



Microsoft Power BI

Шевцов Василий Викторович,
директор ДИТ РУДН, shevtsov_vv@rudn.university

DAX

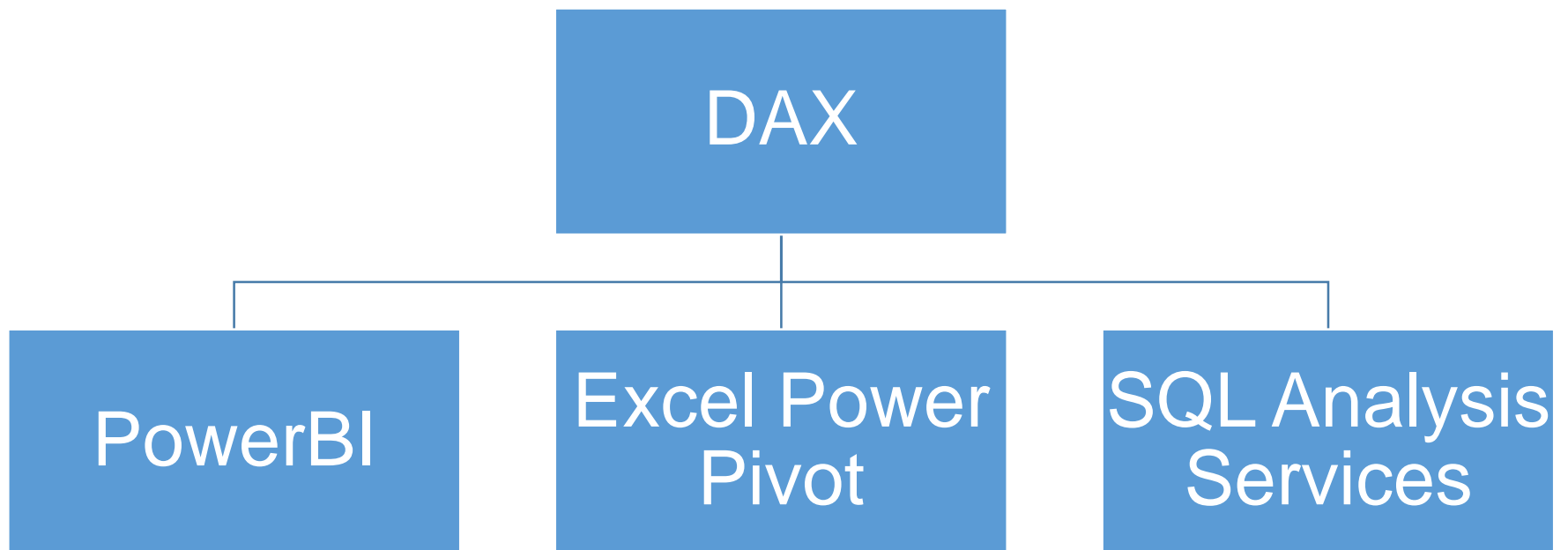
Общие сведения

Data Analysis eXpressions

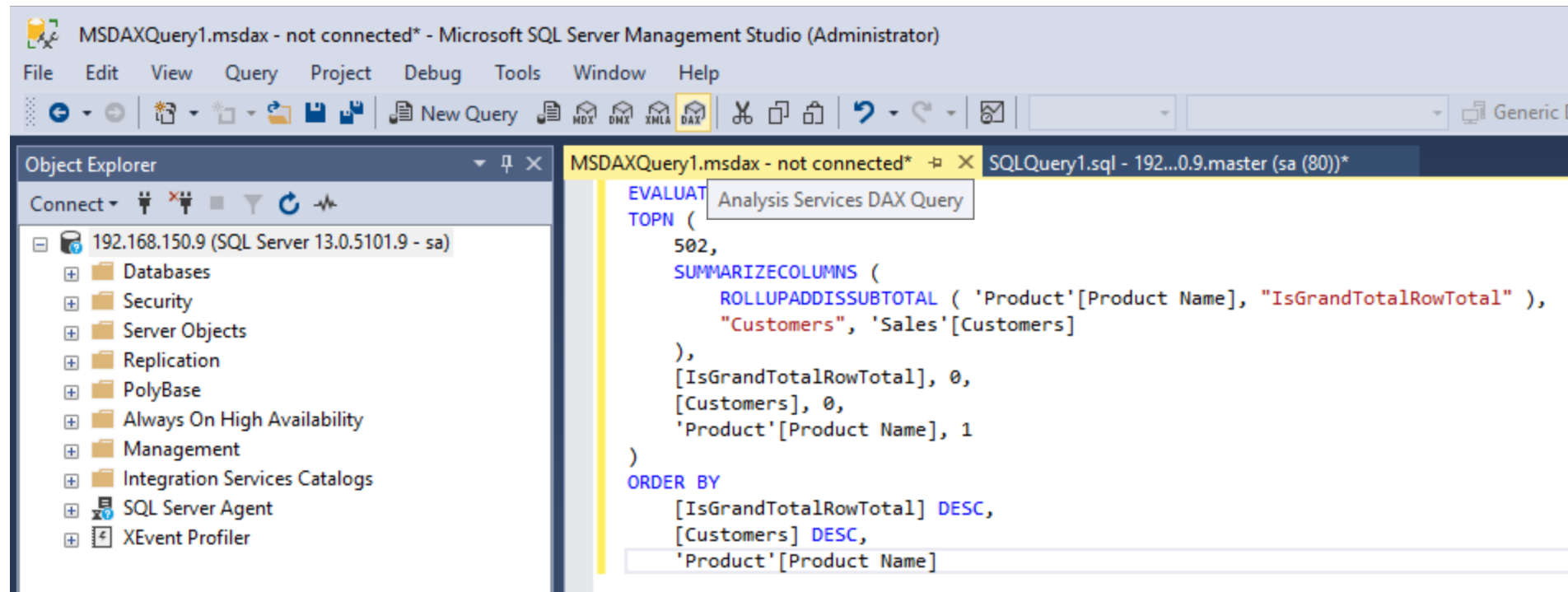
DAX. Общие сведения

- DAX предоставляет функции, схожие по задачам и именам с функциями Excel, с которыми пользователи уже знакомы. Однако эти функции были изменены для использования типов данных DAX и работы с таблицами и столбцами. Кроме того, DAX предоставляет множество специализированных функций для таких конкретных целей, как поиск по связям, итерации по таблице для рекурсивных вычислений и расчеты на основе логики операций со временем.





