

Наглядный подход к изучению и преобразованию данных из множества источников для Power BI и Excel

Power Query – технология, позволяющая соединяться с источниками данных, а также преобразовывать и очищать информацию для ее дальнейшего анализа.

Эта книга научит вас создавать интерактивные отчеты и дашборды посредством Excel и Power BI, используя множество источников данных. Вы узнаете о том, как получать доступ к Power Query из разных версий Microsoft Excel и как установить движок Power BI. Также рассмотрен инструмент Power Pivot; особое внимание уделено трудностям, которые могут возникнуть у пользователей Excel при очистке данных с его помощью. Показано, как Power Query нивелирует недостатки, характерные для Power Pivot. В заключение показаны возможности языка M, – его синтаксис и структура рассмотрены на многочисленных примерах.

Прочитав книгу, вы научитесь автоматизировать отчетность при помощи Power Query и сможете преобразовывать сырье данные в ценную аналитическую информацию.

Для читателей, работающих с отчетами и большими массивами данных.

Интернет-магазин:
www.dmkpress.com

Оптовая продажа:
КТК «Галактика»
e-mail: books@aliens-kniga.ru

Packt
ДМК
издательство
www.dmk.rf

Основные темы:

- преобразование данных с рабочих листов в табличный формат, пригодный для работы с запросами;
- создание динамических связей между базой данных Access и рабочей книгой Excel;
- переформатирование табличных данных путем изменения строк, столбцов и таблиц при помощи различных инструментов Power Query;
- автоматическое создание столбцов из имен файлов и вкладок, а также работа с несколькими файлами Excel;
- упрощение и автоматизация отчетов из нескольких источников данных;
- максимально эффективное использование дашбордов;
- понимание различий между языками запросов DAX и M.



Примеры на сайте
издательства
www.dmkpress.com

ISBN 978-5-97060-905-7



9 785970 609057 >

Изучаем Power Query

Изучаем Power Query

Линда Фоукс, Уоррен Спарроу

ДМК
издательство

Линда Фоукс
Уоррен Спарроу

Изучаем Power Query

Наглядный подход к подключению
и преобразованию данных из множества
источников для Power BI и Excel

Learn Power Query

A low-code approach to connect
and transform data from multiple sources
for Power BI and Excel

Linda Foulkes

Warren Sparrow

Packt

BIRMINGHAM—MUMBAI

Изучаем Power Query

Наглядный подход к подключению
и преобразованию данных из множества
источников для Power BI и Excel

*Линда Фоукс
Уоррен Спарроу*



Москва, 2021

УДК 004.424
ББК 32.372
Ф74

Линда Фоукс, Уоррен Спарроу
Ф74 Изучаем Power Query / пер. с анг. А. Ю. Гинько. – М.: ДМК Пресс, 2021. – 378 с.: ил.

ISBN 978-5-97060-905-7

Power Query – инструмент, позволяющий подключаться к источникам данных, а также подготавливать и обрабатывать информацию для ее дальнейшего анализа. В этой книге вы найдете все, что нужно для работы с Power Query: описание интерфейса и способов подключения к источникам данных, а также извлечения, преобразования и очищения информации при помощи мощных инструментов. Отдельная часть книги посвящена языку формул M, который обеспечивает новый подход к агрегированию данных. В завершение вы научитесь строить дашборды и многомерные отчеты при помощи Power Query.

Книга адресована профессионалам в области бизнес-аналитики и анализа данных, а также пользователям Excel, которые смогут вывести свое владение этой программой на новый уровень благодаря изучению принципов сбора и преобразования данных при помощи Power Query.

УДК 004.424
ББК 32.372

©Packt Publishing 2020. First published in the English language under the title Learn Power Query - (9781839219719).

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN (анг.) 978-1-83921-971-9
ISBN (рус.) 978-5-97060-905-7

© 2020 Packt Publishing
© Оформление, издание, перевод, ДМК Пресс, 2021

Дорогому человеку:

*С написанием этой книги была связана непростая, но крайне плодотворная и приятная часть моей жизни. Я хотела бы поблагодарить дорогого мне человека Мика за воодушевление, понимание и помочь в соблюдении сроков сдачи книги. Ты оказал мне огромную поддержку, и я очень это ценю.
Люблю тебя.*

Моему соавтору:

*Бывает очень непросто найти соавтора, который будет отвечать всем вашим требованиям и к тому же сможет присутствовать на видеоконференциях в Zoom в шесть утра, да еще и при полном параде. Уоррен, я сердечно благодарю тебя за совместную работу, мотивацию, а также твои знания и опыт. Кроме того, спасибо за то, что ближе к концу книги позволил мне взять паузу и завершить мою вторую книгу «Изучаем Microsoft Office 2019» (Learn Microsoft Office 2019), которую я писала параллельно. Мне пришли по душе наши долгие обсуждения и твое чувство юмора. Поздравляю с твоей первой публикацией!
Мы это сделали!*

Линда Фоукс

Я посвящаю эту книгу Мишель – моей супруге и другу, – а также двум моим прекрасным дочерям Данике и Джордан.

Уоррен Спарроу

Об авторах

Линда Фоукс (Linda Foulkes) является сертифицированным тренером по Microsoft Office, обладающим статусом Expert программы Microsoft Innovative Educator и более чем 25-летним опытом преподавания. Также, будучи сертифицированным тренером в области информационных технологий, Линда представляла Южную Африку на Microsoft Global Forum в Редмонде (США) в 2015 году. Кроме того, у нее есть опыт подготовки студентов к конкурсам Microsoft Office Specialist в Техасе. В издательстве Packt вышла книга Линды Фоукс «Изучаем Microsoft Office 2019» (Learn Microsoft Office 2019). Также Линда регулярно присутствует на конференциях, проводит вебинары в SchoolNet SA и участвует в мероприятиях Teach Meets и MicrosoftMeets. В сферу интересов Линды входит удаленное обучение, и она разработала ряд программ в этой области для семейства программ Microsoft Office.

Уоррен Спарроу (Warren Sparrow) является сертифицированным преподавателем Microsoft, Adobe и National Geographic. Он обладает статусом Expert программы Microsoft Innovative Educator на протяжении шести последних лет. Уоррена регулярно приглашают в качестве спикера на различные образовательные мероприятия, включая ежегодную конференцию Microsoft по Office 365, которую он посетил в 2015 году. Главным образом он сосредоточен на области разработки долгосрочных стратегий, преподавании и внедрении новых технологий. Уоррен был вовлечен в разработку и преподавание как в сегменте образования, так и в области бизнеса. Также он был советником в департаменте образования в Западной Капской провинции (ЮАР), способствуя выходу нового программного и аппаратного обеспечения, а также занимался обучением преподавателей и внедрением новых технологий в школах.

Благодарности

Мы с Уорреном остались крайне довольны процессом написания этой книги и хотели бы от всей души поблагодарить команду издательства Packt за содействие. Без вашей поддержки и помощи в написании глав книги, корректуре, установке и соблюдении сроков, а также выпусксе сигнального экземпляра книги все это стало бы совершенно невозможным. Мы благодарим всех и каждого, кто оказывал нам содействие, но отдельно хотелось бы отметить рецензента издательства Карана Гупту (Karan Gupta), который посвящал все свое время нашему проекту на его начальной стадии и помогал на протяжении всего процесса написания книги. Также не забудем отметить редактора разработки контента Тикшу Лэд (Tiksha Lad) за ее профессиональный вклад и Праджакту Наик (Prajakta Naik) за тщательное отслеживание сроков сдачи книги и прочую поддержку. Кроме того, хочется сказать спасибо Вишванатха Музумдару (Vishwanath Muzumdar) за уместную и своевременную критику, которая в конечном итоге помогла сделать книгу лучше.

О редакторе

Вишванатх Музумдар обладает более чем восьмилетним опытом консультирования в области информационных технологий, бизнес-анализа и разработки. Он является признанным разработчиком в среде MS Power BI. Его цель – использовать навыки в расстановке приоритетов, аналитические способности, умение управления командой и опыт работы с Microsoft Power BI для помощи компаниям в достижении организационных целей.

Оглавление

Об авторах	6
Благодарности	7
О редакторе	8
Предисловие от издательства.....	15
Предисловие	16
Для кого предназначена эта книга?	16
Структура книги	16
Как извлечь максимум из книги?.....	18
Загрузите файлы с примерами	18
Видеоверсия глав книги.....	19
Цветные изображения из книги.....	19
Условные обозначения.....	19
Связь.....	20
Отзывы	20
ЧАСТЬ I. ОБЗОР POWER PIVOT И POWER QUERY	21
Глава 1. Установка и настройка.....	23
Требования.....	24
Введение в Power Pivot	24
Power Pivot и версии Office.....	24
Введение в Power BI.....	25
Различия между версиями Power BI.....	25
Введение в Power Query.....	29
Особенности Power Query	30
Версии Office и совместимость с Power Query	30
Установка Power Query в Office 2013 и 2010	32
Повторная активация Power Query / Pivot	33
Запуск Power Query, Power Pivot и Power BI	36
Запуск Power Query из Excel.....	36
Доступ к Power Query из Power BI	36
Заключение	38

Глава 2. Основы Power Pivot, его недостатки и управление данными.....	39
Технические требования.....	39
Создание сводной таблицы.....	40
Построение модели данных в Power Pivot	46
Создание таблицы в Excel	47
Добавление таблиц в модель данных.....	50
Создание связей между таблицами.....	53
Power Query спешит на помощь	55
Создание вычисляемого столбца	55
Создание вычисляемого поля.....	60
Создание сводной таблицы из Power Pivot	61
Недостатки Power Pivot	62
Проблема 1. Множественный выбор элементов	62
Проблема 2. Предпросмотр в Power Pivot	63
Проблема 3. Вычисляемые поля	63
Проблема 4. Версии Microsoft Office	64
Заключение	64
Глава 3. Введение в интерфейс Power Query.....	65
Технические требования.....	65
Интерфейс Power Query и его основные элементы.....	65
Главная лента и вкладки	66
Навигационная панель или список запросов	67
Предпросмотр таблицы данных	70
Панель Query Settings (Параметры запроса)	72
Работа с примененными шагами	72
Исследование вкладки View (Просмотр).....	75
Использование расширенного редактора	79
Создание простого запроса	86
Изучение опций Load To... (Загрузить в...)	90
Изменение параметров загрузки данных в Excel по умолчанию	90
Загрузка запросов на рабочий лист вручную	94
Советы по профилированию данных.....	99
Профиль столбца	101
Качество столбца	102
Распределение столбцов	103
Заключение	104
Глава 4. Подключение к различным источникам данных.....	105
Технические требования.....	105
Краткое введение в теорию баз данных	105
Подключение к таблице или диапазону	108

Подключение к веб-странице	111
Подключение к реляционной базе данных	118
Подключение из Excel	119
Подключение из Power BI.....	121
Пользовательские подключения	124
Подключение к рабочей книге	126
Подключение к папке.....	130
Настройки источников данных	138
Изменение настроек в Excel	139
Заключение	141

ЧАСТЬ II. ПРЕОБРАЗОВАНИЕ ДАННЫХ В POWER QUERY

143

Глава 5. Преобразование данных посредством Power Query.....	145
Технические требования.....	145
Манипулирование данными при помощи операций сведения и отмены свертки	146
Обновление данных	150
Основные инструменты работы со строками и столбцами.....	153
Удаление столбцов.....	153
Удаление верхних или нижних строк	154
Использование столбца с индексом	154
Создание условного столбца при помощи конструкции if...then...else	155
Фильтрация данных с использованием условий And/Or (И/ИЛИ).....	158
Создание фильтров с единственным критерием	160
Создание фильтров с множественными критериями	163
Удаление дублирующихся строк.....	165
Замена пустых значений.....	167
Работа с заголовками строк	167
Разделение столбцов	168
Объединение и добавление запросов	171
Объединение посредством комбинирования	171
Объединение текста и значений в один столбец	174
Добавление таблиц	175
Группировка данных	178
Извлечение данных	181
Определение возраста по дате	181
Извлечение столбцов	182
Извлечение данных из столбцов	183
Заключение	185

Глава 6. Углубленное изучение запросов и функций Power Query	187
Технические требования	187
Использование функции IF в Power Query	188
Создание таблицы параметров для запросов	193
Ежемесячное изменение источника данных	197
Введение в функции Index и Modulo	202
Знакомство с функцией Modulo	202
Введение в индексные функции	205
Добавление нескольких файлов	214
Добавление данных из нескольких вкладок	219
Заключение	223
Глава 7. Автоматизация отчетов в Power Query	225
Технические требования	225
Введение в режимы хранения и типы наборов данных	225
Изменение режима хранения информации в Power BI Desktop	226
Установка режима хранения Import	229
Где Power BI хранит данные	232
Определение статуса Microsoft SQL Server Analysis Services	232
Введение в типы обновлений в Power BI	233
Обновление подключения OneDrive	234
Просмотр и выполнение обновлений в OneDrive	237
Настройка запланированных обновлений	240
Инкрементальное обновление	241
Автоматическое обновление страницы	241
Обновление потоков данных	242
Заключение	242
Глава 8. Создание дашбордов при помощи Power Query	243
Технические требования	244
Создание базовой сводной таблицы и сводной диаграммы	244
Использование Power BI для сбора и подключения к данным	248
Объединение файлов	253
Использование Power BI для добавления таблиц в модель данных	260
Выбор типа визуализации данных и подходящей диаграммы	264
Сохранение, публикация и распространение дашбордов	272
Распространение дашбордов	276
Практические рекомендации	283
Заключение	284

ЧАСТЬ III. ИЗУЧАЕМ ЯЗЫК М	285
Глава 9. Работаем с языком М	287
Технические требования	287
Знакомство с языком М	288
Основы синтаксиса языка М	288
Ключевое слово #shared для получения функций из библиотеки	291
Текстовый тип данных	291
Числовой тип данных	292
Списки	293
Записи	294
Табличный тип данных	295
Поиск данных	296
Импорт файла CSV посредством языка М	297
Заключение	299
Глава 10. Примеры использования языка М	301
Технические требования	301
Объединение данных с использованием конкатенации	302
Преобразование типов данных	307
Установка и настройка SQL Server	309
Установка SQL Server Management Studio	315
Использование параметров	320
Параметризация источника данных	320
Изменение параметров на вкладке Data (Данные)	332
Заключение	333
Глава 11. Создание базовых пользовательских функций	335
Технические требования	335
Создание функций в языке М	335
Изменение пути к файлам в запросе на локальную папку	337
Создание функции	338
Проверка параметрического запроса	340
Создание столбца с датой и временем с использованием трех функций языка М	347
Заключение	355
Глава 12. Различия между DAX и М	357
Технические требования	357
Особенности и функционал языков DAX и М	357
Синтаксис языка DAX	359
Использование формул DAX в Excel	360

Использование IntelliSense	361
Написание формул DAX	362
Введение в движок формул и движок хранилища DAX	363
Создание вычисляемого столбца	363
Создание вычисляемых мер	365
Использование быстрых мер	366
Создание меры на языке DAX с нуля	368
Организация мер	373
Заключение	374
Предметный указатель	375

Предисловие 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 – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится 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 присыпать по адресу dmkpress@gmail.com.

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

Предисловие

Power Query представляет собой технологию подключения к данным, позволяющую соединяться с источниками, а также преобразовывать и очищать информацию для ее дальнейшего анализа. Читая данную книгу, вы совершиете путешествие в мир Power Query, которое начнется с описания недостатков других программных средств в отношении управления и анализа данных. Затем мы погрузимся в интерфейс Power Query и научимся подключаться к источникам данных, извлекать, преобразовывать и очищать информацию при помощи мощных инструментов, после чего познакомимся с языком формул M, который откроет нам целый новый мир в области агрегирования данных. В завершение путешествия мы научимся строить дашборды и многомерные отчеты при помощи Power Query.

Для кого предназначена эта книга?

Данная книга отлично подойдет профессионалам в области бизнес-аналитики, анализа данных, а также пользователям Excel, стремящимся вывести свое владение этим инструментом на новый уровень посредством изучения принципов сбора и преобразования данных при помощи Power Query.

Структура книги

Глава 1. Установка и настройка. Сегодня Power Query интегрирован во все аналитические инструменты от Microsoft, такие как Excel, Analysis Services и Power BI. При помощи него можно исследовать, преобразовывать и очищать данные из различных источников. Во вводной главе вы узнаете, как установить и получить доступ к необходимым инструментам при использовании разных версий офисных программ.

Глава 2. Основы Power Pivot, его недостатки и управление данными. В этой главе вы познакомитесь с инструментом Power Pivot со всеми его недостатками при работе с разнородными данными. В качестве альтернативы будет предложен Power Query, позволяющий комфортно извлекать информацию и представлять ее для анализа в так или иначе трансформированном виде. Мы рассмотрим конкретные задачи, на примере которых будут продемонстрированы различия между Power Query и Power Pivot, а также научимся преобразовывать данные с рабочего листа в таблицу.

Глава 3. Введение в интерфейс Power Query. В данной главе вы познакомитесь с интерфейсом Power Query. Мы пробежимся по вкладкам, создадим простой запрос и посетим вкладку **View (Просмотр)** в Power BI, в которой можно настроить отображение данных. Также мы рассмотрим процесс отправки данных обратно в рабочую книгу Excel.

Глава 4. Подключение к различным источникам данных. В четвертой главе книги будет подробно описан процесс подключения к разнообразным источникам данных при помощи инструмента **Get & Transform** (Получить и преобразовать данные), также известного как Power Query. Еще мы рассмотрим разные варианты настройки источников данных.

Глава 5. Преобразование данных посредством Power Query. В этой главе вы научитесь видоизменять табличные данные, включая работу со строками, столбцами и целыми таблицами, при помощи разных инструментов Power Query.

Глава 6. Продвинутые запросы и функции Power Query. Здесь мы сосредоточимся на более углубленных запросах и функциях Power Query, таких как IF, Index и Modulo. Вы научитесь создавать параметры для изменения пути к запросу, а также добавлять сразу несколько файлов и листов.

Глава 7. Автоматизация отчетов в Power Query. В данной главе мы обсудим инструменты, предлагаемые Power Query для упрощения и автоматизации отчетов из нескольких источников. Мы рассмотрим процесс сведения информации из нескольких файлов, расположенных в одной папке, в единый набор данных, который будет обновляться при добавлении новых данных в исходную папку.

Глава 8. Создание дашбордов при помощи Power Query. Здесь мы погрузимся в мир дашбордов. Дашборды представляют собой собранные на одной странице визуализации на основе табличных данных, позволяющие рассказать историю и выделить важную для компании информацию. В этой главе вы научитесь создавать дашборды на основе данных, к которым выполнено подключение, выбирать нужные вам типы визуализаций и публиковать дашборды. Также мы затронем тему создания многомерных отчетов.

Глава 9. Работаем с языком M. В данной главе вы познакомитесь с языком программирования M и его синтаксисом. В качестве введения мы расскажем о том, как язык M получил свое название и что оно означает. Далее мы подробно поговорим о структуре и синтаксисе языка. Все языки программирования обладают своей строгой структурой, и для хорошего понимания языка просто необходимо ее усвоить. Мы рассмотрим основные типы данных и функции языка M с наглядной демонстрацией использования всех типов, после чего попробуем вместе импортировать CSV-файл при помощи языка M.

Глава 10. Примеры использования языка M. В данной главе мы начнем применять язык M на практике. Мы рассмотрим разные функции языка, включая функцию конкатенации. Для этого мы сначала отметим различия между соответствующими формулами в Excel и Power BI, после чего рассмотрим применение оператора амперсанд (&). Мы также посмотрим, как могут быть использованы функции Text.From и Text.Combine для объединения строк, дат и столбцов. В этой же главе мы расскажем о том, как настроить совершенно легальную версию SQL Server с полной функциональностью для некоммерческого использования. Мы скачаем и установим базу данных AdventureWorks, с которой в дальнейшем сможем работать. В заключение рассмотрим **параметры** и способы их эффективного использования при выполнении фильтрации источников данных, добавлении в управляемые операторы и сортировке объектов и столбцов по возрастанию или убыванию. Также расскажем, как можно производить эти изменения в Power BI.

Глава 11. Создание базовых пользовательских функций. В этой главе вы научитесь писать собственные функции на языке M в Power Query и создавать столбцы с датой и временем при помощи функций.

Глава 12. Различия между DAX и M. В заключительной главе книги мы поговорим о различиях между языками M, который используется в рамках Power Query для доступа к различным источникам данных, и DAX, позволяющим работать с табличными данными наподобие Excel. Мы покажем разницу между двумя языками на примерах, включая создание вычисляемой меры.

КАК ИЗВЛЕЧЬ МАКСИМУМ ИЗ КНИГИ?

Мы предполагаем, что вы являетесь уверенным пользователем Excel и умеете писать и анализировать формулы и функции. В наших глазах вы обладаете полным спектром знаний того, что предлагает Excel по умолчанию, и готовы сделать следующий шаг и больше узнать об анализе данных и управлении данными как об эффективном бизнес-решении. В следующей таблице приведен список программного обеспечения, используемого на протяжении книги:

Таблица 1. Программное обеспечение

Программное обеспечение, используемое в книге	Требования к операционной системе
Office 2016 или выше Office 2010 поддерживается, но потребует установки дополнений	Windows
Power Query, Power Pivot, Power BI Desktop / Power BI Online (если у вас есть подписка)	Windows
SQL Server	Windows SQL Server
Microsoft SQL Server Management Studio	Windows
Microsoft SQL Server Administration Manager	Windows
SQL – AdventureWorks	https://github.com/Microsoft/sql-server-samples/blob/master/license.txt

Если вы читаете книгу в электронном формате, мы советуем вам вводить код самостоятельно или скачать в репозитории GitHub (ссылка будет дана в следующем разделе). Это позволит вам избежать возможных ошибок, связанных с копированием и вставкой текста.

ЗАГРУЗИТЕ ФАЙЛЫ С ПРИМЕРАМИ

Вы можете загрузить файлы с примерами из этой книги со своего аккаунта на сайте <http://www.packt.com>. Если вы приобрели книгу в другом месте, то можете зарегистрироваться в разделе <http://www.packtpub.com/support> и получить файлы по почте.

Для загрузки файлов необходимо сделать следующее.

1. Войдите или зарегистрируйтесь на сайте <http://www.packt.com>.
2. Перейдите на вкладку **Support**.

3. Щелкните по ссылке **Code Downloads**.

4. Введите название книги в поле **Search** и следуйте инструкциям на экране.

После загрузки файлов вам необходимо будет распаковать их с использованием последней версии одной из следующих программ:

- WinRAR / 7-Zip для Windows;
- Zipeg / iZip / UnRARX для Mac;
- 7-Zip / PeaZip для Linux.

Код из книги также находится на GitHub по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>. Все обновления, которые будут сделаны в коде, появятся в репозитории GitHub.

Также мы отдельно храним все коды из наших книг и видеоуроков по адресу: <https://github.com/PacktPublishing>.

ВИДЕОВЕРСИЯ ГЛАВ КНИГИ

Видеофрагменты из глав данной книги можно посмотреть по адресу: <https://bit.ly/2ZeYxfb>.

ЦВЕТНЫЕ ИЗОБРАЖЕНИЯ ИЗ КНИГИ

Также мы разместили на нашем сайте файл PDF с полным набором цветных изображений и диаграмм из книги. Скачать файл можно по адресу: https://static.packt-cdn.com/downloads/9781839219719_ColorImages.pdf.

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

На протяжении книги мы будем использовать ряд условных обозначений.

Код в тексте: так мы будем обозначать вставленный в текст код из листинга, названия таблиц из базы данных, названия папок, имена файлов, их расширения, пути, ссылки и текст для ручного ввода.

Вот пример: базовый синтаксис языка M относительно прост, если правильно употреблять ключевые слова `let` и `in`.

Блоки кода мы будем обозначать следующим образом:

```
let
    Source = ""
in
    Source
```

Жирный шрифт: так мы будем выделять новые термины, важные слова или слова, которые вы видите на экране. Например, жирным шрифтом будут обозначены пункты меню или диалоговых окон. Пример: имя переменной также представлено на панели **APPLIED STEPS** (Примененные шаги) в правой части экрана.

Советы и важные заметки

Будут показаны так.

Связь

Мы всегда рады обратной связи от наших читателей.

Отзывы: если у вас есть вопросы по книге, укажите название книги в теме письма и отправьте его по адресу customercare@packtpub.com.

Ошибки: несмотря на многочисленные проверки правильности написанного, ошибки и опечатки всегда могут попадаться. Если вы обнаружили неточность в книге, мы были бы вам признательны за указание на нее. Перейдите на страницу <http://www.packtpub.com/support/errata>, введите название книги и отправьте нам сведения об ошибке.

Пиратство: если вам попались нелегальные копии наших работ в интернете, мы были бы вам благодарны за ссылки и названия сайтов, где вы их обнаружили. Вы можете отправить сообщения о таких нарушениях на адрес: copyright@packt.com.

Если вы хотите стать автором: если вы являетесь экспертом в той или иной области и хотите написать книгу или стать соавтором, посетите следующий раздел сайта: <http://authors.packtpub.com>.

Отзывы

Пожалуйста, оставьте свой отзыв о книге. Если вы прочитали книгу и использовали полученные знания на практике, почему бы не оставить отзыв о ней на сайте продавца? Потенциальные читатели смогут ознакомиться с вашим непредвзятым мнением и сделать вывод о том, стоит ли приобретать данную книгу. Обратная связь также поможет издательству понять, что вы думаете о выпускаемых книгах, а авторы будут рады прочитать отзыв о своей работе. Спасибо!

Больше информации об издательстве Packt можно почерпнуть на сайте: <http://packt.com>.

Часть I

Обзор Power Pivot и Power Query

Это вводная часть книги. В ней вы научитесь получать доступ к Power Query из разных версий Microsoft Excel, а также установите движок Power BI. Мы рассмотрим инструмент Power Pivot, упомянем все его недостатки и особенно остановимся на трудностях, с которыми сталкиваются пользователи Excel при очистке данных посредством Power Pivot. После этого мы представим вам Power Query и объясним на примере, как именно он способен нивелировать недостатки, характерные для Power Pivot. После знакомства со всеми тремя инструментами семейства Power – Power Query, Power Pivot и Power BI – мы поговорим о базовых возможностях первых двух: Power Query и Power Pivot.

Первая часть книги будет разбита на следующие главы:

- глава 1 «Установка и настройка»;
- глава 2 «Основы Power Pivot, его недостатки и управление данными»;
- глава 3 «Введение в интерфейс Power Query»;
- глава 4 «Подключение к различным источникам данных».

Глава 1

Установка и настройка

Инструмент Power Query может быть полезен в самых разных сферах работы, а на рис. 1.1 мы привели основные причины его использования при подготовке данных для анализа, управления и визуализации.



Рис. 1.1. Возможности Power Query

Сегодня Power Query интегрирована во все инструменты аналитики от Microsoft, такие как Excel, Analysis Services и Power BI. Это позволяет пользователям исследовать, преобразовывать и очищать данные, полученные из разных источников.

В данной главе мы рассмотрим пошаговую процедуру доступа к данным при помощи Power Query в разных версиях Microsoft Excel, а также установим инструмент Power BI. Кроме того, мы поговорим о недостатках Power Pivot, связанных с очисткой данных в Excel, и предложим альтернативное решение в виде Power Query.

В этой главе будут затронуты четыре основные темы:

- введение в Power Pivot, Power BI и Power Query;
- различия между версиями Office и инструментами семейства Power;
- инструкция по работе в разных версиях Excel;
- запуск Power Query, Power Pivot и Power BI.

ТРЕБОВАНИЯ

Желательно, чтобы читатель книги обладал как минимум средним уровнем знаний в области Microsoft Excel, умел работать со строками, столбцами, рабочими листами и книгами, а также писать формулы и функции. Будет плюсом, если он также знаком с построением диаграмм и изменением их элементов.

Сопроводительные материалы можно загрузить с GitHub по следующему адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

Видеофрагмент, соответствующий этой главе книги, находится по адресу: https://www.youtube.com/watch?v=cxKvtorqP1Q&list=PLLeLcvrwLe1860_GJEzs47WaZXwZjwTN83&index=2&t=4s.

ВВЕДЕНИЕ В POWER PIVOT

Очень важно, чтобы вы хорошо понимали термин *модель данных* (data model) перед погружением в работу с Power Pivot. Модель данных образуется путем объединения двух и более таблиц по соответствующим полям или столбцам. Если вам доводилось работать с базами данных наподобие Microsoft Access, значит, вы понимаете, что имеется в виду под связями между таблицами. Объединение таблиц из одного или нескольких источников в едином источнике данных называется моделью данных, как показано на рис. 1.2.



Рис. 1.2. Представление связанных таблиц

Инструмент *Power Pivot* является составной частью семейства Power BI – его мозгом, поскольку в его функции входит моделирование данных, обработка, вычисление и анализ. Если использовать аналогию с автомобилем, то можно назвать это двигателем с моделью данных, хранящей всю доступную информацию. Он способен обрабатывать огромные наборы информации в созданной многотабличной модели данных, после чего может использоваться в качестве источника данных, например при создании сводных таблиц. Работая с данными на рабочем листе Excel, вы можете использовать Power Pivot для построения модели данных и последующего создания связей между таблицами для выполнения расчетов любой степени сложности непосредственно в среде Excel. Power Pivot помогает при моделировании данных, установке связей между таблицами, создании вычисляемых столбцов, мер и показателей эффективности, а также может использоваться для создания куба.

Power Pivot и версии Office

Сначала мы приведем список совместимых версий Office, а затем рассмотрим различия между ними.

Power Pivot Office 2019 (Office 365), 2016 и 2013

Раньше Power Pivot был доступен только в версиях Office Pro Plus 2013/2016, но не так давно компания Microsoft добавила этот инструмент к следующим версиям ПО: Office 365 Home, Office 365 Personal, Office 365 Business Essentials, Office 365 Business, Office 365 Business Premium и Office 365 Enterprise E1.

В этих версиях Power Pivot установлен по умолчанию, но иногда может потребоваться его активация. Об этом мы расскажем позже в разделе «Повторная активация Power Query / Pivot».

Power Pivot Office 2010

Надстройка Power Pivot не была интегрирована в версию Office 2010, но может быть бесплатно загружена и установлена по адресу: <https://www.microsoft.com/en-us/download/details.aspx?id=43348>. У надстроек Power Pivot для Office 2010 было две версии, ожидаемо названные *version* и *version 2* соответственно. Они были написаны в виде плагинов для разработки SQL Server 2012. При загрузке вы увидите следующую надпись: **Microsoft® SQL Server® 2012 SP2 PowerPivot for Microsoft Excel® 2010**. Если у вас нет SQL Server, не беспокойтесь – именно эта версия понадобится вам для установки в Excel 2010. Заметим, что размер скачиваемого файла будет довольно большим.

ВВЕДЕНИЕ В POWER BI

Инструмент *Power BI* предлагается в качестве облачной платформы или *программного обеспечения как услуги* (Software as a Service – SaaS), что позволяет удовлетворить все бизнес-требования заказчиков. Огромным преимуществом такого подхода является возможность без проблем обрабатывать миллионы строк данных – вы можете моделировать и анализировать данные путем объединения таблиц. Это позволяет один раз определить формулу, а затем многократно использовать ее при управлении данными!

Power BI позволяет представлять информацию в удобном для пользователя виде – при помощи создания отчетов и дашбордов. Этот инструмент доступен в облаке, где пользователи могут загружать данные и обмениваться информацией между собой, выполняя запросы к модели данных на естественном языке. Автономная версия инструмента является бесплатной и включает в себя все три приложения: Power Query, Power View и Power Pivot. Power View позволяет создавать интерактивные визуализации посредством переноса элементов мышью в отчет.

Различия между версиями Power BI

Для начала мы приведем список совместимых версий продукта для Windows и macOS, а затем рассмотрим отличия между ними.

Windows

Существует три разные версии Power BI, доступные для загрузки. Давайте по ним пройдемся.

Power BI Desktop / Free

Версия Power BI Desktop, которую часто называют Power BI Free, предназначена для компаний малого и среднего размера. При помощи нее вы можете подключаться к 70 видам источников данных, публиковать отчеты в интернете и экспортить данные в Excel.

Вы можете загрузить эту версию по адресу <https://www.microsoft.com/en-us/download/details.aspx?id=58494> или с Microsoft Store: <https://aka.ms/pbidesktopstore>.

Примечание

Системные требования Power BI Desktop довольно лояльны, и большинство компьютеров им будет удовлетворять: Windows 7 или выше, Internet Explorer 10 или выше, 1 Гб RAM, .Net 4.5 и частота CPU как минимум 1 ГГц. Что вас может удивить, так это то, что разрешение вашего монитора по умолчанию 1024×768 или 1280×800 может оказаться недостаточным для Power BI. Рекомендуемыми разрешениями являются 1440×900 или 1600×900, что обусловлено требованиями некоторых элементов управления.

При загрузке вы также должны указать разрядность вашей операционной системы: 32 или 64 бит.

Заметим, что разрядность касается исключительно данного приложения и может отличаться от разрядности Office.

Загрузите подходящий вам установочный файл Power BI Desktop, после чего запустите его. В окне установки обратите внимание на сообщение о том, какие данные программа Power BI будет собирать при работе. Вы имеете право отказаться от этого, для чего нужно щелкнуть в соответствующем месте установочного окна.

Далее выберите место на диске, куда будет установлен Power BI. Лично я предлагаю оставлять путь по умолчанию. В конце концов, если у вас возникнут проблемы с поиском программы на диске, вам будет легче найти его, используя путь по умолчанию. При установке в другую папку на диске это может быть проблематично.

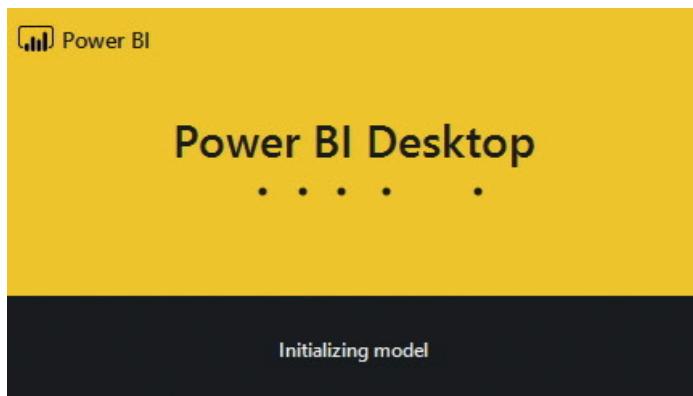


Рис. 1.3. Окно загрузки Power BI Desktop

На этапе первой загрузки Power BI вам будет предложено подпісаться на новостную рассылку, в которой вам будут приходить советы и тонкости использования инструмента.

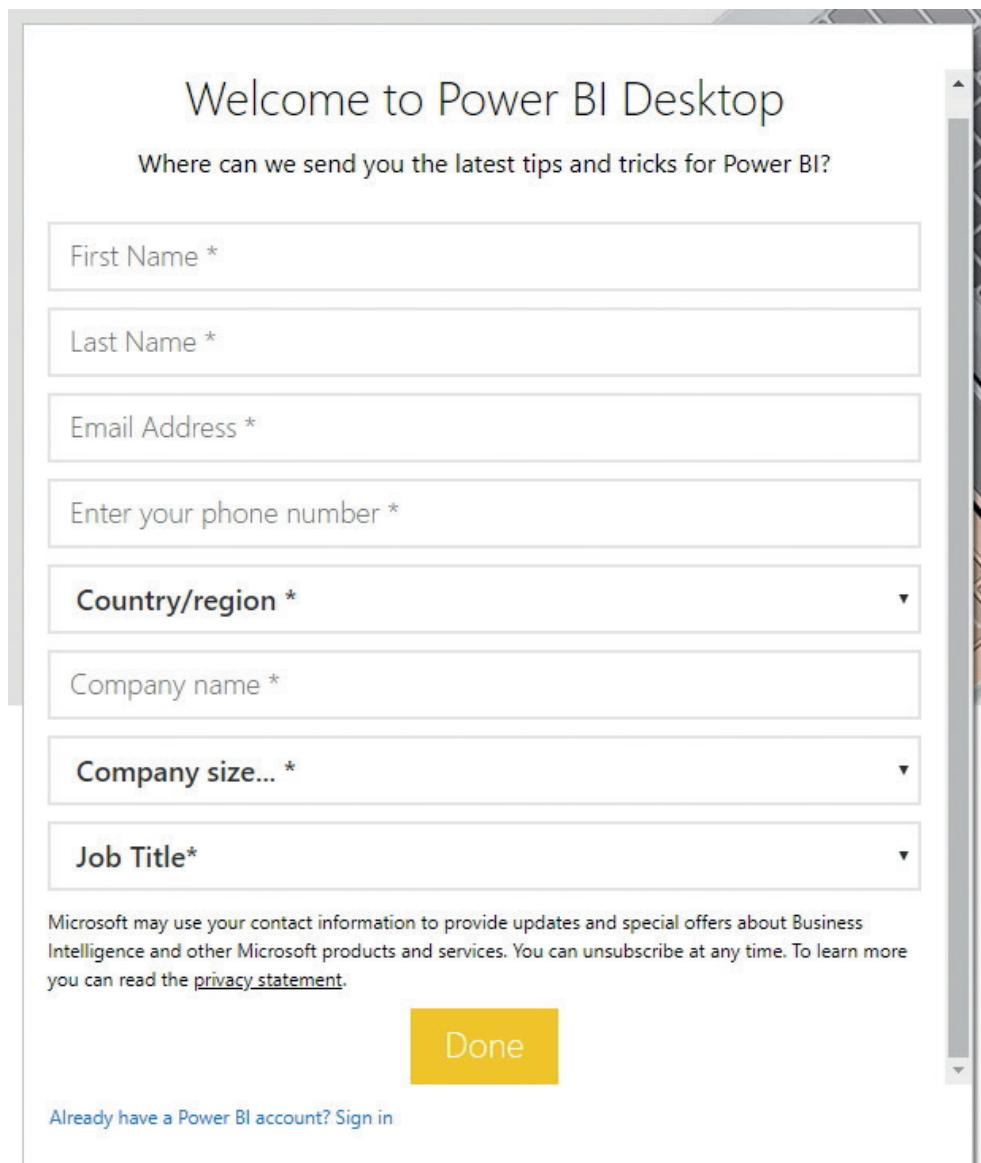


Рис. 1.4. Окно регистрации Power BI Desktop

У вас также есть возможность авторизоваться (используйте эту ссылку для бесплатной пробной подписки на Power BI: <https://powerbi.microsoft.com/en-us/>) или зарегистрироваться.

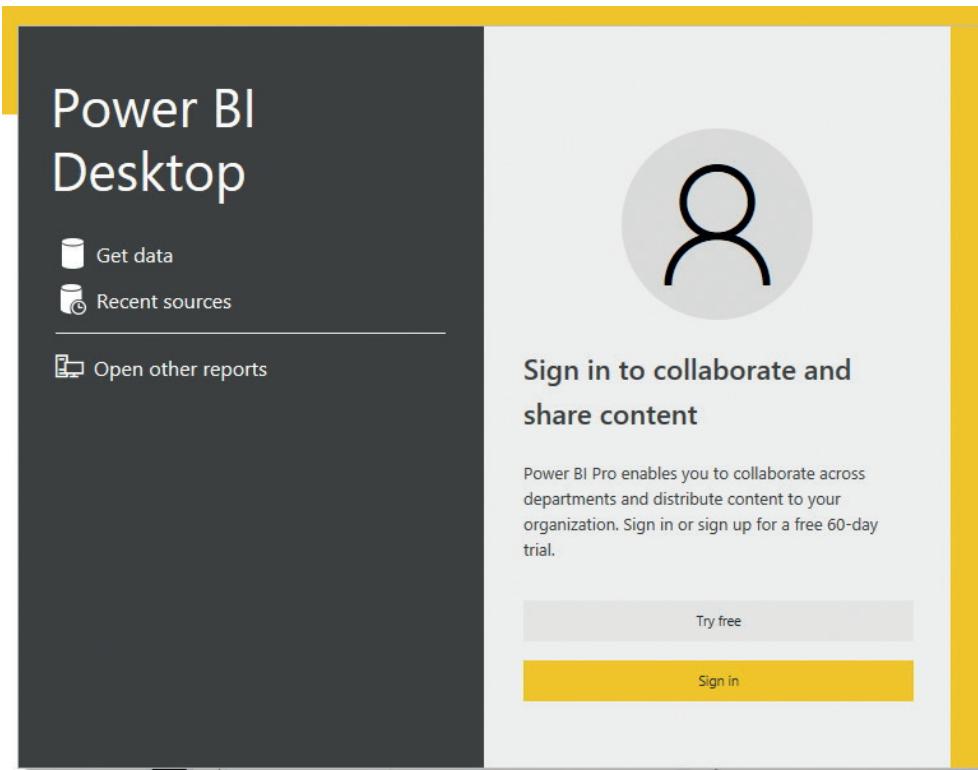


Рис. 1.5. Окно авторизации Power BI Desktop

После этого вы увидите окно загрузки программы, запуск которого вы можете отключить, сняв соответствующий флажок **Show this screen on startup** (Показывать окно загрузки при запуске).

Нам кажется, что легче всего установить Power BI Desktop путем загрузки из Windows Store (доступно для Windows 8 и выше). Единственная проблема в этом случае состоит в том, что вам придется довольствоваться настройками по умолчанию. Так что если вам необходимо что-то изменить, лучше будет выбрать первый способ.

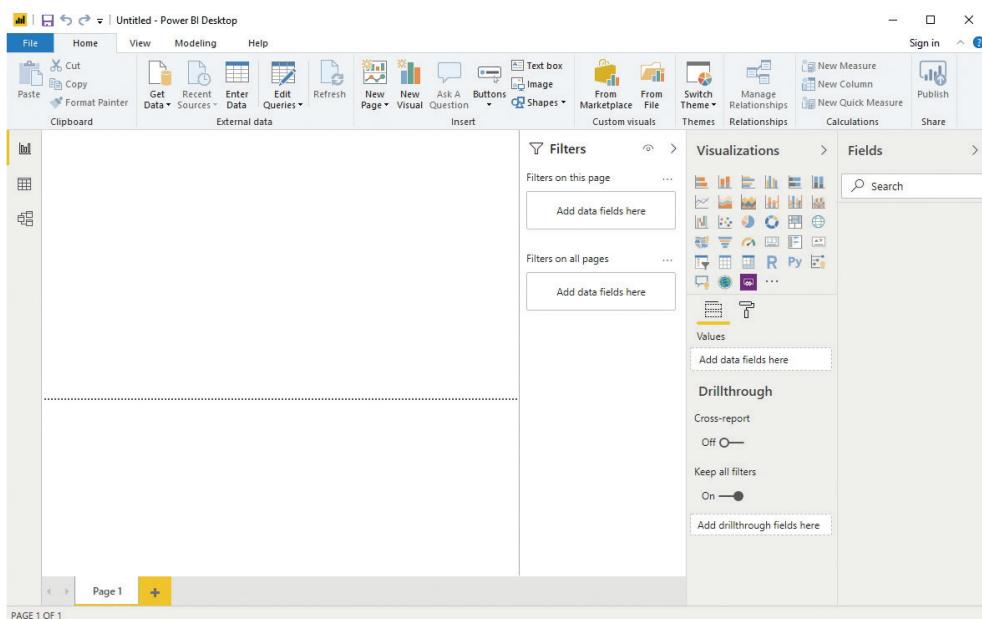


Рис. 1.6. Главный экран Power BI Desktop

Mac/Apple

К сожалению, инструмент Power BI может быть запущен только в Windows. Если у вас Mac, вы не сможете напрямую установить Power BI – придется воспользоваться утилитой Boot Camp, создать локальную виртуальную машину (VM) или использовать стороннее ПО. Также вы можете попробовать для этих целей программы вроде Turbo.net (<http://turbo.net>) или Parallels.

Что касается мобильных платформ, то вы можете установить приложение Power BI на свой iPhone или iPad из App Store по ссылке: <https://apps.apple.com/us/app/microsoft-power-bi/id929738808>.

ВВЕДЕНИЕ В POWER QUERY

Power Query также входит в семейство Power BI и позволяет вам извлекать и загружать данные из огромного количества источников, включая файлы, базы данных и страницы в интернете. Можно рассматривать Power Query как сборщик или преобразователь данных, ведь зачастую в Power Pivot информация поступает именно из него. Работая с Excel, вы можете создать модель данных путем добавления таблиц, загруженных из разных источников посредством Power Query. Для этого нужно установить переключатель **Add to Data Model** (Добавить в модель данных) во время импорта. Затем вы используете Power Pivot для создания связей между таблицами. Вы можете импортировать источники данных как в Power Query, так и в Power Pivot, но Power Query является более предпочтительным вариантом, поскольку предоставляет больше выбора, и файл будет занимать меньше места на диске.

Можно импортировать данные в существующие таблицы Power Pivot или Excel, после чего выполнить преобразование данных, прежде чем использовать их в Excel. Инструмент Power Query доступен в Excel и Power BI, и преобразованные данные могут быть открыты непосредственно в этих программах. Power Query дает возможность подключаться и извлекать данные, настраивать условия, объединять и комбинировать, добавлять и изменять данные, трансформировать их и публиковать. В Power Query используется собственный язык M, который мы изучим в соответствующей главе.

Особенности Power Query

Перед созданием дашбордов в Excel вам необходимо воспользоваться помощью Power Query. Лист Excel ограничен 1 048 576 строками, и если попытаться загрузить более объемные данные из источника, Excel просто не сможет завершить операцию или загрузит столько записей, сколько сможет, что также может оказаться на производительности. Чтобы решить эту проблему, необходимо воспользоваться Power Query, позволяющим вам подключаться непосредственно к источникам данных. Это означает, что данные не будут физически сохраняться в файле Excel, а останутся в источнике.

Power Query дает возможность избежать монотонных действий, повторяющихся снова и снова, например запуска макроса или задачи. Этот инструмент позволяет создавать именованные запросы, содержащие последовательность шагов, которые будут применяться к набору данных, и при этом вам совсем не обязательно уметь программировать. Это способствует снижению затрат и времени для компании, а также исключает необходимость обращаться к сторонним разработчикам.

Еще одной полезной особенностью Power Query является его способность разворачивать данные таким образом, чтобы вы могли создавать несколько отчетов со сводными таблицами, преобразуя данные в табличный формат. Также это позволяет объединять данные из разных таблиц в единую сводную таблицу. Это исключительно полезно, если требуемые вам данные содержатся в разных файлах Excel, создаваемых различными подразделениями компании.

Еще одной причиной использования Power Query является его способность импортировать текстовые файлы, расположенные в одной папке. Этот инструмент быстро и без особыго труда соберет информацию из всех файлов в единый файл Excel. Кроме того, вы можете просто подключаться к ним, что позволит сэкономить немало времени при создании сводных таблиц.

Power Query представлял собой бесплатную надстройку в ранних версиях Excel, а в поздние версии он интегрирован по умолчанию. В следующих разделах данной главы вы узнаете, какие версии Office поддерживают Power Query и где можно скачать этот инструмент для более старых версий Microsoft Excel.

Версии Office и совместимость с Power Query

Сначала мы перечислим совместимые версии Windows и macOS, а затем пройдемся по отличиям в них касательно применения Power Query.

Windows/Android

В следующих версиях программного обеспечения есть полноценная поддержка Power Query:

- Microsoft Office 2019, 2016 и 2013: все версии;
- Microsoft Office 2010 Professional Plus с Software Assurance;
- Power Query Premium: все возможности Power Query доступны для Professional Plus, Office 365 ProPlus и Excel 2013 Standalone;
- Power Query Public: доступен для всех версий Office 2013 Desktop. Включены все возможности Power Query, за исключением Corporate Power BI Data Catalog, источников данных на базе Azure, Active Directory, HDFS, SharePoint Lists, Oracle, DB2, MySQL, PostgreSQL, Sybase, Teradata, Exchange, Dynamics CRM, SAP BusinessObjects и Salesforce;
- **Get & Transform (Power Query)** в настоящее время не поддерживается Android, iOS и Online.

Mac

Поддерживаемые версии для macOS:

- Excel 2011 и Excel 2016 для Mac: Get & Transform (Power Query) не поддерживается;
- Excel для Office 365 для Mac: если вы являетесь подписчиком Office 365 и также подписаны на программу Windows Insider, то можете обновлять существующие запросы Power Query на своем Mac.

Отличия между версиями Office 2019 (Office 365) и Office 2016

Если у вас установлен Office 2019 или Office 2016, значит, у вас уже есть Power Query и Power Pivot. В настоящее время эти инструменты располагаются в группе **Get & Transform** (Получить и преобразовать данные) на вкладке **Data** (Данные).

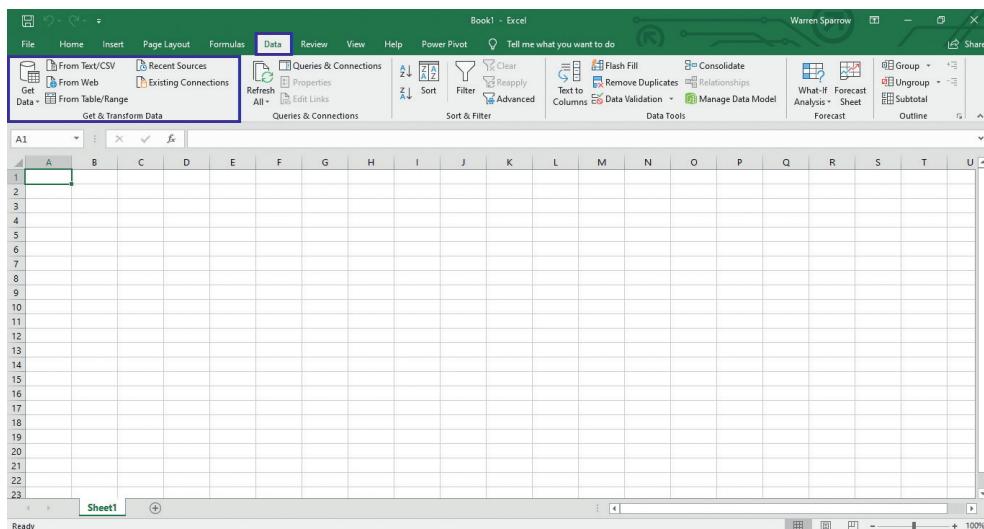


Рис. 1.7. Окно Office 2019

А вот как выглядит окно Office 2016:

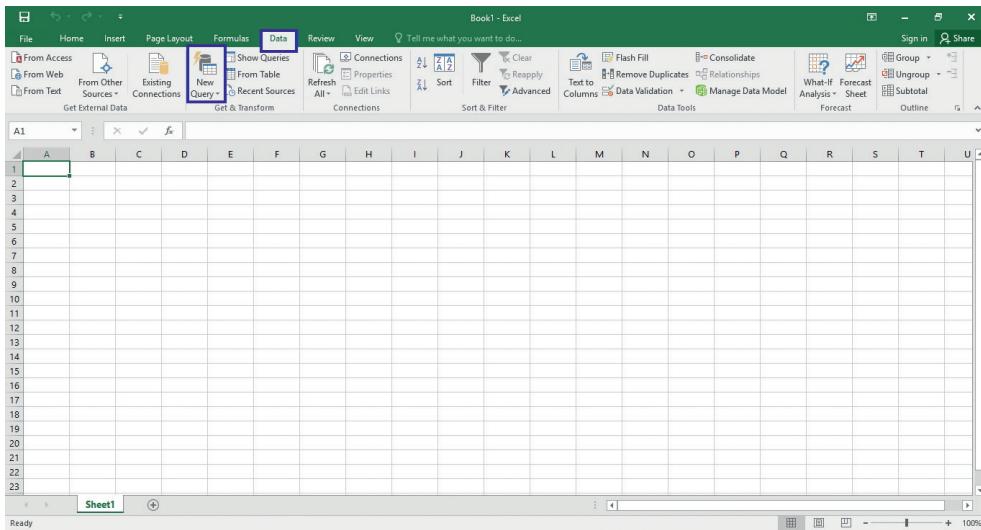


Рис. 1.8. Окно Office 2016

Как видите, между Office 2016 и Office 2019 есть незначительная разница в отношении расположения инструмента Power Query, но это никак не сказывается на его работе. При этом в Office 2016 ваше окно может выглядеть и так, как показано на рис. 1.9.

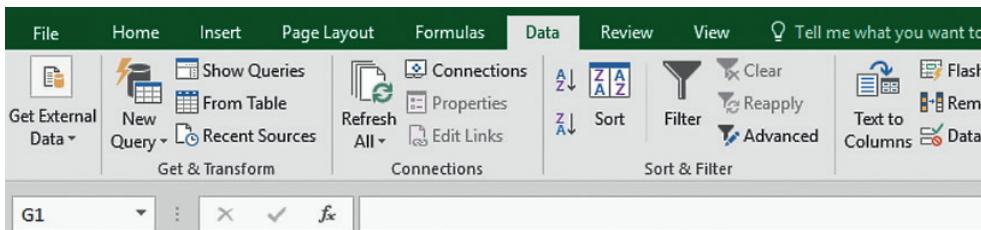


Рис. 1.9. Альтернативное окно Office 2016

Причина этого – в существовании разных версий Excel 2016: с MSI (Windows Installer), в виде подписки Office 365 и версии ProPlus. Этим объясняются незначительные различия во внешнем виде окна программы.

УСТАНОВКА POWER QUERY В OFFICE 2013 И 2010

Для указанных версий Office вам необходимо загрузить надстройку Power Query по следующей ссылке: <https://www.microsoft.com/en-us/download/details.aspx?id=39379>.

После нажатия на кнопку **Download** на этой странице вам будет задан вопрос о том, какой разрядностью обладает установленный у вас Office: 32 или 64 бит. Если вы знаете, какая у вас версия, скачивайте соответствующий файл.

Примечание

Единственное отличие между этими версиями состоит в том, что 32-битная версия может комфортно работать с количеством строк данных, близким к 2 млн. 64-битная версия без труда справляется с объемами данных, превышающими эту цифру в 20 раз. Также стоит помнить, что если вы создадите рабочую книгу в Power Pivot размером в 5 млн строк и дадите доступ к ней человеку с 32-битной версией системы, он откроет рабочую книгу, но не сможет взаимодействовать с таблицей.

После определения своей версии ПО загрузите соответствующий вариант надстройки с сайта Microsoft. Чтобы узнать версию операционной системы, можете воспользоваться следующей ссылкой: <https://support.microsoft.com/en-gb/help/13443/windows-which-version-am-i-running>.

Для установки надстройки Power Query выполните следующие действия.

- Перед тем как запускать загруженный файл, убедитесь, что Excel закрыт. Соглашайтесь со всеми предложенными условиями и жмите кнопку **Next** до завершения установки.

Совет

Для корректной установки надстройки в вашей системе должен быть установлен браузер Internet Explorer версии 9 или выше. Если у вас Internet Explorer более ранней версии, необходимо сначала обновить его, после чего продолжить установку Power Query.

- По окончании установки откройте Excel 2010 или 2013. Вы увидите новую вкладку на ленте с названием Power Query, как показано на рис. 1.10.

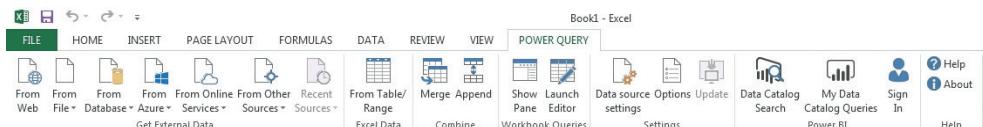


Рис. 1.10. Вкладка **Power Query** на ленте в Excel

ПОВТОРНАЯ АКТИВАЦИЯ POWER QUERY / PIVOT

Иногда случается, что в Excel пропадают вкладки Power Query или Power Pivot. Это происходит из-за проблем с загрузкой надстроек COM. Если это случилось, необходимо заново выбрать нужную вам надстройку в разделе **COM Add-ins** (Надстройки COM). Добраться до этого раздела можно разными способами, но легче всего это будет сделать так:

Примечание

Рекомендованный способ подходит как для Power Query, так и для Power Pivot. Для простоты на иллюстрации мы покажем только метод повторной активации надстройки Power Query.

1. Запустите Excel, войдите в меню **File** (Файл) и выберите пункт **Options** (Параметры). Откроется окно, показанное на рис. 1.11.

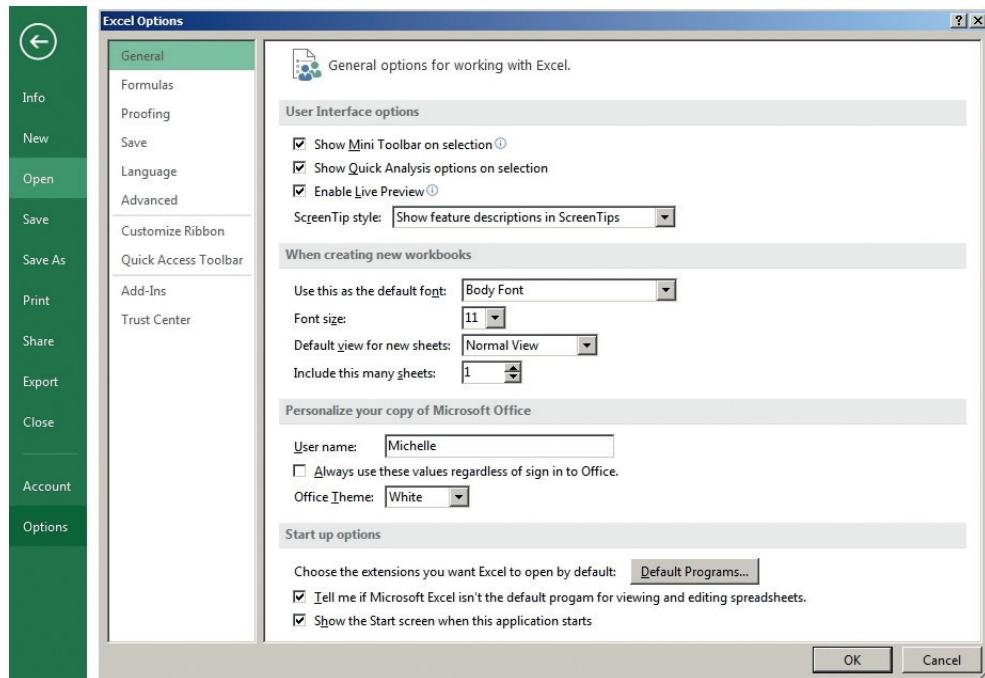


Рис. 1.11. Окно параметров Excel

2. Зайдите в пункт **Add-ins** (Надстройки) в боковом меню, а затем в выпадающем списке **Manage** (Управление) выберите **COM Add-ins** (Надстройки COM), как показано на рис. 1.12, и нажмите кнопку **Go** (Перейти).

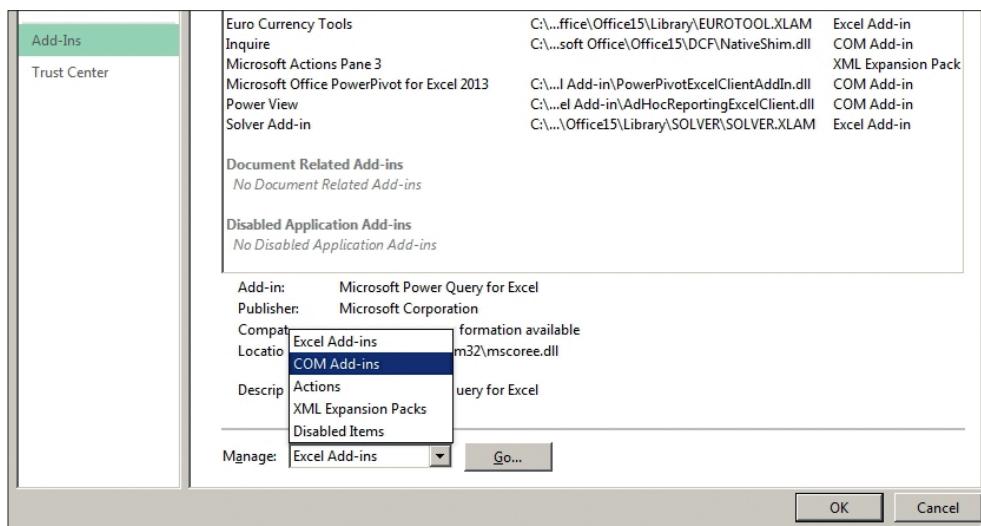


Рис. 1.12. Окно надстроек в Excel

3. Откроется диалоговое окно **COM Add-ins** (Надстройки для модели компонентных объектов (COM)), показанное на рис. 1.13, в котором вы можете установить флажок напротив пункта Power Query, если он снят.

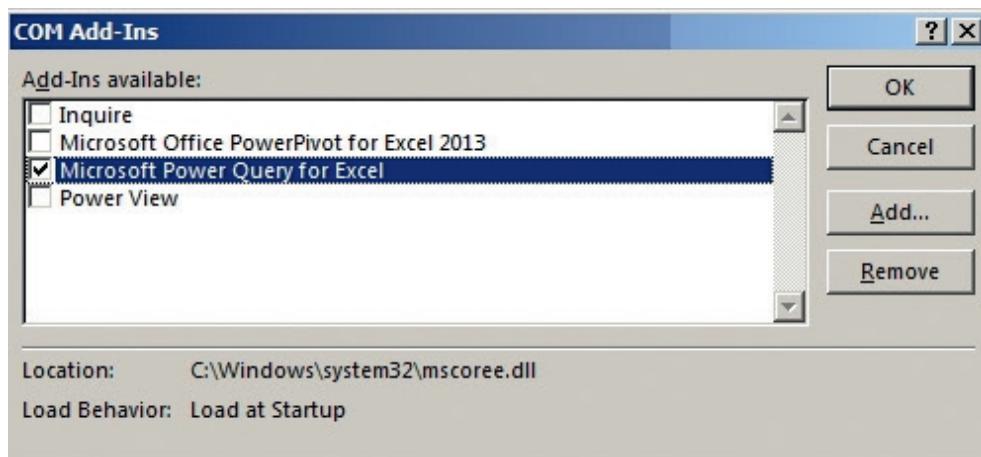


Рис. 1.13. Диалоговое окно надстроек COM в Excel

Запуск Power Query, Power Pivot и Power BI

В данном разделе вы научитесь запускать редактор Power Query (Power Query Editor) из Microsoft Excel и Power BI. Мы покажем этот процесс на примере Microsoft Office 2019.

Запуск Power Query из Excel

Чтобы запустить редактор Power Query в Excel, проделайте следующие действия.

1. Откройте Microsoft Excel 2019.
2. Перейдите на вкладку **Data** в ленте.
3. Нажмите на кнопку **Get Data** (Получить данные), как показано на рис. 1.14. Именно так осуществляется доступ к Power Query.

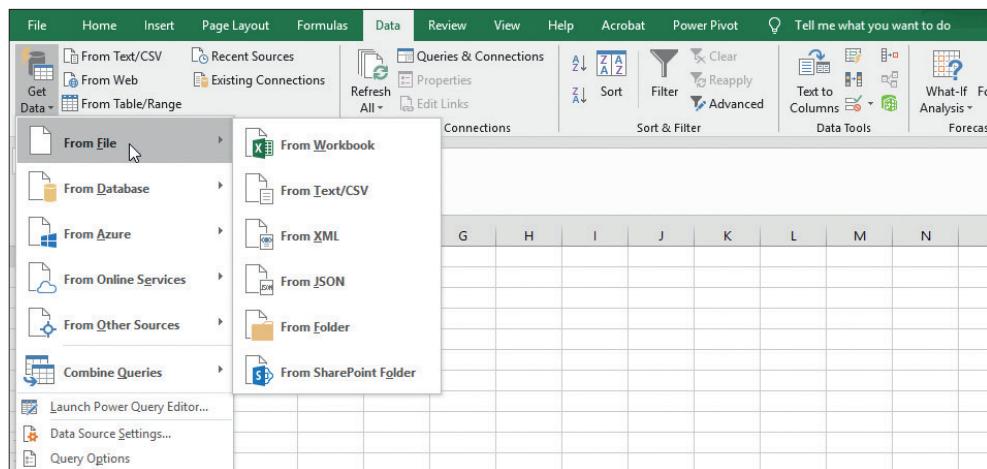


Рис. 1.14. Доступ к Power Query в Excel

4. Выберите в предложенном списке источник данных для подключения посредством Power Query.
5. Пункт **Launch Power Query Editor...** (Запустить редактор запросов) используется для выполнения запуска редактора Power Query.

Доступ к Power Query из Power BI

Выполните следующие шаги для осуществления доступа к инструменту Power Query из Power BI.

1. Запустите программу Power BI Desktop на своем компьютере, как показано на рис. 1.15.

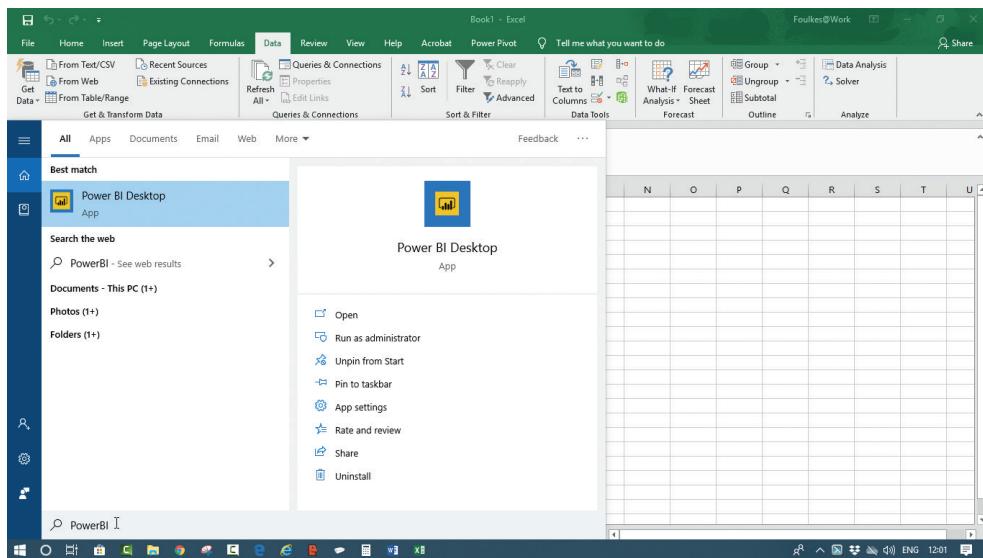


Рис. 1.15. Запуск Power BI Desktop

2. Появится окно загрузки программы, представленное на рис. 1.16.

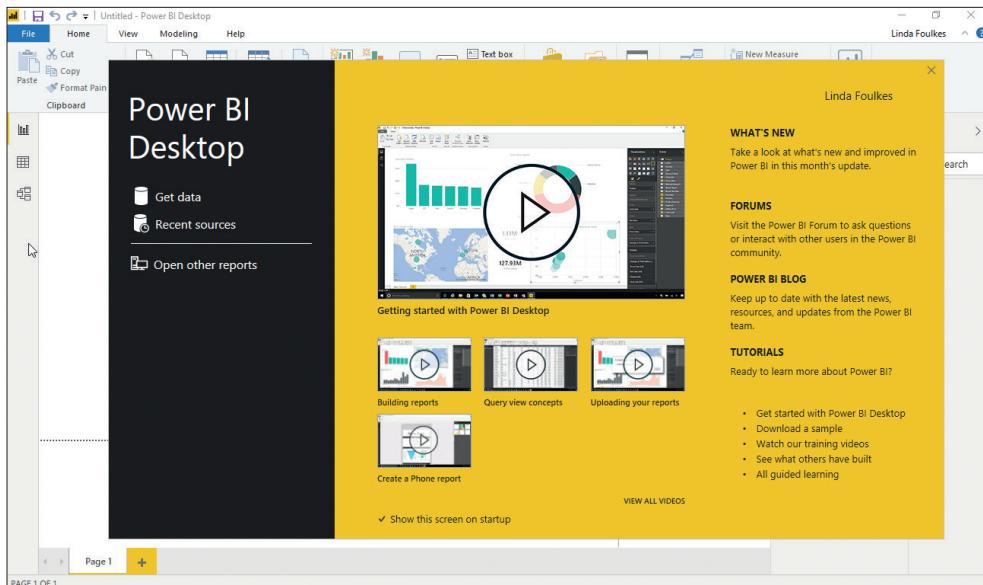


Рис. 1.16. Окно запуска Power BI Desktop

3. Прямо в окне запуска есть пункт **Get Data** (Получить данные), дающий доступ к Power Query. Если вы хотите войти в Power Query из самой программы Power BI, то закройте окно запуска.

4. Находясь в Power BI, нажмите на кнопку **Get Data** (Получить данные) в группе **Data** (Данные) на вкладке **Home** (Главная страница), как показано на рис. 1.17, чтобы получить доступ к Power Query.

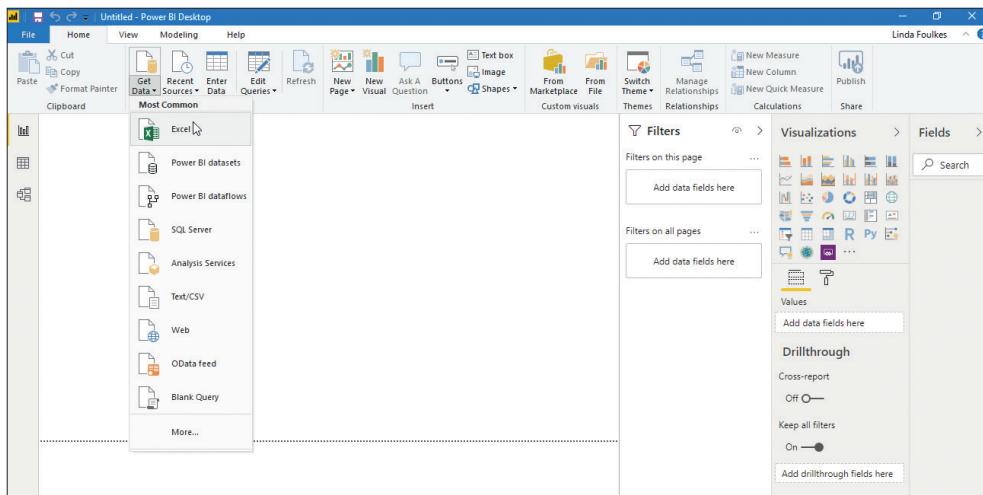


Рис. 1.17. Power BI Desktop

5. Выберите источник данных, чтобы начать работу с Power Query.

ЗАКЛЮЧЕНИЕ

В этой главе вы познакомились с тремя мощными инструментами из семейства Power, позволяющими создавать модели данных, трансформировать и преобразовывать информацию, а также создавать потрясающие дашборды. Вы узнали, что из себя представляют программные продукты Power BI, Power Query и Power Pivot, и научились встраивать эти инструменты в среду Microsoft Office Excel разных версий. Кроме того, вы получили представление о различиях этих инструментов в разных версиях Office. В конце главы вы узнали о том, как запускать Power Query при помощи пункта меню **Get Data** (Получить данные).

Во второй главе книги мы подробно остановимся на недостатках инструмента Power Pivot и объясним, почему Power Query является наиболее приоритетным выбором при необходимости извлекать и преобразовывать данные. Прочитав эту главу, вы поймете всю важность управления и анализа данных на практике. Мы рассмотрим примеры использования инструментов Power Pivot и Power Query. Вы также узнаете, как преобразовывать рабочие листы в таблицы, пригодные для импорта, посредством Power Query.

Глава 2

Основы Power Pivot, его недостатки и управление данными

Как мы уже упоминали, сегодня Power Query присутствует по умолчанию во всех последних версиях аналитических продуктов от Microsoft, включая Excel, Analysis Services и Power BI. Это способствует тому, что пользователи могут с легкостью исследовать, преобразовывать и очищать данные из самых разных источников для их дальнейшего анализа. В данной главе мы затронем следующие темы:

- создание сводной таблицы;
- построение модели данных в Power Pivot;
- создание умных таблиц в Excel;
- настройка связей между таблицами;
- Power Query спешит на помощь;
- недостатки Power Pivot.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Вам понадобится интернет-соединение для загрузки соответствующих файлов с GitHub. Файлы с кодом для этой главы можно скачать по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

Мы будем исходить из предположения о том, что вы обладаете базовыми навыками написания формул в Excel.

Видеофрагмент, посвященный данной главе, располагается по адресу: https://www.youtube.com/watch?v=5TuXxliMehY&list=PLeLcvrwLe186O_GJEZs47WaZXwJwTN83&index=3&t=0s.

В сфере бизнеса во все времена проводился углубленный анализ данных, на основании которого принимались важные управленческие решения. Сегодня на такие решения порой много времени не отводится. Люди не желают долго ждать получения информации для проведения анализа. Вместо этого они хотят получать всю необходимую сводку быстро и мгновенно принимать решения. Долгие годы бизнес-структуры использовали анализ данных для определения

главных трендов и тенденций – даже до появления модного ныне термина «большие данные» (big data). Конечно, сегодня предпринимателям приходится подвергать быстрому анализу огромные объемы информации – это позволяет им сохранять динамику и работать быстро.

Не вдаваясь в подробности того, почему аналитика играет такую существенную роль в бизнесе, поговорим о том, как способствуют проведению анализа инструменты наподобие Power Pivot и Power Query.

Одной из главных причин того, почему анализ данных приобрел такое серьезное значение в сфере бизнеса, является то, что подобный подход ведет к росту экономических показателей. Обычно данные собираются и анализируются внутри компании, при этом анализ может распространяться на все области – начиная от сотрудников и заканчивая глобальными ключевыми показателями эффективности работы организации. Благодаря сбору информации с большого количества людей и технологиям интеллектуального анализа данных, позволяющим быстро обрабатывать огромные массивы информации, процесс принятия решений значительно ускорился, что пошло на пользу компаниям, занимающимся торговлей и рекламой. И Power Query с его возможностями быстрой загрузки и обработки больших пластов данных, безусловно, помог сохраниТЬ и преумножить экономические показатели во многих сферах.

Но давайте приступим к первой теме этой главы, на которой будет основано все дальнейшее повествование.

Создание сводной таблицы

Большинство людей хотя бы раз в своей практике использовали сводные таблицы в Excel для быстрого и легкого агрегирования больших объемов информации. В качестве примера откроем в Microsoft Excel файл PowerQuery.xlsx, который вы можете загрузить с GitHub. Внешний вид файла показан на рис. 2.1.

