Содержание

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt Publishing очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Предисловие

Я очень рад, что именно Сохейл, становление которого в статусе MVP я наблюдал лично, взялся за написание книги на тему экспертного моделирования данных в Power BI. Все студенты любят учиться на конкретных примерах, тогда как пробираться через дебри сухих документаций, извлекая важные сведения, им не очень-то интересно. Многие из них на этом этапе бросают обучение, так и не найдя реальных практических примеров из жизни, объясняющих стоящие перед ними задачи.

В книге Сохейла освещен обширный диапазон тем от начального до экспертного уровня в Power BI – инструменте, занимающем лидирующее место в иерархии продуктов для бизнес-аналитики. Power BI обладает потрясающей гибкостью, и его очень легко использовать. Данная книга подойдет как для начинающих аналитиков, так и для тех, кто проектирует системы, предназначенные для обслуживания тысячей пользователей. Я бы без колебаний посоветовал книгу Сохейла любому, интересующемуся бизнес-аналитикой, вне зависимости от уровня. Он пишет о Power BI в очень захватывающей манере, начиная повествование с элементарных строительных блоков, продолжая различными аспектами моделирования данных, включая описание схемы-звезды, управления двунаправленными связями, отношений «многие ко многим», вычисляемых таблиц, подготовки данных в Power Query, и заканчивая сложными темами, охватывающими RLS, OLS и составные модели данных. Кроме того, каждому студенту должен прийтись по душе сугубо практический подход, использованный в этой книге.

Кристиан Уэйд (*Christian Wade*),
руководитель группы проектов, Microsoft

Об авторе

Сохейл Бахши (Soheil Bakhshi) является основателем сайта Data Vizioner и популярным практикующим консультантом по бизнес-аналитике. Сохейл обладает более чем 20-летним опытом работы с данными в области аналитики в Microsoft, включая работу с хранилищами данных и платформой Power BI. Обладая сертификатами MSCE и MCSA, он также может похвастаться статусом Microsoft MVP. Своими знаниями и страстью Сохейл делится на своем сайте www.biinsight.com, а также на конференциях по Power BI, проходящих по всему миру. В стремлении к простоте и эффективности Сохейл Бахши принял участие в разработке нескольких полезных инструментов, включая Power BI Exporter и Power BI Documenter.

О технических редакторах

Фелипе Вилела (Felipe Vilela) долгое время работал в области системных разработок, после чего более восьми лет назад переключился на бизнес-аналитику и хранилища данных, в основном с использованием продуктов от MicroStrategy. Фелипе работал со многими компаниями из Бразилии и США, внедряя и настраивая продукты от MicroStrategy. Параллельно он преподавал бизнес-аналитику и хранение данных на собственных курсах, а также на официальных курсах компании MicroStrategy. Фелипе ведет блог по адресу www.vilelamstr.com, кроме того, он был одним из разработчиков мобильного приложения для конференций MicroStrategy World 2016 и 2017. Обладает более чем тридцатью сертификатами MicroStrategy, включая сертификат MCEP.

Никита Барсуков (Nikita Barsukov) является опытным специалистом по обработке и анализу данных, сосредоточенным на разработке комплексных аналитических решений. Никита родился и вырос в Украине, а обучение проходил в Финляндии и Швеции. Свой профессиональный путь он начал в области разработки программного обеспечения, но вскоре понял, что его предназначение – это анализ данных и проектирование аналитических инструментов, позволяющих людям лучше разбираться в своих данных. В настоящее время Никита Барсуков работает в Microsoft, где в составе команды разработчиков трудится над аналитическими решениями для Power Platform и Dynamics 365. Помимо работы Никита любит слушать подкасты и аудиокниги, играть в настольные игры с друзьями, а иногда и сам с собой, бегать, пить крафтовое пиво и читать книги. Он живет в Копенгагене с супругой и тремя детьми.

Ана Мария (Ana Maria) является консультантом и тренером по бизнес-аналитике, а также лауреатом звания Microsoft Data Platform MVP, партнером Microsoft Power BI и тренером LinkedIn Learning. Ана Мария находится в индустрии более 25 лет – в 1990-х она разрабатывала решения для FoxPro, а ныне работает, консультирует и преподает в области бизнес-аналитики. Она окончила факультет экономической информатики в Московском государственном университете управления, после чего получила степень магистра в испанском Университете Алькала. Ана Мария специализируется на работе с инструментами бизнес-аналитики от Microsoft, а также с SQL Server, Excel, Azure Machine Learning, R и Power BI. В качестве спикера, организатора или участника ее можно встретить на различных технических форумах и мероприятиях.

Введение

Microsoft Power BI является одним из наиболее популярных инструментов бизнес-аналитики на рынке программного обеспечения для настольных и облачных решений. Книга, которую вы держите в руках, может стать вашим проводником в мир моделирования данных в целом и применительно к Power BI. Вы узнаете, как подключаться к данным в различных источниках, объединять их при помощи связей и строить полноценные модели данных.

Из книги вы поймете, как использовать принципы моделирования данных и техники навигации для определения связей между сущностями и создания модели данных, после чего мы перейдем к вопросам определения новых метрик и выполнения пользовательских вычислений с использованием особенностей модели. С течением глав сложность и эффективность моделей данных будет увеличиваться, и вы научитесь использовать язык запросов DAX, а также новые техники моделирования. С помощью примеров мы покажем вам, как можно создавать новые или адаптировать существующие модели данных с учетом разнообразных бизнес-требований. Наконец, вы освоите применение относительно свежих продвинутых возможностей для оптимизации и расширения своих моделей данных, что позволит вам решать широкий спектр задач. К концу книги вы будете обладать всеми необходимыми знаниями для структурирования и обработки данных, поступающих из разных источников, и создания на их основе полноценных моделей данных, пригодных для построения отчетов и проведения полноценного анализа данных.

Для кого эта книга

Книга предназначена для пользователей систем бизнес-аналитики, а также специалистов и разработчиков в области анализа данных, желающих улучшить свое понимание техник моделирования данных с целью извлечь максимум возможного из Power BI. Наличие базовых знаний в области Power BI и понимание схемы данных «звезда» поможет вам в освоении тем, освещающихся в этой книге.

Структура книги

Глава 1 «Введение в моделирование данных в Power BI». В данной главе мы кратко опишем функционал программного продукта Power BI и расскажем о том, почему так важно уметь моделировать данные. Здесь мы также коснемся вопросов лицензирования Power BI, напрямую влияющих на возможности моделирования данных. Кроме того, мы познакомимся с поняти-

ем итеративного моделирования данных на примере реализации в Power BI.

Глава 2 «DAX и моделирование данных». В этой главе мы будем работать с языком запросов DAX не так много и глубоко, как в третьей и четвертой частях книги. Здесь мы сконцентрируемся на функционале языка, не самом очевидном для понимания, но очень важном с точки зрения моделирования данных. Начнем главу с краткого введения в DAX, после чего сразу перейдем к рассмотрению виртуальных таблиц и функций логики операций со временем, а также их применению в реальных сценариях.

Глава 3 «Подготовка данных с помощью Power Query». Здесь мы быстро пройдемся по базовому функционалу инструмента Power Query и способам его применения на практике. Мы сделаем особый упор на параметры запросов и пользовательские функции, а также рассмотрим несколько примеров, помогающих понять, как эти техники позволяют повысить гибкость и надежность создаваемых моделей данных.

Глава 4 «Получение данных из разных источников». В этой главе мы посмотрим на способы получения данных в Power BI из наиболее распространенных источников. После этого затронем тему сертификации источников данных, которая позволяет выстроить определенные ожидания по поводу типа получаемых данных из источника. Это бывает очень полезно при оценке усилий на проектирование модели данных. Также мы рассмотрим разные режимы подключения к данным.

Глава 5 «Общие шаги по подготовке данных». Здесь мы опишем на примерах наиболее распространенные действия, которые приходится выполнять при преобразовании данных, полученных из источника. В совокупности с уже приобретенными знаниями в более ранних главах описанные здесь шаги позволят вам в будущем проектировать высокоэффективные модели данных. Изучив все описанные возможности, вы сможете по своему усмотрению выбирать способ реализации своей модели данных.

Глава 6 «Подготовка данных в Power Query для схемы "звезда"». В этой главе мы подробно поговорим о вариантах подготовки запросов для создания модели данных типа «звезда» с рассмотрением реальных сценариев. Здесь мы будем активно использовать язык программирования M, встроенный в Power Query. С учетом всех полученных ранее знаний по предварительной подготовке данных в Power Query вам не должно составить труда разобраться с предложенными здесь примерами. Кроме того, вы научитесь создавать таблицы измерений и таблицы фактов, а также денормализовывать запросы при необходимости.

Глава 7 «Эффективные методики подготовки данных». Здесь мы поговорим о типичных шаблонах при преобразовании данных, полученных из разных источников. Использование этих шаблонов позволит вам повысить эффективность создаваемых моделей данных, которые будет легче поддерживать. Прочитав эту главу, вы сможете избегать распространенных ошибок при проектировании моделей данных, что значительно облегчит вам жизнь в будущем.

