующей 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 образце формулы создается мера, которая вычисляет продажи через посредников за предыдущий год.

```
= CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]), SAMEPERIODLASTYEAR(DateTime[DateKey]))
```

## См. также

[Функции операций со временем](#)

[Функции даты и времени](#)

[PREVIOUSYEAR](#)

[PARALLELPERIOD](#)

# STARTOFMONTH

02.10.2020 • 2 minutes to read

Возвращает первую дату месяца в текущем контексте для указанного столбца дат.

## Синтаксис

```
STARTOFMONTH(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция 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 образце формулы создается мера, которая возвращает начало месяца для текущего контекста.

```
= STARTOFMONTH(DateTime[DateKey])
```

## См. также

[Функции даты и времени](#)

[Функции операций со временем](#)

[STARTOFYEAR](#)

[STARTOFQUARTER](#)

# STARTOFQUARTER

02.10.2020 • 2 minutes to read

Возвращает первую дату квартала в текущем контексте для указанного столбца дат.

## Синтаксис

```
STARTOFQUARTER(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция 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 возвращает начало квартала для текущего контекста.

```
= STARTOFQUARTER(DateTime[DateKey])
```

## См. также

[Функции даты и времени](#)

[Функции операций со временем](#)

[STARTOFYEAR](#)

[STARTOFMONTH](#)



# STARTOFYEAR

02.10.2020 • 2 minutes to read

Возвращает первую дату года в текущем контексте для указанного столбца дат.

## Синтаксис

```
STARTOFYEAR(<dates>)
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
даты	Столбец, содержащий даты.
YearEndDate	(необязательно) Значение даты окончания года.

## Возвращаемое значение

Таблица, содержащая один столбец и одну строку со значением даты.

## Remarks

- Аргумент `dates` может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция 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 формула создает меру, которая возвращает начало года для текущего контекста.

```
= STARTOFYEAR(DateTime[DateKey])
```

## См. также

[Функции даты и времени](#)

[Функции операций со временем](#)

[STARTOFQUARTER](#)

[STARTOFMONTH](#)

# TOTALMTD

02.10.2020 • 2 minutes to read

Вычисляет значение **выражения** для дат в месяце до текущей даты в текущем контексте.

## Синтаксис

```
TOTALMTD(<expression>,<dates>[,<filter>])
```

### Параметры

ПАРАМЕТР	ОПРЕДЕЛЕНИЕ
expression	Выражение, возвращающее скалярное значение.
даты	Столбец, содержащий даты.
фильтр	(необязательно) Выражение, задающее фильтр, применяемый к текущему контексту.

## Возвращаемое значение

Скалярное значение, представляющее **выражение**, вычисленное для дат в текущем месяце до текущей даты, учитывая даты в dates.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция CALCULATE](#).
- Выражение **filter** имеет ограничения, описанные в разделе [Функция 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 образце формулы создается мера, которая вычисляет "промежуточный итог месяца" и "промежуточную сумму месяца" для продаж через Интернет.

```
= TOTALMTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

## См. также

[ALL](#)

CALCULATE  
TOTALYTD  
TOTALQTD

# TOTALQTD

02.10.2020 • 2 minutes to read

Вычисляет значение **выражения** для дат в квартале до текущей даты в текущем контексте.

## Синтаксис

```
TOTALQTD(<expression>,<dates>[,<filter>])
```

### Параметры

ПАРАМЕТР	ОПРЕДЕЛЕНИЕ
expression	Выражение, возвращающее скалярное значение.
даты	Столбец, содержащий даты.
фильтр	(необязательно) Выражение, задающее фильтр, применяемый к текущему контексту.

## Возвращаемое значение

Скалярное значение, представляющее **выражение**, вычисленное для всех дат в текущем квартале до текущей даты, учитывая даты в **dates**.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - ссылка на столбец даты и времени;
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция CALCULATE](#).
- Выражение **filter** имеет ограничения, описанные в разделе [Функция 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 образце формулы создается мера, которая вычисляет "промежуточный итог квартала" и "промежуточную сумму квартала" для продаж через Интернет.

```
= TOTALQTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

## См. также

[ALL](#)

CALCULATE  
TOTALYTD  
TOTALMTD

# TOTALYTD

02.10.2020 • 2 minutes to read

Вычисляет значение с начала года для **выражения** в текущем контексте.

## Синтаксис

```
TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])
```

### Параметры

ПАРАМЕТР	ОПРЕДЕЛЕНИЕ
expression	Выражение, возвращающее скалярное значение.
даты	Столбец, содержащий даты.
фильтр	(необязательно) Выражение, задающее фильтр, применяемый к текущему контексту.
year_end_date	(необязательно) Литеральная строка с датой, которая определяет дату окончания года. Значение по умолчанию — 31 декабря.

## Возвращаемое значение

Скалярное значение, представляющее **выражение** вычисляемое для текущего значения **dates** с начала года.

## Remarks

- Аргумент **dates** может принимать любое из следующих значений:
  - Ссылка на столбец даты и времени.
  - табличное выражение, возвращающее один столбец значений даты и времени;
  - логическое выражение, определяющее таблицу значений даты и времени из одного столбца.
- Ограничения, применяемые к логическим выражениям, описаны в разделе [Функция CALCULATE](#).
- Выражение **filter** имеет ограничения, описанные в разделе [Функция CALCULATE](#).
- Параметр **year\_end\_date** — это строковый литерал даты в языковом стандарте клиента, в котором была создана книга. Компонент года в дате не требуется и игнорируется. Например, в следующей формуле указывается конец (финансового) года **year\_end\_date 6/30 (30.06)** в рабочей книге с языковым стандартом EN-US.

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey], ALL('DateTime'), "6/30")
```

В этом примере **year\_end\_date** можно указать как "6/30", "Июн 30", "30 июня" или любую строку, которая разрешается в месяц или день. Однако рекомендуется указать **year\_end\_date**, используя

формат "месяц/день" (как показано), чтобы гарантировать, что строка разрешается в 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 образце формулы создается мера, которая вычисляет "промежуточный итог года" и "промежуточную сумму года" для продаж через Интернет.

```
= TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

## См. также

[ALL](#)

[CALCULATE](#)

[DATESYTD](#)

[TOTALMTD](#)

[TOTALQTD](#)

# Инструкции

22.09.2020 • 2 minutes to read

## В этой категории

	DESCRIPTION
DEFINE	(Ключевое слово) Определяет сущности, которые существуют только на протяжении выполнения запроса DAX.
EVALUATE	(Ключевое слово) Инструкция, которая необходима для выполнения запроса DAX.
ORDER BY	(Ключевое слово) Определяет одно или несколько выражений, используемых для сортировки результатов запроса DAX.
VAR	(Ключевое слово) Сохраняет результат выражения как именованную переменную, которую затем можно передать в качестве аргумента в другие выражения мер.



# DEFINE

22.09.2020 • 2 minutes to read

Ключевое слово, определяющее сущности, которые могут быть применены к одной или нескольким инструкциям EVALUATE [запроса DAX](#).

## Синтаксис

```
DEFINE { <entity> [<name>] = <expression> }
```

### Аргументы

ТЕРМИН	ОПРЕДЕЛЕНИЕ
сущность	MEASURE, VAR, TABLE или COLUMN.
name	Имя сущности. Этот параметр не может быть выражением.
expression	Любое выражение DAX, возвращающее одиночное скалярное значение. В выражении могут использоваться любые определенные сущности. Выражение должно возвращать таблицу. Если требуется скалярное значение, заключите скаляр в функцию ROW() для получения таблицы.

## Remarks

- Сущностями могут быть переменные, меры, таблицы и столбцы.
- Определения обычно предшествуют оператору EVALUATE и допустимы для всех инструкций EVALUATE.
- Определения могут ссылаться на другие определения, находящиеся до или после текущего определения.
- Определения существуют, только пока выполняется запрос.

## См. также раздел

[Запросы DAX](#)  
[ORDER BY](#)  
[VAR](#)

# EVALUATE

22.09.2020 • 2 minutes to read

Инструкция, которая содержит табличное выражение, требуемое для [запроса DAX](#).

## Синтаксис

```
EVALUATE <table>
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
table	Табличное выражение

## Возвращаемое значение

Результат табличного выражения.