	Month	Salesman	Region	Product	No. Customers	Net Sales	Profit / Loss
2	Jan-17	John	North	Dyno1	8	1,592	562
3	Jan-17	John	North	Dyno2	8	1,088	397
4	Jan-17	John	West	Dyno3	8	1,680	753
5	Jan-17	John	West	Dyno1	9	2,133	923
6	Jan-17	John	West	Dyno2	10	1,610	579
7	Jan-17	John	Midlands	Dyno3	10	1,500	570
8	Jan-17	John	Midlands	Dyno1	7	1,316	428
9	Jan-17	John	Midlands	Dyno2	7	1,798	708
10	Jan-17	John	Midlands	Dyno3	8	1,624	621
11	Jan-17	Lucie	North	Dyno1	6	726	236
12	Jan-17	Lucie	North	Dyno2	9	2,277	966
13	Jan-17	Lucie	West	Dyno3	6	714	221
14	Jan-17	Lucie	West	Dyno1	9	2,650	1,023
15	Jan-17	Lucie	West	Dyno2	6	1,500	634
16	Jan-17	Lucie	West	Dyno3	7	917	403
17	Jan-17	Lucie	Midlands	Dyno1	7	1,939	760
18	Jan-17	Lucie	Midlands	Dyno2	6	984	314
19	Jan-17	Lucie	Midlands	Dyno3	9	981	372
20	Jan-17	Jordan	North	Dyno1	10	1,520	476
21	Jan-17	Jordan	North	Dyno2	6	960	380
22	Jan-17	Jordan	North	Dyno3	10	2,800	900
23	Jan-17	Jordan	West	Dyno1	6	1,536	572
24	Jan-17	Jordan	West	Dyno2	8	816	291
25	Jan-17	Jordan	West	Dyno3	9	2,547	781
26	Jan-17	Jordan	Midlands	Dyno1	10	1,810	664
27	Jan-17	Jordan	Midlands	Dyno2	9	2,223	771
28	Jan-17	Jordan	Midlands	Dyno3	9	1,270	451
29	Jan-17	Clive	North	Dyno1	7	903	315
30	Jan-17	Clive	North	Dyno2	9	2,232	828
31	Jan-17	Clive	West	Dyno3	10	2,070	903
32	Jan-17	Clive	West	Dyno1	8	1,592	562

Рис. 2.1. Исходные данные для сводной таблицы

Как видите, в приведенной таблице содержится информация о продажах с указанием продавцов, регионов и товаров, которые были проданы. Давайте используем эти данные для построения нашей первой сводной таблицы. Для этого:

- Выделите всю таблицу, после чего выберите на ленте пункт меню **Insert** (Вставка) и нажмите на кнопку **Pivot Table** (Сводная таблица).

Совет

Вы можете выделить таблицу и после нажатия на кнопку создания сводной таблицы. Для этого необходимо щелкнуть по кнопке со стрелкой справа от поля **Table/Range** (Таблица или диапазон) в диалоговом окне сводной таблицы, выбрать нужный диапазон и снова щелкнуть по указанной кнопке.

- В диалоговом окне создания сводной таблицы вы также можете указать, выводить ли ее на новом листе или существующем. Мы остановимся на существующем листе, чтобы одновременно видеть и исходные данные, как показано на рис. 2.2.

Month	Salesman	Region	Product	No. Customers	Net Sales	Profit / Loss
Jan-17	John	North	Dyno3	8	583	583
Jan-17	John	North	Dyno2	8	1,088	397
Jan-17	John	West	Dyno3	8	1,680	753
Jan-17	John	West	Dyno1	9	2,133	923
Jan-17	John	West	Dyno2	10	1,610	579
Jan-17	John	Midlands	Dyno3	10	1,540	570
Jan-17	John	Midlands	Dyno1	7	1,316	428
Jan-17	John	Midlands	Dyno2	7	1,799	709
Jan-17	Lucie	North	Dyno3	8	1,624	621
Jan-17	Lucie	North	Dyno1	6	726	236
Jan-17	Lucie	North	Dyno2	9	2,277	996
Jan-17	Lucie	West	Dyno3	6	714	221
Jan-17	Lucie	West	Dyno1	9	2,682	1,023
Jan-17	Lucie	West	Dyno2	6	1,500	634
Jan-17	Lucie	Midlands	Dyno3	7	917	403
Jan-17	Lucie	Midlands	Dyno1	7	1,939	760
Jan-17	Jordan	Midlands	Dyno2	6	988	314
Jan-17	Jordan	North	Dyno3	9	981	372
Jan-17	Jordan	North	Dyno1	10	1,520	476
Jan-17	Jordan	North	Dyno2	6	966	330
Jan-17	Jordan	West	Dyno3	10	2,800	903
Jan-17	Jordan	West	Dyno1	8	1,536	572
Jan-17	Jordan	West	Dyno2	9	316	221
Jan-17	Jordan	Midlands	Dyno3	9	2,547	781
Jan-17	Jordan	Midlands	Dyno1	10	1,810	664
Jan-17	Jordan	Midlands	Dyno2	9	2,223	771
Jan-17	Clive	North	Dyno3	9	1,377	415
Jan-17	Clive	North	Dyno1	7	908	315
Jan-17	Clive	North	Dyno2	9	2,232	828
Jan-17	Clive	West	Dyno3	10	2,070	903
Jan-17	Clive	West	Dyno1	10	2,170	832

Рис. 2.2. Поля сводной таблицы

После выбора местоположения сводной таблицы нажмите на кнопку **OK**.

- После этого вы можете выбрать, какие поля включать в сводную таблицу. Давайте используем фильтр по месяцам (Month), регион (Region) отправим в столбцы, а имя продавца (Salesman) – в строки. В качестве значений используем сумму по продажам (Net Sales). Внешне сводная таблица должна выглядеть так, как показано на рис. 2.3.

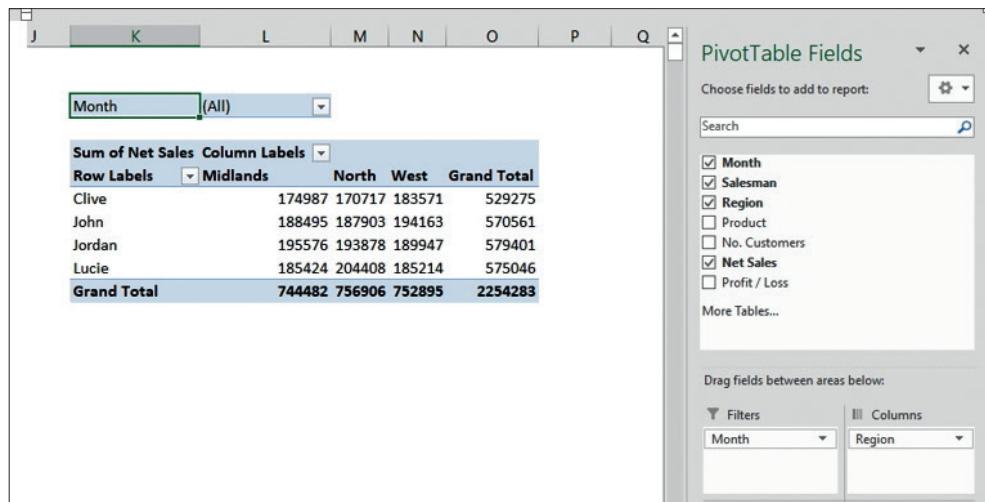


Рис. 2.3. Сводная таблица с заполненными данными

- Если нам необходимо знать, кто из продавцов является лидером продаж по временному диапазону или региону, можно добавить в таблицу итоги. Для этого необходимо щелкнуть правой кнопкой мыши по итоговому полю **Grand Total** (Общий итог) и в меню **Sort** (Сортировка) выбрать пункт **Sort Largest to Smallest** (Сортировка по убыванию), как показано на рис. 2.4.

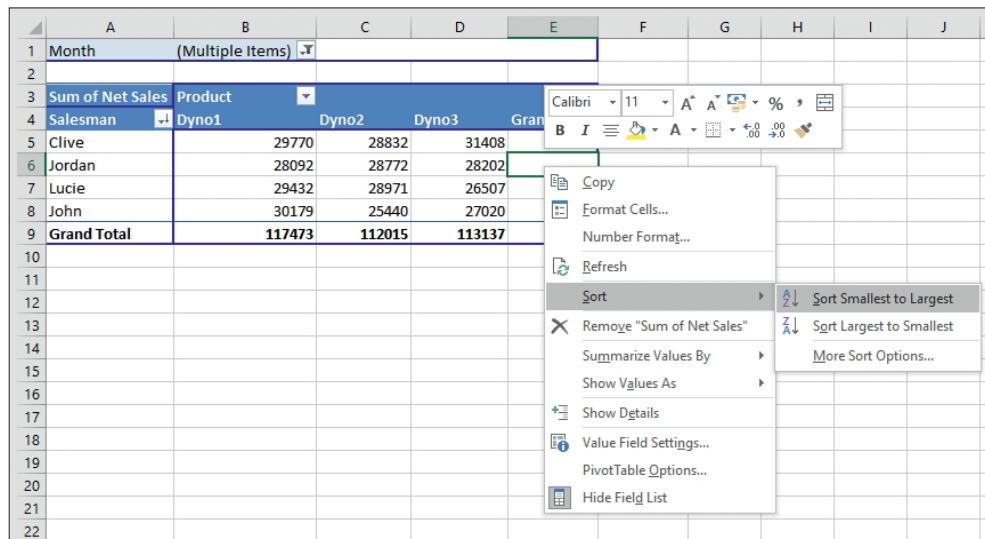


Рис. 2.4. Сортировка данных в сводной таблице

Всего за каких-то 30 секунд мы создали базовую сводную таблицу на основании 1400 строк исходных данных с возможностью быстрого редактирования параметров отчета. Одним из главных преимуществ сводных таблиц является

то, что вам не нужно изучать никаких формул, чтобы изменить критерии формирования отчета. К примеру, если вы хотите узнать, кто продал больше всех товаров в регионе **North**, вам не придется делать ничего особенного – сводная таблица автоматически перестроится, а порядок сортировки столбцов при этом сохранится, как видно на рис. 2.5.

	A	B	C
1	Month	(Multiple Items)	
2			
3	Sum of Net Sales	Region	
4	Salesman	North	Grand Total
5	Lucie	30108	30108
6	Clive	29616	29616
7	Jordan	29220	29220
8	John	27744	27744
9	Grand Total	116688	116688

Рис. 2.5. Автоматическая сортировка данных в сводной таблице при смене фильтра

И хотя сводные таблицы сами по себе довольно информативны, лично я предпочитаю снабжать их *срезами* (slicer). По сути, это то же самое, что выбирать требуемые поля из списка, однако срезы, без сомнений, добавляют сводной таблице элегантности.

Но главной причиной, почему мы показываем вам процесс создания срезов в сводных таблицах, является то, что в Power Query этот аспект полностью автоматизирован, и будет неплохо, если к тому моменту, когда мы доберемся до соответствующей главы, вы будете знать, как срезы создаются вручную. К тому же срезы являются исключительно интерактивными элементами, которые позволяют молниеносно манипулировать данными.

Чтобы добавить срез, вам необходимо выполнить следующие действия.

- Для начала нужно выделить сводную таблицу, с которой вы хотите работать. После этого необходимо в появившейся вкладке **Analyze** (Анализ) выбрать пункт **Insert Slicer** (Вставить срез), как показано на рис. 2.6.

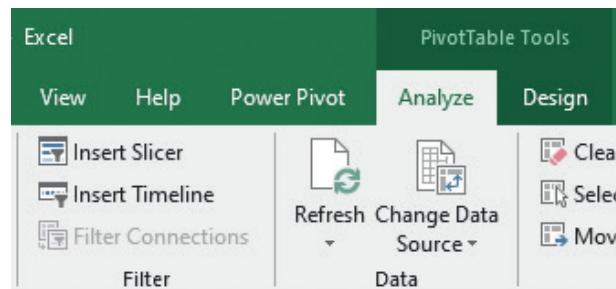


Рис. 2.6. Меню вставки среза в сводную таблицу

- Выберите в появившемся списке поля, по которым вы хотите вставить срезы, как показано на рис. 2.7, и нажмите кнопку **OK**.

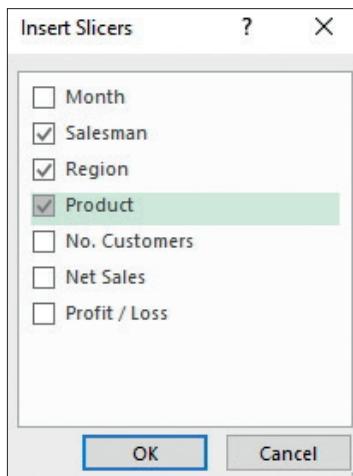


Рис. 2.7. Настройка срезов в сводной таблице

3. Захватив мышью заголовок окна среза, вы можете переместить его в наиболее подходящее место на листе, что видно по рис. 2.8.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Month	(Multiple Items)											
2													
3	Sum of Net Sales	Product											
4	Salesman	Dyno1	Dyno2	Dyno3	Grand Total								
5	Clive	29770	28832	31408	90010								
6	Jordan	28092	28772	28202	85066								
7	Lucie	29432	28971	26507	84910								
8	John	30179	25440	27020	82639								
9	Grand Total	117473	112015	113137	342625								
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													

Рис. 2.8. Позиционирование срезов в сводной таблице

Срезы позволяют вам манипулировать данными более интерактивно по сравнению с обычными фильтрами и сортировкой.

Представьте, что после создания сводной таблицы вы осознали, что забыли добавить в нее категории товаров, указанные для каждого товара на другом листе. Давайте попробуем исправить нашу оплошность и добавить информацию о категориях товаров, показанную на рис. 2.9, в нашу сводную таблицу.

Product	Category
Dyno1	Construction toy
Dyno2	Creative toy
Dyno3	Education toy
Dyno4	Electronic toy
Dyno5	Action Figure

Рис. 2.9. Дополнительные данные по категориям товаров

Для этого нам придется добавить в нашу исходную таблицу новый столбец с формулой ВПР (VLOOKUP) для извлечения данных с другого листа. Формула будет выглядеть так:

```
=ВПР(E3;Categories!B$3:C$7;2;FALSE)
```

В результате мы получим таблицу, показанную на рис. 2.10.

	A	B	C	D	E	F	G	H	I
1	Month	Salesman	Region	Product	Category	No. Custom	Net Sales	Profit / Loss	
2	Jan-17	John	North	Dyno1	Construction toy	8	1,592	563	
3	Jan-17	John	North	Dyno2	Creative toy	8	1,088	397	
4	Jan-17	John	West	Dyno3	Education toy	8	1,680	753	
5	Jan-17	John	West	Dyno1	Construction toy	9	2,133	923	
6	Jan-17	John	West	Dyno2	Creative toy	10	1,610	579	
7	Jan-17	John	Midlands	Dyno3	Education toy	10	1,540	570	
8	Jan-17	John	Midlands	Dyno1	Construction toy	7	1,316	428	
9	Jan-17	John	Midlands	Dyno2	Creative toy	7	1,799	709	
10	Jan-17	Lucie	North	Dyno3	Education toy	8	1,624	621	
11	Jan-17	Lucie	North	Dyno1	Construction toy	6	726	236	
12	Jan-17	Lucie	North	Dyno2	Creative toy	9	2,277	966	
13	Jan-17	Lucie	North						

Рис. 2.10. Добавление данных в таблицу при помощи функции ВПР

Теперь, поскольку наша исходная таблица расширилась, необходимо откорректировать диапазон сводной таблицы, нажав на кнопку **Change Data Source** (Источник данных) на той же вкладке **Analyze** (Анализ) и выбрав новый диапазон, как показано на рис. 2.11.

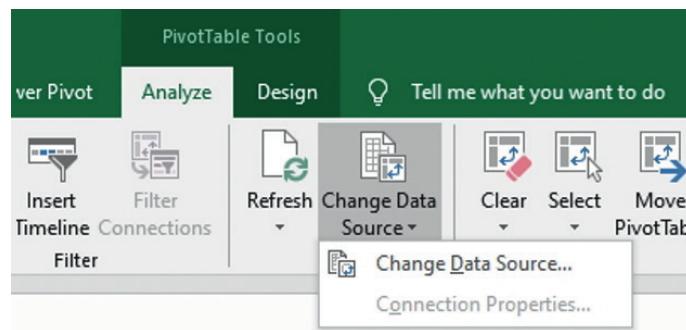


Рис. 2.11. Изменение диапазона сводной таблицы

После обновления сводной таблицы в ней будут показаны новые данные. И хотя это сработало, как мы и ожидали, что видно по рис. 2.12, при одновременном внесении множества изменений бывает легче создать сводную таблицу заново.

The screenshot shows a Microsoft Excel interface with a PivotTable on the left and several Power Pivot ribbon tabs on the right. The PivotTable has 'Month' at the top, followed by 'Row Labels' (with a dropdown menu showing 'Dyno1') and 'Column Labels' (with a dropdown menu showing 'All'). The data area contains five rows for Salesmen (Clive, John, Jordan, Lucie, Grand Total) and five columns for Product categories (Action Figure, Construction toy, Creative toy, Education toy, Electronic toy). The 'Grand Total' row shows the sum of net sales for each product category across all salesmen. The Power Pivot ribbon tabs visible are 'Product', 'Region', 'Salesman', and 'Region' again.

Month	(All)					
Row Labels	Column Labels					
	Dyno1	Dyno2	Dyno3	Dyno4	Dyno5	Grand Total
Clive	162953	161134	170195	17729	17264	529275
Action Figure				17264		17264
Construction toy	162953					162953
Creative toy		161134				161134
Education toy			170195			170195
Electronic toy				17729		17729
John	177588	172834	177797	24065	18277	570561
Action Figure				18277		18277
Construction toy	177588					177588
Creative toy		172834				172834
Education toy			177797			177797
Electronic toy				24065		24065
Jordan	181414	163857	183305	27896	22929	579401
Action Figure				22929		22929
Construction toy	181414					181414
Creative toy		163857				163857
Education toy			183305			183305
Electronic toy				27896		27896
Lucie	174044	173706	178300	31953	17043	575046
Action Figure				17043		17043
Construction toy	174044					174044
Creative toy		173706				173706
Education toy			178300			178300
Electronic toy				31953		31953
Grand Total	695999	671531	709597	101643	75513	2254283

Рис. 2.12. Сводная таблица по исходной таблице с функцией ВПР

Итоговая сводная таблица выглядит довольно неплохо. Но ее поддержка может занять немало времени, особенно если каждый месяц вам придется выполнять одни и те же повторяющиеся действия по обновлению данных.

Существуют и другие проблемы, характерные для сводных таблиц. И главной из них является то, что Excel поддерживает лишь чуть больше миллиона строк. Кто-то из вас подумает, что этого вполне достаточно, но при наличии больших объемов исходных данных нередко может потребоваться и гораздо больше записей для анализа.

ПОСТРОЕНИЕ МОДЕЛИ ДАННЫХ В POWER PIVOT

Давайте создадим простую модель данных в Power Pivot, чтобы понять, чем она отличается от обычной сводной таблицы. В процессе создания модели мы обратим внимание на некоторые преимущества и недостатки инструмента Power Pivot.

Примечание

Когда мы говорим о семействе продуктов Power, включающем в себя также Power View, Power BI Desktop, Power Map, Power BI и Power Query, надо помнить, что Power Pivot представляет движок DAX и по сути является мозгом, отвечающим за процессы создания отчетов, визуализации и анализа.

Процесс создания модели Power Pivot состоит из двух шагов: сначала надо создать таблицу в Excel, а затем добавить ее в модель.

Создание таблицы в Excel

Давайте откроем документ Excel, содержащий ту же информацию, которую мы использовали ранее. Он показан на рис. 2.13. Хотя изначально на листе содержится 1400 строк с данными, я собираюсь продлить заполнение до 1 048 576 строк, чтобы Excel уперся в свое ограничение.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Date	SalesRep	Product	Units	Discount		Product	Category	Price		SalesRep	Region
2	02/04/2014	Nancy	Game1	1	0.007		Dyno1	Construct	14.76		Clive	South
3	24/07/2011	Peter	Dyno1	12	0.012		Dyno2	Creative	29.76		John	West
4	02/03/2011	Harry	Board2	7	0.002		Dyno3	Education	45.2		Jordan	East
5	24/04/2010	Clive	Dyno4	4	0.007		Dyno4	Electronic	12.5		Lucie	North
6	23/05/2014	John	Game8	2	0.006		Dyno5	Action	11.87		Bevan	South
7	17/12/2015	Charlie	Dyno5	3	0.017		Board1	Education	8.55		Claire	South
8	27/09/2011	Peter	Board3	2	0.006		Board2	Electronic	14.34		Olivia	North
9	22/10/2014	Nancy	Game3	5	0.006		Board3	Education	11.23		James	West
10	29/03/2016	Mary	Game4	10	0.010		Board4	Action	9.67		Peter	West
11	27/11/2014	Thelma	Game7	2	0.020		Board5	Creative	8.34		Lars	East
12	23/12/2014	Jordan	Dyno5	9	0.016		Game1	Construct	4.56		Nancy	West
13	09/06/2010	Lucie	Game4	3	0.001		Game2	Creative	8.35		Alfie	North
14	07/09/2010	Bevan	Dyno3	3	0.009		Game3	Education	9.23		Thelma	East
15	06/10/2010	Susan	Game4	12	0.011		Game4	Education	12.51		Sean	North
16	20/01/2015	Lucie	Board5	9	0.002		Game5	Electronic	5.87		Mary	North
17	06/02/2011	Dennis	Dyno2	11	0.015		Game6	Construct	11.34		Harry	East
18	18/05/2015	Xavier	Game7	4	0.001		Game7	Action	5.88		Victoria	West
19	04/03/2011	Xavier	Dyno4	10	0.013		Game8	Creative	5.99		Robert	North
20	21/08/2013	Robert	Game7	5	0.002		Game9	Creative	4.65		Dennis	East
21	26/02/2011	Florence	Board3	13	0.020		Game10	Education	12.67		Susan	South
22	04/10/2012	Sean	Game8	8	0.019						Dale	North
23	31/08/2014	Susan	Dyno3	3	0.018						Helen	East
24	24/04/2011	Victoria	Dyno1	2	0.008						Florence	West
25	24/08/2011	Edgar	Game9	12	0.001						Charlie	North
26	29/03/2013	Clive	Game6	11	0.011						Michael	East
27	07/09/2010	Susan	Game2	2	0.002						Xavier	South
28	24/11/2013	Nancy	Game8	14	0.013						Edgar	North

Рис. 2.13. Данные для Power Pivot – начало листа

На рис. 2.14 показаны заключительные строки документа, включая строку с порядковым номером 1 048 576.

1048567	23/06/2010	Edgar	Dyno4	2	0.006
1048568	27/02/2015	Peter	Dyno2	12	0.001
1048569	04/01/2013	Robert	Game7	5	0.006
1048570	19/01/2011	James	Dyno4	14	0.018
1048571	05/10/2013	Dennis	Board4	12	0.009
1048572	27/10/2010	Harry	Game10	6	0.004
1048573	23/11/2014	Clive	Board2	10	0.015
1048574	30/03/2013	Robert	Game8	7	0.012
1048575	30/08/2015	Peter	Game10	8	0.013

Рис. 2.14. Данные для Power Pivot – окончание листа

Теперь нам необходимо выполнить следующие шаги.

1. Для начала преобразуем данные на листе в так называемую умную таблицу.

Примечание

Слова *данные* и *таблица* пришли из терминологии баз данных и обозначают хранилище информации. Каждая таблица содержит в себе данные об одной сущности (например, о товарах или продавцах), организованные в виде строк и столбцов. Обычно таблицы представляют информацию, достаточную для создания связей с другими таблицами. В терминах баз данных мы именуем хранилище информации *таблицей данных*. И хотя внешне она также состоит из строк и столбцов, в базе данных они называются *записями* и *полями*.

Существует два способа трансформировать данные в таблицу. Первый заключается в том, чтобы щелкнуть мышью внутри диапазона с данными, перейти на вкладку **Insert** (Вставка) и нажать на кнопку **Table** (Таблица), как показано на рис. 2.15. Второй способ проще и состоит в нажатии комбинации клавиш **Ctrl+T**.

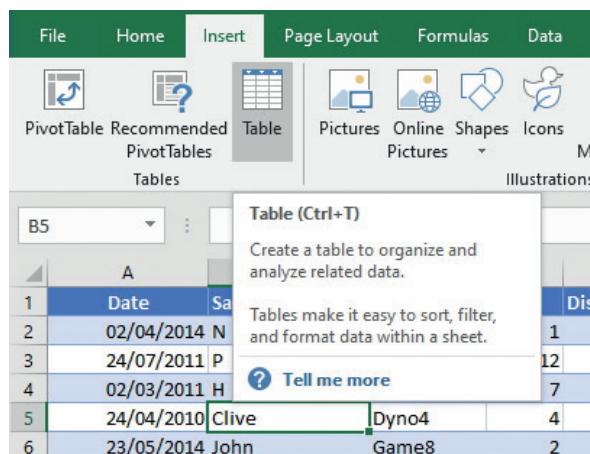


Рис. 2.15. Создание умной таблицы

- В появившемся диалоговом окне, представленном на рис. 2.16, вы можете нажать на кнопку **OK** или клавишу **Enter** на клавиатуре, чтобы подтвердить свои намерения.



Рис. 2.16. Диалоговое окно подтверждения создания таблицы

Примечание

Всегда лучше именовать свои умные таблицы, особенно когда у вас их несколько. Так вам будет легче обращаться к ним в будущем. Но при именовании таблиц необходимо соблюдать несколько простых правил. Имя таблицы должно начинаться с буквы или символа подчеркивания. Также вы не можете вставлять пробелы в названия таблиц и не можете называть их одинаково.

- Мы будем придерживаться правил именования таблиц в базах данных и поставим в начале имени таблицы букву «f», указывающую на то, что имеем дело с *таблицей фактов* (fact table). О различиях между таблицей фактов и измерением будет подробно рассказано в главе 4.

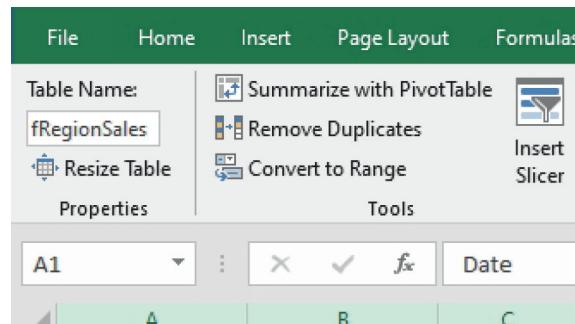


Рис. 2.17. Называем таблицу fRegionSales

- Выделите следующий диапазон данных, представляющий продавцов, и преобразуйте его в умную таблицу, используя один из двух перечисленных выше способов. Если ваша таблица содержит заголовки, не забывайте устанавливать флажок **My table has Headings** (Таблица с заголовками) при создании умной таблицы. Эта таблица представляет собой *измерение* (dimension), поэтому перед названием SalesRep я буду использовать префикс «d».

5. Повторите эти действия применительно к последней таблице, представляющей данные о товарах. В соответствии с выбранными принципами именования объектов назовем эту таблицу dProduct.

Итак, мы создали три таблицы на рабочем листе и теперь можем добавить их в модель данных.

Добавление таблиц в модель данных

Моделирование данных (data modeling) представляет собой процесс объединения таблиц в единую базу данных с определением правил хранения, обработки и обновления данных. И хотя наиболее популярным видом остается реляционная модель данных, существуют и другие, например иерархическая или сетевая модель. К этому шагу нельзя относиться поверхностно, поскольку именно здесь мы определяем структуру данных, которая должна отвечать аналитическим требованиям заказчика. Итак, приступим.

1. Перейдите на вкладку **Power Pivot** и нажмите на кнопку **Add to Data Model** (Добавить в модель данных), как показано на рис. 2.18.

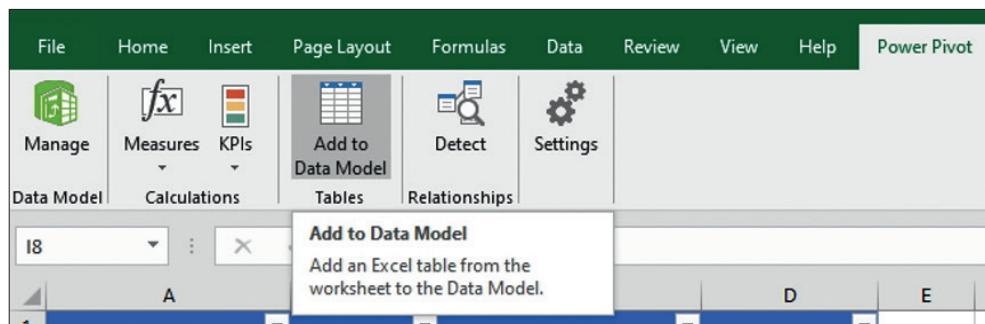


Рис. 2.18. Добавление таблицы в модель данных Power Pivot

Это может занять какое-то время, если объем данных в таблице большой. Важно понимать, что создаваемая модель данных жестко привязывается к нашей исходной таблице, так что любые изменения в таблице Excel отразятся на соответствующей модели данных.

2. После нажатия на кнопку **Add to Data Model** (Добавить в модель данных) откроется окно **Manage Data Model** (Управление моделью данных), в верхней части которого вы увидите три вкладки: **Home** (Главная), **Design** (Конструктор) и **Advanced** (Дополнительно), как показано на рис. 2.19.

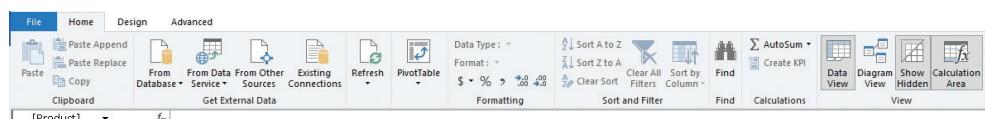


Рис. 2.19. Окно управления моделью данных

3. Вернитесь в Excel, либо нажав комбинацию клавиш **Alt+Tab**, либо щелкнув по иконке **Switch to Workbook** (Переключиться в книгу), расположившейся в верхнем левом углу окна. Добавьте две оставшиеся таблицы в модель данных так же точно, как добавляли первую. В результате появятся три вкладки в Power Pivot, как показано на рис. 2.20.

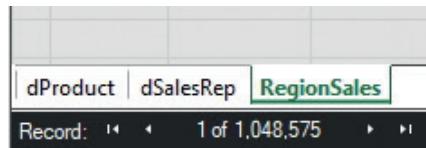


Рис. 2.20. Все три таблицы добавлены в модель данных

- Вы могли обратить внимание на то, что в таблице **fRegionSales** значения дат не поместились в столбец. Вы можете расширить столбец так же точно, как делаете это в Excel.
- После изменения размера столбца давайте поменяем формат отображения, чтобы дата и время не показывались в одной колонке. Для этого откроем выпадающий список с названием **Format** (Формат) на вкладке **Home** (Главная) и выберем четвертый сверху вариант, соответствующий формату **DD/MM/YYYY**, как показано на рис. 2.21. В результате в столбце останутся только даты без указания времени.

The screenshot shows the Microsoft Excel ribbon with the 'Home' tab selected. In the top right corner, there is a dropdown menu for 'Format' under the 'Data Type : Date' section. The dropdown lists various date formats, including '14/03/2001 13:30:55', '14 March 2001', and '14/03/2001'. The '14/03/2001' option is currently selected. Below the ribbon, a table is displayed with columns: Date, SalesRep, Product, Units, and Discount. The 'Date' column contains dates like '02/04/2014 00:00:00' and '24/07/2011 00:00:00'. The 'Format' dropdown is overlaid on the table, specifically covering the first few rows of the 'Date' column.

Рис. 2.21. Изменение формата отображения даты в столбце

6. Теперь давайте изменим формат отображения столбца **Price** в таблице с товарами (**dProduct**). Выберем для него тип **Currency** (Валюта), как показано на рис. 2.22.

The screenshot shows a Microsoft Excel ribbon with the 'Data' tab selected. Below the ribbon is a table named 'dProduct' with columns: Product, Category, Price, and Add Column. The 'Price' column contains numerical values like 14.76, 29.76, etc. A context menu is open over the 'Price' column header, specifically over the cell containing 14.76. The 'Format' dropdown is expanded, showing several options: General, Decimal Number, Whole Number, Currency, Percentage, and Scientific. The 'Currency' option is highlighted with a gray background. The rest of the table and ribbon are visible in the background.

	Product	Category	Price	Add Column
1	Dyno1	Construction	14.76	
2	Dyno2	Creative	29.76	
3	Dyno3	Education	45.2	
4	Dyno4	Electronic	12.5	
5	Dyno5	Action	11.87	
6	Board1	Education	8.55	
7	Board2	Electronic	14.34	
8	Board3	Education	11.23	
9	Board4	Action	9.67	
10	Board5	Creative	8.34	
11	Game1	Construction	4.56	
12	Game2	Creative	8.35	
13	Game3	Education	9.23	
14	Game4	Education	12.51	
15	Game5	Electronic	5.87	
16	Game6	Construction	11.34	
17	Game7	Action	5.88	
18	Game8	Creative	5.99	
19	Game9	Creative	4.65	
20	Game10	Education	12.67	

Рис. 2.22. Изменение формата столбца Price на Валюта

7. Также мы можем отсортировать столбец с датами путем нажатия на кнопку **Sort Oldest to Newest** (Сортировать от старых к новым), как видно на рис. 2.23. Остальные колонки также допустимо сортировать.

The screenshot shows the Power Pivot ribbon with the 'Sort and Filter' tab selected. Below the ribbon is a data grid with 12 rows of data. The columns are labeled: Date, Sale..., Pro..., Units, Discount, and Add Column. The 'Discount' column contains values like 0.007, 0.012, 0.002, etc. A tooltip for the 'Sort Oldest to Newest' button is visible, explaining: 'Sort the selected column so that the lowest values are at the top of the column.'

Date	Sale...	Pro...	Units	Discount	Add Column
1 02/04/2014	Nancy	Game1	1	0.007	
2 24/07/2011	Peter	Dyno1	12	0.012	
3 02/03/2011	Harry	Board2	7	0.002	
4 24/04/2010	Clive	Dyno4	4	0.007	
5 23/05/2014	John	Game8	2	0.006	
6 17/12/2015	Charlie	Dyno5	3	0.017	
7 27/09/2011	Peter	Boards3	2	0.006	
8 22/10/2014	Nancy	Game3	5	0.006	
9 29/03/2016	Mary	Game4	10	0.010	
10 27/11/2014	Thelma	Game7	2	0.020	
11 23/12/2014	Jordan	Dyno5	9	0.016	
12 09/06/2010	Lucie	Game4	3	0.001	

Рис. 2.23. Сортировка столбца

Кроме того, можно было бы поменять формат отображения поля **Discount** в таблице **fRegionSales** на проценты, но мы этого делать не будем.

Создание связей между таблицами

Теперь нам необходимо объединить таблицы, добавленные в модель данных, при помощи связей. Причин для объединения таблиц в модели масса, и важнейшие из них состоят в том, чтобы установить физические связи между таблицами, которые логически связаны, и свести к минимуму количество избыточных данных. Приступим.

- На вкладке **Home** (Главная) нажмите на кнопку **Diagram View** (Представление диаграммы), как показано на рис. 2.24.

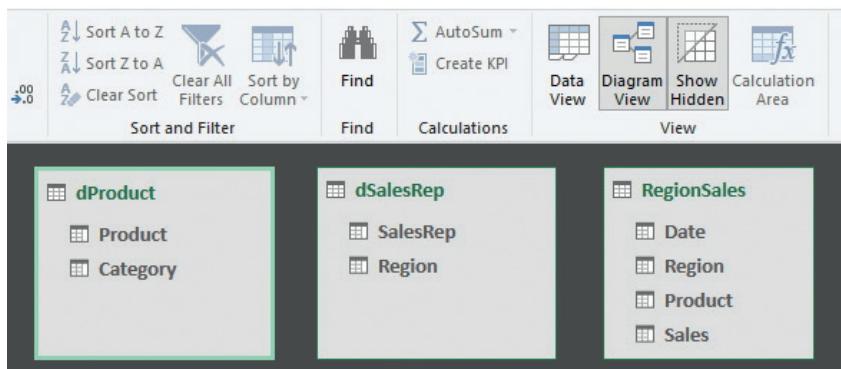


Рис. 2.24. Представление диаграммы в модели данных

- Чтобы создать связь, вам необходимо перетащить мышью поле **SalesRep** в таблице **dSalesRep** на соответствующее поле **SalesRep** в таблице **fRegionSales**. В результате поля будут объединены, и между ними появится связывающая линия.

В Power Pivot вы можете создавать связи различных типов. Та, что мы создали только что, называется *связью «один ко многим»* (one-to-many relationship). В таблице **dSalesRep** поле **SalesRep** является уникальным, тогда как в таблице фактов **fRegionSales** представлено множество дублирующихся продавцов.

3. По тому же принципу создайте связь между таблицами **fRegionSales** и **dProduct**, используя для объединения поле **Product**. Итоговый результат показан на рис. 2.25.

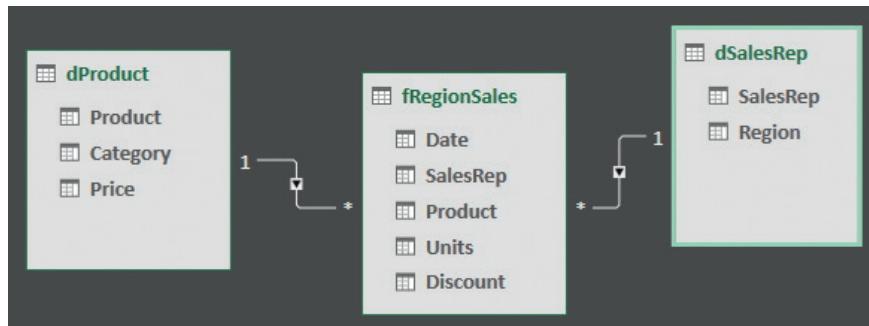


Рис. 2.25. Созданные связи между таблицами

В представлении диаграммы бывает непросто понять, по каким именно полям объединены таблицы. По рис. 2.25 можно подумать, что поле **Category** из таблицы **dProduct** связано с полем **Product** из таблицы **fRegionSales**. Но если щелкнуть мышью по связи, все встанет на свои места.

Если вы хотите отредактировать созданную связь, то можете щелкнуть по ней правой кнопкой мыши и выбрать пункт **Edit Relationship** (Изменить связь), как показано на рис. 2.26. В результате откроется диалоговое окно с таблицами и столбцами с указанием того, по какому полю объединены таблицы.

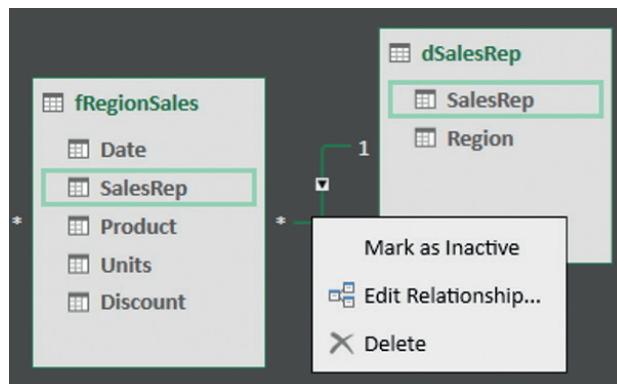


Рис. 2.26. Редактирование связи

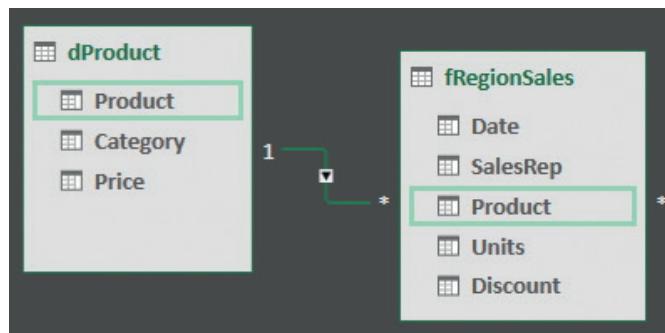


Рис. 2.27. Связи между таблицами

Перед тем как возвращаться на исходный экран **Data View** (Представление данных), необходимо сохранить связи, которые мы создали, – иначе таблицы не будут связаны. Есть несколько способов сохранить созданные связи: вы можете нажать на иконку сохранения в верхней левой части экрана, выбрать пункт **Save** (Сохранить) на вкладке **File** (Файл) или воспользоваться привычным сочетанием клавиш **Ctrl+S**.

POWER QUERY СПЕШИТ НА ПОМОЩЬ

Теперь, когда мы создали связи между таблицами в модели данных, объединим их вместе. Нам необходимо создать поле **NetIncome** в нашей таблице **fRegionSales**. Для поддержания целостности нашей модели мы не можем использовать Power Pivot для редактирования существующих данных. Но мы можем создать новый **вычисляемый столбец** (calculated column) на базе существующих колонок в таблице.

Создание вычисляемого столбца

Щелкните по вкладке **fRegionSales**, чтобы показать соответствующую таблицу. Мы собираемся создать новый вычисляемый столбец. Для этого необходимо дважды щелкнуть мышью на надпись **Add Column** (Добавление столбца) справа от последней колонки в таблице и ввести имя столбца **Net Income**, как показано на рис. 2.28.

	Date	SalesRep	Product	Units	Discount	Net Income	Add Column
1	15/06/2016	John	Game7	10	0.003		
2	15/06/2016	Sean	Game8	3	0.014		
3	15/06/2016	Michael	Game5	10	0.020		
4	15/06/2016	Michael	Game2	12	0.016		
5	15/06/2016	John	Game8	9	0.012		
6	15/06/2016	Mary	Board1	2	0.007		

Рис. 2.28. Создание вычисляемого столбца

Нам бы хотелось в этом столбце перемножить значения из полей **Units** и **Discount** из таблицы **fRegionSales** и **Price** из таблицы **dProduct**. В Excel мы, конечно, воспользовались бы упомянутой ранее функцией ВПР (VLOOKUP). Но такой функции в Power Pivot нет. К счастью, есть другая функция, которая называется *Related*. Начнем.

- Щелкните на первой ячейке новообразованного столбца **Net Income**, наберите на клавиатуре знак равенства (=), как если бы собирались вводить формулу в Excel, и введите название функции **Related**. После открытия скобки Power Pivot автоматически предложит вам на выбор все поля из связанных таблиц, как показано на рис. 2.29, – именно за этим мы и объединили связями наши таблицы.

Date	SalesRep	RELATED(Column Name)
15/06/2016	John	dProduct[Category]
15/06/2016	Sean	dProduct[Price]
15/06/2016	Michael	dProduct[Product]
15/06/2016	Michael	dSalesRep[Region]
15/06/2016	John	dSalesRep[SalesRep]

Рис. 2.29. Создание формулы вычисляемого столбца

- У вас есть два способа завершить формулу. Проще всего будет щелкнуть на поле **dProduct[Price]** из предложенного выпадающего списка. Второй вариант – щелкнуть на вкладке **dProduct** и выбрать столбец **Price** вручную. Я предпочитаю последний способ, проиллюстрированный на рис. 2.30, чтобы убедиться, что выбран правильный столбец.

Product	Category	Price
Dyno1	Construction	£14.76
Dyno2	Creative	£29.76
Dyno3	Education	£45.20
Dyno4	Electronic	£12.50
Dyno5	Action	£11.87
Board1	Education	£8.55
Board2	Electronic	£14.34

Рис. 2.30. Создание формулы при помощи ручного выбора

- Теперь закройте скобку и нажмите на клавишу **Enter**. Заметьте, что в столбце отобразились цены из таблицы товаров, что видно по рис. 2.31.

	[Net Income]	fx	=related(dProduct[Price])				
	Date	SalesRep	Product	Units	Discount	Net Income	Ad
1	15/06/2016	John	Game7	10	0.003	5.88	
2	15/06/2016	Sean	Game8	3	0.014	5.99	
3	15/06/2016	Michael	Game5	10	0.020	5.87	
4	15/06/2016	Michael	Game2	12	0.016	8.35	
5	15/06/2016	John	Game8	9	0.012	5.99	

Рис. 2.31. Столбец Net Income заполнился ценами из справочника товаров

Конечно, вы могли и вручную ввести формулу `=related(dProduct[Price])`.

Примечание

Перед тем как продолжить, необходимо сделать ряд замечаний. В Power Pivot принято обращаться к столбцам исключительно с указанием названия таблицы и имени поля в квадратных скобках.

Если в Excel таблицы обладают структурированными ссылками вроде F6, Power Pivot больше похож на **Access**, в котором также используется обращение через названия таблиц и полей.

Позже мы более подробно обсудим эту особенность, но вам уже необходимо понимать, что значит *контекст строки* (row context). Создавая связи между таблицами, мы даем Power Pivot понять, что в таблице товаров существуют строки, объединенные с нашей таблицей фактов. Например, в первой строке таблицы у нас указан товар **Game7**. Power Pivot просматривает связанную таблицу **dProduct**, обнаруживая при этом цену, соответствующую товару **Game7**, и вставляя ее в нужную ячейку. Далее Power Pivot переходит к следующей строке – и так до конца. Мы будем использовать контекст строки при работе с вычисляемыми столбцами.

Кроме того, стоит отметить, что для каждой ячейки в вычисляемом столбце **Net Income** указана одна и та же формула. В Excel вы бы использовали относительные ссылки на ячейки или что-то еще в своих формулах, а присутствие контекста строки позволяет во всех строках содержать одинаковые формулы.

Теперь, когда мы получили цену товара из таблицы **dProduct**, мы можем завершить написание формулы для получения чистой прибыли.

- Щелкните по строке формул, введите в конце формулы *(1-, нажмите на поле **Discount** и закройте скобку. Осталось перемножить полученный результат на значение поля **Units**. Для этого наберите далее *(, выберите поле **Units** и закройте скобку.

Полученная формула должна выглядеть так:

```
=related(dProduct[Price])*(1-fRegionSales[Discount])*(fRegionSales[Units])
```

Результат вычисления показан на рис. 2.32.

	Date	SalesRep	Product	Units	Discount	Net Income	Add Colu
1	15/06/2016	John	Game7	10	0.003	58.615706463...	
2	15/06/2016	Sean	Game8	3	0.014	17.723319193...	
3	15/06/2016	Michael	Game5	10	0.020	57.530667481...	
4	15/06/2016	Michael	Game2	12	0.016	98.590910377...	
5	15/06/2016	John	Game8	9	0.012	53.277641840...	
6	15/06/2016	Mary	Board1	2	0.007	16.983677940...	
7	15/06/2016	Charlie	Board2	3	0.010	42.575205935...	
8	15/06/2016	Xavier	Dyno5	13	0.015	152.00610909...	
9	15/06/2016	Helen	Dyno2	1	0.017	29.268880248...	
10	15/06/2016	Sean	Board1	11	0.001	93.991070872...	
11	15/06/2016	Peter	Dyno3	1	0.007	44.899078972...	

Рис. 2.32. Окончательная формула получения чистой прибыли

- Глядя на полученные результаты, первое, что хочется сделать, – это конвертировать вывод в формат валюты. Для этого щелкните на новом поле, выберите для него тип **Currency** (Валюта) и укажите ниже конкретную денежную единицу.

Теперь, когда мы научились создавать вычисляемый столбец, пришло время узнать, как создать вычисляемое поле, которое мы сможем использовать для подсчета итоговых продаж по регионам.

Примечание

Вычисляемые поля часто называют *мерами* (measure) или *явными формулами* (explicit formula).

The screenshot shows the Power Query Editor interface with a table of data. The table has columns: Product, Category, Price, and Add Column. The rows contain various products like Dyno1 through Game10, categorized into Construction, Creative, Education, and Electronic. The 'Add Column' column is empty.

Annotations on the right side of the table explain the differences:

- Вычисляемый столбец** (Calculated Column): Points to the 'Add Column' header.
- Вычисляемое поле** (Calculated Field): Points to the empty 'Add Column' cells.
- Область мер** (Measure Area): Points to the bottom-most row of the table.
- Сетка мер** (Measure Grid): Points to the bottom-most row of the table.

The table data is as follows:

	Product	Category	Price	Add Column
1	Dyno1	Construction	£14.76	
2	Dyno2	Creative	£29.76	
3	Dyno3	Education	£45.20	
4	Dyno4	Electronic	£12.50	
5	Dyno5	Action	£11.87	
6	Board1	Education	£8.55	
7	Board2	Electronic	£14.34	
8	Board3	Education	£11.23	
9	Board4	Action	£9.67	
10	Board5	Creative	£8.34	
11	Game1	Construction	£4.56	
12	Game2	Creative	£8.35	
13	Game3	Education	£9.23	
14	Game4	Education	£12.51	
15	Game5	Electronic	£5.87	
16	Game6	Construction	£11.34	
17	Game7	Action	£5.88	
18	Game8	Creative	£5.99	
19	Game9	Creative	£4.65	
20	Game10	Education	£12.67	

Рис. 2.33. Различия между вычисляемыми столбцами и вычисляемыми полями

Теперь научимся создавать вычисляемые поля.

Создание вычисляемого поля

Создать *вычисляемое поле* (calculated field) можно двумя способами. Первый состоит в том, чтобы щелкнуть по *сетке мер* (measured grid) под вашей таблицей и действовать через нее. Допустим, мы хотим рассчитать итоговую чистую прибыль. Хотя это можно сделать и в сводной таблице, лучше создать *явную формулу* (explicit measure) в Power Pivot. Причина этого в том, что созданную формулу вы сможете применять повторно и использовать ее в формулах DAX. Посмотрим.

1. Для начала создадим имя вычисляемого поля. Я напишу текст **Total Net Income**. Несмотря на то что вы предварительно выделили ячейку в *области мер* (measured area), текст на самом деле вводится в строке формул. Добавим к введенному тексту следующий: `=sum(fRegionSales[Net Income])`.

Примечание

Если вы использовали в своей работе **Access**, то знаете, что сначала нужно ввести имя вычисляемого поля, затем поставить двоеточие и написать формулу.

2. После нажатия на клавишу **Enter** и расширения столбца вы заметите, что поле **Total Net Income** отформатировано неправильно, что видно по рис. 2.34.

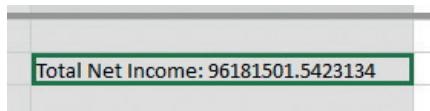


Рис. 2.34. Неправильно отформатированное вычисляемое поле

3. Чтобы это исправить, нажмите символ валюты на вкладке **Home** (Главная). Второй способ создания вычисляемого поля состоит в том, чтобы вернуться в Excel и нажать на выпадающую кнопку **Measures** (Меры) на вкладке **Power Pivot**, как показано на рис. 2.35.

Вы можете выбрать пункт добавления меры или перейти в диалоговое окно редактирования мер. На рис. 2.35 вы видите, что Power Pivot сохранил наше вычисляемое поле **Total Net Income** как валютное. Думаю, первый способ легче, поскольку вы можете щелкнуть и увидеть всю необходимую информацию, вместо того чтобы писать формулу вручную.

На данный момент мы рассмотрели создание связей, вычисляемых столбцов и полей, а также написали пару формул. В следующем разделе мы увидим, как все это можно объединить при создании сводной таблицы из Power Pivot. Попутно мы также отметим некоторые недостатки этого инструмента.

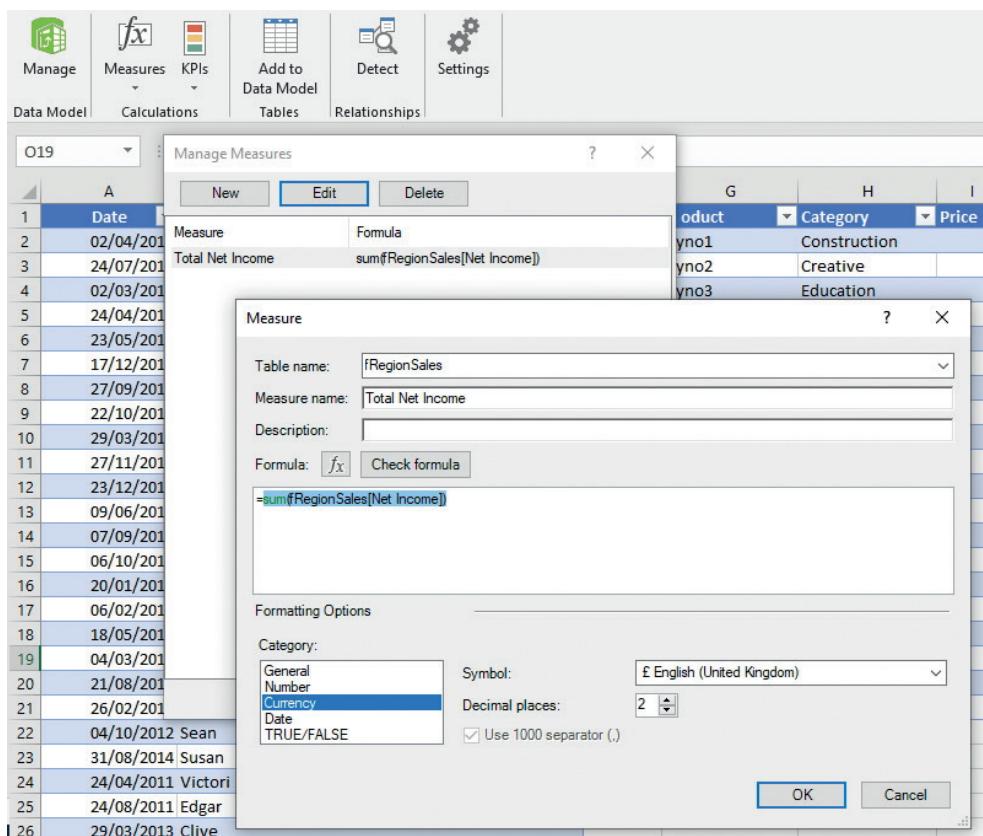


Рис. 2.35. Второй способ создания вычисляемого поля

Создание сводной таблицы из Power Pivot

Все знания, которые мы получили ранее на протяжении этой главы, пригодятся нам сейчас для создания сводной таблицы непосредственно из Power Pivot. Напомним, что мы взяли исходный лист в Excel объемом 1 048 576 строк, добавили таблицы в модель данных, изменили некоторые форматы и объединили таблицы при помощи связей. Затем создали вычисляемые столбцы с расчетом цены товаров и чистой прибыли, а также вычисляемое поле для подсчета продаж по регионам. Также мы вскользь познакомились с языком DAX, когда писали простую формулу `=related(dProduct[Price])`.

Теперь мы готовы собрать эту информацию воедино, чтобы создать сводную таблицу непосредственно из Power Pivot. Если вы уже закрыли рабочую книгу, созданную в предыдущих разделах, вам придется вновь открыть ее – мы продолжим работать с ней.

Чтобы создать сводную таблицу из Power Pivot, нажмите на кнопку **PivotTable** (Сводная таблица) на вкладке **Home** (Главная) и установите переключатель в положение **New Worksheet** (Новый лист). Первое, что вы увидите, – это то, что загрузились все ранее связанные таблицы, как показано на рис. 2.36, так что вы можете использовать информацию из каждой из них.

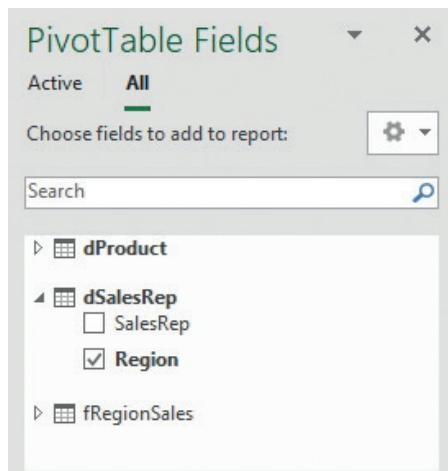


Рис. 2.36. Поля в сводной таблице из связанных таблиц

В сводной таблице, представленной на рис. 2.37, мы выбрали несколько продавцов из региона **South**, определенные товары и вывели созданное нами ранее вычисляемое поле **Total Net Income**.

Total Net Income	Column Labels										Grand Total
	=Dyno1		=Dyno2		=Dyno3		=Dyno4		=Dyno5		
Row Labels	Bevan	Clive	Bevan	Clive	Bevan	Clive	Bevan	Clive	Bevan	Clive	Grand Total
■ South											
Action									£165,200.76	£174,505.01	£339,705.76
Construction	£220,097.49	£211,266.93									£431,364.43
Creative			£431,993.42	£440,613.85							£872,607.26
Education					£655,867.04	£627,322.12					£1,283,189.16
Electronic							£180,695.12	£179,362.33			£360,057.45
Grand Total	£220,097.49	£211,266.93	£431,993.42	£440,613.85	£655,867.04	£627,322.12	£180,695.12	£179,362.33	£165,200.76	£174,505.01	£3,286,924.06

Рис. 2.37. Готовая сводная таблица

Примечание

При расчете мер используется контекст фильтра. В нашем примере поле **Sum of Total Net Income** отфильтровано по выбранному региону.

После всех усилий, которые мы приложили на ранних стадиях, создание сводной таблицы из Power Pivot не составило труда.

Недостатки Power Pivot

Сводные таблицы, созданные из Power Pivot, обладают рядом недостатков, о которых вам просто необходимо знать, чтобы обходить их при работе с такими отчетами.

Проблема 1. Множественный выбор элементов

Первая проблема связана с невозможностью сгруппировать даты по кварталам или годам. Конечно, вы можете воспользоваться флажком **Select Multiple**

Items (Выделить несколько элементов), как показано на рис. 2.38, и выбрать нужные даты вручную, но это весьма и весьма трудозатратно. Кроме того, это может вызвать массу ошибок, связанных с человеческим фактором, ведь в этом случае очень легко ошибиться и выбрать не все нужные даты.

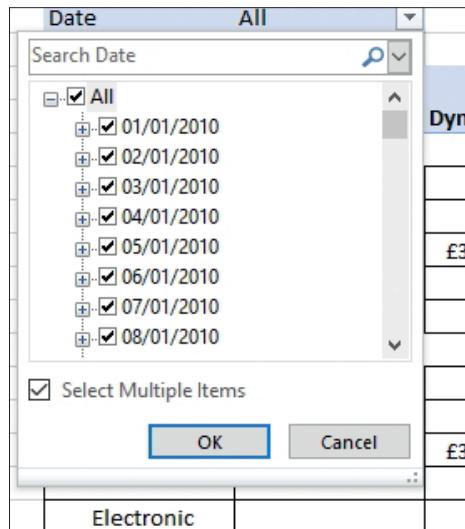


Рис. 2.38. Проблема с выбором дат в Power Pivot

Проблема 2. Предпросмотр в Power Pivot

Вторая проблема связана с тем, что если в обычной сводной таблице вы всегда можете дважды щелкнуть по ячейке и увидеть предпросмотр по всем строкам, составляющим содержимое ячейки, то в сводной таблице, созданной в Power Pivot, предпросмотр ограничен 1000 записей.

	A	B	C	D
1	Data returned for Total Net Income, East - Construction, Dyno1 (First 1000 rows).			
2				
3	fRegionSales[Date] ▾ fRegionSales[SalesRep] ▾ fRegionSales[Product] ▾ fRegionSales[Unit]			
4	27/02/2015 Dennis	Dyno1		
5	01/04/2014 Michael	Dyno1		
6	29/11/2014 Thelma	Dyno1		
7	25/06/2014 Jordan	Dyno1		
8	29/08/2015 Thelma	Dyno1		

Рис. 2.39. Проблема первых 1000 строк в Power Pivot

Проблема 3. Вычисляемые поля

Третья проблема, которую мы наблюдали в нашем предыдущем примере, связана с невозможностью создавать вычисляемые поля и элементы, хотя в обычных сводных таблицах такая возможность присутствует.

Проблема 4. Версии Microsoft Office

Наконец, сводные таблицы, созданные посредством Power Pivot, могут обновляться и конфигурироваться только в версиях Excel 2013 и выше. Таким образом, если таблица, которую вы собираетесь использовать для свода данных, содержит меньше миллиона строк, а вам необходимо воспользоваться какими-то из перечисленных здесь возможностей, лучше будет остановить свой выбор на обычной сводной таблице без использования модели данных.

ЗАКЛЮЧЕНИЕ

Несмотря на то что в Excel можно довольно легко и просто создавать полноценные сводные таблицы, все же они будут обладать определенными недостатками, с которыми может побороться Power Query. Одной из главных проблем Excel является его пресловутое ограничение на один миллион строк на листе. В реальности же производительность Excel начинает снижаться уже после нескольких сотен тысяч строк.

Power Pivot позволяет нам работать с наборами данных гигантских размеров, при этом рабочие книги остаются небольшого размера и, в отличие от обычных сводных таблиц, работают очень быстро. Для достижения подобного быстродействия Power Pivot загружает исходные данные не на рабочий лист, а непосредственно во внутреннюю модель данных. После загрузки таблиц мы можем настроить между ними необходимые связи, что позволит не беспокоиться об использовании функции ВПР (VLOOKUP) для создания единой таблицы на основании многих.

Создавая сводные таблицы из Power Pivot, мы получаем возможность анализировать данные сразу из нескольких исходных таблиц гораздо быстрее и эффективнее.

В следующей главе мы познакомимся с интерфейсом инструмента Power Query, создадим свой первый запрос и узнаем, как отправлять данные обратно в Excel из Power Query.

Глава 3

Введение в интерфейс Power Query

В данной главе мы познакомимся с интерфейсом Power Query и пробежимся по его вкладкам, ленте и пунктам меню. В процессе изучения этого инструмента вы усвоите несколько важных советов и приемов для выполнения преобразования данных. Вы также научитесь создавать простые запросы и узнаете, как использовать вкладку **View** (Просмотр) для тонкой настройки запросов. Кроме того, мы остановимся на различных методах отправки преобразованных данных обратно в рабочую книгу Excel.

Основные темы этой главы:

- интерфейс Power Query и его базовые элементы;
- создание простого запроса в Power Query;
- исследование опции **Load To...** (Загрузить в...);
- советы по профилированию данных.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Изучение этой главы потребует от вас хорошего понимания задач, стоящих перед Power Query, а также знания окружения Excel 2019 и умения работать с файловой системой (папки, файлы и их типы). Кроме того, вы должны хорошо разбираться в ленте Excel 2019 и уметь свободно перемещаться по меню при выполнении различных операций. Также будет плюсом, если у вас есть определенные знания в области теории баз данных, поскольку на протяжении главы мы будем использовать схожую терминологию.

Примеры для этой главы можно загрузить по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

Сопутствующие видеофрагменты располагаются по ссылке: https://www.youtube.com/watch?v=yXC8-jqqqXY&list=PLelcvrwLe1860_GJEzs47WaZXwZjwTN83&index=4&t=1s.

ИНТЕРФЕЙС POWER QUERY И ЕГО ОСНОВНЫЕ ЭЛЕМЕНТЫ

В этом разделе мы познакомимся с главным окном Power Query. Вы узнаете, как использовать те или иные элементы, и научитесь свободно работать с основными пунктами меню.

Как мы уже говорили в предыдущей главе, Power Query запускается в Excel при помощи группы **Get & Transform** (Получить и преобразовать данные) на вкладке **Data** (Данные). Щелкнув на кнопке получения данных, выберите пункт **Launch Power Query Editor** (Запустить редактор запросов) в выпадающем списке. Редактор Power Query откроется в отдельном окне поверх Excel, как показано на рис. 3.1.

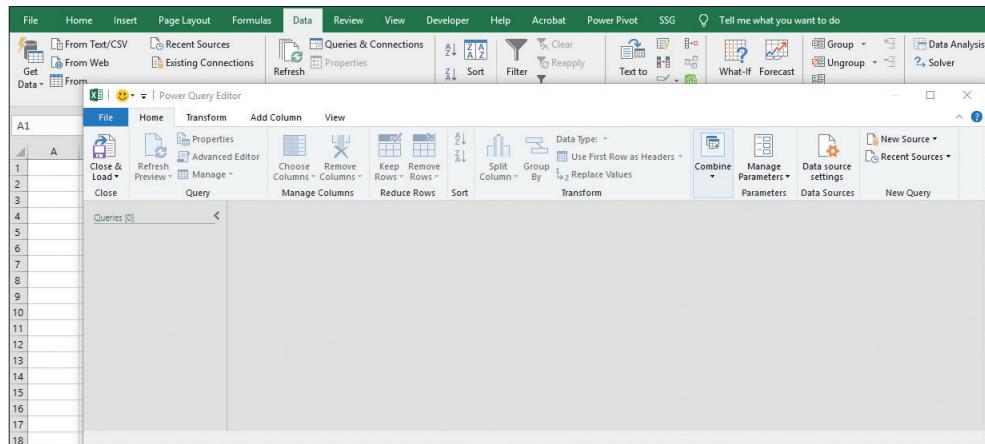


Рис. 3.1. Окно Power Query без загруженных данных

Примечание

Различные методы доступа к Power Query были описаны в предыдущих главах.

В следующих разделах мы опишем элементы интерфейса Power Query:

- основная лента и вкладки;
- навигационная панель (справка **Queries** (Запросы));
- предпросмотр таблицы данных;
- панель **Query Settings** (Параметры запроса).
- Вместе с тем мы обсудим следующие важные темы:
- работа с панелью **APPLIED STEPS** (Примененные шаги);
- исследование вкладки **View** (Просмотр);
- использование расширенного редактора (кнопка **Advanced Editor**).

После этого мы будем готовы к созданию первого запроса в Power Query. Приступим!

Главная лента и вкладки

Главная лента Power Query обеспечивает доступ ко всем возможностям этого инструмента, связанным с очисткой, преобразованием и трансформацией данных из различных источников.

Каждый раз, когда вы используете для действий иконки на главной ленте, выполняемые шаги сохраняются в Power Query и становятся доступными для дальнейшего использования, редактирования или удаления. Это особенно полезно при подготовке данных в Power Query для дальнейшего анализа в других инструментах, например в сводных таблицах или посредством дашбордов.

Что касается главного меню Power Query, то оно насчитывает всего пять вкладок: **File** (Файл), **Home** (Главная), **Transform** (Преобразование), **Add Column** (Добавить столбец) и **View** (Просмотр), как показано на рис. 3.2.



Рис. 3.2. Вкладки Power Query

Ознакомьтесь с содержимым этих вкладок, чтобы понимать, где какие элементы располагаются.

Навигационная панель или список запросов

На боковой панели **Queries** (Запросы), показанной на рис. 3.3, отображаются все созданные вами запросы. Все новые источники данных или запросы, которые добавляются для создания в Power Query, будут показываться на этой панели.

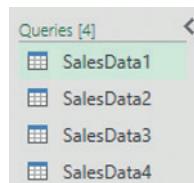


Рис. 3.3. Панель Queries

Давайте посмотрим, как можно добавить/создать новый запрос в Power Query.

Добавление запроса с использованием навигационной панели

Для создания запроса из навигационной панели выполните следующие действия.

1. Щелкните правой кнопкой мыши на панели **Queries** (Запросы).
2. Раскройте выпадающий список пункта меню **New Query** (Новый запрос).
3. Выберите из предложенного списка источник данных и, если потребуется, уточните свой выбор для создания запроса, как показано на рис. 3.4.

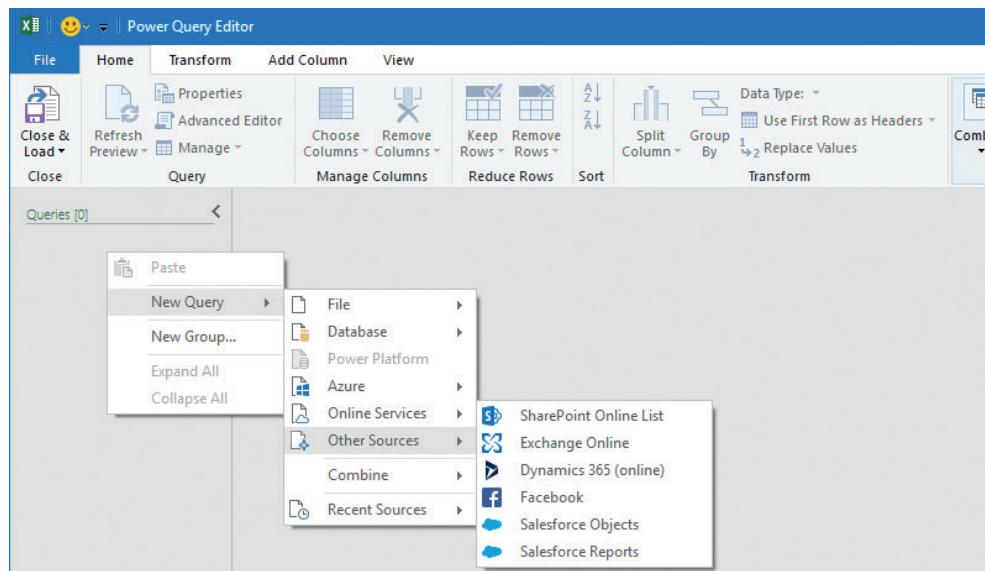


Рис. 3.4. Добавление запроса из навигационной панели

В дальнейшем вам может понадобиться изменить существующий запрос. Об этом мы поговорим в следующем разделе.

Редактирование запросов

Вы можете отредактировать запрос путем нажатия на нем правой кнопкой мыши в навигационной панели. Появится вспомогательное меню, показанное на рис. 3.5, в котором вы можете переименовать, удалить, дублировать, сортировать запросы и даже создать функцию или перейти в расширенный редактор (**Advanced Editor**).

Product	Date	Sales
Board5	06/09/2012	53395.17732
Game6	23/02/2016	116609.6948
Game1	20/09/2013	72524.0953
Game5	24/12/2010	22538.47873
Dyno4	28/03/2012	45616.53282
Game5	23/03/2014	81902.36089
Board5	12/09/2013	72000.3749
Game10	18/03/2015	99227.6778
Dyno1	22/11/2014	93520.54806
Game5	24/12/2013	77445.82665
Dyno5	15/11/2011	38690.90909
Game2	13/03/2014	81105.31419
Dyno1	12/06/2013	66631.66186
Game1	15/03/2011	26633.681
Game10	25/05/2010	11441.31857

Рис. 3.5. Контекстное меню навигационной панели

Если вы работаете сразу с несколькими запросами, относящимися к одной теме, отделу или региону, то можете для удобства объединить их в группу. Посмотрим, как это можно сделать.

Группировка запросов

Вы можете довольно легко и просто объединять запросы в группы. Создать новую группу можно, либо кликнув на свободном пространстве в навигационной панели правой кнопкой мыши и выбрав пункт **New Group** (Создать группу) в контекстном меню, либо через вызов контекстного меню запроса следующим образом:

1. Выберите один или несколько запросов в навигационной панели (во втором случае это необходимо сделать с зажатой клавишей **Ctrl**), которые вы хотите переместить в группу.
2. Вызовите контекстное меню, щелкнув правой кнопкой мыши на любом из выбранных запросов, и выберите пункт меню **Move To Group** (Переместить в группу), как показано на рис. 3.6.

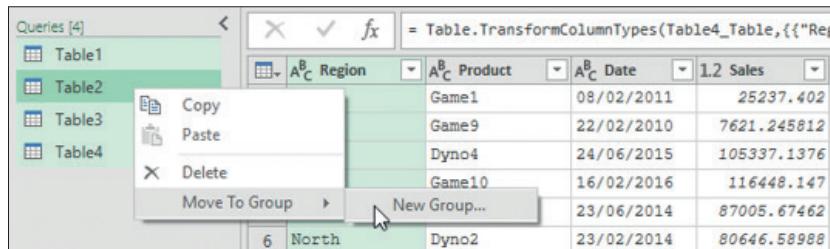


Рис. 3.6. Перемещение запросов в группу

3. Выберите пункт **New Group** (Создать группу).
4. Введите название группы **SalesData** и описание группы, как показано на рис. 3.7.

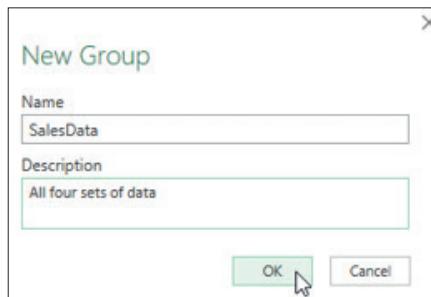


Рис. 3.7. Создание новой группы

Совет

Добавление описания группы мне кажется весьма полезным, поскольку при загрузке данных обратно в Excel вы будете видеть это описание на панели **Queries & Connections** (Запросы и подключения).

5. Нажмите на кнопке **OK** для сохранения группы.
6. Новая группа появится в навигационной панели, а все выбранные запросы будут перенесены в нее, что видно по рис. 3.8.

The screenshot shows the Power Query Editor interface. On the left, there's a navigation pane titled 'Queries [4]' with a tree view. Under 'SalesData [4]', four tables are listed: Table1, Table2, Table3, and Table4. The main workspace displays a table with columns: Region, Product, Date, and Sales. The data rows are: 1 South Board2 16/03/2015 100308.7559, 2 West Game10 02/05/2013 65824.37301, 3 North Game3 02/01/2014 78488.77647, 4 South Dyno2 02/10/2011 36987.84956, 5 North Game8 03/04/2011 28013.17286, 6 North Dyno4 16/09/2010 18192.7951, 7 North Board2 17/12/2015 113945.7647, 8 South Game4 03/04/2013 64663.23978.

Рис. 3.8. Результат создания группы

7. Группа может быть развернута или свернута при помощи стрелки слева от папки.

Предпросмотр таблицы данных

Это самое большое центральное окно в Power Query, в котором представлен предварительный просмотр выбранного в навигационной панели запроса, а также состояние данных после применения шагов, показанных справа. Именно к этому представлению вы обращаетесь, когда выполняете преобразование данных согласно вашим требованиям.

В верхней части окна вы видите заголовки столбцов, а слева – номера строк. В заголовке каждого столбца также показан тип данных слева от названия колонки, что видно по рис. 3.9.

В нашем примере столбцы называются так: Region, Product, Date и Sales. При наведении мышью на иконку с типом данных появится квадратное обрамление, как в случае с колонкой Region на рис. 3.9.

The screenshot shows the Power Query Editor in preview mode. The top bar shows the formula: = Table.TransformColumnTypes(Source, {{"Region", ...}}). The preview area shows the first row of data: 1 South Board2 16/03/2015 100308.7559. The 'Region' column has a tooltip indicating its type is 'Text'.

Рис. 3.9. Типы данных столбцов

Щелчок по иконке приведет к появлению показанного на рис. 3.10 контекстного меню с перечислением допустимых типов данных для столбца. Пункт **Using Locale** (Используя локаль) в нижней части списка позволит вам выбрать тип данных с использованием конкретной локали, например **English (United Kingdom)**.