Глава 8 «Элементы моделирования данных». Эта глава будет посвящена составляющим компонентам моделей данных с точки зрения Power BI, рас-

смотренным на примерах. Здесь мы будем активно использовать DAX, так что базовое понимание этого языка запросов будет крайне желательным. При рассмотрении примеров мы будем иметь дело с полноценной моделью данных типа «звезда». В этой главе мы также коснемся темы особых таблиц, использование которых может позволить обогатить модель данных за счет добавления в нее сложной бизнес-логики.

Глава 9 «Схема "звезда" и распространенные техники при моделировании данных». Здесь мы поговорим о принятых нормах при проектировании моделей данных и постараемся сделать все, чтобы вы не допускали наиболее распространенных ошибок на этапе разработки модели. К примеру, проблему с типом данных в ключевом столбце, используемом в связи, бывает очень непросто обнаружить, тогда как предотвратить ее на этапе проектирования модели проще простого. Так что освоение распространенных техник позволит вам в конечном счете сэкономить драгоценные время и деньги.

Глава 10 «Продвинутые техники моделирования данных». В этой главе мы затронем тему особых приемов при моделировании данных, помогающих в решении поставленных бизнес-задач. Хороший специалист по моделям данных должен быть открыт всему новому. Вы в своей практике легко можете столкнуться с описанными в этой главе или очень похожими на них бизнес-требованиями. И здесь мы пытаемся донести до вас мысль о том, что вы всегда должны с готовностью принимать новые вызовы бизнеса и быть готовы применить все известные вам инновационные приемы для их решения.

Глава 11 «Безопасность на уровне строк». Здесь мы посмотрим, как реализуется в модели данных Power BI *безопасность на уровне строк* (row-level security – RLS). Работать с этой технологией бывает не так просто, и чтобы понять все ее тонкости, необходимо обладать глубокими знаниями в области моделирования данных и распространения фильтров. В данной главе мы поможем вам освоить эти концепции, что позволит вам в будущем проектировать эффективные и надежные модели.

Глава 12 «Дополнительные опции и возможности моделирования данных». В заключительной главе книги мы подробно поговорим о таких дополнительных концепциях моделирования данных, как *медленно меняющиеся измерения* (Slowly Changing Dimensions – SCD), *безопасность на уровне объектов* (Object-Level Security (OLS), *потоки данных* (dataflows) и *составные модели* (composite model).

Как извлечь максимум из книги

Вам необходимо загрузить и установить последнюю версию Power BI Desktop. Все выражения проходили проверку в мартовском релизе Power BI Desktop 2021 года и должны без проблем работать в более поздних версиях программы. В дополнение к Power BI Desktop желательно будет установить программы DAX Studio и Tabular Editor.

Программное/аппаратное обеспечение, используемое в книге	Требования к операционной системе
Power BI Desktop https://powerbi.microsoft.com/en-us/downloads/	– Windows 8.1 / Windows Server 2012 R2 или выше; – .NET 4.6.2 или выше; – Internet Explorer 11 или выше; – память (RAM): минимум 2 Гб, рекомендовано 4 Гб и более; – видеосистема: минимум 1440×900 или 1600×900 (16:9). Более низкие разрешения экрана, такие как 1024×768 или 1280×800, не поддерживаются, поскольку в этом случае некоторые элементы управления (например, закрытие стартовой заставки) не могут быть корректно отражены; – параметры экрана: если вы настроили дисплей таким образом, чтобы масштаб текста, приложений и других элементов мог превышать значение 100 %, вы можете не увидеть некоторых диалоговых окон в Power BI Desktop. Если у вас возникла такая проблема, откройте **Параметры** (Settings) ⇒ **Система** (System) ⇒ **Дисплей** (Display) и установите значение масштаба в 100 %; – процессор (CPU): 1 ГГц 64 бит (x64) или выше
DAX Studio https://daxstudio.org/downloads/	– Windows 7 или выше (рекомендуется Windows 10); – .NET Framework 4.7.1 или выше
Tabular Editor https://github.com/otykier/TabularEditor/releases/tag/2.16.0	– Windows Server 2019; – Windows Server 2016; – Windows 7 или выше (рекомендуется Windows 10); – клиентские библиотеки Azure Analysis Services «AMO» версии 18.4.0.5 или выше

ПРИМЕЧАНИЕ Начиная с 31 января 2021 года Power BI Desktop прекратил поддержку Windows 7.

Для проверки примеров из некоторых глав вам понадобится наличие аккаунта *службы Power BI* (Power BI Service). Вы можете зарегистрироваться в службе и приобрести лицензию в качестве отдельного пользователя. Подробнее почитать можно по адресу https://docs.microsoft.com/en-us/Power-bi/fundamentals/service-self-service-signup-for-Power-bi?WT.mc_id=5003466.

Если вы используете цифровую версию книги, мы советуем вам вводить весь код вручную или скачивать его с сайта www.dmkpress.com на странице с описанием данной книги. Это позволит вам избежать возможных ошибок при копировании и вставке кода.

При написании книги я предполагал, что вы знакомы с терминологией и базовыми принципами хранилищ данных и схемы «звезда». При этом в книге, когда это необходимо, приводятся термины с их кратким описанием.

Загрузите сопроводительные файлы

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.

Условные обозначения

На протяжении книги мы будем использовать следующие условные обозначения и шрифты.

`Код в тексте`: так в тексте книги мы будем обозначать код. Пример: «Таблица `Customertable` широкая и длинная».

Термины: так будут написаны важные термины, названия папок и файлов, пути, пользовательский ввод и прочее.

Блоки кода будут выделены следующим образом:

```
Sequential Numbers =
SELECTCOLUMNS(
    GENERATESERIES(1, 20, 1)
    , "ID"
    , [Value]
    )
```

Жирный шрифт: так будут выделяться новые термины, важные слова и текст, который вы видите на экране. Например, таким образом будут обозначаться пункты меню. Пример: «Нажмите на кнопку **Создать таблицу** (New table) на вкладке **Моделирование** (Modeling)».

СОВЕТЫ ИЛИ ВАЖНЫЕ ПРИМЕЧАНИЯ будут оформлены так.

Часть I
Моделирование данных в Power BI

В этой вводной части мы обсудим общие принципы моделирования данных при помощи Power BI. Мы будем предполагать, что вы уже знаете, что такое и для чего используются Power Query и DAX, а также понимаете базовые принципы схемы «звезда». В этой части вы узнаете, как применять виртуальные таблицы и функции логики операций со временем в DAX, кроме того, мы поговорим о принципах проектирования эффективных моделей данных на основе реальных сценариев.

Содержание этой части:

- глава 1 «Введение в моделирование данных в Power BI»;
- глава 2 «DAX и моделирование данных».

Глава 1

Введение в моделирование данных в Power BI

Power BI – это не просто инструмент для построения отчетов. Это полноценная платформа, предоставляющая богатейший спектр возможностей, – от подготовки исходных данных до их моделирования и визуализации. Кроме того, 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.

Одним из важнейших факторов, влияющих на формирование эффективной экосистемы в Power BI, является правильность лежащих в ее основе данных. 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.

В этой главе мы познакомим вас с таким понятием, как слои Power BI, и вместе посмотрим, как данные перемещаются между слоями, что помогает при эффективном решении потенциальных проблем. После этого мы поговорим о таком важнейшем аспекте платформы Power BI, как моделирование данных. Вы узнаете об ограничениях моделей данных и разных возможностях в зависимости от используемой лицензии. Наконец, мы познакомимся с понятием итеративного моделирования данных и его фазами.

Основные темы, которые мы рассмотрим в этой главе:

- понятие слоев в Power BI Desktop;
- что означает моделирование данных в Power BI;
- варианты лицензирования в Power BI;
- итеративный подход к моделированию данных.

Понятие слоев в Power BI Desktop

Как мы уже сказали ранее, *Power BI* – это не просто инструмент для формирования отчетов. Поскольку главным образом эта книга сконцентрирована на моделировании данных, нам бы не хотелось углубляться в детали Power BI как инструмента, но некоторые основные концепции без внимания мы оставить не можем. 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). Power BI Desktop представляет собой бесплатный продукт от Microsoft, который можно загрузить по адресу https://powerbi.microsoft.com/en-us/downloads/. В этой книге мы будем подразумевать Power BI Desktop, когда говорим Power BI, если не указано иное.

На рис. 1.1 показан типичный процесс построения отчета в Power BI Desktop.

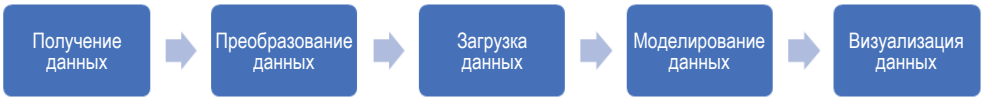

Рис. 1.1 ❖ Формирование нового отчета в Power BI

Для осуществления описанного на рис. 1.1 процесса мы используем различные *концептуальные слои* (conceptual layer) в Power BI. В Power BI Desktop эти слои отражены так, как показано на рис. 1.2.