SQL Analysis Services

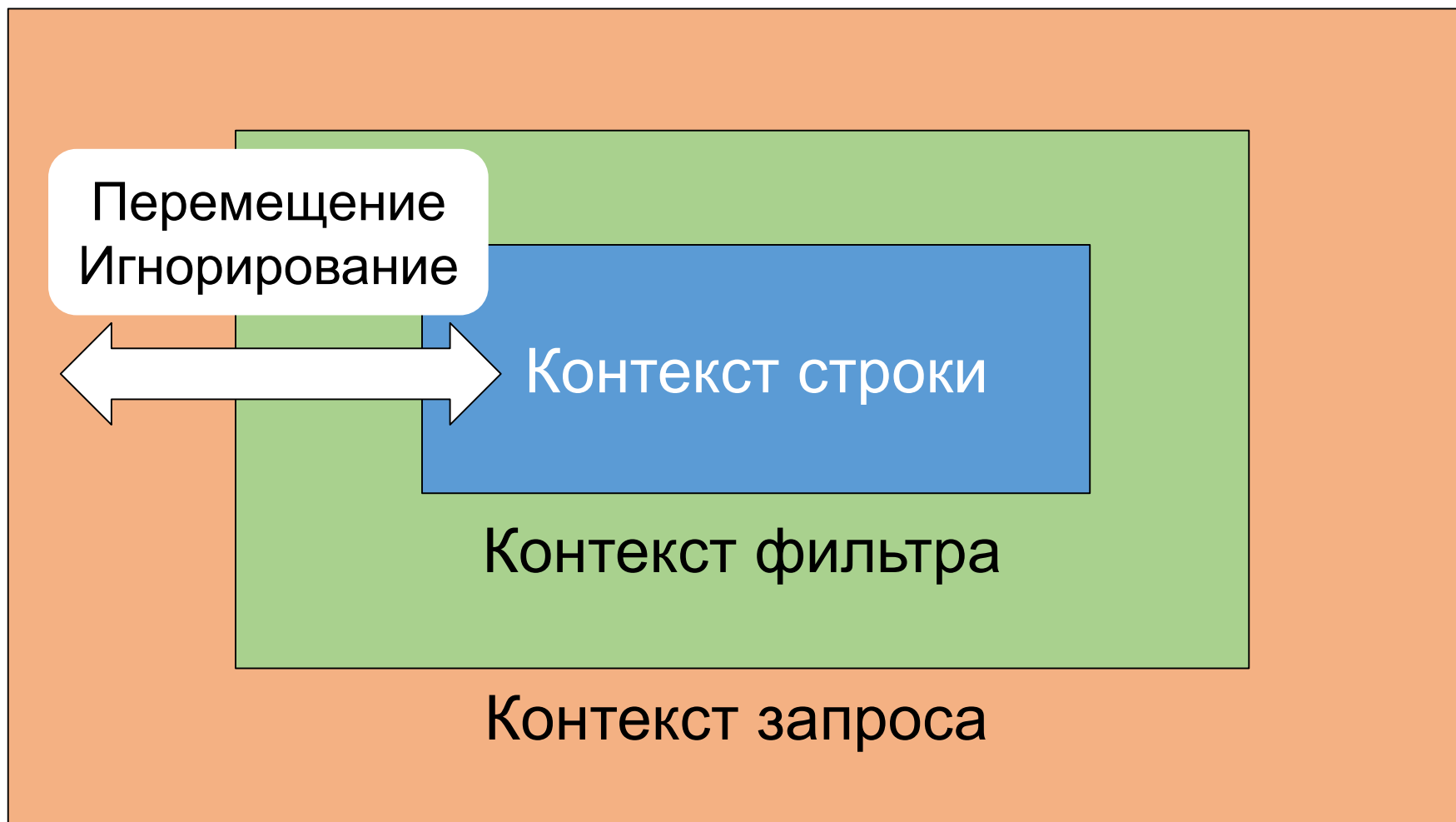


The screenshot displays the Microsoft SQL Server Management Studio (Administrator) interface. The title bar indicates the connection is 'MSDAXQuery1.msdx - not connected*'. The menu bar includes File, Edit, View, Query, Project, Debug, Tools, Window, and Help. The toolbar contains various icons for file operations and query execution. The Object Explorer on the left shows the server hierarchy for '192.168.150.9 (SQL Server 13.0.5101.9 - sa)', including Databases, Security, Server Objects, Replication, PolyBase, Always On High Availability, Management, Integration Services Catalogs, SQL Server Agent, and XEvent Profiler. The SQL Query window on the right shows a DAX query titled 'Analysis Services DAX Query'.

```
EVALUATE
TOPN (
    502,
    SUMMARIZECOLUMNS (
        ROLLUPADDISSUBTOTAL ( 'Product'[Product Name], "IsGrandTotalRowTotal" ),
        "Customers", 'Sales'[Customers]
    ),
    [IsGrandTotalRowTotal], 0,
    [Customers], 0,
    'Product'[Product Name], 1
)
ORDER BY
    [IsGrandTotalRowTotal] DESC,
    [Customers] DESC,
    'Product'[Product Name]
```

Контексты

Контексты



Контекст

Контекст фильтра - это набор значений, разрешенных в каждом столбце в зависимости от ограничений фильтра, которые применяются к строке или определенным критериям фильтра в формуле.

Контекст строки - фактически соответствует понятию текущей строки. Если создан вычисляемый столбец, то контекстом строки служат значение в каждой отдельной строке и значение в столбцах, связанных с текущей строкой.

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.

Внешний, или контекст запроса - включает в себя все фильтры, заданные явно или неявно используемыми столбцами (полями) и фильтрами/срезами сводной (полями визуального элемента Power BI), в общем, все фильтры, которые приходят в формулу "снаружи" (должны примениться к формуле и существуют/установлены ДО её вычисления).

Контекст строки

Контекст строки: работает в вычисляемых столбцах и функциях-итераторах (типа SUMX или FILTER)

В вычисляемых столбцах есть контекст строки. Контекст строки - это значения каждого столбца текущей строки. В этом контексте простое обращение к столбцу понимается как обращение к значению столбца в текущей строке (мы хотим получить конкретное скалярное значение). Однако если мы используем в вычисляемом столбце не просто формулу типа Таблица[Столбец1]+Таблица[Столбец2], а какую-то функцию, и эта функция на вход принимает не скалярное значение, а столбец, то и будет рассматриваться то, что запрошено: столбец таблицы целиком. Самый простой пример: функция SUM требует своим аргументом столбец, и при создании вычисляемого столбца с формулой =SUM(Таблица[Столбец]) она возьмет в рассмотрение весь столбец.

✕	✓	Столбец = SUM('Лист1'[2016])								
Округ	Область	2005	2010	2011	2012	2013	2014	2015	2016	Столбец
Центральный федеральный округ	Белгородская область	1512	1532	1536	1541	1544	1548	1550	1553	146804
Центральный федеральный округ	Брянская область	1327	1275	1264	1254	1242	1233	1226	1221	146804
Центральный федеральный округ	Владимирская область	1486	1441	1432	1422	1413	1406	1397	1390	146804
Центральный федеральный округ	Воронежская область	2361	2335	2332	2330	2329	2331	2333	2335	146804
Центральный федеральный округ	Ивановская область	1102	1060	1054	1049	1043	1037	1030	1023	146804
Центральный федеральный округ	Калужская область	1023	1009	1008	1006	1005	1011	1010	1014	146804
Центральный федеральный округ	Костромская область	700	666	662	659	656	654	651	648	146804

Контекст строки

Если мы используем в формуле вычисляемого столбца функцию CALCULATE, то она преобразует контекст строки в контекст фильтра - значение КАЖДОГО столбца текущей строки становится самостоятельным фильтром для данных этой таблицы. То есть формула =CALCULATE(SUM(Таблица[Столбец])) посчитает сумму столбца только в текущей строке (или, если несколько строк дублируются полностью, посчитает сумму столбца по всем дублирующимся строкам).

Использование меры в расчете вычисляемого столбца равнозначно использованию CALCULATE.

Эта история называется перенос контекста.

Аналогично работает контекст строки и перенос контекста в виртуальных таблицах (например, внутри функции SUMX мы перебираем строки виртуальной таблицы и для каждой строки делаем какие-то вычисления - почти как вычисляемый столбец в обычной таблице).

✕	✓	Столбец = CALCULATE(SUM('Лист1'[2016]))								
Округ	Область	2005	2010	2011	2012	2013	2014	2015	2016	Столбец
Центральный федеральный округ	Белгородская область	1512	1532	1536	1541	1544	1548	1550	1553	1553
Центральный федеральный округ	Брянская область	1327	1275	1264	1254	1242	1233	1226	1221	1221
Центральный федеральный округ	Владимирская область	1486	1441	1432	1422	1413	1406	1397	1390	1390
Центральный федеральный округ	Воронежская область	2361	2335	2332	2330	2329	2331	2333	2335	2335
Центральный федеральный округ	Ивановская область	1102	1060	1054	1049	1043	1037	1030	1023	1023
Центральный федеральный округ	Калужская область	1023	1009	1008	1006	1005	1011	1010	1014	1014
Центральный федеральный округ	Костромская область	700	666	662	659	656	654	651	648	648

Резюме

- Есть какой-то набор строк (набор данных).
- Формула читает (наружный по отношению к ней) набор внешних фильтров, которые создаются полями сводной таблицы или срезами (это внешний контекст). Если формула в вычисляемом столбце обычной таблицы, то набор внешних фильтров пустой - его нет, так как это же не сводная, а просто таблица.
- Далее она при необходимости модифицирует его (например, фильтрующими аргументами внутри CALCULATE). Либо меняет фильтр на каком-то столбце, либо убирает его совсем, либо наоборот, добавляет. Так формируется контекст фильтра. И если у нас есть при этом контекст строки, то он может быть (если мы сказали это делать, при помощи CALCULATE) преобразован в такой же контекст фильтра.
- В итоге весь получившийся набор фильтров - внешний, внутренний, перенесенный - миксуется и вычисляется, наконец, окончательное значение.

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.

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.

Ссылки

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.

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).

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:

- .,,:'\^|?&%\$!+=()[]{}<>

Другие ограничения

Синтаксис, необходимый для каждой функции, и тип операции, которую она может выполнять, сильно зависит 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

Тип оператора	Символ и назначение
Оператор в виде скобок	() (порядок приоритета и группирование аргументов)
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.

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.

Применение фильтров

Дан набор строк (набор данных).

Формула читает (наружный по отношению к ней) набор внешних фильтров, которые создаются полями сводной таблицы или срезами (это внешний контекст). Если формула в вычисляемом столбце обычной таблицы, то набор внешних фильтров пустой - его нет, так как это же не сводная, а просто таблица.