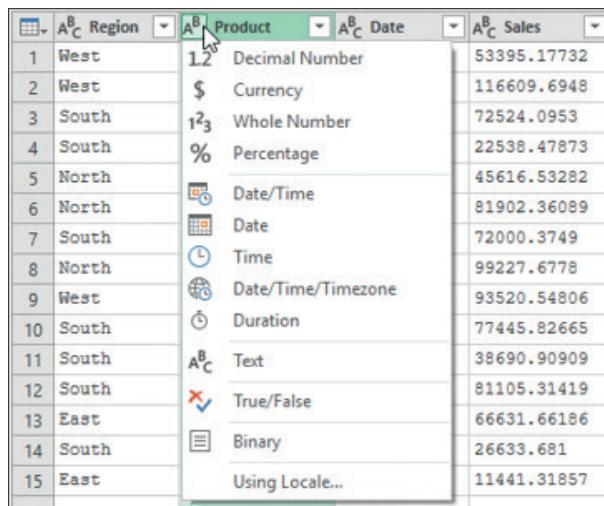


Рис. 3.10. Контекстное меню выбора типа данных

Еще один способ изменения типа данных связан с выбором выпадающего списка **Data Type** (Тип данных), расположенного на вкладке **Home** (Главная) в группе **Transform** (Преобразование). Помните, что сначала нужно выделить столбец, тип данных которого вы желаете изменить.

The screenshot shows the Power Query ribbon with the 'Transform' tab selected. The formula bar displays the formula = Table.TransformColumnTypes(Source,{{"Region", type text}, {"Product", type decimal}, {"Date", type date}, {"Sales", type decimal}}). A context menu is open over the 'Sales' column header, listing the same data type options as in Figure 3.10. The 'Decimal Number' option is selected. The 'Sales' column values are displayed in the table below.

	Region	Product	Date	Sales
1	South	1.2	100308.7559	
2	West	\$	65824.37301	
3	North	123	78488.77647	
4	South	%	36987.84956	
5	North	Date/Time	28013.17286	
6	North	Date	18192.7951	
7	North	Time	113945.7647	
8	South	Date/Time/Timezone	64663.23978	
9	North	Duration	93012.55135	
10	West	Text	35464.07375	
11	South	True/False	54665.24841	
12	East	Binary	34044.0828	
13	South	Using Locale...	8672.084433	
14	North		100861.4465	
15	East		72629.19944	

Рис. 3.11. Изменение типа данных столбца

Навигация по строкам и столбцам в окне предварительного просмотра таблицы данных работает точно так же, как и в Excel. Например, чтобы выделить все содержимое таблицы, достаточно нажать комбинацию клавиш **Ctrl+A**.

Панель Query Settings (Параметры запроса)

Панель **Query Settings** (Параметры запроса) располагается в правой части окна Power Query и при этом разделена на две части: **Properties** (Свойства) и **APPLIED STEPS** (Примененные шаги). Если вы не видите панель на экране, вам стоит активировать ее, нажав на одноименную кнопку **Query Settings** (Параметры запроса) на вкладке **View** (Просмотр), как показано на рис. 3.12.

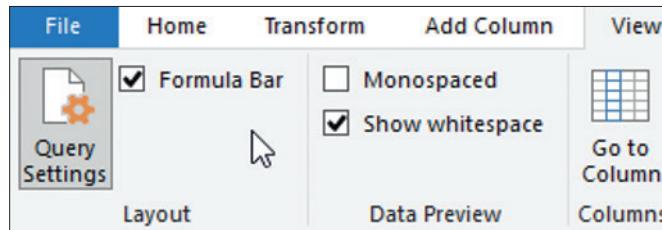


Рис. 3.12. Параметры запроса

Теперь посмотрим, как можно менять свойства запроса.

Изменение свойств запроса

При открытии окна Power Query на правой панели **Properties** (Свойства) будет содержаться имя таблицы, которое вы можете изменить на более подходящее вашему запросу. При закрытии и загрузке измененных данных обратно в Excel новая таблица получит то же имя, что и исходная таблица из Power Query.

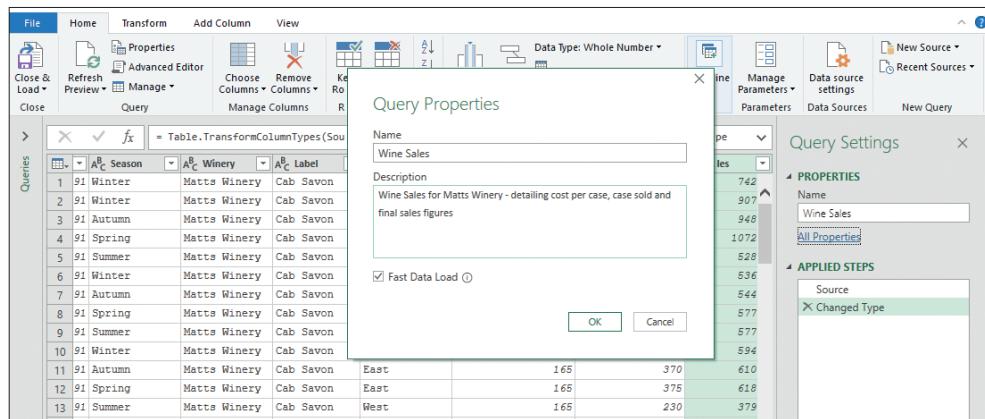


Рис. 3.13. Свойства таблицы

Имя таблицы может быть использовано в коде на языке M при обращении к ней в Power Query.

Работа с примененными шагами

Панель **APPLIED STEPS** (Примененные шаги) в правой части окна Power Query содержит все действия, произведенные над данными в рамках запроса. Эти шаги добавляются в список всякий раз, когда вы выполняете те или иные

операции в ленте. Шаги могут включать в себя операции удаления столбца, подключения к данным или изменения заголовка колонки – все эти действия так или иначе меняют наши данные, что и представляет собой процесс создания запроса.

Последние добавленные шаги могут быть удалены путем нажатия комбинации клавиш **Ctrl+Z** или при помощи крестика слева от шага. При наведении мыши на крестик он окрашивается в красный цвет, как показано на рис. 3.14.



Рис. 3.14. Удаление шага запроса

Все шаги записываются при помощи команд, которые иначе называются запросами на языке M. Именно этот язык используется в Power Query для создания всех без исключения шагов в запросах.

Удаление нескольких шагов одновременно

Вместо того чтобы удалять шаги запроса по одному, что может занять немало времени, у вас есть возможность выделить конкретный шаг на панели примененных шагов и удалить все оставшиеся пункты до конца списка. Для этого нажмите правой кнопкой мыши на нужный шаг и в появившемся контекстном меню выберите пункт **Delete Until End** (Удалить до конца), как показано на рис. 3.15.

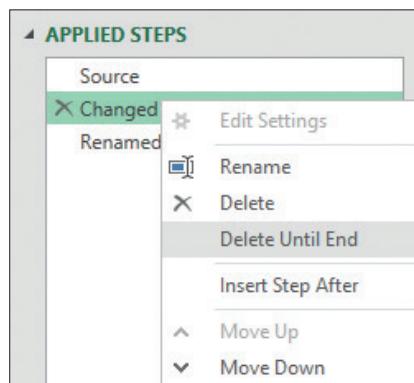


Рис. 3.15. Удаление нескольких шагов запроса одновременно

Теперь посмотрим, как можно документировать и переименовывать шаги запросов.

Документирование и переименование шагов

Этот раздел чрезвычайно важен – в нем будут описаны действия, к которым вы определенно должны привыкнуть, работая с Power Query. Дело в том, что эти

действия позволяют вам аккуратно документировать все шаги, примененные к данным в рамках запроса. К тому же при одновременной работе нескольких людей с одним набором данных или необходимости передать свои запросы коллегам все будут видеть документацию, что поможет понять, для чего служат те или иные шаги.

Все описания, добавленные к свойствам шагов, будут автоматически вставлены в итоговый код, который доступен в окне **Advanced Editor** (Расширенный редактор).

Примечание

Вы не обязаны делать какие-то пометки в синтаксисе для обозначения комментариев к коду, Power Query сделает это за вас.

1. Щелкните правой кнопкой мыши на шаге на панели **APPLIED STEPS** (Примененные шаги) и выберите пункт **Properties** (Свойства) в контекстном меню, как показано на рис. 3.16.

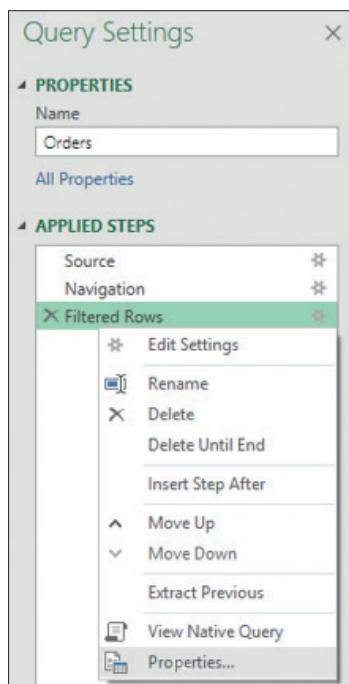


Рис. 3.16. Добавление описания к шагу

2. Откроется диалоговое окно **Step Properties** (Свойства шагов), в котором вы можете ввести описание для именованного шага, как видно на рис. 3.17.

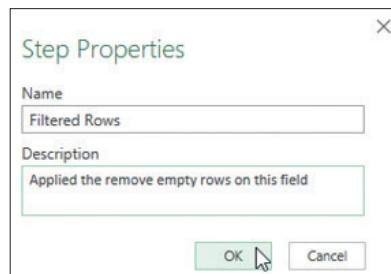


Рис. 3.17. Диалоговое окно со свойствами шагов

3. Нажмите на кнопку **OK** для сохранения свойств шага и возвращения на панель примененных шагов.

Помимо добавления описаний к шагам также важно, чтобы можно было легко понять предназначение того или иного шага при беглом взгляде на список шагов.

Часто бывает, что названия шагов недостаточно хорошо описывают производимые действия. В этом случае полезно будет их переименовать. Для этого необходимо выполнить следующие действия.

1. Щелкните правой кнопкой мыши на шаге на панели **APPLIED STEPS** (Примененные шаги) и выберите пункт **Rename** (Переименовать) в выпадающем списке.
2. Введите новое имя шага.
3. Подтвердите переименование шага клавишей **Enter**.

Если не вы один будете работать с рабочей книгой с ассоциированными запросами, стоит позаботиться о том, чтобы шаги были названы понятно.

Исследование вкладки View (Просмотр)

Здесь мы обратим внимание на три важных элемента, расположенных на вкладке **View** (Просмотр) в Power Query, а именно:

- строку ввода формул;
- инструмент изменения шрифта;
- просмотр зависимостей запроса.

Рассмотрим эти пункты более подробно.

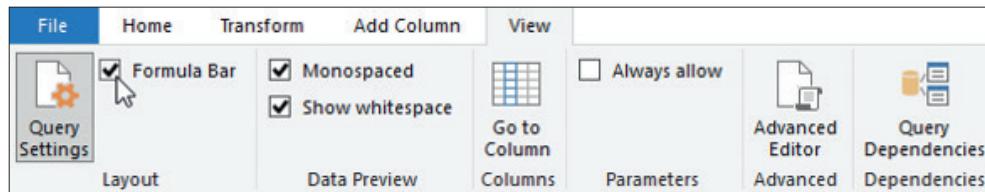
Строка ввода формул

При работе с Power Query бывает очень удобно, если *строка ввода формул*, показанная на рис. 3.18, видна. Даже если вы не имеете ни малейшего понятия о языке программирования Power Query (M), вы сможете получить о нем представление, периодически посматривая в строку формул, поскольку ее содержимое будет постоянно меняться в процессе выполнения преобразований данных. Это неплохой способ научиться программировать в Power Query.

The screenshot shows a Power Query formula bar with a formula: `= Table.TransformColumnTypes(Table1_Table,{{"Region", type text}, {"Product", type v}}`. Below the formula is a preview table with four columns: Region, Product, Date, and Sales. The first row shows 'South' in the Region column, 'Board2' in the Product column, '16/03/2015' in the Date column, and '100308.7559' in the Sales column.

Рис. 3.18. Стока ввода формул

Если при открытии Power Query вы не видите строку формул, перейдите на вкладку **View** (Просмотр) и установите флажок **Formula Bar** (Строка формул) в группе **Layout** (Структура), как показано на рис. 3.19.

Рис. 3.19. Вкладка **View** (Просмотр)

Теперь посмотрим, как можно изменить шрифт.

Изменение шрифта

Моноширинный шрифт отличается одинаковой шириной всех символов в строке и равными интервалами между ними. Эта особенность может существенно помочь при визуальном восприятии столбцов с длинными кодами товаров или при выборе схемы разбиения столбца на несколько колонок.

Для установки моноширинного шрифта в таблице необходимо выполнить следующие действия.

1. Откройте запрос в Power Query.
2. Перейдите на вкладку **View** (Просмотр) и в группе **Data Preview** (Предварительный просмотр данных) установите флажок **Monospaced** (Моноширинный). В окне предпросмотра данных шрифт поменяется на моноширинный.
3. Обратите внимание на отличия в восприятии информации. На рис. 3.20 показано окно предпросмотра со шрифтом по умолчанию.

The screenshot shows the Power Query data preview window with a table of products. The columns are labeled **Product**, **Product Code**, **Date**, and **Sales**. The data includes various product names like 'Board5', 'Game6', 'Game1', etc., along with their corresponding codes and financial values. The font used for the preview is monospaced, as specified in the previous steps.

Рис. 3.20. Коды товаров шрифтом по умолчанию

А на рис. 3.21 шрифт заменен на моноширинный.

A ^B _C Product	A ^B _C Product Code	A ^B _C Date	1.2 Sales
Board5	E729TFO605258916421-600PT	06/09/2012	53395.1773
Game6	I883VXW767844292755-246GX	23/02/2016	116609.6948
Game1	P514YXZ415181521507-575CU	20/09/2013	72524.0953
Game5	I938LN2479834468789-860ON	24/12/2010	22538.4787
Dyno4	H247MKU458573986837-770SK	28/03/2012	45616.5328
Game5	V777ZJI221878500906-685EV	23/03/2014	81902.3609

Рис. 3.21. Коды товаров моноширинным шрифтом

Моноширинный шрифт позволяет гораздо легче сравнивать длины строк и подсчитывать символы. Кроме того, в колонке с кодом товара теперь отчетливо видно, в каком месте можно разбить столбец надвое.

Просмотр зависимостей запроса

В правой части ленты на вкладке **View** (Просмотр) располагается кнопка **Query Dependencies** (Зависимости запроса). Она может оказаться очень полезной при работе с чужими моделями данных. Кнопка выводит в графическом виде перечисление всех таблиц в модели с указанием связей между ними, как показано на рис. 3.22.

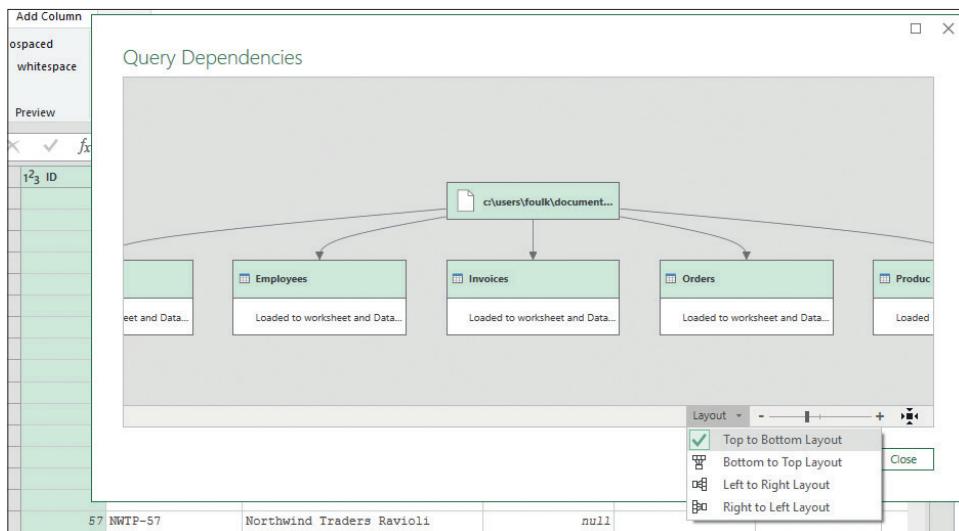


Рис. 3.22. Зависимости запроса

Окно зависимостей запроса показывает все без исключения связи между запросами. Чтобы все таблицы уместились в окне, необходимо нажать на иконку **Fit to Screen** (По размеру экрана), расположенную в нижней правой части окна, что видно по рис. 3.23.

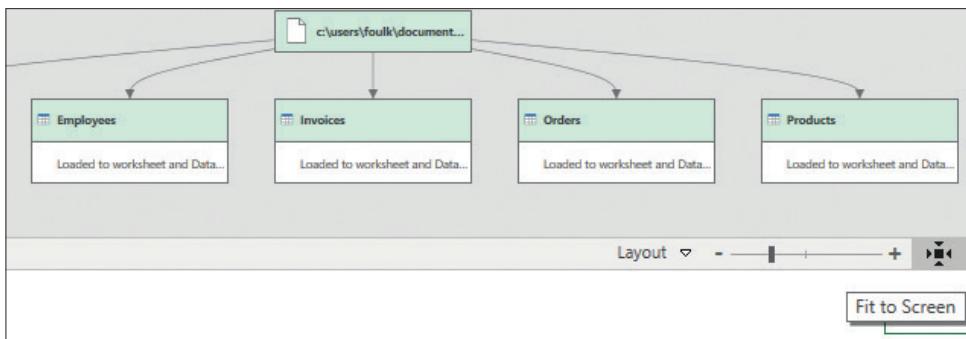


Рис. 3.23. Расположение окон по размеру экрана

Чтобы изменить расположение окон, щелкните на иконке выпадающего списка **Layout** (Структура) и выберите наиболее подходящий способ размещения таблиц в окне, как показано на рис. 3.24.

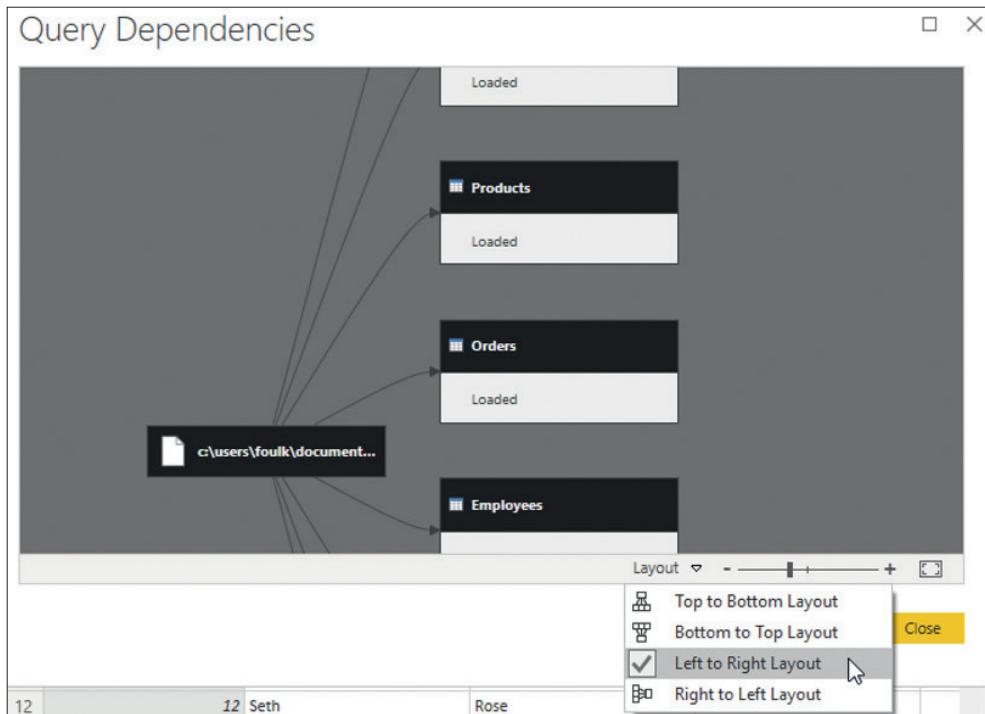


Рис. 3.24. Изменение схемы расположения таблиц в окне

Использование кнопки перехода к столбцу

Возможность перехода к конкретному столбцу бывает очень полезной при работе с огромными таблицами со множеством столбцов. Если вам необходимо быстро найти столбец по наименованию, нажмите на кнопку **Go to Column** (Перейти к столбцу) в группе **Column** (Столбцы), как показано на рис. 3.25.

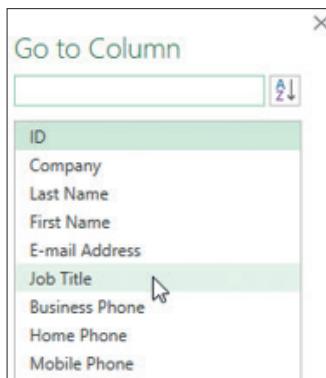


Рис. 3.25. Быстрый поиск столбца по наименованию

Теперь, когда мы ознакомились с вкладкой **View** (Просмотр), пришло время перейти к рассмотрению расширенного редактора.

Использование расширенного редактора

Каждый раз, когда вы производите то или иное изменение в структуре данных в рамках Power Query, вы, по сути, запускаете строку кода. Все эти строки хранятся в окне *расширенного редактора* (**Advanced Editor**), в котором вы можете вручную создавать или изменять запросы при помощи языка Power Query – M.

Очень важно хорошо понимать, что из себя представляет и как работает расширенный редактор Power Query. Зачем? Представьте, что вы активно работаете с данными: редактируете шаги, меняете их местами и т. д., а дойдя до заключительного шага, видите, что возникла ошибка. И чтобы эту ошибку обнаружить, локализовать и исправить, вам придется открыть расширенный редактор, а значит, нужно понимать, как он работает.

Чтобы продемонстрировать расширенный редактор в действии, давайте выполним несколько шагов для преобразования данных, а затем воспользуемся расширенным редактором, чтобы переименовать столбец с предыдущего шага. Это приведет к возникновению ошибки в коде, и нам нужно будет понять, что именно произошло, чтобы решить проблему.

1. Используя файл **SSGFilter.xlsx**, мы преобразуем таблицу данных в рабочую книгу при помощи Power Query.
2. На вкладке **Data** (Данные) в Excel нажмите на кнопку **From Table/Range** (Из таблицы/диапазона) в группе **Get & Transform** (Получить и преобразовать данные).
3. Откроется окно Power Query. Продублируйте столбец **DATE of HIRE**, чтобы мы могли разбить его на части, как показано на рис. 3.26.

The screenshot shows the Power Query ribbon with the 'DATE of HIRE' column selected. A context menu is open, and the 'Duplicate Column' option is highlighted.

A ^B _C DIVISION	A ^B _C DEPT	DATE of HIRE	1.2 GROSS
Munerton	Cobrella	1983/	5
Munerton	Mankay Falls	1985/	5
View Tabue	Slangsgrow	1990/	3
Soningdale	Shewe	1988/	2
View Tabue	Mankay Falls	1990/	5
Soningdale	Cobrella	1983/	6
Munerton	Cobrella	1990/	5
Soningdale	Shewe	1987/	4
Soningdale	Mankay Falls	1987/	6

Рис. 3.26. Создание дубликата колонки

4. Выделите новый столбец DATE of HIRE – Copy, щелкнув по нему.
5. Теперь воспользуемся выпадающей кнопкой **Split Column** (Разделить столбец) и выберем пункт **By Delimiter** (По разделителю), как показано на рис. 3.27, чтобы разбить столбец.

The screenshot shows the Power Query ribbon with the 'DATE of HIRE' column selected. The 'Split Column' dropdown is open, and the 'By Delimiter' option is highlighted.

A ^B _C DIVISION	A ^B _C DEPT	1.2 HRS	1.2 HOURLY RATE
Munerton		/15	21.5
Munerton		/25	12.5
View Tabue		/01	13.3
Soningdale		/12	7.22
View Tabue	Mankay Falls	1990/07/26	42
Soningdale	Cobrella	1983/06/12	40

Рис. 3.27. Разбиение колонки

6. Выберите пользовательский разделитель / (косая черта) и установите переключатель **Split at** (Разделение по) в положение **Left-most delimiter** (Самый левый разделитель), как показано на рис. 3.28.

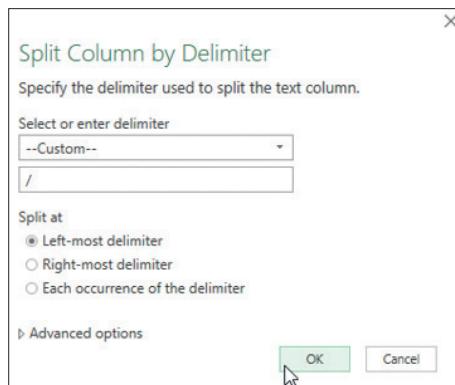


Рис. 3.28. Разбиение колонки по разделителю

7. Нажмите на кнопку **OK**, чтобы продолжить.
8. Удалите дополнительную колонку, нажав на ней правой кнопкой мыши и выбрав пункт **Remove** (Удалить), поскольку нам надо оставить только столбец с указанием года.
9. Переименуйте столбец **Year** в **Year Hired**, дважды щелкнув мышью на его заголовке и введя новое имя. Затем подтвердите действие, нажав на клавишу **Enter**.
10. Заметьте, что на панели **APPLIED STEPS** (Примененные шаги) выстроился целый ряд шагов, что видно по рис. 3.29.

DATE of HIRE - Copy.3"}			
	1.2 HOURLY RATE	1.2 GROSS PAY	1 ² ₃ Year
40	21.5	860	1983
35.5	12.5	443.75	1985
35.5	13.3	472.15	1990
40	7.22	288.8	1988
42	16.75	703.5	1990
40	12.6	504	1983
40	21.5	860	1990
35	24	840	1987
40	12.6	504	1987
35.5	12.5	443.75	1986
35.5	13.3	472.15	1985
32	5.5	176	1990
35.5	13.3	472.15	1984
40	8.75	350	1988

Query Settings

Properties

Name: Table1

All Properties

Applied Steps

- Source
- Navigation
- Changed Type
- Duplicated Column
- Split Column by Delimiter
- Changed Type1
- Renamed Columns
- Removed Columns

Рис. 3.29. Примененные шаги

11. Нажмите на кнопке **Advanced Editor** (Расширенный редактор), находящейся на вкладке **Home** (Главная), чтобы увидеть, какой код был сгенерирован при создании наших шагов. Окно расширенного редактора показано на рис. 3.30.

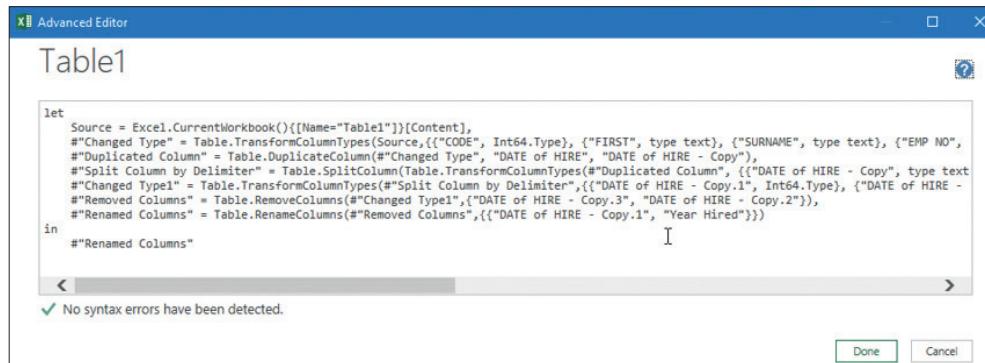


Рис. 3.30. Расширенный редактор

12. Просмотрите код в редакторе, который был создан автоматически, и попытайтесь понять его смысл. Обратите внимание на созданные Power Query метки для каждого шага.
13. Измените в последней строке листинга `Year Hired` на `YEAR HIRED`, поменяв регистр, как показано на рис. 3.31.

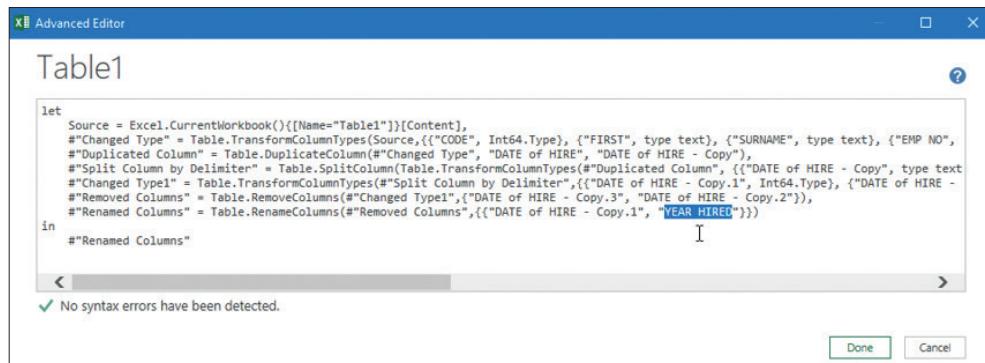


Рис. 3.31. Изменение кода в расширенном редакторе

14. По окончании нажмите на кнопку **OK**, и вы увидите, что данные обновились.
15. Перетащите столбец `YEAR HIRED`, расположив его непосредственно после колонки `DATE of HIRE`, как показано на рис. 3.32.

DEPT	DATE OF HIRE	YEAR HIRED	URLY RATE	GROSS PAY	YEAR HIRED
Cobrella	1983/04/15	40	21.5	860	1983
Mankay Falls	1985/01/25	35.5	12.5	443.75	1985
Slangsgrow	1990/02/01	35.5	13.3	472.15	1990
Shewe	1988/05/12	40	7.22	288.8	1988
Mankay Falls	1990/07/26	42	16.75	703.5	1990
Cobrella	1983/06/12	40	12.6	504	1983
Cobrella	1990/12/30	40	21.5	860	1990
Shewe	1987/06/05	35	24	840	1987

Рис. 3.32. Перемещение столбца

16. Мы забыли переименовать колонку DATE of HIRE в HIRE DATE. Дважды щелкните на поле DATE of HIRE и напишите HIRE DATE, после чего нажмите на кнопку Enter.

17. Данные выглядят нормально, без видимых ошибок.

Но если щелкнуть на последнем шаге на панели примененных шагов, мы увидим ошибку в окне предварительного просмотра. Текст ошибки, которая показана на рис. 3.33, гласит, что столбец с именем DATE of HIRE не был обнаружен.

18. Справа от текста ошибки вы обнаружите кнопку Go to Error (Перейти к ошибке), которую можно нажать, чтобы увидеть дополнительную информацию. Но лучше всего будет открыть расширенный редактор, чтобы понять, что случилось.

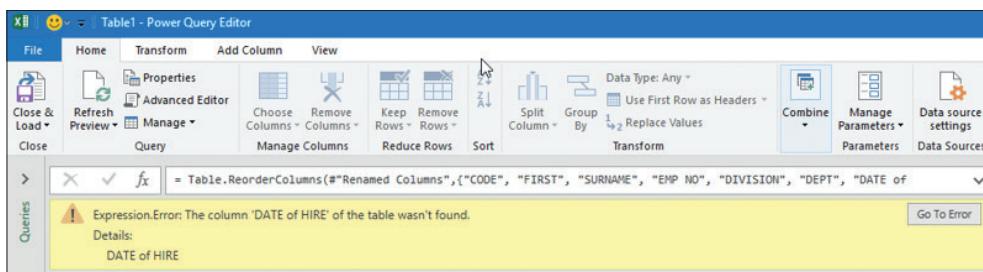


Рис. 3.33. Ошибка в коде

Именно в такие моменты начинаешь понимать, как работает код, как он устроен и как важен порядок шагов при преобразовании данных в Power Query.

```

let
    Source = Excel.CurrentWorkbook(){[Name="Table1"]}[Content],
    #"Changed Type" = Table.TransformColumnTypes(Source,{{"CODE", Int64.Type}, {"FIRST", type text}, {"SURNAME", type text}, {"EMP NO", type text}, {"DIVISION" type text}, {"HIRE DATE", type date}}),
    #"Renamed Column1" = Table.RenameColumns(#"Changed Type",{{"DATE OF HIRE", "HIRE DATE"}}),
    #"Duplicated Column" = Table.DuplicateColumn(#"Renamed Column1", "DATE OF HIRE", "DATE OF HIRE - Copy"),
    #"Split Column by Delimiter" = Table.SplitColumn(Table.TransformColumnTypes(#"Duplicated Column",{{("DATE OF HIRE - Copy", type text)}}, "en-ZA"), "DATE OF HIRE - Copy", 1),
    #"Changed Type1" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{("DATE OF HIRE - Copy.1", Int64.Type), {"DATE OF HIRE - Copy.2", Int64.Type}}}),
    #"Removed Columns" = Table.RemoveColumns(#"Changed Type1",{"DATE OF HIRE - Copy.3", "DATE OF HIRE - Copy.2"}),
    #"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"DATE OF HIRE - Copy.1", "YEAR HIRED"}}),
    #"Reordered Columns" = Table.ReorderColumns(#"Renamed Columns",{"CODE", "FIRST", "SURNAME", "EMP NO", "DIVISION", "DEPT", "DATE OF HIRE", "YEAR HIRED", "HIRE DATE"})
in
#>

```

✓ No syntax errors have been detected.

Done Cancel

Рис. 3.34. Пример кода в расширенном редакторе

19. Определите, где возникла ошибка.

```

let
    Source = Excel.CurrentWorkbook(){[Name="Table1"]}[Content],
    pe1 = Table.TransformColumnTypes(Source,{{"CODE", Int64.Type}, {"FIRST", type text}, {"SURNAME", type text}, {"EMP NO", type text}, {"DIVISION" type text}, {"HIRE DATE", type date}}),
    #"Renamed Column1" = Table.RenameColumns(#"Changed Type",{{"DATE OF HIRE", "HIRE DATE"}}),
    Column = Table.DuplicateColumn(#"Renamed Column1", "DATE OF HIRE", "DATE OF HIRE - Copy"),
    m by Delimiter" = Table.SplitColumn(Table.TransformColumnTypes(#"Duplicated Column",{{("DATE OF HIRE - Copy", type text)}}, "en-ZA"), "DATE OF HIRE - Copy", 1),
    pe1" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{("DATE OF HIRE - Copy.1", Int64.Type), {"DATE OF HIRE - Copy.2", Int64.Type}}}),
    lums" = Table.RemoveColumns(#"Changed Type1",{"DATE OF HIRE - Copy.3", "DATE OF HIRE - Copy.2"}),
    lums" = Table.RenameColumns(#"Removed Columns",{{"DATE OF HIRE - Copy.1", "YEAR HIRED"}}),
    Columns" = Table.ReorderColumns(#"Renamed Columns",{"CODE", "FIRST", "SURNAME", "EMP NO", "DIVISION", "DEPT", "DATE OF HIRE", "YEAR HIRED", "HIRE DATE"})
in
#>

```

✓ No syntax errors have been detected.

Done Cancel

Рис. 3.35. Проблема обнаружена

20. Измените значение, вызвавшее ошибку.

```

let
    Source = Excel.CurrentWorkbook(){[Name="Table1"]}[Content],
    pe1 = Table.TransformColumnTypes(Source,{{"CODE", Int64.Type}, {"FIRST", type text}, {"SURNAME", type text}, {"EMP NO", type text}, {"DIVISION" type text}, {"HIRE DATE", type date}}),
    #"Renamed Column1" = Table.RenameColumns(#"Changed Type",{{"DATE OF HIRE", "HIRE DATE"}}),
    Column = Table.DuplicateColumn(#"Renamed Column1", "DATE OF HIRE", "DATE OF HIRE - Copy"),
    m by Delimiter" = Table.SplitColumn(Table.TransformColumnTypes(#"Duplicated Column",{{("DATE OF HIRE - Copy", type text)}}, "en-ZA"), "DATE OF HIRE - Copy", 1),
    pe1" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{("DATE OF HIRE - Copy.1", Int64.Type), {"DATE OF HIRE - Copy.2", Int64.Type}}}),
    lums" = Table.RemoveColumns(#"Changed Type1", {"DATE OF HIRE - Copy.3", "DATE OF HIRE - Copy.2"}),
    lums" = Table.RenameColumns(#"Removed Columns",{{"DATE OF HIRE - Copy.1", "YEAR HIRED"}}),
    Columns" = Table.ReorderColumns(#"Renamed Columns", {"CODE", "FIRST", "SURNAME", "EMP NO", "DIVISION", "DEPT", "DATE OF HIRE", "YEAR HIRED", "HIRE DATE"})
in
#>

```

✓ No syntax errors have been detected.

Done Cancel

Рис. 3.36. Проблема устранена

21. Нажмите на кнопку **Done** (Готово) по окончании работы, и данные в окне предварительного просмотра должны восстановиться, от ошибки не осталось и следа.

В следующем разделе мы узнаем, как добавлять комментарии в код в расширенном редакторе.

Добавление комментариев в расширенном редакторе

Работая с программным кодом, часто возникает необходимость *добавления комментариев* к тексту – либо для вас самих, либо для тех, кто будет работать с вашими данными и кодом. И хотя подробно с языком программирования, использующимся в Power Query, вы познакомитесь позже в этой книге, тему комментариев мы рассмотрим уже здесь.

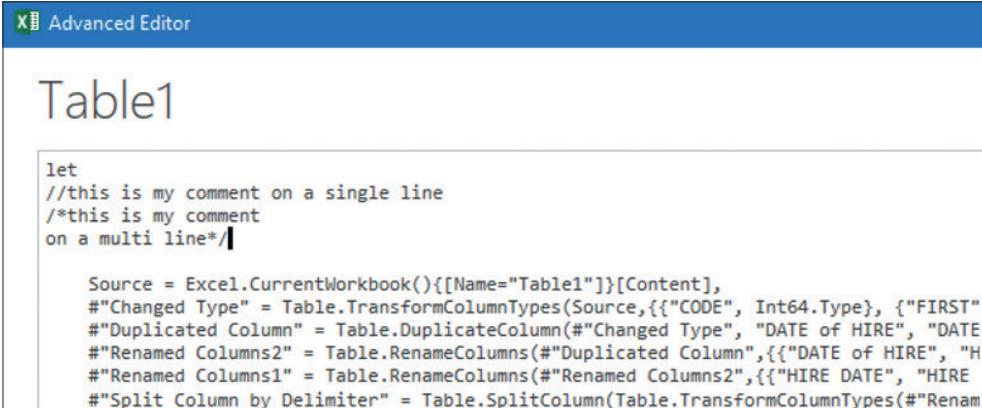
1. Откройте окно расширенного редактора Power Query.
2. Установите курсор в начало текста.
3. Для написания однострочного комментария введите приведенный ниже текст в строку и нажмите на клавишу **Enter**:

```
//this is my comment
```

4. Для создания многострочного комментария введите следующий текст. Обратите внимание на знаки начала и конца комментария.

```
/*this is my
multi-line comment*/
```

На рис. 3.37 показано, как это будет выглядеть на экране.



The screenshot shows the 'Advanced Editor' window for a table named 'Table1'. The code area contains the following:

```
let
    //this is my comment on a single line
    /*this is my comment
    on a multi line*/
    Source = Excel.CurrentWorkbook(){[Name="Table1"]}[Content],
    #"Changed Type" = Table.TransformColumnTypes(Source,{{"CODE", Int64.Type}, {"FIRST", typeText}}),
    #"Duplicated Column" = Table.DuplicateColumn(#"Changed Type", "DATE of HIRE", "DATE #Renamed Columns2"),
    #"Renamed Columns2" = Table.RenameColumns(#"Duplicated Column",{{"DATE of HIRE", "HIRE"}, {"Renamed Columns1", "HIRE DATE"}, {"Split Column by Delimiter", "HIRE"}})
in
    #"Renamed Columns2"
```

Рис. 3.37. Создание комментариев в расширенном редакторе Power Query

Теперь научимся добавлять комментарии в код прямо в строке формул.

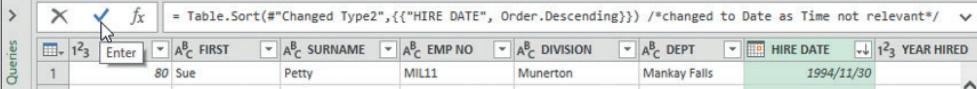
Добавление комментариев в строке формул

Комментарии могут быть добавлены для каждого шага в процессе написания кода с использованием строки формул. Это самый быстрый способ добавить описание к шагам запроса в Power Query. И хотя вы добавляете комментарии в строке формул, видны они будут также и в расширенном редакторе. Для создания комментария в строке формул необходимо выполнить следующие действия.

- Найдите и выделите столбец HIRE DATE.
- Щелкните на заголовке столбца для отображения кода в строке формул.
- Ведите следующий текст в конце строки кода:

```
/*changed to Date as Time not relevant here*/
```

На рис. 3.38 показано, как это будет выглядеть в строке формул.



The screenshot shows the Power Query formula bar with the following content:

```
= Table.Sort(#"Changed Type2",{{"HIRE DATE", Order.Descending}}) /*changed to Date as Time not relevant here*/
```

The cursor is positioned over the word "Enter" in the formula bar, which is highlighted in blue. The formula bar also displays the column headers and data for the "HIRE DATE" column.

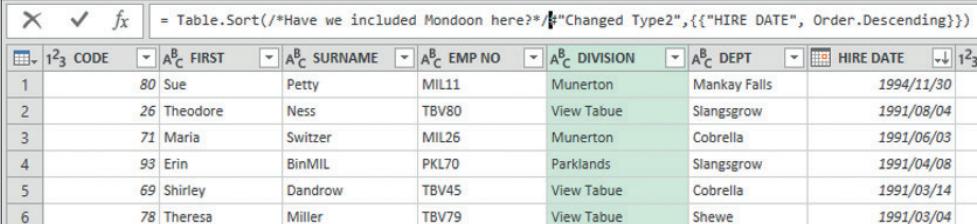
Рис. 3.38. Комментарий в строке формул

- Щелкните на галке слева от строки формул или нажмите на клавишу **Enter** для сохранения комментария.

Комментарий появится при просмотре текста запроса в расширенном редакторе и пропадет из кода столбца HIRE DATE в строке формул.

Как сохранить комментарии в строке формул?

Чтобы комментарии оставались видны в строке формул для конкретного шага запроса, что иногда бывает полезно при выполнении сложных преобразований, просто добавьте комментарий в середину кода, а не в конец, как показано на рис. 3.39.



The screenshot shows the Power Query formula bar with the following content:

```
= Table.Sort(/"Have we included Monday here?"/,"Changed Type2",{{"HIRE DATE", Order.Descending}})
```

The cursor is positioned over the word "Have" in the formula bar, which is highlighted in blue. The formula bar displays the column headers and data for the "HIRE DATE" column.

Рис. 3.39. Комментарий в строке формул

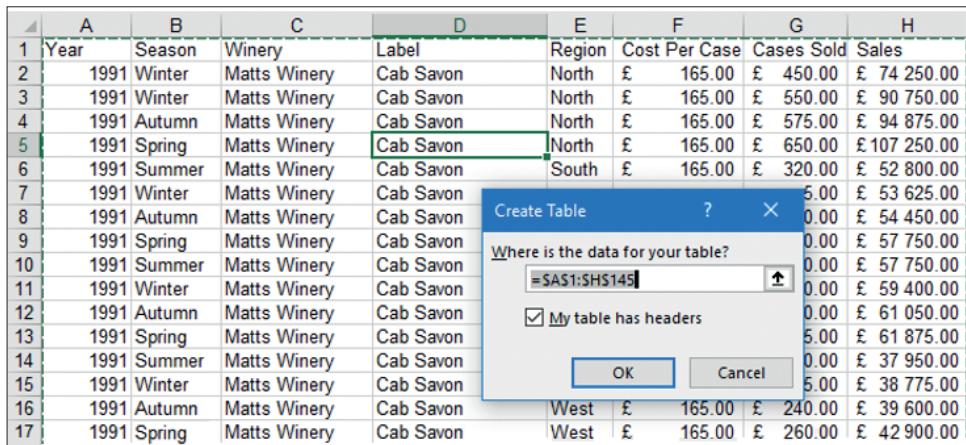
На этом мы завершаем краткий экскурс по элементам интерфейса Power Query. Пришло время научиться создавать запросы непосредственно из Excel.

Создание простого запроса

В предыдущих разделах книги вы достаточно узнали об интерфейсе инструмента Power Query и научились работать с предварительным просмотром. Здесь же мы опишем процесс создания простого запроса с использованием умной таблицы Excel с нуля.

Первое, с чем вам необходимо определиться, – это с тем, откуда брать исходные данные. В данном примере мы будем использовать файл MattsWinery.xlsx.

- После открытия рабочей книги нажмите сочетание клавиш **Ctrl+T** для запуска диалогового окна **Create Table** (Создание таблицы), как показано на рис. 3.40. Также вы можете воспользоваться кнопкой **Table/Range** (Из таблицы/диапазона) из группы **Get & Transform** (Получить и преобразовать данные) на вкладке **Data** (Данные).



The screenshot shows a portion of an Excel spreadsheet with columns labeled A through H. Rows 1 through 17 contain data. Row 1 is a header row with column titles: Year, Season, Winery, Label, Region, Cost Per Case, Cases Sold, and Sales. Rows 2 through 17 contain data points. A green selection box highlights the range from A1 to H17. Overlaid on the spreadsheet is a 'Create Table' dialog box. The dialog has a question mark icon, a close button, and an 'OK' button at the bottom. It contains the text 'Where is the data for your table?' followed by the formula '=S\$1:\$H\$14\$'. There is also a checked checkbox labeled 'My table has headers'. The background of the dialog is semi-transparent.

	A	B	C	D	E	F	G	H
1	Year	Season	Winery	Label	Region	Cost Per Case	Cases Sold	Sales
2	1991	Winter	Matts Winery	Cab Savon	North	£ 165.00	£ 450.00	£ 74 250.00
3	1991	Winter	Matts Winery	Cab Savon	North	£ 165.00	£ 550.00	£ 90 750.00
4	1991	Autumn	Matts Winery	Cab Savon	North	£ 165.00	£ 575.00	£ 94 875.00
5	1991	Spring	Matts Winery	Cab Savon	North	£ 165.00	£ 650.00	£ 107 250.00
6	1991	Summer	Matts Winery	Cab Savon	South	£ 165.00	£ 320.00	£ 52 800.00
7	1991	Winter	Matts Winery	Cab Savon		5.00	£ 53 625.00	
8	1991	Autumn	Matts Winery	Cab Savon		0.00	£ 54 450.00	
9	1991	Spring	Matts Winery	Cab Savon		0.00	£ 57 750.00	
10	1991	Summer	Matts Winery	Cab Savon		0.00	£ 57 750.00	
11	1991	Winter	Matts Winery	Cab Savon		0.00	£ 59 400.00	
12	1991	Autumn	Matts Winery	Cab Savon		0.00	£ 61 050.00	
13	1991	Spring	Matts Winery	Cab Savon		5.00	£ 61 875.00	
14	1991	Summer	Matts Winery	Cab Savon		0.00	£ 37 950.00	
15	1991	Winter	Matts Winery	Cab Savon		5.00	£ 38 775.00	
16	1991	Autumn	Matts Winery	Cab Savon	West	£ 165.00	£ 240.00	£ 39 600.00
17	1991	Spring	Matts Winery	Cab Savon	West	£ 165.00	£ 260.00	£ 42 900.00

Рис. 3.40. Создание умной таблицы

- Убедитесь в том, что вы выбрали правильный диапазон данных для будущего преобразования.
- Под выбранным диапазоном установите флажок **My table has headers** (Таблица с заголовками), чтобы Excel воспринял первую строку выбранного диапазона как заголовки.
- Щелкните на кнопке **OK**, чтобы преобразовать табличные данные в умную таблицу, и запустите Power Query. Окно Power Query появится поверх рабочего листа Excel.
- Для начала изменим имя таблицы, данное ей по умолчанию, на **Wine Sales**.
- Всегда полезно снабжать таблицу описанием, так что нажмите на ссылку **All Properties** (Все свойства) непосредственно под названием таблицы (в правой части окна), введите описание, а затем нажмите на кнопку **OK**, как показано на рис. 3.41.

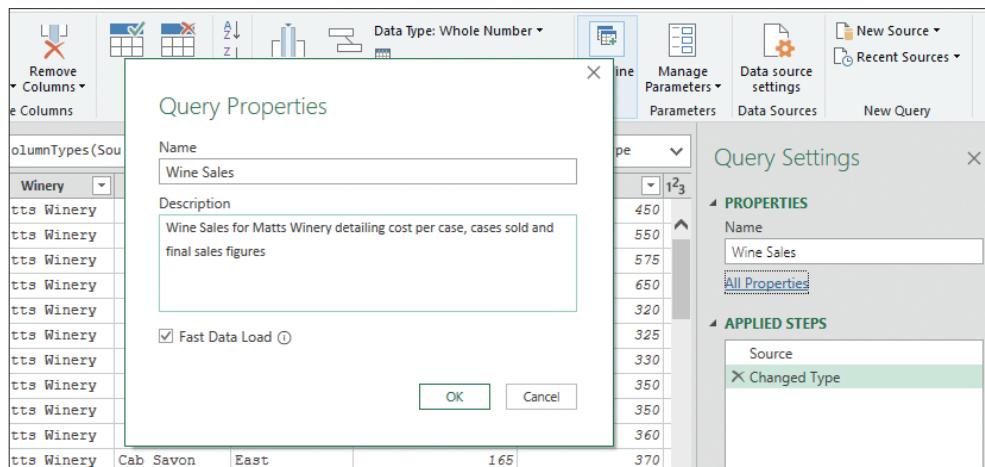


Рис. 3.41. Добавление описания запроса

Теперь можно приступать к преобразованию и очистке таблицы.
Посмотрите на запрос, представленный на рис. 3.42.

	Year	Season	Winery	Label	Region	Cost Per Case	Cases
1	1991	Summer	Matts Winery	Cab Savon	South	165	
2	1991	Winter	Matts Winery	Cab Savon	South	165	
3	1991	Autumn	Matts Winery	Cab Savon	South	165	
4	1991	Spring	Matts Winery	Cab Savon	South	165	
5	1991	Summer	Matts Winery	Cab Savon	East	165	
6	1991	Winter	Matts Winery	Cab Savon	East	165	
7	1991	Autumn	Matts Winery	Cab Savon	East	165	

Рис. 3.42. Так будет выглядеть ваш запрос перед применением следующих шагов

В следующих главах мы подробно поговорим обо всех этих действиях, а сейчас лишь быстро пробежимся по шагам, чтобы вы лучше поняли, как это работает.

- Измените столбцы **Region** и **Season** таким образом, чтобы они отображались в верхнем регистре. Для этого перейдите на вкладку **Transform** (Преобразование), найдите группу **Text Column** (Столбец «Текст»), нажмите на выпадающую кнопку **Format** (Форматировать) и выберите пункт **UPPERCASE** (ВЕРХНИЙ РЕГИСТР).
- Создайте копию запроса **Wine Sales**, открыв слева навигационную панель **Запросы**, щелкнув правой кнопкой мыши на запросе **Wine Sales** и выбрав пункт **Duplicate** (Дублировать).

3. Переименуйте запрос Wine Sales в Matts Sales, а второй запрос – в Prominent Sales, затем отфильтруйте столбец Winery так, чтобы в нем отображалась только соответствующая запросу винодельня.

После завершения преобразования данных вам может понадобиться переслать их обратно в Excel для дальнейшего анализа. Для этого необходимо сделать следующее.

1. В меню **File** (Файл) выберите пункт **Close and Load** (Закрыть и загрузить). Обо всех нюансах использования опции загрузки мы расскажем в следующем разделе.
2. Загрузив запросы обратно в Excel, вы обнаружите в правой части окна новую панель **Queries & Connections** (Запросы и подключения), как видно на рис. 3.43.

The screenshot shows a Microsoft Excel spreadsheet titled 'Matts Winery'. The data is organized into columns: Year, Season, Region, Winery, Type, and Sales. The 'Winery' column contains repeated entries for 'Matts Winery'. A green box highlights the 'Winery' column header. To the right of the spreadsheet is the 'Queries & Connections' pane. It displays two entries: 'Matts Sales' and 'Prominent Sales', each with a note indicating '48 rows loaded.'

Рис. 3.43. Панель **Запросы и подключения** в рабочей книге

Запросы автоматически добавляются на отдельные листы, поскольку в нашем примере мы воспользовались пунктом **Close and Load** (Закрыть и загрузить).

3. На панели **Queries & Connections** (Запросы и подключения) отображаются все запросы, созданные в Power Query. Дважды щелкните на запросе, чтобы открыть окно Power Query и продолжить процесс преобразования данных.
4. При наведении мышью на запрос в панели **Queries & Connections** вы увидите окно предварительного просмотра запроса с данными и основной информацией о запросе, а также кнопки возможных действий, как показано на рис. 3.44.

Prominent Winery

Year	Season	Region	Winery	Type	Cost Per Case	Cases Sold
1991	WINTER	NORTH	Prominent Wines	Merlat	148	660
1991	AUTUMN	NORTH	Prominent Wines	Merlat	148	665
1991	SPRING	NORTH	Prominent Wines	Merlat	148	670
1991	SUMMER	SOUTH	Prominent Wines	Merlat	148	325
1991	WINTER	SOUTH	Prominent Wines	Merlat	148	329
1991	AUTUMN	SOUTH	Prominent Wines	Merlat	148	330
1991	SPRING	SOUTH	Prominent Wines	Merlat	148	341
1991	SUMMER	NORTH	Prominent Wines	Merlat	148	660
1991	WINTER	EAST	Prominent Wines	Merlat	148	481

Columns [8]
Year, Season, Region, Winery, Type, Cost Per Case, Cases Sold, Sales
Last refreshed
09:00
Load status
Loaded to worksheet and Data Model
Data Sources [1]
c:\users\foulkes\documents\foulkes@work\packt\powerquery\chapter three\code\data\matthewinery-c.xlsx

VIEW IN WORKSHEET EDIT ... DELETE

Queries & Connections

Wine Sales 144 rows loaded.

Prominent_Sales 48 rows loaded.

Prominent Winery 48 rows loaded.

Рис. 3.44. Предпросмотр запроса со вспомогательной информацией

5. Справа от кнопки **Edit** (Изменить) вы увидите троеточие. Нажав на эту кнопку, вы сможете выбрать дополнительные действия с запросом.
6. В правой части окна предварительного просмотра располагается кнопка удаления запроса.
7. Если на листе Excel добавились данные, вы можете щелкнуть на панели **Queries & Connections** по соответствующему запросу и нажать на иконку **Refresh** (Обновить) справа от имени запроса, чтобы прогнать новую информацию через Power Query.

Теперь, когда вы научились создавать простой запрос, посмотрим на различные варианты использования опции **Load To...** (Загрузить в...) для отправки результатов запросов обратно в Excel.

Изучение опций Load To... (Загрузить в...)

В этом разделе мы рассмотрим все возможные опции в диалоговом окне **Load To...** (Загрузить в...) и различные варианты обратной загрузки данных из Power Query в Excel.

Для начала посмотрим на опции загрузки данных в Excel по умолчанию.

Изменение параметров загрузки данных в Excel по умолчанию

В Excel существуют определенные параметры загрузки данных по умолчанию. Чтобы посмотреть их и при необходимости изменить, нужно посетить вкладку **Data** (Данные) в Excel 2019. Значения параметров загрузки по умолчанию приведены ниже:

- отдельные запросы загружаются на новый рабочий лист;
- при загрузке множества запросов они будут помещены в модель данных.

Давайте посмотрим, где и как можно изменить эти параметры.

1. В Excel 2019 перейдите на вкладку **Data** (Данные).
2. Если вы уже находитесь в Power Query, откройте вкладку **File** (Файл) и выберите пункт **Query Options** (Параметры запроса) в меню **Options and settings** (Параметры и настройки), как показано на рис. 3.45.

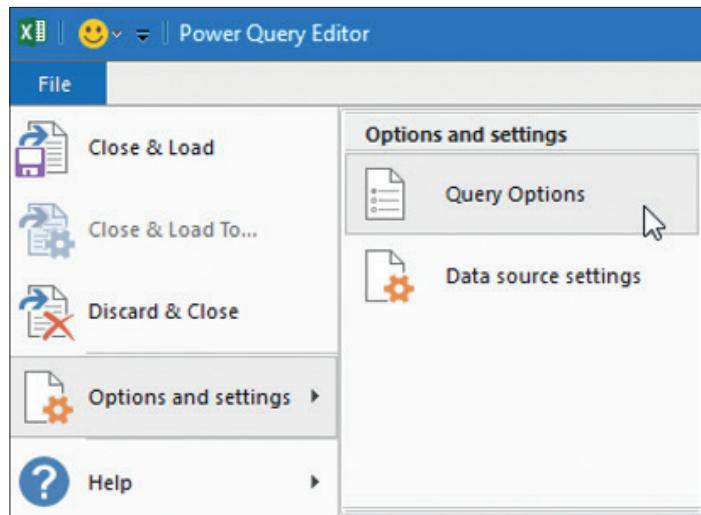


Рис. 3.45. Параметры запроса в Power Query

3. Нажмите на кнопку **Get Data** (Получить данные) и из выпадающего списка выберите пункт **Query Options** (Параметры запроса), как показано на рис. 3.46.

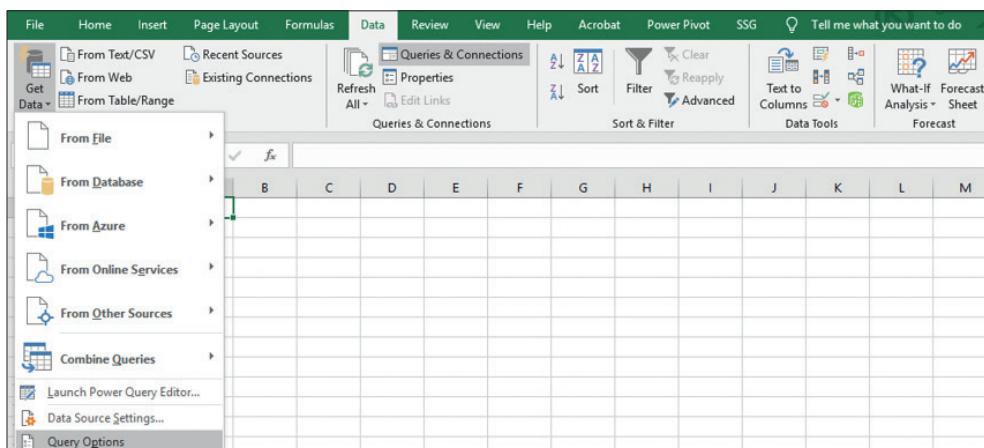


Рис. 3.46. Параметры запроса в Power Query

4. В разделе **GLOBAL** (ГЛОБАЛЬНЫЕ) выберите пункт **Data Load** (Загрузка данных). Вы также можете установить настройки по умолчанию только для текущей рабочий книги – для этого нужно в левой навигационной панели выбрать соответствующий пункт.
5. Обратите внимание, что верхний переключатель по умолчанию установлен в значение **Use standard load settings** (Использовать стандартные параметры загрузки), а это означает, что запрос будет загружаться только на лист.

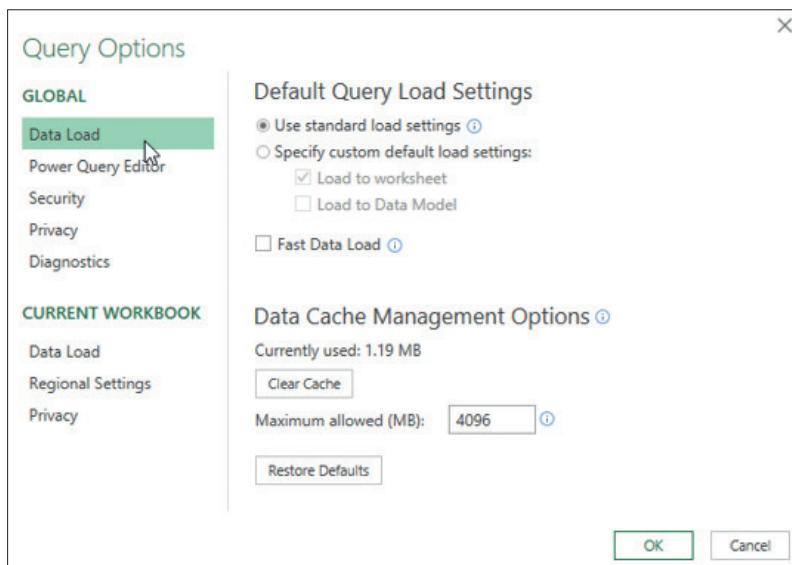


Рис. 3.47. Параметры загрузки запросов по умолчанию

6. Сделать так, чтобы запросы по умолчанию загружались на рабочий лист и в модель данных, можно, установив переключатель в значение **Specify custom default load settings** (Указать пользовательские параметры загрузки по умолчанию) и включив оба флашка: **Load to Worksheet** (Загрузить в лист) и **Load to Data Model** (Загрузить в модель данных).

Примечание

Не загружайте запросы на рабочий лист без необходимости. К примеру, если вы создали 25 запросов и выбрали вариант загрузки по умолчанию на рабочий лист, все 25 запросов будут загружены на отдельные листы в рабочей книге. Это довольно ресурсозатратная операция, и чаще всего вам не понадобятся все эти запросы в книге.

7. Можно загрузить данные быстрее, если установить флашок **Fast Data Load** (Быстрая загрузка данных), расположенный под флашком **Load to Data Model** (Загрузить в модель данных). Эта опция позволит загружать запросы намного быстрее, но во время загрузки производительность Excel 2019 может значительно снизиться.

8. Также важной опцией в разделе параметров загрузки для текущей рабочей книги является *фоновая загрузка* (**Background Data Load**). Установка этой настройки позволит всем запросам в рабочей книге сохраняться в кеше. Конечно, скорость перемещения между шагами в запросе и между разными запросами возрастет, но стоит отметить, что при этом непременно увеличится и нагрузка на центральный процессор и оперативную память.
9. Чтобы установить этот параметр в Power Query, откройте на вкладке **File** (Файл) пункт **Query Options** (Параметры запроса) и убедитесь, что в левой навигационной панели пункт **Data Load** (Загрузка данных) выбран в разделе **CURRENT WORKBOOK** (ТЕКУЩАЯ КНИГА). Затем установите флажок **Allow data preview to download in the background** (Разрешить скачивание в фоновом режиме для предварительного просмотра данных), как показано на рис. 3.48, и нажмите на кнопку **OK**.

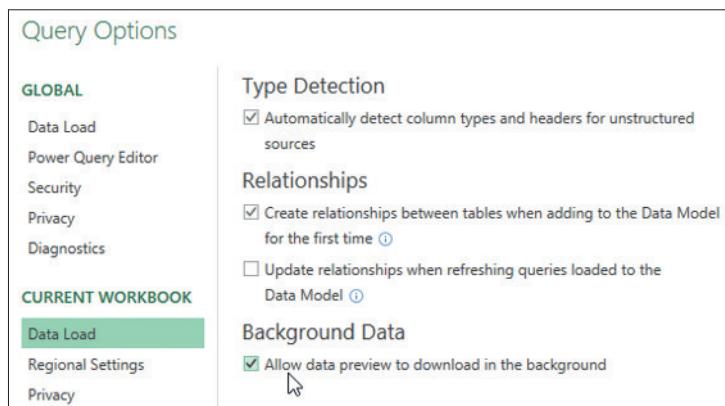


Рис. 3.48. Настройки загрузки данных в текущей рабочей книге

При работе с множеством связанных источников данных бывает полезно время от времени очищать кеш. Если вы получаете ошибки при установке соединения со связанными источниками в онлайне, первое, о чём вам стоит позаботиться, – это об очистке кеша в Power Query.

10. Чтобы очистить кеш данных, необходимо на вкладке **File** (Файл) открыть пункт **Query Options** (Параметры запроса) и убедиться, что в левой навигационной панели пункт **Data Load** (Загрузка данных) выбран в разделе **GLOBAL** (ГЛОБАЛЬНЫЕ). В разделе **Data Cache Management Options** (Параметры управления кешем данных) нажмите на кнопку **Clear Cache** (Очистить кеш), как показано на рис. 3.49, а затем на кнопку **OK**.

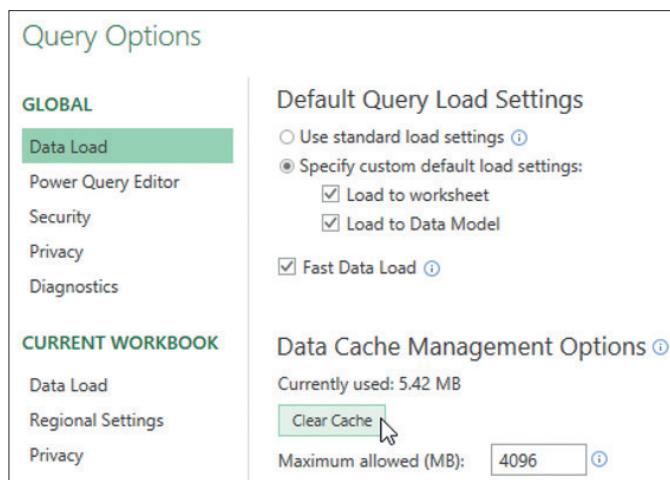


Рис. 3.49. Очистка кеша в Power Query

Теперь, когда мы научились менять параметры загрузки данных по умолчанию, давайте посмотрим, как загружать запросы вручную.

Загрузка запросов на рабочий лист вручную

Для загрузки запросов на рабочий лист в ручном режиме необходимо проделать следующие действия.

1. Откройте рабочую книгу с названием SSGLoadDataM.xlsx.
2. Щелкните на таблице и перейдите на вкладку **Data** (Данные).
3. В группе **Get & Transform** (Получить и преобразовать данные) нажмите на кнопку **From Table/Range** (Из таблицы/диапазона). Откроется окно Power Query.
4. Выполните преобразование данных, подготовив их к обратной отправке в Excel.
5. Перейдите на вкладку **Home** (Главная) и в группе **Close** (Закрыть) нажмите на кнопку **Close & Load To...** (Закрыть и загрузить в...), как показано на рис. 3.50.

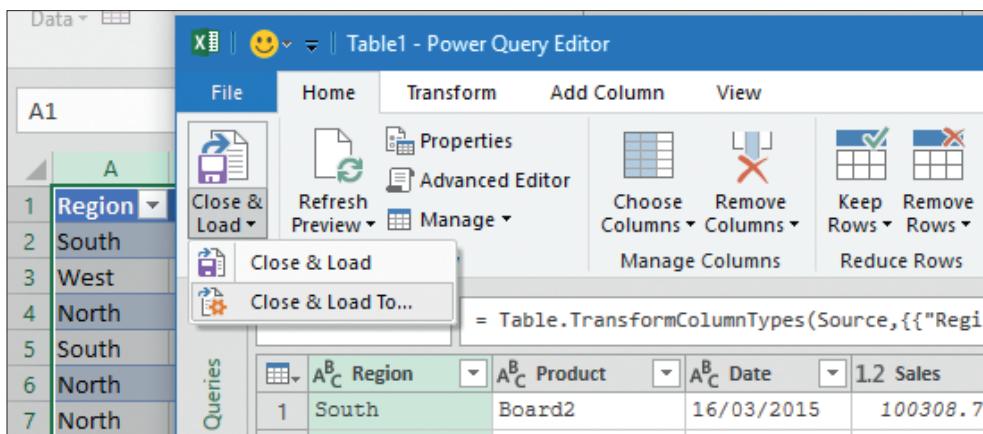


Рис. 3.50. Пункт меню Close & Load To... (Закрыть и загрузить в...)

- Снова откроется Excel, а с ним и диалоговое окно Import Data (Импорт данных) с целым рядом опций, как видно на рис. 3.51.

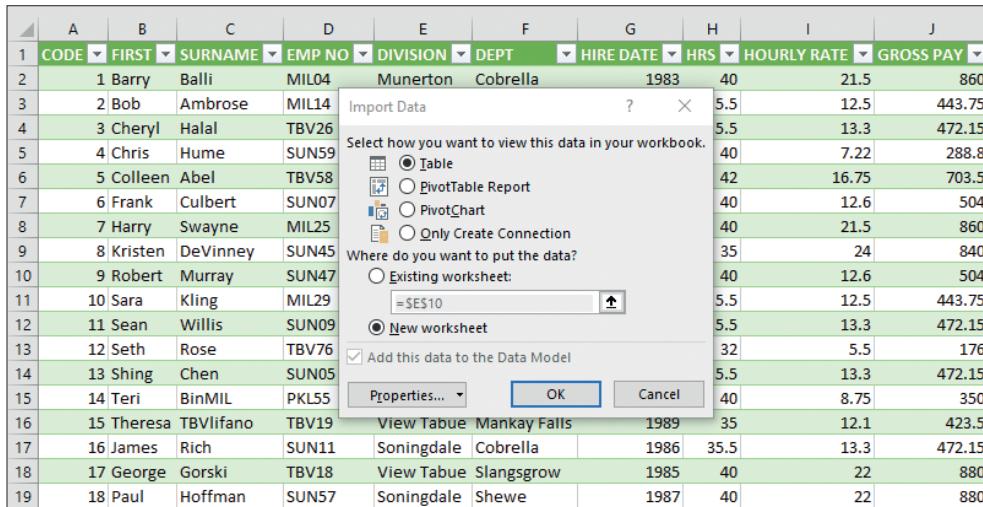


Рис. 3.51. Варианты загрузки данных обратно в Excel

- Выбор в этом окне отвечает за то, как вы будете видеть загружаемые данные и где они будут размещаться на листе. Первый вариант, выбранный по умолчанию, означает, что данные будут загружены в таблицу на листе. Для этого примера ниже в данном же окне вы можете установить второй переключатель в положение Existing worksheet (Имеющийся лист).
- Выберите желаемое расположение данных, нажав на кнопку со стрелкой, которая перенесет вас на лист.

9. Данные будут помещены на лист незамедлительно, справа появится панель **Queries & Connections** (Запросы и подключения), а в строке заголовков – пункт **Query Tools** (Работа с запросами), как показано на рис. 3.52.

The screenshot shows a Microsoft Excel spreadsheet titled 'SSGLoadDataM-C.xlsx'. The 'Data' tab is selected in the ribbon. On the far right of the ribbon, the 'Query Tools' tab is highlighted. To the right of the main workspace, there is a 'Queries & Connections' pane. This pane displays a list of connections, with 'Table1' highlighted, indicating it has 18 rows loaded. The main workspace shows a table with columns labeled A through K, containing data such as names, addresses, and dates.

Рис. 3.52. Загруженные данные в Excel

При этом пункт **Query Tools** (Работа с запросами) будет виден только при выделении мышью загруженной таблицы. При щелчке за пределами таблицы этот пункт будет скрываться.

10. Будучи на рабочем листе Excel, вы сохраните доступ к настройкам параметров загрузки данных **Load To...** (Загрузить в...), поскольку вам может понадобиться выгрузить запрос или выбрать альтернативную опцию загрузки.
11. Чтобы выгрузить существующий запрос с листа, необходимо обратиться к правой панели **Queries & Connections** (Запросы и подключения), чтобы найти загруженную таблицу – в нашем случае это таблица по умолчанию с названием **Table1**.
12. Щелкните по таблице правой кнопкой мыши и выберите пункт **Load To...** (Загрузить в...) из предложенного меню, как показано на рис. 3.53.

The screenshot shows a table of data with columns Q, R, S, T, U, DEPT, HIRE DATE, HRS, HOURLY RATE, and GROSS PAY. The 'Queries & Connections' pane is open, showing 'Table1' with '18 rows loaded.' A context menu is open over 'Table1', with 'Load To...' highlighted.

Q	R	S	T	U	DEPT	HIRE DATE	HRS	HOURLY RATE	GROSS PAY
Cobrella		1983	40	21.5	860				
Mankay Falls		1985	35.5	12.5	443.75				
Slangsgrow		1990	35.5	13.3	472.15				
Shewe		1988	40	7.22	288.8				
Mankay Falls		1990	42	16.75	703.5				
Cobrella		1983	40	12.6	504				
Cobrella		1990	40	21.5	860				
Shewe		1987	35	24	840				
Mankay Falls		1987	40	12.6	504				
Mankay Falls		1986	35.5	12.5	443.75				
Mankay Falls		1985	35.5	13.3	472.15				
Cobrella		1990	32	5.5	176				
Cobrella		1984	35.5	13.3	472.15				
Mankay Falls		1988	40	8.75	350				
Mankay Falls		1989	35	12.1	423.5				
Cobrella		1986	35.5	13.3	472.15				

Рис. 3.53. Опция Load To... (Загрузить в...)

13. Снова откроется диалоговое окно Import Data (Импорт данных).
14. На этот раз выберите пункт Only Create Connection (Только создать подключение). Заметьте, что в этом случае нижние переключатели с новым и имеющимся листами будут недоступны, что видно по рис. 3.54.

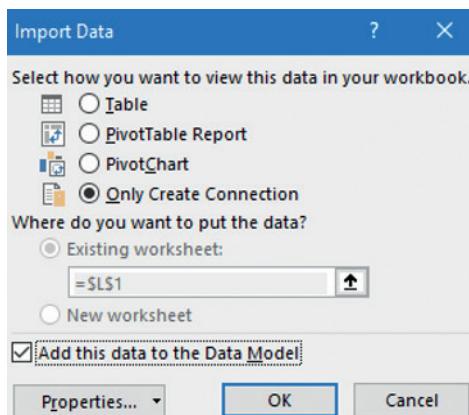


Рис. 3.54. Диалоговое окно Import Data (Импорт данных)

15. Нажмите на кнопку OK, чтобы сохранить изменения.
- Появится информационное диалоговое окно, предупреждающее о том, что таблица будет удалена с рабочего листа, как показано на рис. 3.55.

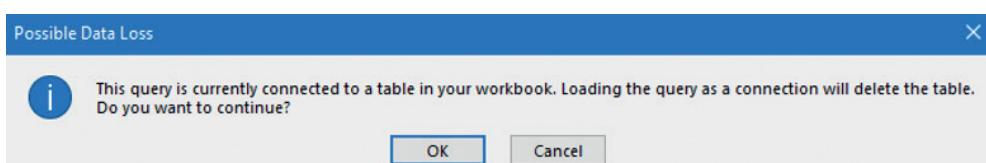


Рис. 3.55. Диалоговое окно, предупреждающее о возможной потере данных

16. Нажмите на кнопку **OK** для подтверждения намерений.

Данные оказались удалены с рабочего листа.

Давайте пройдемся по оставшимся опциям в диалоговом окне **Import Data** (Импорт данных).