## Remarks

Запрос может содержать несколько инструкций EVALUATE.

## Пример

```
EVALUATE(  
    'Internet Sales'  
)
```

Возвращает все строки и столбцы из таблицы Internet Sales в виде таблицы.

## См. также раздел

[Запросы DAX](#)

[DEFINE](#)

[ORDER BY](#)

# ORDER BY

22.09.2020 • 2 minutes to read

Определяет порядок сортировки результатов запроса, возвращаемых инструкцией EVALUATE в [запросе DAX](#).

## Синтаксис

```
ORDER BY {<expression> [{ASC | DESC}]}
```

### Аргументы

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Любое выражение DAX, возвращающее одиночное скалярное значение.
ASC	(По умолчанию) Порядок сортировки по возрастанию.
DESC	(По умолчанию) Порядок сортировки по убыванию.

## Возвращаемое значение

Результат выполнения инструкции EVALUATE в порядке возрастания (ASC) или убывания (DESC).

## Пример

```
EVALUATE(  
    'Internet Sales'  
)  
ORDER BY  
    'Internet Sales'[Order Date]
```

Возвращает все строки и столбцы из таблицы Internet Sales, упорядоченные по Order Date, в виде таблицы.

## См. также раздел

[Запросы DAX](#)

[EVALUATE](#)

# VAR

22.09.2020 • 3 minutes to read

Сохраняет результат выражения как именованную переменную, которую затем можно передать в качестве аргумента в другие выражения мер. Вычисленные итоговые значения для выражения переменной не изменяются, даже если на переменную существует ссылка в другом выражении.

## Синтаксис

```
VAR <name> = <expression>
```

### Параметры

ТЕРМИН	ОПРЕДЕЛЕНИЕ
name	Имя переменной (идентификатор). Разделители не поддерживаются. Например, "имя_переменной" или [имя_переменной] приведет к ошибке. Поддерживаемый набор символов: a–z, A–Z, 0–9. Символы 0–9 не являются допустимыми в качестве первого символа. __ (двойное подчеркивание) можно использовать в качестве префикса к имени идентификатора. Другие специальные символы не поддерживаются. Зарезервированные ключевые слова не допускаются. Имена существующих таблиц не допускаются. Пустые места не допускаются.
expression	Выражение DAX, возвращающее скалярное или табличное значение.

## Возвращаемое значение

Именованная переменная, содержащая результат аргумента выражения.

## Remarks

- Выражение, передаваемое в качестве аргумента функции VAR, может содержать другое объявление VAR.
- При ссылке на переменную:
  - Меры не могут ссылаться на переменные, определенные за пределами выражения меры, но могут ссылаться на переменные функциональной области, определенные в выражении.
  - Переменные могут ссылаться на меры.
  - Переменные могут ссылаться на ранее определенные переменные.
  - К столбцам в табличных переменных нельзя обращаться через синтаксис `ИмяТаблицы[ИмяСтолбца]`.

## Пример

Чтобы вычислить процентное значение роста за год, не используя переменную, можно создать три отдельные меры. Первая мера вычисляет сумму продаж:

```
Sum of SalesAmount = SUM(SalesTable[SalesAmount])
```

Вторая мера вычисляет объем продаж за предыдущий год.

```
SalesAmount PreviousYear =  
    CALCULATE([Sum of SalesAmount],  
        SAMEPERIODLASTYEAR(Calendar[Date])  
    )
```

Затем можно создать третью меру, объединяющую две другие меры для вычисления процента роста. Обратите внимание, что мера суммы продаж используется в двух местах: сначала для определения того, была ли продажа, а затем снова, чтобы вычислить процент.

```
Sum of SalesAmount YoY%: =  
    IF([Sum of SalesAmount] ,  
        DIVIDE(([Sum of SalesAmount] - [SalesAmount PreviousYear]), [Sum of SalesAmount])  
    )
```

С помощью переменной можно создать единую меру, которая вычисляет тот же результат:

```
YoY% = VAR Sales = SUM(SalesTable[SalesAmount])  
  
VAR SalesLastYear =  
    CALCULATE ( SUM ( SalesTable[SalesAmount] ), SAMEPERIODLASTYEAR ( 'Calendar'[Date] ) )  
  
return if(Sales, DIVIDE(Sales - SalesLastYear, Sales))
```

С помощью переменной можно получить тот же результат, но в более удобочитаемом виде. Кроме того, результат выражения сохраняется в переменной при объявлении. Его не нужно пересчитывать каждый раз, когда он используется, в отличие от ситуации без использования переменной. Это может повысить производительность меры.

# Глоссарий по DAX

02.10.2020 • 13 minutes to read

## Аналитический запрос

Визуальные элементы Power BI запрашивают модель данных с помощью *аналитических запросов*. Аналитический запрос стремится уменьшить потенциально большой объем данных и упростить модель с помощью трех отдельных этапов: фильтрация, группирование и обобщение. Аналитический запрос создается автоматически, если поля назначены ячейкам визуальных элементов отчета. Авторы отчетов могут управлять поведением назначений полей путем переименования полей, изменения способа формирования сводных данных или отключения формирования сводных данных для группирования. Во время разработки отчета к самому отчету, странице отчета или визуальному элементу можно добавлять фильтры. В режиме чтения фильтры можно изменить на панели **Фильтры** или с помощью взаимодействия со срезами и другими визуальными элементами (перекрестная фильтрация).

## BLANK

DAX определяет отсутствие значения как BLANK. Это эквивалент NULL в SQL, но его поведение немного отличается. Это больше похоже на способ определения пустой ячейки в Excel. BLANK вычисляется как ноль или пустая строка в сочетании с другими операциями. Например,  $BLANK + 20 = 20$ . Всегда используйте прописные буквы; во множественном числе — BLANKs со строчной s.

## Вычисляемый столбец

Вычисление модели, используемое для добавления столбца в табличную модель путем написания формулы DAX. Формула должна возвращать скалярное значение и оцениваться для каждой строки в таблице. Вычисляемый столбец можно добавить в таблицу с режимом хранения "Импорт" или "DirectQuery".

## Вычисляемая мера

В табличном моделировании нет такого понятия как *вычисляемая мера*. Вместо этого используйте *меру*. Слово *вычисляемый* используется для описания вычисляемых таблиц и вычисляемых столбцов. Этот термин отличает их от таблиц и столбцов, взятых из Power Query. В Power Query нет понятия меры.