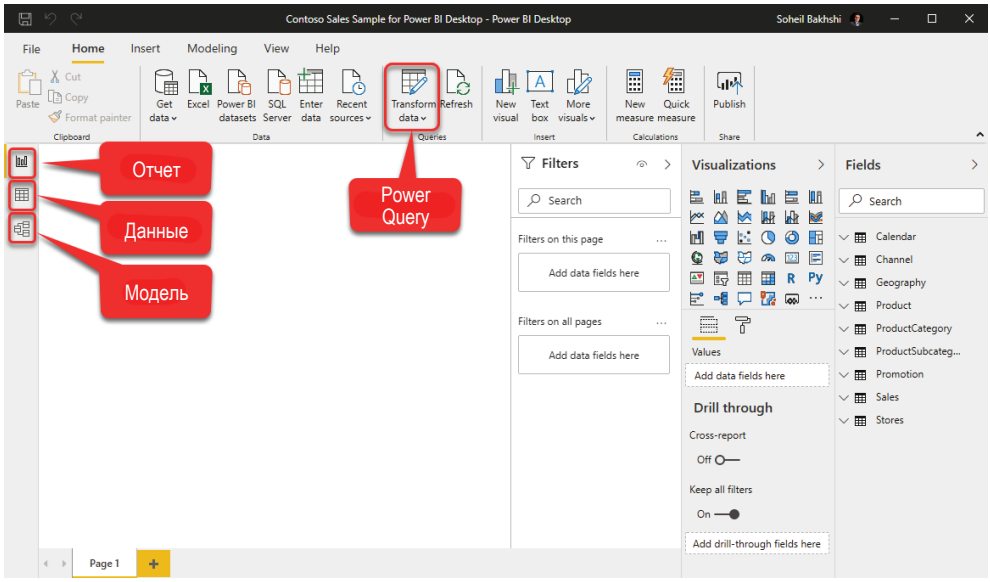

Рис. 1.2 ❖ Слои Power BI

Загрузите пример *Microsoft Contoso Sales* для Power BI Desktop по адресу https://www.microsoft.com/en-us/download/confirmation.aspx?id=46801.

Давайте детально разберем все представленные слои:

- *слой Power Query* (подготовка данных);
- *слой модели данных*;
- *слой визуализации данных*.

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, как показано на рис. 1.3.

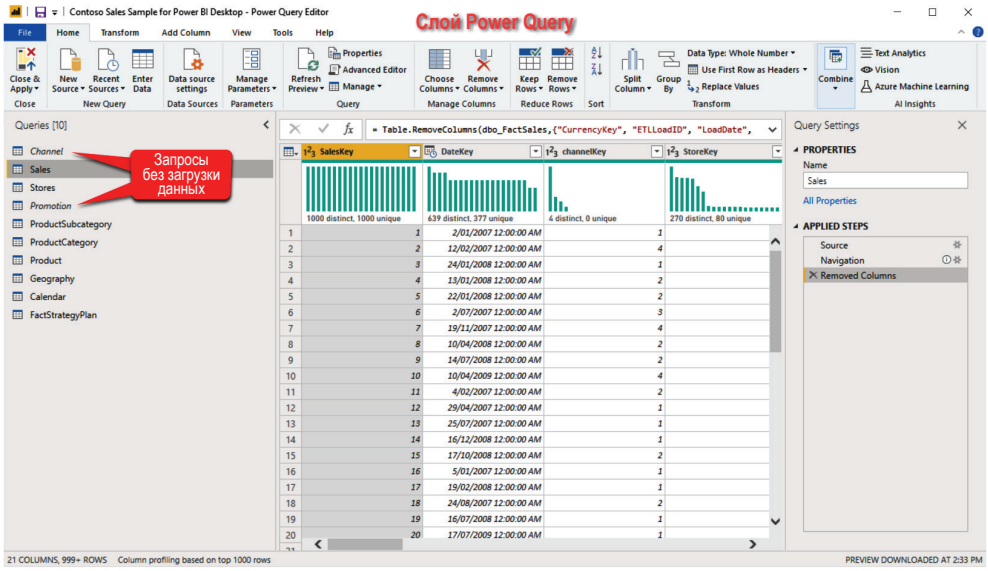

Рис. 1.3 ❖ Power Query

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.

На рис. 1.4 показан внешний вид представления **Данные** (Data) в Power BI Desktop при установленном режиме хранения **Импорт** (Import).

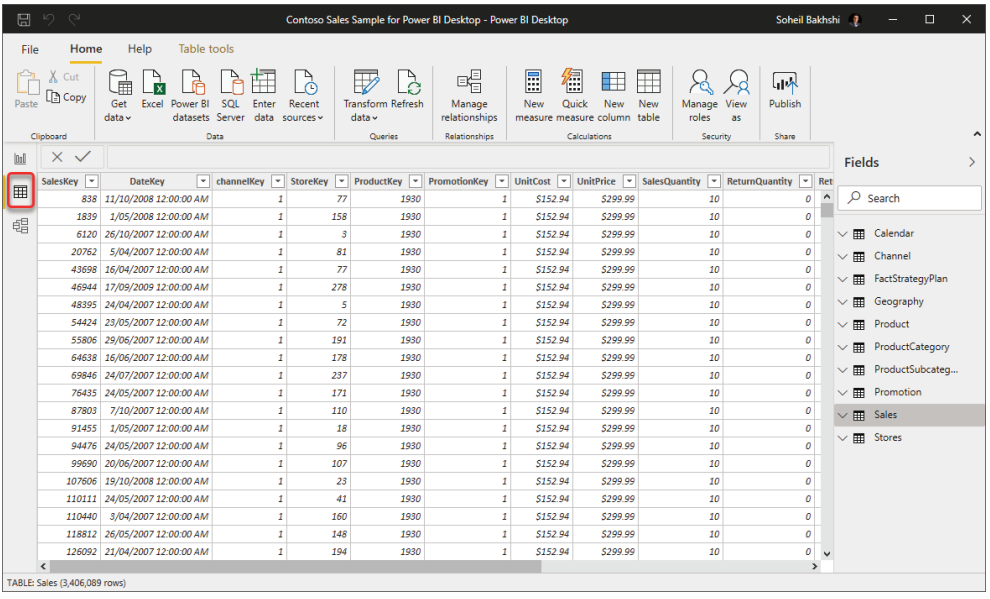

Рис. 1.4 ❖ Представление данных, режим хранения **Импорт**

Исходные данные на вкладке отображаются только в случае выбора для соответствующих таблиц режима хранения **Импорт** (Import). При выборе режима хранения **DirectQuery** информация на вкладке отображаться не будет, что видно по рис. 1.5.

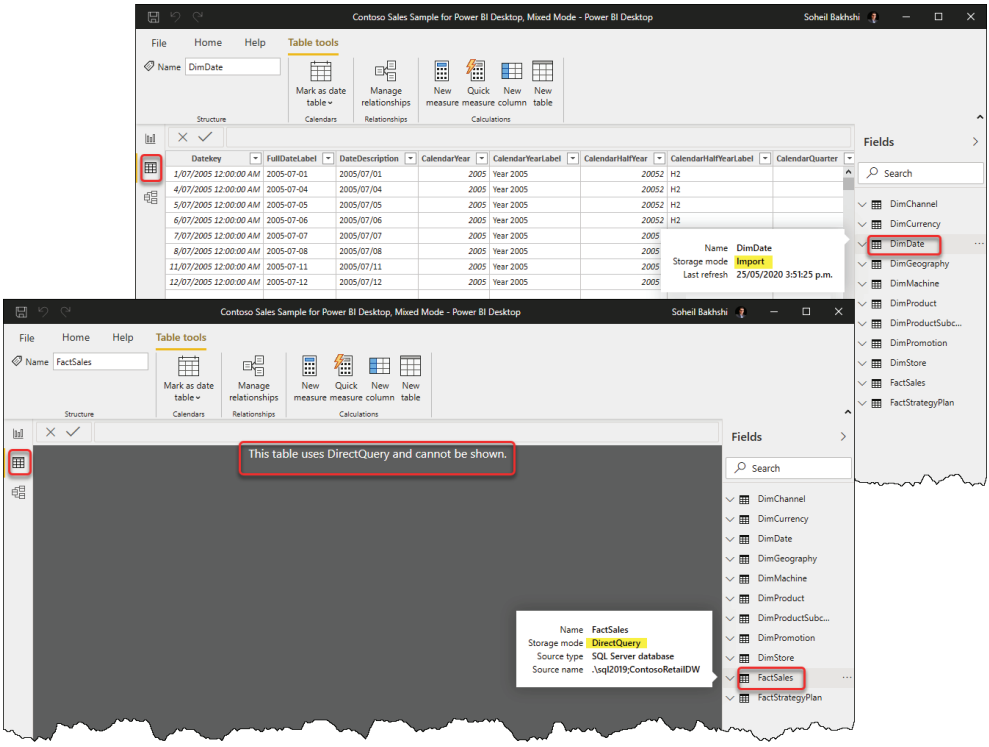

Рис. 1.5 ❖ Представление данных, режим хранения **DirectQuery**

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., что показано на рис. 1.6.