- На этот раз выберем вариант загрузки **PivotTable Report** (Отчет сводной таблицы).
- Щелкните правой кнопкой мыши на подключение **Table1**.
- Выберите вариант **PivotTable Report** (Отчет сводной таблицы) в диалоговом окне **Import Data** (Импорт данных).
- Выберите место на листе для размещения сводной таблицы – мы поместим ее в ячейку L1.
- Нажмите на кнопку **OK**, чтобы сохранить изменения.
- На листе появится пустая сводная таблица вместе с панелью **Fields** (Поля сводной таблицы).
- Закройте панель **Queries & Connections** (Запросы и подключения), поскольку сейчас она вам не нужна. Вы всегда можете открыть ее вновь, щелкнув по иконке **Queries & Connections** (Запросы и подключения) на вкладке **Data** (Данные).
- Выберите на правой панели необходимые поля сводной таблицы, отвечающие вашим требованиям, как показано на рис. 3.56. Заметьте, что при выделении сводной таблицы на листе становится видимым пункт **PivotTable Tools** (Работа со сводными таблицами) в строке заголовков, предоставляющий богатые возможности по настройке сводных таблиц.

	E	F	G	H	I	J	K	L	M	N
	DEPT	DATE OF HIRE	HRS	HOURLY RATE	GROSS PAY		Row Labels	Sum of GROSS PAY	Sum of HRS	Sum
1	DEPT									
2	Munerton	Cobrella	15-Apr-83	40 £	21.50 £	860.00				
3	Munerton	Mankay Falls	25-Jan-85	36 £	12.50 £	443.75				
4	View Tabue	Slangsgrow	01-Feb-90	36 £	13.30 £	472.15				
5	Soningdale	Shewe	12-May-88	40 £	7.22 £	288.80				
6	View Tabue	Mankay Falls	26-Jul-90	42 £	16.75 £	703.50				
7	Soningdale	Cobrella	12-Jun-83	40 £	12.60 £	504.00				
8	Munerton	Cobrella	30-Dec-90	40 £	21.50 £	860.00				
9	Soningdale	Shewe	05-Jun-87	35 £	24.00 £	840.00				
10	Soningdale	Mankay Falls	10-Jun-87	40 £	12.60 £	504.00				
11	Munerton	Mankay Falls	24-Dec-86	36 £	12.50 £	443.75				
12	Soningdale	Mankay Falls	05-Jul-85	36 £	13.30 £	472.15				
13	View Tabue	Cobrella	05-Apr-90	32 £	5.50 £	176.00				
14	Soningdale	Cobrella	08-Aug-84	36 £	13.30 £	472.15				
15	Parklands	Mankay Falls	07-Jun-88	40 £	8.75 £	350.00				
16	View Tabue	Mankay Falls	26-Feb-89	35 £	12.10 £	423.50				
17	Soningdale	Cobrella	11-Oct-86	36 £	13.30 £	472.15				
18	View Tabue	Slangsgrow	07-May-85	40 £	22.00 £	880.00				
19	Soningdale	Shewe	19-Dec-87	40 £	22.00 £	880.00				

Рис. 3.56. Настройка отчета сводной таблицы

9. Если в какой-то момент времени вы заметили, что опция **Close & Load To...** (Закрыть и загрузить в...) стала недоступна в Power Query, используйте вариант **Close & Load** (Закрыть и загрузить), после чего войдите на панель **Queries & Connections** (Запросы и подключения), щелкните правой кнопкой мыши по вашему подключению и выберите пункт **Load To...** (Загрузить в...) из выпадающего меню для изменения типа загрузки данных.

Примечание

Если вы загружаете данные только в модель данных, размер итогового файла будет существенно меньше.

СОВЕТЫ ПО ПРОФИЛИРОВАНИЮ ДАННЫХ

В данном разделе мы поговорим о наборе иконок на вкладке **View** (Просмотр) в Power Query, посвященных *профилированию данных* (data profiling), при обращении из Excel и Power BI. Мы узнаем, как отображать данные моноширинным шрифтом, а также как добавлять информацию о *качестве столбца* (Column Quality), *распределении столбцов* (Column Distribution) и *профиле столбца* (Column Profile) в Power BI и что они означают. В данный момент полный спектр работы в области профилирования данных доступен только в Power BI. Но что дают нам эти данные о данных? Если говорить просто, то они позволяют сделать работу в Power Query более визуальной и обеспечивают необходимыми инструментами при замене и редактировании данных.

Во всех примерах, связанных с профилированием данных, мы будем использовать рабочую книгу *SSGProfiling.xlsx*.

И хотя возможности профилирования данных присутствуют в Power Query также при обращении к нему из Excel, они весьма ограничены по сравнению с использованием Power BI. Так что при описании этих инструментов мы будем пользоваться именно Power BI. На следующих двух рисунках показаны отличия на ленте Power Query при обращении к нему из Excel и Power BI.

На рис. 3.57 мы видим панель инструментов Power Query при открытии из Excel, а на рис. 3.58 – из Power BI. Обратите внимание на группу **Data Preview** (Предварительный просмотр данных) – насколько более богатый выбор функций у вас появляется при переходе из Power BI.

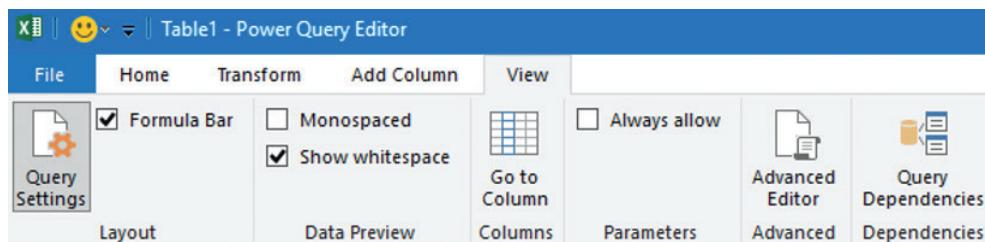


Рис. 3.57. Опции профилирования данных в Power Query при обращении из Excel 2019

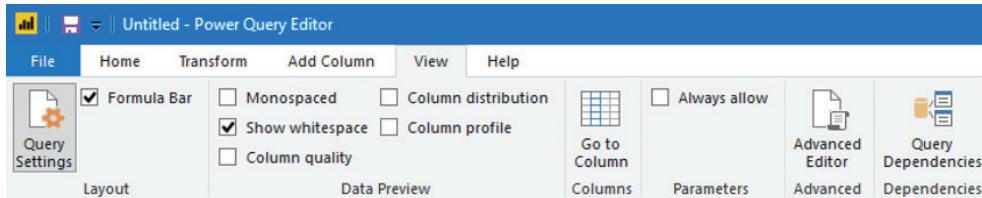


Рис. 3.58. Опции профилирования данных в Power Query при обращении из Power BI

Как видите, на рис. 3.58 показано больше флаажков в области профилирования данных. Убедитесь, что при работе с примерами из этого раздела вы используете именно Power BI.

Как мы уже сказали, опции, связанные с профилированием данных, включают в себя информацию о качестве столбца (Column Quality), распределении столбцов (Column Distribution) и профиле столбца (Column Profile). Давайте установим одноименные флаажки в группе Data Preview (Предварительный просмотр данных) и рассмотрим их использование в Power Query.

1. Откройте рабочую книгу SSGProfiling.xlsx и создайте новый запрос на основании таблицы данных.
2. В окне Power Query перейдите на вкладку View (Просмотр).
3. В группе Data Preview (Предварительный просмотр данных) установите флаажки Column Quality (Качество столбца), Column Distribution (Распределение столбцов) и Column Profile (Профиль столбца), чтобы активировать функции, связанные с профилированием данных.
4. После этого вы обнаружите заметные изменения в интерфейсе Power Query, показанные на рис. 3.59.

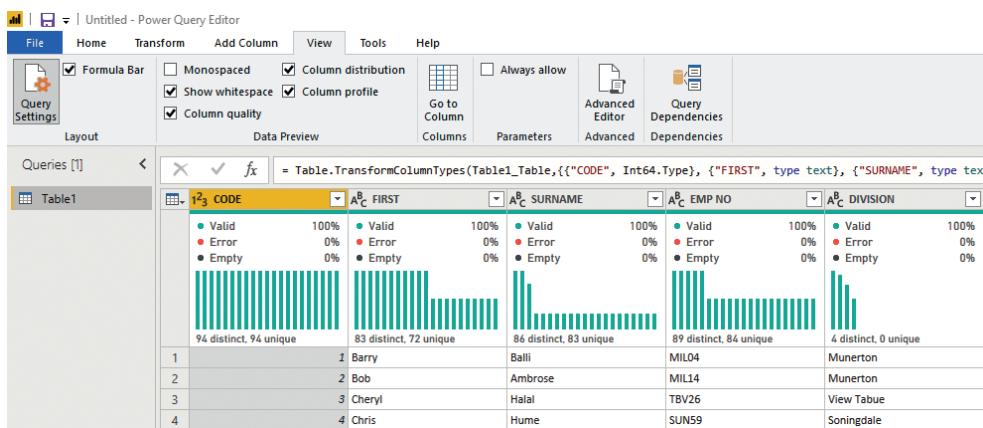


Рис. 3.59. Power Query с активированными возможностями профилирования данных

Рассмотрим каждую из этих опций отдельно и узнаем, как можно воспользоваться связанными с ними преимуществами. Если вы раньше никогда не работали с этим функционалом, вам будет полезно узнать, где какая информация представлена. На рис. 3.60 показано, где какие элементы размещены.

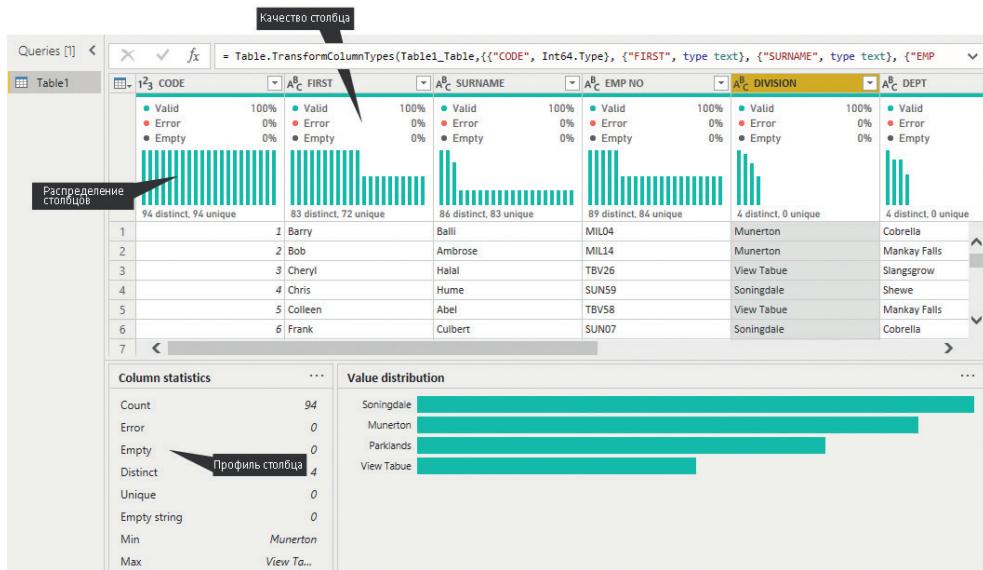


Рис. 3.60. Опции профилирования данных

В следующих разделах мы подробно коснемся каждой опции из области профилирования данных.

Профиль столбца

Чтобы посмотреть *профиль столбца* (column profile), щелкните по нему в таблице. Интерфейс инструмента изменится таким образом, чтобы можно было визуально оценить набор данных и структуру столбца, как показано на рис. 3.61.

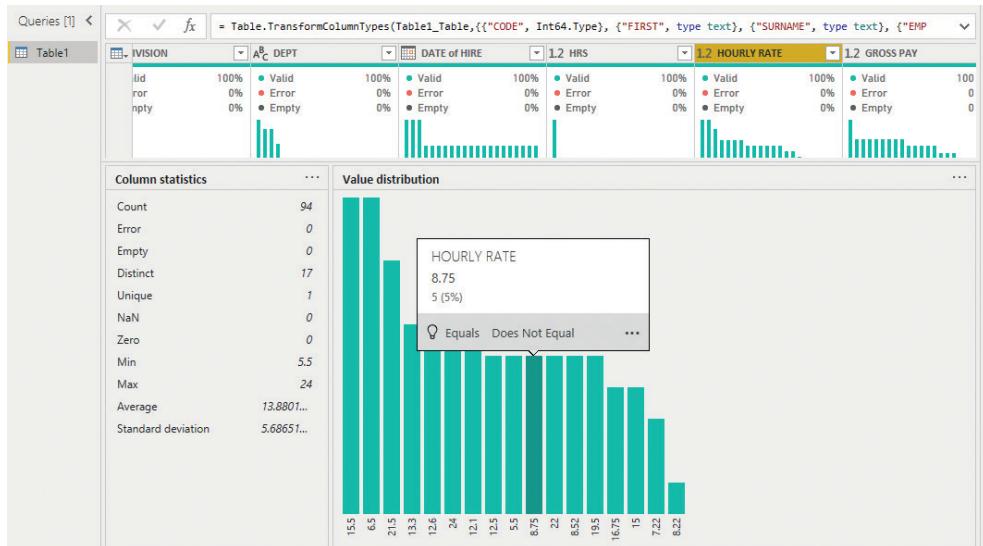


Рис. 3.61. Щелкните на столбце, чтобы увидеть его профиль

По умолчанию при профилировании данных Power Query отображает информацию по первой тысяче строк в таблице. Но вы можете изменить эту настройку, щелкнув мышью по иконке **Status Bar** (Строка состояния) и выбрав пункт **Column profiling based on entire data set** (Проанализировать столбец на основании всего набора данных), как показано на рис. 3.62.

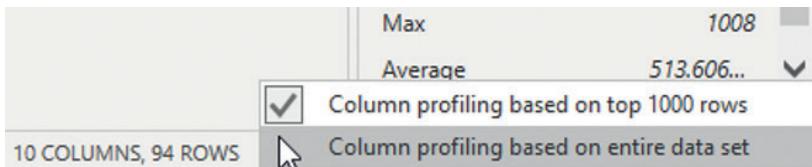


Рис. 3.62. Изменение настройки профилирования данных

Профиль столбца включает в себя статистические данные о колонке, расположющиеся слева от гистограммы распределения значений. Эти области доступны только для просмотра, взаимодействовать и изменять эти сведения нельзя.

Качество столбца

Щелкнув на столбец **CODE** в наборе данных, можно заметить, что он содержит 3 % ошибок, что отражено в статистике под заголовком столбца, показанной на рис. 3.63. Непосредственно под столбцом располагается небольшая диаграмма, демонстрирующая «состояние здоровья» или *качество столбца* (column quality).

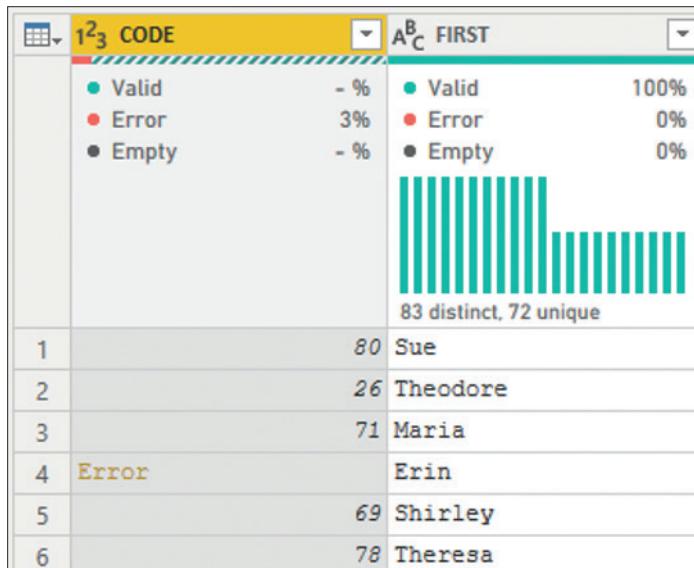


Рис. 3.63. Графическое представление качества данных

При наведении мыши на область отображения качества столбца появится всплывающее окно, в котором вы можете попытаться устранить ошибки в столбце. Выберите пункт меню **Replace Errors** (Заменить ошибки), чтобы устранить проблему в запросе и заменить некорректные значения, как показано на рис. 3.64.

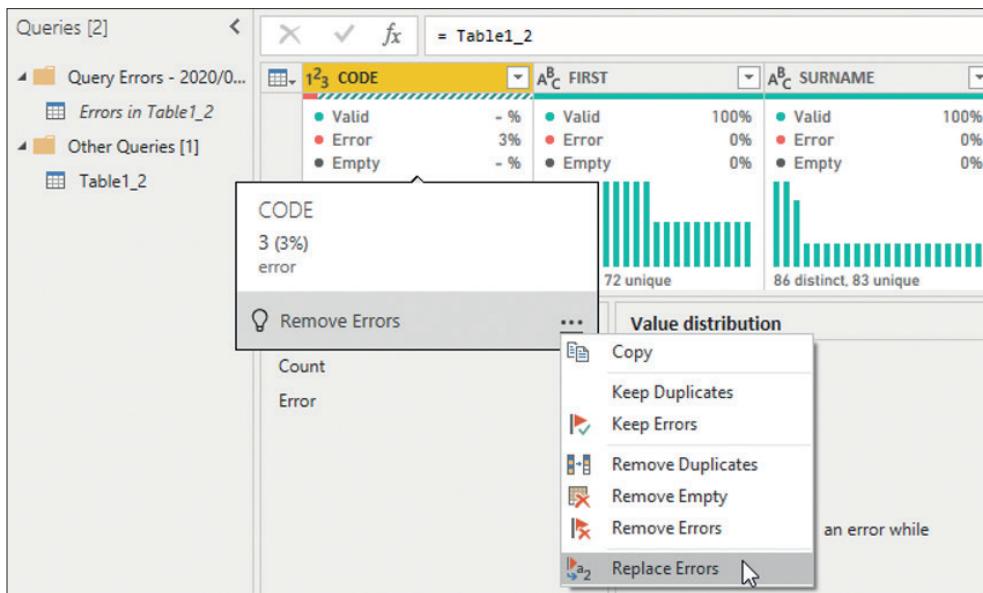


Рис. 3.64. Замена ошибок в столбце CODE

Это самый простой способ исправления ошибок в запросе!

Распределение столбцов

Распределение столбцов (column distribution) представляет собой великолепный визуальный инструмент анализа данных, но на этом этапе он не даст вам дополнительной ценной информации. Щелкнув правой кнопкой мыши на визуализации распределения столбцов, мы увидим меню с некоторыми опциями, которые были доступны в разделе с определением качества столбца, как видно по рис. 3.65.

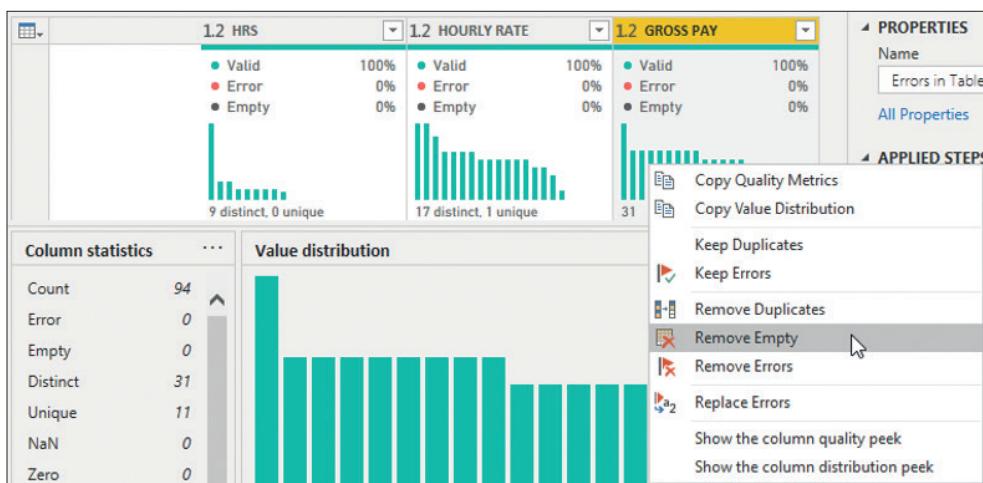


Рис. 3.65. Пример отображения распределения столбцов

Профиль столбца является наиболее мощным инструментом при анализе данных, поскольку, по сути, представляет расширенную версию визуализации распределения столбцов.

ЗАКЛЮЧЕНИЕ

В данной главе вы впервые начали работать непосредственно с Power Query. Вы также много узнали о параметрах запросов, что является важной частью процесса их загрузки. Теперь вы способны производить тонкую настройку окружения загрузки запросов в соответствии с вашими требованиями.

Кроме того, вы уже должны обладать достаточными знаниями для создания простого запроса в Power Query и возвращения преобразованных данных обратно в Excel или Power BI посредством опции **Load To...** (Загрузить в...).

Мы также коснулись области профилирования данных и узнали, как можно улучшить визуальную интерпретацию данных внутри запроса.

В следующей главе мы научимся подключаться к разнообразным источникам данных, среди которых веб-страница, базы данных Access, рабочие книги Excel, текстовые файлы и файлы с разделителями, а также поговорим о множественных подключениях. В завершение покопаемся в настройках источников данных.

Глава 4

Подключение к различным источникам данных

В данной главе мы научимся подключаться к самым разным источникам информации при помощи инструмента **Get & Transform** (Получить и преобразовать данные), также именуемого Power Query, а заодно рассмотрим настройки и параметры подключений. Существует огромное множество способов подключиться к разнообразным источникам данных, и в этой главе мы пройдемся по всему спектру, начиная от простых рабочих книг Excel и заканчивая базами данных. При этом мы будем использовать как Excel, так и Power BI.

В этой главе мы затронем следующие темы:

- подключение к таблице или диапазону;
- подключение к веб-странице;
- подключение к реляционной базе данных;
- пользовательские подключения;
- настройки источников данных.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Примеры для этой главы можно загрузить по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

Сопутствующие видеофрагменты располагаются по ссылке: https://www.youtube.com/watch?v=pyszzEJlpB4&list=PLeLcvrwLe1860_GJEzs47WaZXwZjwTN83&index=5&t=24s.

После прочтения предыдущих глав книги вы должны быть хорошо знакомы с интерфейсом Power Query. Также мы предполагаем, что вы уже умеете создавать базовые запросы и загружать преобразованные данные обратно в Excel или Power BI с использованием опции **Load To...** (Загрузить в...).

КРАТКОЕ ВВЕДЕНИЕ В ТЕОРИЮ БАЗ ДАННЫХ

Существуют разные типы баз данных. Есть, к примеру, *транзакционные базы данных* (transactional database), в которых вся информация хранится исключительно в строках. Допустим, если у вас есть данные о товаре в таблице, то все они будут храниться в одной строке. Транзакционные базы данных обычно

рассчитаны на взаимодействие при помощи строковых операций **Create** (Создать), **Retrieve** (Извлечь), **Update** (Обновить) и **Delete** (Удалить) – **CRUD**. В таких базах данных обычно предусмотрено наличие первичных ключей в таблицах, так что если вам понадобится изменить категорию товаров, достаточно будет сделать это в единственной таблице, тогда как все связанные товары обновятся автоматически. Иногда при использовании хранилищ данных возникает ситуация, когда один товар заменяет другой, который, в свою очередь, становится недоступным или выходит из обращения. На этот случай в таких базах данных обычно предусмотрены системы или операторы, вносящие изменения в данные. Одной из главных проблем подобных баз данных является наличие огромного количества связей между таблицами, что ведет к снижению скорости выполнения запросов при задействовании большого количества таблиц (зачастую в запросах может использоваться до 25 таблиц). В разделе, посвященном подключению к таблице или диапазону, мы поговорим о таблицах подробнее.

Если вернуться к модели данных, построенной нами в главе 2, можно вспомнить, что по центру у нас располагалась таблица фактов (**fRegionSales**), а по бокам от нее – измерения (**dProduct** и **dSalesRep**), как показано на рис. 4.1.

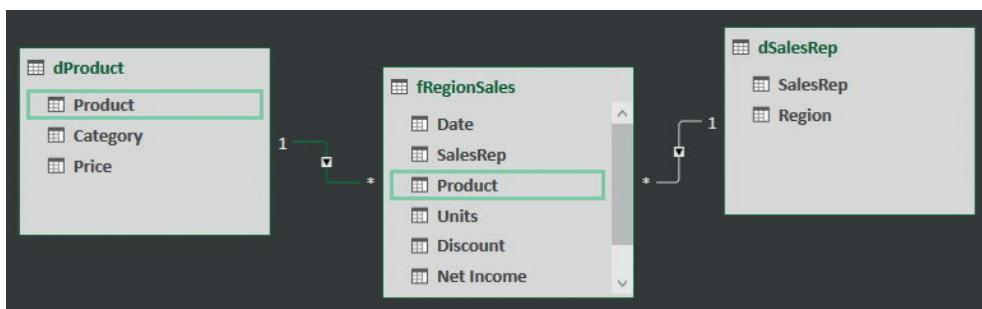


Рис. 4.1. Реляционная база данных

Из этого рисунка мы можем сделать следующие выводы:

- модель данных должна обладать максимально возможной простотой, что положительно скажется на быстродействии запросов. Посередине нашей схемы данных расположилась таблица фактов, в которой собраны все сведения о продажах, и она объединена с двумя таблицами по бокам, именуемыми измерениями. Это одна из самых распространенных схем баз данных, получившая название звезда (star schema). Образно она показана на рис. 4.2;
- таблица фактов в основном будет содержать числовые и *аддитивные* (additive) данные. В нашем примере в ней находится информация о том, сколько единиц товара было продано, с какой скидкой и на какую сумму. Эти значения в основной своей массе являются числовыми и могут быть изменены. Измерения чаще всего содержат описательную информацию. В нашем примере в одном измерении хранятся данные о товарах, их категориях и ценах, а в другом – о продавцах и регионах, в которых они работают. В больших базах данных измерения могут быть также вложенными;

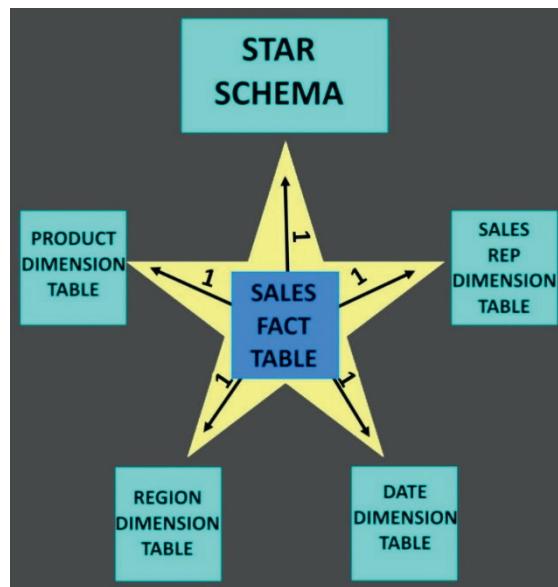


Рис. 4.2. Схема «звезды»

- до этого момента мы применяли *естественные ключи* (natural key) в таблицах, которые было легко и просто использовать в наших примерах с учетом связей между таблицами. Хотя это довольно быстрый способ, он предполагает использование в качестве ключей существующих в таблицах данных. Проблема здесь заключается в том, что когда мы объединяем несколько исходных систем данных, каждая из которых обладает своими бизнес-правилами, мы не сможем диктовать всем хранилищам данных свои условия в отношении реализации. С учетом этих неудобств всегда лучше будет использовать анонимные целочисленные первичные ключи, которые также известны как *суррогатные ключи* (surrogate key). Преимущество их использования заключается в том, что они никак не привязаны в бизнес-логике. Это просто числа, которые должны идти последовательно. В этой нумерации не должно быть никаких параллелей с бизнес-сущностями – смысл ее заключается исключительно в объединении таблиц фактов с измерениями. При этом в таблице фактов этот суррогатный ключ будет храниться как внешний, а поскольку в нем находятся лишь целые числа, места он будет занимать гораздо меньше, а скорость индексации по нему значительно вырастет. А чем выше скорость индексации, тем выше и скорость выполнения запросов. Стоит отметить, что первичные или естественные ключи обычно используются в *средствах оперативной обработки транзакций* (online transactional processing – OLTP), а суррогатные – в *системах оперативной аналитической обработки данных* (online analytical processing – OLAP).

Было время, когда все коммерческие *системы управления базами данных* (database management systems – DBMS) с точки зрения структуры выглядели одинаково, но с тех пор картина изменилась и многие разработали собственные схемы баз данных. Не существует одного оптимального способа создания баз

данных. Но есть вещи, которые могут помочь в повышении производительности баз данных.

Подключение к таблице или диапазону

В Excel инструмент **Get & Transform** (Получить и преобразовать данные) основывается на запросах. Запросы используются для извлечения информации из базы данных. При этом мощь запросов состоит в том, что они способны представить нам ограниченную часть информации из базы данных, которая нам и нужна. На рис. 4.3 показан список товаров на рабочем листе в Excel. Обратите внимание, что в столбце **B** ряд ячеек содержит пустые значения, которые нам необходимо заполнить.

	A	B	C
1		Category	Product
2		Action	Dyno5
3			Board4
4			Game7
5		Construction	Dyno1
6			Game1
7			Game6
8		Creative	Dyno2
9			Board5
10			Game2
11			Game8
12			Game9
13		Education	Dyno3
14			Board1
15			Board3
16			Game3
17			Game4
18			Game10
19		Electronic	Dyno4
20			Board2
21			Game5

Рис. 4.3. Список товаров в Excel

Один из способов сделать это – воспользоваться инструментом **Get & Transform** (Получить и преобразовать данные), находящимся на вкладке **Data** (Данные). Чтобы использовать наш каталог товаров в запросе, можно предварительно преобразовать его в умную таблицу. В этом вам поможет сочетание клавиш **Ctrl+T** или кнопка **Table** (Таблица) на вкладке **Insert** (Вставка). Но делать это необязательно, поскольку инструмент **Get & Transform** (Получить и преобразовать данные) может работать как с таблицей, так и с диапазоном. Посмотрим, как это происходит.

1. Установите курсор внутрь таблицы и выберите пункт **From Table/Range** (Из таблицы/диапазона) на вкладке **Data** (Данные), как показано на рис. 4.4.

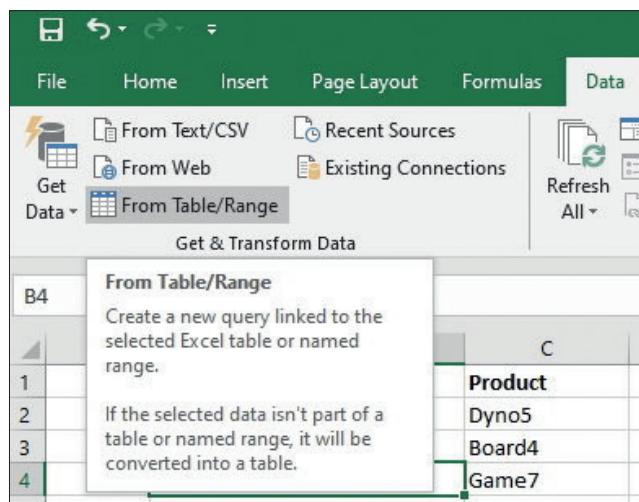


Рис. 4.4. Загрузка данных из таблицы/диапазона

2. При выборе варианта с таблицей/диапазоном Excel автоматически выделит диапазон для преобразования в таблицу. Если вы заранее превратили свой диапазон в умную таблицу, этот шаг можно пропустить. При конвертации диапазона в таблицу Excel обычно выбирает полный диапазон, но если в нем присутствуют пустые ячейки, они могут быть не включены в итоговую таблицу. Так что перед созданием таблицы убедитесь, что выбраны все необходимые данные. На рис. 4.5 показан полностью выбранный диапазон.

	A	B	C	D	E	F	G
1		Category	Product				
2		Action	Dyno5				
3			Board4				
4			Game7				
5		Construction	Dyno1				
6			Game1				
7			Game6				
8		Creative	Dyno2				
9			Board5				
10			Game2				
11			Game8				
12			Game9				
13		Education	Dyno3				
14			Board1				
15			Board3				
16			Game3				
17			Game4				
18			Game10				
19		Electronic	Dyno4				
20			Board2				
21			Game5				

Рис. 4.5. Создание таблицы

3. После нажатия на кнопку **OK** откроется редактор Power Query, и вы увидите свои данные.
4. Нажмите правой кнопкой мыши на заголовок столбца **Category** и выберите пункт **Down** (Вниз) в подменю **Fill** (Заполнить), как показано на рис. 4.6.

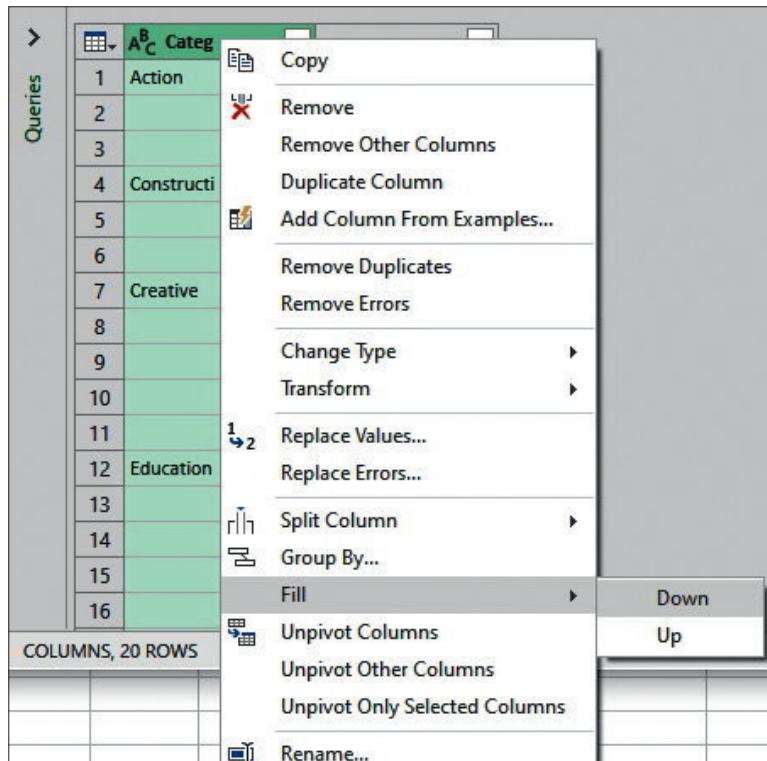


Рис. 4.6. Заполнение столбца значениями вниз

После выполнения этой операции все пустые ячейки в колонке пропадут, что видно по рис. 4.7, и мы сможем использовать ее в дальнейшем.

Подключение из таблицы или диапазона является, пожалуй, самой простой разновидностью соединения с данными, поскольку вся информация уже есть на листе в Excel. А значит, нам нет необходимости загружать их откуда-то еще.

	A	B	C
1	Category	Product	
2	Action	Dyno5	
3	Action	Board4	
4	Action	Game7	
5	Construction	Dyno1	
6	Construction	Game1	
7	Construction	Game6	
8	Creative	Dyno2	
9	Creative	Board5	
10	Creative	Game2	
11	Creative	Game8	
12	Creative	Game9	
13	Education	Dyno3	
14	Education	Board1	
15	Education	Board3	
16	Education	Game3	
17	Education	Game4	
18	Education	Game10	
19	Electronic	Dyno4	
20	Electronic	Board2	
21	Electronic	Game5	

Рис. 4.7. Заполнение столбца значениями вниз завершено

ПОДКЛЮЧЕНИЕ К ВЕБ-СТРАНИЦЕ

Очень здорово, что существует возможность извлекать данные непосредственно с веб-страниц. Кроме того, вас никто не ограничивает одним лишь извлечением данных – далее вы можете очистить их, избавиться от ненужной информации и форматировать по вашему усмотрению перед загрузкой обратно в Excel. Выполнив все эти действия, при очередной необходимости извлечь информацию с веб-страницы вам достаточно будет просто обновить ее, и новые данные будут у вас в руках.

Примечание

При загрузке данных с веб-страницы существует лишь одно ограничение – данные на странице должны быть отформатированы в виде HTML-разметки, а не с использованием JavaScript.

Мы рассмотрим следующие два способа подключения к данным в интернете.

1. Первый способ будет связан с подключением сразу к нескольким таблицам с одного сайта, при этом в каждой из них будет ограниченное количество строк.
2. Второй способ заключается в подключении к файлу CSV, в котором содержится больше сотни тысяч строк.

Будучи типичной южноафриканкой, я неравнодушна к спорту, и особенно к крикету. Давайте взглянем на таблицы с положением команд

в чемпионате с сайта Sky, которые можно найти по адресу: <https://www.skysports.com/cricket/tables>. Они также представлены на рис. 4.8.

The screenshot shows the 'Cricket Tables' section of the Sky Sports website. It features four tables:

- ICC CRICKET WORLD CUP 2019 GROUP**: Shows the top 5 teams (India, Australia, England, New Zealand, Pakistan) with their PLD, W, D, L, and PTS.
- COUNTY CHAMPIONSHIP DIVISION 1 LEAGUE ...**: Shows the top 5 teams (Essex, Somerset, Hampshire, Kent, Yorkshire) with their PLD, W, D, L, and PTS.
- COUNTY CHAMPIONSHIP DIVISION 2 LEAGUE ...**: Shows the top 3 teams (Lancashire, Northamptonshire, Gloucestershire) with their PLD, W, D, L, and PTS.
- ROYAL LONDON ONE-DAY CUP 2019 NORTH G...**: Shows the top 3 teams (Notts Outlaws, Worcestershire Rapids, Lancashire Lightning) with their PLD, W, D, L, and PTS.

Each table includes a 'See Full Table' link and a 'Last Updated' timestamp.

Рис. 4.8. Турнирные таблицы по крикету со Sky

Примечание

Данные в таблицах актуальны на момент написания книги. В день, когда вы ее читаете, информация на сайте будет отличаться.

- Получать данные из интернета достаточно просто. Если вы используете Excel 2016 или выше, перейдите на вкладку **Data** (Данные) и выберите на ленте иконку **From Web** (Из интернета).
В версиях Excel 2013 и ниже для выбора этой иконки вам необходимо сначала перейти на вкладку **POWER QUERY**.
- Вставьте ссылку <https://www.skysports.com/cricket/tables> в диалоговое окно и нажмите на кнопку **OK**, как показано на рис. 4.9.

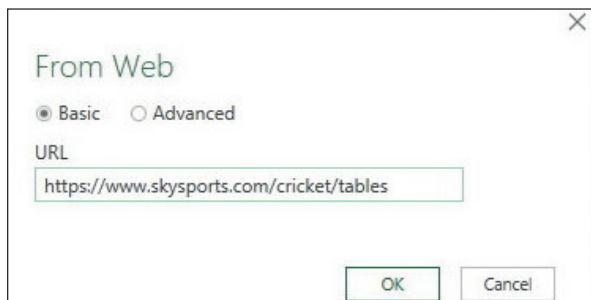


Рис. 4.9. Загрузка данных с веб-страницы

5. Откроется диалоговое окно **Navigator** (Навигатор) с перечислением всех доступных таблиц для подключения. Если бы мы использовали для подключения ссылку <https://www.skysports.com/cricket/tables/3565/icc-cricket-world-cup-2019>, нам была бы предложена единственная таблица **ICC Cricket World Cup**, в которой были бы перечислены все команды, а не только лучшие. По щелчку на таблице **ICC Cricket World Cup 2019** в левой навигационной панели откроется предпросмотр таблицы, как показано на рис. 4.10.

POS	TEAM	PLD	W	D	L	PTS
1	India	9	7	0	2	1
2	Australia	9	7	0	2	2
3	England	9	6	0	3	3
4	New Zealand	9	5	0	3	3
5	Pakistan	9	5	0	3	3

 The 'Load' button is at the bottom left, and 'Transform Data' and 'Cancel' buttons are at the bottom right."/>

Рис. 4.10. Таблица ICC Cricket World Cup

Примечание

Раздел с названием **Document** не содержит в себе таблиц, а представляет собой фрагмент HTML-кода. Во всех остальных разделах присутствуют данные в виде таблиц.

6. На вкладке **Web View** (Веб-представление) можно увидеть предварительный просмотр веб-страницы, к которой вы подключаетесь, как показано на рис. 4.11.

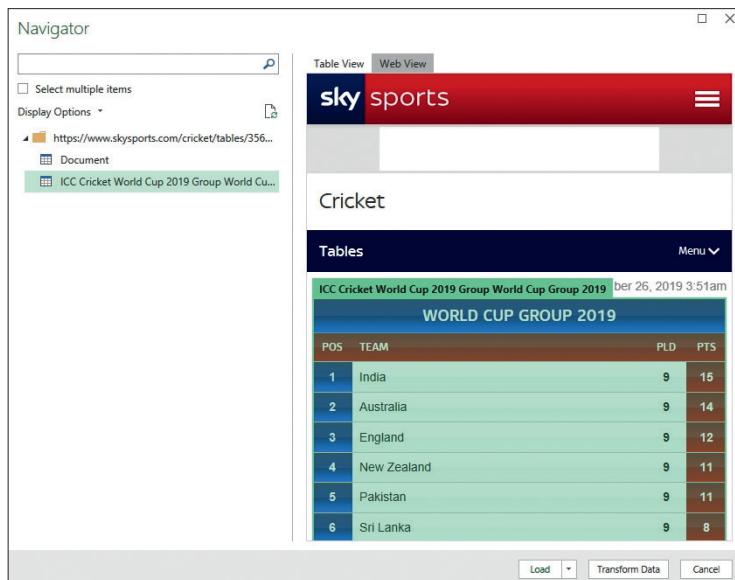


Рис. 4.11. Веб-представление данных

- Если вы хотите перенести более одной таблицы в Excel, можете установить флажок **Select multiple items** (Несколько элементов) и выбрать столько пунктов слева, сколько пожелаете.
- После выбора таблицы **ICC Cricket World Cup 2019** нажмите на кнопку **Transform Data** (Правка). Причина того, почему мы сразу не нажали на кнопку **Load** (Загрузить), состоит в необходимости отредактировать и очистить данные перед их импортированием в Excel или модель данных. В результате этих действий откроется редактор запросов Power Query, как показано на рис. 4.12.

The screenshot shows the 'Power Query Editor' window with the title 'World Cup Group 2019 - Power Query Editor'. The main area displays a table with 10 rows and 7 columns, representing the same data as the table in Figure 4.11. The 'Transform' tab is selected in the ribbon. On the right side, there is a 'Query Settings' pane and an 'APPLIED STEPS' pane. The 'APPLIED STEPS' pane shows a step named 'Changed Type'.

POS	TEAM	PLD	W	D	L	PTS
1	India	9	7	0	1	15
2	Australia	9	7	0	2	14
3	England	9	6	0	3	12
4	New Zealand	9	5	0	3	11
5	Pakistan	9	5	0	3	11
6	Sri Lanka	9	3	0	4	8
7	South Africa	9	3	0	5	7
8	Bangladesh	9	3	0	5	7
9	West Indies	9	2	0	6	5
10	Afghanistan	9	0	0	9	0

Рис. 4.12. Редактор запросов Power Query

9. На данном этапе вы можете преобразовать информацию так, как пожелаете. Все доступные вам действия отображены на рис. 4.12, а это:
- управление запросом;
 - выбор и удаление столбцов и строк;
 - сортировка данных;
 - разделение столбцов;
 - группировка и замена значений;
 - комбинирование таблиц с другими источниками данных;
 - изменение параметров таблицы.
10. После выполнения необходимых действий в таблице перейдите на вкладку **Home** (Главная) и нажмите на кнопку **Close & Load To...** (Закрыть и загрузить в...), как показано на рис. 4.13.



Рис. 4.13. Кнопка загрузки данных в Excel

11. Откроется диалоговое окно **Import Data** (Импорт данных), в котором вы можете выбрать тип загрузки из следующих вариантов: **Table** (Таблица), **PivotTable Report** (Отчет сводной таблицы), **PivotChart** (Сводная диаграмма) и **Only Create Connection** (Только создать подключение). У вас также будет выбор между добавлением таблицы на новый или существующий лист. Мы в нашем примере создадим новый лист, так что окно импорта данных должно выглядеть так, как показано на рис. 4.14.

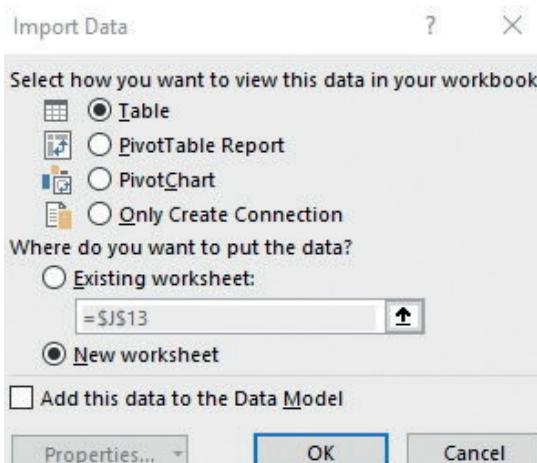


Рис. 4.14. Диалоговое окно импорта данных

Примечание

Если не хотите, чтобы информация дублировалась на рабочем листе и в модели данных, обязательно выбирайте вариант **Only Create Connection** (Только создать подключение).

12. Это делается ради того, чтобы при изменении данных на веб-странице вы могли актуализировать информацию на рабочем листе путем ее обновления. А значит, если вам необходимо формировать один и тот же отчет каждый месяц, достаточно будет нажимать кнопку **Refresh All** (Обновить все) на вкладке **Data** (Данные), как показано на рис. 4.15.

E24	A	B	C	D	E			
1	Position	TEAM	Played	Won	Draw	Lost	Points	
2	1	India	9	7	0	1	15	
3	2	Australia	9	7	0	2	14	

Рис. 4.15. Кнопка обновления данных

13. Если вы используете несколько источников данных, вы можете нажать на кнопку **Queries & Connections** (Запросы и подключения), что приведет к открытию правой панели, на которой будут присутствовать все подключения из разных источников данных с этого листа, как показано на рис. 4.16.
14. Если вы используете Excel версии ниже 2016, нажмите на кнопку **Show Pane** (Показывать области) на вкладке **POWER QUERY**, после чего откроется панель **Workbook Queries** (Запросы книги), как показано на рис. 4.17.

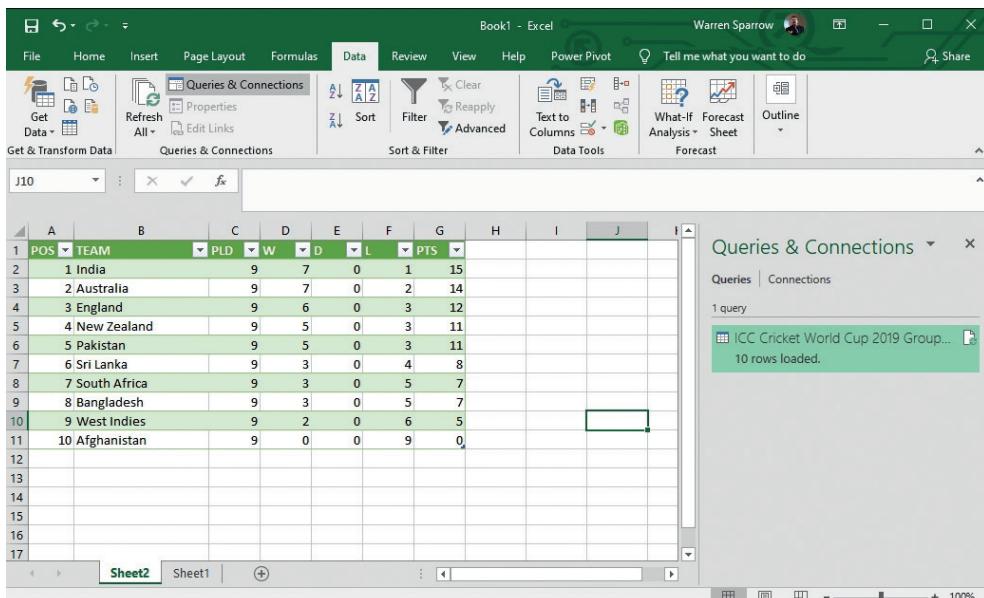


Рис. 4.16. Панель Запросы и подключения

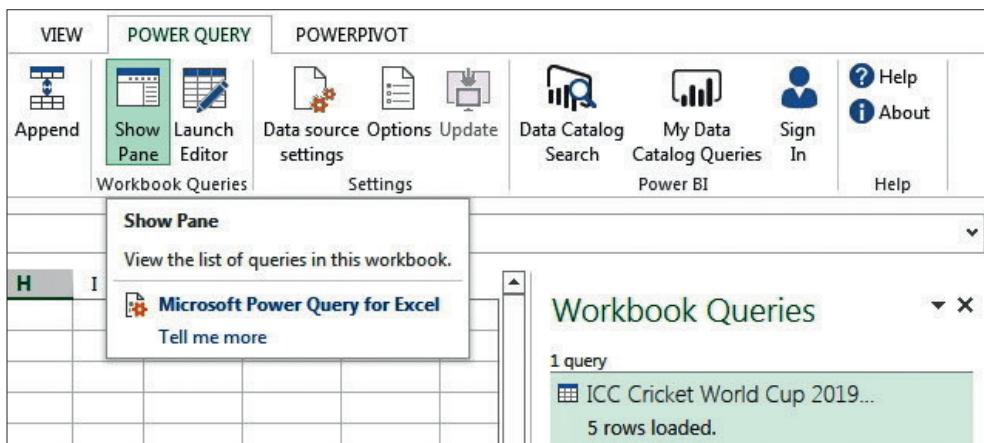


Рис. 4.17. Панель с запросами книги

На панели запросов книги вы увидите подключения примерно в таком же виде, как и в версиях Excel 2016 и выше.

При открытии в более старых версиях Excel книг, созданных в новых версиях, могут возникать ошибки совместимости. Вам будет предложено обновить версию Power Query или Excel, как показано на рис. 4.18. Обычно при обновлении данных ошибка исчезает, но так бывает не всегда. Если ошибка останется, вы будете видеть данные, которые были в книге в момент ее сохранения, и не увидите их обновлений. Если вы выполняли наш пример в версиях Excel ниже 2016, данные из интернета должны обновляться без проблем.

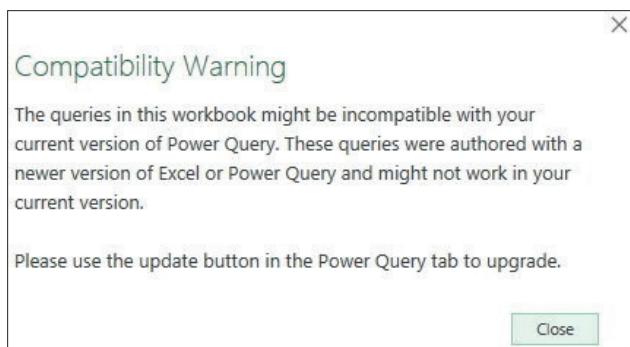


Рис. 4.18. Ошибка совместимости версий

Чтобы обойти проблемы совместимости, вы можете обновить данные из интернета в более ранних версиях Excel. Чтобы это сделать, достаточно нажать на иконку **Refresh** (Обновить) справа от имени запроса или щелкнуть по запросу правой кнопкой мыши и выбрать аналогичный пункт в контекстном меню, как показано на рис. 4.19.

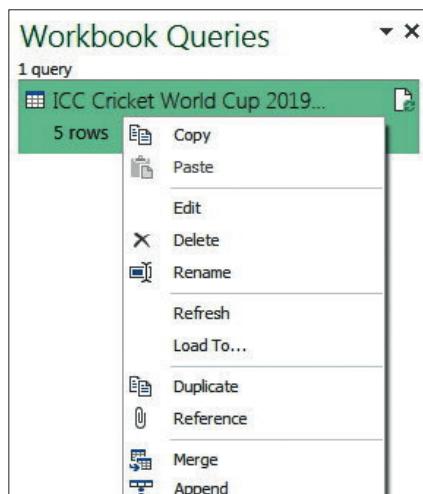


Рис. 4.19. Обновление данных в более ранних версиях Excel

Если открыть этот документ в будущем, когда данные в источнике измениются, Excel предупредит вас об этом, и вы сможете обновить информацию на листе. Так вы сможете располагать наиболее актуальными данными из любых источников в интернете.

ПОДКЛЮЧЕНИЕ К РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ

Для примера извлечения информации из реляционной базы данных будем использовать обращение к базе, поддерживающей язык запросов SQL. И хотя на первый взгляд может показаться, что подключаться к базам данных сложнее,

чем к другим источникам, на самом деле это не так. Самое сложное, что вас ждет, – это правильный ввод пароля при аутентификации.

Существует множество способов подключаться к базам данных SQL, при этом делать это можно как из Excel при помощи инструмента **Get & Transform** (Получить и преобразовать данные), так и из Power BI.

Мы пройдемся по этим вариантам в следующих разделах.

Подключение из Excel

Для подключения к базе данных SQL необходимо в Excel перейти на вкладку **Data** (Данные), нажать на кнопку **Get Data** (Получить данные) и выбрать пункт **From SQL Server Database** (Из базы данных SQL Server) из меню **From Database** (Из базы данных), как показано на рис. 4.20.

Надо сказать, что вы имеете возможность подключаться к огромному количеству баз данных, в числе которых Azure, а также можете извлекать данные из популярных онлайн-служб SharePoint, Exchange и Facebook.

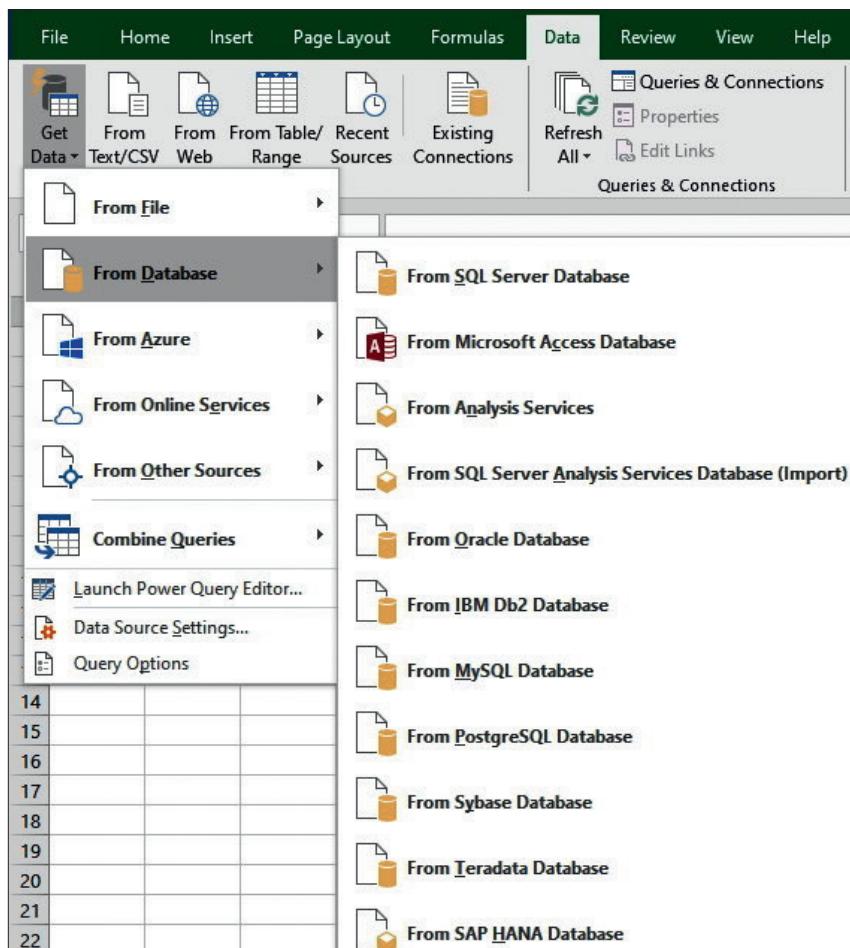


Рис. 4.20. Подключение к базе данных SQL

После выбора пункта **From SQL Server Database** (Из базы данных SQL Server) откроется диалоговое окно **SQL Server database** (База данных SQL Server), показанное на рис. 4.21, в котором вам необходимо ввести параметры подключения к базе данных. Установка и настройка локального экземпляра SQL Server будет подробно описана в главе 10.

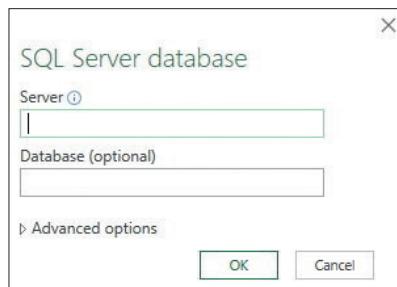


Рис. 4.21. Окно аутентификации SQL Server

Подключившись к нашему экземпляру SQL Server, вы увидите две базы данных **SQLData** и **Test**, в первой из которых среди прочих будут присутствовать таблицы **Orders** и **Customers**, как показано на рис. 4.22.

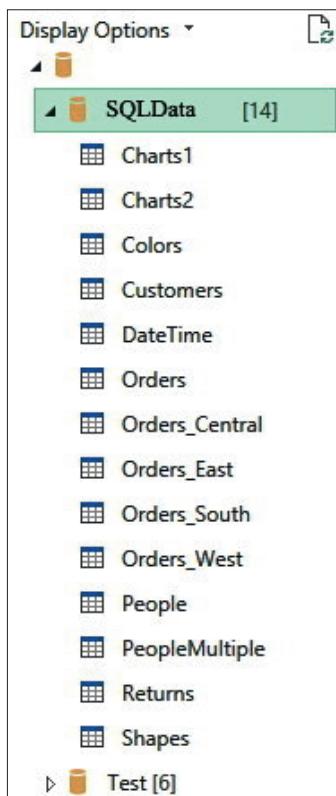


Рис. 4.22. Подключение к базе данных SQLData

Щелкнув по нужной вам таблице, нажмите на кнопку **Transform** (Преобразовать данные), что приведет к открытию редактора запросов Power Query.

Подключение из Power BI

Несмотря на внешнее сходство подключений к базе данных с использованием инструмента **Get & Transform** (Получить и преобразовать данные) в Excel и средств *Power BI*, все же во втором способе есть определенные нюансы, о которых мы расскажем в этом разделе. Для начала выполните следующие шаги.

1. В Power BI нажмите на кнопку **Get Data** (Получить данные) на вкладке **Home** (Главная) и выберите из выпадающего меню пункт **More...** (Другие...).
2. В открывшемся диалоговом окне **Get Data** (Получить данные) выберите в левой навигационной панели пункт **All** (Все), а в правой – **SQL Server database** (База данных SQL Server), после чего нажмите на кнопку **Connect** (Подключить), как показано на рис. 4.23.

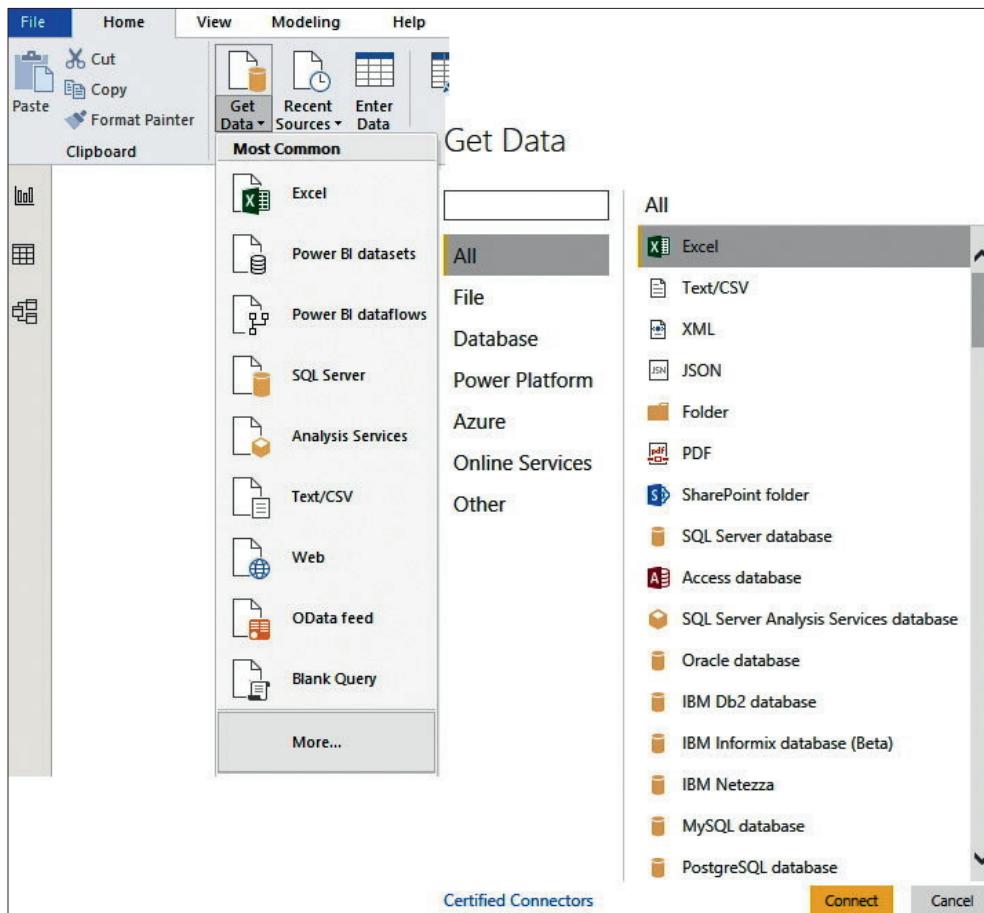


Рис. 4.23. Подключение к базе данных SQL Server из Power BI

Как и в предыдущем разделе, мы будем подключаться к базе данных, при этом Power BI при подключении предлагает дополнительные опции, которых нет в Excel и о которых обязательно нужно иметь представление. При подключении из Power BI вам будет предоставлен выбор между режимами **Import** (импорт) и **DirectQuery**, как показано на рис. 4.24. При выборе режима **DirectQuery** данные не копируются и не импортируются в Power BI, а вместо этого всегда используется наиболее актуальная информация из базы данных. Это означает, что при создании и взаимодействии с визуализациями Power BI будет обращаться за информацией к базе данных. Это приведет к тому, что у вас в наличии всегда будет самая свежая информация, и вы сможете строить визуализации на основе огромных наборов данных, поскольку в режиме **DirectQuery** нет необходимости переносить их целиком в Power BI. Еще одно преимущество использования режима **DirectQuery** состоит в отсутствии ограничения на объем набора данных в 1 Гб.

Однако в этом случае наряду с заботой о наличии быстрого соединения вам придется также думать о том, что каждый раз, когда визуализация будет обновляться, будет запускаться запрос к данным, что в зависимости от серверных мощностей может занимать немало времени. Если речь идет об ожидании в течение 5–30 секунд, это еще приемлемо, но если на выполнение запроса потребуется несколько минут, допустимый тайм-аут истечет, и пользователь получит сообщение об ошибке. Количество одновременно работающих пользователей также влияет на время выполнения запросов, так что этот фактор тоже стоит принимать во внимание.

В режиме **Import** (Импорт), напротив, происходит сохранение копии набора данных в модели, которая используется для формирования и взаимодействия с визуализациями. Обязанности по обновлению данных в этом случае ложатся на Power BI, чтобы вы могли видеть актуальную информацию.

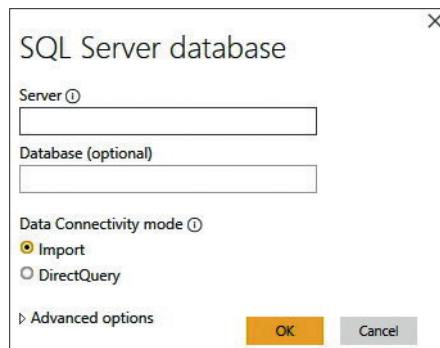


Рис. 4.24. Import или DirectQuery?

Подключившись к серверу, вам необходимо будет пройти процедуру аутентификации, после чего вы сможете выбрать таблицы для загрузки или преобразования. Этот процесс не сильно отличается от процесса подключения к SQL Server из Excel.

Нажав на кнопку **Transform Data** (Преобразовать данные), вы перенесетесь в редактор запросов Power Query, где сможете очистить данные. Нажатие на

кнопку **Load** (Загрузить) приведет к импорту данных в Power BI, как показано на рис. 4.25.



Рис. 4.25. Импорт данных в Power BI

Информация, импортированная в Power BI, отображается в виде отдельных наборов данных в области просмотра данных, располагающейся в правой части окна, как показано на рис. 4.26.

The screenshot shows the Power BI Desktop interface with the 'Visualizations' and 'Fields' panes open on the right side of the workspace. The 'Visualizations' pane displays various chart and report icons. The 'Fields' pane shows a hierarchical list of imported datasets: Charts1, Charts2, Colors, Customers, DateTime, Decimals, Dictionary, and Orders. Under the 'Orders' dataset, specific fields like Category, City, Country, Customer ID, and others are listed. The top navigation bar shows 'File', 'Home', 'Insert', 'Modeling', 'View', 'Help', and 'Table tools'. The 'Table tools' tab is selected, showing options for 'Name' (set to 'Orders'), 'Mark as date table', 'Manage relationships', 'New measure', 'Quick measure', 'New column', and 'New table'. The bottom navigation bar includes 'Page 1' and a '+' button.

Рис. 4.26. Наборы данных в области просмотра

Если нажать на кнопку **Edit Queries** (Преобразование данных), представленную на рис. 4.27, откроется редактор запросов Power Query.

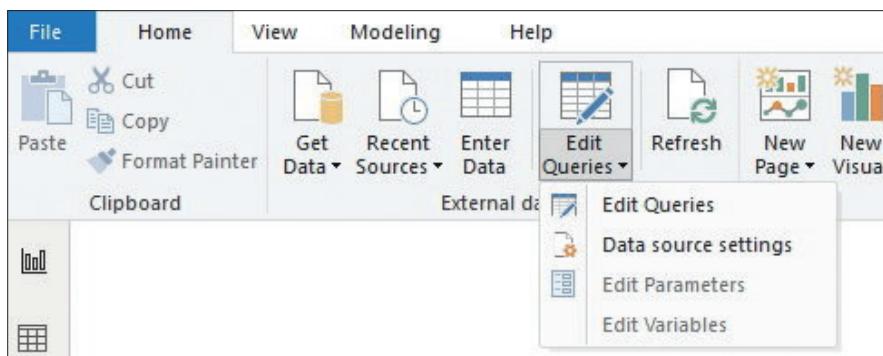


Рис. 4.27. Меню кнопки преобразования данных

Это один из наиболее мощных способов подключения к базе данных SQL, поскольку информация всегда будет актуальной, а работать с ней одновременно смогут сразу несколько пользователей. Разумеется, в этом случае вам может понадобиться дополнительная система безопасности с аутентификацией пользователей и паролями.

ПОЛЬЗОВАТЕЛЬСКИЕ ПОДКЛЮЧЕНИЯ

Существует множество способов подключения к данным в Power BI и Power Query. В наиболее распространенных из них доступ осуществляется посредством кнопки **Get Data** (Получить данные) на вкладке **Home** (Главная), но вы можете *подключаться и к менее очевидным источникам данных*. В этом разделе мы снова подключимся к данным в интернете, но на этот раз нашей целью будет файл CSV, чтобы нам гарантированно потребовалась очистка информации перед ее окончательной загрузкой. Перейдите на вкладку **Data** (Данные) и выберите опцию **From Web** (Из интернета), как мы это делали ранее. Вставьте в поисковую строку адрес <https://github.com/PacktPublishing/Learn-Power-Query/Chapter4-Data/master/SalesData10.csv> и нажмите на кнопку **OK**, как показано на рис. 4.28.

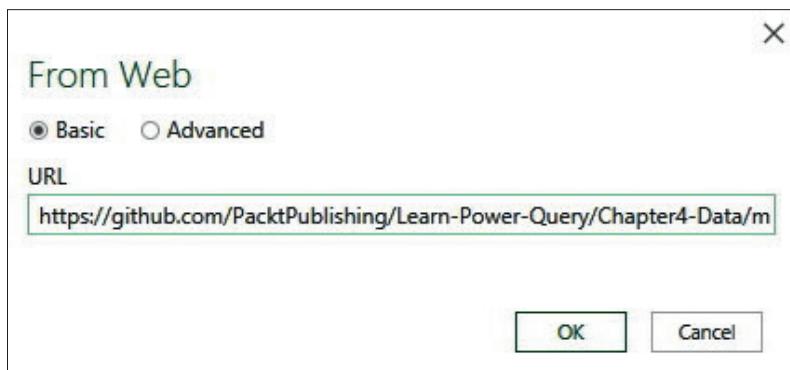


Рис. 4.28. Загрузка данных из интернета

Excel достаточно умен, чтобы догадаться, что вы подключаетесь к файлу с разделителями запятыми. Если вы желаете импортировать данные из файла другого типа или разделителями являются символы, отличные от запятой, вы всегда можете указать альтернативу в выпадающем списке **Delimiter** (Разделитель), как показано на рис. 4.29.

The screenshot shows the 'Get Data' dialog from Microsoft Power Query. At the top, there is a URL: <https://raw.githubusercontent.com/warrensparrow/Chapter4-Data/master/SalesData10.csv>. Below the URL, there are three main sections: 'File Origin', 'Delimiter', and 'Data Type Detection'. The 'File Origin' section shows '65001: Unicode (UTF-8)'. The 'Delimiter' section has a dropdown menu open, with 'Comma' selected. Other options in the dropdown include 'Colon', 'Equals Sign', 'Semicolon', 'Space', 'Tab', '--Custom--', and '--Fixed Width--'. The 'Data Type Detection' section shows 'Based on first 200 rows'. At the bottom of the dialog are three buttons: 'Load', 'Transform Data', and 'Cancel'.

Region	Product	Date	Sales
South	Board2	16/03/2015	10030
West	Game10	02/05/2013	65824
North	Game3	02/01/2014	78488
South	Dyno2	02/10/2011	36987
North	Game8	03/04/2011	28013
North	Dyno4	16/09/2010	1819
North	Board2	17/12/2015	113945.7647
South	Game4	03/04/2013	64663.23978
North	Game10	24/10/2014	93012.55135
West	Dyno5	26/08/2011	35464.07375
South	Board5	16/09/2012	54665.24841
East	Board3	13/08/2011	34044.0828
South	Dyno3	09/03/2010	8672.084433
North	Game1	05/04/2015	100861.4465
East	Dyno4	27/09/2013	72629.19944
West	Board4	13/01/2011	23610.45488
North	Board1	19/04/2014	83492.95682
South	Game6	09/03/2011	26741.61191
North	Dyno4	04/08/2010	15861.37249
South	Game5	18/01/2013	60577.88246
North	Dyno4	31/07/2012	52029.70838

Рис. 4.29. Изменение символа разделителя в файле

Нажмите на кнопку **Transform Data** (Преобразовать данные) и произведите все необходимые действия в открывшемся редакторе запросов Power Query. После этого нажмите на кнопку **Close and Load To...** (Закрыть и загрузить в...) – на этом этапе вы можете создать таблицу данных или загрузить данные в Модель.

	Region	Product	Date	Sales
2	South	Board2	16/03/2015	£ 100,308.76
3	West	Game10	02/05/2013	£ 65,824.37
4	North	Game3	02/01/2014	£ 78,488.78
5	South	Dyno2	02/10/2011	£ 36,987.85
6	North	Game8	03/04/2011	£ 28,013.17
7	North	Dyno4	16/09/2010	£ 18,192.80
8	North	Board2	17/12/2015	£ 113,945.76
9	South	Game4	03/04/2013	£ 64,663.24
10	North	Game10	24/10/2014	£ 93,012.55
11	West	Dyno5	26/08/2011	£ 35,464.07
12	South	Board5	16/09/2012	£ 54,665.25
13	East	Board3	13/08/2011	£ 34,044.08
14	South	Dyno3	09/03/2010	£ 8,672.08
15	North	Game1	05/04/2015	£ 100,861.45
16	East	Dyno4	27/09/2013	£ 72,629.20
17	West	Board4	13/01/2011	£ 23,610.45
18	North	Board1	19/04/2014	£ 83,492.96
19	South	Game6	09/03/2011	£ 26,741.61
20	North	Dyno4	04/08/2010	£ 15,861.37
21	South	Game5	18/01/2013	£ 60,577.88
22	North	Dyno4	31/07/2012	£ 52,029.71
23	North	Dyno4	12/07/2015	£ 106,047.48
24	North	Game3	10/05/2016	£ 121,177.62
25	North	Board4	16/12/2012	£ 58,902.20
26	South	Game10	31/07/2011	£ 34,016.97
27	West	Game7	21/05/2015	£ 103,118.11
28	North	Game9	29/07/2013	£ 70,349.70

Рис. 4.30. Таблица, загруженная из файла CSV

Пользовательские подключения могут быть реализованы в двух видах, которые будут описаны в следующих разделах.

Подключение к рабочей книге

Время от времени появляется необходимость в объединении/слиянии существующей информации, находящейся в одной рабочей книге, в единую таблицу. Препятствием этому служит наличие большого количества пропусков на листе, что делает невозможным создание одной таблицы. На рис. 4.31 показан фраг-

мент исследования, которое проводилось по одному из наших товаров (Game1) на протяжении нескольких дней.

	A	B	C	D	E	F	G
1	Product : Game1						
2							
3			Monday	Overall impression of the product	Quality of the product	Value for money	
4					4	3	3
5					3	3	4
6					4	4	5
7					4	4	
8					4	4	4
9							
10		Average			3.8	3.6	4.0
11							
12							
13			Tuesday	Overall impression of the product	Quality of the product	Value for money	
14					3	4	4
15					3	4	4
16					3	3	4
17					4	3	3
18					3	4	4
19							
20		Average			3.2	3.6	3.8

Рис. 4.31. Исследование товара Game1

В данном исследовании все критерии базируются на шкале от 1 до 5, где 5 – это лучшая оценка, а 1 – худшая. Формализовать такую таблицу традиционными методами было бы довольно затруднительно, но можно сделать это при помощи инструмента **Get & Transform** (Получить и преобразовать данные), что позволит как следует почистить информацию. Посмотрим, как это можно сделать.

1. Создайте новый лист в Excel и перейдите на вкладку **Data** (Данные). После этого проследуйте по пути **Get Data** (Получить данные) → **From File** (Из файла) → **From Workbook** (Из книги), как показано на рис. 4.32.