Далее она при необходимости модифицирует его (например, фильтрующими аргументами внутри CALCULATE). Либо меняет фильтр на каком-то столбце, либо убирает его совсем, либо наоборот, добавляет. Так формируется контекст фильтра. И если у нас есть при этом контекст строки, то он может быть (если мы сказали это делать, при помощи CALCULATE) преобразован в такой же контекст фильтра.

В итоге весь получившийся набор фильтров - внешний, внутренний, перенесенный - миксуется и вычисляется, наконец, окончательное значение.

Соглашения об именах параметров DAX

Параметр	Описание
expression	Любое выражение DAX, возвращающее одно скалярное значение, где выражение должно вычисляться несколько раз (для каждой строки/контекста).
value	Любое выражение DAX, возвращающее одно скалярное значение, где выражение должно быть вычислено ровно один раз перед всеми другими операциями.
table	Любое выражение DAX, возвращающее таблицу данных.
tableName	Имя существующей таблицы с использованием стандартного синтаксиса DAX. Это не может быть выражением.
columnName	Имя существующего столбца, использующего стандартный синтаксис DAX, обычно полное. Это не может быть выражением.
name	Строковая константа, которая будет использоваться для предоставления имени нового объекта.
order	Перечисление, используемое для определения порядка сортировки.
ties	Перечисление, используемое для определения обработки значений связи.
type	Перечисление, используемое для определения типа данных для PathItem и PathItemReverse.

Имена параметров префикса или только префикс

Парамтер	Описание
prefixing	<p>Имена параметров могут быть далее квалифицированы с префиксом, который описателен того, как используется аргумент, и чтобы избежать неоднозначного чтения параметров. Например:</p> <p>столбце result_columnname - ссылается на существующий столбец, используемый для получения значений результатов в LOOKUPVALUE() функция.</p> <p>Параметров search_columnname - относится к существующему столбцу используется для поиска значения в LOOKUPVALUE() функция.</p>
omitting	<p>Имена параметров будут опущены, если префикс достаточно ясен для описания параметра.</p> <p>Например, вместо того, чтобы иметь следующий синтаксис Дата (Year_Value, Month_Value, Day_Value) это понятнее для пользователя, чтобы читать даты (Год, месяц, день); повторяют три раза суффикс значение не добавит ничего к лучшему пониманию функции и захламлений чтение без надобности.</p> <p>Однако, если префикс параметра Year_columnName тогда имя параметра и префикс останется, чтобы убедиться, что пользователь понимает, что параметра должна быть ссылка на существующий столбец лет.</p>

Запросы DAX

Файл

Главная страница

Просмотр

Моделирование

Справка

Формат

Данные и детализация

Макет для телефона

Просмотр страницы ▼

☐ Показывать сетку
 ☐ Панель закладок
 ☒ Анализатор производительности

☐ Привязать объекты к сетке
 ☐ Область выделения

☐ Блокировать объекты
 ☐ Синхронизировать срезы

Отображать

2016 по Округ

2016 по Область

<

Анализатор производительности

✕

▶ Начать за...

🔄 Обновить визуальные элем...

⏸ Останов...

🧼 Очистить

📄 Экспортировать

Имя	Длительность ↓
🕒 Запись начата. (16.02.2020 20:12:45)	-
🗺 Перекрестное выделение	-
☐ 2016 по Область	297
Запрос DAX	29
Визуальное отображение	129
Другое	140
📄 Копировать запрос	
🗺 Перекрестное выделение	-
☐ 2016 по Область	309
Запрос DAX	31
Визуальное отображение	113
Другое	165
📄 Копировать запрос	

Запросы DAX

```
// DAX Query
DEFINE VAR __H0FilterTable =
    TREATAS({"Приволжский федеральный округ"}, 'Численность населения'[Округ])

EVALUATE
    TOPN(
        1001,
        SUMMARIZECOLUMNS(
            'Численность населения'[Область],
            "Sumv2016", CALCULATE(SUM('Численность населения'[2016])),
            "Численность_населенияSumv2016", IGNORE(
                CALCULATE(SUM('Численность населения'[2016]),
                KEEPFILTERS(__H0FilterTable))
            )
        ),
        [Sumv2016],
        0,
        'Численность населения'[Область],
        1
    )

ORDER BY
    [Sumv2016] DESC, 'Численность населения'[Область]
```

Функции с табличным значением

Синтаксис

[DEFINE { MEASURE <tableName>[<name>] = <expression> }

EVALUATE <table>

[ORDER BY {<expression> [{ASC | DESC}]}[, ...]

[START AT {<value>|<parameter>} [, ...]]]

Функции с табличным значением

Параметры

- Предложение DEFINE
Необязательное предложение инструкции запроса, которое позволяет определить меры на время выполнения запроса. Определения могут ссылаться на другие определения, которые появляются перед или после текущего определения.
- tableName
Имя существующей таблицы в стандартном синтаксисе DAX. Не может быть выражением.
- name
Имя новой меры. Не может быть выражением.
- expression
Любое выражение DAX, возвращающее одно скалярное значение.
- Предложение EVALUATE
Содержит табличное выражение, создающее результаты запроса. Выражение может использовать любую из определенных мер.
- Выражение должно возвращать таблицу. Если требуется скалярное значение, автор меры может поместить скалярное значение в функцию ROW(), чтобы сформировать таблицу, содержащую необходимое скалярное значение.

Функции с табличным значением

- Предложение ORDER BY
Необязательное предложение, определяющее выражения, по которым сортируются результаты запроса. Допустимо любое выражение, которое может быть вычислено для каждой из строк.
- Вложенное предложение START AT
Необязательное предложение внутри предложения ORDER BY, определяющее значения, с которых начнутся результаты запроса. Предложение START AT является частью предложения ORDER BY и не может использоваться за его пределами.
- В упорядоченном наборе результатов предложение START AT определяет начальную строку результирующего набора.
- Аргументы START AT имеют соответствие «один к одному» со столбцами в предложении ORDER BY. В предложении START AT может быть столько же аргументов, сколько и в предложении ORDER BY, но не более. Первый аргумент в предложении START AT определяет начальное значение в столбце 1 из столбцов ORDER BY. Второй аргумент в предложении START AT определяет начальное значение в столбце 2 из столбцов ORDER BY внутри строк, соответствующих первому значению для столбца 1.
- value
Значение константы. Не может быть выражением.

SQL → DAX

```
SELECT *  
FROM DimProduct  
WHERE Color = 'Red'
```

```
EVALUATE  
FILTER ( PRODUCT,  
PRODUCT[COLOR] = "RED" )
```

```
SELECT *  
FROM DimProduct  
WHERE Color = 'Red' AND ListPrice > 1000
```

```
EVALUATE  
FILTER (  
    Product,  
    AND ( Product[Color] = "Red", Product[ListPrice] >  
1000 )  
)
```


SQL → DAX

SQL-запрос, который возвращает красные продукты с общей суммой продаж более 100 000 и средней суммой продаж более 3000 в 2006 календарном году.

```
SELECT *
FROM DimProduct p
WHERE
    Color = 'Red'
    AND ( SELECT SUM([SalesAmount])
          FROM [FactInternetSales] s
          INNER JOIN DimDate d
            ON s.OrderDateKey = d.DateKey
          WHERE s.ProductKey = p.ProductKey
            AND d.CalendarYear = 2006
        ) > 100000
    AND ( SELECT AVG([SalesAmount])
          FROM [FactInternetSales] s
          INNER JOIN DimDate d
            ON s.OrderDateKey = d.DateKey
          WHERE s.ProductKey = p.ProductKey
            AND d.CalendarYear = 2006
        ) > 3000
```

```
EVALUATE
FILTER (
    FILTER ( Product, Product[Color] = "Red" ),
    AND (
        CALCULATE (
            SUM ( 'Internet Sales'[Sales Amount] ),
            'Date'[Calendar Year] = 2006
        ) > 100000,
        CALCULATE (
            AVERAGE ( 'Internet Sales'[Sales
Amount] ),
            'Date'[Calendar Year] = 2006
        ) > 3000
    )
)
```

Функции даты и времени

- Функции даты и времени в DAX схожи с функциями даты и времени Microsoft Excel. Однако функции DAX основаны на типах данных datetime, используемых в Microsoft SQL Server.

Функции даты и времени

CALENDAR	CALENDAR (<start_date>, <end_date>)	Возвращает таблицу с одним столбцом с именем "Date", который содержит непрерывный набор дат. Диапазон дат - от указанной даты начала до указанной даты окончания, включая эти две даты.
CALENDARAUTO	CALENDARAUTO ([fiscal_year_end_month])	Возвращает таблицу с одним столбцом с именем "Date", который содержит непрерывный набор дат. Диапазон дат рассчитывается автоматически на основе данных в модели.
UTCNOW	UTCNOW()	Возвращает текущую дату и время в формате UTC
UTCTODAY	UTCTODAY()	Возвращает текущую дату в формате UTC.

