Моделирование данных в Power BI

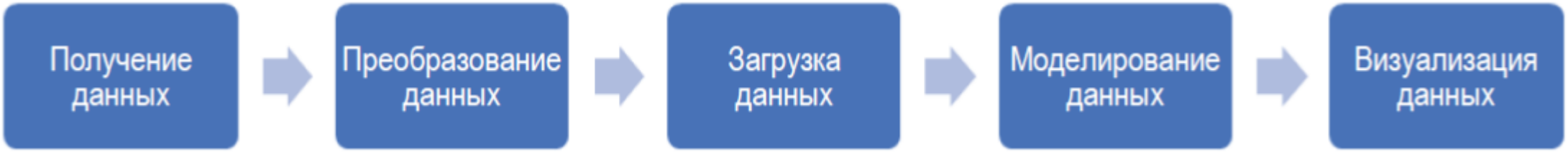
к.э.н., доцент Кучмезов Х.Х.

Power BI представляет собой целую экосистему, позволяющую пользователям вносить собственный вклад в аналитическую политику организации путем обмена наборами данных, отчетами и дашбордами, а также размещения в отчетах комментариев с мобильных устройств и их рассылки конкретным пользователям. Но все это возможно только при правильной настройке экосистемы Power BI. Даже самый красивый в мире отчет ровным счетом ничего не будет стоить, если он показывает неправильные цифры или на его формирование уходит много времени. Пользователи никогда не будут работать с таким отчетом.

В реальных проектах вам зачастую приходится получать данные из различных источников. Но получение данных и их внедрение в систему – это только полдела. Самое главное – объединить эти данные в модель, позволяющую гарантировать целостность исходных сведений и их связь с бизнес-логикой

Понятие слоев в Power BI

Этапы формирование нового отчета в Power BI



Говоря о моделировании данных в Power BI, мы фактически ссылаемся на программный продукт *Power BI Desktop*. Вы можете рассматривать Power BI Desktop как своеобразный аналог *Visual Studio* при разработке *табличной модели* (Tabular model) в *SQL Server Analysis Services* (SSAS).

Для осуществления описанного выше «Этапов формирования нового отчета» мы используем различные *концептуальные слои* (conceptual layer) в Power BI.

В Power BI Desktop эти слои отражены так, как показано на рис. 1.1.