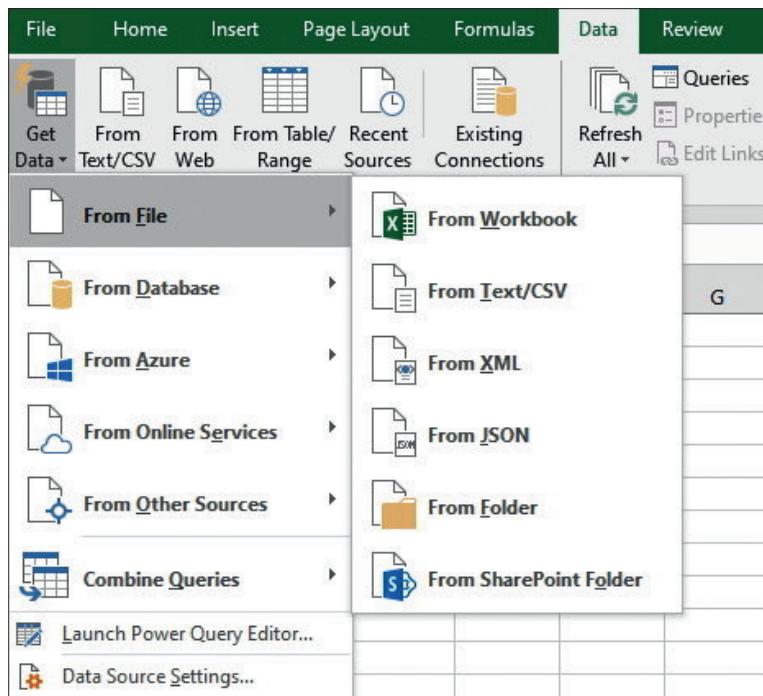


Рис. 4.32. Выбор загрузки из рабочей книги Excel

2. Найдите файл, который вы хотите импортировать, и нажмите на кнопку **Import** (Открыть). После загрузки файла выберите нужный лист и нажмите на кнопку **Transform** (Правка), что приведет к открытию редактора запросов Power Query, как видно на рис. 4.33.

	ABC	Product : Game1	ABC	Column2	ABC	Column3	ABC	Column4	ABC	Column5
1	null		null	Monday		null		null		Overall impression of the product
2	null		null			null				
3	null		null			null				
4	null		null			null				
5	null		null			null				
6	null		null			null				
7	null		null			null				
8	null		null			null				
9	null		null	Average						
10	null		null			null				
11	null		null			null				
12	null		null	Tuesday						Overall impression of the product
										null

Рис. 4.33. Редактор запросов Power Query

3. Теперь можно приступать к очистке данных. Гораздо более подробно о процессе очистки загружаемой информации мы будем говорить в главе 7, посвященной автоматизации отчетов при помощи Power Query. Здесь же мы сосредоточимся на загрузке данных в таблицу или модель данных.
4. Если говорить об очистке данных совсем кратко, то здесь мы можем щелкнуть правой кнопкой мыши на столбце **Column2** и выбрать пункт **Remove** (Удалить), поскольку эта колонка нам не нужна вовсе. То же самое вы можете проделать, если хотите, со столбцами **Column4** и **Product:Game1**.
5. Щелкните по столбцу **Column2** и последовательно выберите пункты **Fill** (Заполнить) → **Down** (Вниз). Вы также можете переименовать столбцы и отфильтровать строки, чтобы избавиться от средних значений и пустых ячеек.
6. Одним из важнейших знаний о Power Query является то, что все ваши действия в редакторе запросов фиксируются справа на панели **APPLIED STEPS** (Примененные шаги). Преимущество такого подхода в том, что если вы сейчас захотите импортировать другой лист Excel, все сохраненные вами шаги будут автоматически применены к новым загруженным данным.

The screenshot shows the Microsoft Power Query Editor interface. The main area displays a table with three columns: 'Overall impression of the game' (containing values like 'Monday', 'Tuesday', etc.), 'Quality of the product' (containing values like '3', '4', '5'), and 'Value for money' (containing values like '3', '4', '5'). The 'Applied Steps' pane on the right lists several steps taken during the transformation process, including 'Filtered Rows1' (which is currently selected).

	Overall impression of the game	Quality of the product	Value for money
1	Monday	4	3
2	Monday	3	3
3	Monday	4	4
4	Monday	4	4
5	Monday	4	4
6	Tuesday	3	4
7	Tuesday	3	4
8	Tuesday	3	4
9	Tuesday	4	3
10	Tuesday	3	4
11	Wednesday	2	2
12	Wednesday	4	3
13	Wednesday	4	4
14	Wednesday	4	4
15	Wednesday	3	4
16	Thursday	3	4
17	Thursday	5	3
18	Thursday	4	3
19	Thursday	3	3
20	Thursday	4	4
21	Friday	4	3
22	Friday	5	4
23	Friday	4	4

Рис. 4.34. Примененные шаги в редакторе запросов Power Query

7. Нажмите на кнопку **Close and Load** (Закрыть и загрузить), чтобы импортировать в Excel отформатированную вами таблицу. Согласитесь, очистить таблицу при помощи инструмента **Get & Transform** (Получить и преобразовать данные) оказалось гораздо проще, чем делать это силами Excel.

	A	B	C	D
1	Column3	Overall impression of the product	Quality of the product	Vlaue for money
2	Monday	4	3	3
3	Monday	3	3	4
4	Monday	4	4	5
5	Monday	4	4	
6	Monday	4	4	4
7	Tuesday	3	4	4
8	Tuesday	3	4	4
9	Tuesday	3	3	4
10	Tuesday	4	3	3
11	Tuesday	3	4	4
12	Wednesday	2	2	2
13	Wednesday	4	4	3
14	Wednesday	4	4	4
15	Wednesday	4	5	4
16	Wednesday	3	4	4
17	Thursday	3	4	4
18	Thursday	5	3	4
19	Thursday	4	4	3
20	Thursday	3	3	3
21	Thursday	4	3	4
22	Friday	4	3	4
23	Friday	5	3	4
24	Friday	4	4	4
25	Friday	4	3	5
26	Friday	4	4	3
27	Saturday	5	4	5
28	Saturday	3	3	4
29	Saturday	4	4	4
30	Saturday	4	3	4

Рис. 4.35. Примененные шаги в редакторе запросов Power Query

Вот и все, мы подключились к рабочей книге.

Подключение к папке

В этом разделе речь пойдет о наиболее простом способе загрузить данные в таблицу или модель данных. Единственное требование заключается в том, что все файлы, которые вы хотите использовать в работе, должны находиться в одной папке. Так что стоит задуматься о том, чтобы создать специальные папки для подобных задач. Это сэкономит вам немало времени и сил в будущем. Также убедитесь, что вы дали каждой папке подходящее и понятное имя, поскольку в результате у вас может быть несколько папок для загрузки данных. На рис. 4.36 показан фрагмент одной из таких папок, в которой я храню файлы CSV, но ничто не мешает мне поместить в нее также текстовые файлы и файлы Excel.

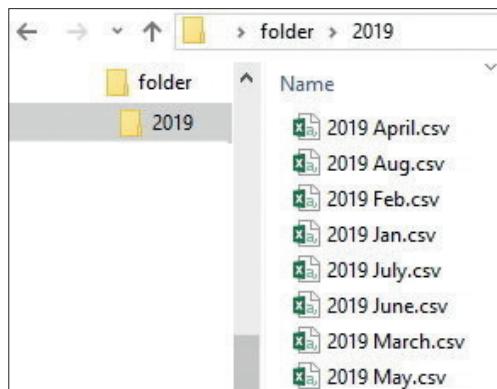


Рис. 4.36. Папка с файлами для загрузки

В каждом из этих файлов хранится информация о продажах в объеме от 10 до 15 тысяч строк. Кроме того, эти файлы содержат четыре столбца, но вы при желании можете осуществлять загрузку и более объемных файлов.

Примечание

Во всех файлах для загрузки должно быть одинаковое количество столбцов с одними и теми же заголовками и в том же порядке. Но если речь идет о загрузке данных из *системы управления взаимодействием с клиентами* (Customer Relationship Management – CRM) или *торговых точек* (point of sale), чаще всего это условие будет соблюдаться по умолчанию.

На рис. 4.37 показан фрагмент одного из файлов.

	A	B	C	D
1	Region	Product	Date	Sales
2	East	Game1	30/06/2019	13490.16
3	West	Game8	30/06/2019	13933.9
4	West	Game10	30/06/2019	13673.88
5	North	Board4	30/06/2019	13864.81
6	South	Prod T	30/06/2019	14153.97
7	North	Dyno2	30/06/2019	14404.17

Рис. 4.37. Файл CSV для загрузки

Нам необходимо выполнить следующие действия.

1. Запустите Excel и перейдите на вкладку **Data** (Данные). Нажмите на кнопку **Get Data** (Получить данные) и в подменю **From File** (Из файла) выберите пункт **From Folder** (Из папки).
2. В версиях Excel ниже 2016 все выглядит немного иначе. Там нам необходимо перейти на вкладку **POWER QUERY** и уже в ней искать кнопку **From File** (Из файла), в которой будет пункт **From Folder** (Из папки), как показано на рис. 4.38.

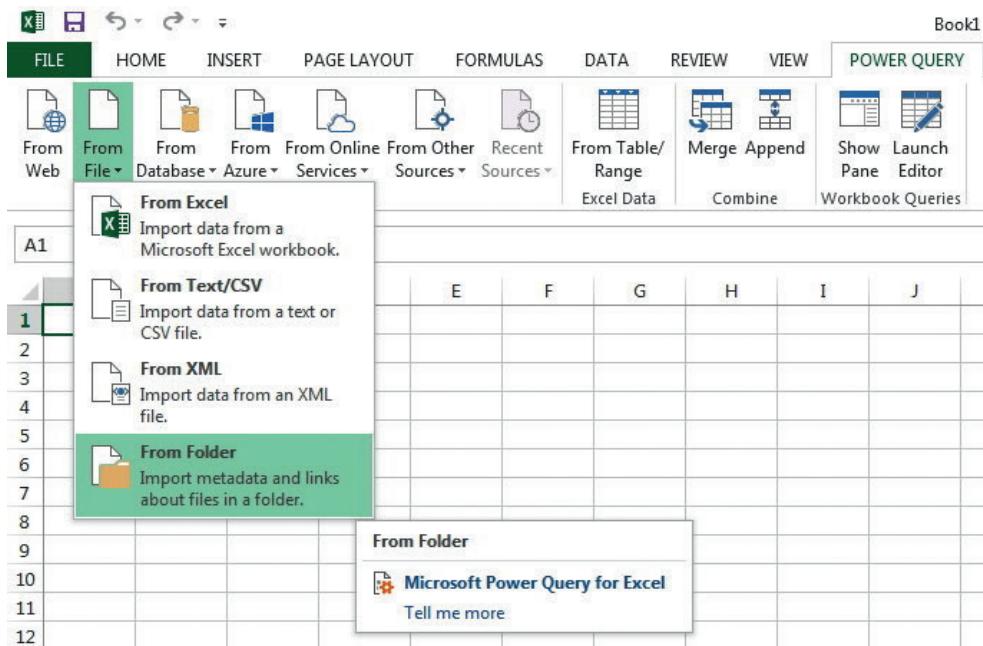


Рис. 4.38. Загрузка из папки в Excel версии ниже 2016

3. Скопируйте и вставьте в открывшееся диалоговое окно путь к нужной папке или нажмите на кнопку **Browse** (Обзор) и найдите папку в проводнике. После этого нажмите на кнопку **OK**.

Вы увидите диалоговое окно с перечислением файлов, находящихся в выбранной папке. Действия, доступные вам в этом окне:

- Combine** (Комбинировать);
- Combine & Edit** (Комбинировать и изменить);
- Combine & Load** (Комбинировать и загрузить);
- Combine & Load To...** (Комбинировать и загрузить в...);
- Load** (Загрузить);
- Load To...** (Загрузить в...)
- Transform Data** (Изменить);
- Cancel** (Отмена).

Я обычно нажимаю на кнопку **Transform Data** (Изменить), что аналогично действию **Combine & Edit** (Комбинировать и изменить).

The screenshot shows the Power Query Editor interface. At the top, there is a table with columns: Content, Name, Extension, Date accessed, Date modified, Date created, Attributes, and Folder Path. Below the table, there are navigation arrows (left, right, first, last). Below the table is a toolbar with buttons: Combine (with a dropdown menu), Load (with a dropdown menu), Transform Data, and Cancel. A context menu is open over the 'Combine' button, listing 'Combine & Edit', 'Combine & Load', and 'Combine & Load To...'. The main area below the toolbar is a grid representing the data from the table.

Рис. 4.39. Меню **Combine & Edit** (Комбинировать и изменить)

- На этом этапе вы можете выбрать и удалить файлы, которые вам не нужны в работе. Мне нужны все файлы без исключения, так что я приступлю к их комбинированию путем нажатия на кнопку с двумя стрелками вниз в столбце **Content**, как показано на рис. 4.40.

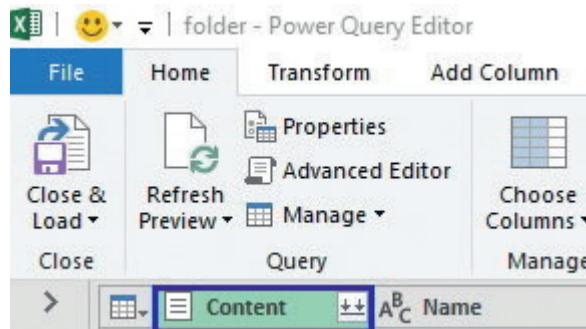


Рис. 4.40. Опции столбца **Content**

- В открывшемся диалоговом окне **Combine Files** (Объединить файлы) появится страница предварительного просмотра первого файла из папки. Поскольку мы имеем дело с файлами с разделителями (CSV), нелишним будет убедиться, что они созданы в правильном формате. Выбор формата, например, поможет Power Query правильно настроить поля даты и времени. Если данные выглядят некорректно, убедитесь, что в выпадающем списке **Delimiter** (Разделитель) выбрано нужное значение. Окно объединения файлов показано на рис. 4.41.

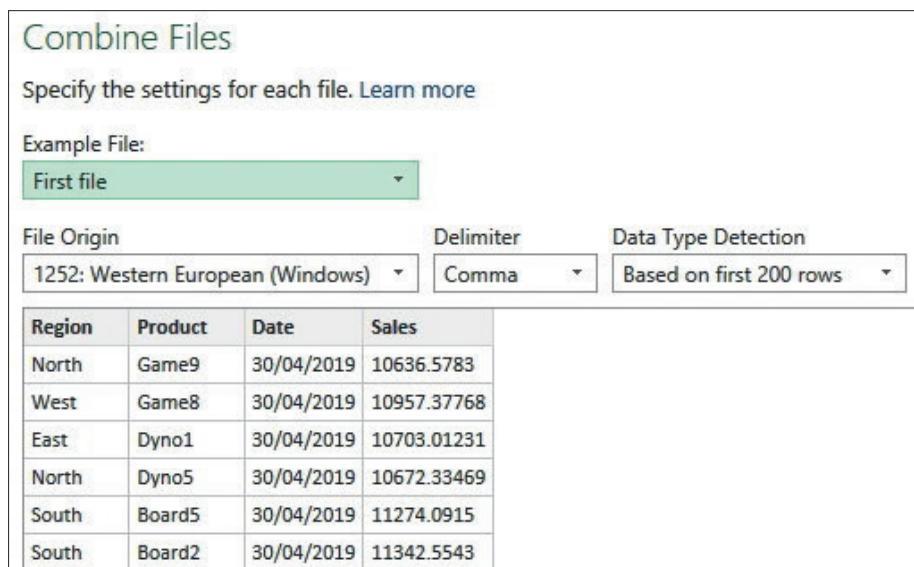


Рис. 4.41. Диалоговое окно объединения файлов

Для комбинирования файлов из папки выполните следующие шаги.

1. Начните с выбора файла, который будет служить образцом. С учетом того, что все наши файлы одинакового формата, это не имеет особого смысла.
2. В случае с файлами CSV разделителями будут запятые, но у вас в папке могут содержаться и текстовые файлы с разделителями в виде пробелов или символов табуляции.
3. Наконец, убедитесь в том, что 200 строк будет достаточно для Power Query для определения типа данных. После завершения операции нажмите на кнопку **OK**.

На рис. 4.42 можно видеть следующее:

- Power Query взял все файлы из папки и объединил их в одну таблицу. В левой части редактора расположена панель с запросами, которые Power Query генерировал, когда вы нажали на кнопку **OK**;
- важным запросом здесь является **Transform Sample file from CSV** (Преобразовать пример файла из CSV), поскольку он является основой для всех остальных файлов и используется в качестве шаблона для существующих и будущих файлов в нашей папке. Если вам захочется изменить внешний вид запроса, вам потребуется обратиться именно к этому пункту;
- последний запрос на левой панели представляет собой итоговый запрос, отвечающий за объединение файлов в одну таблицу. Именно этот запрос вы просматриваете в данный момент. Мы можем преобразовывать данные в этой таблице так же точно, как и в остальных таблицах в редакторе запросов Power Query. Щелкните правой кнопкой мыши на заголовке столбца **Source.Name** и выберите пункт **Remove** (Удалить). Вы также можете изменить тип данных столбца **Sales** на **Currency** (Валюта);

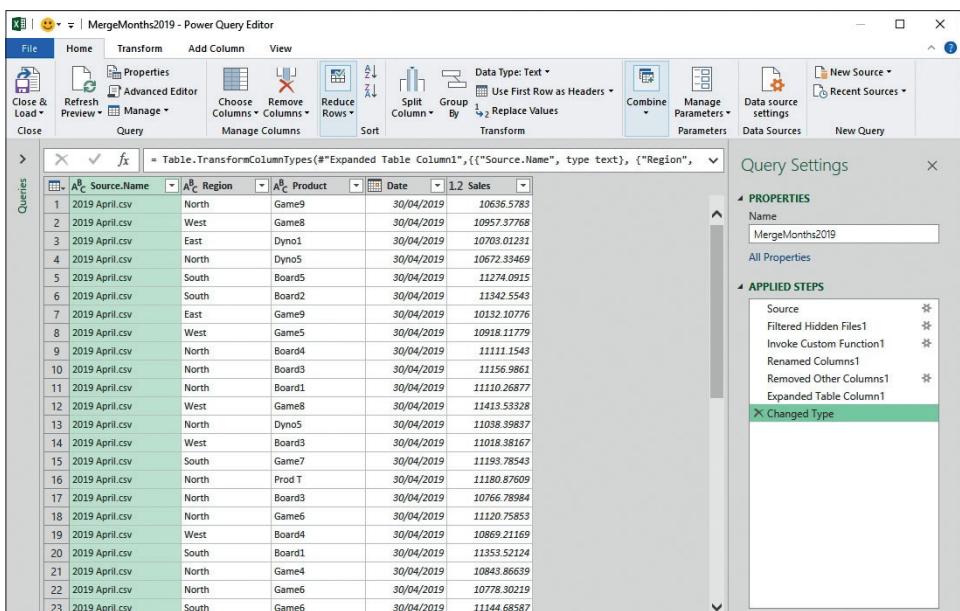


Рис. 4.42. Редактор Power Query

- в правой части редактора располагается панель **Query Settings** (Параметры запроса). Здесь можно дать запросу новое, более подходящее название в поле **Name** (Имя) в разделе **PROPERTIES** (Свойства). После этого вы увидите, что в левой панели наш запрос также изменил название;
- в нижней части правой панели находится область **APPLIED STEPS** (Примененные шаги), представленная на рис. 4.43, в которой вы увидите все шаги, которые выполнили в процессе очистки данных;

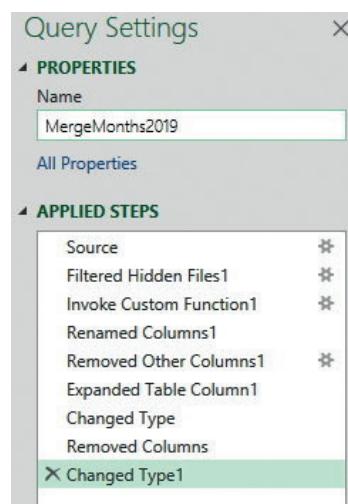


Рис. 4.43. Панель параметров запроса

- после окончания работы с запросами нажмите на кнопку **Close & Load** (Закрыть и загрузить) в верхней левой части редактора. Это приведет к открытию нового рабочего листа по созданному вами запросу. Также вы можете из выпадающего списка выбрать пункт **Close & Load To...** (Закрыть и загрузить в...), что приведет к загрузке запроса в модель данных Power Pivot. Мы остановимся на втором варианте и загрузим данные в модель, как показано на рис. 4.44;

The screenshot shows the Power BI desktop application. On the left is a data grid displaying a table with four columns: Region, Product, Date, and Sales. The table has 30 rows of data. On the right is the 'Queries & Connections' pane, which lists five queries and connections. The 'MergeMonths2019' query is highlighted in green at the bottom, indicating it has been loaded successfully with 110,048 rows.

	A	B	C	D
1	Region	Product	Date	Sales
2	North	Game9	30/04/2019	10636.5783
3	West	Game8	30/04/2019	10957.3777
4	East	Dyno1	30/04/2019	10703.0123
5	North	Dyno5	30/04/2019	10672.3347
6	South	Board5	30/04/2019	11274.0915
7	South	Board2	30/04/2019	11342.5543
8	East	Game9	30/04/2019	10132.1078
9	West	Game5	30/04/2019	10918.1178
10	North	Board4	30/04/2019	11111.1543
11	North	Board3	30/04/2019	11156.9861
12	North	Board1	30/04/2019	11110.2688
13	West	Game8	30/04/2019	11413.5333
14	North	Dyno5	30/04/2019	11038.3984
15	West	Board3	30/04/2019	11018.3817
16	South	Game7	30/04/2019	11193.7854
17	North	Prod T	30/04/2019	11180.8761
18	North	Board3	30/04/2019	10766.7898
19	North	Game6	30/04/2019	11120.7585
20	West	Board4	30/04/2019	10869.2117
21	South	Board1	30/04/2019	11353.5212
22	North	Game4	30/04/2019	10843.8664
23	North	Game6	30/04/2019	10778.3022
24	South	Game6	30/04/2019	11144.6859
25	West	Game5	30/04/2019	11070.3305
26	North	Game3	30/04/2019	11309.3888
27	South	Board3	30/04/2019	11309.8237
28	North	Prod T	30/04/2019	11340.7097
29	West	Game6	30/04/2019	11052.1908
30	South	Dyno5	30/04/2019	11394.4389

Рис. 4.44. Завершенная таблица на основе запроса

- мы загрузили данные о продажах за первые три квартала года, а теперь давайте добавим к нашему анализу оставшийся четвертый квартал;
- перенесите в нашу папку недостающие файлы с последними месяцами года, чтобы она приобрела вид, показанный на рис. 4.45;

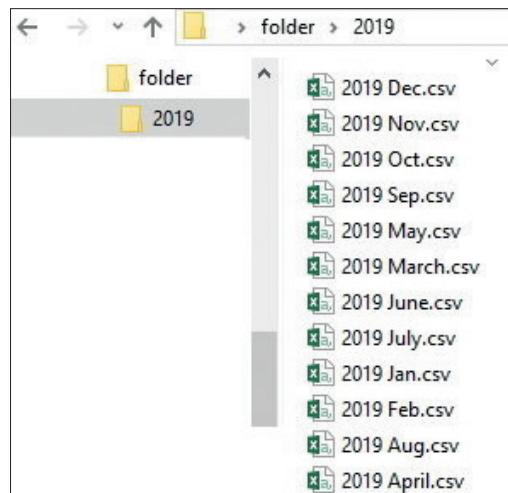


Рис. 4.45. Добавление недостающих месяцев в папку с продажами

- находясь в Excel, щелкните правой кнопкой мыши в любом месте таблицы и выберите пункт **Refresh** (Обновить) из контекстного меню, показанного на рис. 4.46;



Рис. 4.46. Обновление данных запроса

- хотя данные обновляются довольно быстро благодаря наличию шаблона запроса, по сути, происходит полная загрузка всех файлов из папки с выполнением всех примененных шагов из редактора Power Query перед отправкой таблицы в Excel;
- если вы загрузили данные в модель данных Power Pivot, то можете щелкнуть правой кнопкой мыши по запросу **MergeMonths2019** и выбрать пункт **Refresh** (Обновить) в контекстном меню, показанном на рис. 4.47, либо нажать на соответствующую иконку на панели **Queries & Connections** (Запросы и подключения).

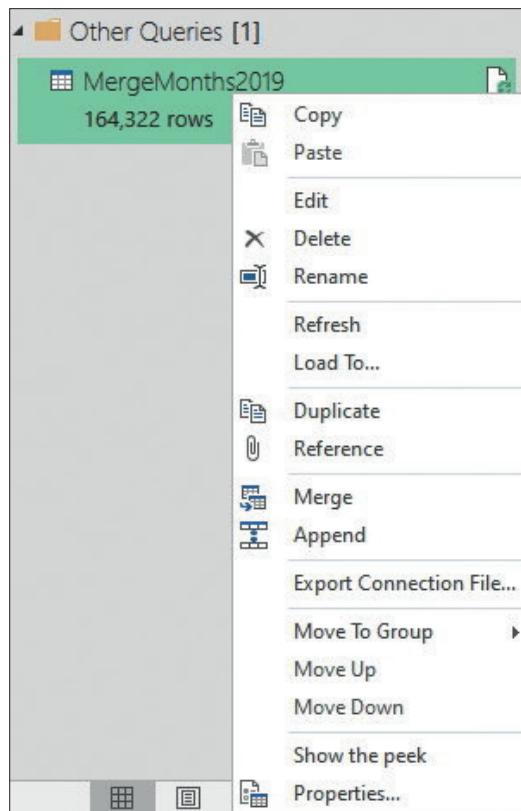


Рис. 4.47. Обновление данных запроса из панели запросов и подключений

Далее мы обсудим различные настройки источников данных.

Настройки источников данных

После подключения к данным вы можете установить настройки для всех существующих подключений, а не только для активного. Например, если ваш источник данных настроен на папку *Sales*, а теперь вам вдруг понадобилось установить подключение на директорию внутри папки *Sales*, вам необходимо изменить настройки подключения таким образом, чтобы Power BI / Power

Query знали где искать информацию. В данном разделе мы посмотрим, где можно изменить эти параметры в Excel.

Изменение настроек в Excel

Практически всегда у вас есть несколько способов выполнить одну и ту же операцию. Здесь я приведу три способа добраться до *изменения настроек подключения*:

- если вы находитесь в Excel и уже загрузили запрос в таблицу, то всегда можете щелкнуть по ней мышью и нажать на кнопку **Edit** (Изменить) на вкладке **Query** (Запрос), показанной на рис. 4.48. После этого откроется привычное окно редактора запросов Power Query;

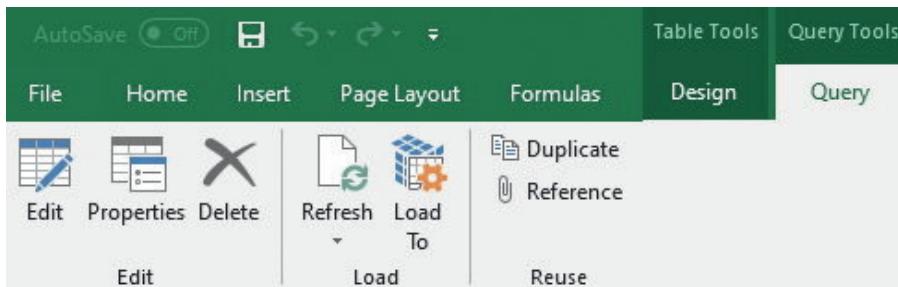


Рис. 4.48. Вкладка **Query** (Запрос)

- также вы можете щелкнуть правой кнопкой мыши на запросе в панели **Queries & Connections** (Запросы и подключения) и выбрать пункт **Edit** (Изменить), что также приведет к открытию редактора запросов Power Query;
- наконец, в самом редакторе Power Query можно нажать на кнопку **Data source settings** (Настройки источника данных) на вкладке **Home** (Главная), показанной на рис. 4.49, после чего будет открыто диалоговое окно **Data source settings** (Настройки источника данных).

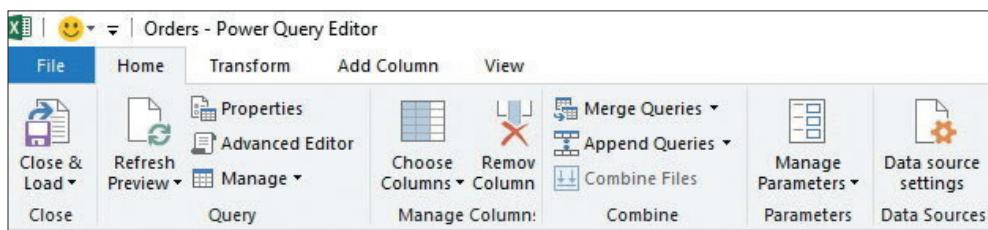


Рис. 4.49. Редактор запросов Power Query

В диалоговом окне **Data source settings** (Настройки источника данных), представленном на рис. 4.50, вы можете изменить источник, а также настроить или очистить права доступа.

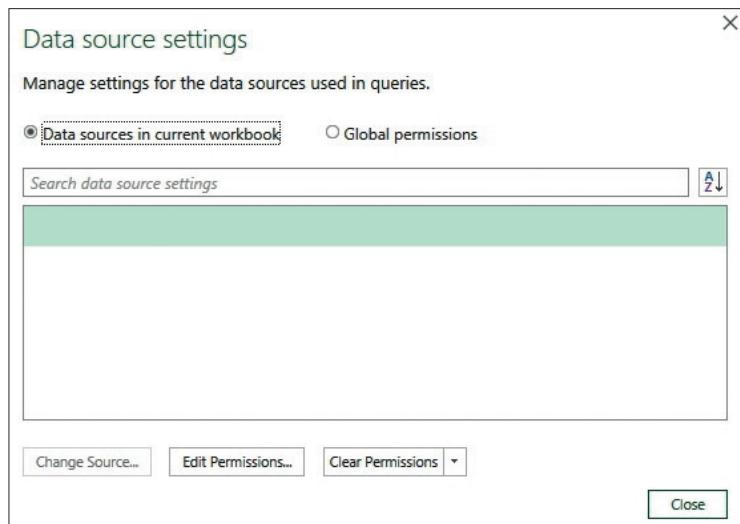


Рис. 4.50. Диалоговое окно настройки источника данных

Вот как это работает:

- при нажатии на кнопку **Change Source...** (Изменить источник...) вы можете поменять исходный источник данных. Для этого вам придется выполнить те же шаги, что и при создании источника;
- кнопка **Edit Permissions...** (Править разрешения...) позволяет настроить аутентификацию, как показано на рис. 4.51.

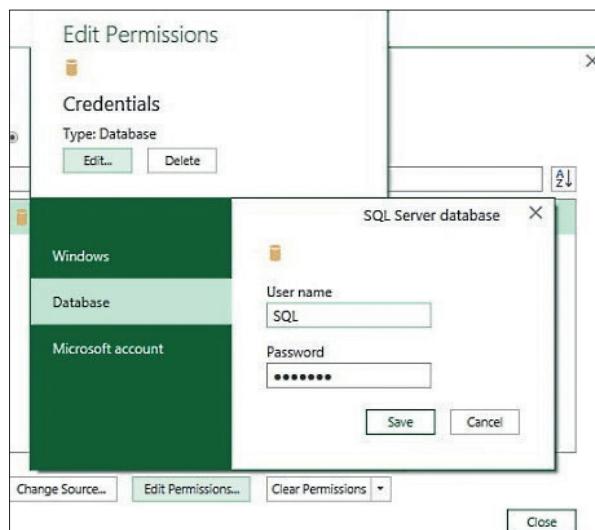


Рис. 4.51. Окно настройки аутентификации для базы данных SQL Server

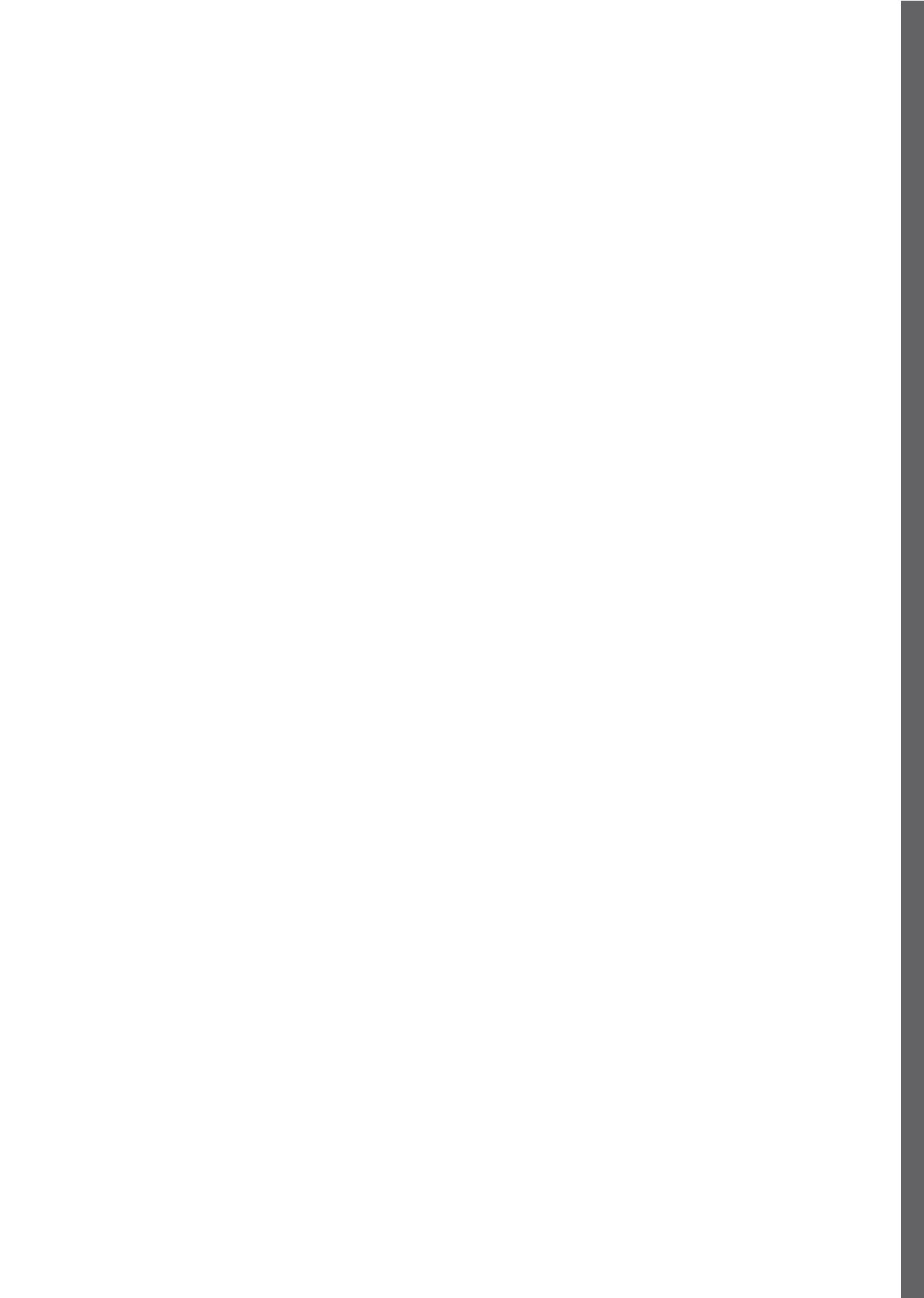
После создания нужных вам подключений обычно не возникает необходимости менять настройки источника данных. Но если вы создали и опублико-

вали отчет в Power BI, после чего вам понадобилось создать еще один отчет с похожей структурой, но с использованием других данных, изменив источник данных, вы сможете сделать это с минимальными затратами времени. Кроме того, иногда может потребоваться повторно подключиться к источнику данных SQL Server или изменить настройки разрешений, что также можно сделать в этом окне.

ЗАКЛЮЧЕНИЕ

Из данной главы вы узнали обо всем разнообразии доступных подключений к данным в Power BI и Excel, включая отдельные файлы, целые папки, базы данных SQL, Azure и онлайн-службы. Одной из наиболее важных особенностей подключений является то, что вы можете соединяться с источниками данных, не представленными в списке, и создавать пользовательские подключения. В процессе подключения к источникам в Power BI создаются наборы данных, которые одновременно представляют и источник, и сами данные. Каждый раз, когда вы подключаетесь к вашим данным, Power BI создает набор данных, который впоследствии используется для создания отчетов и визуализаций.

В следующей части книги вы научитесь преобразовывать данные, загруженные в Power BI и Power Query.



Часть III

Преобразование данных в Power Query

В этой части книги вы научитесь искусству преобразования данных и написания функций с использованием редактора запросов Power Query. Мы рассмотрим большое количество полезных функций, включая IF, Index и Modulo. Вы также научитесь создавать параметры для изменения пути к запросу, поработаете с дашбордами и узнаете, как создавать многомерные и автоматизированные отчеты.

В данной части книги будут представлены следующие главы:

- глава 5 «Преобразование данных посредством Power Query»;
- глава 6 «Продвинутые запросы и функции Power Query»;
- глава 7 «Автоматизация отчетов в Power Query»;
- глава 8 «Создание дашбордов посредством Power Query».

Глава 5

Преобразование данных посредством Power Query

Эта часть книги будет посвящена разнообразным трансформациям данных при помощи Power Query. Прочитав ее, вы должны в полной мере понять, почему так важно выполнять тщательную предварительную подготовку сырых данных перед их анализом. Вы научитесь применять к наборам данных операции *сведения* (pivot) и *отмены свертки* (unpivot) с целью их структурирования в табличном формате. Кроме того, вы приобретете навыки *разделения* (split), *объединения* (merge), *дублирования* (duplicate) и *извлечения* (extract) информации из столбцов, а также сможете использовать условные столбцы, значения в которых отображаются в соответствии с условием *if...then...else*, и применять настройку автоматического фонового обновления. В данной части мы научимся извлекать возраст из столбцов с датами, что обычно требует дополнительных усилий и применения этой операции к каждому столбцу отдельно. В дополнение мы также рассмотрим разные варианты использования разделителей и группировку данных из разных строк в одно значение.

В этой главе мы остановимся на следующих темах:

- манипулирование данными при помощи операций сведения (pivot) и отмены свертки (unpivot);
- основные инструменты для работы со столбцами и строками;
- использование инструментов объединения и добавления запросов;
- группировка данных;
- работа с инструментами извлечения данных.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

С учетом пройденного материала мы будем исходить из предположения о том, что вы обладаете достаточными навыками работы в Excel и умеете сортировать и фильтровать данные в таблицах. Также вы должны уверенно работать с инструментом **Get & Transform** (Получить и преобразовать данные), опцией **Close & Load** (Закрыть и загрузить) и панелью **Queries & Connections** (Запросы и подключения).

Вам понадобится интернет-соединение для загрузки соответствующих файлов с GitHub. Файлы с кодом для этой главы можно скачать по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

Видеофрагмент, посвященный данной главе, располагается по адресу: https://www.youtube.com/watch?v=n1swG3Z44mM&list=PLelCvrlLe1860_GJEzs47WaZXwZjwTN83&index=6&t=515s.

Манипулирование данными при помощи операций сведения и отмены свертки

Преобразование данных подразумевает под собой переименование таблиц и столбцов и приданье данным вида, подходящего для анализа. Читая эту главу, вы научитесь применять инструменты сведения и отмены свертки данных для преобразования их в удобный табличный формат. После применения этих инструментов мы удалим все ненужные столбцы, а также переименуем колонки и запросы. Мы обратим внимание на то, какие шаги будут созданы в процессе выполнения операций, и узнаем, как обновлять источники данных в Power Query. Все это необходимо для предварительной подготовки данных с целью дальнейшего анализа или создания отчетов.

Наборы данных могут быть представлены по-разному. Одни из них можно сделать визуально привлекательными, например отчеты по финансовому планированию с финансовыми годами в строках и месяцами в заголовках столбцов. Другие являются собой более сложные структуры данных, которые используются для анализа при помощи сводных таблиц и/или хранения информации в приложениях, специально для этого предназначенных, вроде Power BI. Всегда необходимо ясно представлять итоговую цель набора данных, с которым вы работаете, и тщательно подготавливать данные перед выполнением вычислений и анализа. И Power Query идеально подходит для выполнения подобных преобразований.

Если не структурировать данные как следует, это может привести к серьезным проблемам при их анализе – мы просто потеряем уйму драгоценного времени на создание различных вычислений. В хорошо структурированном наборе данных этим заниматься не придется.

Подготавливая информацию к анализу при помощи, например, сводных таблиц, необходимо пройти долгий путь от сырых неподготовленных данных к удобным представлениям. Вот краткий план, которому можно следовать:

- убедитесь, что на листе отсутствуют дублирующиеся данные;
- удалите все фильтры, примененные к данным, и снимите установленные группировки в разделе **Outline** (Структура) в Excel;
- форматируйте колонки в Excel должным образом. Это означает, что для столбцов с датами должен быть указан нужный формат, а для числовых столбцов – формат валюта или финансовый, когда это необходимо. Позаботьтесь о том, чтобы заголовки колонок были точными, понятными и лаконичными;
- никогда не включайте в исходные данные расчетные величины, а также агрегированные значения вроде итогов, подытогов и средних значе-

ний. Причина этого в том, что после преобразования и очистки данных в Power Query они поступят в аналитические инструменты Power Pivot, Power BI или Excel, где будут обработаны так, как нам нужно;

- избегайте наличия пустых ячеек в исходных данных;
- последнее, что вам необходимо сделать, – это отформатировать исходные данные в виде таблицы. Причины того, почему это так важно, были подробно описаны в главе 4. Выполнить форматирование в виде таблицы можно до использования инструмента **Get & Transform** (Получить и преобразовать данные) или прямо на этом этапе, чтобы гарантировать, что любые добавленные строки и столбцы будут автоматически включены в исходные данные в момент обновления из разных источников.

Давайте для начала рассмотрим пример плохо структурированного набора данных и подумаем, как можно привести его в порядок при помощи инструмента *отмены свертки* (Unpivot).

На рис. 5.1 представлен тот самый плохо структурированный набор данных, абсолютно не годящийся для анализа средствами Power Pivot или других аналитических инструментов. Эти исходные данные ни в коей мере не соответствуют приведенным выше постулатам.

1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	QUARTER ONE						QUARTER 2						QUARTER 3					
2	Season	Winery	Label	Region	Cost Per Case	JAN SALES	FEB SALES	MAR SALES	Q1 SALES	APR SALES	MAY SALES	JUNE SALES	Q2 SALES	JULY SALES	AUG SALES	SEPT SALES	Q3 SALES	TOTAL SALES
3	Winter	Matts Winery	Cab Savon	North	£165.00	450	526	926	£313 830.00	779	1144	929	£470 580.00	305	1148	333	£294 690.00	£1 079 100.00
4	Winter	Matts Winery	Cab Savon	North	£165.00	550	1038	409	£329 505.00	416	724	971	£348 315.00	530	565	327	£234 630.00	£912 450.00
5	Autumn	Matts Winery	Cab Savon	North	£165.00	575	1025	331	£318 615.00	790	364	1198	£388 080.00	983	692	861	£418 440.00	£1 125 135.00
6	Spring	Matts Winery	Cab Savon	North	£165.00	650	523	723	£312 840.00	774	394	949	£348 975.00	1105	530	576	£364 815.00	£1 026 630.00
7	Summer	Matts Winery	Cab Savon	South	£165.00	320	800	306	£235 290.00	360	363	778	£247 665.00	391	368	1043	£297 330.00	£780 285.00
8	Winter	Matts Winery	Cab Savon	South	£165.00	325	938	558	£300 465.00	723	351	572	£271 590.00	747	1196	1069	£496 980.00	£1 099 035.00
9	Autumn	Matts Winery	Cab Savon	South	£165.00	330	420	648	£230 670.00	851	342	395	£262 020.00	451	403	519	£226 545.00	£719 235.00
10	Spring	Matts Winery	Cab Savon	South	£165.00	350	804	1015	£357 885.00	680	534	529	£287 595.00	792	978	737	£413 655.00	£1 002 540.00
11	Summer	Matts Winery	Cab Savon	East	£165.00	350	768	593	£281 985.00	743	827	508	£342 870.00	768	651	870	£377 685.00	£1 002 540.00
12	Winter	Matts Winery	Cab Savon	East	£165.00	360	1083	851	£378 510.00	449	888	355	£279 180.00	562	562	892	£332 640.00	£990 330.00
13	Autumn	Matts Winery	Cab Savon	East	£165.00	370	780	1164	£381 810.00	532	373	938	£304 095.00	964	1039	864	£473 055.00	£1 158 960.00
14	Spring	Matts Winery	Cab Savon	East	£165.00	375	860	412	£271 752.00	897	830	521	£370 920.00	764	534	969	£374 055.00	£1 016 730.00
15	Summer	Matts Winery	Cab Savon	West	£165.00	230	1080	489	£296 835.00	669	732	426	£301 455.00	760	1018	959	£451 605.00	£1 049 895.00
16	Winter	Matts Winery	Cab Savon	West	£165.00	235	1068	1083	£393 690.00	618	1082	727	£400 455.00	879	787	686	£388 080.00	£1 182 225.00

Рис. 5.1. Плохо структурированные исходные данные

В этом наборе данных содержатся расчетные значения для кварталов и итоговых показателей. От объединенных заголовков столбцов также желательно избавиться, а столбец **Region** можно преобразовать из строк в столбцы в Power Query.

После выполнения минимального количества необходимых шагов по преобразованию данных исходная информация должна принять гораздо более благоприятный вид для анализа, но для полного удовлетворения нужно сделать последний штрих – воспользоваться инструментом *отмены свертки* (Unpivot) в Power Query. Для демонстрации работы этого механизма мы будем использовать столбцы с продажами с января по сентябрь. Вместо отдельных столбцов с цифрами продаж по трем кварталам мы получим два столбца со строками. Начнем.

1. Откройте набор данных *YearlyProductSales.xlsx* в Power Query. Это можно сделать посредством Excel или Power BI.
2. Выделите все девять столбцов с цифрами продаж. Затем на вкладке **Transform** (Преобразование) в группе **Any Column** (Любой столбец) нажмите на кнопку **Unpivot Columns** (Отменить свертывание столбцов), как показано на рис. 5.2.

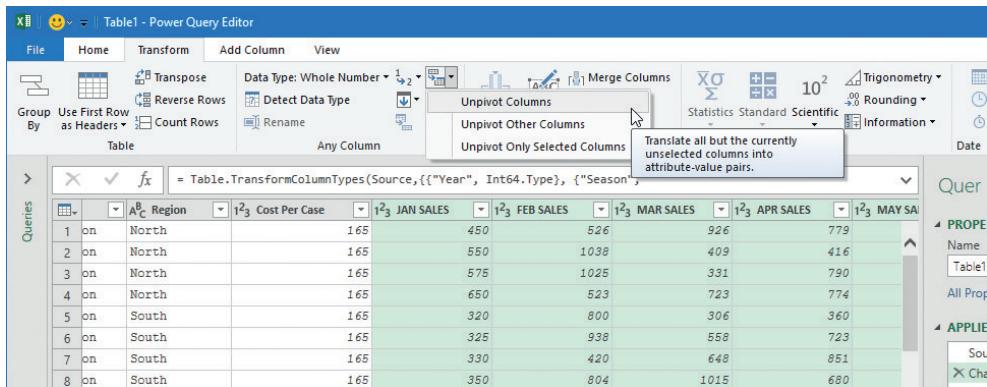


Рис. 5.2. Использование инструмента отмены свертывания столбцов

3. В результате будут созданы столбцы **Attribute** (Атрибут) и **Value** (Значение), в которых разместятся месяцы и соответствующие им значения продаж, как видно на рис. 5.3.

1 ² ₃ Cost Per Case	A ^B _C Attribute	1 ² ₃ Value
165	JAN SALES	450
165	FEB SALES	526
165	MAR SALES	926
165	APR SALES	779

Рис. 5.3. Столбцы с атрибутами и значениями после отмены свертывания

4. Переименуйте созданные столбцы в соответствии с их смыслом. Просто дважды щелкните мышью по заголовку столбца **Attribute** (Атрибут) и измените его. То же самое проделайте с полем **Value** (Значение).
5. Если вам необходимо отменить последнее действие или вы хотите узнать, было ли оно применено к данным в таблице, обратитесь к панели **APPLIED STEPS** (Примененные шаги) в правой части экрана, чтобы просмотреть, удалить, переименовать или изменить порядок следования шагов, примененных к данным. Для удаления шага щелкните на красном крестике слева от названия шага или воспользуйтесь контекстным меню.
6. Теперь давайте выполним *операцию сведения* (Pivot), являющуюся полной противоположностью операции отмены свертывания. В этом случае значения из строк станут названиями новых колонок. Убедитесь, что для этого примера вы используете источник данных *SafestSolutionsLaw.xlsx*.
7. Выделите столбец **CASE TYPE**, чтобы использовать его значения в строках в качестве новых столбцов. После этого перейдите на вкладку **Transform** (Преобразование) и нажмите на кнопку **Pivot Column** (Столбец сведения), как показано на рис. 5.4.

The screenshot shows the Power BI Data Editor ribbon with the 'Transform' tab selected. In the 'Pivot Column' section of the ribbon, the 'Pivot Column' button is highlighted with a mouse cursor. Below the ribbon, a query editor window is open, showing a table with four rows and three columns: 'CASE #', 'CASE TYPE', and 'BALANCE OWED'. The 'CASE TYPE' column has values 'real estate', 'collections', 'corporate', and 'disability'.

Рис. 5.4. Операция сведения

В открывшемся диалоговом окне **Pivot Column** (Столбец сведения) в выпадающем списке **Values Column** (Столбец значений) выберите значение **BALANCE OWED**, как показано на рис. 5.5.

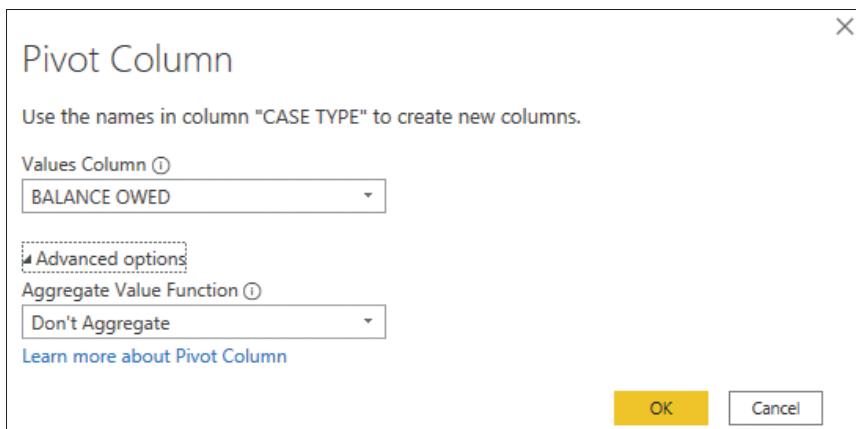


Рис. 5.5. Диалоговое окно сведения столбца

Раскройте раздел **Advanced options** (Расширенные параметры) и выберите в выпадающем списке **Aggregate Value Function** (Функция агрегированного значения) пункт **Don't Aggregate** (Не агрегировать).

Нажмите на кнопку **OK**, чтобы увидеть результат сведения столбца, показанный на рис. 5.6.

The screenshot shows the Power BI Data Editor with a transformed table. The original columns were 'CASE #', 'CASE TYPE', and 'BALANCE OWED'. After applying the 'Pivot Column' operation with 'BALANCE OWED' as the value column and 'Don't Aggregate' as the function, a new table was created with six columns: 'CASE #', 'real estate', 'collections', 'corporate', 'disability', and 'BALANCE OWED'. The 'BALANCE OWED' column now contains the values 12500, 1000, 13000, and 12000 respectively for the four rows.

Рис. 5.6. Результат сведения столбца

После выполнения этой операции в левой панели **Queries** (Запросы) будут созданы два запроса. Переименуйте первый из них в **UnPivot**, а второй в **Pivot**, как показано на рис. 5.7.

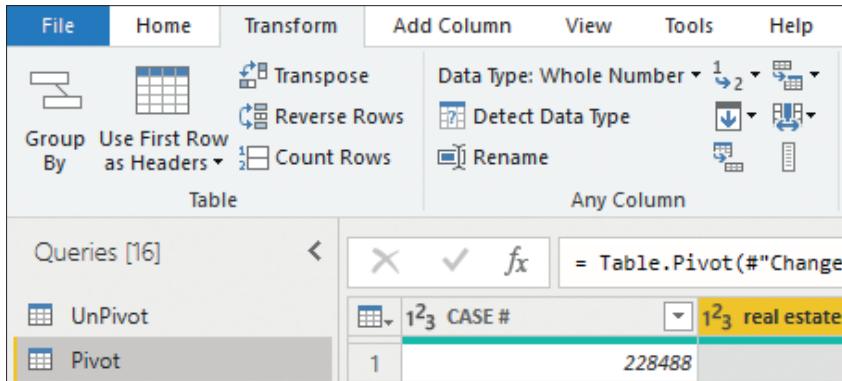


Рис. 5.7. Переименованные запросы в левой панели

В этом разделе вы научились транспонировать таблицы с применением операций сведения и отмены свертывания. В следующем разделе мы рассмотрим разные варианты обновления данных и узнаем, как в целом работает процесс обновления.

Обновление данных

После создания запросов данные, лежащие в их основе, не остаются неизменными. Например, в бухгалтерию могут поступить новые финансовые сведения, которые требуется добавить к существующему набору данных, или данные в интернете, которые мы брали за основу, могут измениться, или же пользователь вручную внес дополнительную информацию в систему. Хорошая новость состоит в том, что вам необходимо лишь раз создать запрос, после чего вы можете много-кратно использовать его, просто обновляя подключение к источнику данных. При выполнении расчетов в Excel зависимые ячейки на листе пересчитываются автоматически при изменении исходных данных. С Power Query дело обстоит иначе – этому инструменту требуется команда на обновление. Это очень важно понимать, поскольку от этого зависят точность и актуальность ваших данных.

При обновлении информации вам нет необходимости помнить все шаги, которые вы применяли к данным, когда последний раз с ними работали. Как вы уже знаете, Power Query бережно сохраняет все выполненные шаги в разделе **APPLIED STEPS** (Примененные шаги) в виде кода на языке M. Так что обновление для Power Query представляет собой простую задачу, состоящую в загрузке или импорте новых данных и последующем выполнении всех шагов.

Если у нас есть подключения к рабочей книге Excel, то для обновления соединений достаточно перейти на вкладку **Data** (Данные) и нажать на кнопку **Refresh All** (Обновить все) в группе **Queries & Connections** (Запросы и подключения).

Чтобы обновить все запросы непосредственно из Power Query, перейдите на вкладку **Home** (Главная) и на выпадающей кнопке **Refresh Preview** (Обновить

предварительный просмотр) выберите пункт **Refresh All** (Обновить все). Можно нажать на кнопку **Refresh Preview** (Обновить предварительный просмотр), чтобы обновить только текущий запрос. Над строкой формул появится область оповещений с напоминаниями о необходимости обновить запросы при работе в Power Query, как показано на рис. 5.8.

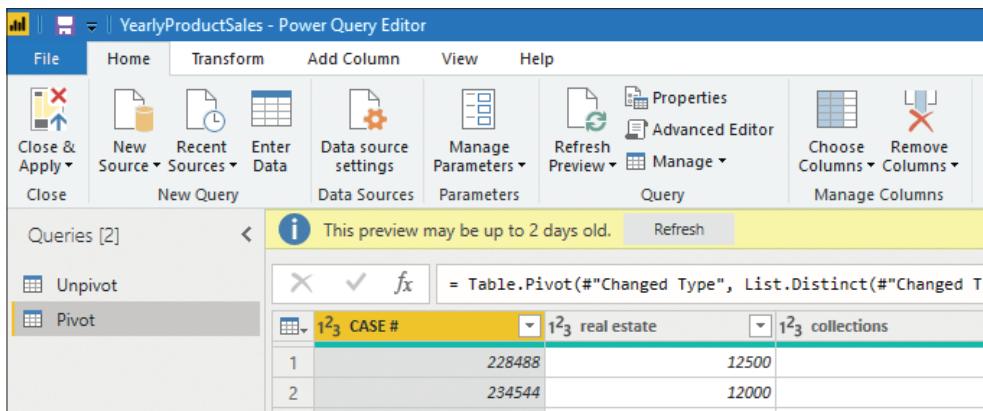


Рис. 5.8. Область оповещений в Power Query

Помните о том, что обновление всех запросов может занять немало времени, особенно если в них используются наборы данных большого размера. Также стоит не забывать, что при обновлении всех запросов эта операция будет выполнена в единственно правильной очередности, тогда как при ручном обновлении запросов вам пришлось бы соблюдать этот порядок самостоятельно.

Если вашим источником данных является внешний файл, в Power Query загружается только его последняя сохраненная версия. Даже если этот файл открыт у вас на компьютере и вы недавно его редактировали, вам необходимо будет сохранить его, чтобы включить последние правки в обновление. В то же время если источник данных и запросы находятся в одной и той же книге, все изменения будут включаться в обновления автоматически – даже без их сохранения. Будьте внимательны при обновлении запросов, базирующихся на других запросах, поскольку в этом случае могут обновиться не все запросы.

Существует также возможность установки опции автоматического обновления в процессе работы. У этой особенности есть свои плюсы и минусы. С одной стороны, всегда приятно, когда данные обновляются в фоновом режиме без вашего ведома, с другой – в процессе работы вы можете изменить какие-то формулы или обновить сводные таблицы, и произведенные изменения могут не включаться в обновления, если фоновый процесс на этот момент не был завершен. Следуйте приведенной ниже инструкции при настройке обновлений.

1. Для установки режима автоматического обновления убедитесь, что Excel открыт и у вас есть активный запрос или подключение. На вкладке **Data** (Данные) нажмите кнопку **Refresh All** (Обновить все) и выберите пункт **Connection Properties...** (Свойства подключения).

2. В открывшемся диалоговом окне **Query Properties** (Свойства запроса) обратите внимание на раздел **Refresh control** (Обновление экрана) на вкладке **Usage** (Использование).
3. Убедитесь, что флажок **Enable background refresh** (Фоновое обновление) установлен. Ниже выберите требуемую частоту обновлений в минутах, как показано на рис. 5.9.

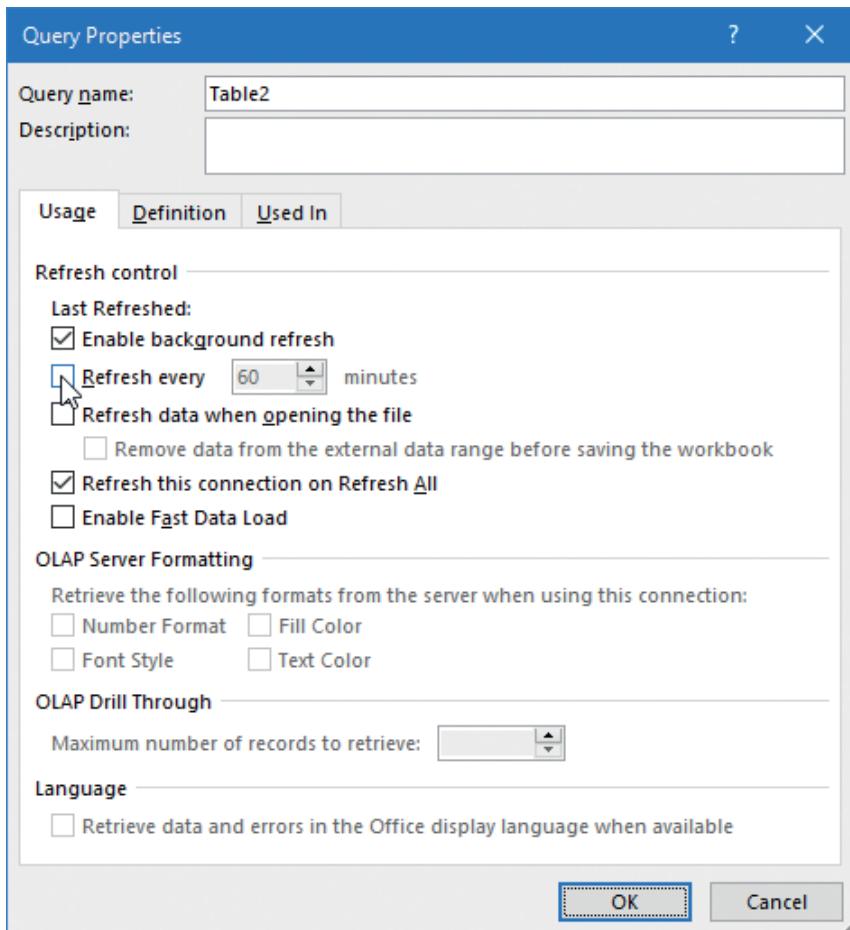


Рис. 5.9. Опции обновления в Excel

4. Также обратите внимание, что у вас есть возможность обновлять данные при открытии файла и включать или не включать запрос в число обновляемых по кнопке **Refresh All** (Обновить все).
5. Нажмите на кнопку **OK**, чтобы подтвердить изменения и вернуться в рабочую книгу Excel.

В данном разделе мы научились обновлять данные и познакомились с опциями диалогового окна **Query Properties** (Свойства запроса). В следующей главе мы подробно остановимся на инструментах работы со столбцами и строками.

Основные инструменты работы со строками и столбцами

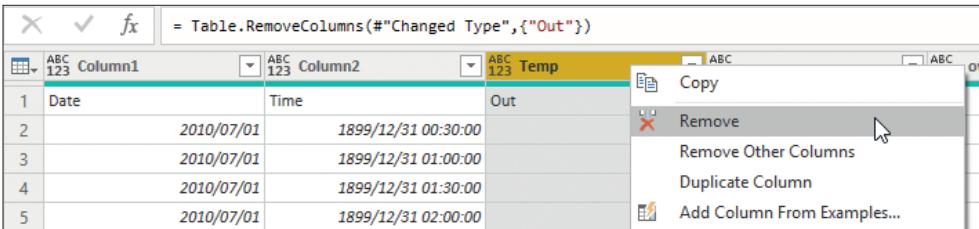
В данном разделе вы познакомитесь с полезными инструментами для работы со столбцами, научитесь удалять колонки, которые вам больше не нужны, и разделять столбцы таким образом, чтобы информация из одного столбца показывалась в разных. Также вы узнаете, как работать с операциями извлечения данных, объединения и индексирования, и научитесь создавать условные столбцы.

Прочитав этот раздел, вы сможете преобразовывать таблицы таким образом, чтобы они были полностью готовы для анализа в таких инструментах, как Power BI и Excel Power Pivot. Процесс преобразования данных включает в себя удаление ненужных колонок, избавление от верхних и нижних строк, в которых часто содержится лишняя информация, использование столбца с индексом с целью проведения анализа, создание столбцов на основании условий, применение фильтров к столбцам с операторами И/ИЛИ, использование фильтров с одним и множеством критериев, удаление дублирующихся и пустых строк, работу со строкой заголовков в запросах и разбивку значений в столбцах в соответствии с символами-разделителями. Все это поможет привести данные в вид, пригодный для проведения полноценного анализа.

Удаление столбцов

Удалять лишние столбцы в Power Query крайне просто, и этот процесс сильно напоминает выполнение аналогичной операции в Excel. Если колонка не придает запросу аналитической ценности, просто избавьтесь от нее, чтобы она не добавляла веса загрузке данных при обновлении и работе с отчетами. Следующие методы работают в Power Query при обращении из Excel или Power BI.

1. Откройте рабочую книгу WeatherData.xlsx.
2. Удалять столбцы можно двумя способами. Первый состоит в удалении выбранных столбцов, а второй – в удалении всех столбцов, за исключением выбранных. Для выполнения этих операций можно использовать кнопки на ленте или воспользоваться контекстным меню. Для удаления выбранных столбцов подойдет пункт меню **Remove Columns** (Удалить столбцы), а для удаления всех столбцов, кроме выбранных, – пункт **Remove Other Columns** (Удалить другие столбцы).
3. Выделите в таблице одну или несколько колонок для удаления. В нашем примере это будет столбец **Out**.
4. Нажмите правой кнопкой на заголовке столбца и выберите пункт **Remove** (Удалить), как показано на рис. 5.10.



The screenshot shows the Power Query Editor interface. A table is displayed with three columns: 'Column1', 'Column2', and 'Temp'. The 'Temp' column is highlighted with a yellow background. A context menu is open over the 'Out' cell in the 'Temp' column, listing options: 'Copy', 'Remove', 'Remove Other Columns', 'Duplicate Column', and 'Add Column From Examples...'. The 'Remove' option is selected, indicated by a grey background and a cursor pointing at it.

	Date	Time	Out
1	2010/07/01	1899/12/31 00:30:00	
2	2010/07/01	1899/12/31 01:00:00	
3	2010/07/01	1899/12/31 01:30:00	
4	2010/07/01	1899/12/31 02:00:00	
5	2010/07/01	1899/12/31 02:00:00	

Рис. 5.10. Удаление столбца

5. Колонка исчезнет из вывода запроса. Не забудьте сохранить запрос для обновления.

В следующем разделе мы научимся избавляться от ненужных строк в верхней и нижней частях запроса.

Удаление верхних или нижних строк

Одной из самых распространенных операций при очистке данных является **удаление верхних и/или нижних строк в таблице**, не несущих важной для анализа информации. Давайте продолжим работать с нашим запросом WeatherData из предыдущего примера. На вкладке **Home** (Главная) существует целый раздел для работы со строками в запросах, скрывающийся под выпадающей кнопкой **Remove Rows** (Сократить строки). Выберите из подменю **Remove Rows** пункт **Remove Top Rows** (Удаление верхних строк), как показано на рис. 5.11. В открывшемся диалоговом окне укажите количество строк для удаления из верхней части запроса. Затем нажмите на кнопку **OK** для подтверждения действия и сохранения запроса.

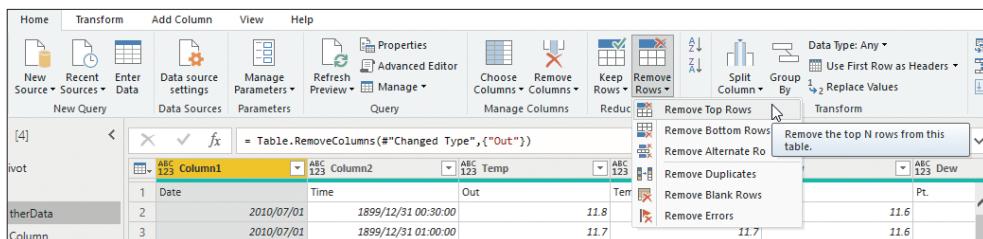


Рис. 5.11. Удаление верхних строк в запросе

Вы также имеете возможность удалять строки по определенному шаблону, воспользовавшись кнопкой **Remove Alternate Rows** (Удалить чередующиеся строки). В этом случае вы можете указать первую строку, количество строк для удаления и число строк, которые необходимо сохранить. Power Query выполнит удаление строк согласно указанному вами шаблону.

Использование столбца с индексом

Можно создать столбец с индексом, который будет выступать в качестве счетчика строк или временной колонки и помогать в проведении анализа. В этом столбце будет содержаться набор чисел, идентифицирующих строки. В данном разделе мы будем использовать запрос **SSGProductsAll** для создания колонки с индексом.

1. Выделите в соответствующем списке запрос **SSGProductsAll**, который был создан в результате наших действий по сведению и отмене свертывания столбцов.
2. Откройте вкладку **Add Column** (Добавить столбец) и в выпадающей кнопке **Index Column** (Столбец индекса) выберите пункт **Custom...** (Насстроить...) для создания столбца с индексом с указанным приращением, как показано на рис. 5.12.

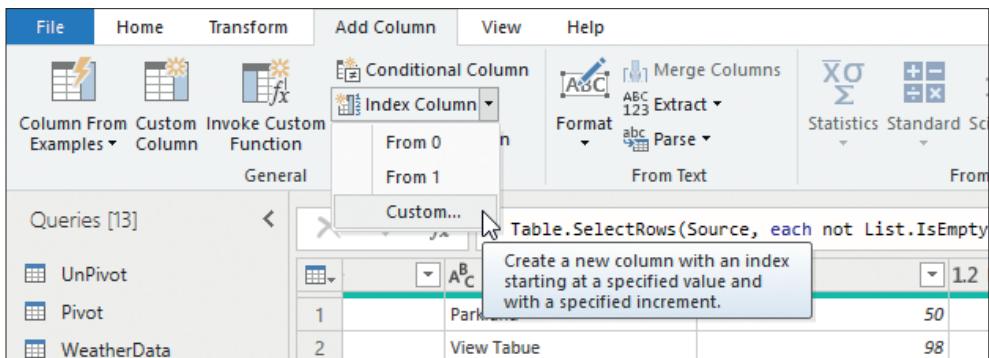


Рис. 5.12. Создание столбца с индексом

3. В открывшемся диалоговом окне **Index column** (Добавить столбец индекса) вы можете указать начальное значение индекса и его приращение (инкремент). На рис. 5.13 показано созданное поле индекса с начальным значением 1 и таким же приращением.

Queries [16]	1	2	3	4	5	6	7	8	9	10	11	12	Index
UnPivot													1
Pivot													2
WeatherData													3
SplitColumn													4
STUDENTS													5
SSGProductsEast													6
SSGProductsWest													7
SSGProductsAll													8
StudentList													9
YEARLY FEES													10
Merge1													11

Рис. 5.13. Столбец индекса, созданный в конце таблицы

4. После создания столбца с индексом он появится в правой части таблицы. Нам же обычно нужно, чтобы эта колонка располагалась слева. Чтобы переместить ее на привычное место, перейдите на вкладку **Transform** (Преобразование) и в выпадающей кнопке **Move** (Переместить) выберите пункт **To Beginning** (В начало).
5. Сохраните запрос, чтобы он обновился.

В этом разделе вы узнали, как создавать колонку с индексом, служащую в качестве счетчика строк в таблице. Далее мы посмотрим, как можно создать так называемый условный столбец, данные в котором будут зависеть от созданного нами условия при помощи конструкции *if / then / else*.

Создание условного столбца при помощи конструкции if...then...else

Условный столбец в Power Query заполняется данными автоматически на основании заданных критериев. Это бывает полезным, когда значения в столбце должны

вычисляться с использованием конструкции `if...then...else` и могут быть дополнены сложными условиями `else...if`. Мы пройдем по всем стадиям создания условного столбца с целью отображения групп сбыта (sales team) по каждому региону, соответствующему продавцам из колонки A в исходных данных. Приступим.

1. Импортируйте файл SalesTeams.xlsx в Power Query с помощью Excel или Power BI.

1	Sales Team West
2	Arthur Payne
3	David Header
4	Brian Donkin
5	James Bartha
6	Carol Harrison
7	Sales Team North
8	Robert Wagner
9	Hillary Hunker
10	Kimberly Hepburn
11	George Houderson
12	Sales Team East
13	Ariel Saywers
14	Monica Essous
15	Sales Team South
16	Gerald Lafasne
17	Angie D'Abrose
18	Andrew Sales
19	Katherine Henderson
20	Gregory Duncan

Рис. 5.14. Структура данных в исходной рабочей книге SalesTeams.xlsx

2. Переименуйте запрос в **SalesTeams**.
3. Теперь нам необходимо создать условный столбец с перечислением соответствующих групп сбыта для каждого продавца в первом столбце. В настоящий момент в этом столбце стоит наименование региона группы сбыта, после чего следуют продавцы из этого региона. И так по всей колонке. Здесь все зависит от того, что именно вы хотите сделать с данными. В нашем случае мы хотели бы, чтобы в одном столбце указывался регион, а во втором – список продавцов из этого региона. Продолжаем двигаться к этой цели.
4. Убедитесь, что вы выбрали запрос **SalesTeams**. После этого перейдите на вкладку **Add Column** (Добавить столбец) и нажмите на ленте кнопку **Add Conditional Column** (Условный столбец).
5. В появившемся диалоговом окне введите в поле **New column name** (Имя нового столбца) значение **Sales Teams**. Далее нам необходимо определить условную конструкцию `if...then...else`. Логика должна быть следующей: если значение в колонке **Column1** начинается со строки **Sales**, выводить в новом столбце значение из колонки **Column1**, иначе выводить **Null**, как

показано на рис. 5.15. На этом шаге будет создан новый столбец, в значения которого будут попадать наименования групп сбыта, а индивидуальные продавцы будут заменены на Null. Нажмите на кнопку **OK** для подтверждения своих действий.

Рис. 5.15. Создание условного столбца

6. Результат будет выглядеть так, как показано на рис. 5.16.

	ABC Column1	Sales Teams
1	Sales Team West	Sales Team West
2	Arthur Payne	null
3	David Header	null
4	Brian Donkin	null
5	James Bartha	null
6	Carol Harrison	null
7	Sales Team North	Sales Team North
8	Robert Wagner	null
9	Hillary Hunker	null
10	Kimberly Hepburn	null
11	George Houderson	null
12	Sales Team East	Sales Team East
13	Ariel Saywers	null
14	Monica Essous	null
15	Sales Team South	Sales Team South
16	Gerald Lafasne	null
17	Angie D'Abrose	null
18	Andrew Sales	null
19	Katherine Henderson	null
20	Gregory Duncan	null

Рис. 5.16. Результат вычисления условного столбца

7. Далее нам нужно заполнить значения в колонке, чтобы избавиться от пустых ячеек, и сопоставить столбцы с соответствующими продавцами. Щелкните правой кнопкой мыши на заголовке столбца **Sales Teams** и в подменю **Fill** (Заполнить) выберите пункт **Down** (Вниз).
8. Осталось отфильтровать строки в столбце **Column1**, чтобы в нем остались только продавцы, а группы сбыта, начинающиеся со слова Sales, не показывались.
9. Щелкните на иконке фильтра справа от названия колонки **Column1**. Появится окно с опциями фильтрации. Выберите в подменю **Text Filters** (Текстовые фильтры) пункт **Does Not Begin With...** (Не начинается с...).
10. В разделе **Keep Rows where 'Column1' does not begin with** (Показать строки, где Column1 не начинается с) введите слово **Sales**.
11. Нажмите на кнопку **OK**, чтобы подтвердить выбор и увидеть результат, показанный на рис. 5.17.

Queries [13]

	Column1	Sales Teams
1	Arthur Payne	Sales Team West
2	David Header	Sales Team West
3	Brian Donkin	Sales Team West
4	James Bartha	Sales Team West
5	Carol Harrinson	Sales Team West
6	Robert Wagner	Sales Team North
7	Hillary Hunker	Sales Team North
8	Kimberly Hepburn	Sales Team North
9	George Houderson	Sales Team North
10	Ariel Saywers	Sales Team East
11	Monica Essous	Sales Team East
12	Gerald Lafasne	Sales Team South
13	Angie D'Arose	Sales Team South
14	Andrew Sales	Sales Team South
15	Katherine Henderson	Sales Team South
16	Gregory Duncan	Sales Team South

Рис. 5.17. Результат создания условного столбца

12. Переименуйте столбец **Column1** в **Salesperson**, если необходимо.