Функции даты и времени

DATE	DATE(<year>, <month>, <day>)	Возвращает указанную дату в формате datetime.
DATEDIFF	DATEDIFF(<start_date>, <end_date>, <interval>)	Возвращает количество границ интервалов, пересекаемых между двумя датами
DATEVALUE	DATEVALUE(date_text)	Преобразует дату в виде текста в дату в формате datetime.
DAY	DAY(<date>)	Возвращает день месяца в виде числа от 1 до 31.
EDATE	EDATE(<start_date>, <months>)	Возвращает дату, отстоящую от заданной даты на указанное число месяцев (до или после). С помощью функции EDATE вычисляются сроки обязательств и платежей, приходящиеся на тот же день месяца, что и начало расчетного периода.
EOMONTH	EOMONTH(<start_date>, <months>)	Возвращает в формате datetime дату окончания месяца (до или после указанного числа месяцев). С помощью функции EOMONTH вычисляются сроки обязательств и платежей, приходящиеся на конец месяца.
HOUR	HOUR(<datetime>)	Возвращает значение часа в виде числа от 0 (12:00 A.M.) до 23 (11:00 P.M.).
MINUTE	MINUTE(<datetime>)	Возвращает число минут от 0 до 59 по заданному значению даты и времени.
MONTH	MONTH(<datetime>)	Возвращает месяц в виде числа от 1 (январь) до 12 (декабрь).

Функции даты и времени

NOW	NOW()	Возвращает текущую дату и время в формате datetime. Функцию NOW полезно использовать, если нужно вывести на листе текущую дату и время или вычислить значение на основе текущей даты или времени, а затем обновлять это значение при каждом открытии листа.
SECOND	SECOND(<time>)	Возвращает для заданного значения времени число секунд в виде числа от 0 до 59.
TIME	TIME(hour, minute, second)	Преобразует часы, минуты и секунды, заданные в виде чисел, во время в формате datetime.
TIMEVALUE	TIMEVALUE(time_text)	Преобразует время в текстовом формате в формат datetime.
TODAY	TODAY()	Возвращает текущую дату.
WEEKDAY	WEEKDAY(<date>, <return_type>)	Возвращает число от 1 до 7, определяющее день недели для даты. По умолчанию дни недели считаются от воскресенья (1) до субботы (7).
WEEKNUM	WEEKNUM(<date>, <return_type>)	Возвращает номер недели для заданных даты и года в соответствии со значением тип_возврата. Номер недели указывает числовую позицию недели относительно начала года.
YEAR	YEAR(<date>)	Возвращает для заданной даты год в виде четырехзначного числа от 1900 до 9999.
YEARFRAC	YEARFRAC(<start_date>, <end_date>, <basis>)	Вычисляет долю года, представленную числом целых дней между двумя датами. Функция листа YEARFRAC позволяет определить долю преимуществ или обязательств за целый год, приходящуюся на указанный срок.

Функции логики 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 вместе со статистическими выражениями или вычислениями, можно создавать осмысленные сравнения по сравнимым периодам времени для продаж, количеству товара и так далее.

Функции логики операций со временем

CLOSINGBALANCEMONTH	CLOSINGBALANCEMONTH (<expression>,<dates> [,<filter>])	Вычисляет выражение expression для последней даты месяца в текущем контексте.
CLOSINGBALANCEQUARTER	CLOSINGBALANCEQUARTER (<expression>,<dates> [,<filter>])	Вычисляет выражение для последней даты квартала в текущем контексте.
CLOSINGBALANCEYEAR	CLOSINGBALANCEYEAR (<expression>,<dates> [,<filter>] [,<year_end_date>])	Вычисляет expression для последней даты года в текущем контексте.
DATEADD	DATEADD (<dates>,<number_of_intervals>,<interval>)	Возвращает таблицу, содержащую столбец дат, сдвинутых в будущее или в прошлое от дат в текущем контексте на указанное количество интервалов.
DATESBETWEEN	DATESBETWEEN (<dates>,<start_date>,<end_date>)	Возвращает таблицу, содержащую столбец с датами, начиная со значения начальная_дата и заканчивая значением конечная_дата.
DATESINPERIOD	DATESINPERIOD (<dates>,<start_date>,<number_of_intervals>,<interval>)	Возвращает таблицу, содержащую столбец дат, который начинается с start_date и продолжается в течение указанного number_of_intervals.
DATESMTD	DATESMTD (<dates>)	Возвращает таблицу, содержащую столбец дат за месяц до текущей даты в текущем контексте.
DATESQTD	DATESQTD (<dates>)	Возвращает таблицу, содержащую столбец дат за текущий квартал в текущем контексте.
DATESYTD	DATESYTD (<dates> [,<year_end_date>])	Возвращает таблицу, содержащую столбец дат за текущий год в текущем контексте.

Функции логики операций со временем

ENDOFMONTH	ENDOFMONTH (<dates>)	Возвращает последний день месяца в текущем контексте для указанного столбца дат.
ENDOFQUARTER	ENDOFQUARTER (<dates>)	Возвращает последний день квартала в текущем контексте для указанного столбца дат.
ENDOFYEAR	ENDOFYEAR (<dates> [,<year_end_date>])	Возвращает последний день года в текущем контексте для указанного столбца дат.
FIRSTDATE	FIRSTDATE (<dates>)	Возвращает первую дату в текущем контексте для указанного столбца дат.
FIRSTNONBLANK	FIRSTNONBLANK (<column>,<expression>)	Возвращает первое значение в столбце column (с фильтрацией по текущему контексту), где выражение не является пустым.
LASTDATE	LASTDATE (<dates>)	Возвращает последнюю дату в текущем контексте для указанного столбца дат.
LASTNONBLANK	LASTNONBLANK (<column>,<expression>)	Возвращает последнее значение в столбце column (с фильтрацией по текущему контексту), где выражение не является пустым.
NEXTDAY	NEXTDAY (<dates>)	Возвращает таблицу, содержащую столбец со всеми датами следующего дня, на основе первой даты, указанной в столбце dates в текущем контексте.
NEXTMONTH	NEXTMONTH (<dates>)	Возвращает таблицу, содержащую столбец со всеми датами следующего месяца на основе первой даты в столбце dates в текущем контексте.

Функции логики операций со временем

NEXTQUARTER	NEXTQUARTER (<dates>)	Возвращает таблицу, содержащую столбец со всеми датами следующего квартала, на основе первой даты, указанной в столбце dates в текущем контексте.
NEXTYEAR	NEXTYEAR (<dates> [,<year_end_date>])	Возвращает таблицу, содержащую столбец со всеми датами следующего года, на основе первой даты в столбце dates в текущем контексте.
OPENINGBALANCEMONTH	OPENINGBALANCEMONTH (<expression>,<dates> [,<filter>])	Вычисляет выражение в первый день месяца в текущем контексте.
OPENINGBALANCEQUARTER	OPENINGBALANCEQUARTER (<expression>,<dates> [,<filter>])	Вычисляет выражение для первой даты квартала в текущем контексте.
OPENINGBALANCEYEAR	OPENINGBALANCEYEAR (<expression>,<dates> [,<filter>] [,<year_end_date>])	Вычисляет expression для первой даты года в текущем контексте.
PARALLELPERIOD	PARALLELPERIOD (<dates>,<number_of_intervals>,<interval>)	Возвращает таблицу, содержащую столбец с датами, представляющими период, параллельный датам в указанном столбце dates, в текущем контексте; даты смещены на несколько интервалов вперед или назад во времени.
PREVIOUSDAY	PREVIOUSDAY(<dates>)	Возвращает таблицу, содержащую столбец со всеми датами предыдущего дня на основе первой даты в столбце dates в текущем контексте.

