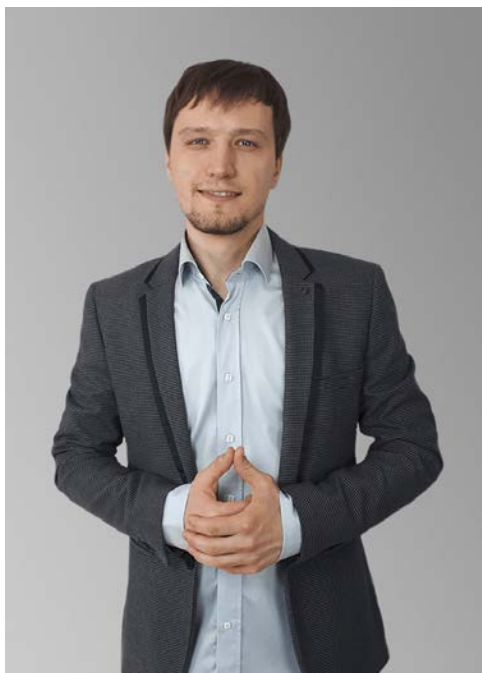


СПРАВОЧНИК  
DAX ФУНКЦИЙ  
НА РУССКОМ ЯЗЫКЕ

ДЛЯ POWER BI  
И POWER PIVOT

БУДУЕВ АНТОН  
[biprosto.ru](http://biprosto.ru)



Дорогие друзья, всем привет!

Меня зовут Будуев Антон.

Вы находитесь в справочнике DAX функций для Power BI и Power Pivot.

Давайте немного познакомимся.

Скажу сразу – я не являюсь каким-то первоисточником, создателем или разработчиком языка DAX.

Я практик, и все, что связано со сквозной BI аналитикой (Power BI, DAX и прочее) - все это я приобретаю на основе своего

личного опыта, на основе личной практики внедрения BI отчетов в своем бизнесе.

Естественно, чтобы все это внедрять у себя – я много учусь. А так как на русскоязычном пространстве очень мало каких-либо материалов по BI аналитике, то в основном, приходится обучаться за рубежом.

И именно поэтому, я и написал данный справочник. Во-первых, чтобы облегчить повседневную работу себе, а во-вторых, чтобы помочь многим из Вас.

Итак, данный справочник состоит из 3 частей:

1. Краткий перечень DAX функций, разделенных на 10 логических категорий. Для ориентира, напротив каждой функции или группы функций, имеется краткое их описание.
2. Название каждой функции кликабельно. При клике по этим ссылкам, Вы сможете перейти ко второй части данного справочника – где индивидуально для каждой функции расписано ее подробное определение, назначение, синтаксис и свойства (если это необходимо).

3. Последняя, третья часть данного справочника – это практика и разбор каждой функции на примерах конкретных формул. Но, так как практика очень большая по объему, то размещать ее в этой книге я не стал. Все практические примеры формул находятся на нашем сайте. А в этой же книге, в описаниях каждой из DAX функций, даны индивидуальные ссылки для перехода на сайт к разбору этих практических примеров.

Если у Вас имеются какие-либо вопросы, задавайте их в комментариях, в соответствующих статьях на сайте, либо же в социальных сетях. Я всегда открыт для общения!

Наши социальные сети: [Вконтакте](#), [FaceBook](#), [Инстаграм](#), [YouTube](#)

**P.S. Если:**

- в Ваших формулах имеются какие-то ошибки, проблемы, а результаты работы формул постоянно не те, что Вы ожидаете и Вам необходима помощь;
- у Вас нет понимания как правильно сочетать DAX функции друг с другом и чем они отличаются друг от друга;
- и, наконец, если Вы не до конца понимаете **суть самого языка DAX и его контекстов**, а это, замечу, в DAX самое главное!



👉 то, записывайтесь в бесплатный экспресс-курс "Быстрый старт в языке функций и формул DAX для Power BI и Power Pivot": [записаться >>>](#)

Начало уже совсем скоро!

# Краткий перечень DAX функций

## 1) Текстовые функции DAX:

- [BLANK](#) — пустое значение
- [CONCATENATE](#) и [CONCATENATEX](#) — объединение текста
- [FORMAT](#) — преобразование других типов данных в текстовый тип
- [VALUE](#) — преобразование текста в числовой тип данных
- [FIND](#) и [SEARCH](#) — поиск текста в составе другого текста
- [FIXED](#) — возвращает в текстовом формате округленное число
- [LEFT, RIGHT](#) и [MID](#) — выводят нужное количество символов
- [LEN](#) — количество символов
- [LOWER](#) и [UPPER](#) — управляют регистрами текста
- [REPLACE](#) и [SUBSTITUTE](#) — замена текста
- [REPT](#) — повтор текста
- [TRIM](#) — удаление пробелов
- [EXACT](#) — сравнение текстовых строк

## 2) Логические функции DAX:

- [IF](#) — условие если
- [TRUE](#) и [FALSE](#) — возвращают логические значения правда и ложь
- [SWITCH](#) — переключатель
- [IFERROR](#) — если ошибка
- [AND, OR](#) — и, или
- [NOT](#) — замена логических значений друг другом

### 3) Математические функции DAX:

- [DIVIDE](#), [QUOTIENT](#) и [MOD](#) — деление
- [SQRT](#) и [POWER](#) — квадратный корень и возведение в степень
- [ABS](#) — абсолютное значение числа
- [SIGN](#) — знак числа
- [EXP](#) — число E, возведенное в степень
- [FACT](#) — факториал числа
- [LN](#), [LOG](#), [LOG10](#) — логарифмы чисел
- [PI](#) — число Пи
- [INT](#), [TRUNC](#), [ROUND](#), [ROUNDDOWN](#), [ROUNDUP](#), [MROUND](#), [FLOOR](#), [CEILING](#), [ISO.CEILING](#) — округление чисел
- [RAND](#) и [RANDBETWEEN](#) — случайное число
- [RADIANS](#) — радианы

### 4) Статистические (функции агрегирования DAX):

- [SUM](#) и [SUMX](#) — операции сложения
- [MAX](#), [MAXA](#) и [MAXX](#) — вычисление максимумов
- [MIN](#), [MINA](#) и [MINX](#) — вычисление минимумов
- [AVERAGE](#), [AVERAGEA](#) и [AVERAGEX](#) — вывод среднего арифметического
- [COUNT](#), [COUNTA](#), [COUNTX](#), [COUNTAX](#), [COUNTBLANK](#), [DISTINCTCOUNT](#) и [COUNTROWS](#) — подсчет количества (значений, ячеек, строк)

### 5) Функции даты и времени в DAX:

- [TODAY](#) и [NOW](#) — сегодняшний день и сейчас (сегодняшний день + текущее время)
- [DATE](#) и [DATEVALUE](#) — возвращают даты в формате datetime и в текстовом формате

- [YEAR, MONTH и DAY](#) — возвращают год, месяц и день из значения даты
- [TIME](#) и [TIMEVALUE](#) — возвращают время в формате datetime и в текстовом формате
- [HOUR, MINUTE и SECOND](#) — возвращают час, минуты и секунды из значения времени
- [WEEKDAY](#) — номер дня недели
- [WEEKNUM](#) — номер недели в году
- [YEARFRAC](#) — доля указанного периода дат в целом году

б) Функции логики операций с датой в DAX (time intelligence):

- [CALENDARAUTO](#) и [CALENDAR](#) — создание календаря
- [DATESBETWEEN](#) и [DATESINPERIOD](#) — ограничение столбцов из дат
- [DATEADD](#) и [PARALLELPERIOD](#) — создание столбца из дат, сдвинутых вперед или назад
- [SAMEPERIODLASTYEAR](#) — создание столбца дат, сдвинутого на 1 год назад
- [PREVIOUSYEAR, PREVIOUSQUARTER, PREVIOUSMONTH и PREVIOUSDAY](#) — создание столбца с датами предыдущего года, квартала, месяца и дня
- [NEXTYEAR, NEXTQUARTER, NEXTMONTH и NEXTDAY](#) — создание столбца с датами следующего года, квартала, месяца и дня
- [DATESYTD, DATESQTD и DATESMTD](#) — создание столбца из дат от начала года, квартала или месяца до текущей даты
- [TOTALYTD, TOTALQTD и TOTALMTD](#) — вычисляет выражение от начала года, квартала или месяца до текущей даты
- [CLOSINGBALANCEYEAR, CLOSINGBALANCEQUARTER и CLOSINGBALANCEMONTH](#) — вычисляет выражение на последнюю дату года, квартала или месяца

- [OPENINGBALANCEMONTH](#), [OPENINGBALANCEQUARTER](#) и [OPENINGBALANCEYEAR](#) — вычисляет выражение на первую дату года, квартала или месяца
- [ENDOFYEAR](#), [ENDOFQUARTER](#) и [ENDOFMONTH](#) — возвращает последний день года, квартала или месяца
- [STARTOFYEAR](#), [STARTOFQUARTER](#) и [STARTOFMONTH](#) — возвращает первый день года, квартала или месяца
- [FIRSTDATE](#) и [LASTDATE](#) — возвращают первую (последнюю) известные даты
- [EOMONTH](#) и [EDATE](#) — возвращают дату с определенным отступом от заданной даты

## 7) Функции фильтров в DAX:

- [CALCULATE](#) и [CALCULATETABLE](#) — вычисляют выражение (табличное выражение) в контексте фильтров, заданных в параметрах
- [FILTER](#) — возвращает отфильтрованную таблицу по заданным фильтрам
- [ALL](#), [ALLEXCEPT](#), [ALLSELECTED](#) и [ALLNOBLANKROW](#) — удаление фильтров с таблиц или столбцов
- [VALUES](#) и [DISTINCT](#) — создание таблицы с одним столбцом из уникальных значений исходной таблицы
- [ISONORAFTER](#) — возвращает все строки, которые соответствуют условиям, заданным в параметрах
- [LOOKUPVALUE](#) — поиск по строкам и вывод значения из столбца по найденной строке
- [RELATED](#) — вывод связанных значений из другой таблицы по связи «Многие к одному»
- [RELATEDTABLE](#) — возвращает связанную таблицу значений из другой таблицы по связи «Один ко многим»

## 8) Табличные функции DAX:

- [ADDCOLUMNS](#) — добавляет столбец к существующей таблице
- [DATATABLE](#) — возвращает таблицу с нужными данными (создание таблицы «с нуля»)
- [SUMMARIZE](#), [SUMMARIZECOLUMNS](#) и [GROUPBY](#) — создание сводной таблицы с итогами по группам
- [ROW](#) — создание таблицы из одной строки с итогами исходной таблицы
- [TOPN](#) — создание таблицы из первых N строк исходной таблицы
- [EXCEPT](#), [INTERSECT](#) и [UNION](#) — создание таблицы на основе объединения двух или более исходных таблиц
- [NATURALINNERJOIN](#) и [NATURALLEFTOUTERJOIN](#) — создание таблицы на основе объединения двух исходных таблиц по внутренней связи в модели данных
- [GENERATESERIES](#) — создание таблицы последовательного ряда чисел
- [CROSSJOIN](#), [GENERATE](#) и [GENERATEALL](#) — создание таблиц из всех пересечений (декартово произведение) строк других таблиц, входящих в параметры

## 9) Информационные функции DAX:

- [ISBLANK](#) — проверка на пустое значение
- [ISNUMBER](#) и [ISEVEN](#) — проверка на числовое значение
- [ISTEXT](#) и [ISNONTEXT](#) — проверка на текстовое и нетекстовое значение
- [ISERROR](#) — проверка на ошибку
- [ISLOGICAL](#) — проверка на логическое значение
- [HASONEVALUE](#) и [HASONEFILTER](#) — проверка на одно отфильтрованное значение прямыми или косвенными фильтрами



- [ISCROSSFILTERED](#) и [ISFILTERED](#) — проверка столбца на наличие или отсутствие прямых или кросс фильтров

#### 10) Прочие функции DAX:

- [FIRSTNONBLANK](#) и [LASTNONBLANK](#) — возвращают первое (последнее) известное значение в рамках непустого результата вычисления выражения
- [EARLIER](#) — возвращает значение столбца на предыдущих уровнях контекста строки
- [VAR](#) и [RETURN](#) — создание переменных в DAX
- [RANKX](#) — возвращает ранг числа для каждой строки в столбце чисел
- [ERROR](#) — выводит сообщение с пользовательским текстом ошибки

**СКОРО...!**

[Бесплатный экспресс-курс]  
**Быстрый старт**  
в языке функций и формул DAX  
для **Power BI** и Power Pivot

Контексты в DAX

➔ **Успейте записаться**

The advertisement features a green background with a film strip effect. A yellow box at the top left contains the text 'СКОРО...!'. Below it, a dark grey box contains the text '[Бесплатный экспресс-курс] Быстрый старт в языке функций и формул DAX для Power BI и Power Pivot'. To the right, a film strip frame shows a man pointing at a screen displaying 'Контексты в DAX'. At the bottom, a red button with a white arrow icon and the text 'Успейте записаться' is centered.

