**Глава 18. Множественные таблицы данных Power Pivot с различной степенью детализации**

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=20299) [Содержание](http://baguzin.ru/wp/?p=19780#content) Следующая глава

Ваши таблицы данных могут иметь различную структуру:

* В детализированных наборах данных каждая строка содержит относительно немного информации. Высокодисперсные наборы данных, как правило, содержат много коротких строк.
* В менее детализированных наборах данных каждая строка представляет собой «более толстый» фрагмент данных, и поэтому эти наборы данных, как правило, содержат меньше строк.
* Если ваши таблицы данных имеют одинаковую степень детализации, предыдущая глава – это все, что вам нужно. Но если у них разные степени детализации, вам пригодятся идеи, содержащиеся в этой главе.
* В общем случае, чем более дисперсен набор данных, тем более мощной и гибкой может быть ваша модель.

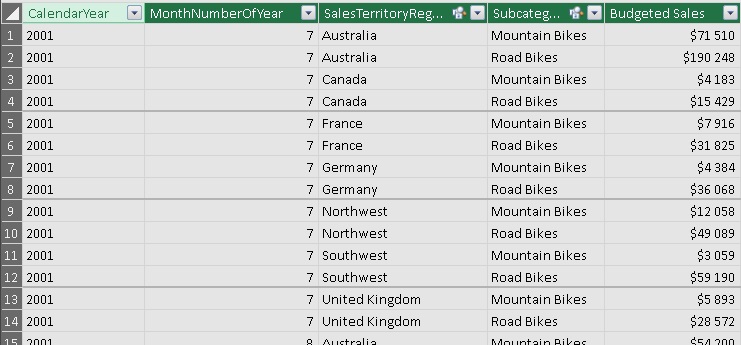


Рис. 18.1. Бюджет продаж, импортированный в модель данных Power Pivot

### Бюджет и факт

У вас есть таблица продаж (фактические данные), где каждая строка представляет отдельную транзакцию. В нашем примере это 60 тысяч строк. У вас также есть бюджет (план продаж), менее детализированный (рис. 18.1). Как правило, нас интересует, как наши продукты продаются по сравнению с бюджетом?

Решение этой проблемы в обычном Excel утомительно. Решение включает в себя создание двух сводных – одной для продаж, другой для бюджета, а затем написание формул, которые индексируют каждую сводную, чтобы сформировать единый отчет «продажи по сравнению с бюджетом». Требуется некоторое время, чтобы подготовить такой отчет. А затем, когда кто-то с неизбежностью захочет увидеть чуть-чуть другой формат или уровень детализации, вы потратите почти столько же времени, как и в первый раз…

В Power Pivot продажи и бюджет могут сосуществовать в одной и той же сводной таблице. При том что в модели данных это будут две разные таблицы. Всё что нам нужно, это создать «правильные» связи. Если это было совсем просто, то не понадобилась бы отдельная глава. К сожалению, нас подстерегает проблема: таблица [Budget] не связывается ни с одной из наших таблиц поиска. Например, попытка связать [Budget] с [Products], используя единственный общий столбец – [Subcategory] (рис. 18.2), приводит к ошибке «многие ко многим».

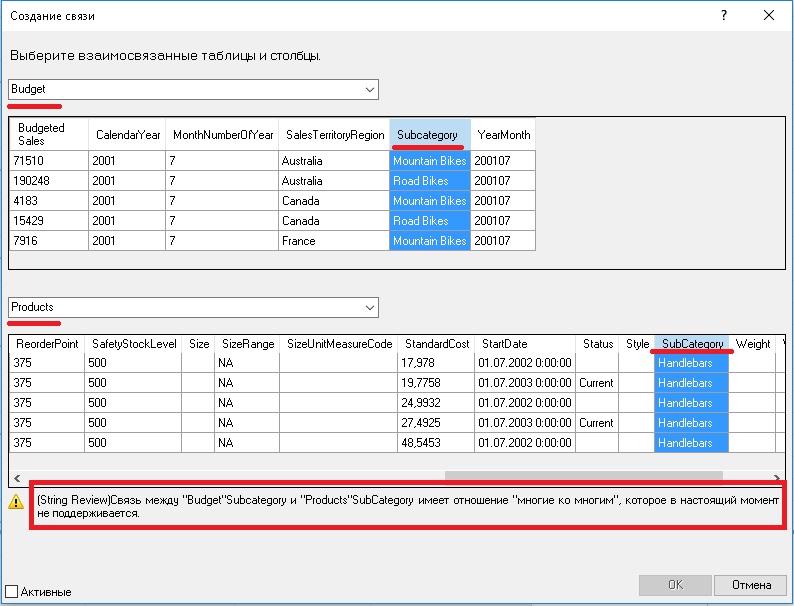


Рис. 18.2. Попытка связать бюджет с продуктами… приводит к ошибке «многие ко многим»

И с этим ничего нельзя поделать, так как каждое значение подкатегории (например, «горные велосипеды») появляется много раз в каждой таблице.

Нам нужна таблица поиска подкатегорий – [Subcategory], которая будет содержать только уникальные записи (рис. 18.3).