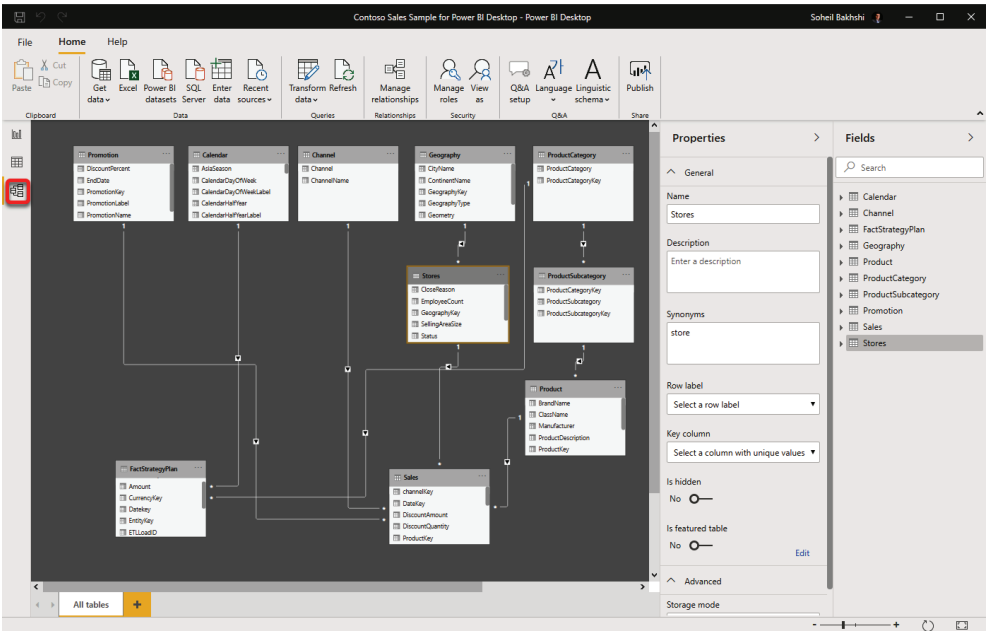

Рис. 1.6 ❖ Представление модели данных

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, как показано на рис. 1.7. 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.

Загрузите файл *Sales & Returns Sample v201912.pbix* по адресу https://docs.microsoft.com/en-us/Power-bi/create-reports/sample-datasets#sales--returns-sample-pbix-file.

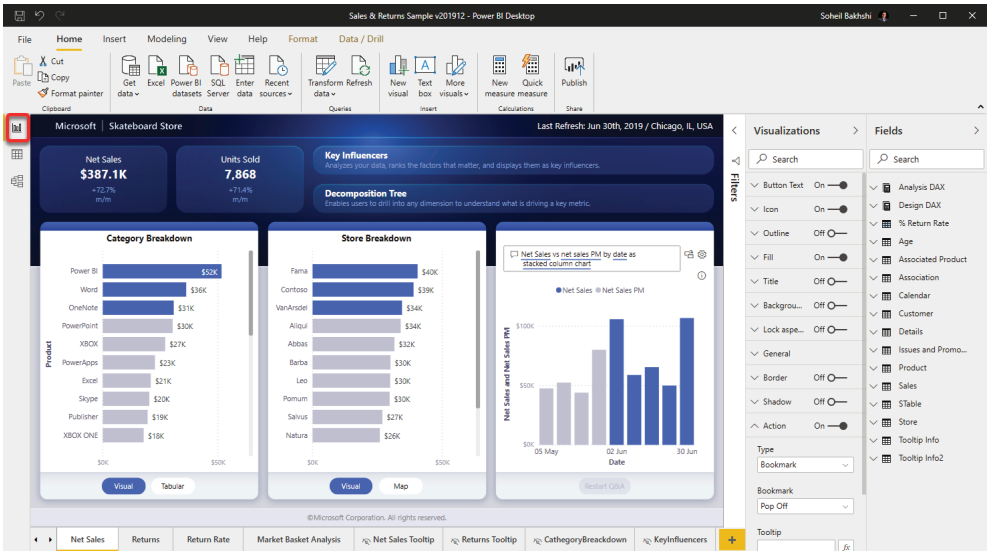

Рис. 1.7 ❖ Вкладка **Отчет** (Report)

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. Мы будем подробно говорить о пользовательских функциях в главе 3.

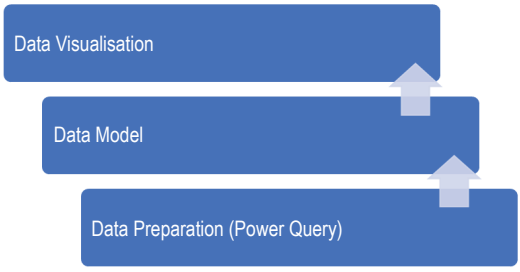

Рис. 1.8 ❖ Поток данных в Power BI

Для лучшего понимания общей картины давайте рассмотрим конкретный сценарий.

В отчете Power BI разработчик определил *параметр запроса* (query parameter), содержащий список заглавных букв **E**, **O** и **P**. Есть запрос *Product* в Power Query, содержащий описательную информацию о товарах. Столбец *Product Name* отфильтрован по списку параметров. Таким образом, когда разработчик выбирает букву **E** в параметре, запрос *Product* фильтрует результаты, оставляя только товары, начинающиеся с этой буквы.

Вы создаете табличный визуальный элемент в отчете с выводом столбца *Product Name*. Можете ли вы добавить срез в отчет, показывающий значения параметра, чтобы пользователь имел возможность выбора и фильтрации списка товаров?

Это довольно распространенный вопрос, который время от времени задают разработчики Power BI. Чтобы ответить на него, необходимо поразмышлять о слоях Power BI. Давайте проведем небольшой анализ:

- параметры запроса определяются в слое подготовки данных (Power Query);
- фильтрация запроса также представляет собой шаг преобразования в Power Query, влияющий на результирующий набор запроса. Таким образом, когда мы импортируем данные в модель данных, результирующий набор запроса *не* изменится, пока мы не вернемся в Power Query и не поменяем значение параметра. Только в этом случае итоговый набор данных изменится, и эти изменения будут загружены в модель данных в момент следующего импорта;
- по умолчанию значения параметров запросов не загружаются в модель данных, если не установить флажок **Включить загрузку** (Enable Load). Включение этой опции позволит загрузить только выбранные значения параметра, а не весь список;
- срез представляет собой элемент визуализации. Таким образом, мы говорим о слое визуализации данных, а значит, срез может пользоваться только значениями, доступными в модели данных.

Следовательно, ответ *нет*. После загрузки результирующего набора запроса в модель только эти данные будут доступны в слое визуализации.

Что означает 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 за-

просов 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 бизнес-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.

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

- источник данных *OData* из 15 таблиц, каждая из которых содержит от 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;
- ваша организация обладает лицензией Power BI Pro.

Уже по одному описанию сценария к представленным источникам данных могут возникать серьезные вопросы, и ниже я перечислил некоторые из них.

OData. Это онлайновый источник, а значит, со скоростью загрузки данных из него могут возникать следующие проблемы:

- таблицы в источнике слишком широкие, что может негативно сказываться на эффективности загрузки;
- с имеющейся лицензией Power BI Pro мы будем ограничены размером файла 1 Гб;
- на следующие вопросы также придется найти ответы. Без этого итоговая модель данных может получиться объемнее, чем это необходимо, а скорость обращения к ней будет страдать, что приведет к недовольству конечных пользователей:
  - нужно ли нам импортировать все столбцы из 15 таблиц?
  - нужно ли нам загружать все данные или хватит их части? Иными словами, если в базе хранятся исторические данные за последние десять лет, стоит ли импортировать их все, или для принятия бизнес-решений будет достаточно ограничиться одним-двумя последними годами?

Excel. Обычно рабочие книги Excel с большим количеством формул бывает достаточно тяжело поддерживать. Так что нам необходимо задаться следующими вопросами:

- сколько из 20 рабочих листов на самом деле содержат данные, необходимые нам для анализа? Некоторые листы мы можем исключить из обработки;
- как часто редактируются формулы на листах Excel? Это очень важно знать, поскольку изменение формул может привести к ошибкам при обработке данных в Power Query. Таким образом, вам нужно быть готовыми к тому, что некоторые формулы придется продублировать в Power BI при необходимости.

Хранилище данных в SQL Server. Всегда полезно иметь в качестве источника *хранилище данных* (data warehouse), поскольку обычно в нем с точки зрения аналитики все структурировано гораздо лучше. Но в нашем сценарии гранулярность обеих таблиц фактов установлена на уровне минуты. Это очень быстро может обернуться проблемами. Помните, что у нас в наличии есть только лицензия Power BI Pro, а значит, мы ограничены размером файла 1 Гб. Таким образом, будет нелишним ответить на ряд бизнес-вопросов, прежде чем принимать такую структуру:

- нужен ли нам анализ всех показателей с точностью до минуты или хватит уровня гранулярности день?
- нужно ли нам загружать все имеющиеся данные в Power BI или достаточно будет загрузить их часть?

Теперь мы знаем, какого рода вопросы нужно задать. Но что, если нам действительно понадобится анализировать все имеющиеся исторические данные? В этом случае можно рассмотреть вариант использования составной модели с агрегатами.

Помимо этого, есть еще один момент, который нужно учитывать. У нас в хранилище уже присутствуют пять измерений. Эти измерения потенциально можно повторно использовать в нашей модели данных. Таким образом, разумно будет рассмотреть и другие источники данных на предмет поиска общностей в шаблонах данных.

У вас могут возникнуть собственные вопросы к бизнесу по поводу будущей структуры данных. Главное – понять, что эти вопросы просто необходимо формулировать и задавать до начала проектирования модели, – в противном случае вы рискуете совершить очень большую ошибку. Существуют и другие аспекты, которые необходимо учитывать при управлении проектом, но они выходят за рамки данной книги.