## DAX функция BLANK

BLANK () — возвращает пустое значение.

Синтаксис:

BLANK ()

[Подробная информация о функции BLANK и примеры формул >>>](#)

## DAX функция CONCATENATE

CONCATENATE () — производит объединение двух текстовых строк в одну единую.

Синтаксис:

CONCATENATE ("Текст 1"; "Текст 2")

Где, «Текст» — это строка, содержащая текст или число. Также, в качестве параметров могут быть ссылки на столбцы с текстовым типом данных.

[Подробная информация о функции CONCATENATE и примеры формул >>>](#)

## DAX функция CONCATENATEX

CONCATENATEX () — объединяет результат выражения, вычисленного для каждой строки таблицы.

Синтаксис:

CONCATENATEX ('Таблица'; Выражение; "Разделитель")

Где:

- 'Таблица' — таблица, содержащая строки, для которых будет вычислено выражение
- Выражение — выражение, вычисляемое для каждой строки таблицы
- «Разделитель» — разделитель объединяемых частей текста (необязательный элемент)

[Подробная информация о функции CONCATENATEX и примеры формул >>>](#)

## DAX функция FORMAT

FORMAT () — меняет различные типы данных (числа, дата, время) на текстовый тип данных с возможностью задать нужный формат вывода информации.

Синтаксис:

FORMAT (Значение; Формат)

Где:

- Значение — значение с типом данных число, дата, время
- Формат — шаблон формата

Шаблоны формата чисел:

- «General Number» — выводит число
- «Currency» — денежный формат. Выводит число, разделяя его на группы разрядов на основании параметров локали ПК
- «Fixed» — округляет число до 2 знаков после запятой
- «Standard» — выводит число, разделенное на группы разрядов, округляя его до 2 знаков после запятой
- «Percent» — умножает число на 100, преобразуя его в проценты (%). После запятой число округляется до 2 знаков
- «Scientific» — преобразует число в научный вид
- «Yes/No» — выводит «Да», если число не равно 0 и «Нет», если равно 0
- «True/False» — выводит «TRUE» (Истина), если число не равно 0 и «FALSE» (Ложь), если равно 0
- «On/Off» — выводит «Вкл», если число не равно 0 и «Выкл», если равно 0

Шаблоны формата дат:

- «D» — выводит номер дня месяца (без нулей)
- «DD» — выводит номер дня месяца (с нулём)
- «DDD» — преобразует исходную дату в день недели (сокращенное название — Пт, Сб, Вс)

- «DDDD» — преобразует исходную дату в день недели (полное название)
- «M» — выводит номер месяца в году (без нулей)
- «MM» — выводит номер месяца в году (с нулём)
- «MMM» — преобразует исходную дату в месяц (сокращенное название — дек)
- «MMMM» — преобразует исходную дату в месяц (полное название)
- «уу» — выводит год в формате 2 цифр
- «уууу» — выводит год в формате 4 цифр

Шаблоны формата времени:

- «H» — преобразует исходное время в часы (без нулей)
- «HH» — преобразует исходное время в часы (с нулём)
- «m» — преобразует исходное время в минуты (без нулей)
- «mm» — преобразует исходное время в минуты (с нулём)
- «s» — преобразует исходное время в секунды (без нулей)
- «ss» — преобразует исходное время в секунды (с нулём)

[Подробная информация о функции FORMAT и примеры формул >>>](#)

## DAX функция VALUE

VALUE () — преобразует число, записанное в текстовом формате в настоящее число (в числовом типе данных). В некотором смысле, можно сказать, что VALUE — обратная функция для функции FORMAT.

Синтаксис:

VALUE ("Число")

Где, «Число» — это значение числа (или столбец со значениями чисел), записанное текстом (в текстовом типе данных)

[Подробная информация о функции VALUE и примеры формул >>>](#)

## DAX функция FIND

FIND () — ищет один текст в составе другого текста с учетом регистра букв и выводит номер позиции первого символа найденного текста в составе символов другого текста.

Синтаксис:

FIND ("Текст 1"; "Текст 2"; Номер Позиции; Значение Нет)

Где:

- Текст 1 — символы текста, которые нужно найти. Используются подстановочные шаблоны: ? — один любой символ, \* — много любых символов. Если в тексте нужно найти знак вопроса (?) или знак звездочка (\*), то перед ними нужно поставить значок тильды ~
- Текст 2 — текст, в котором идет поиск
- Номер Позиции — номер позиции того символа, с которого нужно начать поиск

- Значение Нет — (необязательный параметр) числовое значение или значение BLANK (), которое нужно вывести, если текст не найден

[Подробная информация о функции FIND и примеры формул >>>](#)

## DAX функция SEARCH

SEARCH () — ищет один текст в составе другого текста без учета регистра букв и выводит номер позиции первого символа найденного текста в составе символов другого текста.

Синтаксис:

SEARCH ("Текст 1"; "Текст 2"; Номер Позиции; Значение Нет)

Где:

- Текст 1 — символы текста, которые нужно найти. Используются подстановочные шаблоны: ? — один любой символ, \* — много любых символов. Если в тексте нужно найти знак вопроса (?) или знак звездочка (\*), то перед ними нужно поставить значок тильды ~
- Текст 2 — текст, в котором идет поиск
- Номер Позиции — номер позиции того символа, с которого нужно начать поиск
- Значение Нет — (необязательный параметр) числовое значение или значение BLANK (), которое нужно вывести, если текст не найден

[Подробная информация о функции SEARCH и примеры формул >>>](#)

## DAX функция FIXED

FIXED () — возвращает в текстовом формате округленное число (до указанного значения десятичного числа)

Синтаксис:

FIXED (Число; Округление; Разряд)

Где:

- Число – исходное десятичное число (столбец с числовыми значениями)
- Округление – (необязательный параметр) до скольких знаков после запятой округлить десятичное число. Если число не указано, то по умолчанию 2
- Разряд – (необязательный параметр) если указано 0, то создать разряд чисел из пробелов, если указано 1, то не создавать разряд. По умолчанию — 0

[Подробная информация о функции FIXED и примеры формул >>>](#)

## DAX функции LEFT и RIGHT



LEFT () — выводит нужное количество символов с левой стороны.

RIGHT () — выводит нужное количество символов с правой стороны.

Синтаксис:

LEFT ("Текст"; Количество Символов)

RIGHT ("Текст"; Количество Символов)

Где:

- Текст – исследуемый текст или столбец с текстовыми данными
- Количество Символов – количество символов, которые нужно вывести

[Подробная информация о функциях LEFT, RIGHT и примеры формул >>>](#)

## DAX функция MID

MID () — выводит часть текста заданной длины с учетом первоначальной позиции.

Синтаксис:

MID ("Текст"; Позиция; Количество Символов)

Где:

- Текст – исследуемый текст или столбец с текстовыми данными
- Позиция – номер позиции символа, с которого нужно вывести текст

- Количество Символов – количество символов, которые нужно вывести

[Подробная информация о функции MID и примеры формул >>>](#)

## DAX функция LEN

LEN () — возвращает количество символов в текстовой строке.

Синтаксис:

LEN ("Текст")

Где, «Текст» — любое текстовое значение, выражение или столбец с текстовыми данными. Пробелы также включаются в подсчет символов.

[Подробная информация о функции LEN и примеры формул >>>](#)

## DAX функции LOWER и UPPER

LOWER () – возвращает исходный текст полностью в нижнем регистре.

UPPER () – возвращает исходный текст полностью в верхнем регистре.

Синтаксис:

LOWER ("Текст")

UPPER ("Текст")

Где, «Текст» — текст или столбец, содержащий текстовые значения.

[Подробная информация о функциях LOWER, UPPER и примеры формул >>>](#)

## DAХ функция REPLACE

REPLACE () — заменяет один текст другим, с учетом стартовой позиции и количества символов.

Синтаксис:

REPLACE ("Текст 1"; Позиция; Количество; "Текст 2")

Где:

- Текст 1 – старый текст, который нужно изменить
- Текст 2 – новый текст, которым нужно заменить старый
- Позиция – номер позиции символа, с которого нужно заменять старый текст
- Количество – количество символов, которые нужно заменить

[Подробная информация о функции REPLACE и примеры формул >>>](#)

## DAX функция SUBSTITUTE

SUBSTITUTE () — заменяет один текст другим с учетом регистра.

Синтаксис:

SUBSTITUTE ("Текст 1"; "Текст 2"; "Текст 3"; Номер Вхождения)

Где:

- Текст 1 – исходный текст (столбец с текстовыми данными)
- Текст 2 – часть текста, которую нужно изменить
- Текст 3 – новая часть текста
- Номер вхождения — (необязательный параметр) вхождение (повторы) того слова, которое нужно заменить. Если параметр не указан, то по умолчанию заменяются все вхождения (повторы) искомого слова

[Подробная информация о функции SUBSTITUTE и примеры формул >>>](#)

## DAX функция REPT

REPT () — функция повтора текста с заданным количеством повторов.

Синтаксис:

REPT ("Текст"; Количество Повторов)

Где:

- Текст – исходный текст для повтора
- Количество Повторов – число повторов исходного текста

[Подробная информация о функции REPT и примеры формул >>>](#)

## DAX функция TRIM

TRIM () — удаляет все лишние пробелы из текста, кроме одинарных пробелов между словами

Синтаксис:

TRIM ("Текст")

Где, «Текст» — любое текстовое значение, выражение или столбец с текстовыми данными.

[Подробная информация о функции TRIM и примеры формул >>>](#)

## DAX функция EXACT

EXACT () — сравнение двух текстовых строк. Если строки одинаковые, то функция возвращает TRUE, если разные, то FALSE. EXACT учитывает регистр букв, но не учитывает их формат.

Синтаксис:

EXACT ("Текст 1"; "Текст 2")

Где, «Текст 1» и «Текст 2» — сравниваемые тексты или столбцы, содержащие текстовые значения.

[Подробная информация о функции EXACT и примеры формул >>>](#)

## DAХ функция IF

IF () — если. Выполняет проверку условия. Если условие равно логическому значению TRUE, то функция выполняет одно выражение. Если же условие равно логическому значению FALSE, то функция выполняет второе выражение.

