**Глава 14. Управление датами 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=19926) [Содержание](http://baguzin.ru/wp/?p=19780#content) Следующая глава

Различают стандартный и пользовательский календарь. Эта глава посвящена первому. Под стандартным мы понимаем календарь со следующими свойствами:

* Февраль имеет 28 дней (29 в високосные годы), а все остальные месяцы 30 или 31 день.
* Кварталы состоят из трех месяцев подряд
* В году 365 дней (366 – в високосный год)
* Любой месяц в этом году может иметь больше или меньше выходных (или любых других дней), чем в том же месяце в прошлом году. Другими словами, стандартный календарь – это календарь, который висит у вас на стене.

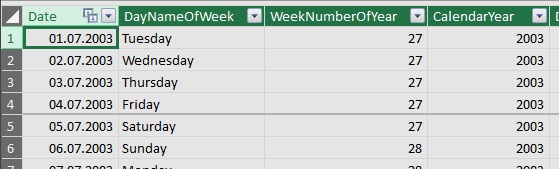


Рис. 14.1. Фрагмент таблицы *Календарь* в Power Pivot

Функции аналитики времени Power Pivot работают в предположении, что используется стандартный календарь.

С другой стороны, стандартный календарь создает много проблем, которые часто неприемлемы:

1. Сравнение этого месяца с прошлым часто «не справедливо», например, когда в прошлом месяце был 31 день, а в этом – только 30. Мы действительно на 3% хуже в этом месяце или это просто из-за разного количества дней?
2. Даже два месяца одинаковой продолжительности часто сравнивать не справедливо, поскольку они содержат разное количество выходных дней.
3. Иногда единица измерения не похожа на настенный календарь; например, семестры в академическом мире и сезоны – в спортивном.
4. Иногда (например, в науке), мы хотим сравнить абсолютно одинаковые периоды времени вместо календарных периодов.

По нашему опыту, по крайней мере, половина организаций измеряет себя пользовательскими календарями. Например, розничные предприятия, очень чувствительны к двум первым проблемам. Поэтому мы займемся пользовательским календарем позже.

### Создание календаря

При работе с датами создайте в Power Pivot отдельную таблицу (ее не обязательно называть календарь, но так всё же удобнее). Существует множество способов ее создания. Вот лишь несколько вариантов:

1. Импортируйте из базы данных, например, [этой](https://powerpivotpro.com/2014/12/sql-date-tables-in-power-pivot/).
2. Создайте в Excel, а затем загрузите в Power Pivot. Может возникнуть проблема, если нужен динамический календарь (например, до текущей даты). Воспользуйтесь [генератором календаря в Excel](http://ppvt.pro/xlCalendar) (потребуется русификация).
3. Сгенерируйте с помощью Power Query, см. [здесь](https://powerpivotpro.com/2014/02/the-ultimate-date-tablerevisited/) и [здесь](https://powerpivotpro.com/2015/02/create-a-custom-calendar-in-power-query/).
4. Импортируйте календарь из Azure DataMarket (или из другого места в Интернете). Есть, по крайней мере, одна таблица календаря, доступная для бесплатной загрузки, созданная Бояном Пеневым, см. [здесь](https://powerpivotpro.com/2011/11/the-ultimate-date-table/).

### Свойства таблицы Календарь

* Содержит хотя бы один столбец с типом данных *Дата*.
* Содержит одну строку на каждый день.
* Содержит все последовательные даты, без пропусков (даже если вы не работаете по выходным, эти дни должны быть в календаре).
* Связана с таблицей данных (типа Sales)
* Содержит множество столбцов для всех ваших мыслимых группировок и меток; например, номер недели/месяца/квартала, день недели, праздник, выходной и др. (но если вы минималист, то ничто не мешает ограничиться одним столбцом с датами).
* В идеале, охватывает только релевантные даты. Если ваш бизнес открылся в 2001 году, зачем начинать календарь с 2000 года!? И если сегодня 13 января 2019 г., то какой смысл иметь в календаре 14 января 2019 г? Это одно из самых сложных требований. Вот почему нам нравится получать календарь из базы данных. Это необязательно, но со временем вы найдете это очень полезным.