Итак, сейчас мы можем выделить три основных момента, которые стоит принимать во внимание при создании модели данных:

- мы должны заранее задать бизнесу интересующие нас вопросы, чтобы впоследствии не возникло проблем и необходимости все переделывать;
- нужно иметь в виду технологические ограничения и предпринять соответствующие меры;
- необходимо хорошо разбираться в вопросах моделирования данных, чтобы осуществить полноценный поиск общих шаблонов данных во избежание перекрытий данных.

Вы можете спросить: «И как мне все это сделать?», на что я отвечу, что первый шаг на этом пути вы уже сделали, взяв в руки данную книгу. Все вопросы, которые мы здесь обсуждали, и многие другие будут максимально подробно освещены в книге. Остальное зависит от вас и от того, как вы применяете на практике то, что изучили теоретически.

Схемы «звезда» (многомерное моделирование) 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, так что в этой книге мы будем использовать в основном его. Здесь мы не ставим себе цель научить вас многомерному моделированию. Однако, несмотря на это, мы рассмотрим общие принципы моделирования данных с использованием техники схемы «звезда» и напомним вам базовые концепции этого подхода.

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.

Давайте продолжим со сценарием.

Скажем, у нас есть транзакционная система для международной сети розничных магазинов. Каждую секунду в нашу систему записываются сотни транзакций с разных магазинов по всему миру. Владелец компании хочет видеть общую сумму продаж за последние полгода.

Звучит несложно. Достаточно применить простую функцию суммирования к продажам. Но не забывайте, что у нас каждую секунду добавляются сотни операций в базу. Даже если взять по минимуму – сто транзакций в секунду, – за день у нас наберется 8 640 000 транзакций, а за полгода – более полутора миллиарда строк. Очевидно, что простая операция суммирования вдруг перестала быть такой уж простой и быстрой.

Следующий шаг сценария. От руководства компании поступает новый запрос. Теперь владельцу хочется посмотреть сумму продаж за полгода в разрезе стран и городов, чтобы определить лидирующие географические направления.

Итак, нам необходимо добавить дополнительное условие к нашему расчету суммы с *объединением* (join) с таблицей географии. Если у вас есть опыт работы с реляционными базами данных, вы должны понимать, что объединение таблиц – это достаточно дорогостоящая операция. Этот сценарий мы

можем развивать до бесконечности, но вы уже видите, как быстро у вас могут возникнуть проблемы с производительностью.

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).

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 рис. 1.9. Именно отсюда и произошло такое ее название. В этой книге мы в основном будем работать с базой *Adventure Works DW* – известным тестовым набором данных от Microsoft, если не указано иное. Это вымышленная компания по продаже велосипедов, ведущая торговлю как онлайн, так и через сеть розничных магазинов. На рис. 1.9 показана таблица фактов *Internet Sales* в окружении измерений на схеме «звезда».

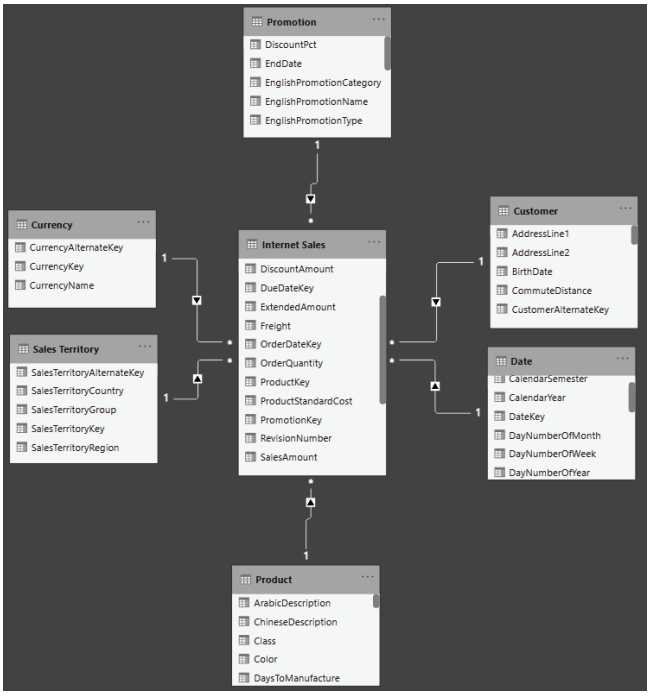

Рис. 1.9 ❖ Модель данных *Adventure Works DW*, таблица фактов *Internet Sales* («звезда»)

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». На рис. 1.10 показана эта схема применительно к модели данных *Adventure Works*.

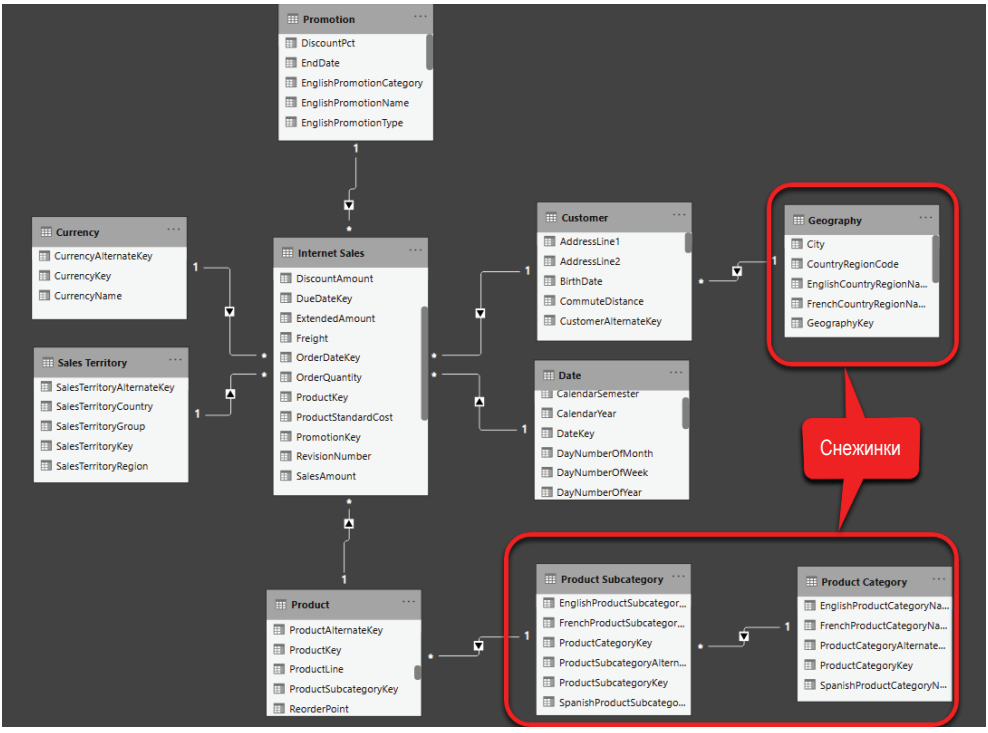

Рис. 1.10 ❖ Модель данных *Adventure Works DW*, таблица фактов *Internet Sales* («снежинка»)

Понятие денормализации

В сценариях из обычной жизни далеко не все могут похвастаться наличием готового хранилища данных, построенного на базе схемы «звезда». Вместо этого в большинстве случаев наличие связанных измерений, ведущее к образованию схемы «снежинка», просто неизбежно. Ваша модель данных может

быть построена на основании разных источников данных, включая транзакционные системы и нетранзакционные, такие как файлы Excel и CSV. Таким образом, вам просто необходимо до определенной степени денормализовывать ваши модели. В зависимости от ваших требований вы можете сочетать нормализацию и денормализацию в нужной вам пропорции. Никакого золотого правила соотношения нормализованных и денормализованных данных в схеме просто не существует. Принято считать, что денормализовать данные необходимо до тех пор, пока измерения не будут полноценно их описывать.

В предыдущем примере с моделью *Adventure Works DW* «снежинка», образуемая ветвью измерений *Product Category* и *Product Subcategory*, может быть легко денормализована в таблице *Product*.

Давайте рассмотрим конкретный пример. Выполните следующие действия, чтобы денормализовать таблицы *Product Category* и *Product Subcategory* в измерении *Product*.

> **ПРИМЕЧАНИЕ** Вы можете загрузить файл *Adventure Works, Internet Sales.pbix* по адресу https://github.com/PacktPublishing/Expert-Data-Modeling-with-Power-BI/blob/master/Adventure%20Works%20DW.pbix.

Откройте файл *Adventure Works, Internet Sales.pbix* и проделайте следующие шаги, как показано на рис. 1.11.

1. Нажмите на кнопку **Преобразование данных** (Transform data) на вкладке **Главная** (Home) в группе **Запросы** (Queries).