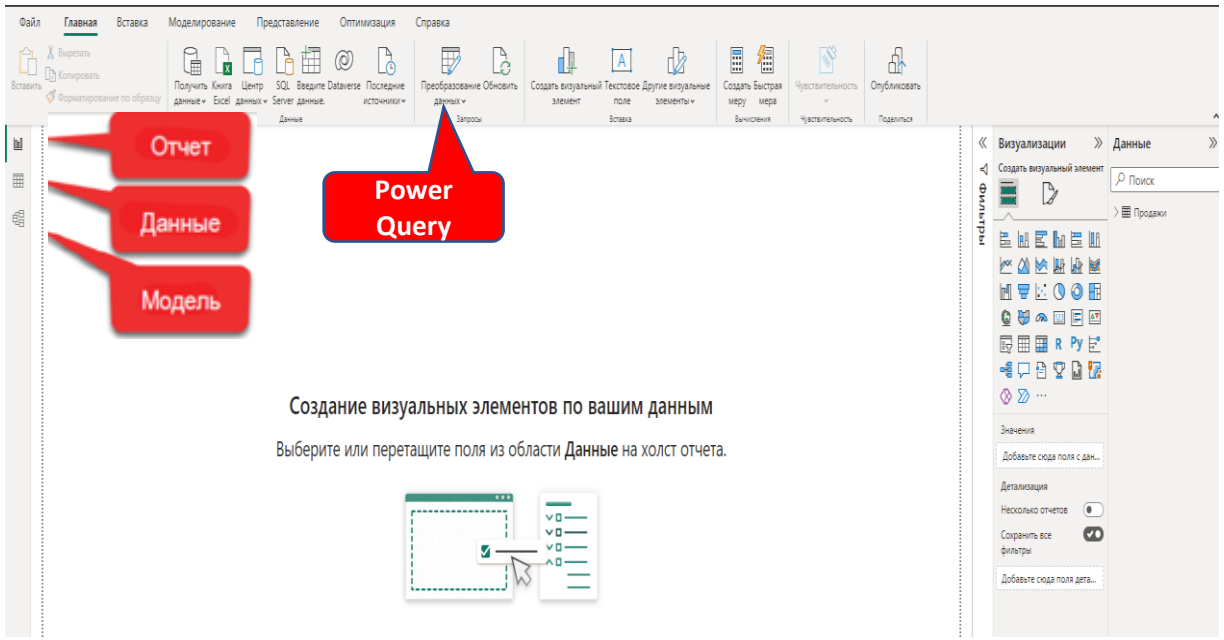
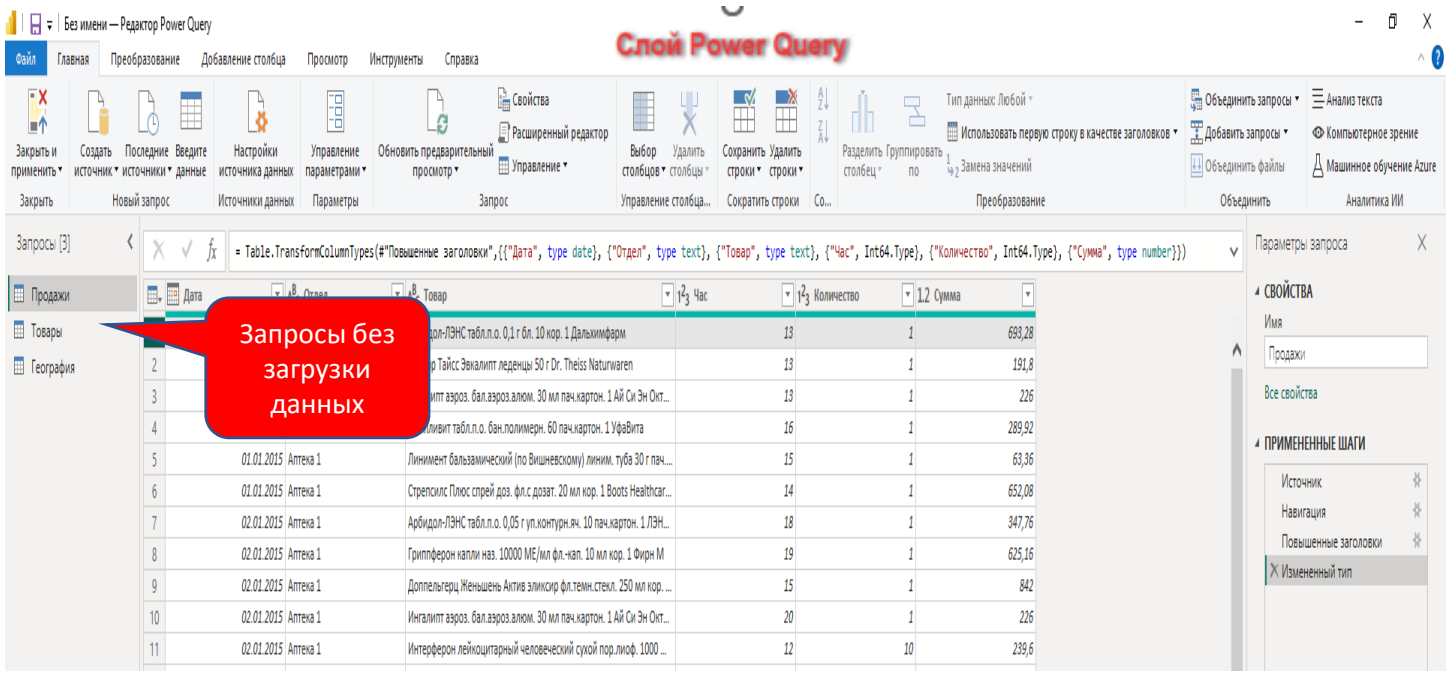


Рис.1.1. Слои Power BI

Слой подготовки данных (Power Query)

На этом слое мы получаем исходные данные из различных источников, преобразовываем, очищаем их и делаем доступными для других слоев. Это первый слой обработки данных, так что он играет важную роль в вашем путешествии по миру Power BI.

В слое Power Query вы определяете, какие запросы будут служить для загрузки данных в вашу модель данных, а какие – выполнять исключительно служебные задачи по трансформации и очистке информации без загрузки в модель