Итак, наш календарь представлен на рис. 14.1. Свяжем его с таблицей Sales, используя столбцы Date:

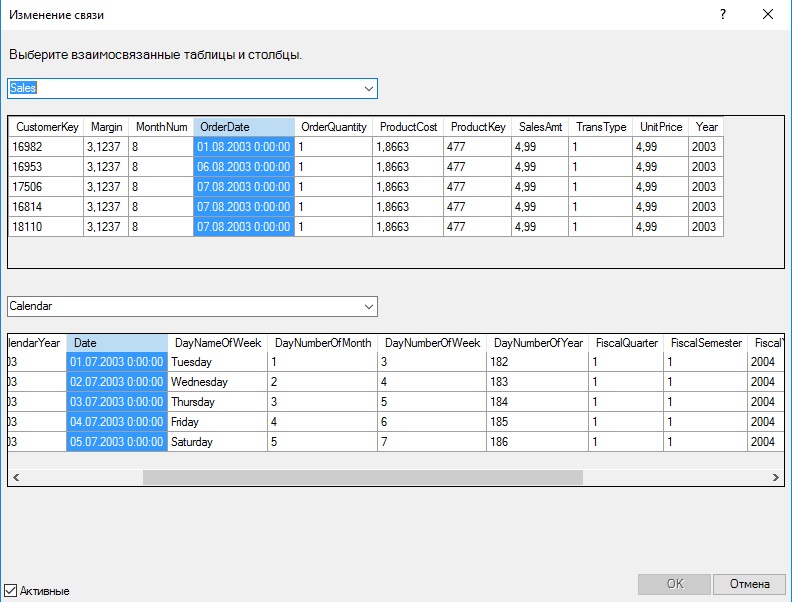


Рис. 14.2. Установлена связь таблицы Sales с Calendar

Желательно (но не обязательно), чтобы столбец OrderDate в таблице Sales имел тип «дата». При этом обязательно, чтобы в таблице Calendar столбец Date имел тип данных «дата».

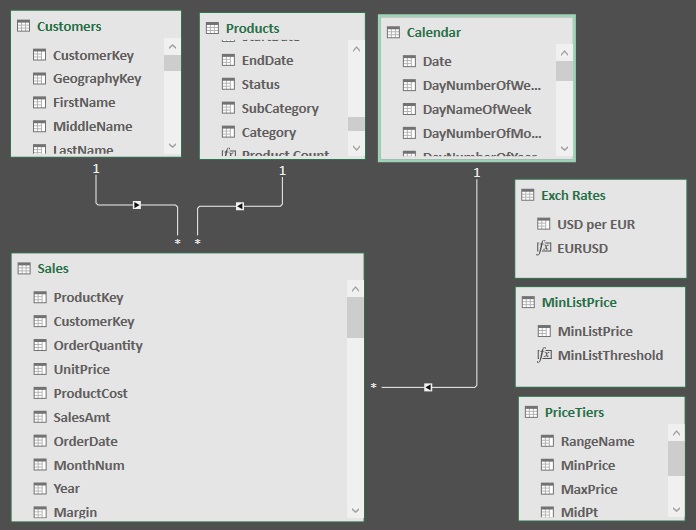


Рис. 14.3. Обновленный вид диаграммы связей; календарь становится третьей таблицей поиска (подстановок)

Календарь работает, как обычная таблица поиска:



Рис. 14.4. Сводная таблица продаж по дням недели

Но есть и особенности. Во-первых, чтобы фильтрация данных с помощью таблицы Calendar была корректной, активируйте таблицу в Power Pivot, перейдите на вкладку *Конструктор* и кликните *Пометить как таблицу дат*. После этого в сводной таблице в Excel при выборе фильтра, вы получите специальные параметры фильтрации дат:

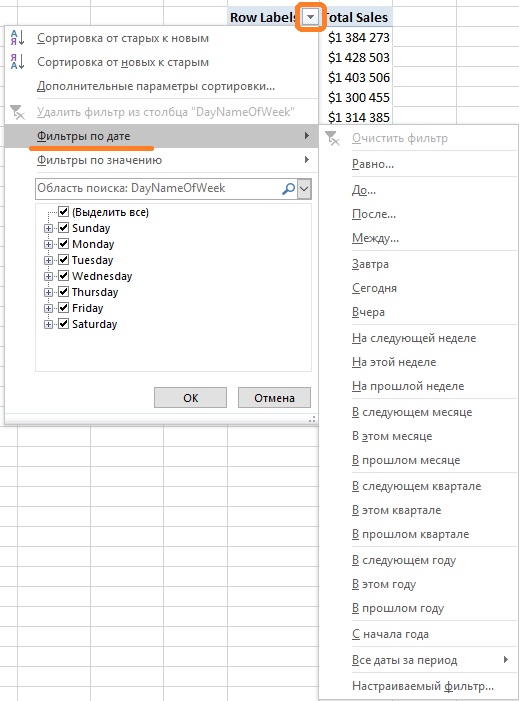


Рис. 14.5. Power Pivot сообщил Excel, что это таблица дат, поэтому Excel контекстно предложил «правильные» фильтры