Функции логики операций со временем

PREVIOUSMONTH	PREVIOUSMONTH (<dates>)	Возвращает таблицу, содержащую столбец со всеми датами предыдущего месяца на основе первой даты в столбце dates в текущем контексте.
PREVIOUSQUARTER	PREVIOUSQUARTER (<dates>)	Возвращает таблицу, содержащую столбец со всеми датами предыдущего квартала, на основе первой даты в столбце dates в текущем контексте.
PREVIOUSYEAR	PREVIOUSYEAR (<dates> [,<year_end_date>])	Возвращает таблицу, содержащую столбец со всеми датами предыдущего года, на основе последней даты в столбце dates в текущем контексте.
SAMEPERIODLASTYEAR	SAMEPERIODLASTYEAR (<dates>)	Возвращает таблицу, содержащую столбец с датами, сдвинутыми на один год назад от дат, указанных в столбце dates в текущем контексте.
STARTOFMONTH	STARTOFMONTH (<dates>)	Возвращает первый день месяца в текущем контексте для указанного столбца дат.
STARTOFQUARTER	STARTOFQUARTER (<dates>)	Возвращает первый день квартала в текущем контексте для указанного столбца дат.
STARTOFYEAR	STARTOFYEAR (<dates>)	Возвращает первый день года в текущем контексте для указанного столбца дат.
TOTALMTD	TOTALMTD (<expression>,<dates> [,<filter>])	Вычисляет значение выражения для интервала с первого числа месяца до текущей даты в текущем контексте.

Функции логики операций со временем

TOTALQTD	TOTALQTD (<expression>,<dates> [,<filter>])	Вычисляет в текущем контексте значение выражения для дат с начала последнего квартала до текущей даты.
TOTALYTD	TOTALYTD (<expression>,<dates> [,<filter>] [,<year_end_date>])	Вычисляет значение выражения expression за текущий год в текущем контексте.

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.

Информационные функции

CONTAINS (DAX)	CONTAINS(<table>, <columnName>, <value>[, <columnName>, <value>]...)	возвращает значение true, если значения для всех столбцов, на которые имеются ссылки, существуют или содержатся в этих столбцах; в противном случае возвращает значение false.
CUSTOMDATA	CUSTOMDATA()	Возвращает содержимое пользовательских данных свойств в строке подключения.
IN Operator / CONTAINSROW Function	<scalarExpr> IN <tableExpr> (<scalarExpr1>, <scalarExpr2>, ...) IN <tableExpr>	Возвращает TRUE, если строка значений существует или содержится в таблице, в противном случае возвращает FALSE. Кроме синтаксиса, оператор IN и функция CONTAINSROW функционально эквивалентны.
ISBLANK	ISBLANK(<value>)	Проверяет, пустое ли значение, и возвращает значение TRUE или FALSE.
ISERROR	ISERROR(<value>)	Проверяет, является ли значение ошибкой, и возвращает TRUE или FALSE.
ISEVEN	ISEVEN(number)	Возвращает TRUE, если number четный, или FALSE, если number нечетный.
ISLOGICAL	ISLOGICAL(<value>)	Проверяет, является ли значение логическим значением (TRUE или FALSE), и возвращает TRUE или FALSE.
ISNONTEXT	ISNONTEXT(<value>)	Проверяет, не является ли значение текстом (пустые ячейки не являются текстом), и возвращает TRUE или FALSE.
ISNUMBER	ISNUMBER(<value>)	Проверяет, является ли значение числом, и возвращает TRUE или FALSE.
ISODD	ISODD(number)	Возвращает TRUE, если number нечетный, или FALSE, если number четный.

Информационные функции

ISONORAFTER	ISONORAFTER(<scalar_expression>, <scalar_expression>sort_order) [,scalar_expression>, <scalar_expression>, [sort_order][,...])	Булева функция, эмулирующая поведение предложения Start At и возвращающая значение true для строки, удовлетворяющей всем параметрам условия. Эта функция принимает переменное число троек, первые два значения в тройке являются выражениями для сравнения, а третий параметр указывает порядок сортировки. Порядок сортировки может быть по возрастанию (по умолчанию) или по убыванию. По порядку сортировки первый параметр сравнивается со вторым. Если порядок сортировки является возрастающим, то сравнение будет выполнено с первым параметром, большим или равным второму параметру. Если порядок сортировки нисходящий, сравнение выполняется по второму параметру, который меньше или равен первому параметру.
ISTEXT	ISTEXT(<value>)	Проверяет, является ли значение текстом, и возвращает TRUE или FALSE.
LOOKUPVALUE (DAX)	LOOKUPVALUE(<result_columnName>, <search_columnName>, <search_value>[, <search_columnName>, <search_value>]...)	Возвращает значение в столбце result_columnName для строки, соответствующей всем критериям, указанным с помощью параметров search_columnName и search_value.
USERNAME	USERNAME()	Возвращает имя домена и имя пользователя из учетных данных, предоставленных системе во время подключения

Логические функции

- Логические функции обрабатывают выражение и возвращают сведения о значениях в выражении. Например, функция TRUE в языке DAX сообщает, возвращает ли вычисляемое выражение значение TRUE.

Логические функции

AND	AND(<logical1>,<logical2>)	Проверяет значения аргументов и возвращает значение TRUE, если оба аргумента имеют значение TRUE. В противном случае возвращает значение FALSE.
FALSE	FALSE()	Возвращает логическое значение FALSE.
IF	IF(logical_test>,<value_if_true>, value_if_false)	Проверяет выполнение условия, указанного в первом аргументе. Возвращает одно значение, если условие равно TRUE, и другое значение, если условие равно FALSE.
IFERROR	IFERROR(value, value_if_error)	Вычисляет выражение и возвращает указанное значение, если выражение ошибочно. В противном случае возвращает значение выражения.
NOT	NOT(<logical>)	Изменяет значение FALSE на TRUE, а TRUE на FALSE.
OR	OR(<logical1>,<logical2>)	Проверяет, является ли один из аргументов TRUE, и если да, то возвращает значение TRUE. Функция возвращает значение FALSE, если оба аргумента имеют значение FALSE.
SWITCH (DAX)	SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])	Вычисляет выражение на основе списка значений и возвращает одно из нескольких возможных результирующих выражений.
TRUE	TRUE()	Возвращает логическое значение TRUE.

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.

Математические и тригонометрические функции

ABS	ABS(<number>)	Возвращает абсолютное значение числа.
ACOS	ACOS(number)	Возвращает арккосинус, или обратный косинус, числа. Арккосинус-это угол, косинус которого равен числу. Возвращаемый угол задается в радианах в диапазоне от 0 (ноль) до π .
ACOSH	ACOSH(number)	Возвращает обратный гиперболический косинус числа. Число должно быть больше или равно 1. Гиперболический арккосинус-это значение, гиперболический косинус которого равен числу, так что $\cosh(\text{acosh}(\text{дубинка}(\text{число})))$ равняется числу.
ASIN	ASIN(number)	Возвращает арксинус, или обратный синус, числа. Арксинус-это угол, синус которого равен числу. Возвращаемый угол задается в радианах в диапазоне от $-\pi/2$ до $\pi/2$.
ASINH	ASINH(number)	Возвращает обратный гиперболический синус числа. Обратный гиперболический синус-это значение, гиперболический синус которого равен числу, так что $\sinh(\text{asinh}(\text{число}))$ равняется числу.
ATAN	ATAN(number)	Возвращает арктангенс или обратную касательную числа. Арктангенс-это угол, тангенс которого равен числу. Возвращаемый угол задается в радианах в диапазоне от $-\pi/2$ до $\pi/2$.

Математические и тригонометрические функции

ATANH	ATANH(number)	Возвращает обратный гиперболический тангенс числа. Число должно быть от -1 до 1 (за исключением -1 и 1). Обратный гиперболический тангенс является гиперболический тангенс является количество, поэтому инструмента $\text{atanh}(\text{TANH}(\text{число}))$ равняется числу.
CEILING	CEILING(<number>, <significance>)	Округляет число в большую сторону до ближайшего целого или до ближайшего числа, кратного заданной значимости.
COMBIN	COMBIN(number, number_chosen)	Возвращает количество комбинаций для заданного количества элементов. Используйте комбинацию для определения общего возможного числа групп для заданного количества элементов.
COMBINA	COMBINA (number, number_chosen)	Возвращает количество комбинаций (с повторениями) для заданного количества элементов.
COS	COS(number)	Возвращает косинус заданного угла.
COSH	COSH(number)	Возвращает гиперболический косинус числа.
CURRENCY (DAX)	CURRENCY(<value>)	Вычисляет аргумент и возвращает результат как тип данных currency.
DIVIDE	DIVIDE(<numerator>, <denominator> [, <alternateresult>])	Выполняет деление и возвращает альтернативный результат или BLANK () при делении на 0.

Математические и тригонометрические функции

EVEN	EVEN(number)	Возвращает число, округленное до ближайшего четного целого числа.
EXP	EXP(<number>)	Возвращает число e , возведенное в степень, заданную числом. Константа e приблизительно равна 2,71828182845904 и служит основанием натуральных логарифмов.
FACT	FACT(<number>)	Возвращает факториал числа (произведение последовательности $1*2*3*...*$, завершающейся заданным числом).
FLOOR	FLOOR(<number>, <significance>)	Округляет число в меньшую сторону (к нулю) до ближайшего числа, кратного заданной значимости.
GCD	GCD(number1, [number2], ...)	Возвращает наибольший общий делитель из двух или более целых чисел. Наибольшим общим делителем является наибольшее целое число, которое делит и number1 и number2 без остатка.
INT	INT(<number>)	Округляет число в меньшую сторону до ближайшего целого.
ISO.CEILING	ISO.CEILING(<number>[, <significance>])	Округляет число в большую сторону до ближайшего целого или до ближайшего числа, кратного заданной значимости.