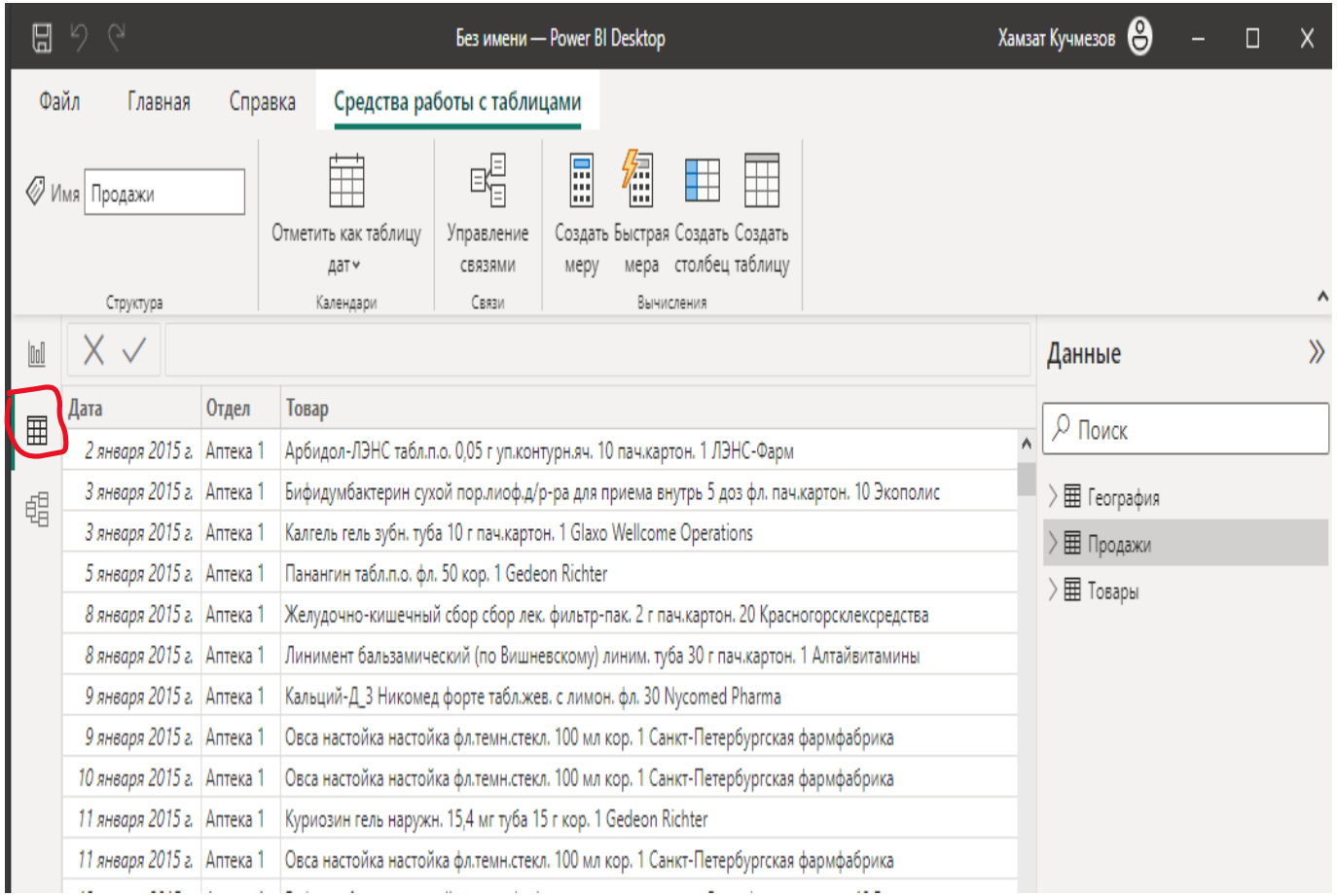


Слой модели данных

Этот слой включает в себя два представления: **Данные** (Data view) и **Модель** (Model view). В первом из них вы можете работать с исходными данными, во втором – с целыми моделями.

Вкладка Данные

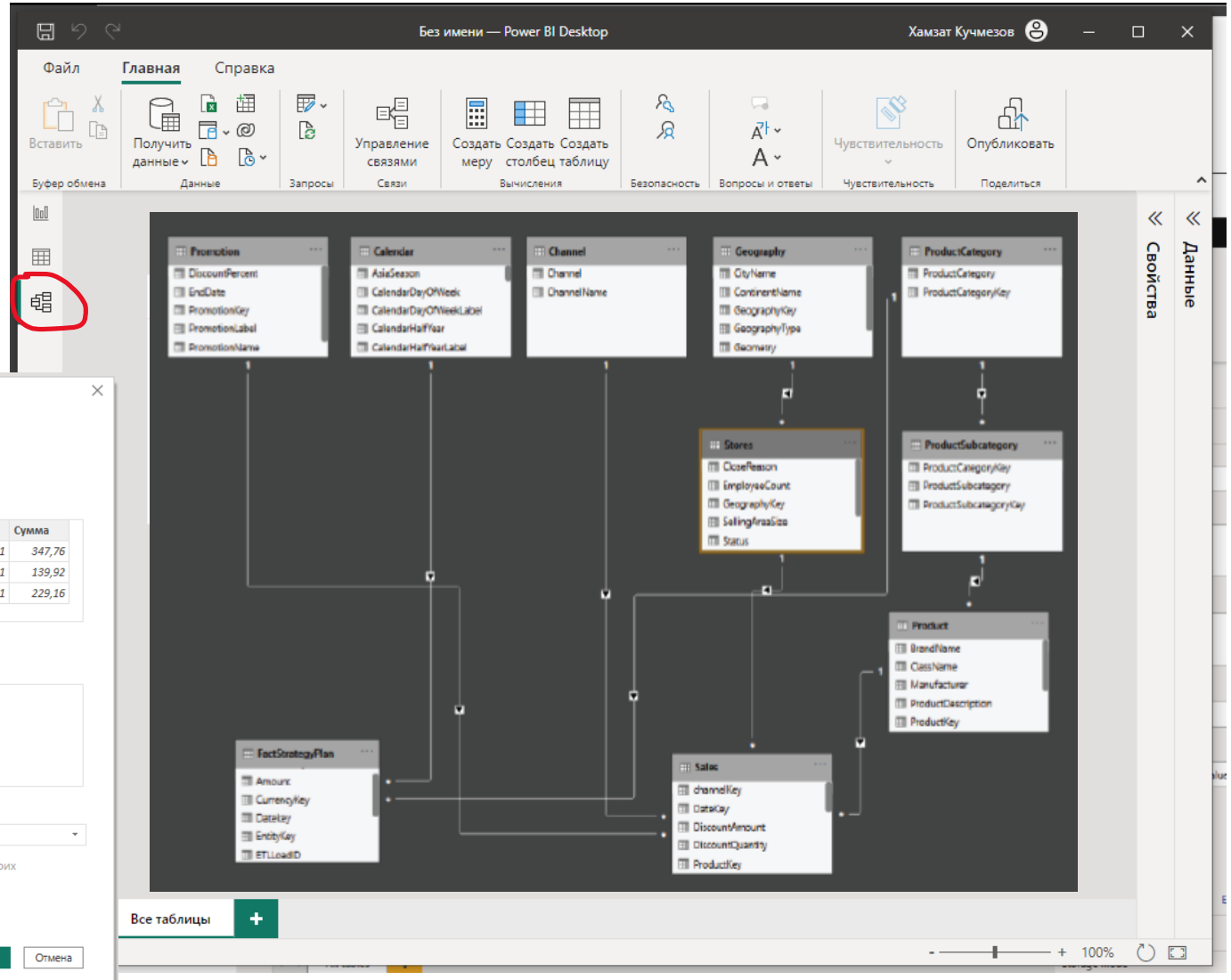
После окончания работы с данными в слое Power Query происходит их загрузка в слой модели данных. На *вкладке с данными* мы видим исходные сведения в том виде, в котором они поступили в модель после их преобразования и очистки. В зависимости от типа подключения эти исходные данные могут быть доступны или нет. Помимо просмотра данных в этой вкладке, мы также можем выполнять сопутствующие действия над ними, создавая объекты аналитики, такие как вычисляемые таблицы, вычисляемые столбцы и меры, и копируя данные из таблиц.