Если вы собираетесь использовать столбец иного типа данных (не дата) в таблице данных (например, Sales) для связи с Calendar, всё равно, необходимо пометить его как столбец дат, иначе многие функции, рассмотренные ниже, не будут работать:

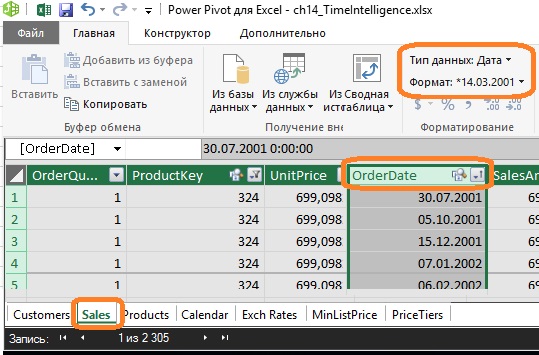


Рис. 14.6. Столбцу OrderDate в таблице Sales присвоен тип данных *Дата*

Вторая особенность таблицы Calendar. Power Pivot предлагает множество функций, связанных со временем:



Рис. 14.7. Функции DAX, относящихся к дате и времени; некоторые из них совпадают с обычными Excel-евскими, но большинство – являются новыми

Остановимся для начала на DATESYTD(). Исходная сводная таблица:

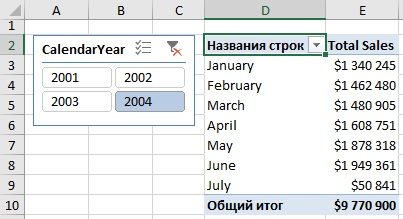


Рис. 14.8. Продажи по месяцам за 2004 г.

Добавим новую меру, которая подсчитывает продажи с начала года до сегодняшней даты:

[Total Sales YTD] =CALCULATE ([Total Sales]; DATESYTD(Calendar[Date]))

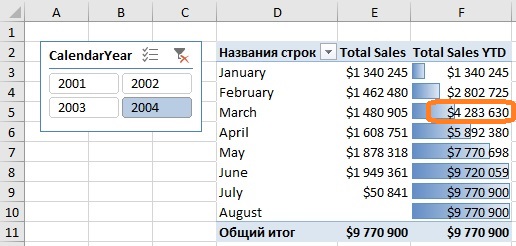


Рис. 14.9. Новая мера показывает продажи нарастающим итогом

Как это работает? Как и обычно мера вычисляется отдельно для каждой ячейки сводной таблицы. Например, для выделенного на рисунке выше значения, DATESYTD() определяет последнюю дату в текущем контексте фильтра (31 марта 2004 г.), а затем «разворачивает» контекст фильтра назад от этой даты до первой даты года (01 января 2004 г.). Затем DATESYTD() изменяет контекст фильтра: «возьмите продажи за период 01.01–31.03.2004».

Удалите [Total Sales] и перетащите год в область *Столбцы*:

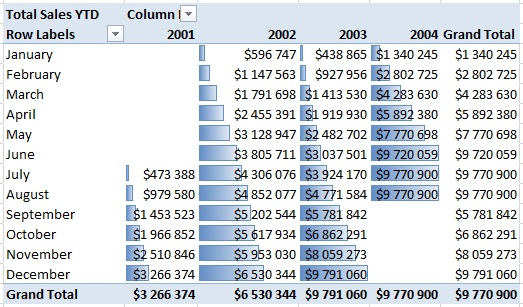


Рис. 14.10. Итоги по строкам ведут себя не совсем, как ожидалось, но, в целом, логично

### Синтаксис функции DATESYTD()

DATESYTD([Дата]; "Конец Года")