Рис. 18.3. Новая таблица поиска [Subcategory]

Затем мы связываем таблицу [Subcategory] с таблицами [Products] и [Budget]. Теперь наша диаграмма выглядит так:

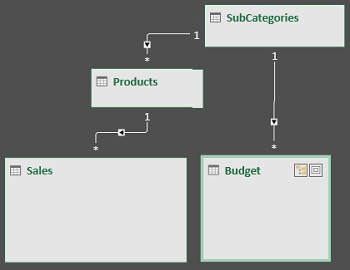


Рис. 18.4. Фрагмент модернизированной модель данных

Контекст фильтра перетекает из подкатегорий в продукты, а затем из продуктов в продажи.

Проще всего создать новую таблицу поиска [Subcategory] с помощью Power Query. Вы можете автоматически создать уникальный список значений подкатегорий, используя запрос к базе данных и преобразовав данные в [Power Query](http://baguzin.ru/wp/?p=19403) (чью команду *Удалить дубликаты* также можно назвать *Создать таблицу поиска из Таблицы данных*). Это избавит вас от ручного обновления в будущем, если появятся новые подкатегории (или удаляться старые).

Поскольку детализация бюджета по датам сводится только к парам год/месяц, нам нужна новая таблица поиска с той же степенью детализации (рис. 18.5).

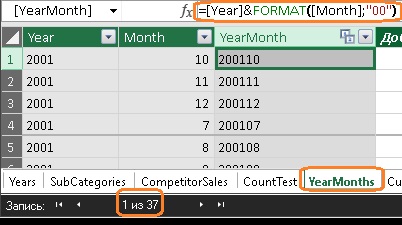


Рис. 18.5. Новая таблица поиска [YearMonths]

[YearMonths] – вычисляемый столбец. Функция FORMAT() используется для добавления нуля перед однозначными числами месяцев. Это не обязательно, но для правильной сортировки значений в столбце [YearMonths] очень полезно.

Мы добавляем такой же вычисляемый столбец в таблицы [Budget] и [Calendar], а затем создаем связи таблицы [YearMonths] с таблицами [Budget] и [Calendar]. Для таблицы [SalesTerritory] нам не нужно создавать новую таблицу поиска. Здесь [Budget] соответствует уже имеющейся детализации таблицы [SalesTerritory]. Поэтому мы просто создаем связь между этими таблицами. Так как таблиц много, отключите отображение деталей. Для этого в правом нижнем углу окна Power Pivot кликните *Отображение*, и снимите галки (рис. 18.6).

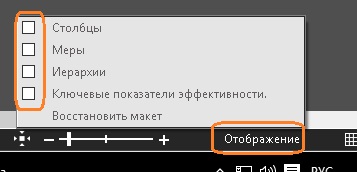


Рис. 18.6. Детали модели данных скрыты

Теперь мы можем построить единую сводную таблицу, используя меры из таблиц [Sales] и [Budget], а в качестве контекста фильтров – только поля из общих таблиц поиска. «Общими» будут те таблицы поиска, которые фильтруют обе наши таблицы данных. В нашем примере существует три общие таблицы поиска, отмеченные звездочками на рис. 18.7.

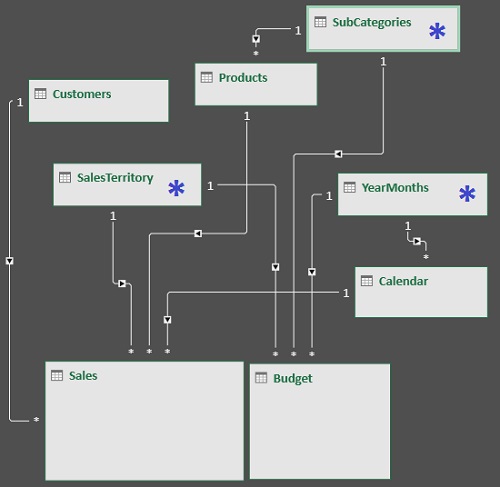


Рис. 18.7. Модель данных и направление общих контекстных фильтров

В таблице [Budget] мы создали меру [Total Budgeted Sales] = SUM(Budget[Budgeted Sales]). Поместим ее в сводную таблицу вместе с [Total Sales] из таблицы Sales:

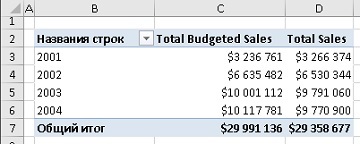


Рис. 18.8. Меры из разных таблиц данных: [Budget] и [Sales]

Обратите внимание, что в область *Строки* сводной вы поместите «правильное» поле Year из таблицы [YearMonths]. Если вы пометите сюда поле Year из таблицы [Calendar] у вас отобразятся неверные данные в столбце Total Budgeted Sales (рис. 18.9). Это связано с тем, что таблица [Calendar] не создает контекста фильтров для таблицы [Budget]. Чтобы уменьшить вероятность таких ошибок, рекомендую переименовывать поля, используя префиксы или суффиксы. Названия подскажут вам из каких таблиц поступают те или иные поля. В нашем примере поле *Год* из таблицы [Calendar] было названо CalendarYear.