Математические и тригонометрические функции

LCM	LCM(number1, [number2], ...)	Возвращает наименьшее число, кратное целым числам. Наименьшим общим числом является наименьшее положительное целое число, кратное всем целочисленным аргументам number1, number2 и т. д. Используйте LCM для добавления дробей с различными знаменателями.
LN	LN(<number>)	Возвращает натуральный логарифм числа. Натуральные логарифмы вычисляются по константе e (2,71828182845904).
LOG	LOG(<number>,<base>)	Возвращает логарифм числа по указанному основанию.
LOG10	LOG10(<number>)	Возвращает логарифм числа по основанию 10.
INT	INT(<number>)	Округляет число в меньшую сторону до ближайшего целого.
MROUND	MROUND(<number>, <multiple>)	Возвращает число, округленное до кратного заданной значимости.
ODD	ODD(number)	Возвращает число, округленное до ближайшего нечетного целого.
PI	PI()	Возвращает значение константы «пи», равное 3,14159265358979 с точностью до 15 знака.
POWER	POWER(<number>, <power>)	Возвращает результат возведения числа в степень.
PRODUCT	PRODUCT(<column>)	Возвращает произведение чисел в столбце.

Математические и тригонометрические функции

PRODUCTX	PRODUCTX(<table>, <expression>)	Возвращает произведение выражения, вычисленное для каждой строки в таблице.
QUOTIENT	QUOTIENT(<numerator>, <denominator>)	Выполняет деление и возвращает только целочисленную часть результата. Эта функция используется, если не нужно учитывать остаток от деления.
RADIANS	RADIANS(angle)	Преобразует градусы в радианы.
RAND	RAND()	Возвращает случайное число, большее или равное 0 и меньшее 1, с равномерным распределением. Возвращаемое число изменяется каждый раз при пересчете ячейки, содержащей данную функцию.
RANDBETWEEN	RANDBETWEEN(<bottom>,<top>)	Возвращает случайное число, расположенное в интервале между двумя заданными числами.
ROUND	ROUND(<number>, <num_digits>)	Округляет число до указанного количества разрядов.
ROUNDDOWN	ROUNDDOWN(<number>, <num_digits>)	Округляет число вниз, к нулю.
ROUNDUP	ROUNDUP(<number>, <num_digits>)	Округляет число в большую сторону, от 0 (нуля).
SIGN	SIGN(<number>)	Определяет знак числа, результата вычисления или значения в столбце. Функция возвращает значение 1, если число положительное, значение 0 (ноль), если число равно нулю, и значение -1, если число отрицательное.

Математические и тригонометрические функции

SQRT	SQRT(<number>)	Возвращает квадратный корень числа.
SUM	SUM(<column>)	Складывает все числа в столбце.
SUMX	SUMX(<table>, <expression>)	Возвращает сумму результатов выражения, вычисляемого для каждой строки в таблице.
TRUNC	TRUNC(<number>,<num_digits>)	Усекает число до целого, удаляя дробную часть.

Текстовые функции

- Текстовые функции DAX во многом схожи со своими аналогами 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.

Текстовые функции

BLANK	BLANK()	Возвращает пустое значение.
CODE	CODE(text)	Возвращает числовой код для первого символа в текстовой строке. Возвращаемый код соответствует набору символов, используемому компьютером.
CONCATENATE	CONCATENATE(<text1>, <text2>)	Соединяет две текстовые строки в одну текстовую строку.
CONCATENATEX	CONCATENATEX (таблица, выражение, разделитель, выражение_упорядочивания1, порядок1,...)	Вычисляет выражение для каждой строки таблицы, затем возвращает полученные значения разделенные разделителем.
EXACT	EXACT(<text1>,<text2>)	Сравнивает две текстовые строки и возвращает TRUE, если они полностью одинаковые, и FALSE в противном случае. Функция EXACT учитывает регистр, но не учитывает отличия форматирования. Можно использовать функцию EXACT, чтобы проверить входящий в документ текст.
FIND	FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])	Возвращает начальную позицию одной текстовой строки внутри другой текстовой строки. Функция FIND учитывает регистр.
FIXED	FIXED(<number>, <decimals>, <no_commas>)	Округляет число до указанного количества десятичных знаков и возвращает результат в виде текста. Можно задать, чтобы результат возвращался с пробелами, делящими число на группы по три цифры, или без них.
FORMAT	FORMAT(<value>, <format_string>)	Преобразует значение в текст в указанном числовом формате.
LEFT	LEFT(<text>, <num_chars>)	Возвращает указанное количество символов с начала текстовой строки.
LEN	LEN(<text>)	Возвращает число символов в текстовой строке.
LOWER	LOWER(<text>)	Преобразует все буквы в текстовой строке в нижний регистр.
MID	MID(<text>, <start_num>, <num_chars>)	Возвращает строку символов из середины текстовой строки с учетом начальной позиции и длины.

Текстовые функции

REPLACE	REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)	Функция REPLACE на основе указанного числа символов заменяет часть текстовой строки другой текстовой строкой.
REPT	REPT(<text>, <num_times>)	Повторяет текст заданное количество раз. Используйте функцию REPT для заполнения ячейки определенным числом экземпляров текстовой строки.
RIGHT	RIGHT(<text>, <num_chars>)	Функция RIGHT возвращает последний символ или последние символы текстовой строки, в зависимости от указанного количества символов.
SEARCH	SEARCH(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])	Возвращает номер позиции, начиная с которой обнаружен указанный символ или текстовая строка, если считать слева направо. При поиске не учитывается регистр и учитываются диакритические знаки.
SUBSTITUTE	SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)	Заменяет существующий текст новым в текстовой строке.
TRIM	TRIM(<text>)	Удаляет все пробелы из текста, за исключением единичных пробелов между словами.
UPPER	UPPER (<text>)	Преобразует все буквы в текстовой строке в верхний регистр.
VALUE	Преобразует текстовую строку, представляющую число, в число.	VALUE(<text>)

Предопределенные числовые форматы для функции FORMAT

Спецификация формата	Описание
"General Number"	Отображает число без разделителя групп разрядов.
"Currency"	Отображает число с разделителями групп разрядов, если они необходимы; 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 разделителем групп разрядов, не менее одной цифры слева и две цифры справа от десятичного разделителя.
"Percent"	Отображает умноженное на 100 число со знаком процента (%) непосредственно справа, всегда отображает две цифры справа от десятичного разделителя.
"Scientific"	Использует стандартное научное обозначение с двумя значащими разрядами.
"Yes/No"	Отображает «Нет», если число равно 0; в противном случае — «Да».
"True/False"	Отображает «False», если число равно 0; в противном случае — «True».
"On/Off"	Отображает «Выкл», если число равно 0; в противном случае — «Вкл».

Пользовательские числовые форматы для функции FORMAT

Спецификация формата	Описание
Нет	Выводит число без форматирования.
0 (ноль)	<p>Заполнитель цифры. Выводит заданную цифру или ноль. Если выражение содержит цифру в позиции, где в строке форматирования находится ноль, то выводится эта цифра. В противном случае в этой позиции выводится ноль.</p> <p>Если число имеет меньше разрядов, чем задано нулей в выражении форматирования (с обеих сторон от десятичного разделителя), то число выводится с начальными или с конечными нулями. Если число имеет больше разрядов справа от десятичного разделителя, чем задано нулей справа от десятичного разделителя в выражении форматирования, то значение округляется до стольких знаков после запятой, сколько задано нулей. Если число имеет больше цифр слева от десятичного разделителя, чем задано нулей слева от десятичного разделителя в выражении форматирования, то дополнительные разряды выводятся без изменений.</p>
#	<p>Заполнитель цифры. Выводит цифру или не выводит никакого символа. Если выражение содержит цифру в позиции, где в строке форматирования находится знак #, то выводится эта цифра. В противном случае в этой позиции не выводится ничего.</p> <p>Этот символ аналогичен цифровому заполнителю «0» с той разницей, что начальные и конечные нули не отображаются в случае, если число содержит меньше разрядов, чем задано символов «#» с обеих сторон от десятичного разделителя в выражении форматирования.</p>

Пользовательские числовые форматы для функции FORMAT

Спецификация формата	Описание
. (точка)	Заполнитель десятичного разделителя. Он определяет, сколько разрядов отображается слева и справа от десятичного разделителя. Если выражение форматирования содержит только знаки # слева от этого символа, то числа меньше 1 начинаются с десятичного разделителя. Чтобы в этом случае в дробных числах отображались ведущие нули, укажите ноль в качестве первого заполнителя цифры слева от десятичного разделителя. В некоторых локалях и региональных стандартах в качестве десятичного разделителя используется запятая. Фактический символ, используемый в качестве десятичного заполнителя в отформатированном значении, зависит от числового формата, распознаваемого системой. Поэтому даже в локали, где десятичным разделителем служит запятая, в выражениях форматирования в качестве десятичного заполнителя следует использовать точку. Отформатированная строка будет выводиться в соответствии с локалью.
%	Заполнитель процента. Умножает выражение на 100. Символ процента (%) вставляется там, где он должен выводиться в строке форматирования.