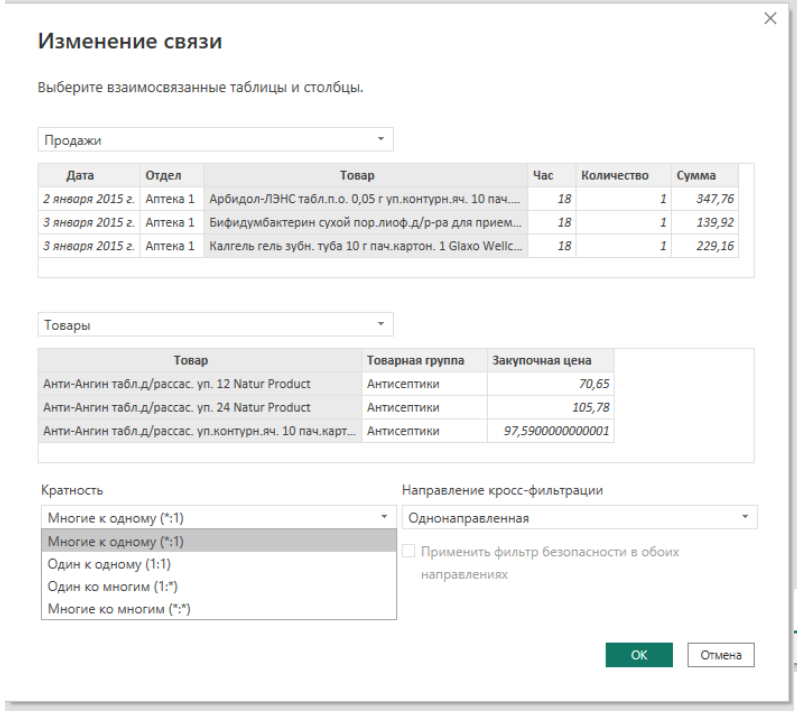


ПРИМЕЧАНИЕ *Все объекты, создаваемые при помощи языка DAX, становятся частью нашей модели данных.*

Вкладка Модель данных

Как ясно из названия, на *этой вкладке* мы сводим все наши исходные данные воедино. При этом мы не только видим, какие таблицы у нас есть и как именно они объединены между собой, но также можем создавать новые связи, форматировать поля и синонимы, показывать/скрывать элементы и т. д.,