Теперь вы умеете создавать условные столбцы с использованием конструкции **if...then...else** на основании нужных вам критериев. В следующем разделе мы поговорим об использовании фильтрации с использованием условий **And/Or** (И/ИЛИ).

Фильтрация данных с использованием условий And/Or (И/ИЛИ)

Вы прекрасно знаете, как работают инструменты фильтрации и сортировки данных в Excel. В Power Query все очень похоже. В данном разделе мы обратимся

к двум типам фильтрации: *И* (And) и *ИЛИ* (Or). Power Query загружает в списке фильтрации первые 1000 строк за один раз, о чем вы будете предупреждены специальным сообщением. Используйте кнопку **Load more** (Загрузить еще) для открытия следующей тысячи записей и т. д. Для создания подобного фильтра необходимо выполнить следующие действия.

1. Импортируйте источник данных SSGFilter.xlsx в Power Query.
2. В заголовке каждого столбца запроса имеется активная иконка фильтра. По нажатии на нее открывается диалоговое окно фильтрации. Все это очень похоже на Excel.
3. Щелкните по иконке фильтра справа от наименования колонки **Division**. В появившемся диалоговом окне выберите в меню **Text Filters** (Текстовые фильтры) пункт **Equals** (Равно) и в открывшемся окне установите переключатель в положение **Advanced** (Дополнительно).
4. Мы будем использовать расширенные опции для фильтрации подразделения **Soningdale** со значением поля **GROSS PAY**, большим чем 213. Выберите в поле **Column** (Столбец) значение **DIVISION**, установите в поле **Operator** (Оператор) значение **equals** (равно), раскройте выпадающий список в поле **Value** (Значение) и выберите пункт **Soningdale**. Убедитесь, что в поле **And/Or** (И либо или) выбрано значение **And** (И), выберите во второй строке поля **Column** (Столбец) значение **GROSS PAY**, установите в поле **Operator** (Оператор) значение **is greater than** (больше или равно) и в поле **Value** (Значение) впишите вручную значение 213, как показано на рис. 5.18.

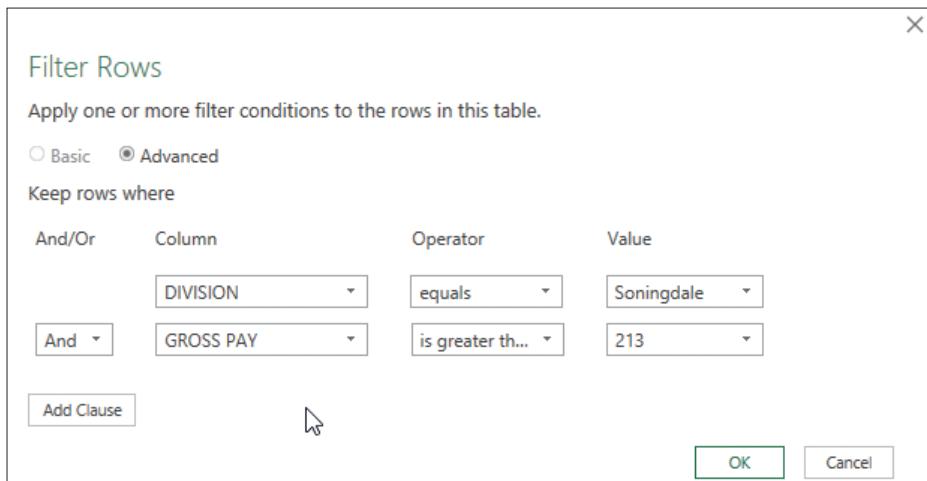


Рис. 5.18. Фильтрация с применением оператора And (И)

5. Нажмите на кнопку **OK**, чтобы увидеть результат выполнения фильтрации. Справа от наименования колонки **DIVISION** появится иконка фильтрации, говорящая о том, что на этом столбце установлен активный фильтр.

6. Теперь самостоятельно добавьте фильтр типа **Or** (ИЛИ), чтобы оставить в поле **DEPT** только значения **Shewe** или **Mankay Falls**, как показано на рис. 5.19.
7. Нажмите на кнопку **OK**, чтобы подтвердить свой выбор.

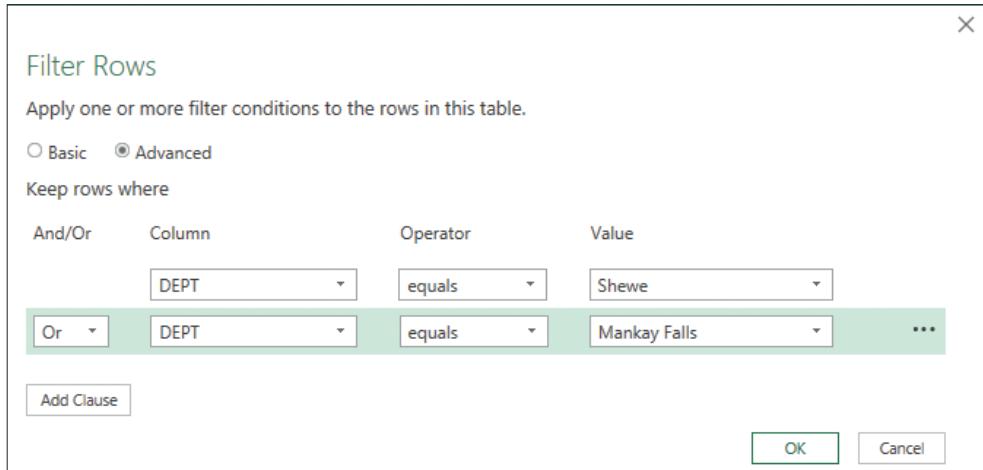


Рис. 5.19. Фильтрация с применением оператора Or (ИЛИ)

8. Теперь в нашем списке должно остаться всего 11 записей, удовлетворяющих указанным критериям, как видно на рис. 5.20.

	CODE	FIRST	SURNAME	EMP NO	DIVISION	DEPT	DATE of HIRE	12 HRS
1	4	Chris	Hume	SUN59	Sondingdale	Shewe	1988/05/12 00:00...	40
2	8	Kristen	DeVinney	SUN45	Sondingdale	Shewe	1987/06/05 00:00...	35
3	9	Robert	Murray	SUN47	Sondingdale	Mankay Falls	1987/06/10 00:00...	40
4	11	Sean	Willis	SUN09	Sondingdale	Mankay Falls	1985/07/05 00:00...	35.5
5	18	Paul	Hoffman	SUN57	Sondingdale	Shewe	1987/12/19 00:00...	40
6	33	Laura	ReaMILn	SUN77	Sondingdale	Mankay Falls	1990/08/12 00:00...	35
7	45	Kyle	Earnhart	SUN16	Sondingdale	Shewe	1984/10/08 00:00...	40
8	57	Bill	Wheeler	SUN05	Sondingdale	Mankay Falls	1981/08/14 00:00...	38
9	58	Todd	MPALters	SUN69	Sondingdale	Shewe	1989/11/23 00:00...	40
10	68	William	Abel	SUN66	Sondingdale	Mankay Falls	1989/10/14 00:00...	40
11	73	Bradley	Howard	SUN12	Sondingdale	Mankay Falls	1984/02/14 00:00...	40

Рис. 5.20. 11 строк, соответствующих выбранным условиям

9. Нажмите на кнопку **Close & Load** (Закрыть и загрузить), чтобы перенести результат обратно в Excel.

В данном разделе вы освоили фильтрацию с применением операторов **AND/OR** (И/ИЛИ). В следующем разделе мы сконцентрируемся на извлечении данных из столбца при помощи единственного критерия.

Создание фильтров с единственным критерием

В этом разделе вы познакомитесь с тем, как создавать *фильтры с единственным критерием* для извлечения данных из столбцов с помощью Power Query.

- Продолжим работать с листом Excel из предыдущего примера. Убедитесь, что в качестве источника данных вы выбрали именно лист Filter.
- В ячейке L4 введите заголовок столбца: **Division**. Непосредственно под этим заголовком введите текст Soningdale, как показано на рис. 5.21. Эти две ячейки будут использованы для формирования фильтра в нашем примере.

J	K	L
GROSS PAY		Division
50 £ 860.00		Soningdale

Рис. 5.21. Табличный критерий для фильтрации с рабочего листа

- Импортируйте источник данных в Power Query и назовите таблицу **SSGFilter**. Нажмите на кнопку **Close & Load** (Закрыть и загрузить). Переименуйте лист в **SSGFilter**.
- Теперь загрузите критерий фильтра и заголовок из ячеек L4:L5 рабочего листа в Power Query.
- Чтобы осуществить фильтрацию по одному подразделению Soningdale, нужно выполнить операцию *детализации* (drill down) для преобразования таблицы в простое значение и тем самым гарантировать, что фильтрация будет выполнена именно по подразделению Soningdale. Нажмите правой кнопкой мыши по текстовой ячейке Soningdale и выберите пункт **Drill Down** (Детализация), как показано на рис. 5.22.

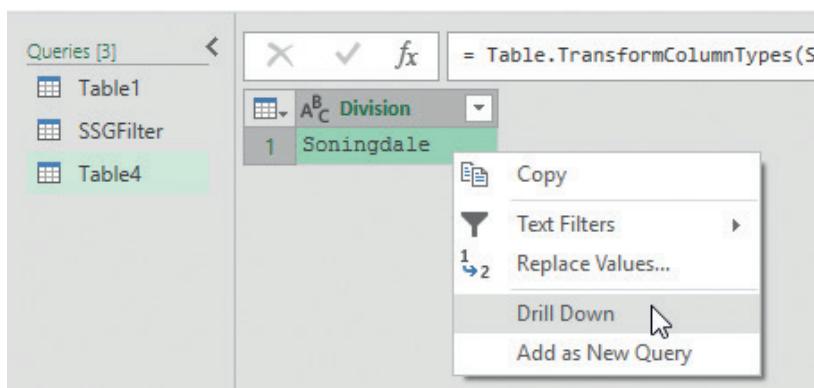


Рис. 5.22. Детализация до простого значения для фильтрации

- В результате выполнения операции детализации первая строка таблицы оказалась преобразована в текст Soningdale, что видно по рис. 5.23. Переименуйте запрос **Table4** в **Filter**.

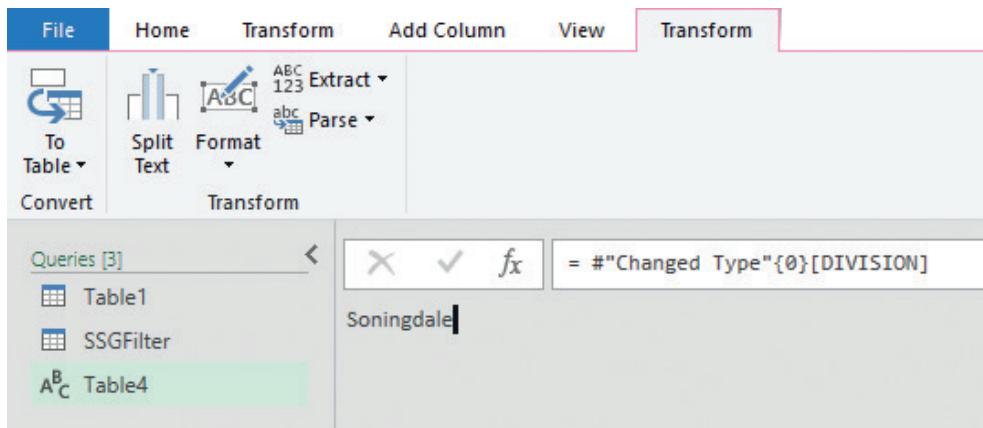


Рис. 5.23. Операция детализации

7. Для подключения к источнику данных **SSGFilter** с критерием фильтра, названным **Filter**, необходимо выполнить два шага. Во-первых, нужно вернуться к запросу **SSGFilter** и создать фильтр по столбцу **Division**. Давайте отфильтруем столбец **Division** по значению **Munerton**, как показано на рис. 5.24. Запрос будет отфильтрован, и в нем останутся только значения **Munerton**.

	CODE	FIRST	SURNAME	EMP NO	DIVISION	DEPT	DATE OF HIRE	1.2 HRS
1	1	Barry	Bally	MIL04	Munerton	Cobrella	1983/04/15 00:00...	40
2	2	Bob						5
3	3	Che						0
4	4	Chr						0
5	5	Col						0
6	6	Fra						0
7	7	Har						0
8	8	Kril						5
9	9	Rob						0
10	10	Sar						5
11	11	Sea						0
12	12	Set						2
13	13	Shi						5
14	14	Ter						0
15	15	The						5
16	16	Jam						5
17	17	George	Gorski	TBV18	View Tabue	Slangsgrow	1985/05/07 00:00...	40
18	18	Paul	Hoffman	SUN57	Sonindale	Shewe	1987/12/19 00:00...	40

Рис. 5.24. Фильтр по полю Division

8. Обратите внимание на значения в столбце **Division**.
9. Во-вторых, взгляните на строку формул – вы увидите, что заключительная часть кода на языке M обращается к следующему фильтру: «, **each** ([**DIVISION**] = «**Munerton**»)». Замените текст **Munerton** в этой формуле таким образом, чтобы был отражен критерий фильтра, созданный нами ранее. Введите слово **Filter** вместо «**Munerton**». Вы увидите, что столбец **Division** обновился и теперь отражает наш критерий: **Sonindale**. Формула в строке приобретет следующий вид:

```
= Table.SelectRows(#"Changed Type", each ([DIVISION] = Filter))
```

Обратите внимание, что мы убрали обрамляющие кавычки, поскольку теперь ссылаемся на запрос, как показано на рис. 5.25.

The screenshot shows the Power Query Editor interface. On the left, under 'Queries [3]', there are three items: 'Table1', 'SSGFilter', and 'Filter'. The 'SSGFilter' item is selected. In the main area, there is a formula bar at the top with the expression: `= Table.SelectRows(#"Changed Type", each ([DIVISION] = Filter))`. Below it is a table with six rows of data. The columns are labeled: CODE, FIRST, SURNAME, EMP NO, DIVISION, and DEPT. The 'DIVISION' column contains values like 'Suningdale', 'Cobrella', etc., and the 'DEPT' column contains values like 'Shewe', 'Mankay Fall', etc. The formula bar indicates that the 'DIVISION' column is being filtered by the value in cell A4.

Рис. 5.25. В строке формул столбец DIVISION связан с критерием фильтра Filter

10. Нажмите на кнопку **Close & Load** (Закрыть и загрузить), чтобы отправить результаты в Excel, разместив запросы **Filter** и **SSGFilter** на листе **SSGFilter**. Запрос **Filter** должен быть размещен в диапазоне **L4:L5**, а таблица **SSGFilter** – в ячейке **A4**.
11. Для проверки фильтрации введите текст **View Tabue** в ячейку **L5**, затем щелкните правой кнопкой мыши по таблице **SSGFilter** и выберите из выпадающего меню пункт **Refresh** (Обновить). Столбец **Division** обновится и будет показывать только строки со значениями **View Tabue**.

Вот вы и научились создавать фильтры с единственным критерием. Посредством них вы можете использовать любой параметр в качестве критерия поиска. В следующем разделе мы сделаем еще один шаг вперед и узнаем, как создавать фильтры с множественными критериями.

Создание фильтров с множественными критериями

В данном разделе мы затронем тему создания *динамических фильтров* для извлечения данных из нескольких столбцов с использованием более одного фильтрующего условия в Power Query.

1. Продолжим работать с листом Excel из предыдущего примера. Убедитесь, что в качестве источника данных вы выбрали лист **Filter**.
2. В ячейках **L9:L11** введите последовательно заголовок **DEPT**, а под ним критерии фильтра **Mankay Falls** и **Cobrella**.
3. Щелкните на **Mankay Falls** и используйте инструмент **Get & Transform** (Получить и преобразовать данные) для импорта таблицы данных в Power Query.
4. Далее нам необходимо продублировать запрос **FilterData**, поскольку мы намереваемся использовать его в качестве источника данных для нашего фильтра с множественными критериями. Когда нам нужно использовать данные из одного запроса для создания другого, мы применяем операцию **Duplicate**. Щелкните правой кнопкой мыши по запросу

FilterData и выберите в контекстном меню пункт **Duplicate**. Удалите последний шаг, **Filtered Rows**, из панели **APPLIED STEPS** (Примененные шаги) в правой части экрана.

5. Настроить запрос таким образом, чтобы он соответствовал двум критериям – **Mankay Falls** или **Corbella**, – можно путем объединения двух запросов с использованием инструмента **Merge Queries**. В результате наборе останутся только два выбранных отдела, а остальные данные будут скрыты. Выделите запрос **FilterData(2)**, откройте вкладку **Home** (Главная) и нажмите на кнопку **Merge Queries** (Объединить запросы).
6. В диалоговом окне **Merge** (Слияние) выделите столбец **DEPT** из запроса **FilterData(2)**, а затем в выпадающем списке выберите **Table5**. Мы это делаем, поскольку хотим видеть только совпадающие строки из двух таблиц, как показано на рис. 5.26.

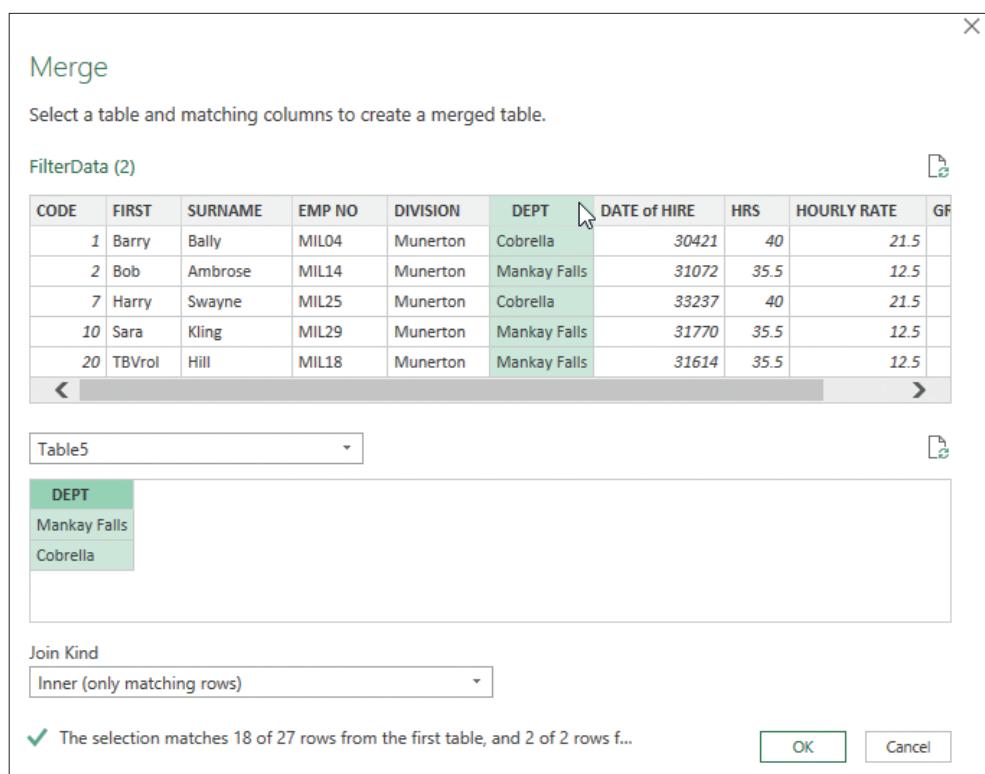


Рис. 5.26. Inner join (внутреннее соединение) дает список совпадающих строк в таблицах

7. Нажмите на кнопку **OK**, чтобы посмотреть результат. Как указано в нижней части диалогового окна **Merge** (Слияние), в первой таблице совпадение с критериями обнаружено в 18 строках из 27, а во второй – в двух строках из двух.

Рис. 5.27. Результат внутреннего объединения запросов

8. На панели **APPLIED STEPS** (Примененные шаги) последний шаг будет показан с именем **Merged Queries**. Удалите столбец **Table5**, который был добавлен после объединения двух таблиц. Для этого выделите столбец и нажмите на клавишу **Delete** на клавиатуре или выберите одноименный пункт в контекстном меню.
9. Для проверки множественных критериев измените значения под ячейкой **DEPT** на **Shewe** и **Slangsgrow**. После этого щелкните правой кнопкой мыши по таблице данных и выберите пункт **Refresh** (Обновить).

Рис. 5.28. Результат применения фильтра с множественными критериями

После прочтения данного раздела вы должны уметь создавать фильтры с множественными критериями для динамического манипулирования выводом отчетов. В следующем разделе мы узнаем, как можно удалить из запроса дублирующиеся строки.

Удаление дублирующихся строк

Пытаясь удалять дублирующиеся строки вручную, вы с большой долей вероятности можете допустить ошибку и даже не заметить этого, особенно если будете делать это с клавиатуры. Инструмент Power Query располагает собственным механизмом удаления дубликатов в таблицах. Чтобы им воспользоваться, нужно выполнить следующие действия.

1. Загрузите данные с листа **SalesData.xlsx** в редактор запросов Power Query при помощи Power BI. Если колонка **Date** импортировалась как текст, используйте локаль для преобразования ее в тип даты. Для этого щелкните правой кнопкой мыши по заголовку столбца и в подменю **Change Type** (Тип изменения) выберите пункт **Using Locale** (Используя локаль). Этот способ прекрасно подходит при обработке текстовых файлов, содержащих даты и числовые показатели в форматах других стран. Мы уже говорили об этом в главе 3.

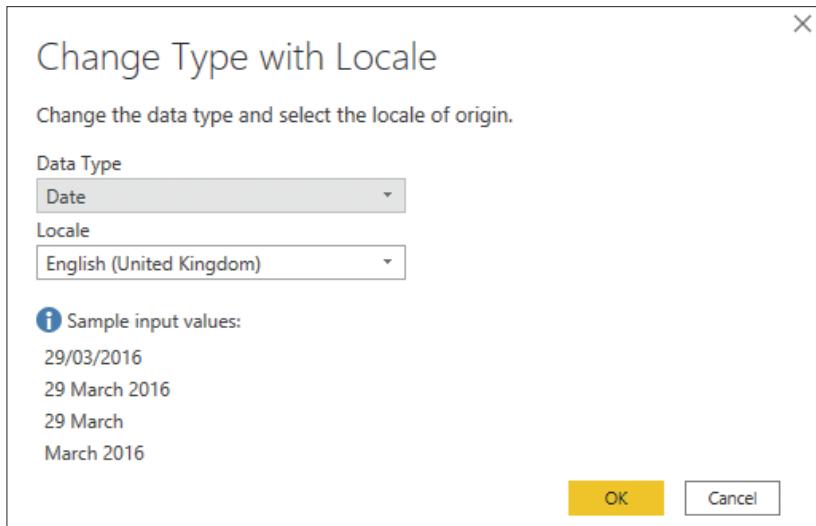


Рис. 5.29. Изменение локали

2. Мы обнаружили, что в этом отчете присутствуют дублирующиеся строки. Например, третья и четвертая строки таблицы полностью идентичны. Чтобы избавиться от дубликатов, перейдите на вкладку **Home** (Главная) и в выпадающей кнопке **Remove Rows** (Сократить строки) выберите пункт **Remove Duplicates** (Удалить дубликаты), как показано на рис. 5.30.

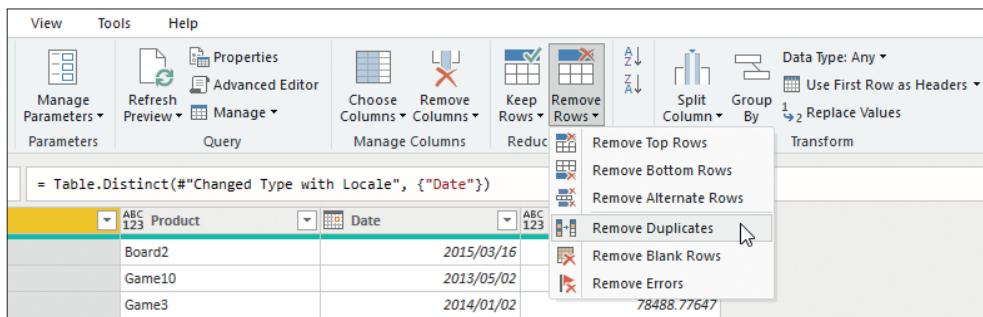


Рис. 5.30. Удаление дублирующихся строк

3. Изначально в таблице содержалось 57 строк. После удаления дубликатов их осталось 54. Заметьте, что мы не выделяли никакие столбцы для выполнения этого действия. Вам нужно быть очень внимательными к тому, какие столбцы вы выделяете перед удалением дубликатов. Например, если выбрать столбцы **Region** и **Product**, а затем запустить операцию удаления дублирующихся строк, результат будет иным. Поэкспериментируйте с этим на наборе данных **SalesData**.

Теперь, когда вы знаете, как избавляться от дублирующихся строк, пришло время научиться заменять пустые значения в запросе.

Замена пустых значений

Для замены пустых значений в редакторе запросов Power Query вам необходимо выполнить те же действия, что и в Excel, а именно:

1. Создайте дубликат запроса SalesTeams и назовите его SalesTeamsNull. Удалите шаги на панели **APPLIED STEPS** (Примененные шаги) до тех пор, пока не увидите сырье данные с пустыми значениями (null). Щелкните на **null**, чтобы выделить значение.
2. Перейдите на вкладку **Home** (Главная) и нажмите на кнопку **Replace Values** (Замена значений).

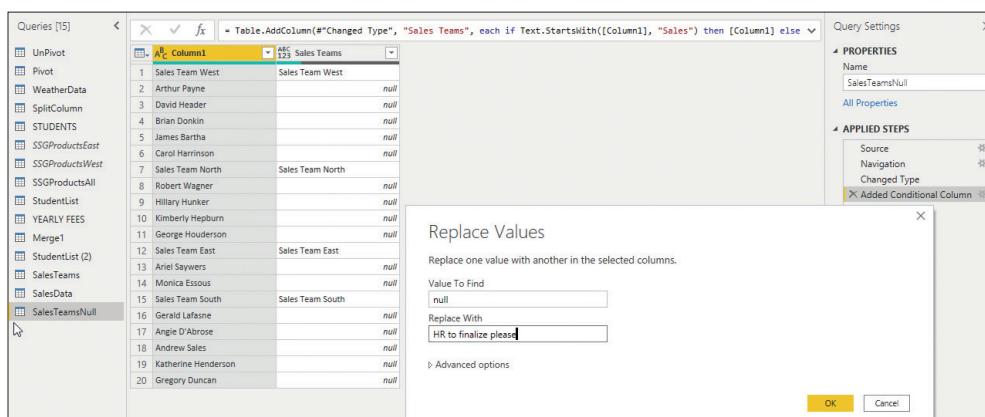


Рис. 5.31. Замена пустых значений

В открывшемся диалоговом окне в поле **Value To Find** (Значение для поиска) уже будет стоять **null**, поскольку мы предварительно выбрали пустую ячейку. Введите текст **HR to finalize please** в поле **Replace With** (Заменить на). Заметьте, что в нижней части диалогового окна есть ссылка на дополнительные опции замены.

3. Нажмите на кнопку **OK** для выполнения замены и просмотра результата.

В данном разделе мы научились заменять одни значения на другие в запросах. Теперь рассмотрим допустимые опции для работы с заголовками строк в Power Query.

Работа с заголовками строк

В Power Query у вас есть возможность использовать первую строку в качестве заголовков и заголовки – в качестве первой строки. Правда, это вам понадобится не так часто, поскольку обычно вы будете прямо во время импорта данных указывать, что первую строку стоит рассматривать как заголовки столбцов. Но при работе с некоторыми наборами данных вам эта опция пригодится. Использовать ее можно так:

1. Импортируйте файл с разделителями **SalesData1.csv** в Power Query.
2. После подключения к источнику данных вы увидите, что столбцы названы по умолчанию: **Column1** (Столбец1), **Column2** (Столбец2), **Column3**

(Столбец3) и **Column4** (Столбец4), при этом заголовки располагаются в строке **Row1** (Строка1).

3. Чтобы сделать содержимое строки **Row1** заголовками, перейдите на вкладку **Home** (Главная) и в соответствующей выпадающей кнопке на ленте выберите пункт **Use First Row as Headers** (Использовать первую строку в качестве заголовков).
4. Теперь содержимое первой строки будет восприниматься как заголовки таблицы.

В следующем разделе мы посмотрим, как можно разделять столбцы, чтобы их содержимое выводилось в нескольких колонках.

Разделение столбцов

Вы можете задаться вопросом, зачем вообще может понадобиться *разделять столбцы* средствами Power Query при наличии в Excel опций мгновенного заполнения (**Flash Fill**), распределения текста по столбцам (**Text to Columns**) и соответствующих формул. Дело в том, что, разделяя столбцы непосредственно в Power Query, а не в Excel, вы получаете дополнительный бонус в виде возможности повторного использования этих инструментов и обновления данных. Да, в Excel вы также можете обновлять данные, применяя формулы, но при этом рискуете столкнуться с проблемой, связанной с разной шириной текста, – например, когда речь идет о городах в столбце.

Кнопка **Split Column** (Разделить столбец) находится в Power Query на вкладке **Home** (Главная), что видно на рис. 5.32. Также можно получить доступ к этому функционалу из контекстного меню столбца.

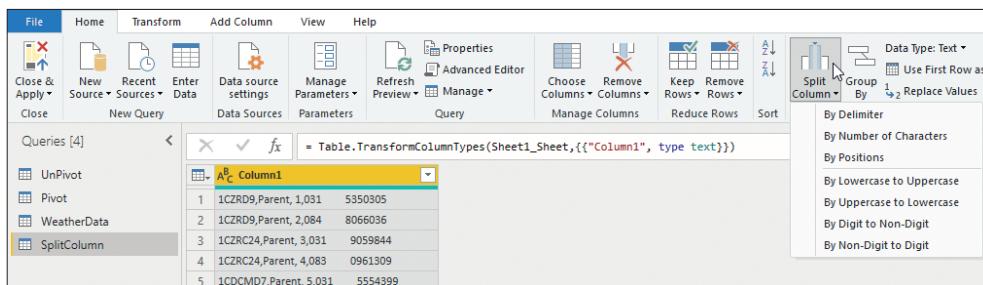


Рис. 5.32. Разделение столбца в Power Query

После вводных слов о разделении столбцов давайте приступим к практике.

1. Загрузите данные из рабочей книги **SplitColumns.xlsx** в Power Query.
2. Выделите столбец **Column1**, перейдите на вкладку **Home** (Главная) и нажмите на кнопку **Split Column** (Разделить столбец).
3. Вам будет предложен дополнительный выбор способа разделения столбца. Мы будем использовать первый пункт – **By Delimiter** (По разделителю) – и выберем в открывшемся диалоговом окне вариант **Comma** (Запятая), как показано на рис. 5.33.

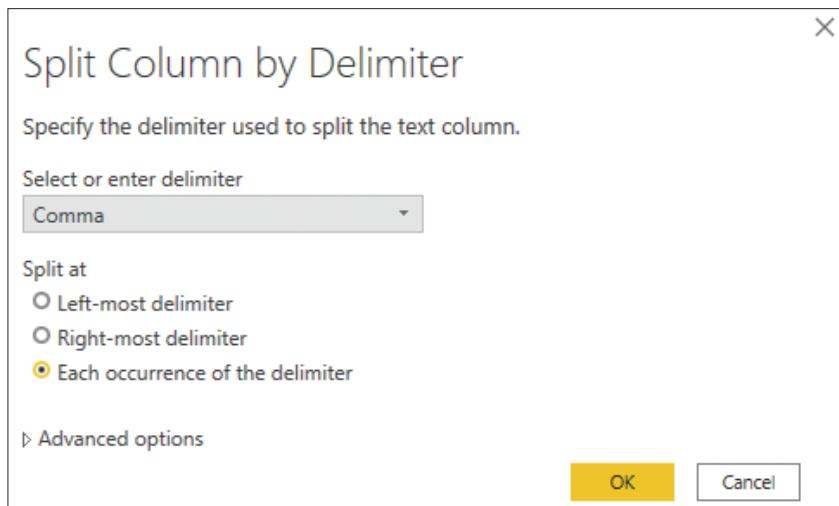


Рис. 5.33. Разделение столбца по разделителю запятой

- Убедитесь, что в разделе **Split at** (Разделение по) выбран пункт **Each occurrence of the delimiter** (По каждому вхождению разделителя). В противном случае лишь первая часть содержимого столбца до запятой будет вынесена в отдельный столбец. А поскольку у нас в тексте содержится три набора данных, разделенных запятыми, нам стоит позаботиться о том, чтобы каждый из них был выделен в свой столбец.
- Нажмите на кнопку **OK**, чтобы увидеть результат, показанный на рис. 5.34.

	A C	Column1.1	A C	Column1.2	1 2 3	Column1.3	A C	Column1.4
1	1CZRD9	Parent			1	031	5350305	
2	1CZRD9	Parent			2	084	8066036	
3	1CZRC24	Parent			3	031	9059844	
4	1CZRC24	Parent			4	083	0961309	
5	1CDCMD7	Parent			5	031	5554399	
6	1CDCMD7	Parent			6	083	8954814	

Рис. 5.34. Результат разделения столбца по запятым

- Теперь можно дать новообразованным столбцам соответствующие им имена и удалить колонку **Column1.2**, поскольку она нам не нужна. Столбец **Column1.4** также должен быть разделен, ведь в нем содержится одновременно и код, и номер. Между этими составляющими располагаются пробелы, от которых мы избавимся путем разделения столбца по позиции. Выделите этот столбец в таблице и из выпадающей кнопки **Split Column** (Разделить столбец) выберите пункт **By Positions** (По позициям).

7. Power Query пройдет по колонке и сделает предположение о позициях для разделения на основании ее содержимого. В нашем случае это предположение оказалось верным, как показано на рис. 5.35.

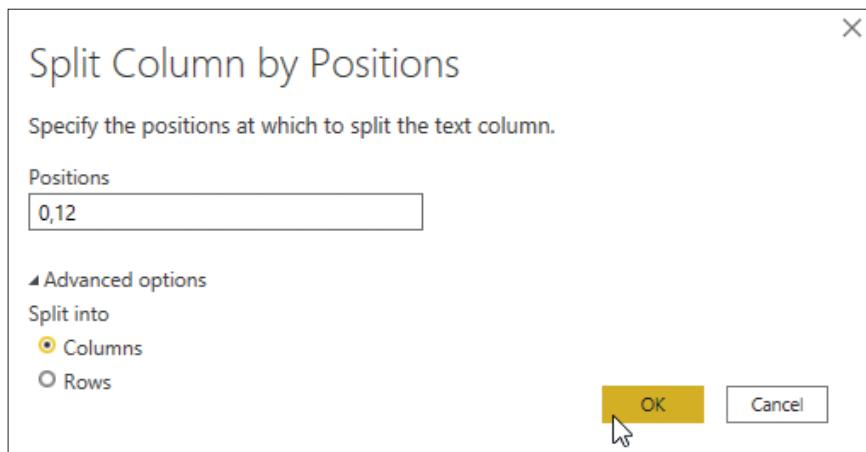


Рис. 5.35. Разделение столбца по позициям

8. Нажмите на кнопку **OK**, чтобы увидеть результат. Теперь у нас есть два столбца: один для кода, второй для номера. Вы также заметите, что ведущие нули в колонке с номерами исчезли, поскольку ее тип был автоматически преобразован из текстового в числовой. Если вам необходимо оставить нули в номерах, просто удалите последний шаг с названием **Changed Type1** на панели **APPLIED STEPS** (Примененные шаги).

При использовании разделения столбца по разделителям стоит помнить, что после обновления запроса новые данные могут не соответствовать настройкам, выставленным в расширенных параметрах операции.

После создания запроса нажмите на иконку с изображением шестеренки справа от шага **Split Column by Delimiter** (Разделить столбец по разделителю) на панели **APPLIED STEPS** (Примененные шаги) для просмотра критериев, установленных Power Query автоматически. В нашем примере колонка была разбита на четыре составляющие. После обновления данных по кнопке **Refresh** (Обновить) вполне может появиться пятый столбец, который не будет отображен в нашем выводе. Для исправления этой ситуации следует открыть раздел **Advanced options** (Расширенные параметры) в диалоговом окне разделения столбцов и в соответствии с новыми требованиями заполнить поле **Number of columns to split into** (Число столбцов, на которое необходимо разделить столбец). В результате данные будут обновлены автоматически.

Теперь, когда мы узнали, как разделять столбцы непосредственно в Power Query, рассмотрим инструменты объединения и добавления запросов.

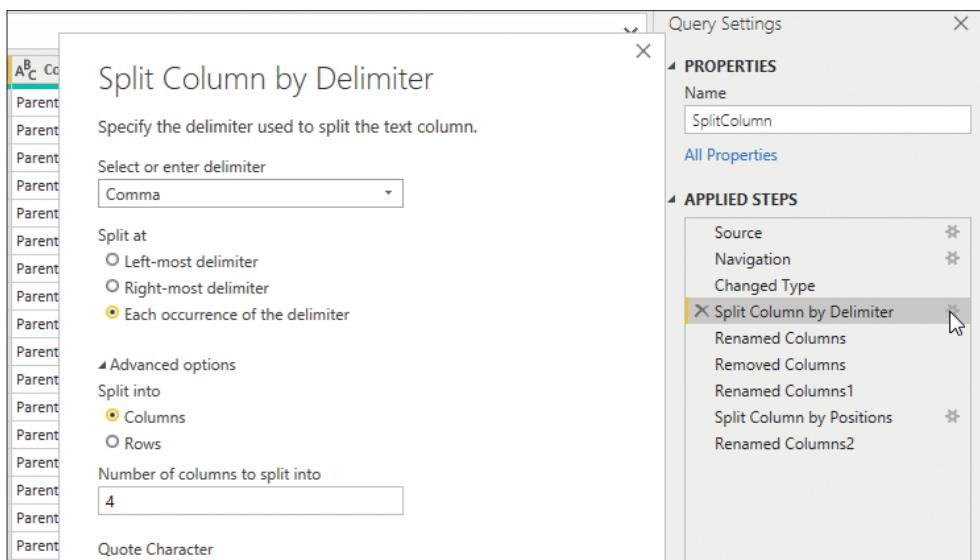


Рис. 5.36. Нажмите на шестеренку для изменения расширенных параметров операции

Объединение и добавление запросов

В данном разделе вы научитесь комбинировать, объединять и извлекать данные из источников при помощи Power Query. Мы поведем речь об исключительно полезных инструментах, которые могут быть эффективно использованы в процессе преобразования данных.

Объединение посредством комбинирования

При помощи этого инструмента допустимо *объединять* (merge) источники данных или таблицы, кардинально отличающиеся друг от друга по структуре. Единственным условием для объединения является наличие связи между источниками. Обычно связь осуществляется по общему столбцу в двух источниках. Эта концепция объединения таблиц ничем не отличается от связей в реляционных базах данных с использованием первичных и внешних ключей.

Допустим, в одной таблице у нас есть информация о студентах, а в другой – данные об оплате за обучение. Мы можем объединить эти данные и выровнять информацию в одной общей таблице. Инструмент *объединения* (**Merge**) дает возможность выбрать общее поле для источников и объединить таблицы в соответствии со значениями этого поля. Приступим.

- Создайте два запроса на основании таблиц **STUDENTS** и **YEARLY FEES**, располагающихся в файле **SSGSchoolAdmin.accdb** из сопутствующих материалов.
- Переименуйте запрос **STUDENTS** в **StudentList**. В обеих исходных таблицах есть общее поле **Code**. Его мы и будем использовать для создания связи между запросами.

Выделите запрос **StudentList**, перейдите на вкладку **Home** (Главная) и выберите в выпадающей кнопке **Merge Queries** (Объединить запросы) пункт **Merge Queries as New** (Объединить запросы в новый).

1. В открывшемся диалоговом окне **Merge** (Слияние) выберите поле **Code** из таблицы **StudentList**. Именно это поле мы будем использовать для объединения таблиц. Во втором выпадающем списке выберите запрос **YEARLY FEES** и затем щелкните на столбце **Code** для идентификации поля для связи. Столбцы, по которым происходит объединение, не обязательно должны одинаково называться. Если вы хотите объединить таблицы по двум и более столбцам, выберите их в верхней и нижней таблицах последовательно с нажатой клавишей **Ctrl** – в одном и том же порядке. Также вы можете выбрать нужное значение в списке **Join Kind** (Тип соединения). При самом распространенном выборе **Left Outer – all from first, matching from second** (Внешнее соединение – все из первой таблицы, совпадающие из второй), показанном на рис. 5.37, будут выбраны все без исключения строки из верхней таблицы в диалоговом окне и только совпадающие по значениям из выбранных столбцов для связки – из нижней.

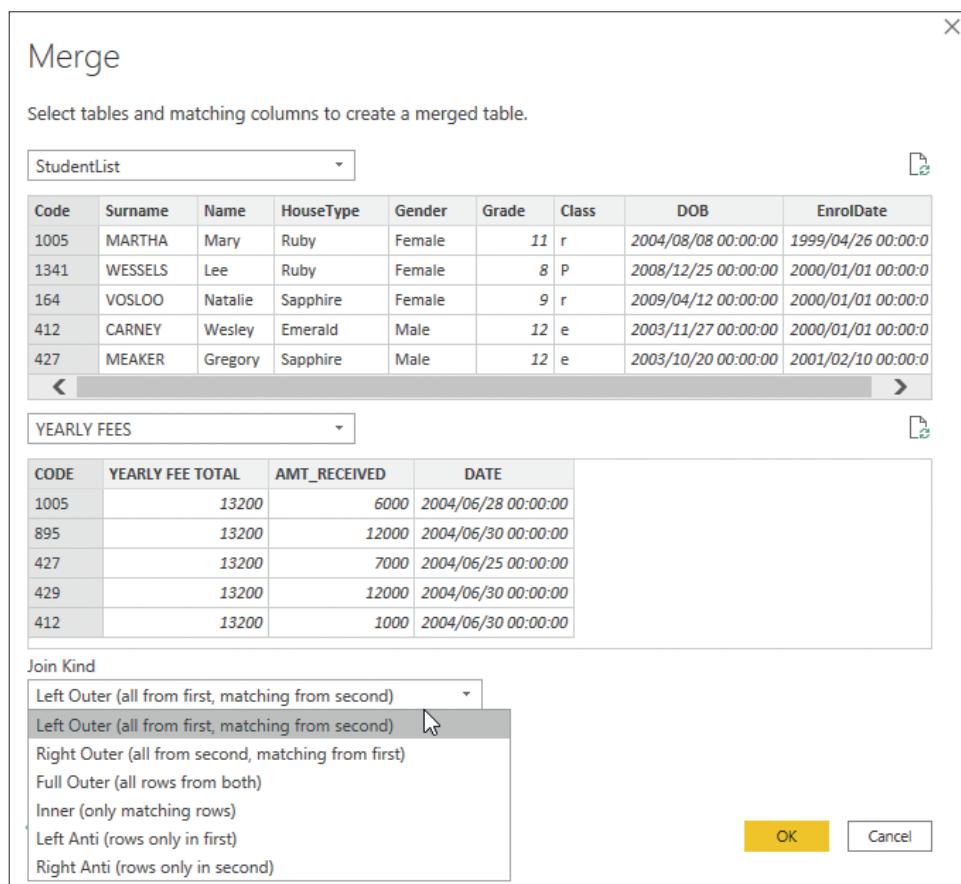


Рис. 5.37. Объединение запросов по совпадающим столбцам

2. Нажмите на кнопку **OK** для подтверждения объединения.
3. После этого будет создан новый запрос с именем **Merge1**, который объединит в себе результат связывания исходных запросов. Если прокрутить таблицу запроса вправо, вы увидите, что к исходной таблице добавилась колонка **YEARLY FEES**. Этот столбец именуется *структурированным* (structured column), поскольку в каждой его строке содержится целая структура данных (таблица), и если щелкнуть мышью по свободному месту в ячейке столбца (не на слове *Table*), то в окне ниже будет показана вложенная таблица. Таким образом, мы можем видеть, что для студента с кодом 1005 в связанной таблице **YEARLY FEE** содержится две записи, соответствующие платежам, как показано на рис. 5.38.

Queries [16]	X ✓ fx	= Table.ExpandTableColumn(Source, "YEARLY FEES", {"CODE", "YEARLY FEES"})		
	A _C Code	A _C Surname	A _C Name	A _C House
UnPivot	1 1005	MARTH	Mary	Ruby
Pivot	2 1005	MARTH	Mary	Ruby
WeatherData	3 1341	WESSELS	Lee	Ruby
SplitColumn	4 164	VOSLOO	Natalie	Sapphire
STUDENTS	5 412	CARNEY	Wesley	Emerald
SSGProductsEast	6 412	CARNEY	Wesley	Emerald
SSGProductsWest	7 412	CARNEY	Wesley	Emerald

Рис. 5.38. Две записи в таблице **YEARLY FEE** для студента с кодом 1005

4. Теперь мы сделаем еще один шаг и выровняем структуру объединенной таблицы, поскольку Power BI просто не поймет такую концепцию с вложенными друг в друга таблицами. Щелкните по иконке **Combine Files** (Объединить файлы) справа от заголовка столбца **YEARLY FEES**, как показано на рис. 5.39. Этот выбор также доступен в группе **Combine** (Комбинировать), расположенной на вкладке **Home** (Главная).

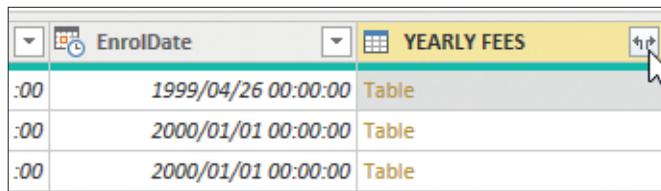


Рис. 5.39. Иконка комбинирования файлов для выравнивания структуры таблицы

5. В появившемся окне, показанном на рис. 5.40, выберите столбцы, которые вы хотите оставить в объединенном запросе.

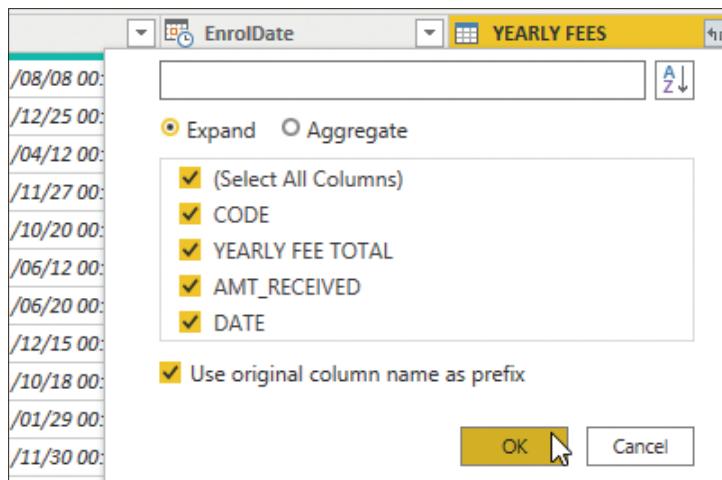


Рис. 5.40. Поля, которые необходимо оставить

В данном примере мы сохраним все столбцы. Нажмите на кнопку **OK** для подтверждения своего выбора.

6. Теперь наши таблицы оказались объединены в единую таблицу, готовую к дальнейшему анализу средствами Excel или Power BI.

Итак, вы изучили инструмент объединения запросов с использованием первичных ключей в рамках Power Query. В следующем разделе посмотрим, как можно объединять несколько столбцов в один.

Объединение текста и значений в один столбец

Используя инструмент *объединения столбцов*, очень важно помнить, что порядок указания колонок для выполнения слияния исключительно важен. Значение из столбца, который вы выделили первым, будет стоять в объединенном столбце на первом месте. Объединение колонок в одну приведет к удалению из итоговой таблицы столбцов-источников. При этом вы можете выделять целый диапазон столбцов с зажатой клавишей **Shift**. В этом случае порядок их объединения будет таким же, как порядок выделения, если вы не укажете иное. Давайте рассмотрим пример.

1. Используем для нашего примера копию запроса **StudentList**. Создав дубликат запроса, вы увидите, что его копия получит имя **StudentList2**.
2. Для объединения колонок воедино необходимо сначала указать столбцы, значения которых мы хотим соединить. Давайте выполним слияние колонок **Grade** и **Class**. Если выбрать их в таблице в обратном порядке, итоговый вывод будет неправильным.
3. Итак, выберите сначала столбец **Grade**, а затем **Class**. Далее откройте вкладку **Transform** (Преобразование) и нажмите на кнопку **Merge Columns** (Объединить столбцы), как показано на рис. 5.41.

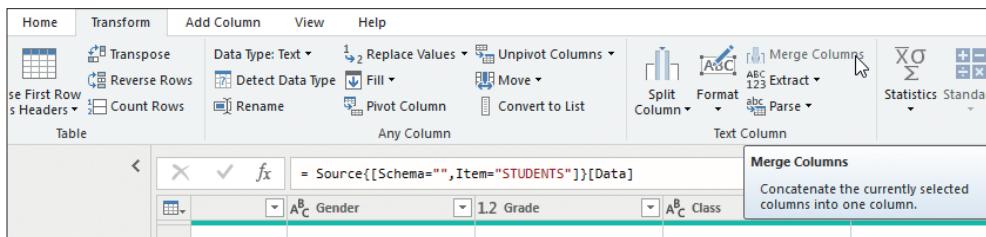


Рис. 5.41. Использование инструмента объединения столбцов

4. В появившемся диалоговом окне, представленном на рис. 5.42, выберите в списке **Separator** (Разделитель) пробел и дайте новой колонке то имя, которое пожелаете. Его можно изменить и после выполнения операции преобразования.

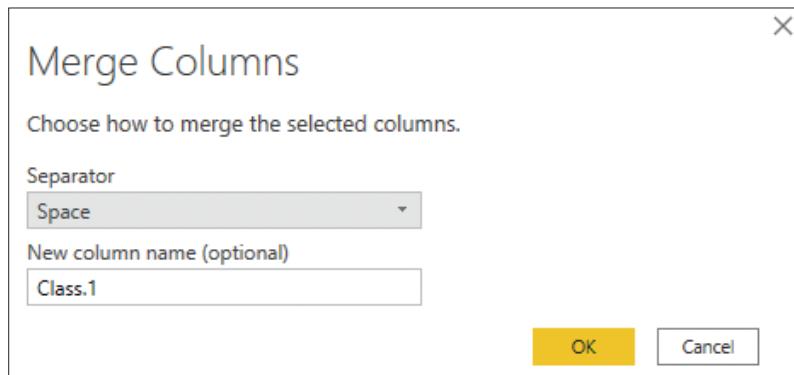


Рис. 5.42. Диалоговое окно объединения столбцов с разделителем

5. Как итог мы получили новый столбец, в котором объединены значения из полей **Grade** и **Class** с пробелом между ними.

В данной главе мы научились выполнять слияние значений двух и более столбцов с указанием разделителя. В следующем разделе посмотрим, как происходит процесс добавления двух источников в одном запросе.

Добавление таблиц

В данном разделе мы узнаем, как можно соединять вместе данные из двух таблиц. Эта операция может быть применена к двум разным источникам данных или разным таблицам. Также данный вид комбинирования данных называется *объединением* (union) и работает по аналогии с соответствующим оператором в реляционных базах данных SQL. Этот инструмент, например, идеально подойдет для сбора воедино информации о студентах разных курсов. При этом к исходным таблицам применяются следующие требования:

- желательно, чтобы структура таблиц так же, как и заголовки столбцов, была идентичной;

- по крайней мере одна колонка в таблице должна соответствовать колонке в другой таблице;
- в обеих таблицах могут присутствовать отличающиеся поля, но лучше всего выполнять объединение таблиц с большинством идентичных столбцов.

Чтобы посмотреть, как это все работает, выполним небольшую предварительную подготовку.

Создайте два отдельных запроса в Power Query на основании рабочих листов из книги ChoclateFlakSalesNW.xlsx. На рис. 5.43 показаны два запроса **SSGProductsEast** и **SSGProductsWest**, предварительно импортированные в Power Query. Эти запросы содержат информацию о нашей компании в восточном (East) и западном (West) регионах соответственно. Состав столбцов в двух таблицах идентичен, за исключением того, что в **SSGProductsEast** есть одна дополнительная колонка **Division**, что видно по рис. 5.43.

	A ^B _C Salesperson	A ^B _C Product Name	A ^B _C Division
1	Berchetta	Cholato Flak	Parklund
2	Zaepoon	Fudgi Browno	View Tabue
3	Nonturnda	Straubory	View Tabue
4	Xanandou	Plain Whito	View Tabue
5	Danderoulla	Plain Whito	View Tabue
6	Pleratodu	Plain Whito	View Tabue
7	Zaepoon	Cholato Flak	Parklund

Рис. 5.43. Результаты запроса в Power Query

Обратите внимание, что в обеих таблицах есть по два дополнительных столбца справа. Можно удалить их, чтобы максимально очистить наш набор данных. Нашей задачей является объединить данные из двух этих таблиц в одну. Приступим.

1. Выберите один из двух запросов, после чего перейдите на вкладку **Home** (Главная) и из выпадающей кнопки **Append** (Добавить) выберите один из пунктов: **Append Queries** (Добавить запросы) или **Append Queries as New** (Добавить запросы в новый). В первом случае после объединения таблиц исходные запросы для востока и запада будут утеряны, а результаты будут собраны в отдельном запросе. Второй вариант позволяет сохранить первоначальные таблицы, а для итогового набора создаст новый запрос.
2. Мы выберем второй путь, а именно кнопку **Append Queries as New** (Добавить запросы в новый). Появится диалоговое окно **Append** (Добавление) с настройками добавления данных из одной таблицы к другой.

Также есть опция объединения трех и более таблиц. В списке **Primary** (Первичная таблица) выбрана текущая открытая таблица, тогда как из списка **Table to append to the primary table** (Таблица, которую нужно добавить к первичной таблице) значение необходимо выбрать вручную, как показано на рис. 5.44.

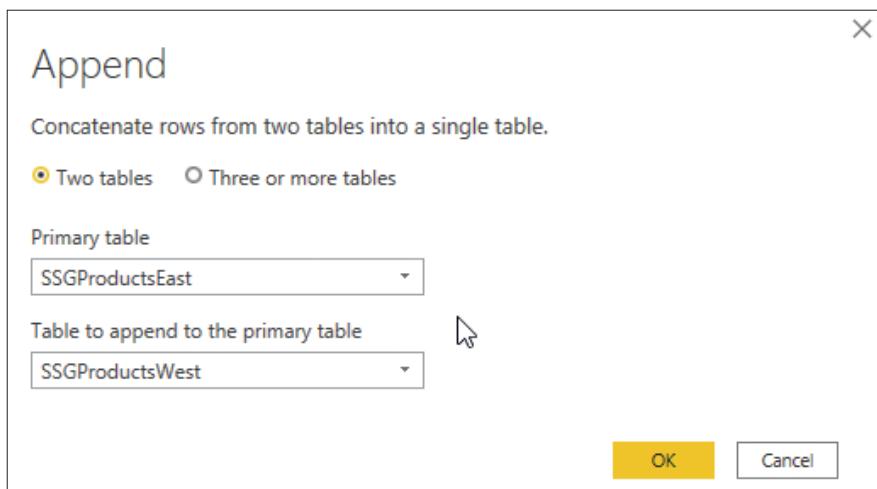


Рис. 5.44. Диалоговое окно добавления таблиц

3. Нажмите на кнопку **OK** для просмотра результата.
4. Появится новый запрос на панели **Queries** (Запросы), который будет называться **Append1**.
5. Поскольку в одной из таблиц у нас содержалось лишнее поле, в итоговом результате будут присутствовать пустые значения. Очистите результат, удалив пустые строки. Для этого выберите запрос **Append1**, перейдите на вкладку **Home** (Главная) и в выпадающей кнопке **Remove Rows** (Удалить строки) выберите пункт **Blank Rows** (Удалить пустые строки). В результате вы получите 84 строки (по 42 строки с востока и запада). Не забудьте сохранить и применить изменения.

Примечание

В столбце **Division** содержалась вспомогательная информация для продавцов из восточного региона, тогда как для продавцов с запада такой информации не было, поскольку эта колонка в соответствующем наборе данных просто отсутствовала. Здесь стоит отметить, что инструмент добавления строк не удаляет дубликаты. Если в объединяемых таблицах окажутся полностью идентичные строки, они обе останутся в результирующем наборе. В связи с этим настоятельно рекомендуем самостоятельно проверять таблицу на дубликаты после объединения запросов.

6. Убедитесь в том, что после объединения таблиц исходные запросы **SSGProductsEast** и **SSGProductsWest** не будут загружаться. Так вы сэкономите драгоценную память и повысите эффективность работы модели данных. Вам нет никакой необходимости загружать в Power BI все три запроса, достаточно будет одного результирующего. Нажмите правой кнопкой мыши на запросе **SSGProductsEast** и снимите флажок **Enable load** (Разрешить загрузку), как показано на рис. 5.45.

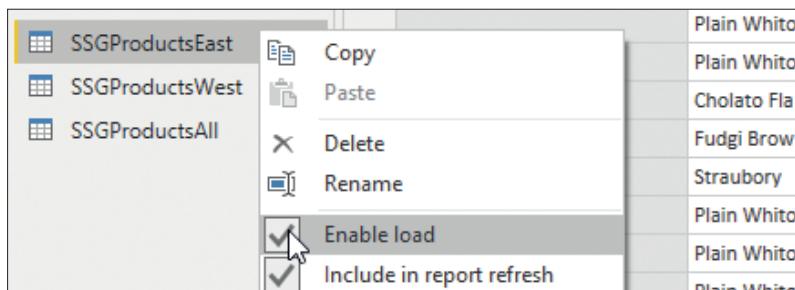


Рис. 5.45. Отмена загрузки запроса в Power BI

7. После этого появится окно, показанное на рис. 5.46, с предупреждением о том, что во время загрузки возможна частичная потеря данных.

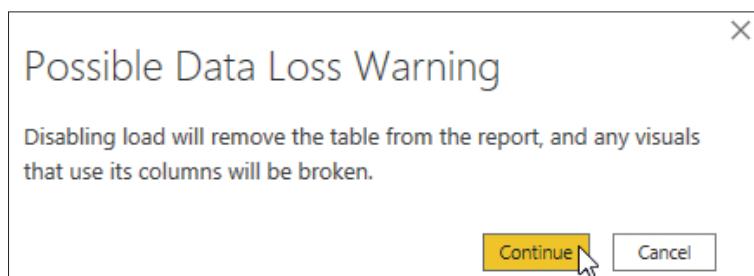


Рис. 5.46. Предупреждение о возможной потере данных

8. Нажмите на кнопку **Continue** (Продолжить). Запросы, которые не будут загружаться в Power BI, будут выделены курсивом. При этом они продолжат обновляться.

В этой главе вы научились объединять данные из разных источников в один запрос. Далее вам предстоит освоить операцию группировки данных.

ГРУППИРОВКА ДАННЫХ

В этом разделе мы посмотрим, как можно *группировать данные* в запросах непосредственно в Power Query, указывая при этом поля для группировки и выбирая операции агрегирования для новых столбцов. Операция группировки в Power Query очень похожа на аналогичную операцию в языке запросов SQL.

В данном разделе мы будем использовать запрос **SalesData**, в котором выполним группировку по регионам и выведем сумму продаж для каждого из них.

1. Выберите в Power Query запрос **SalesData** и создайте его дубликат. Переименуйте новый запрос в **SalesGroup**.
2. Перейдите на вкладку **Home** (Главная) и нажмите на кнопку **Group By** (Группировать по).
3. Для начала оставьте верхний переключатель в открывшемся диалоговом окне в положении **Basic** (Базовый). Откройте выпадающий список с полем для группировки и выберите пункт **Region**. В поле **New column name** (Имя нового столбца) введите **RegionSales** и выберите соответствующую операцию в среднем списке – пусть будет **Sum** (Сумма). В правом выпадающем списке **Column** (Столбец) выберите колонку **Sales**, как показано на рис. 5.47.

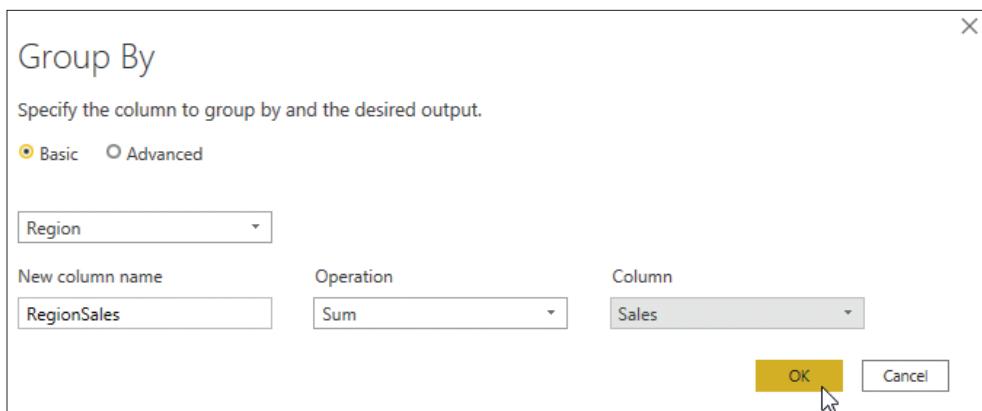


Рис. 5.47. Диалоговое окно группировки

4. Нажмите на кнопку **OK** для просмотра результата, представленного на рис. 5.48.

	A B C Region	1.2 RegionSales
1	South	628066.3657
2	West	493445.8554
3	North	2201098.556
4	East	83370.77893

Рис. 5.48. Результат суммирования продаж по регионам

5. Теперь посмотрим, как работает расширенный режим группировки. На этот раз загрузите в Power Query источник данных **SalesDataAdv.xlsx**.
6. Снова перейдите на вкладку **Home** (Главная) и нажмите на кнопку **Group By** (Группировать по).
7. Установите верхний переключатель в открывшемся диалоговом окне в положение **Advanced** (Дополнительно).

8. На этот раз используйте столбцы **Date** и **Product** в качестве группирующих. В поле **New column name** введите название колонки **DateProductCount** и выберите операцию в среднем выпадающем списке. Здесь мы используем операцию подсчета строк **Count Rows** (Считать строки), как показано на рис. 5.49, чтобы узнать, какое количество товаров было продано в каждую дату.

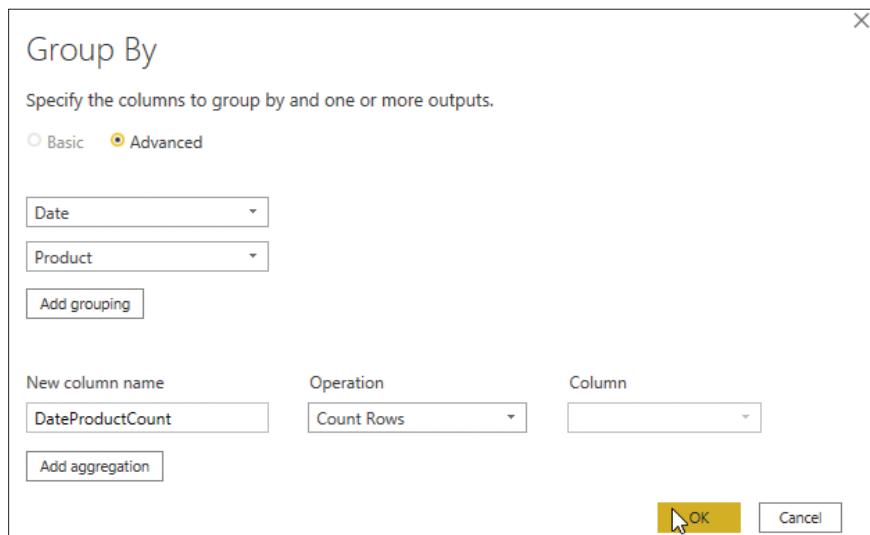


Рис. 5.49. Расширенная группировка

9. Нажмите на кнопку **OK** для просмотра результата, представленного на рис. 5.50.

	Date	Product	DateProductCount
1	16/03/2015	Board2	1
2	02/05/2013	Game10	2
3	02/01/2014	Game3	3
4	02/10/2011	Dyno2	1
5	03/04/2011	Game8	2
6	16/09/2010	Dyno4	1
7	17/12/2015	Board2	1
8	16/03/2015	Game4	1
9	02/01/2014	Dyno5	2

Рис. 5.50. Результат расширенной группировки

10. Количество проданных товаров отражается в новом столбце запроса. Если вам необходимо изменить критерии группировки, просто нажмите на иконку с шестеренкой справа от последнего примененного шага.

В данном разделе вы научились выполнять группировку данных при помощи Power Query. Теперь поговорим об опциях извлечения информации, доступных в этом инструменте.

ИЗВЛЕЧЕНИЕ ДАННЫХ

В какой-то момент вам придется иметь дело с большими таблицами, и вам может понадобиться создать несколько отдельных таблиц из источника данных в Power Query. В этом вам поможет инструмент *извлечения* (extract) данных. Стоит отметить, что решить эту задачу можно разными способами, включая создание дубликата запроса и удаление из него лишних столбцов. Также в этом разделе мы посмотрим, как можно использовать инструмент извлечения данных для вычисления возраста студента из поля с датой в запросе.

Определение возраста по дате

В данном разделе мы постараемся при помощи Power Query извлечь возраст сотрудников компании из столбца с датой, для чего преобразуем формат времени в общее число лет и округлим результат до возраста.

Для начала попробуем определить возраст студентов по дате из списка с зачислением. Для этого выполним следующие действия.

1. Импортируем таблицу со студентами из источника данных `SSSchoolAdmin.accdb` в Power Query.
2. Поскольку при загрузке данных столбец с днями рождения (**DOB**) был автоматически сконвертирован в тип **Date/Time** (Дата и время), установим для него тип **Date** (Дата).
3. Выберите колонку **DOB**, затем перейдите на вкладку **Add Column** (Добавить столбец) и из выпадающей кнопки **Date** (Дата) выберите пункт **Age** (Возраст), как показано на рис. 5.51.

The screenshot shows the Microsoft Power Query ribbon with the 'Transform' tab selected. In the 'Date' group, a tooltip for the 'Age' button is shown, stating: 'Create a new column that contains the duration between the current local time and the values in the selected column.' Below the ribbon, a table named 'STUDENTS' is visible with columns: Gender, Grade, and Class. The 'DOB' column is highlighted, and the 'Age' button in the ribbon is being pointed at by a cursor.

Рис. 5.51. Перевод даты в возраст

4. Столбец с возрастом предстанет перед вами в формате времени.
5. Теперь необходимо преобразовать полученный результат в формат года. Перейдите на вкладку **Transform** (Преобразование) и в выпадающей кнопке **Duration** (Продолжительность) выберите пункт **Total Years** (Общее число лет).

6. Последнее, что нам необходимо сделать, – это округлить годы до возраста. Для этого в группе **Number Column** (Столбец «Количество») на вкладке **Transform** (Преобразование) в выпадающей кнопке **Rounding** (Округление) выберите пункт **Round Down** (Округление с уменьшением).

7. Сохраните запрос.

Таким образом вы можете извлекать возраст из столбцов с датами в Power Query. Далее посмотрим, как можно извлечь столбец из диапазона.

Извлечение столбцов

Теперь давайте воспользуемся инструментом извлечения данных для удаления столбца из диапазона в Excel при помощи Power Query. Выполните следующие действия.

1. Откройте рабочую книгу Excel с названием *SafestSolutions-Law.xlsx*.
2. Вы обнаружите, что на рабочем листе **Sheet2** содержится запрос, настроенный при помощи Power Query. Обратитесь к панели **Queries & Connections** (Запросы и подключения), чтобы посмотреть запрос. Щелкните на нем правой кнопкой мыши и выберите пункт **Edit** (Изменить), при этом откроется редактор запросов Power Query.
3. Мы извлечем из запроса столбец **CASE NO** и преобразуем его в список, состоящий из одной колонки. В расширенном редакторе можно увидеть, что выражение **Source** ссылается на текущую рабочую книгу, а **Output** – обратно на **Source**. Мы изменим выражение **Output** таким образом, чтобы оно ссылалось только на нужный нам столбец **CASE NO**.
4. Нажмите на шаг **Output**, перейдите в строку формул и напишите следующий код: `= Source[CASE NO]`. Таким образом, в этом выражении мы будем ссылаться не на всю таблицу целиком, а только на столбец **CASE NO**. Вы также можете отредактировать формулу выражения **Output** непосредственно в расширенном редакторе.
5. Щелкните на галке слева от строки формул, тем самым подтверждая изменения.

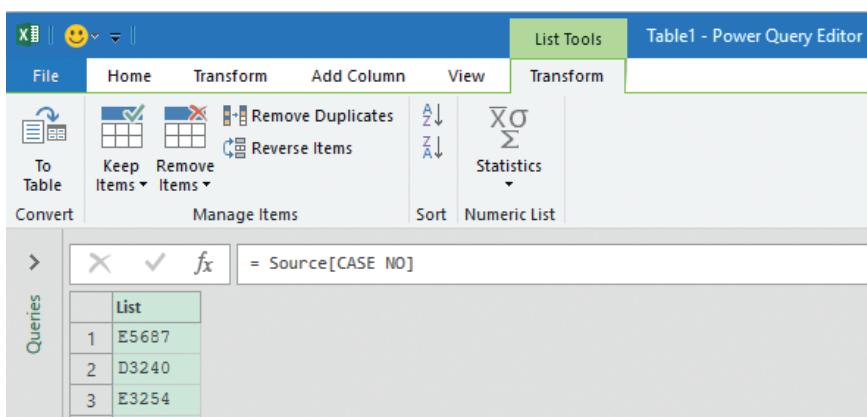


Рис. 5.52. Извлечение столбца в Power Query

6. Вы произвели свое первое извлечение столбца посредством Power Query. Теперь он представлен как список.
7. Осталось преобразовать этот список в таблицу, для чего необходимо на вкладке **Transform** (Преобразование) нажать на кнопку **To Table** (В таблицу). После этого при помощи кнопки **Close & Load** (Закрыть и загрузить) вы можете перенести результаты обратно в Excel.

Теперь, когда вы узнали, как можно преобразовывать данные внутри запроса с целью извлечения столбцов, сконцентрируемся на различных опциях извлечения значений с разделителями.

Извлечение данных из столбцов

Инструмент извлечения данных в Power Query может использоваться для получения и манипулирования различными частями строк в источнике данных при составлении итогового вывода. В данном разделе мы научимся пользоваться этим инструментом для извлечения данных из столбцов.

1. Мне прислали список серийных номеров товаров, определенные части которых необходимо использовать в другом запросе.
2. Откройте рабочую книгу *SerialNo.xlsx* и импортируйте таблицу в Power Query.
3. Перейдите на вкладку **Transform** (Преобразование) и раскройте выпадающую кнопку **Extract** (Извлечь). Список вариантов, который вам будет предложен, показан на рис. 5.53.

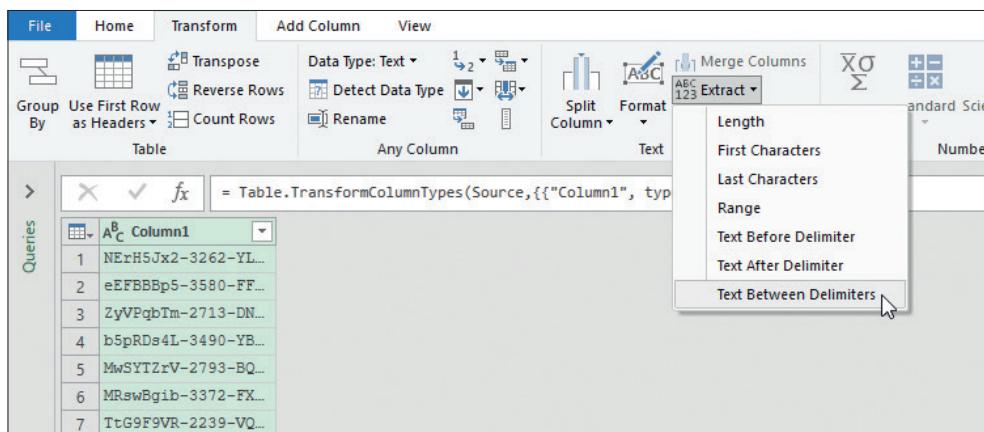


Рис. 5.53. Преобразование данных с использованием инструмента извлечения

4. Попробуем вытащить из номера его среднюю часть. Для этого выберем из предложенного списка пункт **Text Between Delimiters** (Текст между разделителями). Это позволит нам извлечь внутренние составляющие серийных номеров, заключенные в дефисы.
5. Введите символ дефиса в поля **Start delimiter** (Начальный разделитель) и **End delimiter** (Конечный разделитель), как показано на рис. 5.54, и нажмите на кнопку **OK** для просмотра результата.

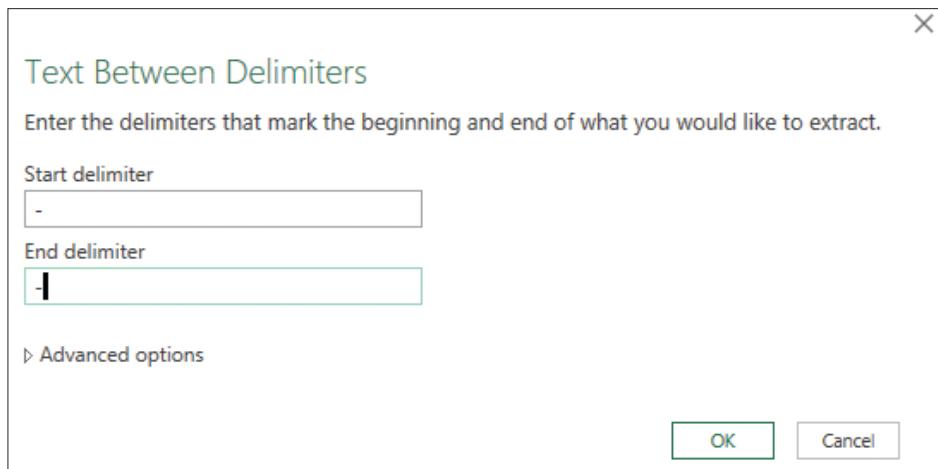


Рис. 5.54. Извлечение текста между разделителями

6. Нажмите на кнопку **Close & Load** (Закрыть и загрузить), чтобы отправить данные обратно в Excel, как показано на рис. 5.55.