## Вычисляемая таблица

Вычисление модели, используемое для добавления таблицы в табличную модель путем написания формулы DAX. Формула должна возвращать объект таблицы. В результате создается таблица, использующая режим хранения "Импорта".

## Вычисление

Намеренный процесс, который преобразует один или несколько входных элементов в один или несколько результатов. В табличной модели данных вычисление может быть объектом модели — вычисляемой таблицей, вычисляемым столбцом или мерой.

## Контекст

Описывает среду, в которой вычисляется формула DAX. Существуют два типа контекста: *контекст строки* и *контекст фильтра*. Контекст строки представляет "текущую строку" и используется для вычисления формул

вычисляемых столбцов и выражений, используемых итераторами таблиц. Контекст фильтра используется для оценки мер и представляет фильтры, применяемые непосредственно к столбцам модели, и фильтры, распространяемые связями модели.

## Куб

См. [Многомерная модель](#).

## Модель данных

Ресурс данных, специально подготовленный для отчетности и аналитики. Она позволяет пользователям отчетов просматривать и изучать данные простым и интуитивно понятным способом. Важно отметить, что она обеспечивает высокую производительность результатов запросов даже для больших объемов данных. Она может интегрировать данные из нескольких источников и использовать вычисления для преобразования данных. Она может применять разрешение на уровне строк, чтобы предоставить разным пользователям доступ к разным данным. Иногда она называется *семантической моделью* или просто *моделью*.

## Разработчик модели данных

Квалифицированный специалист по данным, создающий модели данных. В средстве бизнес-аналитики с самообслуживанием (SSBI) они могут называться бизнес-аналитиками. В корпоративной бизнес-аналитике они могут называться разработчиками бизнес-аналитики.

## DAX

Язык выражений анализа данных (DAX) — это язык формул для Power Pivot в Excel, Power BI, Azure Analysis Services и табличного моделирования в SQL Server Analysis Services. DAX также можно использовать для добавления вычислений модели данных и определения правил безопасности на уровне строк (RLS).

## Динамическая безопасность

При применении правил безопасности на уровне строк (RLS) с помощью удостоверения пользователя отчета. Правила фильтруют таблицы моделей с помощью имени учетной записи пользователя, для чего можно использовать функции USERNAME или USERPRINCIPALNAME. См. статью [Безопасность на уровне строк](#).

## Expression

Единица логики DAX, которая вычисляется и возвращает результат. Выражения могут объявлять переменные. При этом им назначается вложенное выражение, и они должны включать инструкцию RETURN, которая выводит окончательное выражение. Выражения состоят из объектов модели (таблиц, столбцов или мер), функций, операторов или констант.

## Поле

Ресурс модели данных, представленный в области **Поля**. Поля используются для настройки фильтров и визуальных элементов отчетов. Поля состоят из столбцов модели, уровней иерархии и мер.

## Формула

Одно или несколько выражений DAX, используемых для определения вычисления модели. Внутренние выражения называются вложенными выражениями. Во множественном числе это *формулы*.

## Компонент

Функции DAX имеют аргументы, позволяющие передавать параметры. Формулы могут использовать множество вызовов функций, даже вложенных функций в других функциях. В формуле после имен функций следует указывать круглые скобки. В круглых скобках передаются параметры.

## Неявная мера

Автоматически созданное вычисление достигается путем настройки визуального элемента Power BI для суммирования значений столбцов. **Числовые** столбцы поддерживают больше всего методов подсчета сводных данных, в том числе: Sum, Average, Minimum, Maximum, Count (Distinct), Count, Standard deviation, Variance и Median. Рассчитывать сводные данные можно и для других типов данных. **Текстовые** столбцы поддерживают вычисление сводных данных с помощью функций: First (alphabetically), Last (alphabetically), Count (Distinct) и Count. Столбцы **данных** поддерживают вычисление сводных данных с помощью функций: Earliest, Latest, Count (Distinct) и Count. **Логические** столбцы поддерживают вычисление сводных данных с помощью функций: 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 возможности управления обработкой сводных данных в вычислениях модели.

## MDX

Язык многомерных выражений (MDX) — это язык формул для многомерных моделей SQL Server Analysis Services (которые также называются *кубами*). MDX может использоваться для запроса табличных моделей, однако он не может определять неявные меры. Он может запрашивать только те меры, которые уже определены в модели.

## Measure

Вычисление, которое приводит к сводным данным. Меры бывают *неявными* и *явными*. Явная мера — это вычисление, добавляемое в табличную модель данных путем написания формулы DAX. Формула меры должна возвращать скалярное значение. В области **Поля** явные меры отмечены значком калькулятора. Явные меры необходимы, если запрос к модели написан на языке многомерных выражений (MDX), как в случае использования функции анализа в Excel. Явная мера обычно называется просто мерой.

## Группа мер

Таблица модели, которая содержит по крайней мере одну меру и не имеет иерархий или видимых столбцов. В области **Поля** каждая группа мер обозначена значком с несколькими калькуляторами. Группы мер перечислены в верхней части области **Поля** и сортируются в алфавитном порядке по имени.

## Модель

См. [Модель данных](#).

## Разработчик модели

См. [Разработчик модели данных](#).

## Вычисление модели



Именованная формула, используемая для добавления вычисляемой таблицы, вычисляемого столбца или меры в табличную модель данных. Она имеет структуру `<NAME> = <FORMULA>`. Большинство вычислений добавляются разработчиками моделей данных в Power BI Desktop, но меры также можно добавить в отчет с динамическим подключением. См. [Меры отчета](#).

## Многомерная модель

Модель данных, разработанная для SQL Server Analysis Services (многомерный режим). Она состоит из измерений и мер. Часто ее называют просто *куб*.

## Быстрые меры

Функция в Power BI Desktop, которая устраняет необходимость написания формул DAX для часто определяемых мер. Быстрые меры включают в себя среднее значение для категории, ранжирования и отличие от базовых показателей.

## Меры отчета

Также называются *мерами на уровне отчета*. Они добавляются в отчет с динамическим подключением в Power BI Desktop путем написания формулы DAX, но только для соединений с моделями Power BI или табличными моделями Analysis Services.

## Безопасность на уровне строк

Также называется *RLS*. Методика проектирования для ограничения доступа к подмножествам данных для конкретных пользователей. В табличной модели это достигается путем создания ролей модели. У ролей есть правила, которые являются выражениями DAX для фильтрации строк таблицы.

## Семантическая модель

См. [Модель данных](#).

## Сводка

Операция, применяемая к значениям столбца. См. [мера](#).

## Табличный куб

Нет существует такого понятия, как *табличный куб*. Это называется [табличная модель](#).

## Табличная модель

Модель данных, разработанная в Power Pivot в Excel, Power BI, Azure Analysis Services или SQL Server Analysis Services (табличный режим).

## логика операций со временем

Логика операций со временем относится к вычислениям с течением времени, например с начала года (YTD).

## Функция логики операций со временем

В DAX имеется множество функций логики операций со временем. Каждая функция логики операций со временем производит результат путем изменения контекста фильтра для фильтров дат. Примеры функций: TOTALYTD и SAMEPERIODLASTYEAR.

## Значение, значения

Данные для визуализации.

## Параметр What-if

Компонент Power BI Desktop, позволяющий принимать данные, вводимые пользователем, с помощью срезов. Каждый параметр создает вычисляемую таблицу с одним столбцом и меру, возвращающую одно значение. Мера может использоваться в вычислениях модели для реагирования на входные данные пользователя.