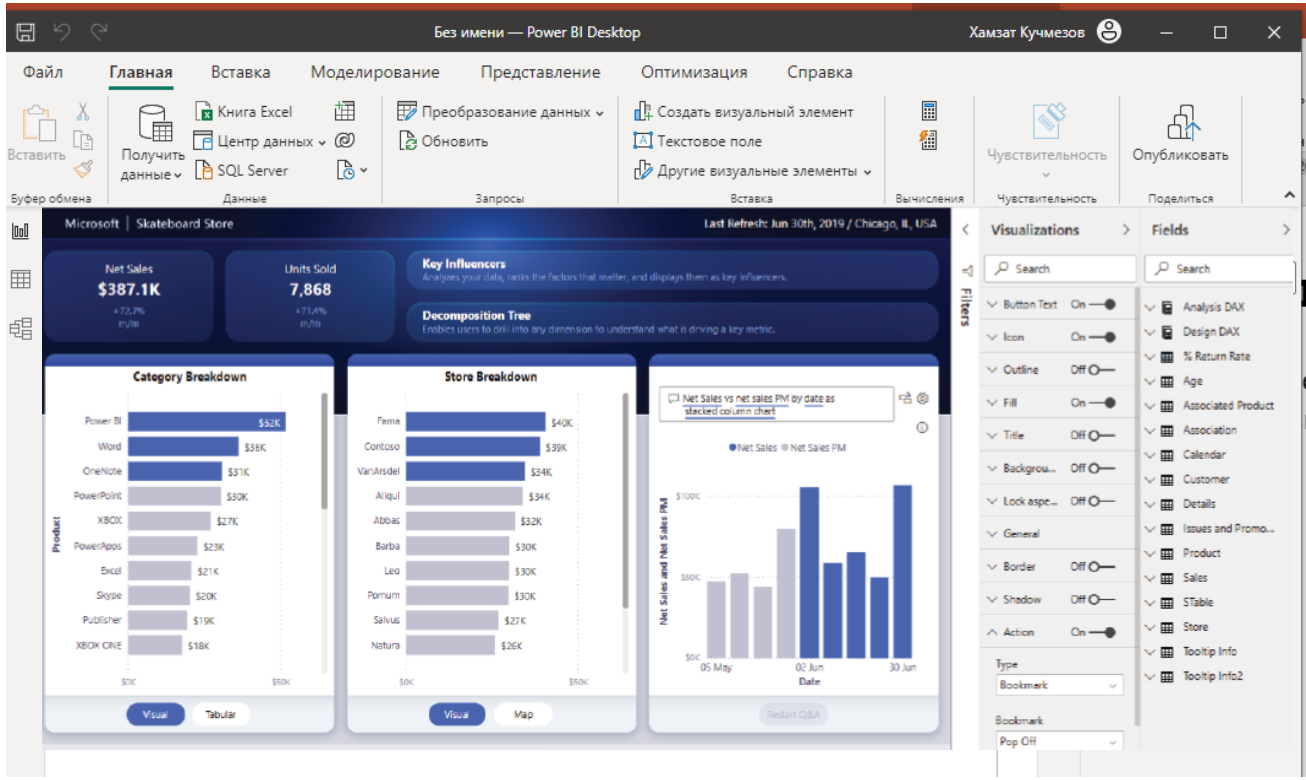


Слой визуализации данных

В этом слое мы возвращаем наши исходные данные к жизни, создавая наполненные смыслом визуализации. Доступ к этому слою осуществляется при помощи вкладки **Отчет** (Report), которая в Power BI Desktop открывается по умолчанию.

Вкладка Отчет

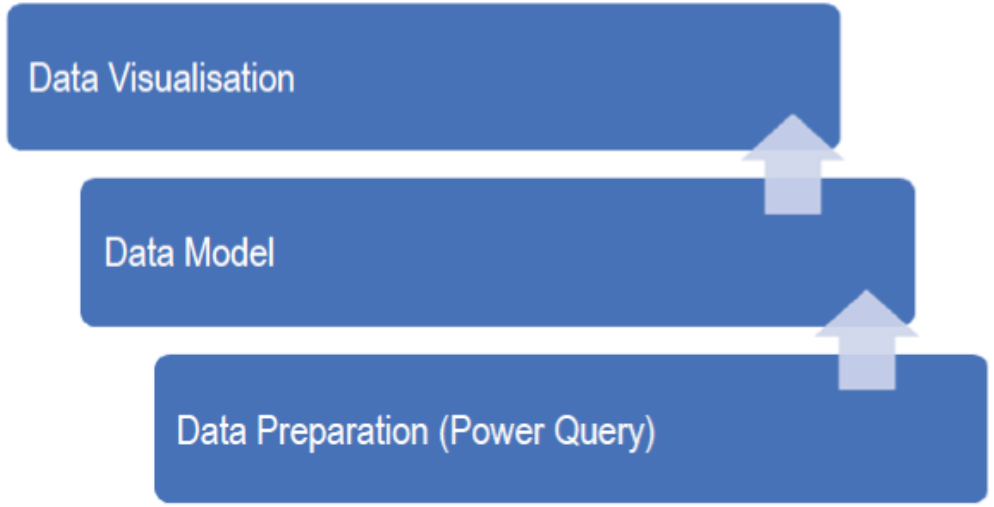
На *вкладке* **Отчет** (Report) мы можем строить визуализации разной степени сложности, помогающие бизнесу принимать решения на основании имеющихся данных. Еще здесь можно создавать аналитические вычисления с использованием языка DAX, такие как вычисляемые таблицы и столбцы, а также меры. Но это не значит, что эти вычисляемые объекты становятся частью слоя визуализации. **Фактически они принадлежат слою модели данных.**



Поток данных в Power BI

Осознание того, как данные перемещаются внутри Power BI, очень важно в плане понимания общей картины происходящего. К примеру, если вы видите, что с каким-то вычислением в отчете возникла проблема, вы должны быстро уметь разобраться в причинах и добраться до уровня, на котором эту проблему стоит решать. Допустим, если вы видите, что на графике выводятся неправильные цифры, и знаете, что этот график использует для расчетов меры, базирующейся на вычисляемом столбце, то должны понимать, что не стоит искать этот столбец в слое Power Query, поскольку объекты, создаваемые в модели данных, недоступны в Power Query. Также вы никогда не станете искать меру в слое подготовки данных или пользовательскую функцию в слое модели данных.