The screenshot shows an Excel spreadsheet with a single column named 'Column1'. The data in the column is as follows:

	Column1
1	3262
2	3580
3	2713
4	3490
5	2793
6	3372
7	2239
8	2575
9	2680

To the right of the spreadsheet is the 'Queries & Connections' ribbon tab. It shows a list of '1 query' named 'Table1' with the message '1 000 rows loaded.'

Рис. 5.55. Результат выгрузки обработанных данных в Excel

Попробуем еще один вариант. Перейдите на рабочий лист **Sheet1** и импортируйте диапазон для создания второго запроса в Power Query.

7. На этот раз мы будем использовать пункт **Range** (Диапазон) из выпадающего списка для определения начала и конца извлекаемого фрагмента текста. В поле **Starting Index** (Начальный индекс) введите цифру 9, чтобы избавиться от первых девяти символов в серийном номере. В поле **Number of Characters** (Число символов) введите цифру 4, что позволит оставить четыре символа и удалить остаток, как показано на рис. 5.56.
8. Нажмите на кнопку **OK**, чтобы посмотреть результаты, а затем отправьте обработанные данные обратно в Excel посредством опции **Close & Load** (Закрыть и загрузить).

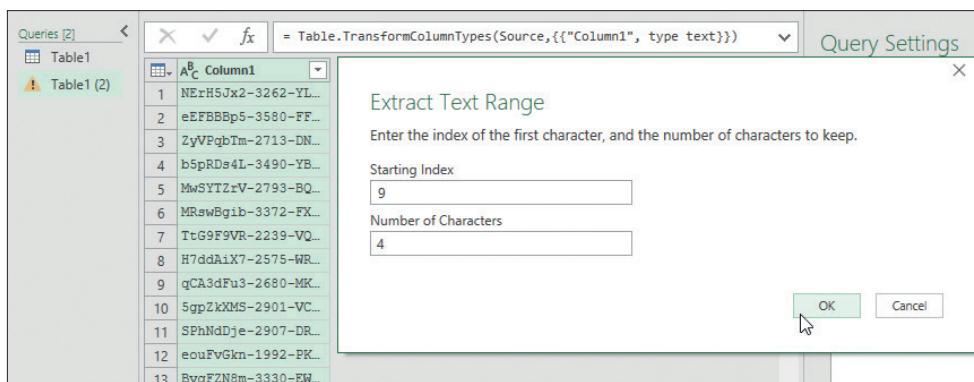


Рис. 5.56. Извлечение подстроки по диапазону

В заключительном разделе данной главы мы показали, как можно использовать инструмент извлечения данных применительно к текстовому содержимому столбцов. Поэкспериментируйте с оставшимися вариантами из выпадающего списка, этот функционал может оказаться очень полезным в работе.

ЗАКЛЮЧЕНИЕ

В этой главе вы научились пользоваться инструментами сведения и отмены свертывания столбцов с целью придания данным подходящей структуры для дальнейшего анализа. Также мы прошлись по всем доступным опциям обновления информации, рассмотрев их преимущества и недостатки.

В разделе, посвященном инструментам для работы со столбцами и строками, вы познакомились со способами очистки данных путем удаления верхних и нижних строк, а также дубликатов, замены пустых значений и добавления строки заголовков. Кроме того, вы узнали, как можно избавиться от лишних столбцов и добавить колонку с индексом, помогающую при проведении анализа.

Также мы показали, как можно создавать условные столбцы с использованием конструкции `if...then...else`. В дальнейшем вам пришлось воспользоваться своими навыками фильтрации данных в Excel, чтобы научиться пользоваться фильтрами и сортировкой в Power Query с использованием операторов AND/OR. Помимо этого, вы узнали, как создавать динамические фильтры с применением множества критериев для извлечения нужных данных. Попутно вы овладели искусством разделения столбцов в соответствии с разделителями.

После этого мы обратились к инструменту объединения данных для комбинирования столбцов, а также освоили группировку данных в Power Query в базовом и расширенном вариантах. В заключение мы посмотрели, как можно извлекать возраст из колонок с датами, и научились разбивать серийные номера товаров по разделителям и иным признакам.

В следующей главе мы продолжим работать с запросами и функциями в Power Query. Вы узнаете, как можно эффективно использовать функцию `If`, и научитесь создавать параметры. Кроме того, мы познакомимся с полезными функциями `Modulo` и `Index`, а также рассмотрим варианты загрузки нескольких файлов.

Глава 6

Углубленное изучение запросов и функций Power Query

В этой главе мы поговорим о применении полезных функций в Power Query, таких как IF, Index и Modulo. С их помощью вы сможете преобразовывать неотформатированные диапазоны на листах Excel в четко структурированные таблицы, в которых будет только та информация, которая необходима для ваших запросов.

Вы научитесь создавать параметры для изменения пути запроса, а также добавлять сразу несколько файлов или листов. Это позволит вам менять пути запросов «на лету», не покидая Excel, и все, что вам необходимо будет делать, – это дополнять свои файлы в папке.

В данной главе мы изучим следующие темы:

- использование функции IF в Power Query;
- создание таблицы параметров для запросов;
- применение функций Index и Modulo;
- добавление нескольких файлов и листов.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Вам понадобится интернет-соединение для загрузки соответствующих файлов с GitHub. Файлы с кодом для этой главы можно скачать по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

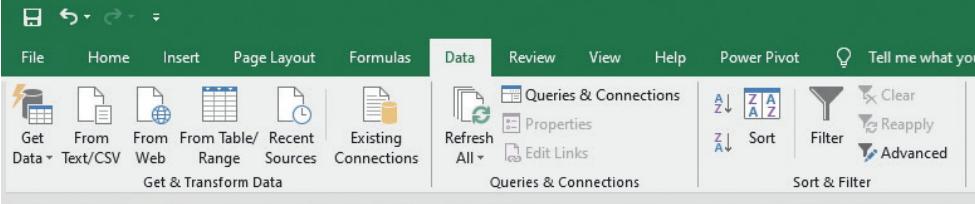
Видеофрагмент, посвященный данной главе, располагается по адресу: https://www.youtube.com/watch?v=QSTjKk1MeKY&list=PLLeLcvrwLe1860_GJEzs47WaZXwZjwTN83&index=7&t=0s.

В этой главе мы предполагаем, что у вас есть понимание о работе базовых функций в Excel, таких как IF, OR и AND. Также вам необходимо будет уметь загружать данные в Power Query и добавлять пользовательские столбцы.

Использование функции IF в Power Query

Хотя применение функции **IF** в Power Query может серьезно отличаться от того, к чему вы привыкли в Excel, начав ее использовать, вы быстро освоитесь и не будете испытывать дискомфорта. Вам даже может показаться, что применять ее в Power Query легче, чем в Excel.

Взгляните на таблицу данных, изображенную на рис. 6.1.



	A	B	C	D	E	F	G	H	I	J
1	Order No	Order Date	Order Type	Delivery Date	Salesman	Region	Product	No. Customers	Net Sales	Profit / Loss
2	13658	01/01/2019	Online	null	Jordan	West	Dyno1	4	984	625
3	13663	01/01/2019	Pickup		John	North	Dyno3	8	2,253	653
4	13667	01/01/2019	Online	null	Jordan	West	Dyno2	4	920	569
5	13669	01/01/2019	Pickup		Lucie	Midlands	Dyno1	4	951	756
6	13670	01/01/2019	In Store	05/01/2019	Jordan	North	Dyno3	9	1,233	486
7	13673	01/01/2019	Pickup		Clive	North	Dyno3	7	1,064	436
8	13676	02/01/2019	Online	null	John	Midlands	Dyno2	8	1,736	690
9	13679	02/01/2019	Pickup		Clive	Midlands	Dyno3	10	2,540	835
10	13682	02/01/2019	Pickup		John	North	Dyno3	8	1,528	574
11	13684	02/01/2019	In Store	08/01/2019	Clive	Midlands	Dyno3	9	2,592	857
12	13688	03/01/2019	Online	null	Clive	Midlands	Dyno4	1	1,866	1,011
13	13689	03/01/2019	Pickup		Lucie	Midlands	Dyno2	10	1,430	534
14	13694	03/01/2019	Pickup		John	West	Dyno3	6	684	225
15	13695	03/01/2019	Online	null	Jordan	Midlands	Dyno3	9	1,512	503
16	13698	03/01/2019	Pickup		Clive	Midlands	Dyno2	10	2,240	722
17	13700	04/01/2019	Pickup		John	North	Dyno3	6	1,983	735

Рис. 6.1. Исходный лист перед обработкой

Представим, что у нас есть такой лист и нам необходимо определить, был ли доставлен размещенный заказ. В исходном файле у нас нет соответствующей колонки, и хотя мы могли бы добавить ее в Excel, сделаем это при помощи Power Query. Одной из причин этого является то, что данный файл может быть всего лишь шаблоном для компании, а значит, нам придется каждый раз добавлять одну и ту же колонку. Сделав это в Power Query, мы автоматизируем процесс создания нужного нам столбца для всех запросов, использующих этот шаблон.

Как вы уже поняли, для определения статуса заказа нам нужно взглянуть на поле **Delivery Date**. Если заказ доставлен не был, надо это как-то отметить. Таким образом, нам понадобится дополнительный столбец **Delivery Status**.

Для загрузки данных из диапазона Excel в Power Query мы преобразуем его в таблицу, после чего на вкладке **Data** (Данные) нажмем на кнопку **From Table/Range** (Из таблицы/диапазона). Откроется редактор запросов Power Query, как показано на рис. 6.2.

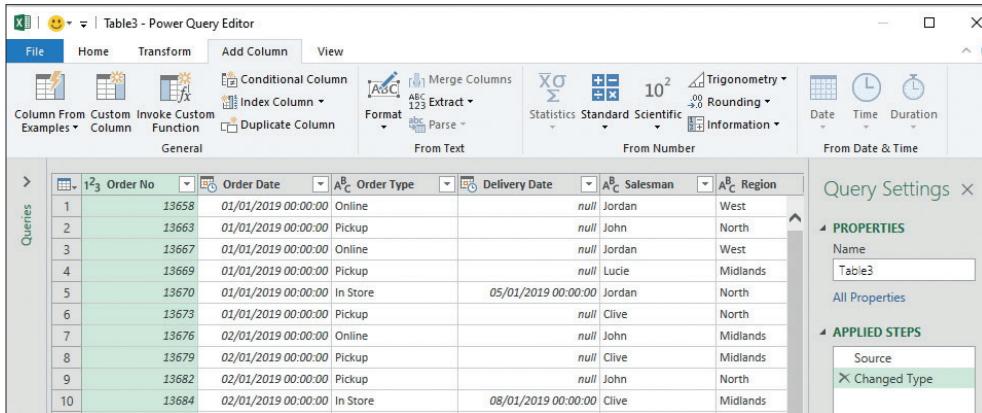


Рис. 6.2. Редактор запросов Power Query

Примечание

Загрузка данных из диапазона Excel была подробно освещена в главе 2.

В редакторе Power Query перейдите на вкладку **Add Column** (Добавить столбец) и нажмите на кнопку **Conditional Column** (Условный столбец). Откроется диалоговое окно **Add Conditional Column** (Добавление условного столбца), в котором нужно ввести данные, как показано на рис. 6.3.

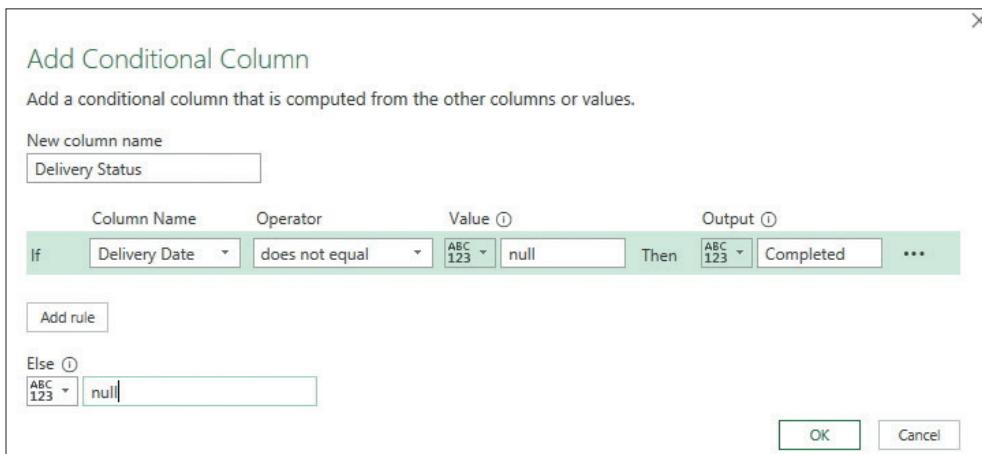


Рис. 6.3. Диалоговое окно добавления условного столбца

На представленном рисунке отчетливо видно, как работает выражение **If: If** (если) **Delivery Date does not equal** (не равно) **null**, **Then** (тогда) вернуть значение **Completed**. **If** оно равно **null**, вернуть **null**. После заполнения формы нажмите на кнопку **OK**.

Как видно на рис. 6.4, в таблице появился столбец **Delivery Status** со значениями **null** и **Completed**.

	Order No	Order Date	Order Type	Delivery Date	Delivery Status	Salesman
1	13658	01/01/2019 00:00:00	Online		null	Jordan
2	13663	01/01/2019 00:00:00	Pickup		null	John
3	13667	01/01/2019 00:00:00	Online		null	Jordan
4	13669	01/01/2019 00:00:00	Pickup		null	Lucie
5	13670	01/01/2019 00:00:00	In Store	05/01/2019 00:00:00	Completed	Jordan
6	13673	01/01/2019 00:00:00	Pickup		null	Clive
7	13676	02/01/2019 00:00:00	Online		null	John
8	13679	02/01/2019 00:00:00	Pickup		null	Clive

Рис. 6.4. Добавление столбца Delivery Status

Совет

Если вы хотите перенести новый столбец в другое место в таблице, можете сделать это путем перетаскивания его мышью, как в Excel.

Обратившись к расширенному редактору, можно увидеть сгенерированную формулу с использованием функции **If**, при этом правила составления формул в языке M несколько отличаются от Excel. Взгляните на рис. 6.5.

```

Excel.CurrentWorkbook(){[Name="Table2"]}[Content],
Type" = Table.TransformColumnTypes(Source,{{"Order No", Int64.Type}, {"Order Date", Int64.Type}, {"Order Type", type_text}, {"Delivery Date", type_text}, {"Delivery Status", type_text}},
conditional.Column" = Table.AddColumn#"Changed Type", "Delivery Status", each if [Delivery Date] <> null then "Completed" else null),
ed Columns" = Table.ReorderColumns#"Added Conditional Column",{"Order No", "Order Date", "Order Type", "Delivery Date", "Delivery Status", "Salesman"}]
```

Рис. 6.5. Код на языке M в расширенном редакторе

Как видите, создавать условный столбец при помощи редактора запросов очень просто, но иногда вам будет необходимо делать это вручную, используя расширенный редактор.

Примечание

Разницу в написании формул в Excel и Power Query легко отследить на следующем примере:

Excel: =IF(test, ValueIfTrue, ValueIfFalse)

Power Query: =if test then ValueIfTrue else ValueIfFalse

Как видите, в Power Query вам нет необходимости вводить обрамляющие скобки, зато нужно использовать ключевые слова **then** и **else**.

При этом ключевое слово **if** в Power Query требуется писать строчными буквами, а зарезервированное слово **null** означает **blank** или пустоту.

Как видно по рис. 6.4, если тип заказа стоит Pickup, его статус не будет помечен как завершенный, что неправильно. Значит, нам необходимо изменить выражение if на if or, чтобы правильно отображать статус доставки товара.

Чтобы внести эти правки, нам предстоит воспользоваться добавлением *настраиваемого столбца* (custom column) в таблицу. От условного столбца он отличается в том числе тем, что формулу нам необходимо будет писать вручную, а не с использованием среды интерфейса. Проделайте все те же действия, что и раньше, но на этот раз на вкладке **Add Column** (Добавить столбец) нажмите на кнопку **Custom column** (Настраиваемый столбец), как показано на рис. 6.6.

		Order Type	Delivery Date	Salesman	Region	Product	No. Customers	Net Sa
1	13663	01/01/2019 00:00:00	Pickup	null	Jordan	West	Dyno1	4
2	13667	01/01/2019 00:00:00	Online	null	John	North	Dyno3	8
3	13669	01/01/2019 00:00:00	Pickup	null	Jordan	West	Dyno2	4
4	13670	01/01/2019 00:00:00	In Store	05/01/2019 00:00:00	Jordan	Midlands	Dyno1	4
5	13673	01/01/2019 00:00:00	Pickup	null	Clive	North	Dyno3	9
6	13675	02/01/2019 00:00:00	Online	null	John	Midlands	Dyno2	8
7	13679	02/01/2019 00:00:00	Pickup	null	Clive	Midlands	Dyno3	10
8	13682	02/01/2019 00:00:00	Pickup	null	John	North	Dyno3	8
9	13684	02/01/2019 00:00:00	In Store	08/01/2019 00:00:00	Clive	Midlands	Dyno3	9
10	13688	03/01/2019 00:00:00	Online	null	Clive	Midlands	Dyno4	1
11	13689	03/01/2019 00:00:00	Pickup	null	Lucie	Midlands	Dyno2	10
12	13694	03/01/2019 00:00:00	Pickup	null	John	West	Dyno3	6
13	13695	03/01/2019 00:00:00	Online	null	Jordan	Midlands	Dyno3	9
14	13698	03/01/2019 00:00:00	Pickup	null	Clive	Midlands	Dyno2	10
15	13700	04/01/2019 00:00:00	Pickup	null	John	North	Dyno3	6
16	13702	04/01/2019 00:00:00	Pickup	null	Jordan	North	Dyno3	6
17								

Рис. 6.6. Создание настраиваемого столбца

В открывшемся диалоговом окне вам нужно написать код, приведенный ниже, как показано на рис. 6.7:

```
If [Delivery Date] <> null or [Order Type] = "Pickup" then "Completed" else null
```

При этом вы можете не писать названия колонок вручную, а выбирать их из списка справа.

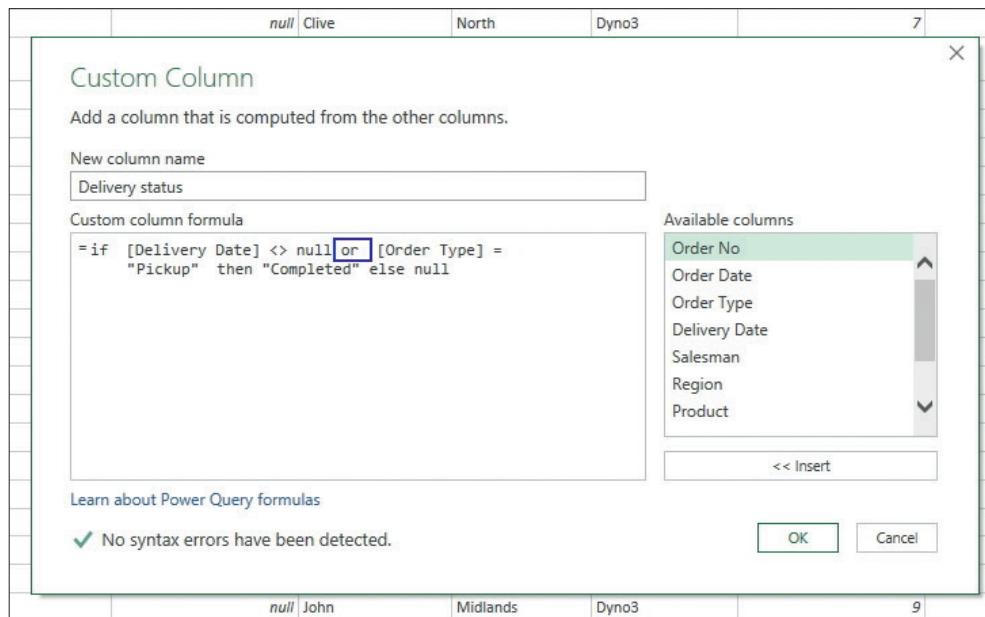


Рис. 6.7. Диалоговое окно настраиваемого столбца

Эта формула говорит о том, что статус **Completed** должен ставиться, либо если поле **Delivery Date** не пустое, либо если в поле **Order Type** стоит значение **Pickup**. В противном случае ячейка должна оставаться пустой, как показано на рис. 6.8.

	Order No	Order Date	Order Type	Delivery Date	Delivery Status	Salesman	Region	Product
1	13658	43466	Online	null	null	Jordan	West	Dyno1
2	13653	43466	Pickup	06/01/2019..	Completed	John	North	Dyno3
3	13667	43466	Online	null	null	Jordan	West	Dyno2
4	13669	43466	Pickup	04/01/2019..	Completed	Lucie	Midlands	Dyno1
5	13670	43466	In Store	05/01/2019..	Completed	Jordan	North	Dyno3
6	13673	43466	Pickup	08/01/2019..	Completed	Clive	North	Dyno3
7	13676	43467	Online	null	null	John	Midlands	Dyno2
8	13679	43467	Pickup	07/01/2019..	Completed	Clive	Midlands	Dyno3
9	13682	43467	Pickup	05/01/2019..	Completed	John	North	Dyno3
10	13684	43467	In Store	08/01/2019..	Completed	Clive	Midlands	Dyno3
11	13688	43468	Online	null	null	Clive	Midlands	Dyno4
12	13689	43468	Pickup	06/01/2019..	Completed	Lucie	Midlands	Dyno2
13	13694	43468	Pickup	08/01/2019..	Completed	John	West	Dyno3
14	13695	43468	Online	null	null	Jordan	Midlands	Dyno3
15	13698	43468	Pickup	07/01/2019..	Completed	Clive	Midlands	Dyno2
16	13700	43469	Pickup	08/01/2019..	Completed	John	North	Dyno3
17	13702	43469	Pickup	07/01/2019..	Completed	Jordan	North	Dyno3
18	13705	43469	Online	null	null	Lucie	North	Dyno3
19	13709	43469	Online	null	null	Clive	North	Dyno1
20	13712	43472	Pickup	10/01/2019..	Completed	Clive	West	Dyno3
21	13715	43472	Pickup	13/01/2019..	Completed	Clive	West	Dyno1
22	13720	43472	Pickup	11/01/2019..	Completed	Jordan	North	Dyno2
23	13725	43472	Pickup	14/01/2019..	Completed	Lucie	West	Dyno3

Рис. 6.8. Результат применения функции IF/OR

Выражение `if` позволяет вам создавать простые формулы прямо в редакторе запросов Power Query, тогда как более сложные формулы нужно писать вручную. Между формулами на языке M и написанными в Excel есть существенные различия, при этом стоит отметить, что при чтении первых гораздо лучше понятно, какие действия выполняет та или иная функция.

В данном разделе мы научились добавлять настраиваемый столбец к таблице. В следующем затронем вопросы создания параметров, позволяющих обновлять запросы, не покидая лист Excel.

СОЗДАНИЕ ТАБЛИЦЫ ПАРАМЕТРОВ ДЛЯ ЗАПРОСОВ

Создавая запросы для каких-то специфических отчетов, например для отчета о месячных продажах, мы обычно ориентируемся исключительно на нужный нам месяц. Но в следующем месяце нам может понадобиться снова воспользоваться нашим запросом, тогда как источник данных будет уже другим. Конечно, нам бы не хотелось переписывать заново весь запрос, лучше было бы просто адаптировать его к новому месяцу. Именно с этой целью в Power Query можно использовать *параметры* (parameter) – они здорово облегчают жизнь разработчикам и пользователям.

В этом разделе мы продолжим использовать исходный файл `SalesData.csv`, с которым работали в главе 4. Подключимся к нему, как показано на рис. 6.9.

	A	B	C	D	E	F	G
1	Region	Product	Date	Sales			
2	North	Game9	30/11/2019	22234.17908			
3	North	Dyno1	30/11/2019	21972.21878			
4	North	Game5	30/11/2019	21539.92663			
5	South	Game9	30/11/2019	22277.71876			
6	West	Dyno1	30/11/2019	21414.00365			
7	North	Board3	30/11/2019	21489.95132			
8	West	Game8	30/11/2019	21379.73065			
9	South	Game3	30/11/2019	22108.10035			
10	South	Game8	30/11/2019	22001.99663			
11	East	Board3	30/11/2019	21299.83249			
12	West	Board2	30/11/2019	21628.38237			
13	North	Board2	30/11/2019	21607.10233			
14	East	Game10	30/11/2019	21224.12249			
15	North	Dyno2	30/11/2019	21745.40375			
16	South	Game7	30/11/2019	22142.00196			
17	North	Board5	30/11/2019	21842.66254			
18	South	Dyno1	30/11/2019	21750.05037			
19	East	Board1	30/11/2019	21229.50994			

Рис. 6.9. Импортированный в запрос файл CSV

Здесь мы почти ничего не правили; единственное, что сделали, – это изменили тип данных в столбце `Sales` на валюту.

Щелчок по иконке с шестеренкой справа от шага **Source** (Источник) на панели **APPLIED STEPS** (Примененные шаги) приведет к открытию диалогового окна со свойствами, показанного на рис. 6.10.

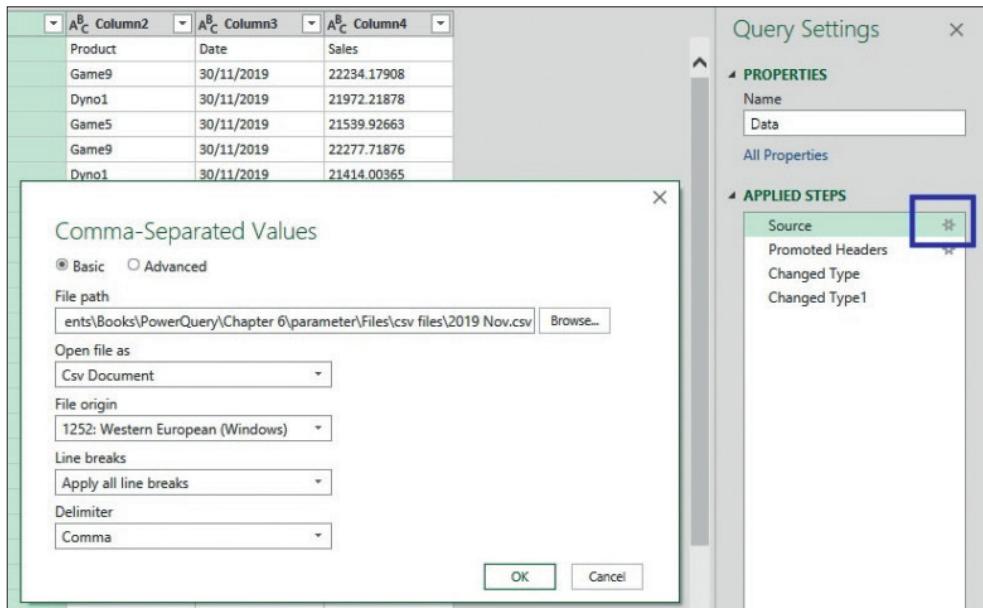


Рис. 6.10. Свойства исходного файла

Здесь вы можете видеть путь к файлу источника в поле **File path** (Путь к файлу), а также информацию о типе файла и разделителях.

В данный момент путь к файлу источника жестко прописан в нашем запросе. Это означает, что при открытии запроса он всегда будет пытаться открыть указанный файл.

Мы же *создадим параметр* с именем **FilePath**, который сможем менять каждый месяц. Для этого нажмите на выпадающую кнопку **Manage Parameters** (Управление параметрами) и выберите пункт **New Parameter** (Создать параметр), как показано на рис. 6.11.

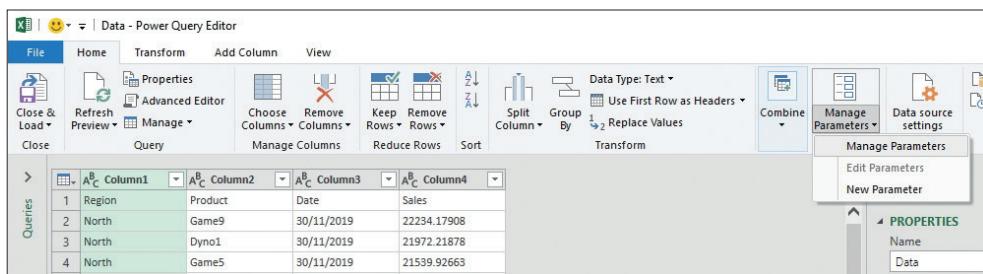


Рис. 6.11. Управление параметрами

Это приведет к открытию диалогового окна **Parameters** (Параметры), представленного на рис. 6.12.

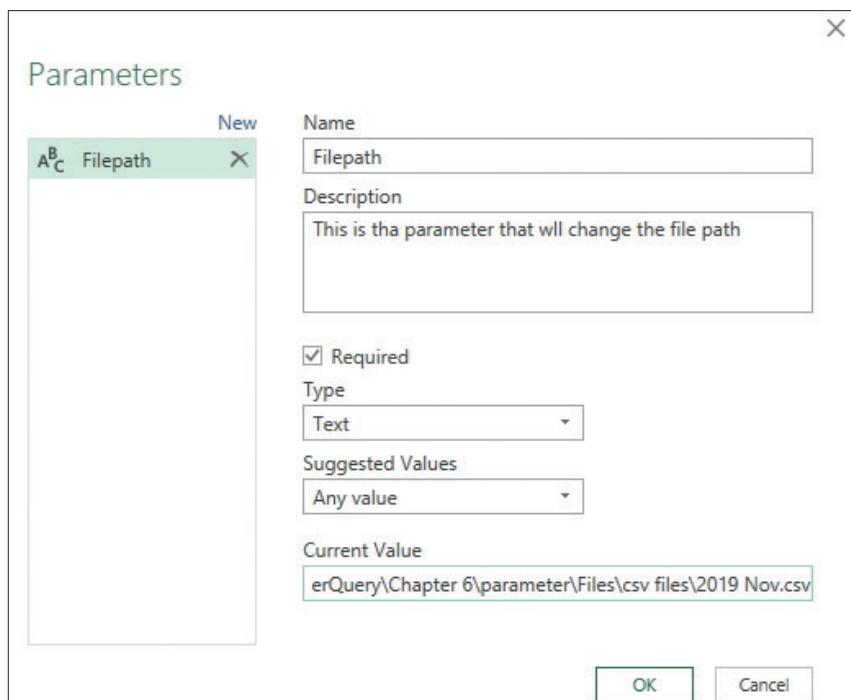


Рис. 6.12. Диалоговое окно управления параметрами

Введите в поле **Name** (Имя) название нового параметра (**FilePath**), а в выпадающем списке **Type** (Тип) укажите тип **Text** (Текст), поскольку мы будем иметь дело со строковым параметром. И наконец, в поле **Current Value** (Текущее значение) введите текущий путь к файлу источника.

Нажмите на кнопку **OK**, после чего созданный параметр появится на панели запросов в левой части экрана, как видно на рис. 6.13.

Рис. 6.13. Новый параметр **FilePath**

Щелчок мышью по запросу **Data** вновь перенесет вас в окно редактирования запроса, показанное на рис. 6.10. Снова нажмите на иконку с шестеренкой справа от шага **Source** (Источник) на панели **APPLIED STEPS** (Примененные шаги), что приведет к открытию знакомого диалогового окна со свойствами. Что мы собираемся сделать, так это заменить статический путь к файлу источника (`C:\Users\...\parameter\ Files\csv files\2019 Nov.csv`) на созданный параметр **Filepath**, как показано на рис. 6.14.

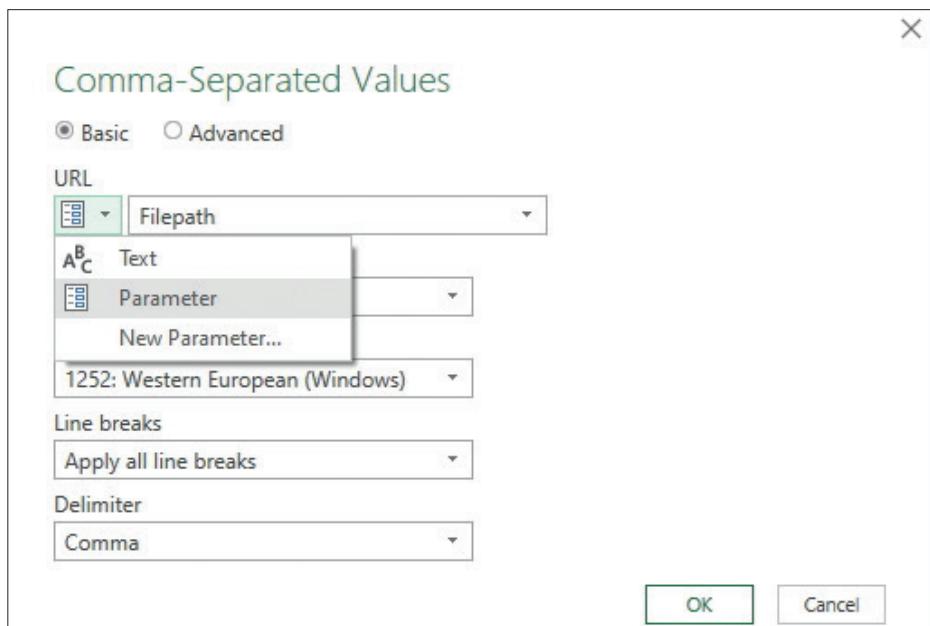


Рис. 6.14. Выбор параметра в качестве пути к файлу

Для этого достаточно на выпадающей кнопке слева от пути к файлу выбрать пункт **Parameter** (Параметр) вместо **Text** (Текст). Легче всего сразу указать вариант с параметром, а затем выбрать нужный параметр из предложенного списка. После этого вы можете нажать на кнопку **Close and Load** (Закрыть и загрузить), чтобы загрузить запрос в Excel, как показано на рис. 6.15.

The screenshot shows a Microsoft Excel interface with the 'Data' tab selected. A 'Queries & Connections' dialog box is open on the right side of the screen. The dialog box has two tabs: 'Queries' (selected) and 'Connections'. Under the 'Queries' tab, it says '2 queries' and shows a list with one item: 'Filepath (C:\Users\Sparrow\Docume... Connection only.)'. Below this, there is a note: '13,300 rows loaded.' The main Excel window displays a table with columns: Region, Product, Date, and Sales. The data includes various regions like North, South, and West, products like Game9, Dyno1, etc., dates like 30/11/2019, and sales values like 22234.1791.

	A	B	C	D	E	F
1	Region	Product	Date	Sales		
2	North	Game9	30/11/2019	22234.1791		
3	North	Dyno1	30/11/2019	21972.2188		
4	North	Game5	30/11/2019	21539.9266		
5	South	Game9	30/11/2019	22277.7188		
6	West	Dyno1	30/11/2019	21414.0036		
7	North	Board3	30/11/2019	21489.9513		
8	West	Game8	30/11/2019	21379.7306		
9	South	Game3	30/11/2019	22108.1004		
10	South	Game8	30/11/2019	22001.9966		
11	East	Board3	30/11/2019	21299.8325		
12	West	Board2	30/11/2019	21628.3824		
13	North	Board2	30/11/2019	21607.1023		
14	East	Game10	30/11/2019	21224.1225		
15	North	Dyno2	30/11/2019	21745.4038		
16	South	Game7	30/11/2019	22142.002		
17	North	Board5	30/11/2019	21842.6625		
18	South	Dyno1	30/11/2019	21750.0504		
19	East	Board1	30/11/2019	21229.5099		
20	North	Game9	30/11/2019	21946.1079		

Рис. 6.15. Запрос, подключенный через параметр Filepath

Вы увидите, что запрос по-прежнему подключен к вашему исходному файлу, но на этот раз не по прямой ссылке, а посредством параметра.

Ежемесячное изменение источника данных

Теперь представим, что наступил следующий месяц и мы хотим воспользоваться созданным в Power Query шаблоном запроса для построения отчета. Для этого нам нужно как-то указать на источник данных, соответствующий актуальному месяцу. Внесем некоторые изменения. Взгляните на рис. 6.16.

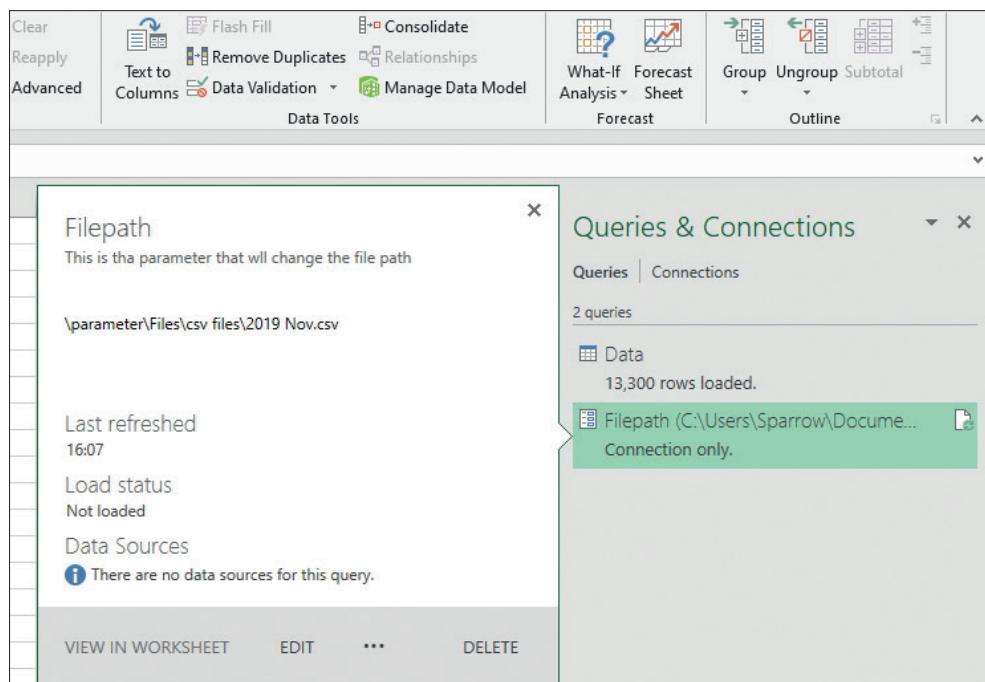


Рис. 6.16. Редактирование параметра FilePath

Сначала нажмем на кнопку **EDIT** (Изменить) во всплывающем окне, появляющемся при наведении на параметр **Filepath** на панели **Queries & Connections** (Запросы и подключения). Откроется диалоговое окно, в котором вы можете изменить значение параметра **FilePath**, воспользовавшись иконкой **Manage Parameter** (Управление параметром), как показано на рис. 6.17.

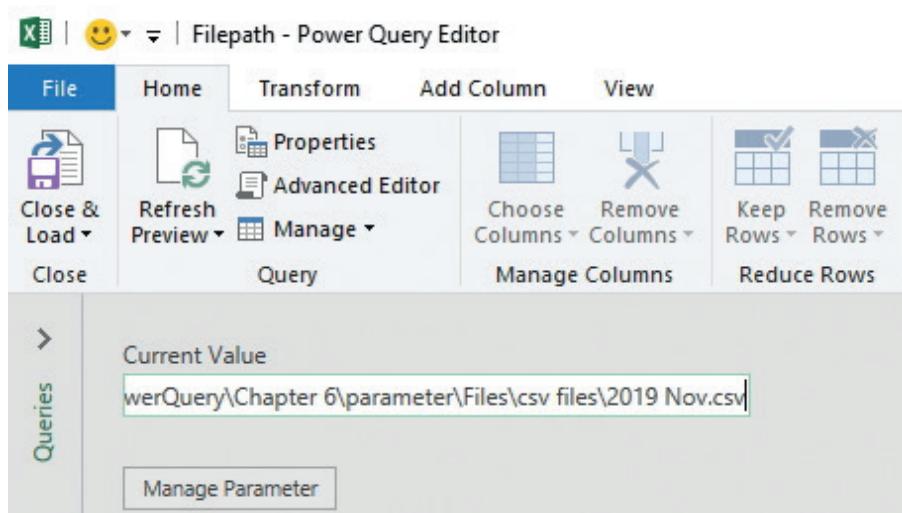


Рис. 6.17. Изменение значения параметра

После изменения значения параметра так, чтобы оно указывало на новый файл, нажмите на кнопку **Close and Load** (Закрыть и загрузить), что приведет к обновлению данных на листе Excel.

И хотя удобно иметь возможность в одной строке менять значения параметров, каждый раз для этого открывать редактор запросов Power Query может быть довольно утомительно. Чтобы этого не делать, достаточно создать так называемую *таблицу параметров* (parameter table), показанную на рис. 6.18, что серьезно упростит весь процесс. Посмотрим, как это можно сделать.

Parameter	Value
Folder	C:\Users\Sparrow\Documents\Books\PowerQuery\Chapter 6\parameter\Files\csv files\
File name	2019 Dec.csv

Region	Product	Date	Sales
North	Game9	30/11/2019	22234.1791
North	Dyno1	30/11/2019	21972.2188
North	Game5	30/11/2019	21539.9266
South	Game9	30/11/2019	22277.7188
West	Dyno1	30/11/2019	21414.0036
North	Board3	30/11/2019	21489.9513
West	Game8	30/11/2019	21379.7306
South	Game3	30/11/2019	22108.1004

Рис. 6.18. Таблица параметров

Создадим простую таблицу с именем **ParameterTable** и столбцами **Parameter** и **Value** для названий и значений параметров. Мы планируем в итоге привязать существующий запрос к нашей таблице параметров. Это позволит нам изменять значения параметров, а данные в основной таблице будут при этом обновляться автоматически. Посмотрим, как это можно реализовать.

1. Создайте новый пустой запрос. Для этого нажмите на кнопку **Get Data** (Получить данные) и выберите из меню **From Other Sources** (Из других источников) пункт **Blank Query** (Пустой запрос).

Дайте созданному запросу имя **fParameter**. Это будет имя функции, которую я собираюсь использовать для получения значений из таблицы параметров.

2. Перейдите на вкладку **View** (Просмотр) и нажмите на кнопку **Advanced Editor** (Расширенный редактор). Замените текст запроса на следующий:

```
let Parameter=(TableName, ParameterLabel) =>
let
    Source = Excel.CurrentWorkbook(){[Name=TableName]}
```

```
[Content],
    value = Source{[Parameter= ParameterLabel]}[Value]
in
    value
in Parameter
```

Это функция запроса, и если внимательно изучить код, становится ясно, что здесь есть два входных параметра **TableName** и **ParameterLabel**, при помощи которых осуществляется подключение к нашей таблице параметров. По колонке **Parameter** мы будем осуществлять поиск, а возвращать будем значение из столбца **Value**. Нажмите на кнопку **Done** (Готово).

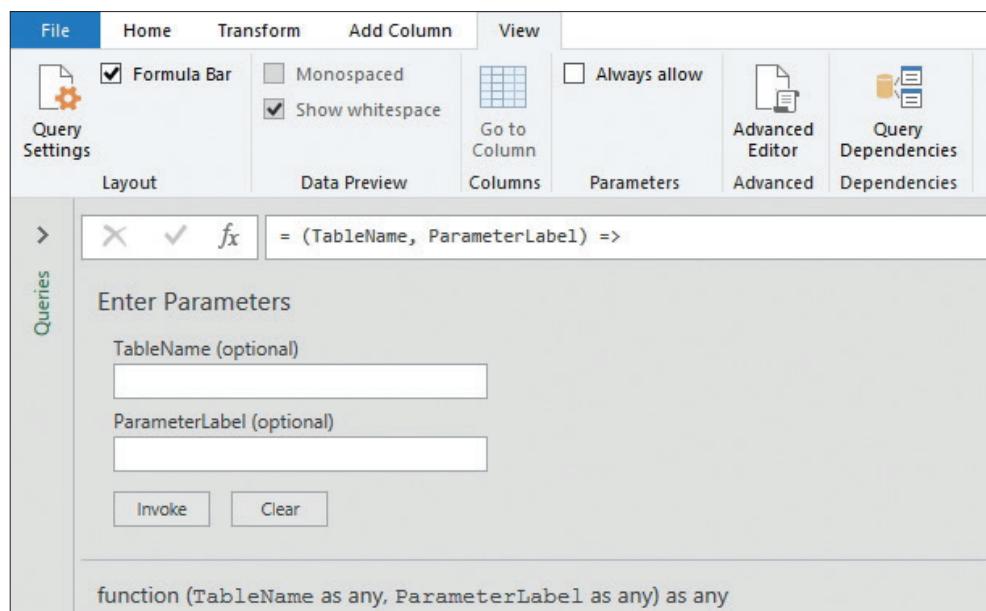


Рис. 6.19. Создание параметров

Теперь нам необходимо изменить параметр **FilePath**, созданный ранее, и написать дополнительный код, который позволит менять значения параметров в таблице и автоматически обновлять данные на листе. Выберите на соответствующей панели запрос **Data** и выделите шаг **Source** (Источник). Мы собираемся удалить параметр **FilePath** и написать дополнительный код.

The screenshot shows the Power Query Editor interface. A query named 'Data' is selected. The main area displays a table with four columns: Region, Product, Date, and Sales. The 'Query Settings' pane on the right shows a parameter named 'FilePath' with the value 'C:\Users\Sp...'. Other parameters listed are 'fParameter("Parameter", "Folder") & "\\" & fParameter("Parameter", "FileName")'.

Рис. 6.20. Изменение параметра FilePath

Удалите упоминание параметра **FilePath** в строке формул, но не удаляйте обрамляющие его скобки, а вставьте в них следующий код:

```
fParameter("Parameter", "Folder") & "\\" &
fParameter("Parameter", "FileName")
```

Здесь мы обращаемся к таблице параметров **ParameterTable**, извлекаем из нее путь к папке и имя файла и сцепляем их вместе.

После этого нажмите на кнопку **Close and Load** (Закрыть и загрузить), что перенесет вас обратно в Excel. Теперь вы сможете менять в таблице параметров путь к папке и имя файла, и данные будут автоматически загружаться в главную таблицу, как показано на рис. 6.21.

The screenshot shows an Excel spreadsheet with a table in the foreground and another table in the background. The foreground table has columns 'Parameter' and 'Value', with rows for 'Folder' (value 'C:\Users\Sp...') and 'FileName' (value '2019Dec.csv'). The background table shows sales data with columns 'Region', 'Product', 'Date', and 'Sales'.

Parameter	Value
Folder	C:\Users\Sp...
FileName	2019Dec.csv

Рис. 6.21. Изменение данных в таблице параметров

Использование таблицы параметров позволит вам быстро менять путь к исходному файлу без необходимости каждый раз править источник данных.

Стоит отметить, что таблицы параметров функционируют корректно, если исходная таблица нормально отформатирована. Однако зачастую вам придется приводить таблицы в порядок при помощи Power Query, прежде чем начать эффективно использовать таблицы параметров. В следующем разделе мы рассмотрим полезные функции `Index` и `Modulo`, которые помогут вам избавиться от лишней информации в данных.

ВВЕДЕНИЕ В ФУНКЦИИ INDEX И MODULO

В данном разделе мы познакомимся с функцией индексирования и модулярной функцией. Этот инструментарий позволит нам удалять строки и создавать новые с возможностью сортировки, что поможет в вопросах фильтрации важных для нас данных. Также мы узнаем, как переименовывать и удалять шаги, чтобы лучше разобраться в последовательности шагов и их функционировании.

Знакомство с функцией Modulo

Прежде чем знакомиться с примерами использования функции `MOD` (`OCTAT`) в Excel, полезно будет узнать, что она из себя представляет и для чего применяется. Термин *модуль* (*modulo*) используется в математике для вычисления остатка от деления одного числа на другое. Представим, что нам нужно поделить 11 на 4. Ответом будет 2 с остатком 3. Эта операция применима, например, если вам необходимо поделить студентов на группы. Использование формулы `=mod(A3, B3)` в ячейке C3 на листе, представленном на рис. 6.22, приведет к результату 1. Это означает, что при равном количестве студентов в группах один из них останется без группы.

	A	B	C	D
1				
2	Number of students in classes	Group size	Left Over	
3		35	17	1
4		27	13	1
5		22	10	2
6		29	13	3
7		26	13	0
8		24	11	2
9		23	11	1
10		19	9	1

Рис. 6.22. Пример применения функции MOD

Расширить использование функции получения остатка по модулю можно, употребляя ее совместно с функциями Sum (СУММ) и Column (СТОЛБЕЦ). Взглядите на таблицу с продажами, представленную на рис. 6.23. Допустим, нам нужно вычислить сумму продаж по всем четным месяцам.

Month	A	B	C	D	E	F	G	H	I	J	K	L	M
1	MOD	January	0	February	1	March	0	April	1	May	0	June	1
2		41	45	29	33	36	42	42	38	27	31	38	30
3		38	44	28	39	38	32	28	31	47	35	29	26
4		37	25	52	34	47	36	40	38	38	31	40	40
5		45	41	39	28	48	36	46	31	45	28	35	36
6		44	34	44	35	28	35	45	25	41	49	25	42
7		49	42	43	34	50	33	34	35	32	38	37	46
8		43	29	40	48	39	41	35	42	34	31	37	44
9		46	44	25	39	35	29	36	33	26	37	50	33
10													
11													
12					Total Sales in even months		1735						

Рис. 6.23. Таблица продаж

Формула, которую мы использовали, имеет следующий вид: $=\text{SUM}(B2:M9*(\text{MOD}(\text{COLUMN}(B2:M9), 2)=1))$ =СУММ(B2:M9*(ОСТАТ(СТОЛБЕЦ(B2:M9), 2)=1)). Функция COLUMN (СТОЛБЕЦ) вернет номер колонки (на рис. 6.23 обозначено жирным шрифтом в заголовках). В свою очередь, функция MOD (ОСТАТ) разделит номер колонки на 2. Например, в четвертом столбце остаток от деления 4 на 2 будет 0. В пятой колонке остаток будет уже 1. Если мы захотим считать сумму за каждые три месяца, то должны будем изменить в формуле 2 на 3. Таким образом, результатом всегда будет 1 или 0. В первом случае мы получим проверку на условие $1 = 1$, а во втором – на условие $0 = 1$. В итоге мы суммируем только числа из ячеек в четных месяцах.

Примечание

Вместо клавиши **Enter** по окончании ввода формулы вам необходимо нажать сочетание клавиш **Ctrl+Shift+Enter**, поскольку здесь мы имеем дело с формулой массива.

Теперь, когда вы освежили в памяти, как функция MOD работает в Excel, давайте посмотрим, как ее можно использовать в Power Query.

Зачастую исходные файлы для наших отчетов бывают отформатированы совсем не так, как нам бы того хотелось. Бывает, что на очистку файла от ненужной информации уходит уйма времени. На рис. 6.24 представлен пример файла с продажами за декабрь 2019 года.

A	B	C	D	E	F	G
1	Dyno Sales					
2	Monthly End Report					
3	Dec-19					
5	Order Number	Product Sold	Sales Person			
6	13658	Dyno1	Jordan			
7	Number of Customers	Net Sales	Profit			
8	4	984	625.28			
9	Order Number	Product Sold	Sales Person			
10	13663	Dyno3	John			
11	Number of Customers	Net Sales	Profit			
12	8	2253	653.2			
13	Order Number	Product Sold	Sales Person			
14	13682	Dyno3	John			
15	Number of Customers	Net Sales	Profit			
16	8	1528	573.6112			
17	Order Number	Product Sold	Sales Person			
18	13684	Dyno3	Clive			
19	Number of Customers	Net Sales	Profit			
20	9	2592	857.4336			
21	Order Number	Product Sold	Sales Person			

Рис. 6.24. Отчет о месячных продажах

При помощи Power Query мы можем довольно легко привести подобный отчет в вид, допустимый для анализа.

Начнем с выбора всего заполненного диапазона на этом листе, включая заголовок отчета, после чего на вкладке **Data** (Данные) в Excel нажмем кнопку **From Table/Range** (Из таблицы/диапазона). В результате откроется редактор запросов Power Query, как показано на рис. 6.25.

The screenshot shows the Power Query Editor interface with the following details:

- File**: Close & Load, Close, Refresh, Advanced Editor, Preview, Manage, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Group By, Split Columns, Sort, Data Type: Any, Use First Row as Headers, Merge Queries, Append Queries, Combine Files, Combine, Manage Parameters, Parameters, Data source settings, Data Sources, New Source, Recent Sources, New Query.
- Home**: The active tab.
- Transform**: The ribbon tab above the steps list.
- Query Settings**: Properties panel showing Name: Table1, All Properties, Applied Steps.
- Applied Steps**: A list containing a single step: "Source" followed by "Changed Type".
- Table1 - Power Query Editor**: The main workspace showing the data from Figure 6.24.

Рис. 6.25. Редактор запросов Power Query

Удалите первые четыре строки с заголовком отчета. Для этого перейдите на вкладку **Home** (Главная), выберите в выпадающей кнопке **Remove Rows** (Удалить строки) пункт **Remove Top Rows** (Удалить верхние строки) и в по-

явившемся диалоговом окне введите цифру 4, как показано на рис. 6.26. По нажатии на кнопку **OK** верхние строки будут удалены.

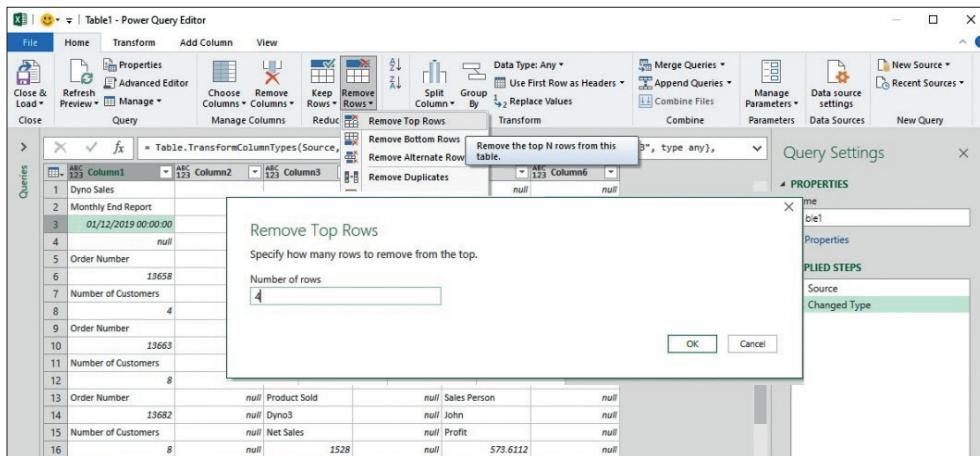


Рис. 6.26. Удаление верхних строк в Power Query

Введение в индексные функции

Одной из мощнейших особенностей Power Query является возможность использования столбца с индексом, который служит сквозным счетчиком строк в таблице. При этом нам хотелось бы использовать индекс совместно с функцией *Modulo*, чтобы можно было понимать, через сколько строк следует повторяющаяся информация.

Чтобы добавить в таблицу столбец с индексом, перейдите на вкладку **Add Column** (Добавить столбец) и нажмите на кнопку **Index Column** (Столбец индекса). При этом вы можете раскрыть кнопку создания индекса и воспользоваться дополнительными опциями, показанными на рис. 6.27: **From 0** (От 0) или **From 1** (От 1). Также вы можете выбрать пункт **Custom** (Настроить) для тонкой настройки столбца индекса.

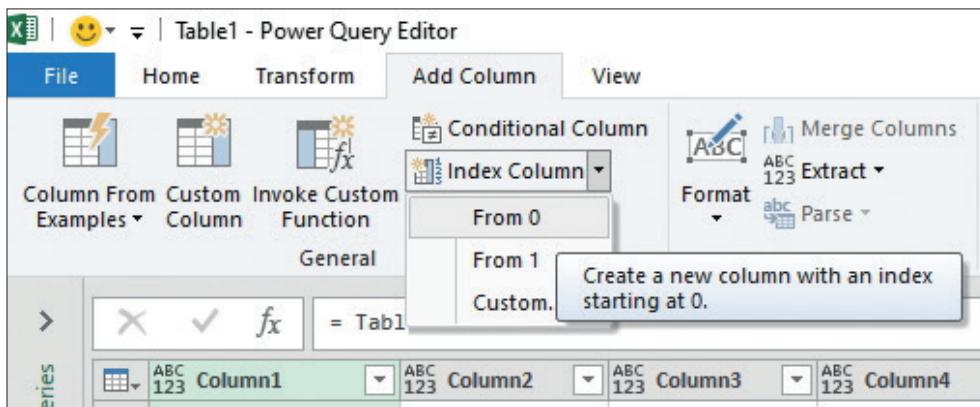


Рис. 6.27. Добавление столбца индекса

При взгляде на рис. 6.28 несложно понять, что первый номер заказа (Order Number) расположен в строке с индексом 1, а следующий – с индексом 5.

	ABC 123 Column1	ABC 123 Column2	ABC 123 Column3	ABC 123 Column4	ABC 123 Column5	ABC 123 Column6	1.2 Index
1	Order Number		null Product Sold		null Sales Person	null	0
2		13658	null Dyno1		null Jordan	null	1
3	Number of Customers		null Net Sales		null Profit	null	2
4		4	null	984	null	625.28	3
5	Order Number		null Product Sold		null Sales Person	null	4
6		13663	null Dyno3		null John	null	5
7	Number of Customers		null Net Sales		null Profit	null	6
8		8	null	2253	null	653.2	7
9	Order Number		null Product Sold		null Sales Person	null	8
10		13682	null Dyno3		null John	null	9

Рис. 6.28. Определение разрывов между номерами заказов

Именно здесь нам и пригодится функция *Modulo* (Остаток от деления) – при помощи нее мы можем сообщить Power Query, через какое количество строк пойдут повторы. В нашем случае это четыре строки, и именно эту цифру мы введем при создании столбца с остатком от деления, что видно по рис. 6.29.

Нажмите на выпадающую кнопку **Standard** (Стандартные) на вкладке **Add Column** (Добавить столбец) и выберите пункт **Modulo** (Остаток от деления).

File Home Transform Add Column View

Column From Examples Custom Column Invoke Custom Function Conditional Column Index Column Duplicate Column Format Parse Statistics Standard Scientific Trigonometry Rounding Date Time From Text From Number From Date

ABC 123 Column1 ABC 123 Column2 ABC 123 Column3 ABC 123 Column4 ABC 123 Column5 ABC 123 Column6 1.2 Index

	ABC 123 Column1	ABC 123 Column2	ABC 123 Column3	ABC 123 Column4	ABC 123 Column5	ABC 123 Column6	1.2 Index
1	Order Number		null Product Sold		null Sales Person	null	0
2		13658	null Dyno1		null Jordan	null	1
3	Number of Customers		null Net Sales		null Profit	null	2
4		4	null	984	null	625.28	3
5	Order Number		null Product Sold		null Sales Person	null	4
6		13663	null Dyno3		null John	null	5
7	Number of Customers		null Net Sales		null Profit	null	6
8		8	null	2253	null	653.2	7
9	Order Number		null Product Sold		null Sales Person	null	8
10		13682	null Dyno3		null John	null	9
11	Number of Customers		null				
12		8	null				
13	Order Number		null				
14		13684	null				
15	Number of Customers		null				
16		9	null				
17	Order Number		null				
18		13702	null				
19	Number of Customers		null				
20		6	null				
21	Order Number		null Product Sold		null Sales Person	20	
22		13705	null Dyno3		null Lucie	null	21

Рис. 6.29. Добавление столбца с остатком от деления

При взгляде на столбец с остатком от деления можно заметить, что каждая новая запись начинается со строки со значением 0. Кроме того, важно отметить, что каждое значение в столбце с модулем будет соответствовать строго определенному показателю, например количеству покупателей (поле **Number of Customers**), что видно по рис. 6.30.

	ABC 123 Column1	ABC 123 Column2	ABC 123 Column3	ABC 123 Column4	ABC 123 Column5	ABC 123 Column6	1.2 Index	1.2 Modulo
1	Order Number		null	Product Sold		null	0	0
2	13658		null	Dyno1		null	1	1
3	Number of Customers		null	Net Sales		null	2	2
4	4		null	984		625.28	3	3
5	Order Number		null	Product Sold		null	4	0
6	13663		null	Dyno3		null	5	1
7	Number of Customers		null	Net Sales		null	6	2
8	8		null	2253		653.2	7	3
9	Order Number		null	Product Sold		null	8	0
10	13682		null	Dyno3		null	9	1
11	Number of Customers		null	Net Sales		null	10	2
12	8		null	1528		573.6112	11	3
13	Order Number		null	Product Sold		null	12	0
14	13684		null	Dyno3		null	13	1
15	Number of Customers		null	Net Sales		null	14	2
16	9		null	2592		857.4336	15	3
17	Order Number		null	Product Sold		null	16	0

Рис. 6.30. Цикличность столбца с модулем

Мы будем использовать функцию IF применительно к столбцу **Modulo** с целью создания дополнительных столбцов для полей, которые нам необходимо оставить в наборе данных, например для **Order Number**, **Product Sold** и т. д.

Сначала создадим новый настраиваемый столбец, нажав на кнопку **Custom Column** (Настраиваемый столбец) на вкладке **Add Column** (Добавить столбец). Попробуем извлечь значения из поля `Order number` – это же имя введем и в поле **New column** (Имя нового столбца) диалогового окна. Теперь напишем формулу, в которой будет указано извлекать данные из столбца **Column1** только в случае, если остаток от деления в соответствующей колонке равен 1. Код получится следующим:

```
= if [Modulo]=1
then [Column1]
else null
```

После нажатия на кнопку **OK** мы увидим, что в созданном столбце заполнены только номера заказов. Сделаем то же самое для остальных столбцов, которые нам понадобятся для анализа, а именно **Product** и **Sales Person**. Формула для колонки **Product Sold** будет такой:

```
= if [Modulo]=1
then [Column3]
else null
```

В итоге получим результат, показанный на рис. 6.31.

Order Number	Product Sold	Sales Person	null	0	0	null	null	null
1	13658	Dyno1	null	1	1	13658	Dyno1	Jordan
2	13658	Dyno1	Jordan	null	2	null	null	null
3	Number of Customers	Net Sales	Profit	null	3	3	null	null
4	4	984	625.28	null	4	0	null	null
5	Order Number	Product Sold	Sales Person	null	5	1	13663	Dyno3
6	13663	Dyno3	John	null	6	2	null	null
7	Number of Customers	Net Sales	Profit	null	7	3	null	null
8	8	2253	653.2	null	8	0	null	null
9	Order Number	Product Sold	Sales Person	null	9	1	13682	Dyno3
10	13682	Dyno3	John	null	10	2	null	null
11	Number of Customers	Net Sales	Profit	null	11	3	null	null
12	8	1528	573.6112	null	12	0	null	null
13	Order Number	Product Sold	Sales Person	null	13	1	13684	Dyno3
14	13684	Dyno3	Clive	null	14	2	null	null
15	Number of Customers	Net Sales	Profit	null	15	3	null	null
16	9	2592	857.4336	null	16	0	null	null
17	Order Number	Product Sold	Sales Person	null	17	1	13702	Dyno3
18	13702	Dyno3	Jordan	null	18	2	null	null
19	Number of Customers	Net Sales	Profit	null	19	3	null	null
20	6	717	857.5448	null	20	0	null	null
21	Order Number	Product Sold	Sales Person	null	21	1	13705	Dyno3
22	13705	Dyno3	Lucie	null	22	2	null	null
23	Number of Customers	Net Sales	Profit	null	23	3	null	null
24	9	2268	903.1176	null				

Рис. 6.31. Первые три столбца

Для извлечения следующих трех колонок придется внести изменения в условную часть формулы, изменив 1 на 3, поскольку мы будем получать информацию из другой строки. К примеру, формула для извлечения данных из поля **Number of Customers** будет следующей:

```
= if [Modulo]=3
then [Column1]
else null
```

Если вы допустите ошибку при создании настраиваемых столбцов, вы всегда можете удалить ошибочный шаг и повторить его. Но легче будет щелкнуть по иконке с шестеренкой справа от нужного шага и внести необходимые изменения в открывшемся диалоговом окне **Add Conditional Column** (Настраивааемый столбец), как показано на рис. 6.32.

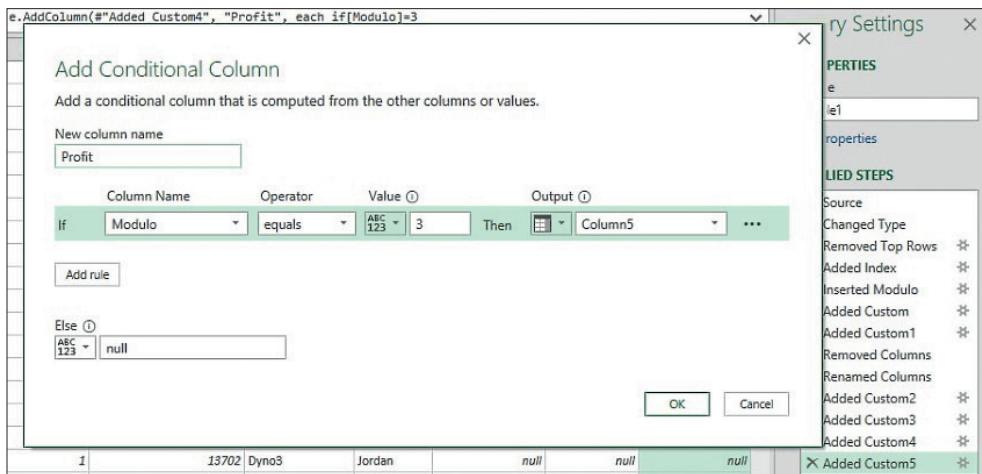


Рис. 6.32. Диалоговое окно создания и редактирования настраиваемого столбца

Завершив процесс создания настраиваемых столбцов, мы получим все нужные нам колонки, что видно по рис. 6.33.

1.2 Index	1.2 Modulo	1.2 Product Sold	1.2 Sales Person	1.2 Number of Customers	1.2 Net Sales	1.2 Profit
1	0	0	null	null	null	null
2	1	1	Jordan	null	null	null
3	2	2	null	null	null	null
4	3	3	null	null	4	984
5	4	0	null	null	null	625.28
6	5	1	13663 Dyno3	John	null	null
7	6	2	null	null	null	null
8	7	3	null	null	8	2253
9	8	0	null	null	null	653.2
10	9	1	13682 Dyno3	John	null	null
11	10	2	null	null	null	null
12	11	3	null	null	8	1528
13	12	0	null	null	null	null
14	13	1	13684 Dyno3	Clive	null	null
15	14	2	null	null	null	null
16	15	3	null	null	9	2592
17	16	0	null	null	null	857.4356
18	17	1	13702 Dyno3	Jordan	null	null

Рис. 6.33. Результирующий набор столбцов

На этом этапе в нашем распоряжении есть все необходимые для дальнейшего анализа столбцы, но данные в них разнесены по разным строкам. Осталось собрать их воедино.

Для удобства мы можем переименовать выполненные шаги на панели **APPLIED STEPS** (Примененные шаги), чтобы они носили говорящие имена, как показано на рис. 6.34.

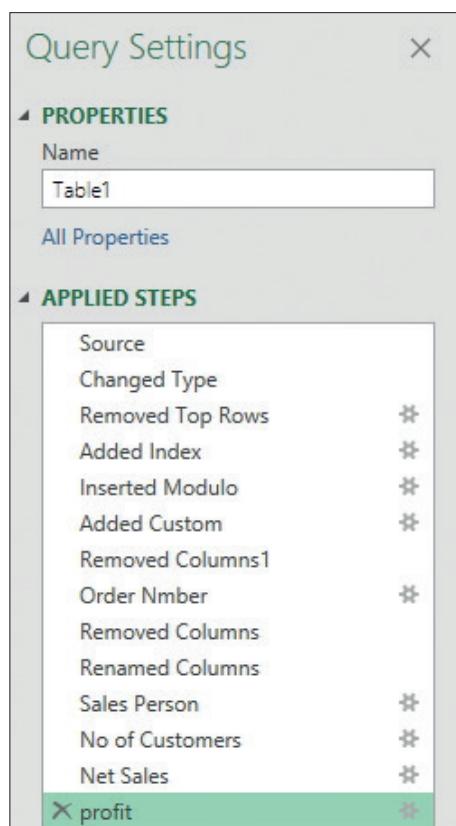


Рис. 6.34. Переименование шагов

Выделите все добавленные столбцы при помощи мыши и зажатой клавиши **Ctrl**, после чего нажмите правой кнопкой на любом из них и в подменю **Fill** (Заполнить) выберите пункт **Down** (Вниз), как показано на рис. 6.35.

Columns		Reduce Rows	Sort	Transform		Combine		Parameters		Data Sources
{#"Net Sales", "Profit", each if [Modulo]=3				Person	ABC 123 No of Customers	ABC 123 Net Sales	ABC 123 Profit			
0	ABC 123 Order Numbr	ABC	ABC	null	null	null	null			
1					null	null	null			
2					null	null	null			
3					null	4	984	625.28		
0					null	null	null	null		
1						null	null	null		
2						null	null	null		
3						8	2253	653.2		
0						null	null	null		
1						null	null	null		
2						8	1528	573.6112		
3						null	null	null		
0						null	null	null		
1						null	null	null		
2						null	null	null		
3						9	2592	857.4336		
0						null	null	null		
1	13702	Dyno3	Jordan							
2	null		null							
3	null		null							
0	null		null			6	717	857.5448		
1	13705	Dyno3	Lucie							
2	null		null							

Рис. 6.35. Меню заполнения значений в столбцах

Поскольку мы можем оставить только данные из строк, соответствующих остатку от деления, равному 3, отфильтруем нужную нам информацию, как показано на рис. 6.36.

1.2 Modulo	ABC 123 Order Number #	ABC 123 Product Sold	ABC 123 Sales
	A↓ Sort Ascending		
	Z↓ Sort Descending		
	Clear Sort		
	Clear Filter		
	Remove Empty		
	Number Filters		
	Search		
	(Select All)		
	<input type="checkbox"/> 0		
	<input type="checkbox"/> 1		
	<input type="checkbox"/> 2		
	<input checked="" type="checkbox"/> 3		
	⚠ List may be incomplete.		
	Load more		
	OK Cancel		

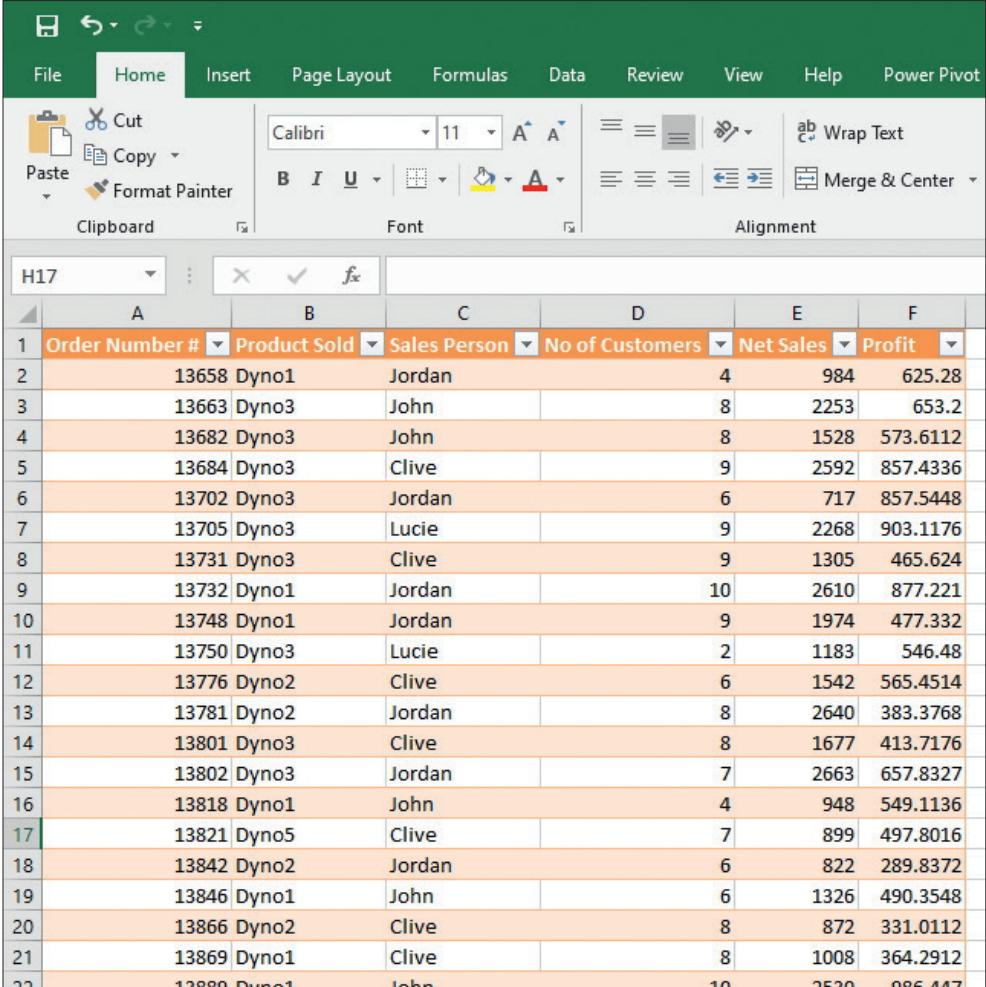
Рис. 6.36. Фильтрация столбца с остатком от деления

Теперь мы можем выбрать из всего набора данных только нужные нам столбцы. Для этого необходимо нажать на кнопку **Choose Columns** (Выбор столбцов) на вкладке **Home** (Главная) и отметить фляжками только добавленные нами колонки. Результат представлен на рис. 6.37.

The screenshot shows the Microsoft Power Query Editor interface. On the left, there's a list of queries. The main area displays a table with columns: Order Number #, Product Sold, Sales Person, No of Customers, Net Sales, and Profit. A filter bar at the top of the table says "Keep or remove columns from this table: {Select Sold", "Sales Person", "No of Customers", "Net Sales", "Profit"}". To the right of the table is the "Query Settings" pane, which includes sections for "PROPERTIES" (Name: Table1) and "APPLIED STEPS". The "APPLIED STEPS" section lists various steps taken during the query process, such as Source, Changed Type, Removed Top Rows, Added Index, Inserted Modulo, Added Custom, Removed Columns, Order Number, Removed Columns, Renamed Columns, Sales Person, No of Customers, Net Sales, profit, Filled Down, Filtered Rows, and Removed Other Column. The "Removed Other Column" step is highlighted with a green background. At the bottom of the editor, there's a status bar showing "PREVIEW DOWNLOADED AT 11:37" and a date/time stamp "01/04/2020 11:37 ENG".

