**Глава 13. Введение в DAX-функцию FILTER()**

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=19903) [Содержание](http://baguzin.ru/wp/?p=19780#content) Следующая глава

Функция FILTER() используется, когда в аргументе <filter> функции CALCULATE() вам необходим более сложный расчет, чем проверка типа "<столбец> равен <фиксированному значению>", "<столбец> больше <фиксированного значения>" и т.п. Примеры логических выражений <filter>, требующих использования функции FILTER(): < столбец> = <мера>, <столбец> = <формула>, <столбец> = <столбец>, <мера> = <мера>, < мера> = <формула>, <мера> = <фиксированное значение> (вместо знака равенства может стоять другой разрешенный оператор сравнения).



Рис. 13.1. Сводная по продуктам со стоимостью по прайс-листу выше, чем порог, выбранный на срезе

Можно также использовать функцию FILTER() в качестве аргумента <table> для таких функций, как COUNTROWS() и SUMX(), чтобы эти функции работали с подмножеством таблицы, а не со всеми строками в текущем наборе фильтров. Такое использование будет описано позже.

Синтаксис функции: FILTER(<table>; <single "rich" filter>)

Мы рекомендуем:

1. Когда вы используете FILTER(), используйте его для таблиц поиска (подстановки), но не для таблиц данных.
2. Никогда не используйте FILTER(), когда функция CALCULATE() сама выполнит работу.

FILTER() – это [итератор](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80), который проходит строку за строкой в таблице (первый аргумент функции) для оценки условий фильтра (второй аргумент). Фильтр не одинок в этом, существует целое семейство функций-итераторов, или X-функций, с которыми вы познакомитесь позже.

Давайте вернемся к нашему [примеру из предыдущей главы](http://baguzin.ru/wp/?p=19903#example), где мы хотели включить только те продукты, у которых столбец Products[ListPrice] был >= наша мера [MinListThreshold] (рис. 13.1).

Формула, которую мы использовала для меры [Product Sales Above Selected List Price]:

CALCULATE (

[Total Sales];

FILTER (Products;

Products[ListPrice] >= [MinListThreshold])

)

Давай проверим, следуем ли мы рекомендациям использования функции FILTER():

1. Products – это таблица поиска, а не данных (как, например, Sales).
2. Мы сравниваем столбец Products[ListPrice] с мерой [MinListThreshold], что невозможно сделать в аргументе функции CALCULATE().

Оба правила соблюдены.

Перейдем теперь ко второй мере [Products Above Selected List Price]. Она подсчитывает количество продуктов с ценой по прайс-листу выше порогового значения, выбранного на срезе (см. рис. 13.1). Поэтому, для начала нам нужна базовая мера, которая просто подсчитывает количество продуктов:

[Product Count] =COUNTROWS(Products)

Обратите внимание, что мы определили эту меру таблице Products, так как мера подсчитывает строки именно в этой таблице:



Рис. 13.2. Мера [Product Count]

[Products Above Selected List Price] =

CALCULATE(

[Product Count];

FILTER ( Products;

Products[ListPrice] >= [MinListThreshold] )

)

Давайте проверим, корректно ли работает мера. Для начала просто изменим выбор среза и убедимся, что значения сводной тоже изменились: сравните рис. 13.1 и 13.3.



Рис. 13.3. Мы ожидали, что обе меры вернут б***о***льшие значения при фильтре $20, что и произошло

Далее давайте поместим Products[ProductKey] в строки сводной таблицы и установим срез на $3000. С такими установками продуктов должно быть немного:



Рис. 13.4. В самой дорогой категории (порог выше $3000) нашлось лишь 13 продуктов

И, наконец, давайте для последней таблицы поместим в строки еще и саму цену ListPrice:



Рис. 13.5. Мы видим, что в списке только продукты с ценой по ListPrice более $3000

Пока все наши проверки были основаны на сводной таблице в Excel. А теперь давайте перейдем в Power Pivot на закладку Products и в колонке ListPrise установим фильтр *Больше или равно 3000*:



Рис. 13.6. Фильтр в колонке ListPrise таблицы Products в Power Pivot

В нижней части окна Power Pivot мы увидим количество отобранных по фильтру записей:



Рис. 13.7. По фильтру в в Power Pivot отобрано 13 записей

### Выбор верхней и нижней границ диапазона

Усложним наш пример, и создадим в Power Pivot таблицу PriceTiers с нижней, верхней границами и названием диапазонов:



Рис. 13.8. Новая несвязанная таблица

Обратите внимание, ценовые уровни перекрываются. То есть некий продукт может принадлежать более чем одному диапазону, что делает использование связанных таблиц невозможным. Решение возможно только с использованием несвязанных таблиц.

Теперь мы определим две пороговые меры для этой таблицы:

[PriceTierMin] =MIN(PriceTiers[MinPrice])

и

[PriceTierMax] =MAX(PriceTiers[MaxPrice])

Поместим эти срезы в область *Значений* сводной таблицы и используем столбец RangeName в качестве среза:



Рис. 13.9. Название диапазона, выбранное на срезе, выбирает два пороговых значения

Если диапазоны отсортированы по алфавиту, кликните правой кнопкой мыши на срезе, и выберите *Сортировать как в источнике данных* (естественно, в Power Pivot диапазоны в таблице PriceTiers должны быть размещены в «правильном» порядке):



Рис. 13.10. Сортировка диапазонов «по смыслу»

Теперь нам нужны меры для количества продуктов (в таблице Products) и объема продаж (в таблице Sales), которые учитывают выбор пользователя в срезе:

[Product Count MinMaxTier] =

CALCULATE(

[Product Count];

FILTER(

Products;

Products[ListPrice] >= [PriceTierMin]

&& Products[ListPrice] <= [PriceTierMax]

)

)

Так как FILTER() поддерживает только одно сравнение, мы используем оператор &&. Строка таблицы Products должна соответствовать обоим условиям фильтра, чтобы быть отобранной.

Поскольку CALCULATE() поддерживает несколько аргументов <filter>, мы могли бы сделать это без оператора &&, используя две функции FILTER: CALCULATE(<мера>; FILTER(...); FILTER(...)). Это дало бы тот же результат, но… с потерей производительности.

[Total Sales MinMaxTier] =

CALCULATE(

[Total Sales];

FILTER(

Products;

Products[ListPrice] >= [PriceTierMin]

&& Products[ListPrice] <= [PriceTierMax]

)

)

Теперь мы создадим сводную таблицу с этими двумя мерами:



Рис. 13.11. Сводная с двумя новыми мерами

### Способ визуализации несвязанных таблиц

Несвязанные таблицы по определению не имеют связей с другими таблицами в модели. Если мы посмотрим на диаграмму, то увидим, что таблица PriceTiers «болтается неприкаянной»:



Рис. 13.12. Таблица ценовых уровней PriceTiers не имеет стрелок связи (как и ожидалось)

Но когда мы используем меры [PriceTierMin] и [PriceTierMax], таблица PriceTiers действует как таблица поиска, так как фильтр PriceTiers (выбор пользователя в срезе) влияет на вычисления. Поэтому мы часто говорим, что несвязанные таблицы имеют связь *пунктирная линия* с таблицами, содержащими соответствующие меры c функцией FILTER():



Рис. 13.13. Вы должны думать о несвязанных таблицах, как о связанных косвенно, через фильтры мер