Синтаксис:

IF (Условие; Выражение 1; Выражение 2)

Где:

- Условие — логическое условие, результат которого равен либо значению TRUE, либо — FALSE
- Выражение 1 — выражение, которое вступит в работу, если результат проверки условия будет равен TRUE
- Выражение 2 — выражение, которое вступит в работу, если результат проверки условия будет равен FALSE

[Подробная информация о функции IF и примеры формул >>>](#)

## DAX функции TRUE и FALSE

TRUE () — возвращает логическое значение TRUE (истина, да, правда, правильно).

FALSE () — возвращает логическое значение FALSE (ложь, нет, неправда, неправильно).

Синтаксис:

TRUE ()

FALSE ()

[Подробная информация о функциях TRUE \(\), FALSE \(\) и примеры формул >>>](#)

## DAX функция SWITCH

SWITCH () — переключатель. Сравнивает результат вычисления выражения с заранее подготовленными значениями и возвращает итоговый результат, относительно найденного подготовленного значения.

Синтаксис:

SWITCH (

    Выражение;

    Значение 1; Результат 1;

Значение 2; Результат 2;

Значение N; Результат N;

Иначе

)

Где:

- Выражение — вычисляемое выражение, результат которого сравнивается с заранее подготовленными значениями
- Значение — заранее подготовленное значение
- Результат — заранее подготовленный результат, который возвратится, если результат вычисления выражения будет равен соответствующему заранее подготовленному значению
- Иначе — заранее подготовленный результат, который возвратится, если результат вычисления выражения не будет равен ни одному заранее подготовленному значению

[Подробная информация о функции SWITCH и примеры формул >>>](#)

## DAX функция IFERROR

IFERROR () — если ошибка. Производит вычисление выражения и если во время вычисления возникла ошибка, то функция выводит значение из второго параметра, если ошибок нет, то возвращается результат вычисления самого выражения.



Синтаксис:

IFERROR (Выражение; Значение Если Ошибка)

[Подробная информация о функции IFERROR и примеры формул >>>](#)

## DAX функции AND и OR

AND () — и. Возвращает логическое значение TRUE, если оба параметра, входящие в эту функцию, в режиме «и», также, возвращают TRUE. Если же хотя бы один параметр возвращает FALSE, то сама функция, также, возвращает логическое значение FALSE. Аналогом в языке DAX является логический оператор && (и).

OR () — или. Возвращает логическое значение TRUE, если хотя бы один из двух параметров, входящих в эту функцию, также, возвращает логическое значение TRUE. Если же оба параметра возвращают FALSE, то сама функция, также, возвращает логическое значение FALSE. Аналогом в языке DAX является логический оператор || (или).

Синтаксис:

AND (Значение 1; Значение 2)

OR (Значение 1; Значение 2)

Данные функции обычно используют тогда, когда нужно в каких-либо условиях или фильтрах использовать в одном условии или параметре сразу 2 или более условий. Если нужно больше 2 условий, то тогда эти функции вкладываются друг в друга, например, так:

AND (Значение 1; AND (Значение 2; Значение 3))

OR (Значение 1; OR (Значение 2; Значение 3))

Но, в этих случаях предпочтительнее использовать уже не данные функции AND или OR, а соответствующие операторы DAX, например, так:

Сочетание в режиме "и": Значение 1 && Значение 2 && Значение 3

Сочетание в режиме "или": Значение 1 || Значение 2 || Значение 3

[Подробная информация о функциях AND, OR и примеры формул >>>](#)

## DAX функция NOT

NOT () — заменяет логические значения друг другом. Если переданное в функцию выражение равно логическому значению FALSE, то функция возвращает TRUE и наоборот, если переданное в функцию выражение равно логическому значению TRUE, то функция возвращает FALSE. Иначе говоря, меняет FALSE на TRUE и наоборот.

Синтаксис:

NOT (Выражение)

[Подробная информация о функции NOT и примеры формул >>>](#)

## DAX функция DIVIDE

DIVIDE () — производит деление с обработкой ошибки «деление на 0». Обработка ошибки заключается в выводе альтернативного результата в случае возникновения ситуации деления на ноль.

Синтаксис:

DIVIDE (Делимое Число; Делитель; Альтернатива)

Где, альтернатива — (необязательный параметр) значение, которое нужно вывести в случае ошибки деления на ноль (0). По умолчанию выводится пустое значение BLANK ().

[Подробная информация о функции DIVIDE и примеры формул >>>](#)

## DAX функция QUOTIENT

QUOTIENT () — выполняет деление чисел, входящих в параметры функции и возвращает целочисленную часть от деления.

Синтаксис:

QUOTIENT (Делимое Число; Делитель)

[Подробная информация о функции QUOTIENT и примеры формул >>>](#)

## DAX функция MOD

MOD () — возвращает остаток от деления со знаком делителя.

Синтаксис:

MOD (Делимое Число; Делитель)

[Подробная информация о функции MOD и примеры формул >>>](#)

## DAX функция SQRT

SQRT () — функция квадратного корня из числа.

Синтаксис:

SQRT (Число)

Где, число — числовое значение или столбец, содержащий числовые значения.

[Подробная информация о функции SQRT и примеры формул >>>](#)

## DAX функция POWER

POWER () — возводит число в степень.

Синтаксис:

POWER (Число; Степень)

Где:

- число — числовое значение или столбец, содержащий числовые значения
- степень — число необходимой степени

[Подробная информация о функции POWER и примеры формул >>>](#)

## DAX функция ABS

ABS () — функция абсолютного значения числа. Иначе говоря, преобразует отрицательное число в положительное.

Синтаксис:

ABS (Число)

Где, число — числовое значение или столбец, содержащий числовые значения.

[Подробная информация о функции ABS и примеры формул >>>](#)

## DAX функция SIGN

SIGN () — возвращает знак числа в столбце или числа, получившегося в результате вычисления выражения. Значение 1 соответствует положительному числу, значение 0 соответствует нулю и -1 соответствует отрицательному числу.

Синтаксис:

SIGN (Число)

Где, число — числовое значение или столбец, содержащий числовые значения.

[Подробная информация о функции SIGN и примеры формул >>>](#)

## DAX функция EXP

EXP () — возводит число E (2,71828182845904) в нужную степень.

Синтаксис:

EXP (Степень)

Где, степень — числовое значение или столбец, содержащий числовые значения нужной степени.

[Подробная информация о функции EXP и примеры формул >>>](#)

## DAX функция FACT

FACT () — факториал числа (произведение последовательности целых чисел начиная с 1 и до указанного числа).

Синтаксис:

FACT (Число)

Где, число — числовое значение или столбец, содержащий числовые значения, указывающее для какого числа производить факториал.

[Подробная информация о функции FACT и примеры формул >>>](#)

## DAX функции LN, LOG, LOG10

LN () — вычисляет натуральный логарифм числа по константе E (2,71828182845904).

LOG () — вычисляет логарифм числа по заданному в параметре функции основанию.

LOG10 () — вычисляет логарифм числа по основанию 10.

Синтаксис:

LN (Число)

LOG (Число; Основание)

LOG10 (Число)

[Подробная информация о функциях LN, LOG, LOG10 и примеры формул >>>](#)

## DAX функция PI

PI () — возвращает значение Пи (3,14159265358979)

Синтаксис:

PI ()

## DAX функция INT

INT () — производит округление числа до ближайшего целого в сторону уменьшения.

Синтаксис:

INT (число)

Где, число — числовое значение или столбец с числовыми значениями.

[Подробная информация о функции INT и примеры формул >>>](#)



## DAX функция TRUNC

TRUNC () — округляет число до целого, обрезая его дробную часть

Синтаксис:

TRUNC (Число; ТО)

Где:

- Число — числовое значение или столбец с числовыми значениями
- ТО — (необязательный параметр) число, задающее точность округления. По умолчанию = 0. Если ТО положительное, то оно отсекает дробную часть, если отрицательное, то округляет правую часть целого числа относительно запятой, если равно 0, то возвращает само целое число, без дроби

[Подробная информация о функции TRUNC и примеры формул >>>](#)

## DAX функция ROUND

ROUND () — производит математическое округление числа до указанной точности округления.

ROUND очень похожа на функцию TRUNC, только TRUNC — просто обрезает дробную часть, а ROUND именно округляет математически.

Синтаксис:

## ROUND (Число; ТО)

Где:

- Число — числовое значение или столбец с числовыми значениями
- ТО — число, задающее точность округления. Если ТО положительное, то оно математически округляет дробную часть, если отрицательное, то округляет правую часть целого числа относительно запятой. Если равно 0, то возвращает математически округленное целое число, без дроби

[Подробная информация о функции ROUND и примеры формул >>>](#)

## DAX функция ROUNDDOWN

ROUNDDOWN () — производит округление числа вниз до указанной точности округления.

ROUNDDOWN похожа на функцию ROUND, но ROUND — производит математическое округление (и вверх, и вниз), а ROUNDDOWN, исходя из названия — округляет всегда вниз.

Также, ROUNDDOWN похожа на функцию INT, так как INT, также, округляет число вниз, но всегда только до целого числа, в то время как, ROUNDDOWN может округлять не только до целого числа, но и до дробного, управляя точностью округления.

Синтаксис:

## ROUNDDOWN (Число; ТО)

Где:

- Число — числовое значение или столбец с числовыми значениями
- ТО — число, задающее точность округления. Если ТО положительное, то оно округляет дробную часть, если отрицательное, то округляет правую часть целого числа относительно запятой, если равно 0, то возвращает округленное целое число, без дроби

[Подробная информация о функции ROUNDDOWN и примеры формул >>>](#)

## DAX функция ROUNDUP

ROUNDUP () — производит округление числа вверх до указанной точности округления.

ROUNDUP похожа на функцию ROUNDDOWN, но ROUNDDOWN — производит округление только вниз, а ROUNDUP, исходя из названия — округляет всегда вверх.

Синтаксис:

ROUNDUP (Число; ТО)

Где:

- Число — числовое значение или столбец с числовыми значениями

- TO — число, задающее точность округления. Если TO положительное, то оно округляет дробную часть, если отрицательное, то округляет правую часть целого числа относительно запятой, если равно 0, то возвращает округленное целое число, без дроби

[Подробная информация о функции ROUNDUP и примеры формул >>>](#)

## DAX функция MROUND

MROUND () — выполняет округление числа, кратно заданному значению точности.

Синтаксис:

MROUND (Число; Кратное)

Где:

- Число — числовое значение или столбец с числовыми значениями
- Кратное — число, задающее значение, кратно которому будет округляться исходное число

! — Исходное число и кратное число, входящие в параметры функции MROUND должны быть одного знака — или оба положительные, или оба отрицательные, иначе возвратится ошибка.

[Подробная информация о функции MROUND и примеры формул >>>](#)

## DAX функция FLOOR

FLOOR () — выполняет округление числа в меньшую к нулю сторону, кратно заданному значению точности.

Синтаксис:

FLOOR (Число; Кратное)

Где:

- Число — числовое значение или столбец с числовыми значениями
- Кратное — число, задающее значение, кратно которому будет округляться исходное число

! — Если исходное число положительное, то кратное число, входящее в параметры функции FLOOR, также, должно быть положительным. Если исходное число отрицательное, то кратное число может быть как отрицательным, так и положительным.

! — Кратное число не может равняться 0

[Подробная информация о функции FLOOR и примеры формул >>>](#)

## DAX функция CEILING

CEILING () — выполняет округление числа в большую сторону до ближайшего целого, кратно значению из второго параметра.