Поток данных в Power BI



Моделирование данных в Power BI

Моделирование данных (data modeling) является одной из важнейших составляющих процесса разработки в Power BI. Цель моделирования данных в Power BI существенно отличается от создания моделей в транзакционных системах. В последнем случае моделирование направлено на оптимизацию процесса фиксирования транзакций. В то же время хорошо спроектированная модель данных в Power BI служит целям оптимизации выполнения запросов к данным и снижения размера результирующих наборов за счет агрегирования данных.

Далеко не у всех есть доступ **к готовому хранилищу данных**, так что зачастую нам приходится проектировать модель данных непосредственно в Power BI. При этом многим хочется просто взять все имеющиеся данные из источников и импортировать их в Power BI. Но в этом случае формирование запросов к модели будет занимать достаточно много времени, что в условиях бизнеса **неприемлемо**. Таким образом, рекомендуется отказаться от соблазна загрузки всех доступных данных в модель, а решать проблемы по мере их поступления. В идеале ваша модель данных должна включать в себя все элементы, достаточные и необходимые для того, чтобы отвечать на требования бизнеса в максимально короткие сроки.

Моделируя данные в Power BI, вы должны делать это в строгом соответствии с имеющейся бизнес-логикой. Для этого вам может понадобиться объединить некоторые таблицы и до определенной степени агрегировать исходные данные. Но это бывает проблематично, когда данные из различных источников, объединенные общей логикой, имеют разную гранулярность.

В связи с этим перед загрузкой данных в Power BI их бывает полезно преобразовать, и лучше других с этой задачей может справиться Power Query. После очистки данных мы получим удобную и лаконичную модель данных, работать с которой будет очень просто.

Семантическая модель

Истоки Power BI восходят к моделям ***Power Pivot*** и ***SSAS*** **(SQL Server Analysis Services)** Tabular. Все они используют в своей основе движок ***xVelocity***, представляющий собой обновленную версию движка *VertiPaq*. Он был разработан для обработки данных в оперативной памяти и содержит объекты *семантической модели* (semantic model), такие как **таблицы, связи, иерархии и меры, хранящиеся в памяти с применением *колоночной индексации* (column store indexing).** В этой связи вы могли бы ожидать значительного прироста производительности по сравнению с сильно сжатыми данными, не так ли?

Но здесь есть свои нюансы. Ваши отчеты будут отличаться высокой скоростью и производительностью только при условии, что вы эффективно преобразовали данные для поддержки бизнес-логики. Путем загрузки данных в модель данных Power BI вы строите семантическую модель, содержащую всю заложенную в информацию логику. Это унифицированная модель, предлагающая бизнес-контексты для ваших данных.

К семантической модели можно осуществлять доступ из разных инструментов визуализации без необходимости повторно преобразовывать данные. Таким образом, после публикации отчета в *службе Power BI* (Power BI service) вы можете анализировать набор данных при помощи Excel или использовать сторонние инструменты, такие как Tableau, для подключения к набору данных Power BI при наличии лицензии Premium и их визуализации.

Построение эффективной модели данных в Power BI

Эффективная модель данных способна с минимальными временными затратами отвечать на все интересующие вас вопросы, а также она проста для понимания и поддержки.

Давайте разберемся, что это значит. Ваша модель должна:

- быстро реагировать и выполнять вычисления;
- быть построена с учетом существующих бизнес-требований;
- обладать минимально возможным уровнем сложности (быть легкой для понимания);
- обеспечивать необходимую поддержку с минимальными затратами. Рассмотрим озвученные требования применительно к реальному сценарию.

Пример: Вам поставили задачу создать отчет на базе трех следующих источников данных:

- источник данных *OData* из 15 таблиц, каждая из которых содержит 50 до 250 столбцов;
- файл Excel с 20 зависящими друг от друга рабочими листами с множеством формул;
- хранилище данных в SQL Server, в котором вас интересуют пять измерений и две таблицы фактов:
- из этих измерений одно содержит даты, второе – время. Гранулярность измерения времени исчисляется часами и минутами;
- в таблицах фактов содержится от 50 до 200 млн строк. Гранулярность обеих таблиц фактов в отношении даты и времени исчисляется днями, часами и минутами;

Уже по одному описанию сценария к представленным источникам данных могут возникать серьезные вопросы

Многомерное моделирование (схемы: звезда и снежинка)

Сразу хочется отметить, что термины схема «звезда» (star schema) и многомерное моделирование (dimensional modeling) относятся к одному и тому же.
Применительно к Power BI термин схема «звезда» употребляется чаще,

Транзакционные модели против схемы «звезда»

В *транзакционных системах* (transactional system) главной целью является повышение производительности при создании новых записей и редактировании/удалении существующих. Таким образом, при проектировании транзакционных систем очень важно провести процесс *нормализации* (normalization) данных с целью снижения избыточности данных и повышения производительности ввода информации. Обычно при нормализации мы разбиваем все таблицы на главные и подчиненные.

В то же время перед системами бизнес-аналитики стоит совершенно иная задача. Здесь на первый план выходит эффективность запросов к данным, и именно с этим расчетом выполняется оптимизация модели данных.