Пользовательские числовые форматы для функции FORMAT

Спецификация формата	Описание
, (запятая)	<p>Разделитель групп разрядов. Разделитель групп разрядов отделяет тысячи от сотен в числе с четырьмя или более разрядами слева от десятичного разделителя. При стандартном использовании разделителя групп разрядов он заключается в заполнители цифр (0 или #).</p> <p>Если разделитель групп разрядов расположен непосредственно слева от десятичного разделителя (независимо от того, указаны ли десятичные знаки) или в конце строки, это равносильно делению числа на 1 000 с последующим округлением. Для чисел, больших или равных 500 и меньших 1 000, выводится 1, а для чисел, меньших 500, выводится 0. Если в этой позиции рядом находятся два разделителя групп разрядов, то число делится на 1 миллион и далее на 1 000 за каждый дополнительный разделитель.</p> <p>Несколько разделителей групп разрядов в любой позиции, кроме позиции непосредственно слева от десятичного разделителя или конечной позиции в строке, интерпретируются просто как разделители групп разрядов. В некоторых локалях в качестве разделителя групп разрядов используется точка. Фактический символ, используемый в качестве разделителя групп разрядов в отформатированном значении, зависит от числового формата, распознаваемого системой. Поэтому даже в локалях, где разделителем групп разрядов служит точка, в выражениях форматирования следует использовать запятую. Отформатированная строка будет выводиться в соответствии с локалью.</p> <p>Рассмотрим три строки форматирования:</p> <p>«#,0.», где с помощью разделителя групп разрядов число «сто миллионов» форматируется в виде строки «100,000,000»;</p> <p>«#0,», где путем деления на тысячу число «сто миллионов» представляется в виде строки «100000»;</p> <p>«#,0,», где с помощью разделителя групп разрядов и деления на тысячу число «сто миллионов» форматируется в виде строки «100,000».</p>

Пользовательские числовые форматы для функции FORMAT

Спецификация формата	Описание
: (двоеточие)	Разделитель компонентов времени. В некоторых локалях в качестве разделителя компонентов времени могут использоваться другие символы. Разделитель компонентов времени разделяет часы, минуты и секунды при форматировании значений времени. Фактический символ, используемый в качестве разделителя компонентов времени в отформатированном значении, зависит от параметров системы.
/ (косая черта)	Разделитель компонентов даты. В некоторых локалях и региональных стандартах в качестве разделителя компонентов даты могут использоваться другие символы. Разделитель компонентов даты разделяет день, месяц и год при форматировании значений даты. Фактический символ, используемый в качестве разделителя компонентов даты в отформатированном значении, зависит от параметров системы.
E-, E+, e-, e+	Экспоненциальный формат. Если выражение форматирования содержит минимум один заполнитель цифры (0 или #) слева от символов E-, E+, e- или e+, то число выводится в экспоненциальном формате, а между числом и показателем степени вставляется символ «E» или «e». Количество заполнителей цифр слева определяет количество разрядов в показателе степени. Чтобы указывать знак «минус» для отрицательных показателей степени, используются заполнители E- и e-. Заполнители E+ и e+ позволяют указывать знак «минус» для отрицательных показателей степени и знак «плюс» для положительных показателей степени. Для правильного форматирования также необходимо указывать заполнители цифр справа от этого символа.

Пользовательские числовые форматы для функции FORMAT

Спецификация формата	Описание
- + \$ ()	Литералы. Эти символы выводятся так же, как в строке форматирования. Чтобы вывести символ, которого нет в списке, укажите перед ним обратную косую черту (\) либо заключите его в двойные кавычки (").
\ (обратная косая черта)	Следующий символ в строке формата. Чтобы вывести специальный символ в качестве литерала, укажите перед ним обратную косую черту (\). Сама обратная косая черта не отображается. Использование обратной косой черты аналогично заключению выводимого символа в двойные кавычки. Чтобы вывести саму обратную косую черту, укажите две черты подряд (\\). Примерами символов, которые не выводятся в виде литералов, являются символы форматирования даты и времени (a, c, d, h, m, n, p, q, s, t, w, y, / и :), символы числового форматирования (#, 0, %, E, e, запятая и точка) и символы форматирования строк (@, &, <, > и !).
"ABC"	Выводит строку, заключенную в двойные кавычки ("). Чтобы включить строку в аргумент стиля из программного кода, необходимо заключить текст в символы Chr(34) (код 34 соответствует знаку кавычки (")).

Пользовательские числовые форматы для функции FORMAT

Аргумент стиля для функции Format	Отформатированное представление для числа 5	Отформатированное представление для числа -5	Отформатированное представление для числа «0,5»	Отформатированное представление для числа «0»
Строка нулевой длины ("")	5	-5	0.5	0
0	5	-5	1	0
0.00	5.00	-5.00	0.50	0.00
#,##0	5	-5	1	0
\$#,##0;(\$#,##0)	\$5	(\$5)	\$1	\$0
\$#,##0.00;(\$#,##0.00)	\$5.00	(\$5.00)	\$0.50	\$0.00
0%	500%	-500%	50%	0%
0.00%	500.00%	-500.00%	50.00%	0.00%
0.00E+00	5.00E+00	-5.00E+00	5.00E-01	0.00E+00
0.00E-00	5.00E00	-5.00E00	5.00E-01	0.00E00
"\$#,##0;;\Z\е\r\o"	\$5	\$-5	\$1	Zero

Стандартные форматы даты и времени для функции FORMAT

Спецификация формата	Описание
"General Date"	Отображает дату и/или время. Например, 3/12/2008 11:07:31 AM. Отображение даты определяется текущим значением культуры в приложении.
"Long Date" или "Medium Date"	Отображает дату в соответствии с полным форматом даты в текущей культуре. Например, Wednesday, March 12, 2008.
"Short Date"	Отображает дату в соответствии с кратким форматом даты в текущей культуре. Например, 3/12/2008.
"Long Time" или "Medium Time"	Выводит время в соответствии с полным форматом времени в текущей культуре. Обычно указываются часы, минуты и секунды. Например, 11:07:31 AM.
"Short Time"	Отображает время в соответствии с кратким форматом времени в текущей культуре. Например, 11:07 AM.

Пользовательские форматы даты и времени для функции FORMAT

Спецификация формата	Описание
(:)	Разделитель компонентов времени. В некоторых локалях в качестве разделителя компонентов времени могут использоваться другие символы. Разделитель компонентов времени разделяет часы, минуты и секунды при форматировании значений времени. Символ, фактически используемый в качестве разделителя компонентов времени в отформатированном значении, зависит от текущего значения культуры в приложении.
(/)	Разделитель компонентов даты. В некоторых локалях и региональных стандартах в качестве разделителя компонентов даты могут использоваться другие символы. Разделитель компонентов даты разделяет день, месяц и год при форматировании значений даты. Символ, фактически используемый в качестве разделителя компонентов даты в отформатированном значении, зависит от текущего значения культуры в приложении.
(%)	Указывает, что следующий символ должен считываться в однобуквенном формате без учета последующих букв. Также может показывать, что однобуквенный формат считывается как определяемый пользователем формат. Дополнительные сведения см. далее.
d	Выводит день месяца в виде числа без ведущих нулей (например, 1). Если в определяемом пользователем числовом формате используется только один символ, используйте %d.
dd	Выводит день месяца в виде числа с ведущими нулями (например, 01).
ddd	Выводит сокращенное название дня недели (например, вс).
dddd	Выводит полное название дня недели (например, воскресенье).
M	Выводит месяц в виде числа без ведущих нулей (например, январь представляется числом 1). Если в определяемом пользователем числовом формате используется только один символ, используйте %M.
MM	Выводит месяц в виде числа с ведущими нулями (например, 01/12/01).
MMM	Выводит сокращенное название месяца (например, янв).
MMMM	Выводит полное название месяца (например, январь).