Синтаксис:

CEILING (Число; Кратное)

Где:

- Число — числовое значение или столбец с числовыми значениями
- Кратное — число, задающее значение, кратно которому будет округляться исходное число

! — Если исходное число положительное, то кратное число, входящее в параметры функции CEILING, также, должно быть положительным. Если исходное число отрицательное, то кратное число может быть как отрицательным, так и положительным.

! — Если исходное число отрицательное и кратное число, также, отрицательное, то округление происходит в меньшую сторону.

[Подробная информация о функции CEILING и примеры формул >>>](#)

## DAX функция ISO.CEILING

ISO.CEILING () — выполняет округление числа в большую сторону до ближайшего целого, кратно значению из второго параметра.

Синтаксис:

ISO.CEILING (Число; Кратное)

Где:

- Число — числовое значение или столбец с числовыми значениями
- Кратное — число, задающее значение, кратно которому будет округляться исходное число

ISO.CEILING — по всем параметрам идентична функции CEILING, за исключением того фактора, когда исходное число отрицательное и кратное число, также, отрицательное — ISO.CEILING, в отличие от CEILING, округление производит в большую сторону.

[Подробная информация о функции ISO.CEILING и примеры формул >>>](#)

## DAX функции RAND И RANDBETWEEN

RAND () — возвращает случайное число от 0 до 1 или равное 0.

RANDBETWEEN () — возвращает случайное число между двумя числами, прописанными в параметрах функции.

Синтаксис:

RAND ()

RANDBETWEEN (Число от; Число до)

Где, числа «от» и «до» — это числа, между которыми возвратится случайное число в Power BI.

[Подробная информация о функциях RAND и RANDBETWEEN и примеры формул >>>](#)

## DAX функция RADIANS

RADIANS () — преобразует градусы в радианы.

Синтаксис:

RADIANS (Значение угла)

Где, значение угла — угол в градусах, который необходимо преобразовать в радианы.

[Подробная информация о функции RADIANS и примеры формул >>>](#)

## DAX функция SUM

SUM () — производит сумму всех чисел в столбце. Если строка в столбце содержит значение с иным типом данных, нежели числовой, то для работы SUM возвращается пустое значение.

Синтаксис:

SUM ([Столбец])

[Подробная информация о функции SUM и примеры формул >>>](#)



## DAX функция SUMX

SUMX () — вычисляет сумму результатов построчного выполнения выражения.

Синтаксис:

SUMX ('Таблица'; Выражение),

Где:

- 'Таблица' — исходная таблица или табличное выражение, по строкам которой будет вычисляться выражение из второго параметра функции
- Выражение — любое выражение, которое необходимо выполнить по строкам таблицы, входящей в первый параметр функции

SUMX работает в 2 действия. Для начала, вычисляется выражение из второго параметра функции для каждой строки таблицы, входящей в первый параметр. По мере построчного вычисления, результаты временно запоминаются в некой виртуальной таблице, а затем SUMX вторым действием складывает все результаты из этой виртуальной таблицы и выдает итоговую сумму. После этого, временная виртуальная таблица стирается из памяти.

[Подробная информация о функции SUMX и примеры формул >>>](#)

## DAX функция MAX

MAX () — возвращает максимум среди числовых значений. MAX работает в двух режимах и эту особенность необходимо всегда учитывать в ее работе. Режимы работы зависят от вида синтаксиса.

Синтаксис:

MAX ([Столбец]) — возвращает максимум среди значений столбца (особенность — не учитывает контекст строки)

MAX (Выражение 1; Выражение 2) — возвращает максимум среди значений из двух параметров функции. Если работа производится в таблице, то MAX работает построчно, вычисляя максимум в рамках только одной строки (особенность — учитывает контекст строки)

[Подробная информация о функции MAX и примеры формул >>>](#)

## DAX функция MAXA

MAXA () — вычисляет максимум с обработкой пустых ячеек и ячеек, содержащих логический тип данных. Причем TRUE считается как 1, а FALSE — как 0.

Синтаксис:

MAXA ([Столбец])

[Подробная информация о функции MAXA и примеры формул >>>](#)

## DAX функция MAXX

MAXX () — выводит максимальное значение из результатов построчного вычисления выражения.

Синтаксис:

MAXX ('Таблица'; Выражение)

Где:

- 'Таблица' — исходная таблица или табличное выражение, по строкам которой будет вычисляться выражение из второго параметра функции
- Выражение — любое выражение, которое необходимо выполнить по строкам таблицы, входящей в первый параметр функции

Первым действием MAXX вычисляет выражение из второго параметра по каждой строке таблицы из первого параметра, а затем вычисляет максимальное значение из получившихся результатов первого действия.

[Подробная информация о функции MAXX и примеры формул >>>](#)

## DAX функция MIN

MIN () — вычисляет минимум среди числовых значений. Также, как и функция MAX, MIN работает в двух режимах, зависящих от вида синтаксиса.

Синтаксис:

MIN ([Столбец]) — возвращает минимум среди значений столбца (не учитывает контекст строки)

MIN (Выражение 1; Выражение 2) — возвращает минимум среди значений из двух параметров функции. Если работа производится в таблице, то MIN работает построчно, вычисляя минимум в рамках только одной строки (учитывает контекст строки)

[Подробная информация о функции MIN и примеры формул >>>](#)

## DAX функция MINA

MINA () — возвращает минимум с обработкой пустых ячеек и ячеек, содержащих логический тип данных. TRUE считается как 1, а FALSE — как 0.

Синтаксис:

MINA ([Столбец])

[Подробная информация о функции MINA и примеры формул >>>](#)

## DAX функция MINX

MINX () — выводит минимальное значение из результатов построчного вычисления выражения.

Синтаксис:

MINX (Таблица'; Выражение)

Вычисления производит в два действия — сначала вычисляет выражение из второго параметра по каждой строке таблицы из первого параметра, а потом возвращает минимум по получившимся результатам первого действия.

[Подробная информация о функции MINX и примеры формул >>>](#)

## DAX функция AVERAGE

AVERAGE () — высчитывает среднее арифметическое значение числовых данных столбца (сумма значений деленное на количество этих значений):

- если в столбце текстовый тип данных, то AVERAGE возвращает пустое значение
- если в столбце пустые ячейки или логический тип данных, то в расчет среднего они не берутся
- если в столбце находятся ячейки с 0, то такие ячейки учитываются, как обычные

Синтаксис:

AVERAGE ([Столбец])

[Подробная информация о функции AVERAGE и примеры формул >>>](#)

## DAX функция AVERAGEA

AVERAGEA () — высчитывает среднее арифметическое значение числовых данных столбца с учетом текстовых и логических типов данных:

- логические выражения равные TRUE (Правда) обрабатываются как 1
- логические выражения равные FALSE (Ложь) обрабатываются как 0
- текстовый тип данных обрабатывается как 0

Синтаксис:

AVERAGEA ([Столбец])

[Подробная информация о функции AVERAGEA и примеры формул >>>](#)

## DAX функция AVERAGEX

AVERAGEX () — вычисляет среднее арифметическое среди результатов построчного выполнения выражения.

Синтаксис:

AVERAGEX ('Таблица'; Выражение)

Где:

- 'Таблица' — исходная таблица или табличное выражение, по строкам которой будет вычисляться выражение из второго параметра функции
- Выражение — любое выражение, которое необходимо выполнить по строкам таблицы, входящей в первый параметр функции

Функция AVERAGEX работает в 2 этапа. На первом этапе вычисляется выражение из второго параметра для каждой строки таблицы, указанной в первом параметре. На втором этапе AVERAGEX высчитывает среднее значение по данным, получившимся на первом этапе работы.

[Подробная информация о функции AVERAGEX и примеры формул >>>](#)

## DAX функция COUNT

COUNT () — подсчитывает в столбце количество ячеек, которые содержат в себе числовое значение. В качестве числового значения признаются числа, даты и число, записанное в текстовом типе данных.

Если в строке учитываемых значений нет, то функция выдаст 0. Если в таблице отсутствуют строки, то COUNT выдаст пустое значение.

Синтаксис:

COUNT ([Столбец])

[Подробная информация о функции COUNT и примеры формул >>>](#)

## DAX функция COUNTA

COUNTA () — подсчитывает непустые ячейки в столбце. То есть, количество тех ячеек, которые в себе содержат хоть какое-то значение: числа, даты, любой текст или значения логического типа.

Синтаксис:

COUNTA ([Столбец])

[Подробная информация о функции COUNTA и примеры формул >>>](#)

## DAX функция COUNTX

COUNTX () — подсчитывает количество строк, содержащие в себе числовое значение, получившееся в результате построчного



выполнения выражения. В качестве числового значения признаются числа, даты и число, записанное в текстовом типе данных.

Синтаксис:

COUNTX ('Таблица'; Выражение)

Где:

- 'Таблица' — исходная таблица или табличное выражение, по строкам которой будет вычисляться выражение из второго параметра функции
- Выражение — любое выражение, которое необходимо выполнить по строкам таблицы, входящей в первый параметр функции

После вычисления выражения по каждой строке таблицы, на основе получившихся построчных результатов, COUNTX подсчитывает количество тех самых числовых значений.

[Подробная информация о функции COUNTX и примеры формул >>>](#)

## DAX функция COUNTAX

COUNTAX () — работает точно также, как и функция выше COUNTX, но в данном случае, уже на основе получившихся построчных результатов, COUNTAX подсчитывает непустые значения (числа, даты, текст, значения логического типа)

Синтаксис:

COUNTAX ('Таблица'; Выражение)

[Подробная информация о функции COUNTAX и примеры формул >>>](#)

## DAX функция COUNTBLANK

COUNTBLANK () — подсчитывает количество пустых ячеек в столбце.

Синтаксис:

COUNTBLANK ([Столбец])

[Подробная информация о функции COUNTBLANK и примеры формул >>>](#)

## DAX функция DISTINCTCOUNT

DISTINCTCOUNT () — подсчитывает количество уникальных значений ячеек в столбце

Синтаксис:

DISTINCTCOUNT ([Столбец])

[Подробная информация о функции DISTINCTCOUNT и примеры формул >>>](#)

## DAX функция COUNTROWS

COUNTROWS () — часто используемая на практике функция, которая подсчитывает количество строк в таблице.

Синтаксис:

COUNTROWS ('Таблица')

[Подробная информация о функции COUNTROWS и примеры формул >>>](#)

## DAX функция TODAY

TODAY () — сегодня. Возвращает текущую дату. Данная функция очень простая и не содержит в себе ни одного параметра.

Синтаксис:

TODAY ()

[Подробная информация о функции TODAY и примеры формул >>>](#)

## DAX функция NOW

NOW () — сейчас. Возвращает текущую дату и время. Это простейшая функция, не содержащая параметров.

Синтаксис:

NOW ()

[Подробная информация о функции NOW и примеры формул >>>](#)

## DAX функция DATE

DATE () — возвращает в формате datetime прописанную в параметрах дату.

Синтаксис:

DATE (год; месяц; день)

! — Если месяц не соответствует числам 1-12, то лишние (недостающие) месяца добавляются (убавляются) к году.

! — Если дни не соответствуют числам 1-31 в месяце, то лишние (недостающие) дни добавляются (убавляются) к месяцу и к году.

[Подробная информация о функции DATE и примеры формул >>>](#)

## DAX функция DATEVALUE

DATEVALUE () — используя локаль даты ПК, возвращает в формате datetime прописанную в параметрах текстовую дату.