[Дата] – столбец из дат или выражений, возвращающих даты; "Конец Года" – текстовая дата, записанная в виде «01/05» («день/месяц»). Определяет дату окончания года (по умолчанию 31 декабря). Необязательный параметр. Он позволяет, например, рассматривать 30 июня как последний день года, что иногда встречается в финансовых календарях:

[Total Sales Fiscal YTD] =

CALCULATE (

[Total Sales];

DATESYTD(Calendar[Date]; ”06/30/2004” )

)

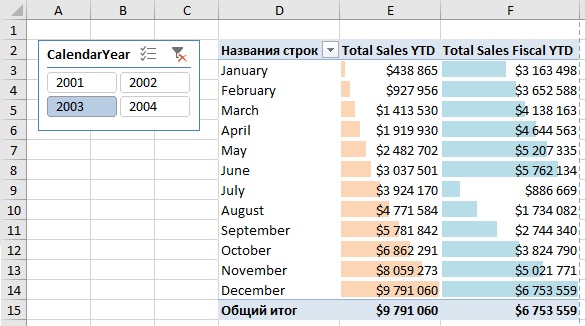


Рис. 14.11. Мера [Total Sales YTD] с начала года начинается с 0, а [Total Sales Fiscal YTD] уже учла продажи за 1.07–31.12.2002

Переопределив конец года во втором аргументе функции DATESYTD(), мы переопределили и начало года, сделав его равным 01.07.2002.

### DATESMTD() и DATESQTD()

Эти функции являются версиями DATESYTD(), которые возвращают набор дат от начала месяца/начала квартала до текущей даты. Правда, у них нет второго аргумента.

### TOTALYTD()

TOTALYTD() является заменой для CALCULATE(), если последняя основана на DATESYTD().

Например, наше первоначальная мера считалась так:

[Total Sales YTD] =CALCULATE([Total Sales]; DATESYTD(Calendar[Date]))

А могла использовать следующую формулу:

[Total Sales YTD] =TOTALYTD([Total Sales]; Calendar[Date])

Вторая форма более читабельна и короче.

Функции управления временем в Power Pivot сгруппированы в семейства. Функции из одного семейства используют похожие принципы работы. Только что мы рассмотрели семейство функций DATESYTD(). Перейдем к следующим семействам.

### FIRSTDATE() и LASTDATE()

Это версии функций MIN() и MAX() для дат. Например, определим две меры…

[FIRSTDATE Example] =FIRSTDATE(Calendar[Date])

[LASTDATE Example] =LASTDATE(Calendar[Date])

…и посмотрим на их значения в сводной таблице:

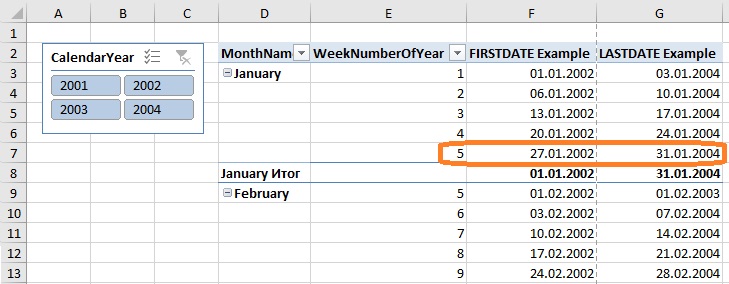


Рис. 14.12. FIRSTDATE() и LASTDATE() в действии

Меры возвращают даты, а не данные о продажах или количестве продуктов. Возможно, вам покажется странным, почему по 5-й неделе возвращены такие даты. Перейдите в Power Pivot и в закладке Calendar установите фильтры на столбцы: [MonthName]=January, [WeekNumberOfYear]=5. Вы увидите, что действительно в выбранном контексте фильтров самая ранняя дата равна 27.01.02, а самая поздняя – 31.01.04:

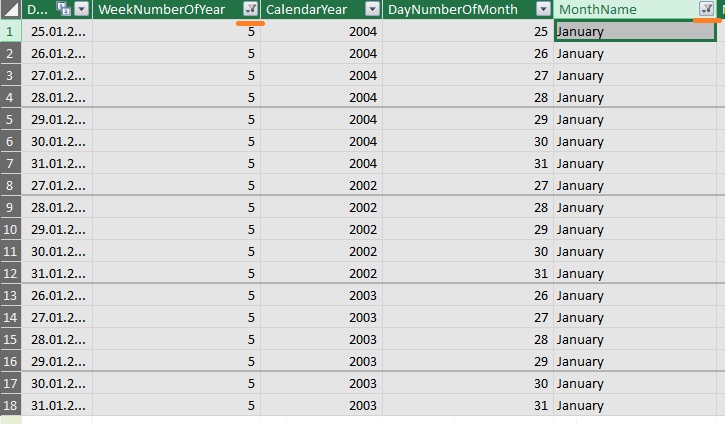


Рис. 14.13. Таблица Calendar с фильтрами столбцов [MonthName]=January, [WeekNumberOfYear]=5

### ENDOFMONTH(), STARTOFYEAR() и др.

Функции этого семейства возвращают отдельные даты, специфически обрабатывая периоды времени различного «размера».

ENDOFMONTH Measure =ENDOFMONTH(Calendar[Date])

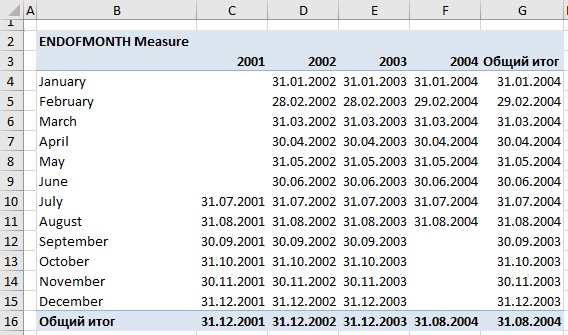


Рис. 14.14. Последняя дата месяца

Даты в итогах по строкам соответствуют вашим ожиданиям?

Поменяем месяц на квартал в строках:



Рис. 14.15. Последняя дата квартала

Поскольку последняя дата в календаре 31.08.04, не получилось отразить «правильные» даты конца 3-го и 4-го кварталов 2004 г. и в итогах по строкам.

Когда вы «вскармливаете» функции ENDOFMONTH() более чем за один месяц, она найдет последнюю дату в последнем месяце. Но когда вы предлагаете ей контекст фильтра «размером» менее месяца, вы всё равно получите последнюю дату месяца:

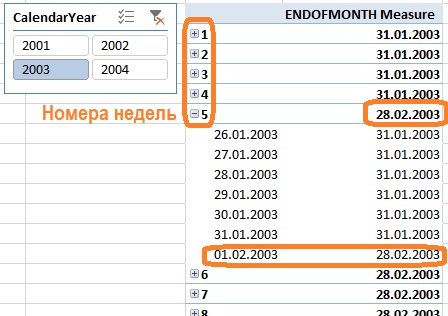


Рис. 14.16. Функция ENDOFMONTH() возвращает последний день месяца, даже если этот день не является частью текущего контекста фильтра

### CLOSINGBALANCEMONTH(), CLOSINGBALANCEYEAR() и др.

Эти функции заменяют CALCULATE(), если нужно вычислить объем продаж (или что-то иное) в соответствии с фильтрами ENDOFMONTH(), STARTOFYEAR() и т.п.

Синтаксис: CLOSINGBALANCEMONTH(<мера>; <даты>; <необязательный фильтр>)

Например:

[Total Sales CLOSINGBALANCEMONTH] =CLOSINGBALANCEMONTH([Total Sales]; Calendar[Date])



Рис. 14.17. CLOSINGBALANCEMONTH() всегда возвращает значение базовой меры в последний день месяца в текущем контексте фильтра (в примере использован объем продаж, но на самом деле эти функции более полезны с такими показателями, как запасы или баланс денежных средств)

### DATEADD()

Эта функция используется в качестве аргумента <filter> для CALCULATE() и сдвигает контекст фильтра вперед или назад во времени. Синтаксис:

DATEADD(<Date Column>; <number of intervals>; <interval type>)

<Date Column> – поместите сюда столбец дат из таблицы Calendar

<Number of Intervals> – количество интервалов; значение 1 для перемещения на один интервал к более поздним датам, и -1 для перемещения назад.

<Interval type> – тип интервала: Year, Quarter, Month или Day (без кавычек).