В схеме «звезда» все нужные нам объединения таблиц уже произведены на основании бизнес-требований. Данные агрегированы и загружены в *денормализованные* (denormalized) таблицы. В описанном выше сценарии руководство компании не интересуют продажи с детализацией до секунды. Таким образом, мы можем агрегировать данные по дням, что позволит уменьшить объем представленной информации с полутора миллиардов до нескольких тысяч строк за интересующий нас период в полгода. Вряд ли нужно объяснять, что на таком объеме данных операция суммирования будет выполняться куда быстрее.

Идея схемы «звезда» состоит в разделении данных на числовые, хранящиеся в *таблицах фактов* (fact table), и описательные, которые размещаются в *таблицах измерений* (dimension table).

Идея схемы «звезда» состоит в разделении данных на числовые, хранящиеся в ***таблицах фактов*** (fact table), и описательные, которые размещаются в *таблицах измерений* (dimension table).

Обычно на схеме таблицы фактов располагаются по центру – в окружении **измерений**, описывающих эти факты. При взгляде на такую схему невольно возникает ассоциация со звездой, что видно по рисунке справа. Именно отсюда и произошло такое ее название.

На рисунке показана таблица фактов ***Internet Sales*** в окружении измерений на схеме «звезда».

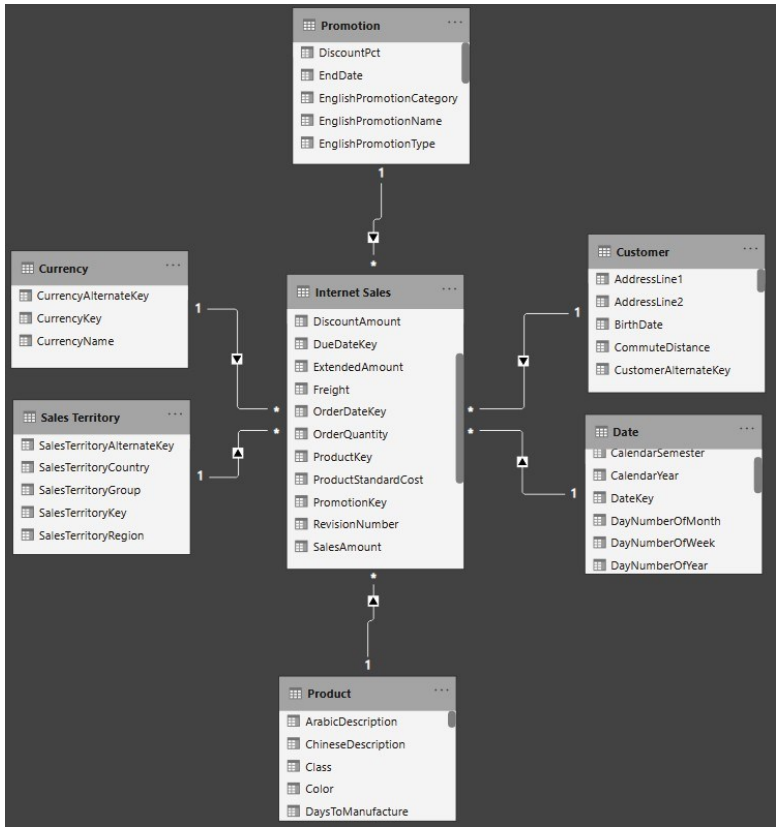
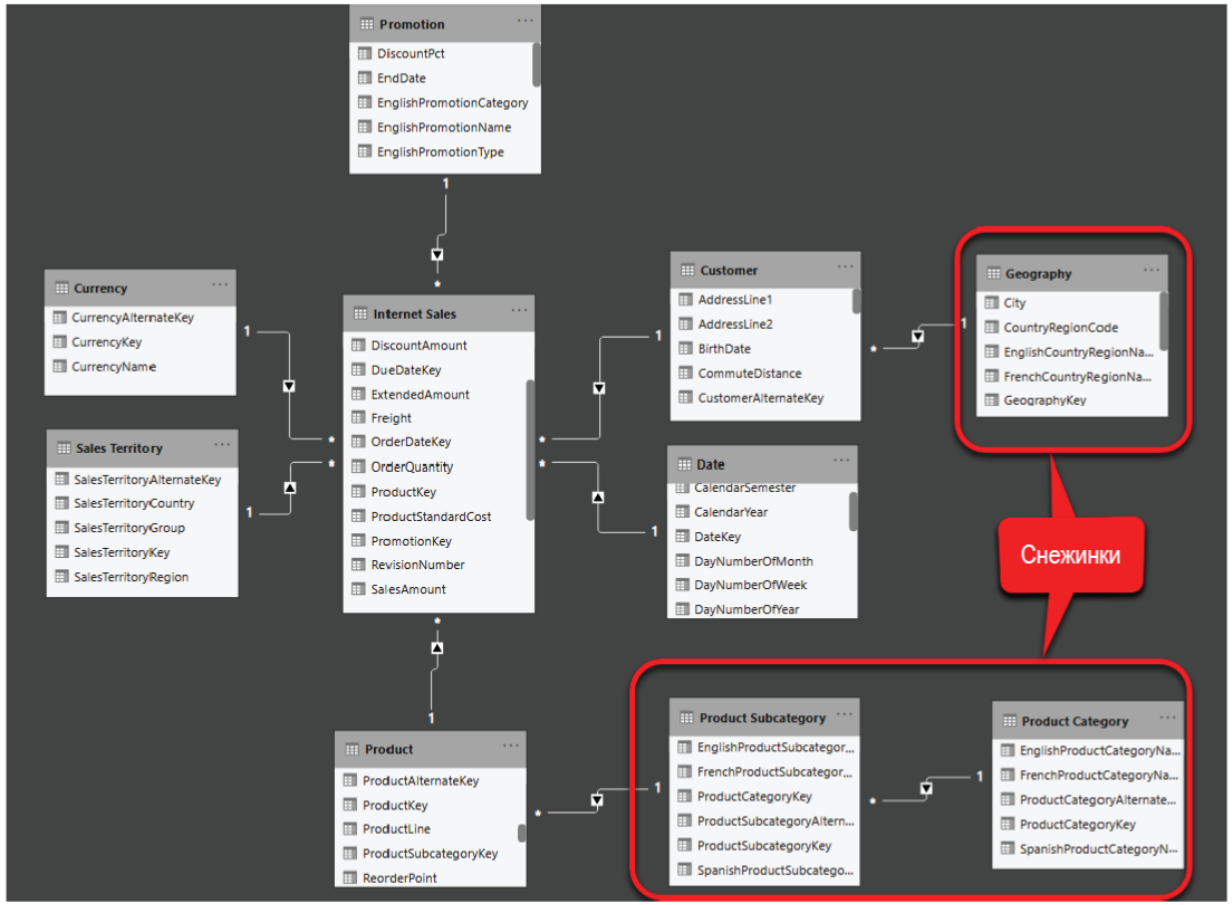


