**Глава 26. Продвинутые вычисляемые столбцы в Power Pivot**

Это продолжение перевода книги Роб Колли. Формулы DAX для Power Pivot. Главы не являются независимыми, поэтому рекомендую начать сначала.

[Предыдущая глава](http://baguzin.ru/wp/?p=20570) [Содержание](http://baguzin.ru/wp/?p=19780#content) Следующая глава

Вычисляемые столбцы не являются сильной стороной DAX. Меры – это да! (поэтому почти вся книга посвящена им). А вот столбцы есть и в обычном Excel, поэтому, по большому счету, здесь мало что добавишь. На самом деле, вычисляемые столбцы в Power Pivot даже немного сложнее, чем обычные столбцы Excel, потому что Power Pivot не хватает ссылки в стиле "A1". А начнем мы с относительно простых трюков.



Рис. 26.1. Продажи в зависимости от температурного диапазона

### Группировка по значениям

Наш любимый пример группировки – объем продаж в зависимости от температуры. Мы импортируем таблицу данных о температуре (по дням), связываем ее с таблицей продаж, а затем выводим в сводную таблицу меру [Sales per Day], в зависимости от температурного диапазона (рис. 26.1). Вот формула, которую мы используем в вычисляемом столбце TempRange таблицы Temperature:

=SWITCH(TRUE()

 ; [Avg Temp] < 40; "Холодно"

 ; [Avg Temp] < 55; "Прохладно"

 ; [Avg Temp] < 70; "Тепло"

 ; "Жарко"

)



Рис. 26.2. Столбец группировки температуры по диапазонам (значения градусов в столбце AvgTemp указано по шкале Фаренгейта; t°С = (tF – 32)/1,8; таким образом, холодно – ниже 4°С, прохладно = 4–10°С, тепло = 10–21°С, жарко – более 21°С)

Что не так с нашей сводной таблицей на рис. 26.1? Диапазоны отсортированы по алфавиту, а хотелось бы по возрастанию температур. Попробуем отсортировать столбец TempRange по значениям температуры в столбце AvgTemp. Для этого в Power Pivot перейдите в таблицу Temperature, выделите столбец TempRange, и кликните кнопку *Сортировка по столбцам*. В открывшемся окне выберите *Сортировать по*: AvgTemp.



Рис. 26.3. Сортировка диапазонов столбца TempRange по значениям температуры в столбце AvgTemp

Это приводит к ошибке:



Рис. 26.4. Для сортировки каждому значению в столбце TempRange должно соответствовать только одно значение в столбце AvgTemp

Ну что ж, добавим еще один столбце в таблицу Temperature с уникальным кодом диапазона:

=SWITCH(TRUE()

 ; [Avg Temp] < 40; 1

 ; [Avg Temp] < 55; 2

 ; [Avg Temp] < 70; 3

 ; 4

)



Рис. 26.5. Кандидат на роль сортировочного столбца – TempRangeSeguence

Повторите сортировку (см. рис. 26.3). Но теперь установите *Сортировать по*: TempRangeSeguence. Получим желаемый результат:



Рис. 26.6. Теперь диапазоны выстроились в правильном порядке

### Сортировка по вычисляемому ключу

Рассмотрим столбец QtrYearLabel в таблице Periods:



Рис. 26.7. Обратите внимание, как каждому значению в столбце QtrYearLabel соответствует несколько значений в столбце PeriodID

Если мы хотим «правильно» сортировать по столбцу QtrYearLabel, столбец PeriodID не подходит, и нам нужен новый вычисляемый столбец. Но на этот раз функция SWITCH() не спасет нас (поняли, почему?). Нам нужен какой-то шаблон для расчета уникальной последовательности кварталов. Например, такой: [Year] \* 4 + [Qtr]. А в общем виде: <Year column> \* <number of periods per year> + <period column>. Здесь периодом может быть квартал (4 за год), месяц (12), неделя (52), семестр (2) и т.д.