# Операторы DAX

17.11.2020 • 12 minutes to read

В языке выражений анализа данных (DAX) операторы используются для создания выражений, которые сравнивают значения, выполняют арифметические вычисления или работают со строками.

## Типы операторов

Существует четыре различных типа вычислительных операторов: арифметические, сравнения, объединения текста и логические.

### Арифметические операторы

Для выполнения основных математических операций, таких как сложение, вычитание или умножение, объединения чисел и получения числовых результатов применяются перечисленные ниже арифметические операторы.

АРИФМЕТИЧЕСКИЙ ОПЕРАТОР	ЗНАЧЕНИЕ	ПРИМЕР
+ (знак плюса)	Сложение	3+3
– (знак минуса)	Вычитание или знак числа	3–1–1
* (звездочка)	Умножение	3*3
/ (косая черта)	Деление	3/3
^ (крышка)	Возведение в степень	16^4

#### NOTE

Знак плюса может функционировать и как *бинарный*, и как *унарный оператор*. Бинарный оператор требует числа с обеих сторон и выполняет их сложение. При использовании значений в формуле DAX с обеих сторон бинарного оператора DAX пытается привести их к числовым типам данных, если они нечисловые. Напротив, унарный оператор можно применять к аргументу любого типа. Символ "плюс" не влияет на тип или значение и просто игнорируется, в то время как оператор "минус", примененный к числовому значению, делает его отрицательным.

### Операторы сравнения

Сравнить два значения можно с помощью приведенных ниже операторов. Результатом является логическое значение (TRUE или FALSE).

ОПЕРАТОР СРАВНЕНИЯ	ЗНАЧЕНИЕ	ПРИМЕР
=	Равно	[Регион] = "Россия"
==	Строго равно	[Регион] == "Россия"
>	Больше чем	[Дата продажи] > "янв 2009"
<	Меньше чем	[Дата продажи] < "1 янв 2009"

ОПЕРАТОР СРАВНЕНИЯ	ЗНАЧЕНИЕ	ПРИМЕР
> =	Больше или равно	[Объем] > = 20000
< =	Меньше или равно	[Объем] < = 100
< >	Не равно	[Регион] < > "Россия"

Все операторы сравнения, кроме ==, считают значение BLANK равным числу 0, пустой строке "", дате DATE(1899, 12, 30) или значению FALSE. Поэтому выражение [Столбец] = 0 будет иметь значение true, если значение [Столбец] равно 0 или BLANK. Для сравнения [Столбец] == 0 имеет значение true, только когда значение [Столбец] равно 0.

**Оператор объединения текста**

Используйте амперсанд ( & ) для объединения или сцепления двух или нескольких текстовых строк в один фрагмент текста.

ОПЕРАТОР ДЛЯ РАБОТЫ С ТЕКСТОМ	ЗНАЧЕНИЕ	ПРИМЕР
& (амперсанд)	Соединяет или сцепляет два значения для получения одного непрерывного текстового значения.	[Регион] & ", " & [Город]

**Логические операторы**

Используйте логические операторы (&&) и (||) для объединения выражений с целью получить один результат.

ОПЕРАТОР ДЛЯ РАБОТЫ С ТЕКСТОМ	ЗНАЧЕНИЕ	ПРИМЕРЫ
&& (двойной амперсанд)	Объединяет два выражения, каждое из которых имеет логический результат, условием И. Если оба выражения возвращают TRUE, комбинация выражений также возвращает значение TRUE. В противном случае комбинация возвращает FALSE.	(([Регион] = "Франция") && ([Покупатель] = "да"))
(двойной символ вертикальной черты)	Объединяет два выражения условием ИЛИ. Если любое из выражений возвращает значение TRUE, результатом будет TRUE. Результат будет FALSE, только если оба выражения имеют значение FALSE.	(([Регион] = "Франция")    ([Покупатель] = "да"))
IN	Объединяет все строки, сравниваемые с таблицей, логическим условием ИЛИ. Обратите внимание, что в синтаксисе конструктора таблицы используются фигурные скобки.	'Продукт'[Цвет] IN { "Красный", "Синий", "Черный" }

**Операторы и порядок приоритета**

В некоторых случаях порядок, в котором выполняется вычисление, может влиять на возвращаемое

значение. Поэтому важно понимать, как определяется порядок и как можно изменить его, чтобы получить нужные результаты.

### Порядок вычисления

Выражение вычисляет операторы и значения в определенном порядке. Выражения всегда начинаются со знака равенства (=). Знак равенства указывает на то, что последующие символы образуют выражение.

После знака равенства находятся вычисляемые элементы (операнды), разделенные вычислительными операторами. Выражения всегда считываются слева направо, но порядок, в котором группируются элементы, можно в некоторой степени контролировать с помощью круглых скобок.

### Приоритет операторов

Если в одной формуле имеется несколько операторов, операции производятся в порядке, указанном в таблице ниже. Если операторы имеют одинаковый приоритет, они выполняются слева направо. Например, если выражение содержит оператор умножения и оператор деления, они вычисляются в том порядке, в котором они следуют в выражении слева направо.

ОПЕРАТОР	ОПИСАНИЕ
$\wedge$	Возведение в степень
$-$	Знак (как в $-1$ )
$*$ и $/$	Умножение и деление
$!$	NOT (унарный оператор)
$+$ и $-$	Сложение и вычитание
$\&$	Объединяет две строки текста (сцепление)
$=, =, <, >, <=, >=, <>$	Сравнение

### Управление порядком вычисления с помощью круглых скобок

Чтобы изменить порядок вычисления, следует заключить в круглые скобки часть формулы, которая должна быть вычислена в первую очередь. Например, приведенная ниже формула возвращает значение 11, так как умножение выполняется до сложения. Формула умножает 2 на 3, а затем прибавляет 5 к результату.

`=5+2*3`

Если же добавить круглые скобки, порядок изменится: сначала складываются числа 5 и 2, а затем результат умножается на 3, и получается 21.

`=(5+2)*3`

В приведенном ниже примере из-за круглых скобок вокруг первой части формулы сначала вычисляется выражение `(3 + 0.25)`, а затем результат делится на результат выражения `(3 - 0.25)`.

`=(3 + 0.25)/(3 - 0.25)`

В приведенном ниже примере согласно правилам приоритета операторов сначала применяется оператор

возведения в степень, а затем оператор знака. Результат этого выражения равен -4.

```
= -2^2
```

Чтобы сначала применить оператор знака к числовому значению, можно использовать круглые скобки, как показано в примере ниже. Результат этого выражения равен 4.

```
= (-2)^2
```

## Совместимость

DAX легко обрабатывает и сравнивает различные типы данных, так же как Microsoft Excel. Однако вычислительная подсистема основана на SQL Server Analysis Services и предоставляет дополнительные возможности реляционного хранилища данных, включая расширенную поддержку типов даты и времени. Поэтому в некоторых случаях результаты вычислений или поведение функций могут отличаться от Excel. Кроме того, DAX поддерживает больше типов данных, чем Excel. В этом разделе описаны основные различия.

### Приведение типов данных операндов

Как правило, операнды слева и справа от любого оператора должны иметь один и тот же тип данных. Однако если типы данных различаются, DAX в некоторых случаях преобразует их в общий тип данных для применения оператора.