Схема «снежинка»

***Схема «снежинка»* (snowflake)** образуется тогда, когда при окружении таблиц фактов измерениями не получается идеальная звезда. Зачастую бывает, что описательная информация хранится в нескольких таблицах – в виде уровней. В таких ситуациях традиционные измерения оказываются объединены связями с другими измерениями, в которых находится более детализированная информация. По сути, «снежинка» представляет собой процесс нормализации таблиц измерений.

Бывают случаи, когда образование такой схемы неизбежно, **но в общем случае при моделировании данных в Power BI следует любыми способами избегать образования схемы «снежинка».**



Группы вычислений

Группы вычислений (calculation groups) аналогичны вычисляемым элементам (calculated member) в MDX. Изначально группы вычислений были представлены в табличных моделях SSAS 2019. Также они доступны в службах Azure Analysis Services и Power BI.
Зачастую разработчикам Power BI приходится создавать некоторые базовые меры, после чего на их основе плодить большое количество мер, производящих идентичные вычисления с использованием логики операций со временем. Так например в примере про моделирование данных были использованы следующие меры:

- Product cost: SUM('Internet Sales'[TotalProductCost]);
- Order quantity: SUM('Internet Sales'[OrderQuantity]);
- Internet sales: SUM('Internet Sales'[SalesAmount])

В то же время бизнес требует создания следующих расчетов с использованием логики операций со временем для каждой из перечисленных мер:

- ✓ накопительная сумма с начала года (Year to date);
- ✓ накопительная сумма с начала квартала (Quarter to date);
- ✓ накопительная сумма с начала месяца (Month to date);
- ✓ накопительная сумма с начала года в предыдущем периоде (Last year to date);
- ✓ предыдущая накопительная сумма с начала квартала в предыдущем периоде (Last quarter to date);
- ✓ предыдущая накопительная сумма с начала месяца в предыдущем периоде (Last month to date);
- ✓ сравнение годов (Year over year);
- ✓ сравнение кварталов (Quarter over quarter);
- ✓ сравнение месяцев (Month over month).

Интерактивных подход к моделированию данных

Как и в случае с разработкой любого программного обеспечения, моделирование данных представляет собой непрерывный процесс. Вы начинаете **с переговоров с руководством**, после чего реализуете определенную бизнес-логику в модели данных. Далее мы продолжаем разработку решения в Power BI.

Часто после построения визуальных элементов вы понимаете, что возможно добиться лучших результатов, если внести определенные изменения в модель данных. Да и реализованная в модели бизнес-логика нередко не соответствует тому, что на самом деле нужно бизнесу.

После осуществления первых нескольких итераций довольно часто от руководство можно услышать следующую фразу

Все выглядит прекрасно, но это не то, что мне нужно!

Именно поэтому при разработке сценариев в Power BI лучше всего применять *пошаговый динамический подход*.

Добавочная загрузка данных

Одной из самых полезных возможностей в Power BI является настройка *добавочной* (*инкрементной*) *загрузки данных* (incremental data load). Эта операция была унаследована Power BI от SSAS для работы с объемными моделями данных. При правильной настройке этого параметра Power BI не будет каждый раз импортировать данные с нуля. Вместо этого будет производиться загрузка только измененных с момента последнего импорта данных. Такой подход позволяет значительно повысить эффективность обновления данных и минимизировать вычислительную нагрузку на сервер. Добавочная загрузка данных доступна в лицензиях **Professional и Premium.**