Синтаксис:

DATEVALUE ("Дата")

«Дата» — может быть записана в различных вариантах, в том числе, и сокращенных.

[Подробная информация о функции DATEVALUE и примеры формул >>>](#)

## DAX функции YEAR, MONTH, DAY

YEAR (), MONTH () и DAY () — возвращают год, месяц и день в формате чисел из значения даты.

Синтаксис:

YEAR ([Дата])

MONTH ([Дата])

DAY ([Дата])

Где [Дата] — столбец, содержащий даты, либо дата в текстовом или datetime форматах.

[Подробная информация о функциях YEAR, MONTH, DAY и примеры формул >>>](#)

## DAХ функция TIME

TIME () — возвращает в формате datetime прописанное в параметрах время.

Синтаксис:

TIME (часы; минуты; секунды)

! — Если часы указаны числом, выходящим за пределы 23, то в качестве часа будет указан остаток от деления этого числа на 24

! — Если минуты указаны числом, выходящим за пределы 59, то это значение будет пересчитано в часы и минуты

! — Если секунды указаны числом, выходящим за пределы 59, то это значение будет пересчитано в часы, минуты и секунды

[Подробная информация о функции TIME и примеры формул >>>](#)

## DAХ функция TIMEVALUE

TIMEVALUE () — используя локаль времени ПК, возвращает в формате datetime прописанное в параметрах текстовое время.

Синтаксис:

TIMEVALUE ("Время")

[Подробная информация о функции TIMEVALUE и примеры формул >>>](#)

## DAX функции HOUR, MINUTE И SECOND

HOUR (), MINUTE () и SECOND () — возвращают часы, минуты и секунды в формате чисел из значения времени.

Синтаксис:

HOUR ([Время])

MINUTE ([Время])

SECOND ([Время])

Где:

[Время] — столбец, содержащий время, либо время в текстовом или datetime форматах.

[Подробная информация о функциях HOUR, MINUTE, SECOND и примеры формул >>>](#)

## DAX функция WEEKDAY

WEEKDAY () — возвращает день недели в формате чисел 1-7 (0-6). По умолчанию неделя начинается с воскресенья (1) и заканчивается субботой (7).

Синтаксис:

WEEKDAY (Дата; Начало Недели)

Где:

- Дата — дата в формате datetime. Дату необходимо вводить формулой DATE() или другим выражением, возвращающим формат datetime
- Начало Недели (по умолчанию — 1):
  - 1 — начало недели в воскресенье (1) и конец в субботу (7);
  - 2 — начало недели в понедельник (1) и конец в воскресенье (7);
  - 3 — начало недели в понедельник (0) и конец в воскресенье (6)

[Подробная информация о функции WEEKDAY и примеры формул >>>](#)

## DAX функция WEEKNUM

WEEKNUM () — возвращает номер недели года (относительно начала года).



Синтаксис:

WEEKNUM (Дата; Начало Недели)

Где:

- Дата — дата в формате datetime. Дату необходимо вводить формулой DATE() или другим выражением, возвращающим формат datetime
- Начало Недели:
  - 1 — неделя начинается с воскресенья (по умолчанию);
  - 2 — неделя начинается с понедельника

[Подробная информация о функции WEEKNUM и примеры формул >>>](#)

## DAX функция YEARFRAC

YEARFRAC () — вычисляет долю указанного периода дат в целом году.

Синтаксис:

YEARFRAC (Стартовая Дата; Конечная Дата; Базис)

Где:

- «Стартовая Дата» и «Конечная Дата» — даты в формате datetime
- «Базис» — способ вычисления дня (необязательный параметр):
  - параметр = 0 — американский стандарт (NASD), 30 дней / 60 дней
  - параметр = 1 — фактический период / фактический период
  - параметр = 2 — фактический период / 360 дней

- параметр = 3 — фактический период / 365 дней
- параметр = 4 — европейский стандарт, 30 дней / 360 дней

[Подробная информация о функции YEARFRAC и примеры формул >>>](#)

## DAX функция CALENDARAUTO

CALENDARAUTO () — автоматический календарь в DAX. Возвращает таблицу с одним столбцом [Date], содержащим непрерывные даты от начала и до конца года. Рассчитывается исходя из тех дат, которые находятся во всех таблицах модели данных.

То есть, CALENDARAUTO в модели данных ищет самую старую (минимальную) и самую новую (максимальную) даты. И на основе года из этих дат, строит полный календарь от 1 января того года, который указан в самой старой (минимальной) дате и до 31 декабря того года, который указан в самой новой (максимальной) дате из всех таблиц модели данных.

! — Если в модели данных нет значений даты, то функция CALENDARAUTO возвратит ошибку.

! — Будьте внимательны, CALENDARAUTO для расчетов берет любые даты, например, даты дней рождения. Поэтому, если в Вашей модели данных будет человек с днем рождения в 1925 году, то функция создаст календарь, начиная с 1925 года и далее до самой последней даты в модели данных.

Синтаксис:

CALENDARAUTO ()

CALENDARAUTO (Номер Месяца)

Где, Номер Месяца — это номер месяца в году.

Если в CALENDARAUTO параметров нет, то возвращается календарь, кратный целому году с 1 января по 31 декабря.

Если в параметрах прописан номер месяца, то возвращается календарь, кратный целому году (12 месяцев), где первый месяц выведенного года равен следующему месяцу, после номера месяца, который прописан в параметре функции.

[Подробная информация о функции CALENDARAUTO и примеры формул >>>](#)

## DAX функция CALENDAR

CALENDAR () — календарь, создаваемый вручную. Возвращает таблицу с одним столбцом [Date], содержащим непрерывные даты от стартовой до конечной дат, указанных в параметрах.

Синтаксис:

CALENDAR (Стартовая Дата; Конечная Дата)

Где, стартовые и конечные даты — это любые выражения, возвращающие значения в формате datetime (в том числе, ссылки на исходные столбцы с датами).

[Подробная информация о функции CALENDAR и примеры формул >>>](#)

## DAХ функция DATESBETWEEN

DATESBETWEEN () — ограничитель календаря по произвольным датам. Создает ограниченную таблицу из дат, начиная со стартовой и заканчивая конечной датами, расположенными в исходном столбце.

Синтаксис:

DATESBETWEEN ([Дата]; Стартовая Дата; Конечная Дата)

Где:

- [Дата] — столбец дат в исходной таблице
- Стартовая и Конечная Даты — дата в формате datetime

! — Стартовые и конечные даты, указанные в параметрах функции, входят в состав выводимых дат.

[Подробная информация о функции DATESBETWEEN и примеры формул >>>](#)

## DAX функция DATESINPERIOD

DATESINPERIOD () — ограничитель календаря по типовым интервалам. Создает ограниченную таблицу из дат, начиная со стартовой даты, расположенной в исходном столбце и продолжая в течение указанного типового интервала (количеством дней, месяцев, кварталов, лет).

Синтаксис:

DATESINPERIOD ([Дата]; Стартовая Дата; Количество Интервалов; Интервал)

Где:

- [Дата] — столбец дат в исходной таблице
- Стартовая Дата — дата в формате datetime (указанная дата входит в состав выводимых дат)
- Количество Интервалов — целое число, характеризующее количество интервалов
- Интервал — наименование интервала: year (год), quarter (квартал), month (месяц), day (день)

DATESINPERIOD очень похожа на DATESBETWEEN. Разница заключается лишь в том, что DATESINPERIOD ограничивает исходный набор дат именно определенными целыми интервалами.

[Подробная информация о функции DATESINPERIOD и примеры формул >>>](#)

## DAX функция DATEADD

DATEADD () — создает таблицу со столбцом из дат, сдвинутых назад (в прошлое) или вперед (в будущее) на заданное количество интервалов от даты текущего контекста.

Синтаксис:

DATEADD ([Дата]; Количество Интервалов; Интервал)

Где:

- [Дата] — столбец из дат или выражений, возвращающих даты
- Количество Интервалов — целое число, характеризующее количество интервалов, которое нужно добавить (вычесть) к дате в текущем контексте. Если указано положительное число — то интервалы добавляются, если указано отрицательное число — то интервалы вычитаются
- Интервал — тип интервала: year (год), quarter (квартал), month (месяц), day (день)

! — Для правильной работы функции необходимо в качестве ее первого параметра [Дата] использовать столбец из таблицы непрерывных дат в Power BI, то есть, создавать отдельную связанную таблицу «Календарь» с непрерывным перечислением всех дат.

[Подробная информация о функции DATEADD и примеры формул >>>](#)

## DAX функция PARALLELPERIOD

PARALLELPERIOD () — возвращает таблицу из дат, смещенных во времени вперед или назад параллельно текущей дате в текущем контексте на заданное количество интервалов.

Синтаксис:

PARALLELPERIOD ([Дата]; Количество Интервалов; Интервал)

Где:

- [Дата] — столбец из дат или выражений, возвращающих даты
- Количество Интервалов — целое число, характеризующее количество интервалов, которое нужно добавить (вычесть) к дате в текущем контексте. Если указано положительное число — то интервалы добавляются, если указано отрицательное число — то интервалы вычитаются
- Интервал — тип интервала: year (год), quarter (квартал), month (месяц)

! — Для правильной работы функции необходимо в качестве ее первого параметра [Дата] использовать столбец из таблицы непрерывных дат в Power BI, то есть, создавать отдельную связанную таблицу «Календарь» с непрерывным перечислением всех дат.

[Подробная информация о функции PARALLELPERIOD и примеры формул >>>](#)

## DAX функция

# SAMEPERIODLASTYEAR

SAMEPERIODLASTYEAR () — возвращает таблицу из дат, смещенных на 1 год назад относительно дат текущего контекста.

Синтаксис:

SAMEPERIODLASTYEAR ([Дата])

На самом деле, SAMEPERIODLASTYEAR это упрощенный вариант DAX функции DATEADD, а именно, упрощенный ее вариант с конкретными настройками параметров:

SAMEPERIODLASTYEAR ([Дата]) = DATEADD ([Дата]; -1; year)

[Подробная информация о функции SAMEPERIODLASTYEAR и примеры формул >>>](#)

## DAX функции PREVIOUSYEAR,

## PREVIOUSQUARTER,

## PREVIOUSMONTH, PREVIOUSDAY



Все функции группы PREVIOUS возвращают таблицу со столбцом из дат предыдущего периода на основе даты текущего контекста. Где, предыдущий период в:

- PREVIOUSYEAR () — равен предыдущему году
- PREVIOUSQUARTER () — равен предыдущему кварталу
- PREVIOUSMONTH () — равен предыдущему месяцу
- PREVIOUSDAY () — равен предыдущему дню

Синтаксис:

PREVIOUSYEAR ([Дата]; "Конец Года")

PREVIOUSQUARTER ([Дата])

PREVIOUSMONTH ([Дата])

PREVIOUSDAY ([Дата])

Где:

- [Дата] — столбец из дат или выражений, возвращающих даты
- «Конец Года» — текстовая дата, записанная в виде «02/01» («день/месяц»). Определяет дату окончания года (по умолчанию 31 декабря). Необязательный параметр

! — Для бесперебойной работы группы функций PREVIOUS необходимо в качестве их параметров [Дата] использовать столбец из календаря непрерывных дат в Power BI, то есть, создавать отдельную связанную таблицу «Календарь» с непрерывным перечислением всех дат.