Например,

[Total Sales DATEADD 1 Year Back] =CALCULATE([Total Sales]; DATEADD(Calendar[Date]; -1; YEAR)

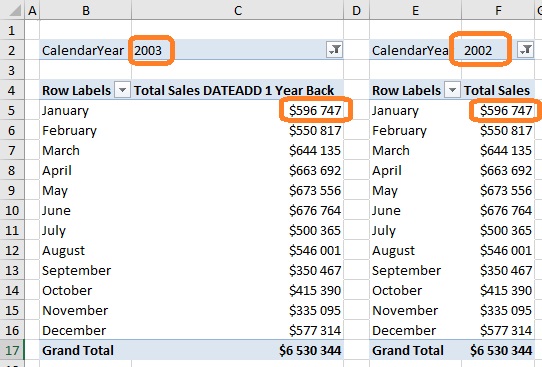


Рис. 14.18. Мера [Total Sales DATEADD 1 Year Back] за 2003 год возвращает те же значения, что и [Total Sales] за 2002 год

### Рост по сравнению с прошлым годом (год к году) и т.п.

Одним из очевидных применений DATEADD() и аналогичных функций является расчет роста по сравнению с предыдущим годом (Year-Over-Year, YOY).

[Pct Sales Growth YOY] =

([Total Sales] - [Total Sales DATEADD 1 Year Back]) / [Total Sales DATEADD 1 Year Back]

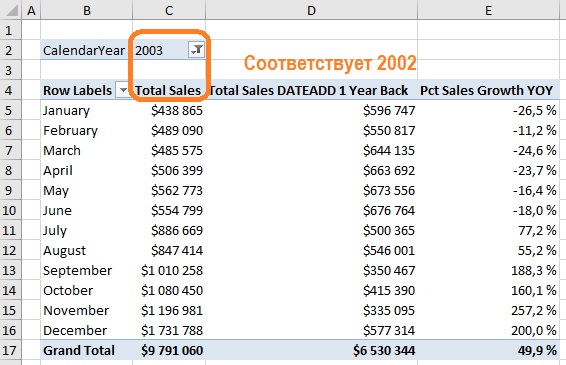


Рис. 14.19. Динамика роста продаж в 2003 году и по сравнению с 2002 годом

Предостережение. Использовать DATEADD() вам следует только на непрерывных диапазонах дат сводной таблицы. Посмотрите, что произойдет, если вы отфильтруете один или несколько месяцев:

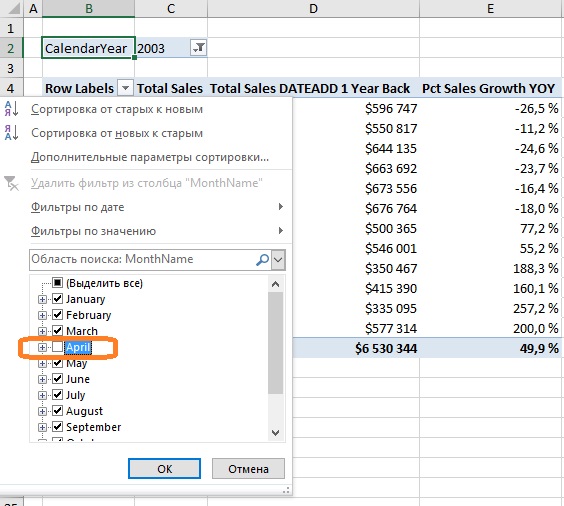


Рис. 14.20. Попытка исключить Апрель из строк сводной таблицы…

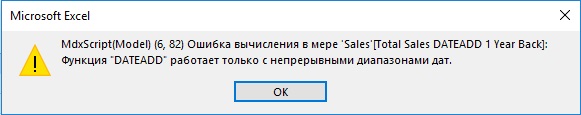


Рис. 14.21. … приводит к ошибке

Если вам всё же нужны прерывистые диапазоны, используйте функцию IF().