Рис. 26.8. QtrSequence – уникальный столбец-идентификатор для сортировки столбца QtrYearLabel

### Суммирование в таблице поиска

Как правило, мы используем меры, чтобы суммировать продажи по продуктам. Но бывают ситуации, когда полезно создать вычисляемый столбец в таблице Products, отражающий продажи по каждому продукту. Вы уже видели такой столбец в [главе о контексте строк](http://baguzin.ru/wp/?p=20521):

=CALCULATE(SUM(Sales[SalesAmt]))



Рис. 26.9. Функция CALCULATE превращает контекст фильтра в контекст строки, и возвращает общий объем продаж для каждого продукта

Усложним задачу: добавим в таблицу Products столбец с объемом продаж, соответствующим общим продажам для категории, к которой принадлежит продукт.

### Использование функции EARLIER()

Это сложная для понимания функция, но, чтобы начать ее использовать, достаточно познакомиться с шаблоном:

1 =CALCULATE(

2 SUM(Sales[SalesAmt])

3 ; FILTER(

4 Products;

5 Products[Category]=EARLIER(Products[Category])

6 )

7 )

Благодаря этой формуле в столбце TotalCaregorySales в каждой строке выводится общий объем продаж всех продуктов соответствующей категории (рис. 26.10). Для аксессуаров он один, для велосипедов – другой, но для одной и той же категории – одинаков.



Рис. 26.10. Каждая строка в соответствующей категории возвращает сумму продаж всей категории

Обратите внимание, что в формуле в строке 4 нам не нужно использовать снятие фильтра с помощью ALL(Products), как мы делали [ранее](http://baguzin.ru/wp/?p=20548). Дело в том, что ранее у нас была формула меры, которая наследует фильтры из сводной таблицы. А сейчас у нас формула вычисляемого столбца, для которой не существует исходного контекста фильтра (столбцы в Power Pivot не имеют [контекста фильтра](http://baguzin.ru/wp/?p=20521)). Таким образом, нет необходимости очищать фильтр с помощью ALL().

Прежде чем разобраться в том, какое условие проверяется в строке 5, еще раз проговорим, как действует описанные выше формула. Функция CALCULATE суммирует продажи по всем строкам, отобранным функцией FILTER. Мы могли бы просто суммировать =SUM(Sales[SalesAmt]), но тогда в каждой строке таблицы Products было бы одно и то же число. Функция SUM() не знает контекста строки. А вот CALCULATE использует контекст строки, и возвращает сумму только по тому ProductKey, который представлен в строке.

Функция FILTER не использует контекст строки, т.е. любое условие внутри нее будет работать на всей таблице Products. При этом функция FILTER – итератор, т.е. она будет проходить таблицу Products строка за строкой. Функция EARLIER как бы выходит наружу этой вложенности (FILTER внутри CALCULATE), и использует контекст строки функции CALCULATE. Таким образом, внутри функции FILTER берется вся таблица Products, а затем строка за строкой отбираются те, для которых категория совпадает с той, что в строке, для которой рассчитывается вся сумма CALCULATE(SUM(…);FILTER(…)).

Проиллюстрируем картинками. CALCULATE, скажем, для строки с ProductKey=597, задает контекст строки Category = "Bikes":



Рис. 26.11. Исходный контекст строки

Вложенная внутрь функции CALCULATE функция FILTER (рис. 26.12) по мере прохождения по таблице Products оценивает условие для каждой строки и, если сравнение дает ИСТИНА, то включает эту строку в возвращаемую таблицу.



Рис. 26.12. Итеративная функция FILTER, проходя по таблице Products, проверяет условие

Например, когда FILTER находится в строке ProductKey=234 (рис. 26.12), условие в формуле в строке 5 сравнивает Products[Category] = "Clothing" (текущая строка функции-итератора FILTER), с контекстом строки для CALCULATE. Именно в этом назначение функции EARLIER() – выйти из контекста строки вложенной функции, и обратиться к контексту строки внешней функции.

Для строки ProductKey=234 контекст строки внутри и вне FILTER не совпадает, и строка не будет возвращена, чтобы принять участие в подсчете продаж велосипедов:



Рис. 26.13. Функция EARLIER помогает нам перейти к «внешнему» контексту строки, или, как говорят, к контексту строки, существующему на один уровень выше текущей вложенности

Если все это слишком сложно, не переживайте. Просто применяйте шаблон. Понимание придет с практикой.