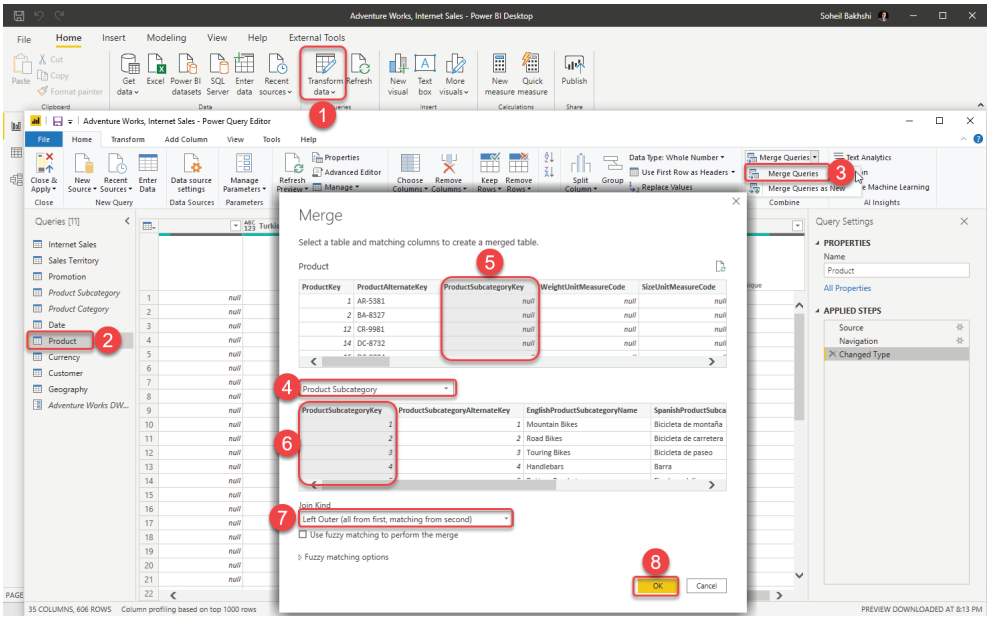

Рис. 1.11 ❖ Объединение таблиц *Product* и *Product Subcategory*

2. Выберите запрос *Product*[1].
3. Нажмите на кнопку **Объединить запросы** (Merge Queries) на вкладке **Главная** (Home) в группе **Объединить** (Combine).
4. Выберите в выпадающем списке запрос *Product Subcategory*.
5. В таблице *Product* выделите столбец *ProductSubcategoryKey*.
6. В таблице *Product Subcategory* выделите столбец *ProductSubcategoryKey*.
7. Выберите тип **Внешнее соединение слева (все из первой таблицы, совпадающие из второй)** в выпадающем списке **Тип соединения** (Join Kind).
8. Нажмите на кнопку **ОК**.

В результате на панели **Примененные шаги** (Applied steps) добавится новый шаг **Объединенные запросы** (Merged Queries). При этом в новом столбце *Product Subcategory*, являющемся *структурированным* (Structured Column), во всех строках будет содержаться табличное значение *Table*. Больше о структурированных столбцах вы узнаете в главе 3.

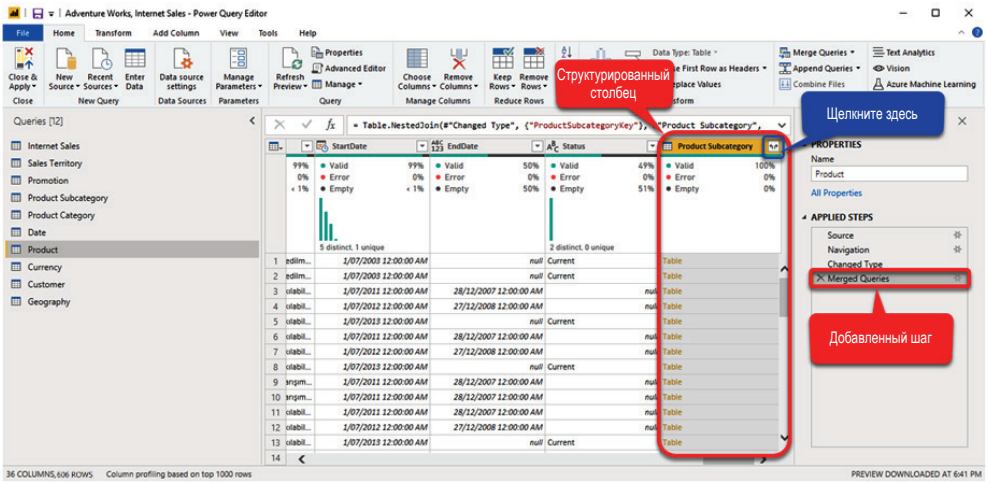

Рис. 1.12 ❖ Объединение таблиц *Product* и *Product Subcategory*

Теперь давайте посмотрим, как развернуть структурированный столбец в редакторе запросов.

1. Нажмите на кнопку **Развернуть** (Expand) в заголовке столбца *Product Subcategory*, как показано на рис. 1.13.
2. Установите флажок напротив поля *ProductCategoryKey*.
3. Установите также флажок для поля *EnglishProductSubcategoryName* и снимите все остальные флажки.
4. Снимите флажок **Использовать исходное имя столбца как префикс** (Use original column names as prefix).
5. Нажмите на кнопку **ОК**.

[1] Предварительно необходимо прописать путь к локальному файлу Excel в параметрах Power BI. – *Прим. ред.*

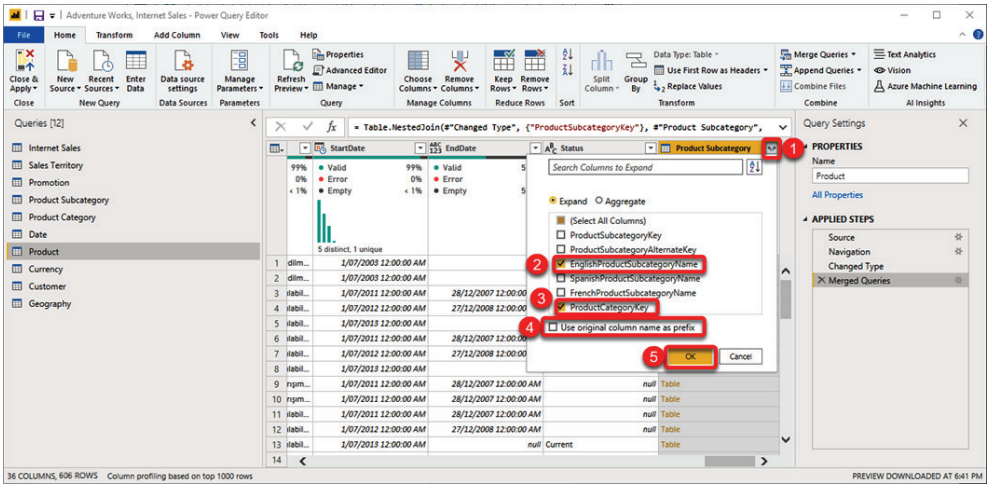

Рис. 1.13 ❖ Разворачивание структурированного столбца в редакторе запросов

В результате мы добавили столбцы *EnglishProductSubcategoryName* и *ProductCategoryKey* из запроса *Product Subcategory* в запрос *Product*. Теперь нужно добавить столбец *EnglishProductCategoryName* из запроса *Product Category*. Для этого необходимо выполнить объединение запросов *Product* и *Product Category*, как показано на рис. 1.14.

1. Снова нажмите на кнопку **Объединить запросы** (Merge Queries).
2. Выберите в выпадающем списке запрос *Product Category*.
3. В таблице *Product* выделите столбец *ProductCategoryKey*.
4. В таблице *Product Category* выделите столбец *ProductCategoryKey*.

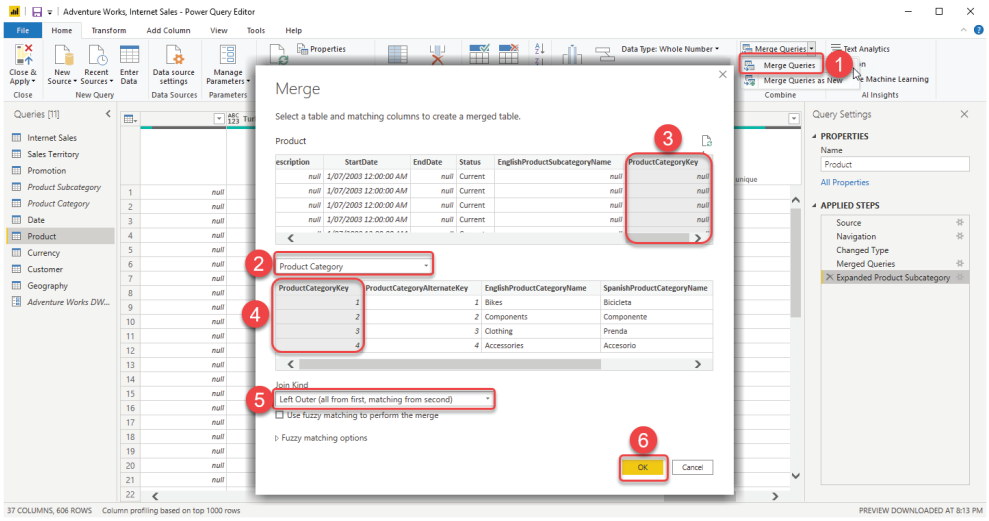

Рис. 1.14 ❖ Объединение таблиц *Product* и *Product Category*

5. Выберите тип **Внешнее соединение слева (все из первой таблицы, совпадающие из второй)** в выпадающем списке **Тип соединения** (Join Kind).
6. Нажмите на кнопку **OK**.