1. Оба операнда преобразуются в самый большой из возможных общих типов данных.
2. Затем применяется оператор, если это возможно.

Например, предположим, что необходимо перемножить два числа. Одно из них получено из формулы, например `= [Price] * .20`, и результат может иметь много знаков после запятой. Другое является целым числом, предоставленным как строковое значение.

В этом случае DAX преобразует оба числа в вещественные, используя самый большой числовой формат, позволяющий хранить оба числа. Затем DAX выполняет умножение.

В зависимости от комбинации типов данных приведение типов может не применяться при операциях сравнения. Полный список типов данных, поддерживаемых DAX, см. в статьях [Типы данных, поддерживаемые в табличных моделях](#) и [Типы данных в Power BI Desktop](#).

При сравнении числовыми считаются типы Integer, Real Number, Currency, Date/time и Blank. При выполнении сравнения значение Blank считается равным нулю. Для операций сравнения поддерживаются перечисленные ниже комбинации типов данных.

ТИП ДАННЫХ В ЛЕВОЙ ЧАСТИ	ТИП ДАННЫХ В ПРАВОЙ ЧАСТИ
Числовой	Числовой
Логический	Логический
Строка	Строка

Другие операции сравнения со смешанными типами данных возвращают ошибку. Например, формула `= "1" > 0` возвращает ошибку *Операции сравнения DAX не поддерживают сравнение значений типа Text со значениями типа Integer*.

ТИПЫ ДАННЫХ, ИСПОЛЬЗУЕМЫЕ В DAX	ТИПЫ ДАННЫХ, ИСПОЛЬЗУЕМЫЕ В EXCEL
Числа (I8, R8)	Числа (R8)
Строка	Строка
Логический	Логический
DateTime	Variant
Валюта	Валюта

### Различия в порядке приоритета

Порядок приоритета операций в формулах DAX в основном аналогичен тому, который используется в Microsoft Excel, но некоторые операторы Excel, например оператор процента, не поддерживаются. Кроме того, не поддерживаются диапазоны.

Поэтому при копировании и вставке формул из Excel внимательно проверяйте их, так как некоторые операторы или элементы могут быть недопустимыми. Если есть сомнения касательно очередности выполнения операций, рекомендуется использовать круглые скобки, чтобы устранить неоднозначность и получить ожидаемый результат.

### См. также:

[Синтаксис DAX](#)

[Именованние параметров DAX](#)

# Запросы DAX

02.10.2020 • 7 minutes to read

С помощью запросов DAX можно запрашивать и получать данные, определенные в виде табличного выражения. Клиенты отчетов создают запросы DAX каждый раз, когда в область отчета помещается поле или когда применяется фильтр или вычисление. Запросы DAX можно также создавать и выполнять в [SQL Server Management Studio](#) (SSMS) и в средствах с открытым кодом, таких как [DAX Studio](#). Запросы DAX выполняются в SSMS, и DAX Studio возвращает результаты в виде таблицы.

Прежде чем приступить к изучению запросов, важно иметь четкое представление об основах DAX. Если вы этого еще не сделали, обязательно ознакомьтесь с [общими сведениями о DAX](#).

## Синтаксис

```
[DEFINE { MEASURE <tableName>[<name>] = <expression> }  
        { VAR <name> = <expression> }]  
EVALUATE <table>  
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]  
[START AT {<value>|<parameter>} [, ...]]]
```

## Ключевые слова

### EVALUATE (обязательное)

На самом базовом уровне запрос DAX является инструкцией EVALUATE, содержащей табличное выражение. Однако запрос может содержать несколько инструкций EVALUATE.

#### Синтаксис

```
EVALUATE <table>
```

#### Аргументы

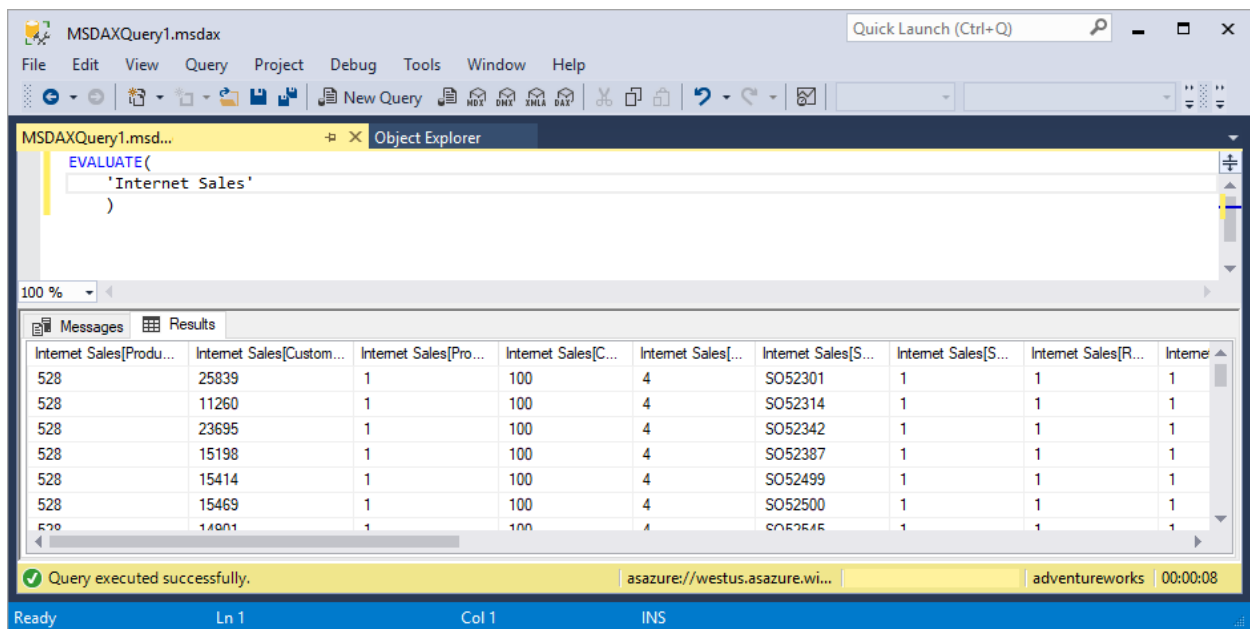
ТЕРМИН	ОПРЕДЕЛЕНИЕ
таблица	Табличное выражение.

#### Пример

```
EVALUATE(  
    'Internet Sales'  
)
```

Возвращает все строки и столбцы из таблицы Internet Sales в виде таблицы.





## ORDER BY (необязательное)

Необязательное ключевое слово **ORDER BY** определяет одно или несколько выражений, используемых для сортировки результатов запроса. Допускается любое выражение, которое можно вычислить для каждой строки результата.

### Синтаксис

```
EVALUATE <table>  
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]
```

### Аргументы

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Любое выражение DAX, возвращающее одиночное скалярное значение.
ASC	(По умолчанию) Порядок сортировки по возрастанию.
DESC	(По умолчанию) Порядок сортировки по убыванию.

### Пример

```
EVALUATE(  
    'Internet Sales'  
)  
ORDER BY  
    'Internet Sales'[Order Date]
```

Возвращает все строки и столбцы из таблицы Internet Sales, упорядоченные по Order Date, в виде таблицы.

