**Глава 10. Power Pivot. Мышление в нескольких таблицах**

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

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

В Excel исходные данные желательно собрать в одной таблице, а уже затем строить на ее основе сводную. Поскольку данные часто находятся в нескольких таблицах, Power Pivot приспособлен к их обработке. Более того, мы рекомендуем хранить «разношерстные» данные внутри Power Pivot в разных таблицах. А сами таблицы связывать между собой.

Откройте Excel-файл ch10\_MultipleTables.xlsx, перейдите на вкладку Power Pivot, и кликните кнопку *Управление*. В окне Power Pivot перейдите на вкладку *Конструктор* и кликните кнопку *Создание связи*. В окне *Создание связи* выберите таблицы и столбцы, как указано ниже:



Рис. 10.1. Окно *Создание связи* в Power Pivot

### Представление *Диаграммы*

В Power Pivot существует два основных представления: данных и диаграммы. Переключение осуществляется кнопками на ленте (меню *Главная* –> *Просмотр*) или внизу в правой части окна:



Рис. 10.2. Кнопки переключения отображения



Рис. 10.3. Представление Диаграммы. Отображаются три таблицы, две из которых связаны

Обратите внимание, что при выделении связи показываются поля, по которым осуществляется связь. Звездочка означает множество, а 1 – единицу. В нашем примере установлена связь «один ко многим» от таблицы Products к таблице Sales. Т.е., значения в поле ProductKey являются уникальными в таблице Products, но будут повторяться в таблице Sales.

В Excel 2010 и 2013 дизайн окна *Создание связи* немного отличался: явно определялась основная таблица и таблица подстановки (аналогично функции ВПР в Excel). Попытка выбрать в верхнем поле таблицу Products, а в нижнем – Sales, приводило к ошибке. В Excel 2016 такой ошибки не возникает. Power Pivot сам исправляет ситуацию, и в любом случае создаст связь «один ко многим», как на рис. 10.3.

### Использование связанных таблиц Power Pivot в сводной таблице Excel

Создадим сводную таблицу, которая использует [ProductKey] в строках:



Рис. 10.4. [ProductKey] не несет особого смысла

Удалим [ProductKey] и добавим [ProductName] из таблицы Products:



Рис. 10.5. C [ProductName] таблица стала яснее

Благодаря установленной связи мы можем использовать в нашей сводной таблице любые поля из исходных таблиц Sales и Products (рис. 10.6). Почему это работает? Потому что набор фильтров передается из таблицы в таблицу.



Рис. 10.6. Группировка по цветам (еще одному столбцу в таблице Products)

Рассмотрим ячейку С7 меры [Normal Sales] и проследим, как работает фильтр. Сначала фильтр Color="Red" применяется к таблице Products:



Рис. 10.7. Фрагмент таблицы Products, отфильтрованной по Color="Red"

Столбец [ProductKey] не фильтруется напрямую, но он был уменьшен до подмножества из 63 значений, соответствующих фильтру Color="Red" в таблице Products. Этот отфильтрованный набор [ProductKey] через связь между таблицами передается в таблицу Sales. А затем выполняется агрегирование по отфильтрованной по столбцу [ProductKey] таблице Sales. Мы можем прояснить шаг 4 золотого правила С: фильтры следуют за связями. Другими словами, фильтры передаются из таблицы в таблицу, если последние связаны.

И еще. В обратном порядке (из таблицы Sales в таблицу Products) фильтр не передается. *Фильтр следует за связью*, направление которой явным образом указано на рис. 10.3 (Products–>Sales).

Для лучшей визуализация фильтров мы всегда будем располагать таблицы поиска (подстановки), над таблицами данных. Таким образом, фильтры перетекают *вниз* в таблицы данных (см. рис. 10.3).

Вернемся к нашим трем таблицам (см. рис. 10.3) и создадим связь Customers–>Sales. На этот раз мы сделаем это в *представлении диаграммы*. Для этого кликните на поле CastomKey в таблице Customers и перетащите его в таблицу Sales на поле с именем CastomKey. Можно сделать и наоборот: перетащить поле CastomKey из таблицы Sales в таблицу Customers. Как и в окне *Создание связи* Power Pivot сам определяет *правильное* направление связи и настроит ее соответствующим образом.



Рис. 10.8. Две таблицы поиска, обе *над* таблицей данных, для которой они поставляют фильтры

Давайте применим фильтры из всех связанных таблиц поиска в таблице данных:



Рис. 10.9. Продажи в разрезе субкатегорий товаров и семейного положения (М – семейные, S – одинокие)

### Фильтры функции CALCULATE() также передаются по направлению связей

Давайте определим новую меру, используя столбец [NumberChildrenAtHome] в таблице подстановки [Customers]:

[Sales to Parents] = CALCULATE([Total Sales]; Customers[NumberChildrenAtHome] > 0 )

И сравните ее с базовой мерой – [Total Sales]:



Рис. 10.10. Фильтры функции CALCULATE() также передаются по направлению связей

Мы видим, что [Sales to Parents] возвращает меньшие числа, чем базовая мера [Total Sales]. Фильтры в CALCULATE() применяются до того, как они будут проходить через связи. Таким образом, мы можем уточнить 4-й шаг третьего золотого правила мер DAX: *примените фильтры к таблицам подстановки; это приведет к тому, что в таблице данных будут отобраны только строки, соответствующие набору всех фильтров.*