В результате будет добавлен еще один шаг к запросу и еще один структурированный столбец с именем *Product Category*. Развернем его, как показано на рис. 1.15.

1. Разверните новый столбец при помощи соответствующей кнопки в заголовке.
2. Выделите в списке только поле *EnglishProductCategoryName*.
3. Снимите флажок **Использовать исходное имя столбца как префикс** (Use original column names as prefix).
4. Нажмите на кнопку **OK**.

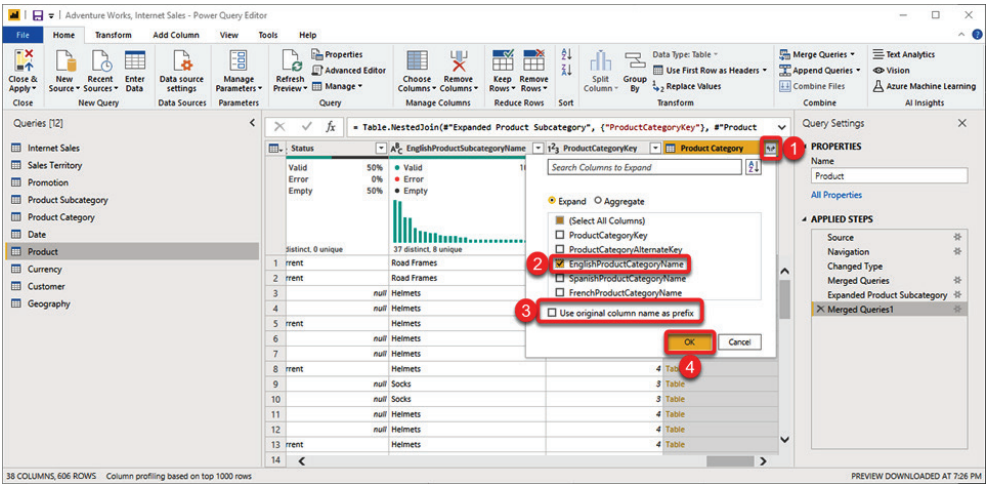

Рис. 1.15 ❖ Разворачивание структурированного столбца

Осталось избавиться от столбца *ProductCategoryKey*, в котором больше нет необходимости. Для этого проделайте следующие действия, показанные на рис. 1.16.

1. Выделите столбец *ProductCategoryKey*.
2. Нажмите на кнопку **Удалить столбцы** (Remove Columns) в группе **Управление столбцами** (Manage Columns) на вкладке **Главная** (Home).

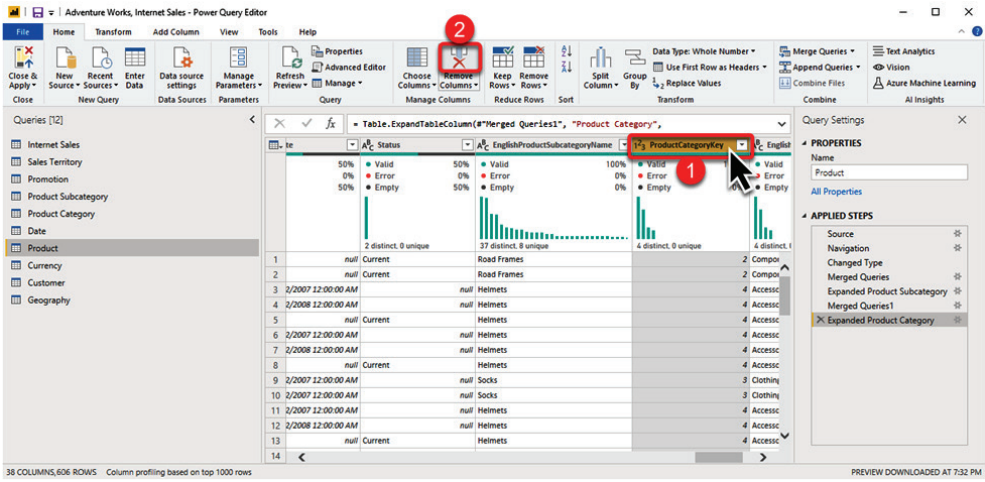

Рис. 1.16 ❖ Удаление столбца в редакторе запросов

Итак, мы объединили лучи снежинки *Product Category* и *Product Subcategory* в запросе *Product*, тем самым денормализовав нашу снежинку.

На заключительном шаге нам необходимо исключить запросы *Product Category* и *Product Subcategory* из загрузки. Для этого выполните следующие действия, показанные на рис. 1.17.

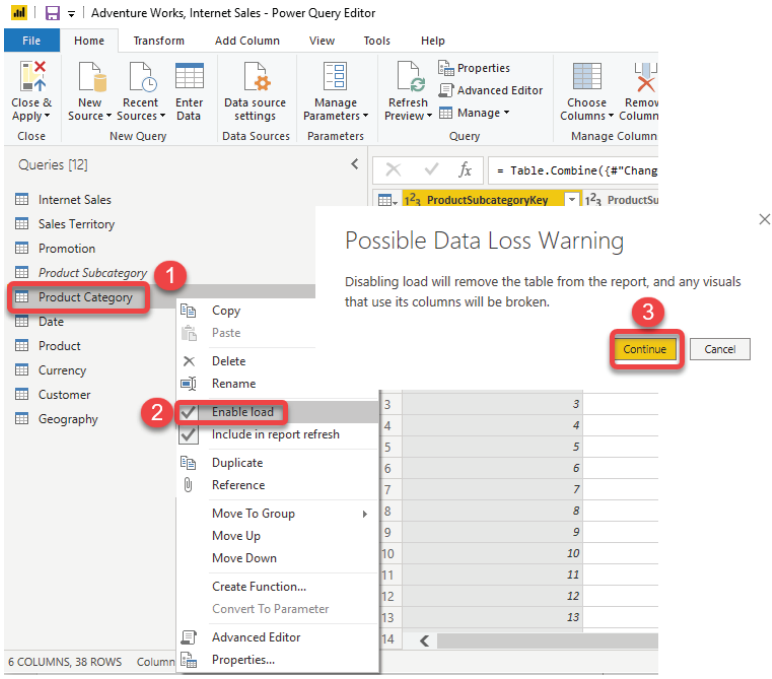

Рис. 1.17 ❖ Исключение запросов из загрузки в редакторе запросов

1. Щелкните правой кнопкой мыши по каждому из этих запросов.
2. Снимите флажок **Включить загрузку** (Enable load) в контекстном меню.
3. Нажмите кнопку **Продолжить** (Continue) в появившемся окне с предупреждением о возможной потере данных.

Все, что нам осталось сделать, – это импортировать данные в модель путем нажатия на кнопку **Закрыть и применить** (Close & Apply), как показано на рис. 1.18.

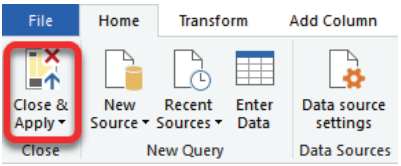

Рис. 1.18 ❖ Импорт данных в модель

Итак, мы достигли поставленной цели – денормализовали таблицы *Product Category* и *Product Subcategory*, сведя их к столбцам *EnglishProductCategoryName* и *EnglishProductSubcategoryName* в измерении *Product*.

Варианты лицензирования в Power BI

Вы наверняка подумали, как могут быть связаны варианты лицензирования в Power BI с моделированием данных. Оказывается, они связаны напрямую, поскольку выбор лицензии непосредственно влияет на характеристики, от которых зависят особенности будущей модели данных. В то же время, какой бы тип лицензии вы ни использовали, Power BI Desktop был и остается бесплатным продуктом. В этом разделе мы рассмотрим некоторые аспекты, связывающие моделирование данных с выбранным типом лицензии.

В табл. 1.1 приведена упрощенная версия сравнения лицензий на Power BI с сайта Microsoft.

Таблица 1.1. Сравнение типов лицензирования Power BI

Лицензия Power BI	Максимальный размер набора данных (Гб)	Добавочная загрузка данных	Группы вычислений	Общие наборы данных	Потоки данных Power BI
Free	1	Нет	Нет	Нет	Нет
Professional	1	Да	Нет	Да	Да
Power BI Report Server	2	Да	Нет	Н/А	Н/А
EM1/A1	3	Да	Да	Да	Да
EM2/A2	5	Да	Да	Да	Да
EM3/A3	10	Да	Да	Да	Да

Таблица 1.1 *(окончание)*

Лицензия Power BI	Максимальный размер набора данных (Гб)	Добавочная загрузка данных	Группы вычислений	Общие наборы данных	Потоки данных Power BI
P1/A4	25	Да	Да	Да	Да
P2/A5	50	Да	Да	Да	Да
P3/A6	100	Да	Да	Да	Да
P4	200	Да	Да	Да	Да
P5	400	Да	Да	Да	Да
PPU	100	Да	Да	Да	Да

Максимальный размер набора данных

Как видно по табл. 1.1, при наличии лицензий Power BI Free и Professional вы будете ограничены в отношении каждого опубликованного набора данных размером 1 Гб. А значит, этому аспекту придется уделить повышенное внимание при проектировании модели данных. Существует несколько способов удерживаться в рамках установленного ограничения:

- импортировать в модель данных только нужные столбцы;
- импортировать только часть данных. Опишите имеющиеся у вас ограничения руководству и узнайте, какие данные не являются критически важными для принятия решений. К примеру, мало какой отрасли бизнеса может понадобиться оперативный анализ данных за последние десять лет, так что вы с большой долей вероятности можете избавиться от лишних периодов;
- используйте агрегации. В большинстве случаев данные в вашем хранилище будут находиться с максимальной степенью детализации (низкой гранулярностью). В то же время при анализе обычно требуется повышать уровень гранулярности данных. Таким образом, вы можете агрегировать данные, повысив их гранулярность, после чего загружать в модель данных. К примеру, в источнике ваши данные могут храниться с детализацией до минуты, а руководству компании информация предоставляется с точностью до дня;
- рассмотрите вариант отключения настройки автоматических даты и времени в Power BI Desktop;
- оптимизируйте типы данных.