CALCULATE(<функция вычисления, например, SUM()>;

FILTER (<таблица>);

<условие с использованием EARLIER()>

)

)

Простейшее условие: table[column] = EARLIER(table[column]). Где знак равенства можно заменить на иной оператор сравнения: >, <, <=, >=, <>. Вы также можете использовать несколько условий, перечисленных через && (логическое И) или || (логическое ИЛИ).

### Скользящее среднее

Подход, описанный выше, был использован нами в одном медико-биологическом исследовании дыхания крыс))

=CALCULATE(MIN(Data[value]);

 FILTER(Data;

 Data[Rat]=EARLIER(Data[Rat]) &&

 Data[TimeID] <= EARLIER(Data[TimeID]) +5 &&

 Data[TimeID] >= EARLIER(Data[TimeID]) -5

 )

)

Обратите внимание на несколько условий, которые перечислены с использованием логического И – &&. Первое условие, как и в предыдущем примере – только строки для выбранной крысы должны быть подсчитаны, иначе мы учтем чье-то чужое дыхание. Второе и третье условия говорят «считайте только пять строк, до меня, и пять строк, сразу после меня». Т.е., отбираемый диапазон составляет 11 строк. В результате формула возвращает наименьшее значение в текущем «окне» из 11 строк. На основе этого другой вычисляемый столбец – MatchesMin – может определить, соответствует ли значение в столбце Value текущей строки минимуму из 11 строк.



Рис. 26.14. Нахождение пиков



Рис. 26.15. Отражение пиков в сводной таблице

Дополнительные материалы на эту тему, см. <http://ppvt.pro/PkSniff>, <http://ppvt.pro/Peak2Freq>, <http://ppvt.pro/FzzyTime>.

### Вычисляемые столбцы являются статическими

Вычисляемые столбцы являются статическими, значения вычисляются и сохраняются в столбце. Есть только два события, которые запускают вычисление / пересчет вычисляемого столбца:

* Определение или переопределение: когда вы пишите (или редактируете) формулу для вычисляемого столбца и нажимаете Enter.
* Обновление данных: при обновлении таблицы Power Pivot, содержащей вычисляемый столбец.

В отличие от этого, фактические значения мер никогда не сохраняются, а всегда вычисляются динамически на основе построенной сводной таблицы.

Давайте посмотрим, как изменяются числа в сводной (рис. 26.16):

1. Вычисляемый столбец SalesPerProduct = CALCULATE(SUM(Sales[SalesAmt]))
2. Мера [SalesPerProduct Measure]=CALCULATE(SUM(Sales[SalesAmt]))



Рис. 26.16. Сходство между вычисляемыми столбцом и мерами только поверхностное

Фильтр в срезе не влияет на вычисляемый столбец, но изменяет значение меры.

### Затраты ресурсов на вычисляемые столбцы

Вычисляемые столбцы «пожирают» большой объем оперативной памяти, и могут снижать скорость работы. Это может ощущаться при изменении координат / срезов в сводной таблице. Мы ранее обсудили это в [главе о производительности](http://baguzin.ru/wp/?p=20351).

Однако при обновлении данных с вычисляемыми столбцами возникает еще одна проблема – их полный пересчет. Если у вас очень большая модель, возможно, потребуется запускать ее обновление в нерабочее время.

Некоторые виды вычисляемых столбцов могут съедать очень много оперативной памяти. Вспомните наш пример нахождения пиков. Формула написана для выделения предыдущих и следующих пяти строк плюс текущей строки. Но чтобы найти эти 11 строк, Power Pivot каждый раз просматривает всю таблицу, по одной строке за раз, и решает, принадлежит ли каждая строка к этим одиннадцати.

В обычном Excel этой цели служит относительная ссылка. В Power Pivot нет ссылок в стиле А1, и с относительным позиционированием Power Pivot справляется хуже, чем Excel. Если в таблице миллион строк, то для расчета каждой из них в вычисляемом столбце потребуется миллион итераций. А это уже 1012 операций!