DATEADD() корректно обрабатывает високосные года. Умные функции управления датами «знают», сколько дней в каждом месяце, даже високосном. Создадим простую меру, чтобы показать число дней в месяце в сводной таблице: [Number of Days] =COUNTROWS(Calendar):

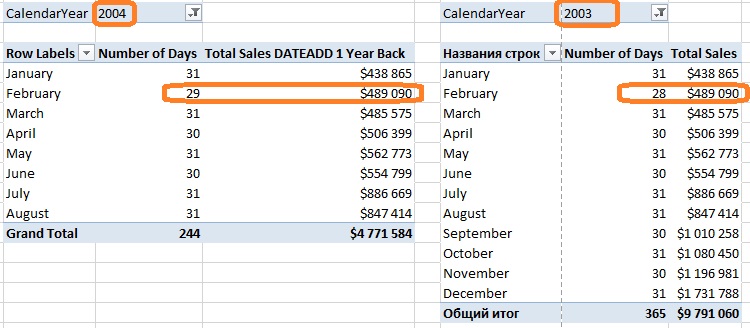


Рис. 14.22. DATEADD() возвращает в левой сводной продажи за 28 дней в феврале прошлого 2003 года, несмотря на то, что в феврале 2004 года 29 дней

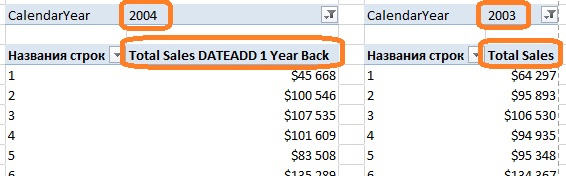


Рис.14.23. А вот по неделям DATEADD() работает некорректно

Чтобы понять, почему, давайте добавим дни недели:

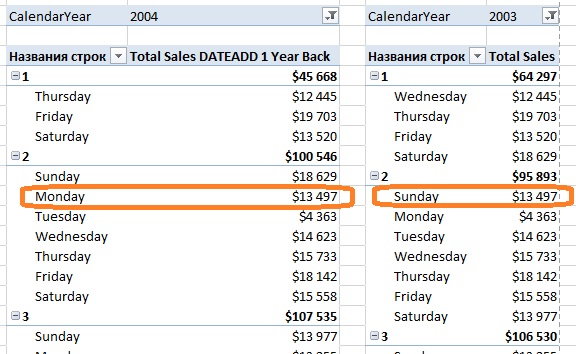


Рис. 14.24. Обе сводных возвращают продажи с воскресенья по понедельник (неделя на американский манер), но мера DATEADD() дает воскресные продажи 2003 года в контексте понедельника 2004 года

Иначе говоря, недели смещены на один день.

В то время как функции управления временем «знают», что мы подразумеваем под месяцем/кварталом/годом, они полагаются на таблицу Calendar для всех других понятий, поэтому при навигации с помощью DATEADD() в контексте фильтра по неделям, нет никакого «волшебного исправления».

### SAMEPERIODLASTYEAR()

Функция SAMEPERIODLASTYEAR(<Date Column>) возвращает тот же период прошлого года. Это, в некотором роде, избыточная функция, которая возвращает то же значение, что и функция DATEADD() с аргументами "-1; Year":

SAMEPERIODLASTYEAR(Calendar[Date])

DATEADD (Calendar[Date]; -1; YEAR )

### PARALLELPERIOD(), NEXTMONTH(), PREVIOUSYEARO и др.

Синтаксис: PARALLELPERIOD(<Date Column>; <number of intervals>; <interval type>)

Покажем работу функции на примере. Создадим меру:

[Total Sales PARALLELPERIOD Back 1 Year] =

CALCULATE(

[Total Sales];

PARALLELPERIOD(Calendar[Date]; -1; YEAR)

)

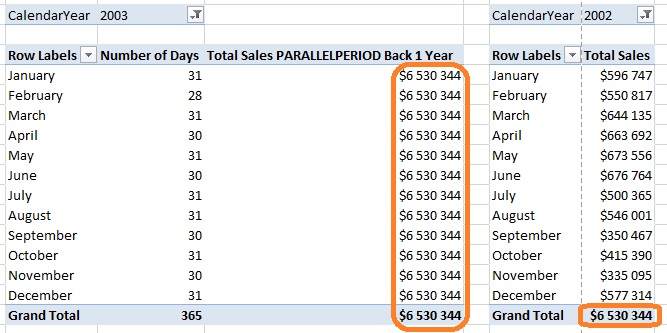


Рис. 14.25. PARALLELPERIOD() всегда выбирает полный год (если третий аргумент Year), независимо от того, каков контекст фильтра (в нашем примере – месяц)

Таким образом, функция PARALLELPERIOD() перемещается так же, как DATEADD(), но когда она достигает своего «назначения», он расширяет контекст фильтра до размера указанного <типа интервала> – года, квартала или месяца.