Подробнее о приведенных способах мы будем говорить в следующих главах книги.

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 каж-

дый 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.

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. В рассматриваемом примере у нас есть три 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).

Таким образом, нам пришлось бы создавать по девять вычислений на базе каждой из трех представленных мер. Несложно подсчитать, что в результате в нашей модели данных добавилось бы 9 × 3 = 27 мер. Как видите, количество мер в моделях способно увеличиваться очень быстро, так что не удивляйтесь, если вам кто-то скажет, что у него в модели сотни мер.

Еще один сценарий связан с учетом нескольких валют. Без групп вычислений вам пришлось бы преобразовывать значения в текст, чтобы снабдить суммы нужным знаком валюты при помощи функции `FORMAT()` в DAX. А если совместить учет валют с вычислениями на основе логики операций со временем, проблемы умножатся.

Группы вычислений способны легко и элегантно решать подобные вопросы. Вы научитесь пользоваться ими в главе 10.

Общие наборы данных

Как понятно из названия, *общий набор данных* (shared dataset) может совместно использоваться в разных отчетах в новых рабочих областях (пришедших на смену классическим) в рамках службы Power BI. Таким образом, эта возможность доступна только для обладателей лицензионных планов Professional и Premium. Использование данной технологии предоставляет больше гибкости при создании обобщенных наборов данных, отвечающих за несколько бизнес-сущностей, вместо содержания множества наборов, обладающих схожими особенностями.

Потоки данных Power BI

Потоки данных (dataflows), также называемые *Power Query Online*, представляют собой централизованный механизм подготовки данных в рамках службы Power BI, плодами которого могут пользоваться все сотрудники компании. Подобно тому, как Power Query используется в Power BI Desktop для подготовки локальных данных, можно готовить, очищать и преобразовывать данные в потоках. И если запросы, создаваемые в Power BI Desktop при помощи Power Query и публикуемые в службе Power BI, остаются изолированными в рамках набора данных, в потоках данных вы можете совместно использовать операции очистки и манипулирования данными.

Создавать потоки данных Power BI можно внутри рабочей области, так что этот функционал доступен только владельцам лицензий Professional и Premium. О потоках данных Power BI мы будем подробнее говорить позже в этой книге.

Итеративный 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 вы продолжаете 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. Мы часто слышим следующую фразу от руководства после осуществления первых нескольких итераций:

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

Именно поэтому при разработке сценариев в Power BI лучше всего применять *пошаговый динамический подход*. На рис. 1.19 показан один из способов реализации подобного подхода.

Рис. 1.19 ❖ Итеративный подход к моделированию данных

Сбор информации от руководства

Как и в случае с любым другим программным обеспечением, процесс разработки в Power BI начинается со сбора информации от постановщиков задачи для общего понимания бизнес-требований. В реальных условиях этим может заниматься бизнес-аналитик, но многие пользователи Power BI сами являются бизнес-аналитиками. Вне зависимости от того, занимаетесь ли вы бизнес-анализом или просто моделируете данные, вам необходимо знать, какие требования к сценарию выдвигает бизнес изначально. Вы должны уметь задавать правильные вопросы и намечать определенные решения. Кроме того, вам должны быть понятны очевидные риски, и вы должны обсуждать их с пользователями продукта. Только после сбора всей необходимой информации и обсуждения со сторонами возможных решений, рисков и технических ограничений вы можете переходить к следующему шагу.

Подготовка данных на основе бизнес-логики

По завершении первого пункта в вашем распоряжении будет достаточно ценной информации. Теперь вам необходимо собрать данные из различных источников и подготовить их для анализа, пользуясь сведениями, полученными на предыдущем шаге. К примеру, если бизнес-требования предполагают подключение к источнику OData и извлечение данных из нескольких

столбцов, вы можете выполнить соответствующую подготовку подключений, минимизировав риски и приняв во внимание все технические ограничения. После подготовки данных можно приступать к следующему шагу – моделированию.

Моделирование данных

Если вы тщательно выполнили все предшествующие подготовительные действия, ваша модель данных должна получиться максимально компактной и эффективной. На этом этапе пришло время задуматься об аналитической стороне дела. Одновременно с этим вы должны учитывать все собранные на предыдущих этапах нюансы относительно бизнес-требований, рисков и технических ограничений. К примеру, если руководство не готово мириться с более чем пятиминутными задержками в данных, вам стоит задуматься об использовании режима DirectQuery. Но этот режим сопряжен со своими ограничениями и рисками. Таким образом, вам необходимо выработать такие подходы к разработке модели данных, которые будут максимально удовлетворять требованиям. Подробно о режиме хранения DirectQuery мы будем говорить в главе 4.

Проверка логики

Это один из самых важных шагов в моделировании данных, заключающийся в проверке реализованной бизнес-логики на предмет соответствия требованиям. При этом вам необходимо не только проверить расчеты и числа, вам нужно также протестировать систему на предмет производительности и удобства использования конечными пользователями. На этом этапе вы должны быть готовы к разнообразной обратной связи от пользователей вплоть до жесткой критики, даже если считаете свое решение оптимальным.

Демонстрация бизнес-логики в базовой визуализации

При моделировании данных нам нет необходимости заботиться об их визуализации. Самый быстрый способ убедиться в том, что бизнес-логика реализована правильно, – получить подтверждение этого от самого бизнеса. А самый простой способ сделать это – продемонстрировать логику при помощи визуальных элементов – таблиц и матриц – с добавкой в виде срезов. Помните, что в данном случае речь идет именно о подтверждении корректности реализации бизнес-логики, а не о конечном продукте. Обычно процесс демонстрации данных приводит к пониманию того, какая информация не была учтена, после чего начинается вторая итерация сбора данных от представителей бизнеса.

Раз за разом повторно проходя по всем этим шагам, вы постепенно будете развивать навыки моделирования данных. В следующем разделе мы поговорим о том, как мыслят профессиональные разработчики моделей данных.

> **ПРИМЕЧАНИЕ** В данной книге также реализован итерационный подход, так что мы будем то и дело возвращаться от главы к главе при рассмотрении сценариев.

Думай как профессиональный разработчик моделей данных

Раньше, в середине 90-х, я работал с транзакционными системами баз данных. В то время одним из ключевых навыков считалась нормализация модели данных как минимум до уровня *третьей нормальной формы* (third normal form). Иногда мы приводили данные к *нормальной форме Бойса–Кодда* (Boyce-Codd normal form). Я вел множество проектов, разрабатывал модели данных, допускал ошибки и учился на них. Со временем я научился визуализировать модели данных вплоть до второй или даже третьей нормальной формы прямо в голове во время сбора информации от потенциальных клиентов. Все подходы к моделированию данных, с которыми мне приходилось работать и о которых я читал, основывались на реляционных моделях, независимо от их использования, будь то транзакционные модели, схема «звезда» или хранилища данных с архитектурой Inmon или Data Vault. Все они основываются на реляционных принципах моделирования. И моделирование данных в Power BI ничем не отличается. Профессионалы могут без труда визуализировать будущие модели данных в голове уже на этапе первичного сбора информации у заказчика. Но этот навык приходит с опытом.

Более того, с приобретением опыта вы также научитесь задавать правильные вопросы заказчикам, сравнивая новые сценариями с уже готовыми и заранее предвосхищая возможные ловушки. Правильные вопросы помогут вам избежать большого количества доработок в будущем. Помимо этого, вы можете сами подсказать заказчику новые идеи для реализации тех или иных возможностей. Зачастую требования к решению будут меняться в процессе работы над системой, и вам не стоит удивляться, когда это будет происходить.

Заключение

В данной главе мы поговорили о различных слоях Power BI и обсудили присущие каждому слою элементы и требования. Таким образом, теперь при возникновении проблемы вы будете точно знать, куда смотреть. Также мы узнали, что при разработке модели данных мы фактически выстраиваем семантический слой в Power BI. Дополнительно мы рассмотрели различия между схемами «звезда» и «снежинка», которые позволяют очень эффектив-

но моделировать данные. При обсуждении вопросов моделирования данных мы также узнали, какие ограничения могут накладываться теми или иными видами лицензии на Power BI. Наконец, мы вкратце обсудили особенности итеративного подхода к моделированию данных, который позволяет проектировать более жизнеспособные и надежные модели.

В следующей главе мы рассмотрим язык запросов DAX применительно к моделированию данных. Мы обсудим такую неоднозначную тему, как виртуальные таблицы, и пройдемся по нескольким сценариям с применением логики операций со временем, которые помогут вам при моделировании данных.