MSDAXQuery1.msdx - Quick Launch (Ctrl+Q)

File Edit View Query Project Debug Tools Window Help

MSDAXQuery1.msdx... Object Explorer

```

EVALUATE(
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Order Date]

```

100 %

Internet Sales[S...	Internet Sales[T...	Internet Sales[Fr...	Internet Sales[C...	Internet Sales[C...	Internet Sales[Order Date]	Internet Sales[D...	Internet Sales[S...	Internet S...
99.99	271.9992	84.9998			12/29/2010 12:00:00 AM	1/10/2011 12:0...	1/5/2011 12:00...	1487.835
78.27	286.2616	89.4568			12/29/2010 12:00:00 AM	1/10/2011 12:0...	1/5/2011 12:00...	1406.975
99.99	271.9992	84.9998			12/29/2010 12:00:00 AM	1/10/2011 12:0...	1/5/2011 12:00...	1487.835
99.99	271.9992	84.9998			12/29/2010 12:00:00 AM	1/10/2011 12:0...	1/5/2011 12:00...	1487.835
9.0982	55.9279	17.4775			12/29/2010 12:00:00 AM	1/10/2011 12:0...	1/5/2011 12:00...	285.9515
99.99	271.9992	84.9998			12/30/2010 12:00:00 AM	1/11/2011 12:0...	1/6/2011 12:00...	1487.835
74.99	269.9992	84.3748			12/30/2010 12:00:00 AM	1/11/2011 12:0...	1/6/2011 12:00...	1476.895
78.27	286.2616	89.4568			12/30/2010 12:00:00 AM	1/11/2011 12:0...	1/6/2011 12:00...	1406.975

Query executed successfully. | azure://westus.azure.wi... | adventureworks | 00:00:08

Ready Ln 1 Col 1 INS

## START AT (необязательное)

Необязательное ключевое слово **START AT** используется в предложении **ORDER BY**. Оно определяет значение, с которого начинаются результаты запроса.

### Синтаксис

```

EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]
[START AT {<value>|<parameter>} [, ...]]

```

### Аргументы

ТЕРМИН	ОПРЕДЕЛЕНИЕ
значение	Значение-константа. Не может быть выражением.
parameter	Имя параметра в инструкции XMLA с префиксом @.

Аргументы **START AT** однозначно соответствуют столбцам в предложении **ORDER BY**. В предложении **START AT** может быть столько же аргументов, сколько в предложении **ORDER BY**, но не более. Первый аргумент в **START AT** определяет начальное значение в первом столбце предложения **ORDER BY**. Второй аргумент в **START AT** определяет начальное значение во втором столбце предложения **ORDER BY** в строках, соответствующих первому значению в первом столбце.

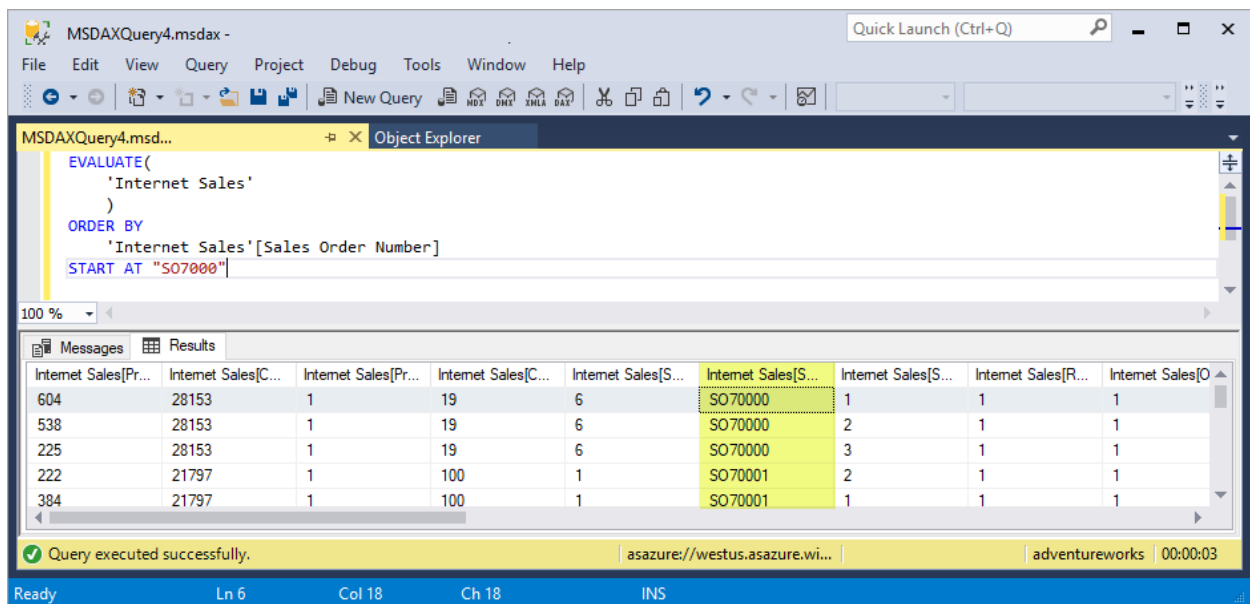
### Пример

```

EVALUATE(
    'Internet Sales'
)
ORDER BY
    'Internet Sales'[Sales Order Number]
START AT "SO7000"

```

Возвращает все строки и столбцы из таблицы **Internet Sales**, упорядоченные по **Sales Order Number**, начиная с **SO7000**.



В одном запросе можно указать несколько предложений EVALUATE/ORDER BY/START AT.

### DEFINE (необязательное)

Необязательное ключевое слово DEFINE определяет сущности, которые существуют только на протяжении выполнения запроса. Определения допустимы для всех инструкций EVALUATE. Сущностями могут быть переменные, меры, таблицы и столбцы. Определения могут ссылаться на другие определения, находящиеся до или после текущего определения. Определения обычно предшествуют инструкции EVALUATE.

#### Синтаксис

```
[DEFINE { MEASURE <tableName>[<name>] = <expression> }
        { VAR <name> = <expression>}]
EVALUATE <table>
```

#### Аргументы

ТЕРМИН	ОПРЕДЕЛЕНИЕ
tableName	Имя существующей таблицы с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.
name	Имя новой меры. Этот параметр не может быть выражением.
expression	Любое выражение DAX, возвращающее одиночное скалярное значение. В выражении могут использоваться любые определенные меры. Выражение должно возвращать таблицу. Если требуется скалярное значение, заключите скаляр в функцию ROW() для получения таблицы.
VAR	Необязательное выражение в виде именованной переменной. VAR можно передавать в качестве аргумента в другие выражения.

#### Пример

```

DEFINE
MEASURE 'Internet Sales'[Internet Total Sales] = SUM('Internet Sales'[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    'Date'[Calendar Year],
    TREATAS({2013, 2014}, 'Date'[Calendar Year]),
    "Total Sales", [Internet Total Sales],
    "Combined Years Total Sales", CALCULATE([Internet Total Sales], ALLSELECTED('Date'[Calendar Year]))
)
ORDER BY [Calendar Year]

```

Возвращает вычисленный общий объем продаж за 2013 и 2014 годы, а также совокупный вычисленный общий объем продаж за 2013 и 2014 годы в виде таблицы. Мера Internet Total Sales в инструкции DEFINE используется как в выражении Total Sales, так и в выражении Combined Years Total Sales.