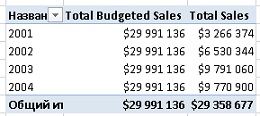


Рис. 18.9. Поле *Год* взято из неверной таблицы – [Calendar]

### Гибридные меры с данными из таблиц с различной детализацией

Мы можем создать новые меры, которые ссылаются (и сравнивают) меры из таблиц Budget и Sales, несмотря на их различную детализацию.

[Sales vs. Budget] = ([Total Sales] - [Total Budgeted Sales])/[Total Budgeted Sales]

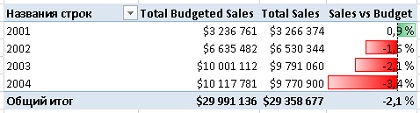


Рис. 18.10. Продажи в сравнении с бюджетом

Мы добавили условное форматирование (которое не было автоматическим). Теперь можно удалить две исходные меры, оставив только [Sales vs. Budget] и поиграть со срезами. Например, так:

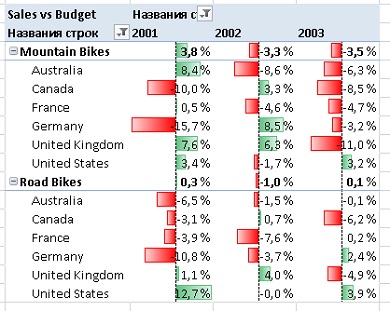


Рис. 18.11. Sales vs. Budget по линейкам продуктов

### Использование третьего загадочного аргумента функции RANKX()

Допустим, мы получили данные о продажах конкурента: насколько хорошо их велосипеды продавались в течение последних нескольких лет (рис. 18.12). И да, информация скупа.

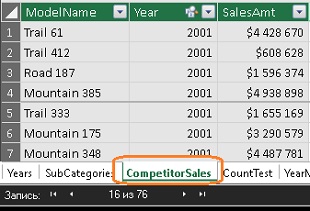


Рис. 18.12. Продажи конкурента

Задача – совместно ранжировать продажи конкурентов и собственные. Например, если одна из наших моделей показала объем продаж 3 млн. долл., а три лучшие модели конкурента – 4, 3,5 и 2,5 млн. долл., это означает, что наша модель занимает 3-е место.

Несоответствие детализации года в таблицах CompetitorSales и Sales требует создать новую таблицу подстановки – Years (рис. 18.13; напомним, что таблица данных Sales фильтруется таблицей подстановки YearMonths, в которой поле *Год* не является уникальным).

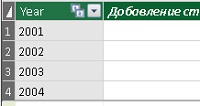


Рис. 18.13. Новая таблица подстановки – Years

Добавим в таблицу CompetitorSales меру [Compete Sales] = SUM(CompetitorSales[SalesAmt]). А затем создадим кросс-ранговую меру в таблице Sales (или в таблице Products, если вам так больше нравится):

[Model Sales Rank vs Competition] =

RANKX(

VALUES(CompetitorSales[ModelName]);

[Compete Sales];

[Total Sales]

)

Формула начинается так, как будто она собирается просто ранжировать продукты конкурентов по объему продаж (см. подробнее о [RANKX](http://baguzin.ru/wp/?p=20080#myRANKX)). Аргумент VALUES(CompetitorSales[ModelName]) возвращает таблицу, в которой каждой модели соответствует объем продаж из таблицы CompetitorSales. Аргумент [Compete Sales] означает, что измерение, с помощью которого модели будут ранжироваться – мера [Compete Sales]. До сих пор мы имели дело с обычным использованием функции RANKX().

А вот аргумент [Total Sales] означает, что мы собираемся взять значение [Total Sales] в нашем текущем контексте фильтра (который расположен в левой колонке сводной таблицы и представлен ModelName нашей компании) и вставить его в порядок, установленный первыми двумя аргументами (рис. 18.14).



Рис. 18.14. Кросс-ранговое сравненное объемов продаж наших и конкурентных моделей. Наша топовая модель занимает 14-е место, уступая первым 13 моделям конкурента.

И, так как обе сводные таблицы фильтруются по таблице подстановки Year, мы можем добавить *Год* в качестве среза для обеих сводных (рис. 18.15). В 2003 году наши показатели получше!

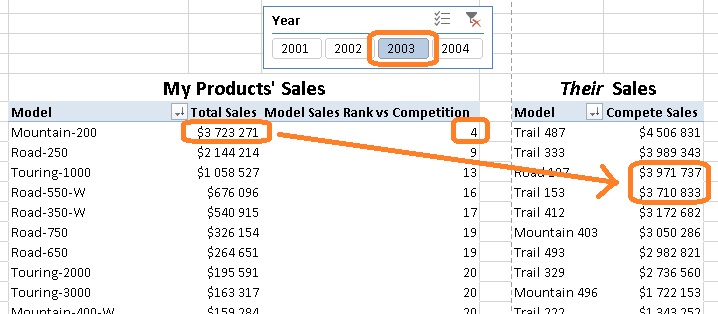


Рис. 18.15. Кросс-ранговое сравненное для 2003 г.