[Подробная информация о функциях PREVIOUSYEAR, PREVIOUSQUARTER, PREVIOUSMONTH, PREVIOUSDAY и примеры формул >>>](#)

# DAX функции NEXTYEAR, NEXTQUARTER, NEXTMONTH И NEXTDAY

Все функции группы NEXT возвращают таблицу со столбцом из дат следующего периода на основе даты текущего контекста. Где, следующий период в:

- NEXTYEAR () — равен следующему году
- NEXTQUARTER () — равен следующему кварталу
- NEXTMONTH () — равен следующему месяцу
- NEXTDAY () — равен следующему дню

Синтаксис:

NEXTYEAR ([Дата]; "Конец Года")

NEXTQUARTER ([Дата])

NEXTMONTH ([Дата])

NEXTDAY ([Дата])

Где:

- [Дата] — столбец из дат или выражений (табличных, логических и прочих), возвращающих даты
- «Конец Года» — (необязательный параметр) текстовая дата, записанная в виде «15/06» («день/месяц»). Определяет дату окончания года (по умолчанию «31/12», то есть, 31 декабря)

! — Для безошибочной работы группы функций NEXT необходимо в качестве их параметров [Дата] использовать столбец из календаря

непрерывных дат в Power BI, то есть, создавать отдельную связанную таблицу «Календарь» с непрерывным перечислением всех дат.

[Подробная информация о функциях NEXTYEAR, NEXTQUARTER, NEXTMONTH и NEXTDAY и примеры формул >>>](#)

## DAХ функции DATESYTD, DATESQTD И DATESMTD

DATESYTD () — создает таблицу, со столбцом из дат от начала года до текущего дня в рамках текущего контекста.

DATESQTD () — создает таблицу, со столбцом из дат от начала квартала до текущего дня в рамках текущего контекста.

DATESMTD () — создает таблицу, со столбцом из дат от начала месяца до текущего дня в рамках текущего контекста.

Синтаксис:

DATESYTD ([Дата]; "Конец Года")

DATESQTD ([Дата])

DATESMTD ([Дата])

Где:

- [Дата] — столбец из дат или выражений, возвращающих даты
- «Конец Года» — текстовая дата, записанная в виде «01/05» («день/месяц»). Определяет дату окончания года (по умолчанию 31 декабря). Необязательный параметр

! — Для безошибочной работы функций DATESYTD, DATESQTD и DATESMTD необходимо в качестве их параметров [Дата] использовать столбец из календаря непрерывных дат в Power BI, то есть, создавать отдельную связанную таблицу «Календарь» с непрерывным перечислением всех дат.

[Подробная информация о функциях DATESYTD, DATESQTD, DATESMTD и примеры формул >>>](#)

## DAX функции TOTALYTD, TOTALQTD, TOTALMTD

TOTALYTD () — вычисляет заданное выражение от начала года до текущего дня в рамках текущего контекста.

TOTALQTD () — вычисляет заданное выражение от начала квартала до текущего дня в рамках текущего контекста.

TOTALMTD () — вычисляет заданное выражение от начала месяца до текущего дня в рамках текущего контекста.

Синтаксис:

TOTALYTD (Выражение; [Дата]; Фильтр; "Конец Года")

TOTALQTD (Выражение; [Дата]; Фильтр)

TOTALMTD (Выражение; [Дата]; Фильтр)

Где:

- Выражение — выражение, возвращающее единственное скалярное значение
- [Дата] — столбец из дат или выражений, возвращающих даты
- Фильтр — (необязательный параметр) необходимые фильтры для вычисления выражения
- «Конец Года» — (необязательный параметр) текстовая дата, записанная в виде «01/07» («день/месяц»). Определяет дату окончания года (по умолчанию «31/12»)

! — Для корректной работы функций TOTALYTD, TOTALQTD и TOTALMTD необходимо в качестве их параметров [Дата] использовать столбец из календаря непрерывных дат в Power BI, то есть, создавать отдельную связанную таблицу «Календарь» с непрерывным перечислением всех дат.

[Подробная информация о функциях TOTALYTD, TOTALQTD, TOTALMTD и примеры формул >>>](#)

## DAX функции

### CLOSINGBALANCEYEAR,

### CLOSINGBALANCEQUARTER И

### CLOSINGBALANCEMONTH

CLOSINGBALANCEYEAR () — вычисляет заданное выражение на последнюю дату года текущего контекста.

CLOSINGBALANCEQUARTER () — вычисляет заданное выражение на последнюю дату квартала текущего контекста.

CLOSINGBALANCEMONTH () — вычисляет заданное выражение на последнюю дату месяца текущего контекста.

Синтаксис:

CLOSINGBALANCEYEAR (Выражение; [Дата]; Фильтр; "Конец Года")

CLOSINGBALANCEQUARTER (Выражение; [Дата]; Фильтр)

CLOSINGBALANCEMONTH (Выражение; [Дата]; Фильтр)

Где:

- Выражение — выражение, возвращающее единственное скалярное значение
- [Дата] — столбец из дат или выражений, возвращающих даты
- Фильтр — (необязательный параметр) необходимые фильтры для вычисления выражения
- «Конец Года» — (необязательный параметр) текстовая дата, записанная в виде «01/06» («день/месяц»). Определяет дату окончания года (по умолчанию «31/12»)

[Подробная информация о функциях CLOSINGBALANCEYEAR, CLOSINGBALANCEQUARTER, CLOSINGBALANCEMONTH и примеры формул >>>](#)

## DAX функции

### OPENINGBALANCEYEAR,

### OPENINGBALANCEQUARTER И

### OPENINGBALANCEMONTH

OPENINGBALANCEYEAR () — вычисляет заданное выражение на первую дату года текущего контекста.

OPENINGBALANCEQUARTER () — вычисляет заданное выражение на первую дату квартала текущего контекста.

OPENINGBALANCEMONTH () — вычисляет заданное выражение на первую дату месяца текущего контекста.

Синтаксис:

OPENINGBALANCEYEAR (Выражение; [Дата]; Фильтр; "Конец Года")

OPENINGBALANCEQUARTER (Выражение; [Дата]; Фильтр)

OPENINGBALANCEMONTH (Выражение; [Дата]; Фильтр)

Где:

- **Выражение** — выражение, возвращающее единственное скалярное значение
- **[Дата]** — столбец из дат или выражений, возвращающих даты
- **Фильтр** — (необязательный параметр) необходимые фильтры для вычисления выражения
- **«Конец Года»** — (необязательный параметр) текстовая дата, записанная в виде «01/06» («день/месяц»). Определяет дату окончания года (по умолчанию «31/12»)

[Подробная информация о функциях OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR и примеры формул >>>](#)

## DAX функции ENDOFYEAR, ENDOFQUARTER И ENDOFMONTH

ENDOFYEAR () — возвращает последнюю известную дату года в рамках текущего контекста.

ENDOFQUARTER () — возвращает последнюю известную дату квартала в рамках текущего контекста.

ENDOFMONTH () — возвращает последнюю известную дату месяца в рамках текущего контекста.

Синтаксис:

ENDOFYEAR ([Дата]; "Конец Года")

ENDOFQUARTER ([Дата])

ENDOFMONTH ([Дата])

Где:

- [Дата] — столбец из дат или выражений, возвращающих даты
- «Конец Года» — (необязательный параметр) текстовая дата, записанная в виде «01/06» («день/месяц»). Определяет дату окончания года (по умолчанию «31/12»)



[Подробная информация о функциях ENDOFYEAR, ENDOFQUARTER, ENDOFMONTH и примеры формул >>>](#)

# DAX функции STARTOFYEAR, STARTOFQUARTER, STARTOFMONTH

STARTOFYEAR () — возвращает первую известную дату года в рамках текущего контекста.

STARTOFQUARTER () — возвращает первую известную дату квартала в рамках текущего контекста.

STARTOFMONTH () — возвращает первую известную дату месяца в рамках текущего контекста.

Синтаксис:

STARTOFYEAR ([Дата]; "Конец Года")

STARTOFQUARTER ([Дата])

STARTOFMONTH ([Дата])

Где:

- [Дата] — столбец из дат или выражений, возвращающих даты
- «Конец Года» — (необязательный параметр) текстовая дата, записанная в виде «01/06» («день/месяц»). Определяет дату окончания года (по умолчанию «31/12»)

[Подробная информация о функциях STARTOFYEAR, STARTOFQUARTER, STARTOFMONTH и примеры формул >>>](#)

## DAX функции

### FIRSTDATE И LASTDATE

FIRSTDATE () — возвращает первую известную дату для указанного столбца дат в текущем контексте.

LASTDATE () — возвращает последнюю известную дату для указанного столбца дат в текущем контексте.

Синтаксис:

FIRSTDATE ([Дата])

LASTDATE ([Дата])

[Подробная информация о функциях FIRSTDATE, LASTDATE и примеры формул >>>](#)

## DAX функция EOMONTH

EOMONTH () — возвращает дату конца месяца с учетом прописанного отступа в месяцах от указанной в параметрах даты.

Синтаксис:

EOMONTH (Дата; Отступ)

Где:

- Дата — дата, записанная в текстовом или datetime форматах
- Отступ — отступ в месяцах (отрицательное значение — отступ до текущей даты, положительное значение — после текущей даты)

[Подробная информация о функции EOMONTH и примеры формул >>>](#)

## DAX функция EDATE

EDATE () — возвращает дату, прописанную в параметрах и перемещенную на указанное количество месяцев до прописанной даты или после.

Синтаксис:

EDATE (Дата; Отступ)

Где:

- Дата — дата, записанная в текстовом или datetime форматах
- Отступ — отступ в месяцах (отрицательное значение — отступ до текущей даты, положительное значение — после текущей даты)

[Подробная информация о функции EDATE и примеры формул >>>](#)

## DAX функция CALCULATE

CALCULATE () — вычисляет выражение, измененное внутренними фильтрами.

Синтаксис:

CALCULATE (Выражение; Фильтр 1; Фильтр 2; ...; Фильтр N)

Где:

- Выражение — то выражение, которое нужно вычислить (обязательный параметр для CALCULATE, без него эта функция работать не будет)
- Фильтр — условия фильтров (необязательный параметр, количество фильтров может быть от 0 до многих и все они сочетаются в режиме «и»)

Фильтры в CALCULATE не могут ссылаться на различные меры или вложенные функции CALCULATE. При этом, в условиях фильтров могут использоваться какие-либо другие функции DAX, вычисляющие одно скалярное значение или создающие запрос уточнения одного скалярного значения.

Если говорить проще про функцию CALCULATE, то она при помощи своих условий фильтров способна дополнить, заменить или полностью удалить все предыдущие наложенные фильтры на выражение, указанное в первом параметре.

[Подробная информация о функции CALCULATE и примеры формул >>>](#)

## DAX функция CALCULATETABLE

CALCULATETABLE () — вычисляет табличное выражение, измененное внутренними фильтрами.

Синтаксис:

CALCULATETABLE (Табличное выражение; Фильтр 1; Фильтр 2; ...; Фильтр N)

Где:

- Табличное выражение — то табличное выражение, которое нужно вычислить (обязательный параметр для CALCULATETABLE, без него эта функция работать не будет)
- Фильтр — условия фильтров (необязательный параметр, количество фильтров может быть от 0 до многих и все они сочетаются в режиме «и»)

Функцию CALCULATETABLE нельзя путать с функцией фильтра таблиц в DAX — FILTER. Эта функция просто фильтрует таблицы по заданным условиям фильтра. А CALCULATETABLE, в свою очередь, не просто фильтрует, а заменяет или удаляет фильтры, что в итоге дает совершенно другой результат.

[Подробная информация о функции CALCULATETABLE и примеры формул >>>](#)

## DAX функция FILTER

FILTER () — возвращает таблицу, фильтруя исходную таблицу по заданным в параметрах фильтрам. Фильтрация производится по строкам, то есть, возвращаются все столбцы исходной таблицы, а строки только те, которые удовлетворяют условию фильтра.