The screenshot shows the SSDT interface with a DAX query in the Object Explorer. The query is as follows:

```

DEFINE
MEASURE 'Internet Sales'[Internet Total Sales] = SUM('Internet Sales'[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    'Date'[Calendar Year],
    TREATAS({2013, 2014}, 'Date'[Calendar Year]),
    "Total Sales", [Internet Total Sales],
    "Combined Years Total Sales", CALCULATE([Internet Total Sales], ALLSELECTED('Date'[Calendar Year]))
)
ORDER BY [Calendar Year]

```

The Results pane shows the following table:

Date[Calendar Y...	[Total Sales]	[Combined Years...
2013	16351550.34	16397245.06
2014	45694.72	16397245.06

The status bar at the bottom indicates "Query executed successfully." and "adventureworks | 00:00:01".

## Параметры в запросах DAX

Хорошо построенную инструкцию запроса DAX можно параметризовать, а затем использовать многократно, изменяя только значения параметров.

Метод **Execute (XMLA)** имеет коллекцию **элементов Parameters (XMLA)**, которая позволяет определять параметры и присваивать им значения. В этой коллекции каждый **элемент Parameter (XMLA)** определяет имя параметра и его значение.

Для ссылки на параметры XMLA необходимо указывать перед их именами префикс **@**. Вызов параметра можно использовать в любом месте, в котором синтаксис допускает значение. Все параметры XMLA имеют текстовый тип.

### IMPORTANT

Параметры, определенные в разделе параметров и не используемые в элементе **<STATEMENT>**, создают ответ об ошибке в XMLA. Параметры, которые используются, но не определены в элементе **<Parameters>**, создают ответ об ошибке в XMLA.

См. также:

FILTER  
SUMMARIZECOLUMNS  
TREATAS  
VAR

# Соглашения об именовании параметров DAX

17.11.2020 • 2 minutes to read

С целью упростить понимание и использование функций имени параметров в справочнике по DAX стандартизованы.

## Имена параметров

ТЕРМИН	ОПРЕДЕЛЕНИЕ
expression	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется несколько раз (для каждой строки или контекста).
value	Любое выражение DAX, возвращающее одно скалярное значение, в котором выражение вычисляется только один раз перед выполнением остальных операций.
table	Любое выражение DAX, возвращающее таблицу данных.
tableName	Имя существующей таблицы с использованием стандартного синтаксиса DAX. Этот параметр не может быть выражением.
columnName	Имя существующего столбца с использованием стандартного синтаксиса DAX (обычно полное имя). Этот параметр не может быть выражением.
name	Строковая константа, которая будет использоваться для указания имени нового объекта.
order	Перечисление, используемое для определения порядка сортировки.
ties	Перечисление, используемое для определения обработки связанных значений.
type	Перечисление, используемое для определения типа данных для PathItem и PathItemReverse.

## Использование префиксов перед именами параметров или отдельно

ТЕРМИН	ОПРЕДЕЛЕНИЕ
--------	-------------

ТЕРМИН	ОПРЕДЕЛЕНИЕ
prefixing	<p>Имена параметров можно уточнять с помощью префиксов, которые указывают, как используется аргумент, и позволяют избежать неоднозначного прочтения параметров. Пример:</p> <p>Result_ColumnName — указывает существующий столбец, используемый для получения результирующих значений в функции LOOKUPVALUE().</p> <p>Search_ColumnName — указывает существующий столбец, используемый для поиска значения в функции LOOKUPVALUE().</p>
omitting	<p>Имена параметров опускаются, если префикса достаточно для описания параметра.</p> <p>Например, вместо синтаксиса DATE(Год_Значение, Месяц_Значение, День_Значение) будет более понятным синтаксис DATE (Год, Месяц, День). Повторение суффикса "Значение" не делает функцию понятнее и лишь затрудняет ее восприятие.</p> <p>Однако, если параметр с префиксом имеет вид Год_имяСтолбца, то нужны и имя параметра, и префикс, чтобы пользователь понимал, что параметр должен указывать на существующий столбец с годами.</p>

# Синтаксис DAX

17.11.2020 • 15 minutes to read

В этой статье описывается синтаксис и требования для языка формул DAX.

## Требования к синтаксису

Формула DAX всегда начинается со знака равенства (=). После знака равенства можно указать любое выражение, результатом которого является скаляр, или выражение, которое можно преобразовать в скалярное. Вот некоторые из них.

- Скалярная константа или выражение, в котором используется скалярный оператор (+, -, \*, /, >=, <, & &, ...).
- Ссылки на столбцы или таблицы. В языке DAX в качестве входных данных функций всегда используются таблицы и столбцы, но не массивы и не произвольные наборы значений.
- Операторы, константы и значения, предоставляемые в составе выражения.
- Результат функции и ее обязательные аргументы. Некоторые функции DAX возвращают таблицу, а не скаляр. Их необходимо заключать в функцию, которая вычисляет таблицу и возвращает скаляр. Если таблица содержит один столбец и одну строку, то она рассматривается как скалярное значение.

Большинству функций DAX требуется один или несколько аргументов, которыми могут быть таблицы, столбцы, выражения и значения. Однако некоторые функции, такие как PI, не нуждаются в аргументах. В этом случае все равно необходимо использовать пустые круглые скобки. Например, необходимо вводить PI(), а не PI. Функции можно также вкладывать в другие функции.

- Выражения. Выражение может содержать все следующие элементы или некоторые из них: операторы, константы или ссылки на столбцы.

Например, все приведенные ниже формулы допустимы.

ФОРМУЛА	РЕЗУЛЬТАТ
= 3	3
= "Продажи"	Sales
= 'Продажи'[Объем]	Если использовать эту формулу в таблице "Продажи", будет получено значение текущей строки в столбце "Объем" этой таблицы.
= (0,03 *[Объем]) =0,03 * [Объем]	Три процента от значения в столбце "Объем" текущей таблицы.  Хотя эту формулу можно использовать для вычисления процента, результат не отображается в виде процента, если в таблице не применено форматирование.
= PI()	Значение константы Пи.



#### NOTE

Формулы могут вести себя по-разному в зависимости от того, используются ли они в вычисляемом столбце или в мере в сводной таблице. 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 образуют базу данных, хранящуюся в подсистеме аналитики в памяти (VertiPaq). В пределах базы данных все таблицы должны иметь уникальные имена. Имена столбцов также должны быть уникальными в пределах каждой таблицы. В именах всех объектов *регистр не учитывается*, например, имена **ПРОДАЖИ** и **Продажи** представляют одну и ту же таблицу.

Каждый столбец и каждая мера, добавляемые в существующую модель данных, должны принадлежать к определенной таблице. Таблица, содержащая столбец, указывается либо неявно при создании вычисляемого столбца в таблице, либо явно при создании меры. В последнем случае вы указываете имя таблицы, в которой должно храниться определение меры.

При использовании таблицы или столбца в качестве входных данных функции, как правило, необходимо указывать *полное* имя столбца. *Полное* имя столбца — это имя таблицы, за которым в квадратных скобках следует имя столбца, например 'Продажи в США'[Товары]. Полное имя требуется всегда при ссылке на столбец в следующих контекстах:

- в качестве аргумента функции VALUES;
- в качестве аргумента функции ALL или ALLEXCEPT;
- в аргументе фильтра для функции CALCULATE или CALCULATETABLE;
- в качестве аргумента функции RELATEDTABLE;
- в качестве аргумента любой функции логики операций со временем;

*Неполное* имя столбца — это просто имя столбца, заключенное в квадратные скобки, например [Объем продаж]. Например, неполное имя столбца можно использовать при ссылке на скалярное значение из той же строки текущей таблицы.