NEXTMONTH() и PREVIOUSYEAR() являются некоторым упрощением PARALLELPERIOD(). Они содержат только один аргумент, так как направление перемещения и интервал перемещения определяются самой функцией. Например, мера…

[Total Sales NEXTMONTH] =CALCULATE([Total Sales]; NEXTMONTH(Calendar[Date]))

… возвращает продажи следующего месяца:

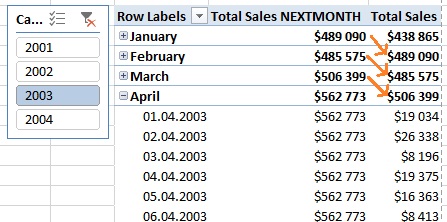


Рис. 14.26. NEXTMONTH() всегда захватывает полный следующий месяц, даже если наш контекст – один день

### DATESBETWEEN()

Функция возвращает даты между начальной и конечной. Синтаксис:

DATESBETWEEN(<date column>; <start date expr>; <end date expr>)

Например:

[Total Sales First Half 2003] =

CALCULATE([Total Sales];

DATESBETWEEN(Calendar[Date]; "1/1/2003"; "6/30/2003")

)

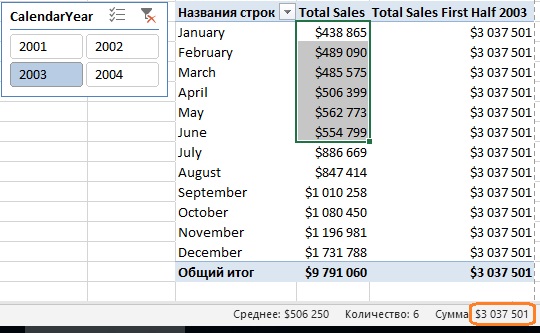


Рис. 14.27. DATESBETWEEN() полностью переопределяет существующий контекст фильтра в таблице Calendar

### Расчеты на текущий день

Функция DATESBETWEEN() позволяет использовать формулы в аргументах <начальная дата> и <конечная дата>:

[Total Sales Life to Date] =

CALCULATE(

[Total Sales];

DATESBETWEEN(

Calendar[Date];

”1/1/1900”, LASTDATE(Calendar[Date])

)

)

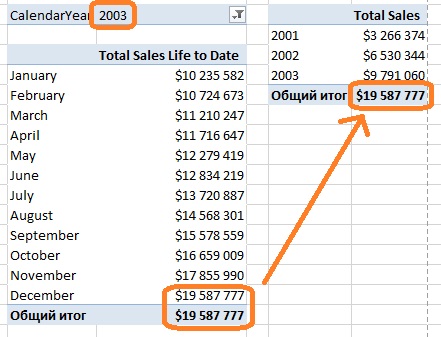


Рис. 14.28. Контекст фильтра с использованием DATESBETWEEN(), как и ожидалось, соответствует общему итогу за 2001–2003 годы

Можно усовершенствовать формулу меры и убрать фиксированное начальное значение ”1/1/1900”:

[Total Sales Life to Date] =

CALCULATE (

[Total Sales];

DATESBETWEEN(

Calendar[Date];

FIRSTDATE(ALL(Calendar[Date]));

LASTDATE(Calendar[Date])

)

)

Почему ALL(Calendar[Date])? Потому что FIRSTDATE(Calendar[Date]) вернет первую дату в контексте фильтра (в примере на рис. 14.28 – 1 января 2003 года). Нам нужно применить ALL(), чтобы очистить текущий контекст фильтра и найти первую дату во всей таблице Calendar.

Мы не используем ALL() в третьем аргументе внутри функции LASTDATE(). Здесь наоборот нам нужна не последняя дата всего календаря, а текущая дата в контексте фильтра.

### DATESBETWEEN() в несвязанной таблице

[Ранее](http://baguzin.ru/wp/?p=19926) мы рассмотрели пороговые значения в несвязанных таблицах. То же самое можно сделать с датами, используя несвязанную таблицу диапазонов дат, обычную таблицу Calendar и функцию DATESBETWEEN(), использующую в качестве аргументов меры из несвязанной таблицы. [Здесь](https://powerpivotpro.com/2011/04/ab-campaign-analysis-with-start-end-date-slicers/) можно найти подробный пример на эту тему.