Синтаксис:

FILTER ('Таблица'; Фильтр)

Где:

- 'Таблица' — исходная таблица или табличное выражение, которую необходимо отфильтровать
- Фильтр — логическое выражение, которое сравнивается с каждой строчкой таблицы, указанной в первом параметре

[Подробная информация о функции FILTER и примеры формул >>>](#)

## DAX функция ALL

ALL () — возвращает полную исходную таблицу или столбец, игнорируя все, ранее наложенные на них, фильтры. Иначе говоря, удаляет все ранее наложенные на таблицу или столбец фильтры.

Синтаксис:

ALL ('Таблица') — возвращает все строки исходной таблицы

ALL ([Столбец 1]; [Столбец 2]; ...; [Столбец N]) — возвращает столбец (столбцы) со всеми уникальными значениями исходного столбца (столбцов)

Исходя из определения синтаксиса функции ALL, который описан выше, можно сказать, что ALL работает в двух разных режимах.

1. Когда в параметре указана просто таблица — то возвращается исходная таблица в том виде, какая она есть на самом деле.
2. Когда в параметре функции ALL указан столбец (столбцов может быть от 1 до нескольких), то возвращается уже не полностью исходный столбец (его копия), а только уникальные значения исходного столбца, но с учетом того, что ранее наложенные фильтры удаляются.

[Подробная информация о функции ALL и примеры формул >>>](#)

## DAХ функция ALLEXCEPT

ALLEXCEPT () — удаляет все наложенные фильтры с указанной таблицы в первом параметре, кроме тех столбцов, которые указаны во втором и последующих параметрах.

Синтаксис:

ALLEXCEPT ('Таблица'; [Столбец 1]; [Столбец 2]; ...; [Столбец N];)

[Подробная информация о функции ALLEXCEPT и примеры формул >>>](#)

## DAX функция ALLSELECTED

ALLSELECTED () — удаляет последний наложенный уровень фильтра.

Синтаксис:

ALLSELECTED () — удаляет последний наложенный уровень фильтра со всех таблиц модели данных

ALLSELECTED ('Таблица') — удаляет последний наложенный уровень фильтра с указанной таблицы

ALLSELECTED ([Столбец]) — удаляет последний наложенный уровень фильтра только с одного указанного столбца

[Подробная информация о функции ALLSELECTED и примеры формул >>>](#)

## DAX функция ALLNOBLANKROW

ALLNOBLANKROW () — возвращает полную исходную таблицу или столбец без учета пустых строк, игнорируя все, ранее наложенные фильтры.



Синтаксис:

ALLNOBLANKROW ('Таблица') — возвращает все строки исходной таблицы без учета пустых строк

ALLNOBLANKROW ([Столбец 1]; [Столбец 2]; ...; [Столбец N]) — возвращает столбец (столбцы) со всеми уникальными значениями исходного столбца (столбцов) без учета пустых строк

ALLNOBLANKROW () — функция, полностью идентичная функции ALL, за исключением того, что ALLNOBLANKROW не учитывает, автоматически создаваемые DAX, пустые строки.

Ситуация с автоматически создаваемыми DAX пустыми строками возможна тогда, когда между связанными таблицами различаются значения ключевых столбцов.

[Подробная информация о функции ALLNOBLANKROW >>>](#)

## DAX функция VALUES

VALUES () — функция, которая создает таблицу, состоящую из одного столбца с набором уникальных значений исходной таблицы или столбца.

Синтаксис:

VALUES ('Таблица')

VALUES ([Столбец])

[Подробная информация о функции VALUES и примеры формул >>>](#)

## DAX функция DISTINCT

DISTINCT () — DAX функция, которая возвращает столбец с уникальными значениями исходного столбца или таблицы. В своей практике я эту функцию очень часто использую при создании так называемых таблиц справочников (измерений) на основе исходного столбца из таблицы фактов.

Таблицы фактов обычно содержат очень большой объем данных (миллионы строк), которые могут повторяться. Например, таблица фактов «Заявки», в которой в каждой строке указана информация по одной заявке — сумма, затраты, менеджер, дата сделки и т.д.

Таблицы справочников (измерений) гораздо меньше, количество строк в них может составлять десятки или сотни, в крайних случаях (в больших корпорациях) несколько тысяч. В них содержатся только уникальные значения, расшифровывающие то или иное измерение, например, менеджеров (имя, фамилия, телефон, адрес проживания и т.д.)

Синтаксис:

DISTINCT (['Таблица'])

DISTINCT ([Столбец])

[Подробная информация о функции DISTINCT и примеры формул >>>](#)

## DAX функция RELATED

RELATED () — находясь в одной таблице, позволяет в рамках контекста строки получить связанное значение из второй таблицы по связи «Многие к одному».

Синтаксис:

RELATED ([Столбец])

[Подробная информация о функции RELATED и примеры формул >>>](#)

## DAX функция RELATEDTABLE

RELATEDTABLE () — находясь в одной таблице, возвращает связанную таблицу значений из второй таблицы по связи «Один ко многим», где одна строка соответствуем многим строкам.

Синтаксис:

RELATEDTABLE ('Таблица')

Функция RELATEDTABLE, в отличие от RELATED, уже не возвращает какое-то одно скалярное значение, она возвращает именно связанную таблицу значений, с которыми возможно совершать те или иные арифметические действия, например посчитать количество строк.

[Подробная информация о функции RELATEDTABLE и примеры формул >>>](#)

## DAX функция ADDCOLUMNS

ADDCOLUMNS () — название этой функции говорит само за себя: добавить столбец. Она создает вычисляемый столбец в таблице.

Синтаксис:

```
ADDCOLUMNS (  
    'Таблица';  
    "Имя столбца 1"; Выражение 1;  
    "Имя столбца 2"; Выражение 2;  
    "..."; ...;  
    "Имя столбца N"; Выражение N  
)
```

Где:

- 'Таблица' — существующая таблица или табличное выражение к которому добавляется столбец
- «ИмяСтолбца» — имя создаваемого столбца
- Выражение — вычисляемое выражение для каждой строки создаваемого столбца

Функцию ADDCOLUMNS использовать саму по себе нет никакого смысла, она используется только совместно с какими-либо другими функциями в DAX, создавая виртуальный вычисляемый столбец, который будет использоваться другой функцией.

Это нужно для того, чтобы не создавать вычисляемый столбец в физической памяти, чтобы он не занимал место оперативной памяти ПК.

[Подробная информация о функции ADDCOLUMNS и примеры формул >>>](#)

## DAX функция DATATABLE

DATATABLE () — возвращает таблицу с нужными данными (позволяет создать таблицу «с нуля»).

Синтаксис:

DATATABLE (

"Имя столбца 1"; Тип Данных Столбца 1;

"Имя столбца 2"; Тип Данных Столбца 2;

"Имя столбца N"; Тип Данных Столбца N;

{

{Значение 1; Значение 2; Значение N};

{Значение 1; Значение 2; Значение N};

{Значение 1; Значение 2; Значение N};

{Значение 1; Значение 2; Значение N}

}

)

Где:

- «Имя столбца» — имя создаваемого столбца в таблице
- Тип Данных Столбца — тип данных создаваемого столбца: INTEGER (целые числа), DOUBLE (дробные числа), STRING

(текстовая строка), BOOLEAN (логический тип), CURRENCY (денежный тип), DATETIME (дата и время)

- {Значение 1; Значение 2; Значение 3} — строки создаваемой таблицы
- Значение 1 — любое значение или выражение, соответствующие создаваемому столбцу и его типу данных

[Подробная информация о функции DATATABLE и примеры формул >>>](#)

## DAX функция SUMMARIZE

SUMMARIZE () — создает сводную таблицу с агрегированными итогами по выбранным группам.

Синтаксис:

```
SUMMARIZE (  
    'Таблица';  
    [Столбец 1]; [Столбец 2]; ...; [Столбец N];  
    "Имя столбца 1"; Выражение 1;  
    "Имя столбца 2"; Выражение 2;  
    ...; ...;  
    "Имя столбца N"; Выражение N  
)
```

Где:

- 'Таблица' — исходная существующая таблица или выражение, возвращающее таблицу, значения которой мы хотим сгруппировать
- [Столбец] — столбец для группировки
- «Имя столбца» — имя создаваемого столбца для значений группировки
- Выражение — вычисляемое выражение для значений группировки

[Подробная информация о функции SUMMARIZE и примеры формул >>>](#)

## DAX функция

# SUMMARIZECOLUMNS

SUMMARIZECOLUMNS () — очень похожая DAX функция на GROUPBY, а тем более на SUMMARIZE. Она также создает сводную таблицу, но, в данном случае, с возможностью фильтрации группируемых столбцов.

Синтаксис:

SUMMARIZECOLUMNS (

[Столбец 1]; [Столбец 2]; ...; [Столбец N];

Фильтр;

"Имя Столбца 1"; Выражение 1;

"Имя Столбца 1"; Выражение 1;

```
"..."; ...;
```

```
"Имя Столбца N"; Выражение N
```

```
)
```

Синтаксис функции SUMMARIZECOLUMNS очень похож на синтаксис GROUPBY, за исключением лишь той разницы, что:

- в первом параметре не нужно прописывать исходную таблицу;
- в выражении мы можем использовать не только X функции, но и любые другие функции агрегирования
- здесь появился еще один параметр — фильтр, по которому производится фильтрация столбцов для группировки

[Подробная информация о функции SUMMARIZECOLUMNS и примеры формул >>>](#)

## DAX функция GROUPBY

GROUPBY () — создает сводную таблицу, сгруппированную по указанным столбцам (название столбцов конфигурируется из названия самой исходной таблицы и названия исходного столбца для группировки).

Синтаксис:

```
GROUPBY (
```

```
'Таблица';
```

```
[Столбец 1]; [Столбец 2]; ...; [Столбец N];
```



"Имя столбца 1"; Выражение 1;

"Имя столбца 2"; Выражение 2;

"..."; ...;

"Имя столбца N"; Выражение N

)

Где:

- 'Таблица' — исходная существующая таблица или табличное выражение, значения которых мы хотим сгруппировать
- [Столбец] — столбец для группировки
- «Имя столбца» — имя создаваемого столбца для значений группировки
- Выражение — вычисляемое выражение для значений группировки

! – Выражение обязательно должно содержать статистическую DAX функцию формата X (SUMX, MAXX, AVERAGEX...), во входных параметрах которой, в качестве таблицы подставляется служебное выражение CURRENTGROUP ().

[Подробная информация о функции GROUPBY и примеры формул >>>](#)

## DAX функция ROW

ROW () — функция итогов. Создает таблицу из одной строки с итогами по выбранным столбцам исходной таблицы (итоги по сумме, количеству, средним значениям и так далее).

Синтаксис:

ROW (

"Имя столбца 1"; Выражение 1;

"Имя столбца 2"; Выражение 2;

"Имя столбца N"; Выражение N

)

Где:

- «Имя столбца» — имя столбца в создаваемой новой таблице
- Выражение — выражение агрегации для вычисления значения столбца

[Подробная информация о функции ROW и примеры формул >>>](#)

## DAX функция TOPN

TOPN () — создает таблицу из первых N строк исходной таблицы, с возможностью выбора столбца исходной таблицы, по которой будут выбираться первые N строк.