Пользовательские форматы даты и времени для функции FORMAT

Спецификация формата	Описание
gg	Отображает строку эры (например, н.э.)
h	Выводит часы в виде числа без ведущих нулей в 12-часовом формате (например, 1:15:15 PM). Если в определяемом пользователем числовом формате используется только один символ, используйте %h.
hh	Выводит часы в виде числа с ведущими нулями в 12-часовом формате (например, 01:15:15 PM).
H	Выводит часы в виде числа без ведущих нулей в 24-часовом формате (например, 1:15:15). Если в определяемом пользователем числовом формате используется только один символ, используйте %H.
HH	Выводит часы в виде числа с ведущими нулями в 24-часовом формате (например, 01:15:15).
m	Выводит минуты в виде числа без ведущих нулей (например, 12:1:15). Если в определяемом пользователем числовом формате используется только один символ, используйте %m.
mm	Выводит минуты в виде числа с ведущими нулями (например, 12:01:15).
s	Выводит секунды в виде числа без ведущих нулей (например, 12:15:5). Если в определяемом пользователем числовом формате используется только один символ, используйте %s.
ss	Выводит секунды в виде числа с ведущими нулями (например, 12:15:05).
f	Выводит доли секунды. Например, ff соответствует сотым долям секунды, а ffff — десятитысячным долям. В определяемом пользователем формате можно использовать до семи символов f. Если в определяемом пользователем числовом формате используется только один символ, используйте %f.
t	Выводит время в 12-часовом формате, добавляя букву A в верхнем регистре для времени до полудня и букву P в верхнем регистре для времени от полудня до 23:59. Если в определяемом пользователем числовом формате используется только один символ, используйте %t.

Пользовательские форматы даты и времени для функции FORMAT

Спецификация формата	Описание
tt	В локалях, использующих 12-часовой формат времени, выводит сочетание AM в верхнем регистре для времени до полудня и сочетание PM в верхнем регистре для времени от полудня до 23:59. В локалях, использующих 24-часовой формат времени, не выводит ничего.
y	Выводит год без ведущих нулей (0-9). Если в определяемом пользователем числовом формате используется только один символ, используйте %y.
yy	Выводит год в виде двух цифр, в случае необходимости указываются ведущие нули.
yyy	Выводит год в формате из четырех цифр.
yyyy	Выводит год в формате из четырех цифр.
z	Выводит сдвиг часового пояса без ведущих нулей (например, -8). Если в определяемом пользователем числовом формате используется только один символ, используйте %z.
zz	Выводит сдвиг часового пояса с ведущими нулями (например, -08).
zzz	Выводит полный сдвиг часового пояса (например, -08:00).

Функции родители-потомки

PATH (DAX)	PATH(<ID_columnName>, <parent_columnName>)	Возвращает разделенную строку текста с идентификаторами всех родителей текущего идентификатора, начиная с самого старого и продолжая до текущего.
PATHCONTAINS (DAX)	PATHCONTAINS (<path>, <item>)	Возвращает значение True , если указанный элемент не существует в указанном пути.
PATHITEM (DAX)	PATHITEM(<path>, <position>[, <type>])	Возвращает элемент в указанную позицию строки в результате оценки пути функции. Позиции подсчитываются слева направо.
PATHITEMREVERSE (DAX)	PATHITEMREVERSE(<path>, <position>[, <type>])	Возвращает элемент в указанную позицию строки в результате оценки пути функции. Позиции отсчитываются назад справа налево.
PATHLENGTH (DAX)	PATHLENGTH(<path>)	Возвращает количество родителей к указанному элементу в результате заданного пути

Прочие функции

DATATABLE	DATATABLE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}, {ValueN, ValueN+1...}...})	Предоставляет механизм объявления встроенного набора значений данных.
ERROR	ERROR(<text>)	Вызывает ошибку с сообщением об ошибке.
EXCEPT	EXCEPT(<table_expression1>, <table_expression2>)	Возвращает строки одной таблицы, которые не отображаются в другой таблице.
GENERATESERIES	GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])	Возвращает таблицу с одним столбцом, содержащую значения арифметического ряда, т. е. последовательность значений, в которой каждое отличается от предыдущего постоянным количеством. Имя возвращаемого столбца-Value.
GROUPBY	GROUPBY (<table>, [<groupBy_columnName1>], [<name>, <expression>]...)	Функция GROUPBY аналогична функции суммирования. Однако GROUPBY не выполняет неявное вычисление для добавляемых столбцов расширения. GROUPBY позволяет использовать новую функцию CURRENTGROUP () внутри агрегатных функций в добавляемых им столбцах расширения. GROUPBY пытается повторно использовать данные, которые были сгруппированы, что делает его высокоэффективным.

Прочие функции

INTERSECT	INTERSECT(<table_expression1>, <table_expression2>)	Возвращает пересечение строк двух таблиц с сохранением дубликатов.
ISEMPTY	ISEMPTY(<table_expression>)	Проверяет, пуста ли Таблица.
NATURALINNERJOIN	NATURALINNERJOIN (<leftJoinTable>, <rightJoinTable>)	Выполняет внутреннее соединение таблицы с другой таблицей. Таблицы соединяются на общих столбцах (по имени) в двух таблицах. Если две таблицы не имеют общих имен столбцов, возвращается ошибка.
NATURALLEFTOUTERJOIN	NATURALLEFTOUTERJOIN (<leftJoinTable>, <rightJoinTable>)	Выполняет внутреннее соединение таблицы с другой таблицей. Таблицы соединяются на общих столбцах (по имени) в двух таблицах. Если две таблицы не имеют общих имен столбцов, возвращается ошибка.

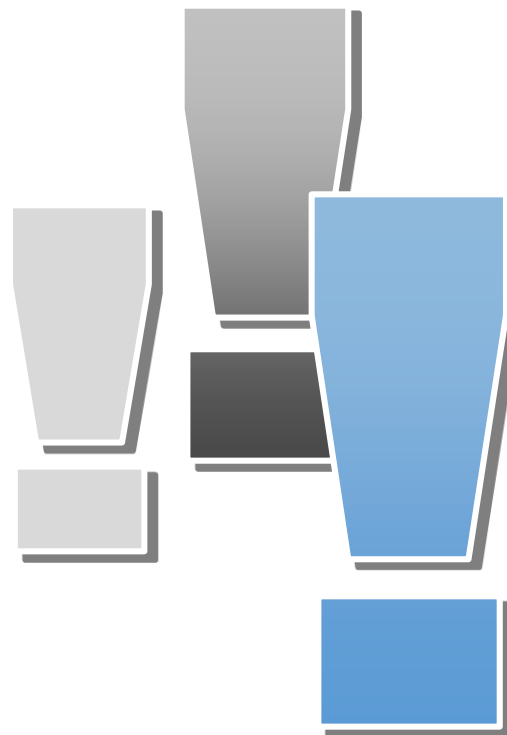
Прочие функции

SUMMARIZECOLUMNS	SUMMARIZECOLUMNS (<groupBy_columnName> [, <groupBy_columnName >]..., [<filterTable>]...[, <name>, <expression>]...)	Возвращает сводную таблицу по набору групп.
Table Constructor	{ <scalarExpr1>, <scalarExpr2>, ... } { (<scalarExpr1>, <scalarExpr2>, ...), (<scalarExpr1>, <scalarExpr2>, ...), ... }	Возвращает таблицу из одного или нескольких столбцов.
TREATAS	TREATAS(table_expression, <column>[, <column>[, <column>[,...]]])	Применяет результат табличного выражения в качестве фильтров к столбцам из несвязанной таблицы.
UNION	UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)	Создает таблицу объединения (соединения) из пары таблиц.
VAR	VAR <name> = <expression>	Сохраняет результат выражения в виде именованной переменной, которая затем может быть передана в качестве аргумента другим выражениям меры. После вычисления результирующих значений для выражения переменной эти значения не изменяются, даже если на переменную ссылается другое выражение.

Операторы и константы DAX

Тип оператор	Символ и назначение
Оператор скобок	() порядок очередности и группирование аргументов
Арифметические операторы	+ (сложение) - (вычитание/ знак) * (умножение) / (деление) ^ (возведение в степень)
Операторы сравнения	= (равно) > (больше, чем) < (меньше, чем) >= (больше или равно) <= (меньше или равно) <> (не равно)
Оператор объединения текста	& (объединение)
Логические операторы	&& (и) (или)

Спасибо за внимание!



Шевцов Василий Викторович

shevtsov_vv@rudn.university
+7(903)144-53-57