Если имя таблицы содержит пробелы, зарезервированные ключевые слова или недопустимые символы, его следует заключить в одинарные кавычки. Также необходимо заключать в кавычки имена таблиц, если они содержат символы вне диапазона буквенно-цифровых символов ANSI, независимо от того, поддерживает ли ваш языковой стандарт этот набор символов. Так, при открытии книги, содержащей имена таблиц на кириллице, например "Таблица", имя таблицы должно быть заключено в кавычки, даже если оно не содержит пробелов.

#### NOTE

Чтобы упростить ввод полных имен столбцов, используйте функцию автозаполнения в редакторе формул.

#### Таблицы

- Имена таблиц являются обязательными, если столбец относится не к текущей таблице. Имена таблиц должны быть уникальными в пределах базы данных.
- Имена таблиц должны заключаться в одинарные кавычки, если они содержат пробелы, другие специальные символы или буквенно-цифровые символы, не относящиеся к английскому алфавиту.

**Меры**

- Имена мер всегда должны заключаться в квадратные скобки.
- Имена мер могут содержать пробелы.
- Имя каждой меры должно быть уникальным в пределах модели. Поэтому при ссылке на существующую меру имя таблицы является необязательным перед именем меры. Однако при создании меры необходимо всегда указывать таблицу, в которой будет храниться определение меры.

**Столбцы**

Имена столбцов должны быть уникальными в контексте таблицы. Однако в нескольких таблицах могут быть столбцы с одинаковыми именами (неоднозначность устраняется посредством имени таблицы).

В общем случае на столбцы можно ссылаться, не указывая базовую таблицу, к которой они относятся. Исключением являются случаи, когда возможен конфликт имен, или определенные функции, требующие указания полных имен столбцов.

**Зарезервированные ключевые слова**

Если имя, используемое для таблицы, совпадает с зарезервированным ключевым словом Analysis Services, возникает ошибка и таблицу необходимо переименовать. Однако ключевые слова можно использовать в именах объектов, если они заключены в квадратные скобки (для столбцов) или кавычки (для таблиц).

**NOTE**

В зависимости от приложения кавычки могут быть представлены различными символами. При вставке формул из внешнего документа или веб-страницы обязательно проверьте коды ASCII символов, которые используются для открывающих и закрывающих кавычек. Они должны совпадать. В противном случае DAX может не распознать символы как кавычки, и ссылка будет недействительной.

**Специальные символы**

Следующие символы и типы символов недопустимы в именах таблиц, столбцов и мер:

- начальные и конечные пробелы, если только имя не заключено в разделители имен, квадратные скобки или одинарные апострофы;
- управляющие символы;
- следующие символы, которые недопустимы в именах объектов:

`.,':"/?&%$!+=()[]{}<>`

**Примеры имен объектов**

В таблице ниже приведены примеры имен объектов.

ТИПЫ ОБЪЕКТОВ	ПРИМЕРЫ	КОММЕНТАРИЙ
Имя таблицы	Sales	Если имя таблицы не содержит пробелов или других специальных символов, его не обязательно заключать в кавычки.
Имя таблицы	'Продажи в Канаде'	Если имя содержит пробелы, символы табуляции или другие специальные символы, заключите его в одинарные кавычки.

ТИПЫ ОБЪЕКТОВ	ПРИМЕРЫ	КОММЕНТАРИЙ
Полное имя столбца	<b>Продажи[Объем]</b>	Имени столбца предшествует имя таблицы, и имя столбца заключено в квадратные скобки.
Полное имя меры	<b>Продажи[Доход]</b>	Имени меры предшествует имя таблицы, и имя меры заключено в квадратные скобки. В некоторых контекстах полное имя обязательно.
Неполное имя столбца	<b>[Объем]</b>	Неполное имя — это просто имя столбца в квадратных скобках. К контекстам, в которых можно использовать неполное имя, относятся формулы в вычисляемом столбце в той же таблице или в статистической функции, которая проверяет ту же таблицу.
Полное имя столбца в таблице с пробелами	<b>'Продажи в Канаде'[Количество]</b>	Имя таблицы содержит пробелы, поэтому должно быть заключено в одинарные кавычки.

### Другие ограничения

Синтаксис, необходимый для каждой функции, и тип операции, которую она может выполнять, сильно зависят от функции. Однако, как правило, ко всем формулам и выражениям применяются указанные ниже правила.

- Формулы и выражения DAX не могут изменять или вставлять отдельные значения в таблицах.
- С помощью DAX нельзя создавать вычисляемые строки. Можно создавать только вычисляемые столбцы и меры.
- При определении вычисляемых столбцов можно вкладывать функции на любом уровне.
- В DAX есть несколько функций, возвращающих таблицы. Как правило, возвращаемые ими значения используются в качестве входных данных для других функций, требующих таблицы на входе.

## Операторы и константы DAX

В приведенной ниже таблице перечислены операторы, поддерживаемые в DAX. Дополнительные сведения о синтаксисе отдельных операторов см. в статье [Операторы DAX](#).

ТИП ОПЕРАТОРА	СИМВОЛ И НАЗНАЧЕНИЕ
Оператор в виде скобок	() (порядок приоритета и группирование аргументов)

ТИП ОПЕРАТОРА	СИМВОЛ И НАЗНАЧЕНИЕ
Арифметические операторы	+ (сложение) – (вычитание или знак числа) * (умножение) / (деление) ^ (возведение в степень)
Операторы сравнения	= (равенство) > (больше чем) < (меньше чем) > = (больше или равно) < = (меньше или равно) < > (не равно)
Оператор объединения текста	& (объединение)
Логические операторы	&& (и)    (или)

## Типы данных

Тип данных столбца или значения, используемого в формуле DAX, не нужно приводить, преобразовывать или указывать иным образом. При использовании данных в формуле DAX типы данных упоминаемых столбцов и вводимых значений определяются автоматически и неявно преобразуются, если это необходимо для выполнения указанной операции.

Например, при попытке прибавить число к значению даты подсистема интерпретирует операцию в контексте функции и преобразует числа в общий тип данных, а затем представит результат в требуемом формате, то есть в виде даты.

Однако существуют некоторые ограничения на преобразование значений. Если значение или столбец имеет тип данных, несовместимый с текущей операцией, DAX возвращает ошибку. Кроме того, DAX не предоставляет функций, которые позволяли бы явно изменять, преобразовывать или приводить тип существующих данных, импортированных в модель данных.

### IMPORTANT

DAX не поддерживает использование типа данных `variant`. Таким образом, при загрузке или импорте данных в модель данных предполагается, что значения в каждом столбце имеют один тип данных.

Некоторые функции возвращают скалярные значения, в том числе строки, тогда как другие работают с числами (как целыми, так и вещественными), а также датами и значениями времени. Типы данных, необходимые для каждой функции, описаны в статье [Справочник по функциям DAX](#).

В качестве аргумента функции можно использовать таблицу, содержащую несколько столбцов и строк. Некоторые функции также возвращают таблицы, которые хранятся в памяти и могут использоваться в качестве аргументов других функций.