Синтаксис:

TOPN (N строк; 'Таблица'; [Столбец]; Порядок Сортировки)

Где:

- N строк — число первых строк (если 0, то возвратится пустая таблица)
- 'Таблица' — исходная таблица

- [Столбец] — столбец, по которому выбирается количество N строк (необязательный параметр)
- Порядок Сортировки — порядок сортировки выбора N строк (ASC — по возрастанию, DESC — по убыванию)

! — Результаты в самой выводимой таблице по выводу не сортируются

! — При наличии одинаковых значений в исходной таблице, в создаваемой таблице выводятся все строки с одинаковыми значениями

[Подробная информация о функции TOPN и примеры формул >>>](#)

## ДAX функция EXCEPT

EXCEPT () — создает таблицу из строк левой таблицы, которых нет в правой. В обеих таблицах должна быть идентичная структура столбцов.

Синтаксис:

EXCEPT ('Левая Таблица'; 'Правая Таблица')

[Подробная информация о функции EXCEPT и примеры формул >>>](#)

## DAX функция INTERSECT

INTERSECT () — создает таблицу из строк левой таблицы, которые присутствуют в правой таблице (пересечение строк двух таблиц с сохранением дубликатов). В обеих таблицах должна быть идентичная структура столбцов.

Синтаксис:

INTERSECT ('Левая Таблица'; 'Правая Таблица')

[Подробная информация о функции INTERSECT и примеры формул >>>](#)

## DAX функция UNION

UNION () — создает новую таблицу, объединяя любое количество таблиц с единой структурой столбцов. То есть, объединяет строки нескольких одинаковых таблиц в одну единую таблицу.

Синтаксис:

UNION ('Таблица 1'; 'Таблица 2'; ...; 'Таблица N')

[Подробная информация о функции UNION и примеры формул >>>](#)

## DAX функция

# NATURALINNERJOIN

NATURALINNERJOIN () — создает новую таблицу, объединяя столбцы левой и правой таблицы и выводит только те строки, которые имеют одно и то же значение. Функция работает на основе внутренних DAX связей в Power BI или Power Pivot.

Синтаксис:

NATURALINNERJOIN ('Левая Таблица'; 'Правая Таблица')

! — Названия столбцов в левой и правой таблицах должны быть разными

! — Обе таблицы должны быть объединены внутренней связью в DAX

[Подробная информация о функции NATURALINNERJOIN и примеры формул >>>](#)

## DAX функция

# NATURALLEFTOUTERJOIN

NATURALLEFTOUTERJOIN () — создает новую таблицу, объединяя столбцы левой и правой таблицы и выводит все строки из правой таблицы, а также из левой, которые имеют одно и то же значение с правой. Функция работает на основе внутренних DAX связей в Power BI или Power Pivot.

Синтаксис:

NATURALLEFTOUTERJOIN ('Левая Таблица'; 'Правая Таблица')

! — Названия столбцов в левой и правой таблицах должны быть разными

! — Обе таблицы должны быть объединены внутренней связью в DAX

[Подробная информация о функции NATURALLEFTOUTERJOIN и примеры формул >>>](#)

## DAX функция GENERATESERIES

GENERATESERIES () — создает таблицу последовательных значений (ряд чисел).

Синтаксис:

GENERATESERIES (Стартовое Число; Конечное Число; Инкремент)

Где:

- Стартовое Число — стартовое число последовательности
- Конечное Число — конечное число последовательности
- Инкремент — приращение (шаг, увеличение) последовательности (! — инкремент должен быть больше 0)

[Подробная информация о функции GENERATESERIES и примеры формул >>>](#)

## DAX функция CROSSJOIN

CROSSJOIN () — создает таблицу из всех пересечений строк всех таблиц, входящих в параметры функции.

Синтаксис:

CROSSJOIN ('Таблица 1'; 'Таблица 2'; 'Таблица N')

Где: Таблица — исходная таблица или табличное выражение

! — Названия столбцов во всех таблицах должны быть разными

[Подробная информация о функции CROSSJOIN и примеры формул >>>](#)

## DAX функции GENERATE и GENERATEALL

GENERATE () и GENERATEALL () — создают таблицу из всех пересечений строк двух таблиц, входящих в параметры функций. То есть, для каждой строки первой таблицы возвращаются (по очереди вычисляются) все строки из второй.

Синтаксис:

GENERATE ('Таблица 1'; 'Таблица 2')

GENERATEALL ('Таблица 1'; 'Таблица 2')

Где:

Будуев Антон © [biprosto.ru](http://biprosto.ru) / BI — это просто

Страница 95

Таблица — исходная таблица или табличное выражение

! — Названия столбцов в обеих таблицах должны быть разными

Обе функции GENERATE и GENERATEALL работают практически одинаково, за исключением того, как они обрабатывают пустую вторую таблицу.

[Подробная информация о функциях GENERATE, GENERATEALL и примеры формул >>>](#)

## DAX функция ISBLANK

ISBLANK () — проверяет переданное в параметрах значение. Если значение пусто, то функция возвращает TRUE, в противном случае возвращает FALSE.

Синтаксис:

ISBLANK (Значение)

[Подробная информация о функциях ISBLANK и примеры формул >>>](#)



## DAX функция ISNUMBER

ISNUMBER () — проверяет переданное в параметрах значение. Если значение является числом, то функция возвращает TRUE, во всех остальных случаях возвращает FALSE

Синтаксис:

ISNUMBER (Значение)

[Подробная информация о функциях ISNUMBER и примеры формул >>>](#)

## DAX функция ISEVEN

ISEVEN () — проверяет переданное в параметрах число. Если число четное, то функция возвращает TRUE, во всех остальных случаях возвращает FALSE

Синтаксис:

ISEVEN (Число)

[Подробная информация о функциях ISEVEN и примеры формул >>>](#)

## DAX функции ISTEXT, ISNONTEXT

ISTEXT () — проверяет переданное в параметрах значение. Если значение является текстом, то функция возвращает TRUE, во всех остальных случаях возвращает FALSE.

ISNONTEXT () — проверяет переданное в параметрах значение. Если значение не является текстом, то функция возвращает TRUE, если значение является текстом, то возвращает FALSE

То есть, функции ISTEXT и ISNONTEXT обратные друг другу.

Синтаксис:

ISTEXT (Значение)

ISNONTEXT (Значение)

[Подробная информация о функциях ISTEXT, ISNONTEXT и примеры формул >>>](#)

## DAX функция ISERROR

ISERROR () — относится к информационным функциям DAX. Она выводит значение TRUE (Истина), если значение, входящее в ее параметр вычисляется с ошибкой, а также, значение FALSE (Ложь), если ошибок нет.

Синтаксис:

ISERROR (Значение)

[Подробная информация о функции ISERROR и примеры формул >>>](#)

## DAX функция ISLOGICAL

ISLOGICAL () — проверяет переданное в параметрах значение. Если значение является логическим, то функция возвращает TRUE, во всех остальных случаях возвращает FALSE.

Синтаксис:

ISLOGICAL (Значение)

[Подробная информация о функции ISLOGICAL и примеры формул >>>](#)

## DAX функции HASONEVALUE и HASONEFILTER

HASONEVALUE () — проверяет столбец и если в результате всех перекрестных фильтров в нем осталось одно единственное значение (одна строка), то функция возвращает TRUE, во всех остальных случаях — FALSE.

HASONEFILTER () — проверяет столбец и если в результате прямого фильтра в нем осталось одно единственное значение (одна строка), то функция возвращает TRUE, во всех остальных случаях — FALSE.

Синтаксис:

HASONEVALUE ([Столбец])

HASONEFILTER ([Столбец])

[Подробная информация о функциях HASONEVALUE, HASONEFILTER и примеры формул >>>](#)

## DAX функции ISFILTERED и ISCROSSFILTERED

ISFILTERED () — проверяет столбец и если он находится под воздействием прямых фильтров, возвращает логическое значение TRUE, во всех остальных случаях — FALSE.

ISCROSSFILTERED () — проверяет столбец и если он находится под воздействием прямых или косвенных (кросс) фильтров, возвращает логическое значение TRUE, во всех остальных случаях — FALSE.

Синтаксис:

ISFILTERED ([Столбец])

ISCROSSFILTERED ([Столбец])

[Подробная информация о функциях ISCROSSFILTERED, ISFILTERED и примеры формул >>>](#)

## DAX функции FIRSTNONBLANK и LASTNONBLANK

FIRSTNONBLANK () — возвращает первое известное значение из столбца, где вычисленное выражение является непустым.

LASTNONBLANK () — возвращает последнее известное значение из столбца, где вычисленное выражение является непустым.

Синтаксис:

FIRSTNONBLANK ([Столбец]; Выражение)

LASTNONBLANK ([Столбец]; Выражение)

[Подробная информация о функциях FIRSTNONBLANK, LASTNONBLANK и примеры формул >>>](#)

## DAX функция EARLIER

EARLIER () — возвращает по указанному в параметрах столбцу, его значение на высших уровнях контекста строки.

Синтаксис:

EARLIER ([Столбец]; Номер Уровня Контекста)

[Подробная информация о функции EARLIER и примеры формул >>>](#)

## VAR И RETURN – переменные в DAX

Любая переменная в DAX всегда состоит из обязательного синтаксиса:

VAR Name 1 = Код Переменной 1

VAR Name 2 = Код Переменной 2

RETURN

Name 1 \* Name 2

Где:

- Name — имя создаваемой переменной
- Код Переменной — любая DAX формула, характеризующая саму переменную
- VAR — служебное слово (специальная функция), всегда предшествующая объявлению имени переменной при ее создании

- RETURN — служебное слово (специальная функция), оканчивающая создание (объявление) блока переменных, после него идет основной код формулы
- Name 1 \* Name 2 — основной код формулы (меры, вычисляемого столбца или таблицы)

[Подробная информация о создании переменных в DAX при помощи VAR и RETURN и примеры формул >>>](#)

## DAX функция RANKX

RANKX () — возвращает ранг выражения, вычисляемого в рамках текущего контекста исходного списка значений для каждой строки текущей таблицы.

Синтаксис:

RANKX ('Таблица'; Выражение; Значение; Порядок; Равные Значения)

Где:

- Таблица — исходная таблица или табличное выражение
- Выражение — любое выражение, вычисляемое для каждой строки исходной таблицы с целью формирования полного списка возможных значений для ранжирования
- Значение — (необязательный параметр) любое выражение, ранг которого необходимо найти. По умолчанию, если этот параметр не указан, вместо этого параметра возвращается значение выражения из текущей строки

- Порядок — (необязательный параметр) вид ранжирования параметра «Значение»:
  - DESC — упорядоченный порядок рангов по убыванию (по умолчанию);
  - ASC — по возрастанию
- Равные Значения — (необязательный параметр) способ определения ранга при наличии равных значений:
  - Skip — (по умолчанию) ранг значения, идущего после ряда одинаковых значений, определяется как ранг равных значений плюс количество этих равных значений. Например, имеется 3 равных значения с рангом 5, следующее значение будет иметь ранг 8 (5+3);
  - Dense — ранг значения, идущего после ряда одинаковых значений, определяется как следующий ранг. Например, имеется 3 равных значения с рангом 5, следующее значение будет иметь ранг 6

[Подробная информация о функции RANKX и примеры формул >>>](#)

## DAX функция ERROR

ERROR () — останавливает выполнение DAX кода и выводит заранее определенную пользователем ошибку (предупреждение).

Синтаксис:

ERROR («Текст ошибки»)

[Подробная информация о функции ERROR и примеры формул >>>](#)



**СКОРО...!**

[Бесплатный экспресс-курс]

**Быстрый старт**

**в языке функций и формул DAX  
для Power BI и Power Pivot**



**Успейте записаться**