Рис. 6.37. Отфильтрованные строки

После завершения работы можно нажать на кнопку **Close and Load** (Закрыть и загрузить) или **Close and Load To...** (Закрыть и загрузить в...). Теперь в нашем распоряжении простая таблица со всеми необходимыми нам данными, показанная на рис. 6.38, и мы можем легко и просто анализировать ее. Здорово и то, что в следующем месяце нам не придется повторять все эти рутинные действия – достаточно будет сменить источник данных, и все будет готово.



The screenshot shows a Microsoft Excel spreadsheet with the 'Home' tab selected in the ribbon. The table has six columns: Order Number #, Product Sold, Sales Person, No of Customers, Net Sales, and Profit. The rows are numbered from 1 to 22. The data includes various sales records for different products like Dyno1, Dyno3, and Dyno2 across different sales persons (Jordan, John, Clive, Lucie) and different numbers of customers (e.g., 4, 8, 10). The 'Net Sales' and 'Profit' columns show numerical values. Row 17 is highlighted in green, indicating it is the current row being edited.

	A	B	C	D	E	F
1	Order Number #	Product Sold	Sales Person	No of Customers	Net Sales	Profit
2	13658	Dyno1	Jordan	4	984	625.28
3	13663	Dyno3	John	8	2253	653.2
4	13682	Dyno3	John	8	1528	573.6112
5	13684	Dyno3	Clive	9	2592	857.4336
6	13702	Dyno3	Jordan	6	717	857.5448
7	13705	Dyno3	Lucie	9	2268	903.1176
8	13731	Dyno3	Clive	9	1305	465.624
9	13732	Dyno1	Jordan	10	2610	877.221
10	13748	Dyno1	Jordan	9	1974	477.332
11	13750	Dyno3	Lucie	2	1183	546.48
12	13776	Dyno2	Clive	6	1542	565.4514
13	13781	Dyno2	Jordan	8	2640	383.3768
14	13801	Dyno3	Clive	8	1677	413.7176
15	13802	Dyno3	Jordan	7	2663	657.8327
16	13818	Dyno1	John	4	948	549.1136
17	13821	Dyno5	Clive	7	899	497.8016
18	13842	Dyno2	Jordan	6	822	289.8372
19	13846	Dyno1	John	6	1326	490.3548
20	13866	Dyno2	Clive	8	872	331.0112
21	13869	Dyno1	Clive	8	1008	364.2912
22	13889	Dyno1	John	10	2530	986.447

Рис. 6.38. Окончательная таблица

Согласитесь, работать с такими данными будет гораздо легче, чем с представленными изначально. Хотя мы и потратили несколько минут на приведение информации в порядок, мы одновременно сэкономили намного больше времени на повторных ежемесячных преобразованиях данных. И в этом состоит прелесть Power Query – настроив отчет один раз, вы можете пользоваться им сколько угодно.

Приятно и то, что любую допущенную в процессе преобразования таблицы ошибку мы можем легко исправить, вернувшись к нужному шагу и отредактировав его.

До этого момента мы лишь объединяли различные файлы вместе, но нередко возникает необходимость добавления новых файлов, что не совсем одно и то же. И в следующем разделе мы научимся выполнять эту операцию.

ДОБАВЛЕНИЕ НЕСКОЛЬКИХ ФАЙЛОВ

Существует два способа для комбинирования таблиц в Power Query или Power BI: это *объединение* (merge) и *добавление* (append). В компьютерной терминологии слово добавление означает присоединение новой информации к уже существующей в таблице без перезаписывания.

При объединении таблиц мы использовали ключи для осуществления связей и создания новых столбцов. При добавлении таблиц данные присоединяются к исходным снизу.

Когда мы добавляем таблицу к другой таблице в Power Query или Power BI, мы получаем в итоге запрос, состоящий из двух и более запросов. В качестве примера давайте рассмотрим две таблицы из разных рабочих книг, представленные на рис. 6.39. Мы собираемся объединить их в один запрос.

	A	B	C	D	E	F	G	H	
1	Number	Region	Product	Date	Sales	Discount%	Discount	Total	
2	1	North	Game5	43555	41564.56	9.62%	3998.843	37565.71	
3	2	North	Game1	43555	41433.35	0.30%	122.3973	41310.95	
4	3	North	Game4	43555	41426.51	0.29%	121.8273	41304.68	
5	4	North	Game2	43555	41426.51	0.29%	121.8273	41304.68	
6	5	West	Game3	43555	41426.51	0.29%	121.8273	41304.68	
7	6	North	Dyno5	43555	9786.894	5.18%	506.7376	9280.157	
8	7	West	Game2	43555	9700.701	9.40%	911.9569	8788.745	
9	8	East	Game6	43555	9953.764	5.38%	535.4767	9418.287	
10	9	South	Game2	43555	9545.53	3.31%	316.2927	9229.237	
11	10	North	Game2	43555	9692.732	9.26%	897.4193	8795.313	
	7	6	North	Game5	43555	9616.868	1.77%	170.4682	9446.4
	8	7	North	Board4	43555	9635.518	4.79%	461.1172	9174.401
	9	8	East	Board1	43555	8933.086	4.68%	417.7248	8515.361
	10	9	East	Game9	43555	9101.107	3.50%	318.1835	8782.923
	11	10	South	Dyno4	43555	9847.687	4.73%	465.906	9381.781

Рис. 6.39. Две таблицы для добавления

При этом мы пойдем еще на шаг дальше и загрузим не только файлы Excel, но и файлы CSV. Посмотрим, как это можно сделать.

1. Откройте Power BI, загрузите запросы и выберите нужные из них для открытия. Нет никакой разницы, какие источники данных вы будете использовать при добавлении таблиц в модель данных.
2. Нажмите на выпадающую кнопку меню **Append Queries** (Добавить запросы), расположенную непосредственно под кнопкой **Merge Queries** (Объединить запросы).

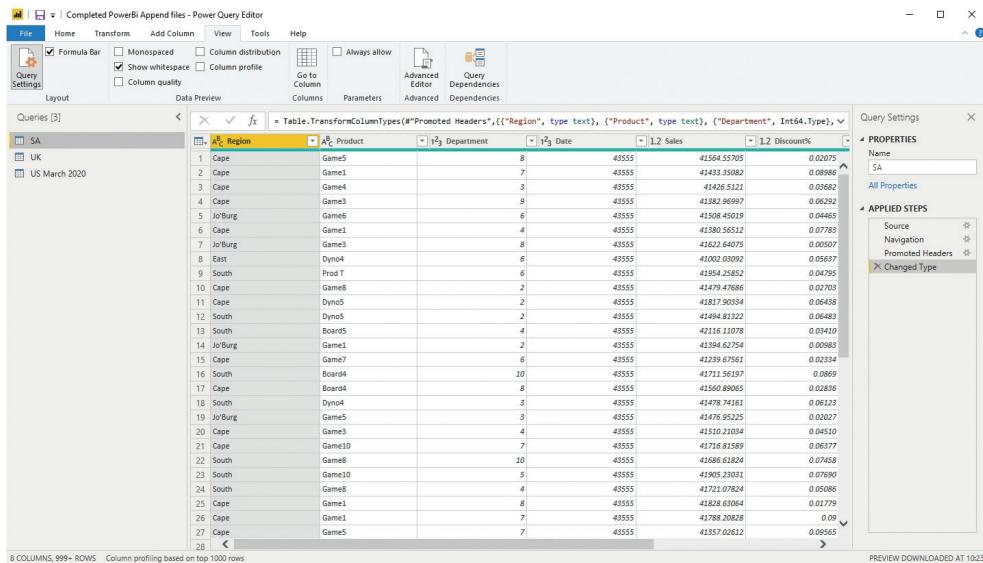


Рис. 6.40. Редактор запросов Power Query

При этом вы можете выбрать один из следующих вариантов: **Append Queries** (Добавить запросы) или **Append Queries as New** (Добавить запросы в новый). Во втором варианте исходные запросы не изменяются, а объединение выполняется за счет создания нового запроса. Мы выберем именно его.

3. Появится диалоговое окно **Append** (Добавление), показанное на рис. 6.41.

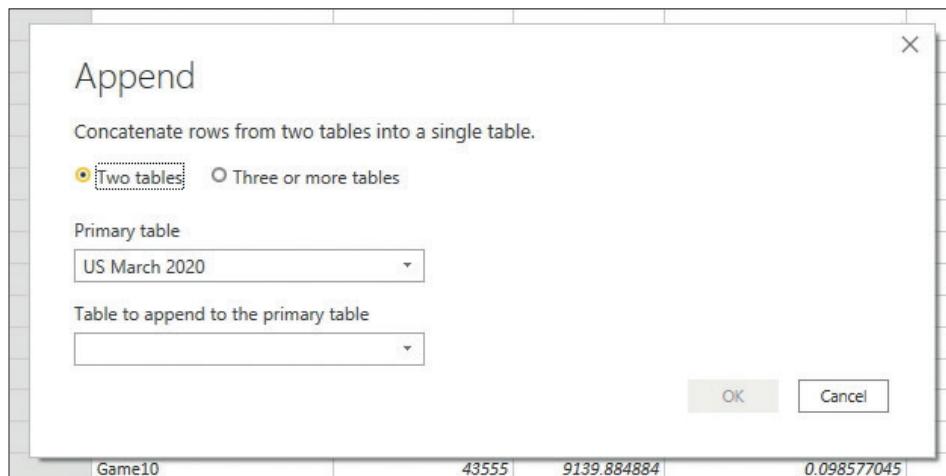


Рис. 6.41. Диалоговое окно добавления запросов

Далее вам нужно выбрать действие в зависимости от того, сколько запросов вы хотите добавить. Если запроса два, вам необходимо выбрать в выпадающем списке первичную таблицу и таблицу, которую вы хотите добавить к первичной.

Первичной обычно является таблица или запрос, который вы выбрали первым, но это также может быть главная таблица, данные из которой должны оказаться наверху. Взгляните на рис. 6.42.

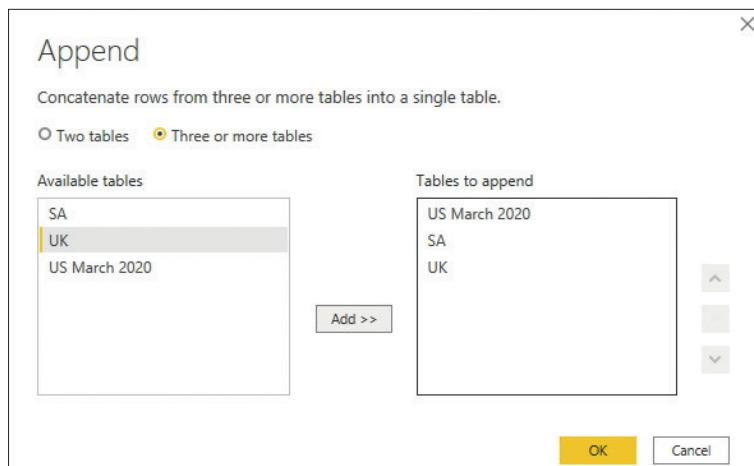


Рис. 6.42. Выбор таблиц для добавления

Как видите, мы собираемся добавить три разные таблицы в запрос, для чего установили переключатель в положение **Three or more tables** (Три таблицы или больше). Выберите нужные таблицы в списке слева и нажмите на кнопку **Add** (Добавить) для переноса их в правую часть.

Рис. 6.43. Итоговый запрос

В итоговом запросе с именем **Append1** (Добавить1) будут объединены исходные запросы строка за строкой, а поскольку колонки в них были одинаковые, в результате мы получили столько же столбцов, как в объединяемых запросах. При этом в одном из исходных запросов (**SA**) был дополнительный столбец **Department**, который также попал в итоговую таблицу со значениями **null** в строках, соответствующих запросам, где он отсутствовал. На рис. 6.44 видно, что в столбце **Department** содержится много пустых значений.

	Sales	Discount%	Discount	Total	Department
1	43555	9786.894289	0.051777159	506.7375842	9280.156705
2	43555	9700.701419	0.094009379	911.9569119	8788.744507
3	43555	9953.764031	0.053796408	535.4767476	9418.287283
4	43555	9545.525987	0.033135164	316.2926982	9229.237172
5	43555	9692.732076	0.092586829	897.4193268	8795.312749
6	43555	9616.868165	0.01772596	170.4682223	9446.399943
7	43555	9635.517974	0.047855987	461.1172276	9174.400746
8	43555	8933.08625	0.04676153	417.7247814	8515.361469
9	43555	9101.10663	0.034960974	318.1835486	8782.923081
10	43555	9847.686509	0.047311214	465.9060033	9381.780506
11	43555	9549.603971	0.024078375	229.9389447	9319.665026
12	43555	9087.243995	0.009466945	86.02843776	9001.215557
13	43555	9348.288541	0.011666993	102.0664155	9239.222126
14	43555	9003.088959	0.023524429	223.554745	9279.534214
15	43555	9857.074395	0.051638565	509.0051817	9348.069213
16	43555	9849.032373	0.08663223	853.2436411	8995.788732

Рис. 6.44. Столбец **Department** со значениями **null**

У нас есть возможность удалить этот столбец или выполнить объединение с другими запросами. Операция добавления запросов не удаляет дублирующиеся строки, так что для этого нам придется воспользоваться пунктами **Group By...** (Группировать по...) или **Remove Duplicates** (Удалить дубликаты) из контекстного меню, как показано на рис. 6.45.

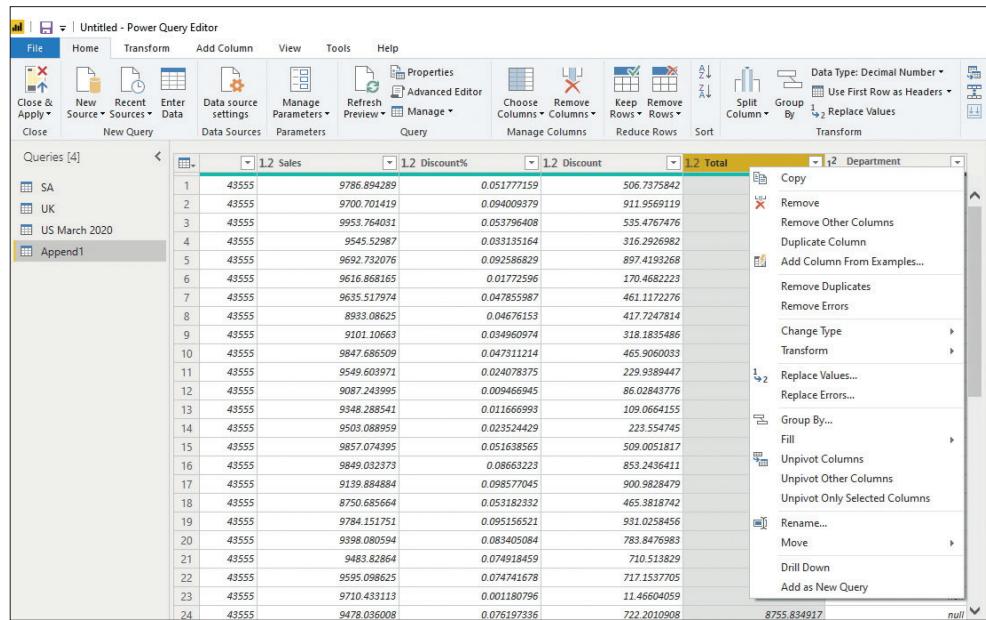


Рис. 6.45. Удаление дубликатов при помощи контекстного меню

Примечание

В Power Query и Power BI комбинирование таблиц выполняется на основании имен столбцов. При этом первая выбранная таблица воспринимается как шаблон для остальных таблиц. Если в других таблицах будет больше колонок, чем в шаблоне, они будут проигнорированы, а если меньше, Power Query и Power BI автоматически заполнят пустующие строки значениями **null**.

Стоит отметить, что столбцы в добавляемых таблицах не обязательно должны следовать в одном и том же порядке, поскольку в Power Query и Power BI для комбинирования запросов используются исключительно имена колонок, а не их порядок.

Теперь можно нажать на кнопку **Close and Apply** (Закрыть и применить), чтобы данные для анализа загрузились в Power BI, как показано на рис. 6.46.

TABLE: Appendix1 (13,603 rows, 1,273 filtered rows) COLUMN: Region (4 distinct values, 4 filtered distinct values)

UPDATE AVAILABLE (CLICK TO DOWNLOAD)

Рис. 4.46. Итоговый запрос

При добавлении запросов Power BI и Power Query выполняют поиск столбцов с полностью совпадающими именами. В случае успешного нахождения таких столбцов происходит добавление всех строк из источников в эту колонку. Если будут обнаружены несовпадающие колонки, как в случае с **Department**, Power BI создаст новый столбец.

При этом необходимо проявлять осторожность в случае, если в запросах есть столбцы, имена которых внешне выглядят похожими, но на самом деле отличаются. Например, колонки **SalesPerson** и **Sales Person** будут считаться различными. Также стоит помнить, что имена столбцов чувствительны к регистру.

Итак, вы научились добавлять таблицы из нескольких файлов. Теперь посмотрим, как добавить данные из одного файла, находящиеся на разных вкладках.

ДОБАВЛЕНИЕ ДАННЫХ ИЗ НЕСКОЛЬКИХ ВКЛАДОК

В этом разделе вы узнаете, как добавить в запрос данные сразу с нескольких вкладок одной рабочей книги. Это можно сделать разными способами, и в своем варианте мы будем использовать код на языке M.

Для начала откроем рабочую книгу **sample1.xlsx**, в которой хранится информация о продажах в разных регионах Южной Африки. При этом каждому региону (Cape Town, Durban и т. д.) отведена своя вкладка в рабочей книге.

1. Создайте пустой запрос, как вы уже делали ранее. Для этого нажмите на кнопку **Get Data** (Получить данные) и выберите из меню **From Other Sources** (Из других источников) пункт **Blank Query** (Пустой запрос).

2. В строке формул введите `=Excel.CurrentWorkbook()`, после чего будут показаны все листы в текущей рабочей книге, что видно по рис. 6.47. Пожалуйста, помните о том, что строка формул в Power Query чувствительна к регистру.

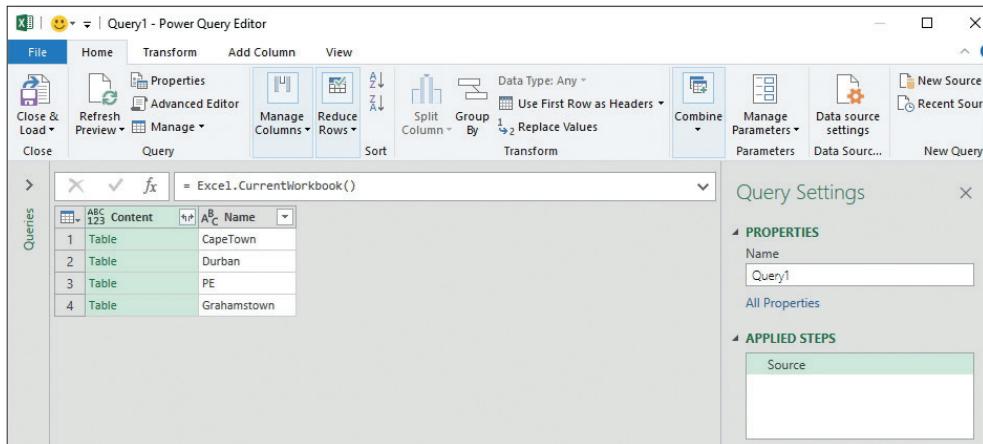


Рис. 6.47. Вкладки рабочей книги

Эта формула сканирует текущую рабочую книгу на предмет листов, таблиц, именованных диапазонов и подключений, которые и отображаются. Как видите, с каждым объектом здесь сопоставляется целая таблица.

3. Чтобы открыть таблицы, необходимо щелкнуть на слове **Table** в каждой строке. Результат показан на рис. 6.48.

Рис. 6.48. Расширенная таблица

4. Чтобы вернуться к исходному списку, необходимо удалить два последних шага на панели **APPLIED STEPS** (Примененные шаги) в правой части экрана.

Power Query хранит все эти таблицы в модели данных.

The screenshot shows the Power Query Editor interface. The title bar reads "Query1 - Power Query Editor". The ribbon has tabs: File, Home, Transform, and Add Column. Under the Home tab, there are buttons for Close & Load (dropdown), Refresh, Preview (dropdown), Manage (dropdown), and Manage Columns (dropdown). The main area is titled "Query" and contains a "Queries" list. The list shows four entries, each represented by a small icon and the text "Table". To the right of the list is a preview pane showing a table with two columns: "Content" and "Name". The "Content" column has four rows labeled 1, 2, 3, and 4. The "Name" column has four corresponding entries: "CapeTown", "Durban", "PE", and "Grahamstown".

	Content	Name
1	Table	CapeTown
2	Table	Durban
3	Table	PE
4	Table	Grahamstown

Рис. 6.49. Исходные таблицы

5. Далее необходимо нажать на кнопку с двумя стрелками в заголовке столбца **Content**, что позволит выбрать нужные нам столбцы, как показано на рис. 6.50.

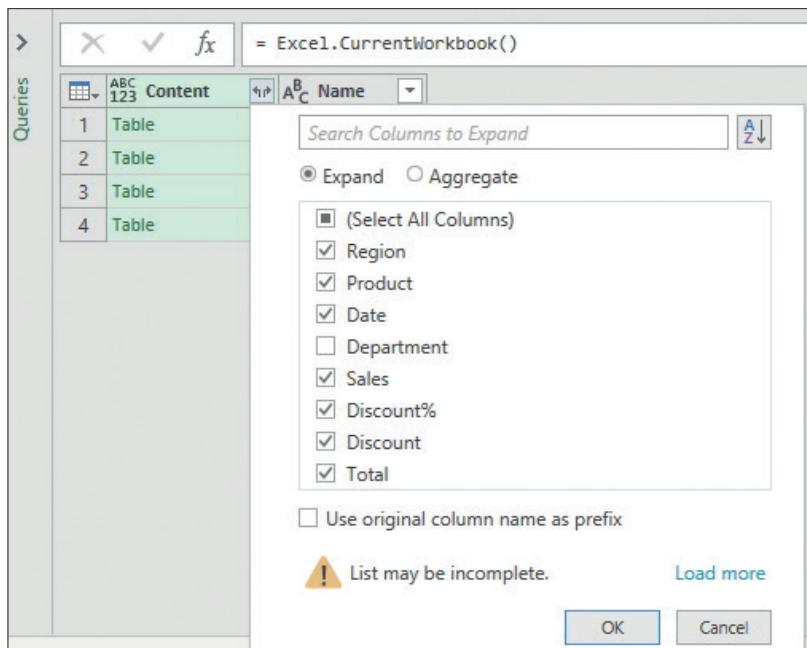


Рис. 6.50. Выбор колонок для объединения

6. Снимем флажок с поля **Department**, поскольку оно используется только в регионе Cape Town, и отключим галку **Use original column name as prefix** (Использовать исходное имя столбца как префикс).
7. Это необходимо сделать по причине того, что Power Query по умолчанию будет добавлять имя столбца **content** ко всем названиям столбцов. Например, колонка **Region** получит имя **content.Region**. После этого нажмите на кнопку **OK**.
8. Пройдитесь по листу и проверьте, что все типы данных выбраны корректно, после чего нажмите на кнопку **Close and Load** (Закрыть и загрузить). Это приведет к созданию нового листа в существующей рабочей книге.

Также мы создадим еще один лист с названием **Pretoria** и добавим его в нашу рабочую книгу. Обновление данных в запросе приводит к автоматическому обновлению всех данных, что обеспечивает нам наличие самой актуальной информации. В данном разделе мы научились добавлять и объединять данные из разных источников, а также увидели, что при изменении источника данных Power Query автоматически создает новый запрос на основании новых файлов в папке или новых вкладок в рабочей книге Excel.

Есть люди, предпочитающие работать со множеством рабочих книг одновременно, я же стараюсь всю необходимую мне информацию держать на разных вкладках одной рабочей книги. Этот вариант мне кажется наиболее удобным с точки зрения поддержания актуальности данных.

ЗАКЛЮЧЕНИЕ

В данной главе мы познакомились с полезными функциями **IF**, **Index** и **Modulo**, а также научились создавать таблицы параметров и объединять и добавлять файлы и вкладки.

Как вы увидели, использование функции **IF** в Power Query не так сильно отличается от Excel, и здесь тоже есть возможность осуществлять ветвление, как было показано на примере выбора доставленных товаров с использованием оператора **OR**.

Далее мы рассмотрели создание таблицы параметров, которая может оказаться очень полезной при построении ежемесячных отчетов на основании одного шаблона с периодической сменой источника данных. Разместив таблицу параметров прямо на листе Excel, вы получаете возможность менять источник данных, не покидая отчета, что здорово экономит время.

Функции **Index** и **Modulo**, с которыми мы познакомились в дальнейшем, могут пригодиться в случае, если вам приходится работать с файлами неудобного формата или содержащими данные не в колонках. С использованием упомянутых функций можно легко построить столбцы индексов по определенным правилам, что поможет извлекать разнородную информацию из таблиц самого разного формата.

В заключительной части главы мы познакомились с различными вариантами объединения и добавления данных из разных файлов, вкладок и таблиц, что позволяет поддерживать источники данных в актуальном состоянии.

На протяжении данной главы мы много раз обращались к коду на языке M и отметили несколько важных отличий между ним и языком формул Excel. Одним из главных отличий является то, что в языке M обращение к столбцам осуществляется посредством квадратных скобок, тогда как в Excel по большей части функции применяются с круглыми скобками. Эту особенность двух языков мы подробнее разберем в главе 9.

В будущих главах мы еще не раз коснемся важных формул, функций и советов относительно использования языка M. Следующая же глава будет целиком посвящена автоматизации отчетов, где очень важно будет следить за тем, что все обновления выполняются четко и без проблем. Также мы научимся разбираться в исходном коде автоматизации и искать ошибки, приводящие к сбоям.

Глава 7

Автоматизация отчетов в Power Query

Power Query позволяет упростить и автоматизировать процесс формирования отчетов из различных источников данных. В данной главе мы рассмотрим три режима хранения (storage mode) данных и опишем преимущества и недостатки каждого из них. Также мы создадим отчет на основании нескольких файлов, который будет ежемесячно обновляться.

Основные темы этой главы:

- режимы хранения и типы наборов данных;
- выбор режима хранения **Import**;
- типы обновлений Power BI.

К концу главы вы будете хорошо понимать, что из себя представляет процесс автоматизации отчетов, и осознаете важность связки Excel с Power Query.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Мы будем исходить из предположения о том, что вы обладаете базовыми знаниями о среде Windows и, в частности, умеете обращаться с *Диспетчером задач* (Task Manager). Кроме того, вы должны уметь загружать различные источники данных в Power BI и просматривать их, пользуясь элементами интерфейса.

Вам понадобится интернет-соединение для загрузки соответствующих файлов с GitHub. Файлы с кодом для этой главы можно скачать по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

Видеофрагмент, посвященный данной главе, располагается по адресу: https://www.youtube.com/watch?v=KbDnA3WPjQ8&list=PLLeLcvrwLe186O_GJEzs47WaZXwZjwTN83&index=8&t=0s.

ВВЕДЕНИЕ В РЕЖИМЫ ХРАНЕНИЯ И ТИПЫ НАБОРОВ ДАННЫХ

В Power BI учтены различные требования к обновлению данных при помощи режимов хранения и типов наборов данных. Обычно, чтобы увидеть изменения, произошедшие в источнике, вам необходимо заново импортировать данные в Power BI. Однако существует и режим обращения к данным в источнике напрямую.

При использовании инструмента Power BI Desktop бывает нeliшним иметь возможность управлять режимом хранения данных, чтобы понимать, как именно ваши данные кешируются в памяти при формировании тех или иных отчетов. Рассмотрим определенные сценарии, в которых манипулирование режимами хранения информации может быть преимуществом:

- осуществление контроля за снижением времени обновления с целью выполнения кеширования только тех данных, которые больше всего в этом нуждаются с точки зрения требований бизнеса;
- при правильной стратегии кеширования данных возрастает скорость выполнения запросов в Power BI, что особенно важно при осуществлении доступа к отчетам одновременно с выполнением запросов DAX над наборами данных;
- при работе с большими источниками данных вы можете четко указать, какие таблицы должны кешироваться, а какие нет, что позволит не использовать память без особой необходимости;
- пользователь может вносить изменения, которые будут отображаться непосредственно в некешированных таблицах.

Теперь, когда вы понимаете, как изменение режима хранения информации может влиять на скорость обновления данных, посмотрим, как он, собственно, устанавливается.

Изменение режима хранения информации в Power BI Desktop

Давайте найдем в Power BI настройку режима хранения. Он может быть установлен отдельно для каждой таблицы в модели данных Power BI. Эта настройка используется для контроля кеширования таблиц в модели данных.

1. Щелкните на левой панели по иконке отображения модели данных, как показано на рис. 7.1, и выберите таблицу, для которой хотите изменить режим хранения.

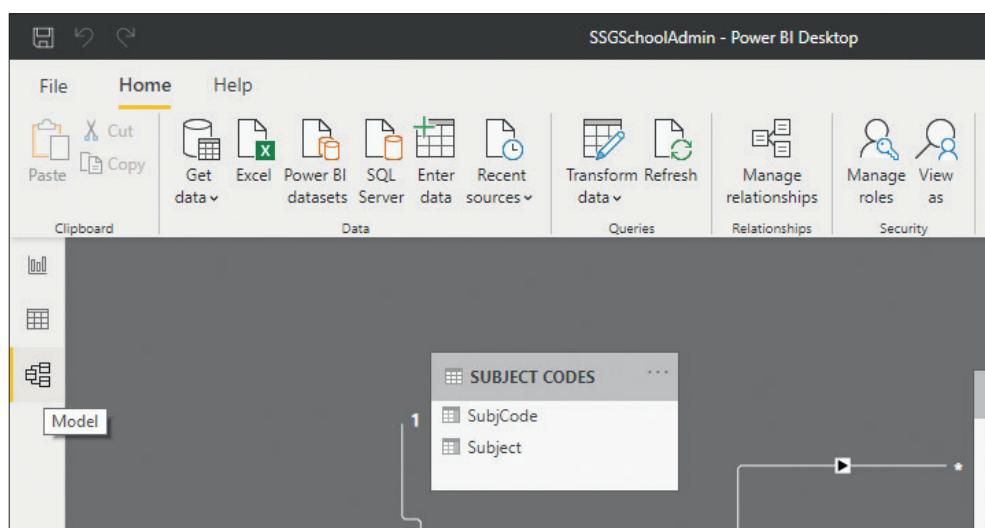


Рис. 7.1. Выбор таблицы в модели данных Power BI

2. Справа от модели данных появится панель **Properties** (Свойства), в которой вам необходимо раскрыть область **Advanced** (Дополнительно), расположенную в нижней части окна.
3. В выпадающем списке **Storage mode** (Режим хранения) выберите требуемый режим для таблицы.

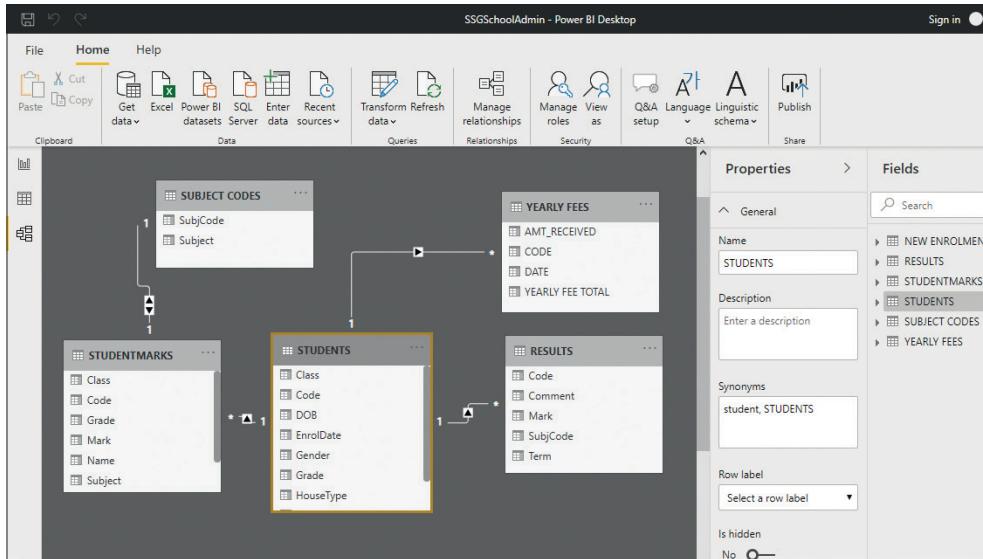


Рис. 7.2. Панель свойств таблицы

4. Вам будет предоставлено на выбор три варианта хранения данных: **Import** (Импорт), **DirectQuery** и **Dual** (Двойной), как показано на рис. 7.3.

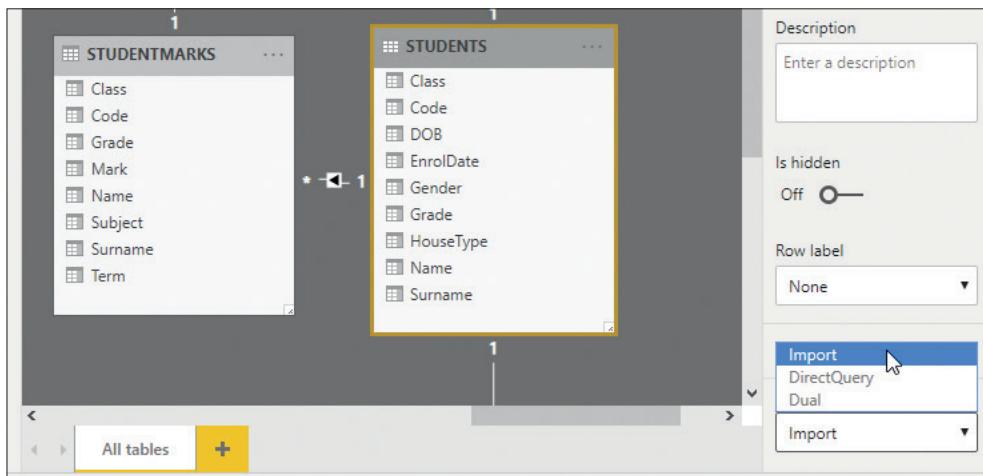


Рис. 7.3. Выбор режима хранения данных

Присмотримся к каждому из режимов внимательнее:

Import. В этом режиме все импортируемые таблицы кешируются и физически загружаются в модель данных. Это гарантирует, что данные будут возвращены посредством импорта таблиц при выполнении запроса к набору данных из Power BI. Режим импорта – единственный тип подключения, доступный вне зависимости от типа источника данных. Заметьте, что, выбрав эту опцию, вы не сможете отменить свое действие. Имеется в виду, что вам будут недоступны другие режимы хранения данных в выпадающем списке после выбора режима импорта. На рис. 7.3 отчетливо видно, что пункты **DirectQuery** и **Dual** выбрать невозможно. В данном случае причина этого в том, что к файлам Excel применим только режим импорта, а остальные режимы недоступны в принципе;

DirectQuery. В этом случае таблицы не кешируются, и каждый запрос к данным, выполненный в Power BI, транслируется в язык запросов, характерный для конкретного источника. Этот режим может быть более медленным. Плюсом же является то, что источник данных не нуждается в обновлении. В то же время при использовании этого режима хранения данных вы будете ограничены по части применения языка DAX в плане моделирования данных и применения преобразований в Power Query;

Dual. Это максимально гибкий вариант, который позволяет применять кеширование или отказаться от него, чтобы можно было в полной мере воспользоваться преимуществами заранее сохраненной информации и при необходимости обратиться к данным в конкретном источнике с использованием понятного ему языка запросов. Одним из таких языков может быть SQL. Этот режим доступен в Power BI для приложения Desktop и для онлайн-службы Power BI Service. Двойной режим может быть использован с целью повышения производительности и ограничения количества слабых связей, возникающих в модели данных при наличии таблиц **DirectQuery**. Модели данных, в которых насчитывается больше одного режима подключения, называются *составными* (composite model). В составной модели может присутствовать смесь режимов, например **DirectQuery** и **Import**. В этом случае данные, хранящиеся в режиме **DirectQuery**, не будут физически загружаться в Power BI, обращение к ним будет осуществляться напрямую при помощи языка SQL, а таблицы Excel будут импортированы и потребуют обновления при каждом обращении.

Здесь стоит отметить, что, говоря о режиме хранения, мы имеем в виду тип подключения, который устанавливаем с источником данных. Можно импортировать разные данные в Power BI, после чего создать подключения с источниками с использованием режимов **Import**, **DirectQuery** или **Dual**. При этом одни подключения могут поддерживать более одного режима хранения, другие – нет.

Примечание

Выбор режима хранения информации – очень важный шаг, и нужно с самого начала определиться с тем, какой тип подключения будет использован. Это позволит избежать ненужных хлопот с повторным импортом и установкой подключения по ходу работы. В Power BI режим хранения данных не может быть изменен «на лету».

В данном разделе вы познакомились с существующими режимами хранения информации и узнали, как устанавливать подключения. Далее мы посмотрим, как можно определить тип подключения, примененный к каждой таблице в наборе данных.

Быстрый просмотр режима хранения

Режим хранения для каждого импортированного элемента можно посмотреть на панели **Fields** (Поля), расположенной в правой части интерфейса Power BI. Для этого нужно сделать следующее:

1. Откройте панель **Fields** (Поля).
2. Наведите мышь на любой импортированный элемент.
3. Вы увидите всплывающее окошко слева от элемента, в котором будет содержаться информация об импорте и режиме хранения элемента, как показано на рис. 7.4.

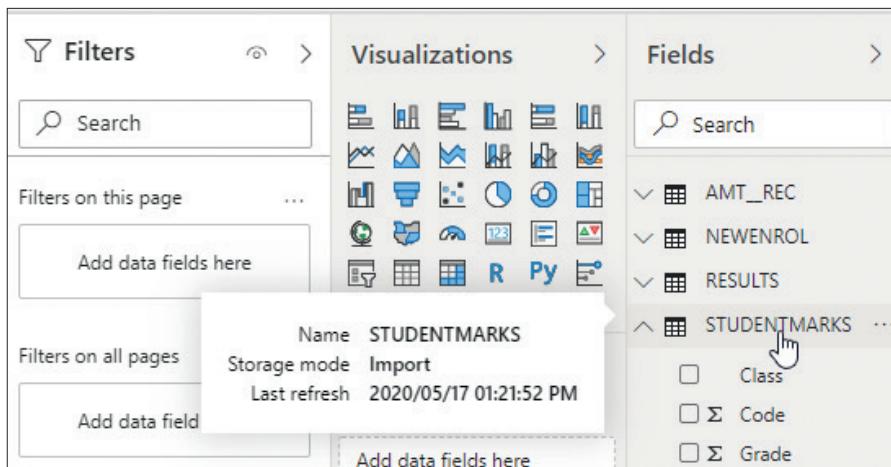


Рис. 7.4. Наведите мышью на элемент для просмотра его режима хранения

Вы узнали, как можно быстро просматривать режим хранения элемента в Power BI. В следующем разделе поговорим о том, как устанавливать режим хранения **Import** (Импорт).

УСТАНОВКА РЕЖИМА ХРАНЕНИЯ IMPORT

На протяжении книги мы импортировали в Power BI далеко не один набор данных, но при этом ни разу подробно не описывали этот процесс. Давайте освежим в памяти последовательность действий при загрузке таблиц, уделив особое внимание установке режима хранения.

1. Откройте Power BI Desktop и на вкладке **Home** (Главная страница) нажмите на кнопку **Get Data** (Получить данные), выбрав пункт Excel.
2. Откройте файл для импорта **SSGSchool.xlsx**.

3. Выберите таблицы, которые хотите импортировать. Мы выберем по одной все таблицы из списка, установив напротив них флажки, как показано на рис. 7.5.

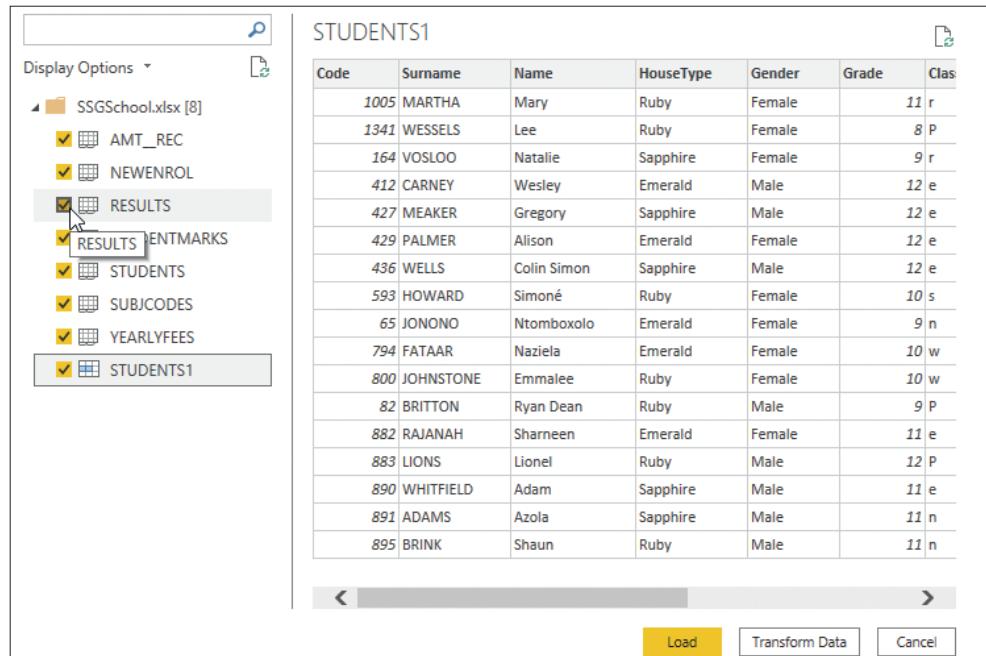


Рис. 7.5. Выбор таблиц для импорта

4. Нажмите на кнопку **Load** (Загрузить) для импорта набора данных.
 5. В процессе загрузки вы увидите сопроводительное окно, показанное на рис. 7.6.

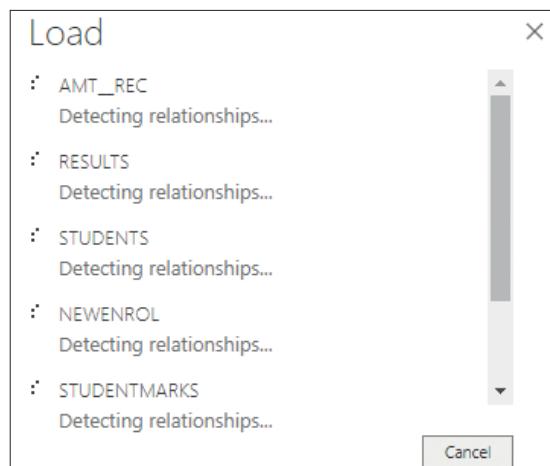


Рис. 7.6. Окно загрузки данных

6. Если вы используете подключение к источнику данных SQL, появится окно, представленное на рис. 7.7, в котором переключатель **Data Connectivity mode** (Режим подключения к данным) по умолчанию будет установлен в положение **Import** (Импорт).

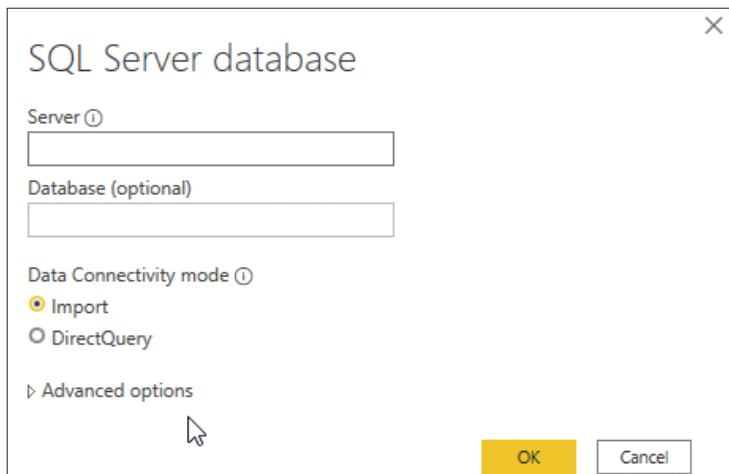


Рис. 7.7. Опции импорта при загрузке данных из SQL Server

Если установить переключатель в положение **DirectQuery**, таблицы не будут физически импортироваться в Power BI. В целом процесс загрузки данных в этом случае мало чем будет отличаться, разве что в режиме **DirectQuery** после подключения к базе данных вы увидите таблицы, к которым у вас будет доступ. При этом создано будет лишь подключение к данным, обновление которых будет производиться путем выполнения запросов напрямую к источнику. Поскольку в нашем примере мы загружаем информацию из Excel, единственным доступным режимом хранения будет **Import** (Импорт).

7. Нажмите на кнопку **OK** для выполнения загрузки данных.
8. После выбора режима подключения к данным появится левая навигационная панель с тремя иконками: **Report** (Отчет), **Data** (Данные) и **Model** (Модель). Если подключение к данным выполняется в режиме **DirectQuery**, вкладка **Data** (Данные) будет неактивной, поскольку данные в нашей модели храниться не будут, как видно на рис. 7.8.

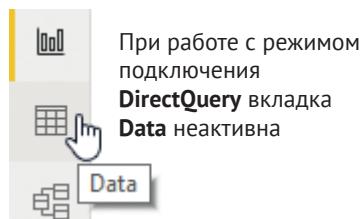


Рис. 7.8. Видимость вкладки **Data** (Данные) в Power BI

Теперь давайте поговорим о том, как Power BI хранит импортируемые данные и где можно их найти и посмотреть в среде Windows.

Где Power BI хранит данные

Когда отчет открыт в Power BI Desktop, набор данных физически хранится в памяти компьютера, на котором запущен этот отчет. После публикации отчета на сайте Power BI информация перетекает в облако. Уверены, при работе с Power BI вы неоднократно видели предупреждения о том, что данные нуждаются в обновлении. Именно поэтому мы иногда называем режим импорта отложенным обновлением.



Рис. 7.9. Хранение данных в Power BI

В следующем разделе мы посмотрим, как при помощи Диспетчера задач можно определить, запущена ли служба Microsoft SQL Server Analysis Services, ведь именно здесь хранятся данные при использовании режима подключения Import.

Определение статуса Microsoft SQL Server Analysis Services

Чтобы увидеть, где Power BI хранит свои данные при использовании режима **Import** (Импорт), выполните следующие действия.

1. Щелкните правой кнопкой мыши по панели задач Windows.
2. Выберите в предложенном меню пункт **Task Manager** (Диспетчер задач).
3. В открывшемся диалоговом окне прокрутите список вниз к задаче **Power BI Desktop**. Нажмите на раскрывающуюся стрелку слева от задачи, чтобы раскрыть список опций.
4. Найдите процесс **Microsoft SQL Server Analysis Services**, как показано на рис. 7.10.
5. Закройте окно диспетчера задач.

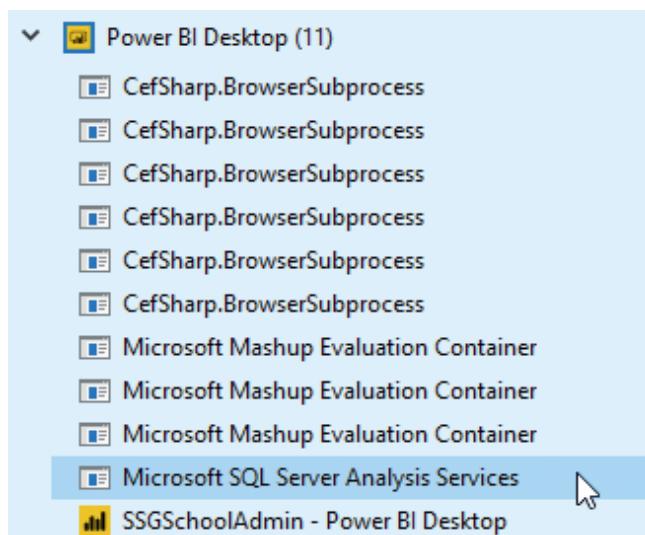


Рис. 7.10. Поиск процесса Microsoft SQL Server Analysis Services в диспетчере задач

В следующем разделе мы рассмотрим различные типы обновлений в Power BI.

ВВЕДЕНИЕ В ТИПЫ ОБНОВЛЕНИЙ В POWER BI

Критически важно следить за тем, чтобы данные в ваших отчетах в Power BI были максимально актуальными, насколько это возможно, поскольку нашим пользователям необходимо опираться на последние сведения при принятии решений. В данном разделе мы обсудим различные типы обновлений, что поможет вам понять, на что у Power BI может уходить время при выполнении актуализации данных.

В Power BI Online есть переключатель **New look** (Новый вид), который позволяет обновить расположение элементов в интерфейсе. В зависимости от выбранного у вас режима вы можете обнаружить иное расположение элементов по сравнению с нашими примерами. Сам переключатель **New look** располагается в строке заголовка Power BI Online ближе к правому краю, как показано на рис. 7.11.

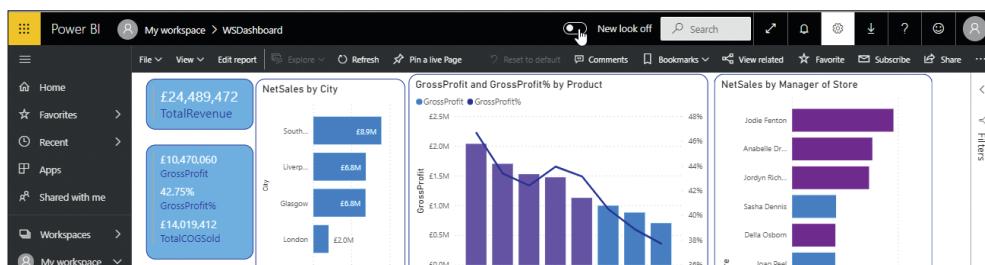


Рис. 7.11. Переключатель **New look** в Power BI Online

Набор данных в Power BI может состоять из нескольких потоков данных с разными режимами хранения. Это обуславливает разницу в требованиях к обновлению.

Данные, загруженные в режиме **Import** (Импорт), кешируются (а не импортируются заново), что приводит к значительному росту их объема при хранении. И проблемы с обновлением станут очевидны, когда размер набора данных дойдет до установленного предела. При использовании режима **LiveConnect / DirectQuery** данные просто не импортируются и не обновляются, как упоминалось выше. При этом сами визуальные элементы обновляются с периодичностью по умолчанию в час или около того, чтобы в отчетах всегда отображались наиболее актуальные данные. В то же время можно сделать определенную настройку набора данных, чтобы обновления дашбордов и отчетов производились вручную, – по нажатии на кнопку **Refresh Now** (Обновить сейчас). Мы поговорим об этой настройке далее.

Обновление подключения OneDrive

Любые отчеты и наборы данных, построенные в Power BI Desktop на основе файлов Excel или CSV и опубликованные с использованием OneDrive или Sharepoint, обновляются в Power BI посредством опций обновления OneDrive. При таком типе обновления Power BI обычно примерно раз в полчаса проверяет необходимость обновления подключения к файлам в OneDrive.

Примечание

Power BI не импортирует данные из OneDrive; он **синхронизирует** наборы данных и отчеты с обновлениями исходных файлов.

Для этого примера мы будем использовать отчет с дашбордом из файла `WSDashboard.pbix`, распространенный посредством OneDrive. Последовательно пройдемся по шагам с настройкой, после чего обратимся к опциям обновления.

1. Откройте Power BI Online по ссылке: <https://app.powerbi.com/home>.
2. Мы будем использовать для этого примера дашборд, построенный в главе 6. Этот отчет был импортирован в Power BI Online при помощи получения данных (Get Data) и подключен через онлайн-службу OneDrive. На рис. 7.12 показан внешний вид загруженного дашборда.

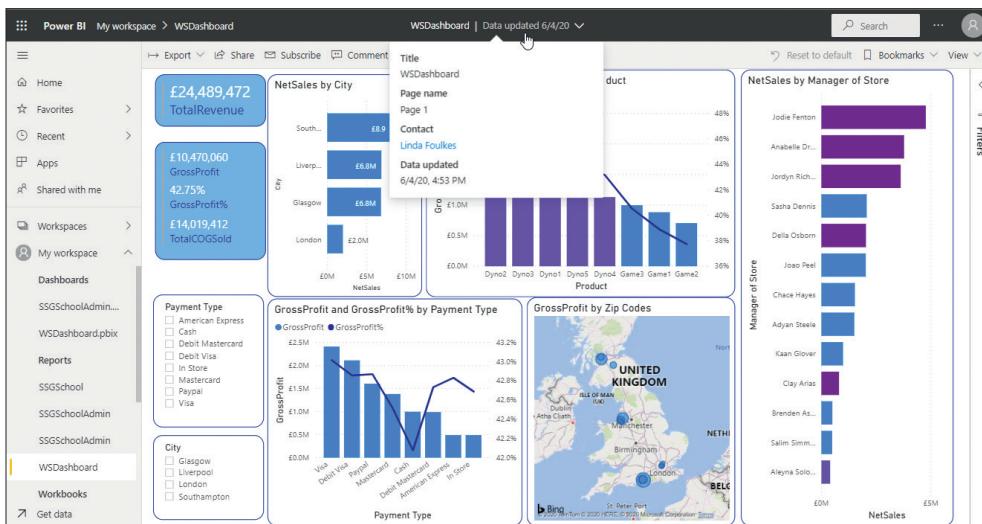


Рис. 7.12. Подключение к дашборду посредством OneDrive

Теперь давайте посмотрим, что нужно сделать, чтобы подключить ваш отчет через OneDrive.

- Нажмите на ссылку **Get data** (Получить данные) на левой навигационной панели Power BI Online.
- На вкладке **Files** (Файлы) нажмите на кнопку **Get** (Получить), как показано на рис. 7.13.

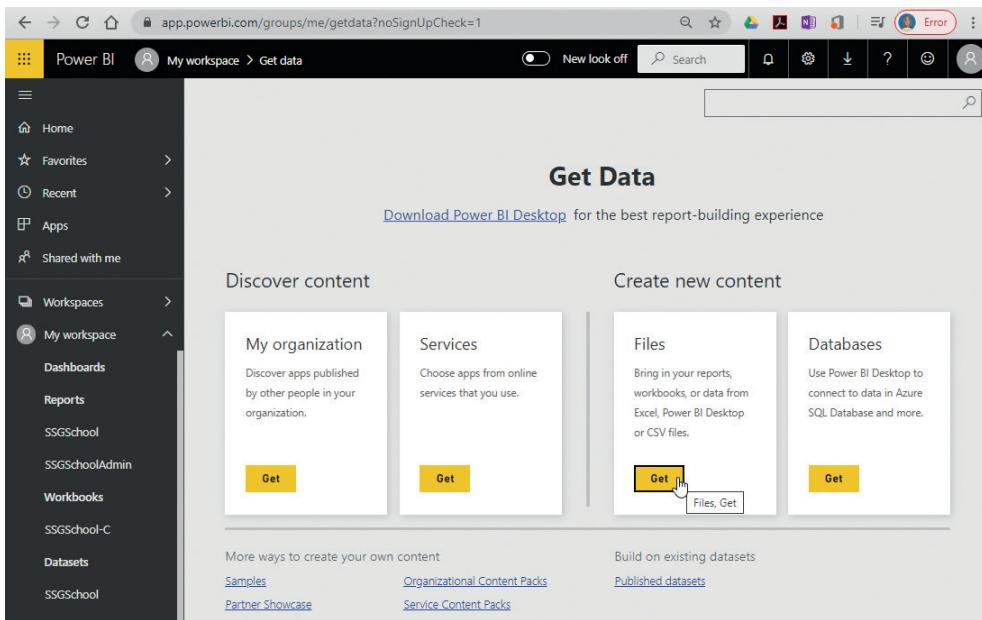


Рис. 7.13. Создание подключения к новому контенту

3. Выберите пункт **OneDrive – Personal** (OneDrive – персональный), как на рис. 7.14.

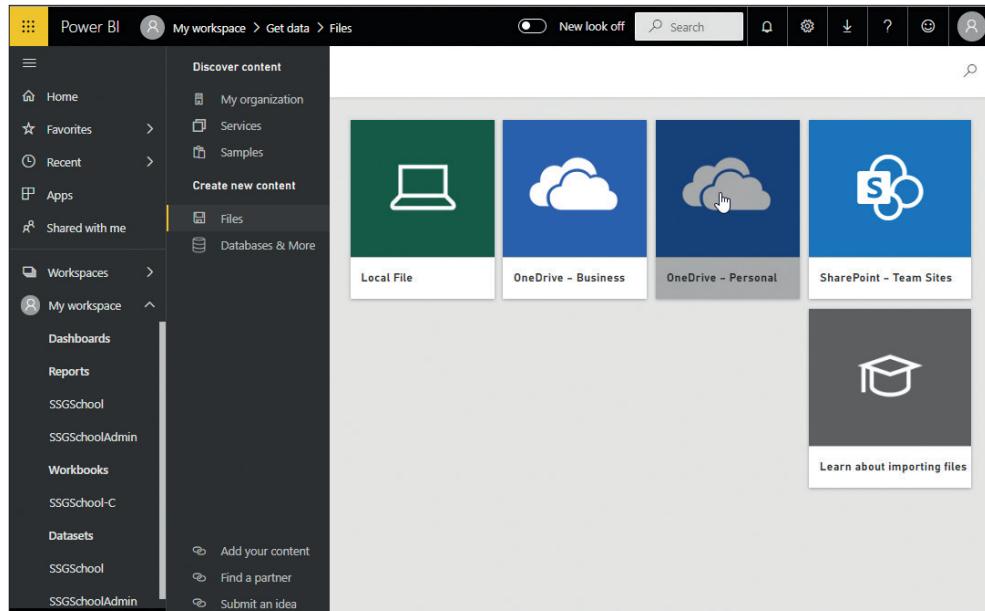


Рис. 7.14. Подключение OneDrive – персональный

4. Откроется диалоговое окно с установкой доступа для Power BI к данным, представленное на рис. 7.15. Нажмите на кнопку **Yes** (Да) для продолжения.

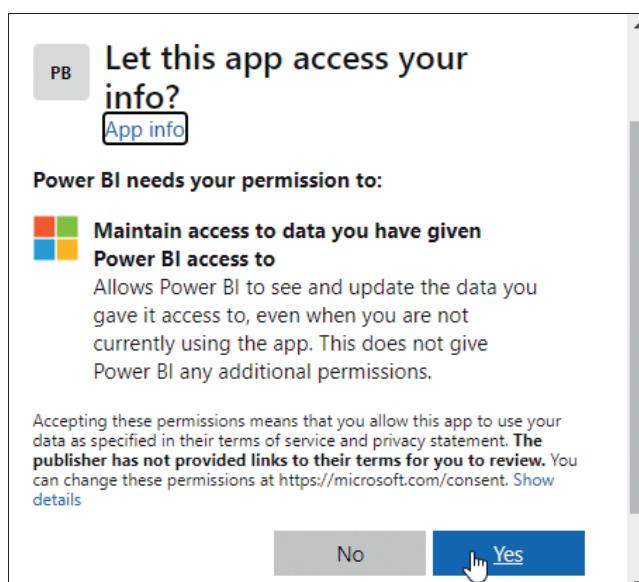


Рис. 7.15. Диалоговое окно безопасности доступа к данным

5. Выберите набор данных Power BI, к которому хотите подключиться, после чего нажмите на кнопку **Connect** (Подключение), как показано на рис. 7.16.

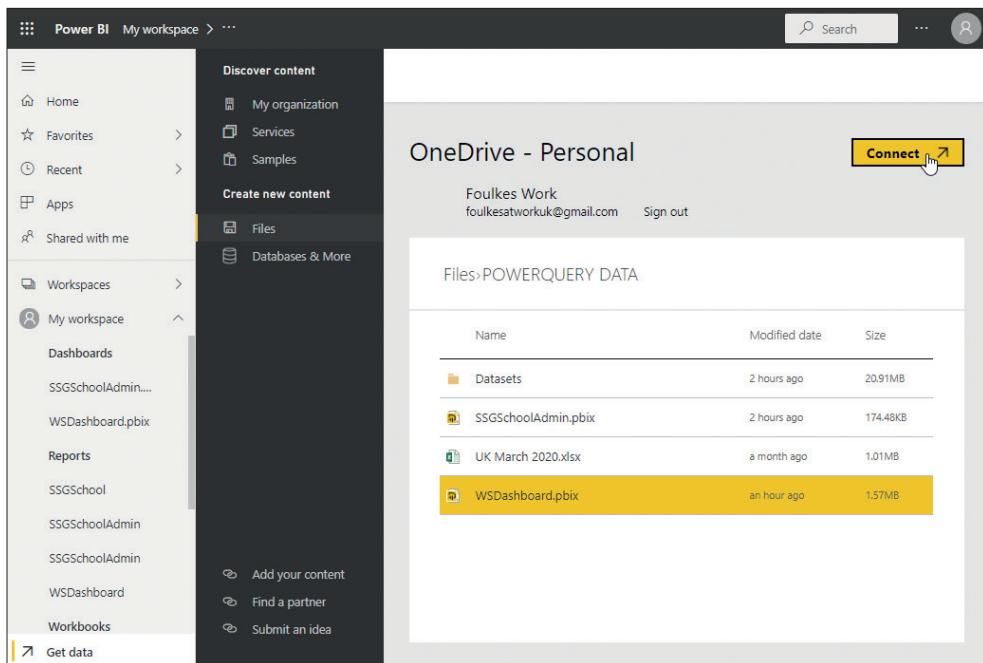


Рис. 7.16. Выбор набора данных из OneDrive

После этого Power BI Online подключится к выбранному набору данных, и будет открыт отчет под заголовком вашего рабочего пространства.

В данной главе мы пошагово рассмотрели подключение к отчету в OneDrive. Далее мы перейдем к теме обновлений.

Просмотр и выполнение обновлений в OneDrive

Можно несколькими способами узнать, когда в последний раз обновлялся подключенный через OneDrive набор данных в Power BI. Ниже мы покажем самый простой из них и расскажем, как обновлять подключение.

Взгляните на верхнюю панель Power BI Online, и вы обнаружите дату последнего обновления набора данных, по которой можно щелкнуть и посмотреть детальную информацию, как показано на рис. 7.17.

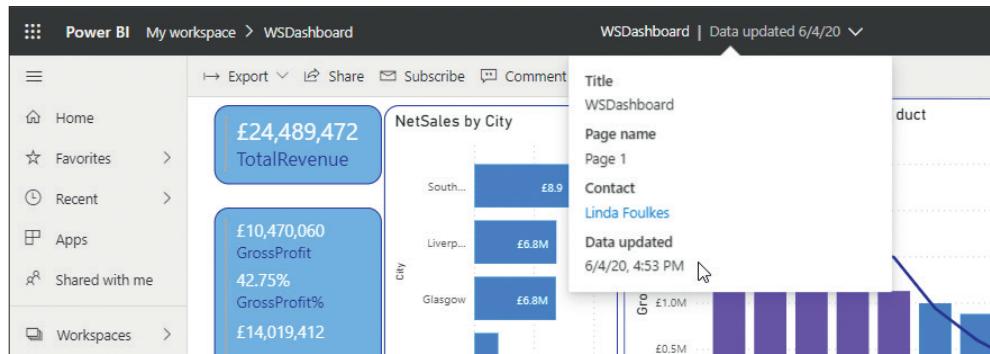


Рис. 7.17. Последнее обновление набора данных в Power BI

Теперь щелкните на троеточие в меню рабочего пространства и выберите из меню пункт обновления данных, как видно на рис. 7.18.

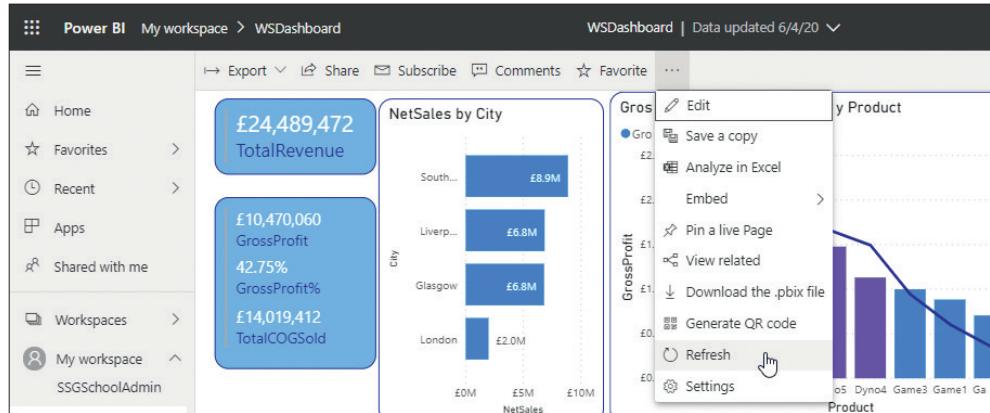


Рис. 7.18. Обновление данных из меню

Теперь откройте меню **Workspace** (Моя рабочая область) и выберите в горизонтальном меню пункт **Datasets + dataflows** (Наборы данных + потоки данных), что приведет к открытию содержимого в рабочей области ниже. Наведите мышь на интересующий вас набор данных, и рядом с именем набора появятся иконки **Refresh now** (Обновить) и **Schedule refresh** (Запланировать обновление), а также троеточие, ведущее к дополнительным опциям.

The screenshot shows the Power BI workspace interface. On the left, there's a navigation pane with options like Home, Favorites, Recent, Apps, Shared with me, Workspaces, and My workspace. Under My workspace, there are entries for SSGSchoolAdmin, WSDashboard, Workbooks, and SSGSchool-C. The main area is titled 'My workspace' and shows a list of datasets and dataflows. One dataset, 'WSDashboard', has a refresh icon and a tooltip 'Refresh now'. At the top right, there are buttons for View, Filters, and Search.

Рис. 7.19. Наборы и потоки данных с доступными действиями

Опции обновления и настройки данных также доступны по нажатии на троеточие в левой навигационной панели напротив нужного набора данных, что видно по рис. 7.20.

This screenshot shows a context menu for a dataset named 'WSDashboard'. The menu includes options like Analyze in Excel, Create report, Delete, Get quick insights, Security, Refresh now (which is highlighted with a cursor), Rename, Schedule refresh, Settings, Download .pbix, Manage permissions, and View lineage. To the right of the menu, there are several visualizations: a bar chart for Gross Profit by city, a bar chart for Gross Profit by payment type, and a map of Ireland.

Рис. 7.20. Доступ к действиям с набором данных через навигационную панель

Теперь, когда вы знаете, как обновлять наборы данных, подключенные посредством OneDrive, давайте научимся работать с запланированными обновлениями.

Настройка запланированных обновлений

Вы можете контролировать обновление наборов данных посредством встроенного в Power BI Online инструмента **запланированных обновлений**. Для начала выберите пункт **Settings** (Параметры) из меню с тремя точками в левой навигационной панели напротив набора данных SSGSchoolAdmin. Далее следуйте этой инструкции:

1. Раскройте пункт **Scheduled refresh** (Запланированное обновление) на панели **Settings** (Настройки).
2. Активируйте переключатель **Keep your data up to date** (Поддерживать актуальность данных).
3. Выберите из выпадающего списка **Refresh frequency** (Периодичность обновления) один из вариантов: Daily (Ежедневно) или Weekly (Еженедельно), как показано на рис. 7.21.

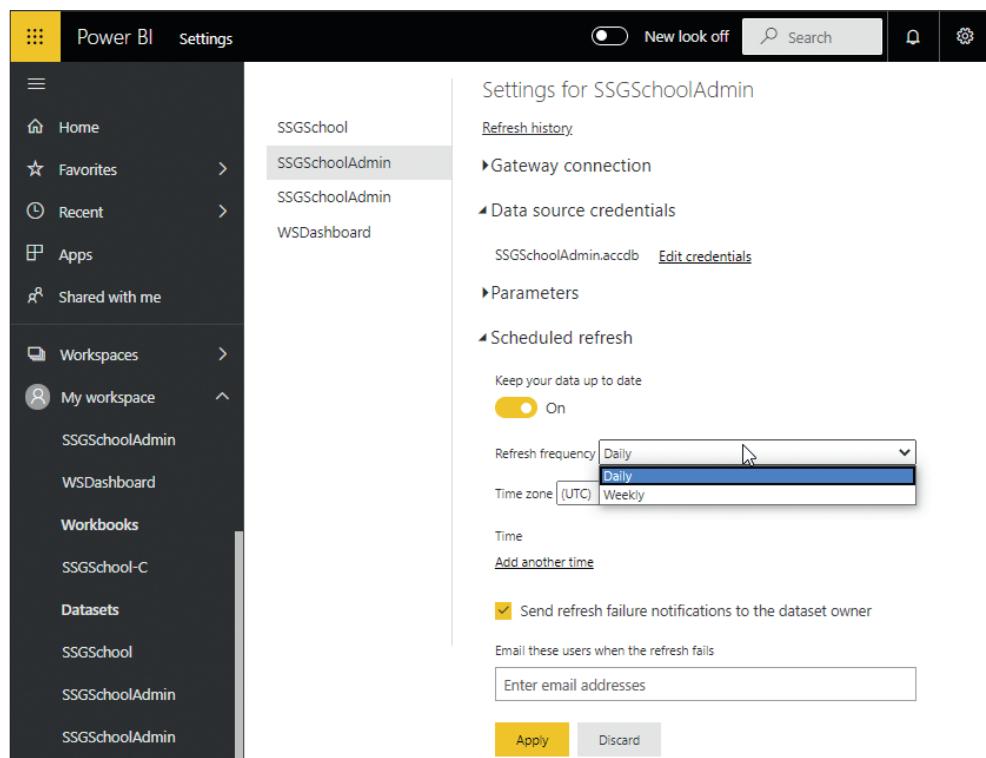


Рис. 7.21. Установка запланированных обновлений

4. Заметьте, что у вас есть возможность оповещать об ошибках ответственных за обновления людей по электронной почте. Для этого установите флажок **Send refresh failure notifications to the dataset owner** (Отправлять уведомления об ошибках обновления владельцу набора данных) и введите адреса электронной почты в соответствующее поле.

В этом разделе мы научились обновлять наборы данных немедленно и настраивать запланированные обновления. Далее мы продолжим рассматривать расширенные возможности обновлений.

Инкрементальное обновление

Инкрементальное обновление (Incremental refresh) – довольно новая опция в Power BI, которая нуждается в настройке. Этот вариант используется с целью ускорения обновлений при использовании объемных таблиц в Power BI Premium.

Главным преимуществом такого метода является то, что обновлениям подвергаются только данные, нуждающиеся в этом. Разработчик волен сам настроить политику обновлений в отношении строк, а также определять, должны ли отслеживаться изменения данных в рамках определенного периода, если максимальное значение в столбце **DateTime** изменилось. Этот тип обновления работает только после публикации отчета в службе Power BI. Подробнее о данном методе и дополнительных настройках можно почитать по адресу: <https://docs.microsoft.com/en-us/power-bi/admin/service-premium-incremental-refresh>.

Автоматическое обновление страницы

Автоматическое обновление страницы применимо только к источникам данных с типом подключения **DirectQuery**. После настройки этого режима Power BI Desktop будет автоматически посыпать запросы источнику (помните, что речь только об источниках **DirectQuery**) и через короткое время получать ответы, вместе с тем обновляя визуальные элементы на дашбордах. Заметим, что вы можете сами устанавливать интервалы обновления страницы для отчетов, созданных в Power BI Desktop и опубликованных в службе Power BI. Единственное ограничение здесь состоит в том, что интервал не должен быть меньше, чем время, требуемое на поступление новых данных в источник. Так, если данные поступают каждые 5 секунд, то и интервал обновления должен равняться 5 секундам.

Вы должны принимать во внимание время, требуемое для возвращения запросов к визуальным элементам, поскольку существует опасность загрузки слишком большого объема данных, и источник просто не сможет с этим справиться. Используйте уравнение для расчета времени возвращения запросов в отчеты, учитывающее количество визуальных элементов на дашборде и число пользователей, одновременно просматривающих отчеты. В результате вы получите количество запросов, которые вам необходимо поддерживать, и сможете адекватно установить интервалы автоматического обновления страницы. Подробнее об этой теме можно почитать по адресу: <https://docs.microsoft.com/en-us/power-bi/create-reports/desktop-automatic-page-refresh>.

Для настройки автоматического обновления страницы в Power BI Desktop необходимо выполнить следующие действия.

1. Перейдите на страницу, для которой нужно настроить автоматическое обновление.
2. Щелкните на иконку **Formatting** (Формат) в виде малярного валика на панели **Visualizations** (Визуализации).
3. Перейдите в раздел **Page refresh** (Обновление страницы) в нижней части панели.

4. Установите переключатель в верхней правой части раздела в положение **On** (Вкл.).
5. Введите интервал обновления. Минимальным значением является 1 секунда, а значением по умолчанию – 30 минут.
6. Теперь ваш отчет будет обновляться в соответствии с вашими настройками.

Из этого раздела вы узнали, как настраивать автоматическое обновление страниц в Power BI Desktop, а теперь посмотрим, как обновлять потоки данных.

Обновление потоков данных

Иногда, отслеживая значения **Start** (Начало) и **End** (Конец) в истории обновлений Power BI, вы будете обнаруживать определенные задержки. Также вы можете заметить задержки непосредственно в процессе обновления в окне применения изменений запроса. Обновление потоков данных позволяет ускорить ход обновления за счет запуска процесса Power Query отдельно от отчетов Power BI. В целом же обновления будут проходить быстрее, если настроить облачное хранилище Microsoft, именуемое Azure Data Lake Storage. Подробнее об этом можно почитать по адресу: <https://azure.microsoft.com/en-gb/services/data-lake-analytics>.

ЗАКЛЮЧЕНИЕ

В данной главе мы рассмотрели доступные в Power BI режимы хранения и типы наборов данных. Также мы узнали, как автоматизировать отчеты с использованием режимов в Power Query и зачем это делать.

Кроме того, мы научились просматривать текущий режим хранения данных и настраивать правила кеширования информации в Power BI. Подробно рассмотрели три режима хранения данных: **Import** (Импорт), **DirectQuery** и **Dual** (Двойной). Мы также узнали, как Power BI хранит свои данные и как обнаружить службу Microsoft SQL Server Analysis Services, в которой Power BI содержит модели данных.

В заключительной части главы мы познакомились с доступными типами обновлений в Power BI, включая **Import**, **LiveConnect/DirectQuery** и обновление источника **OneDrive**. Мы затронули тему запланированных обновлений, позволяющих контролировать расписание загрузки новых данных в отчеты. Также научились подключать отчеты к службе OneDrive и обновлять подключение путем синхронизации наборов данных, после чего познакомились с потоками данных, а еще с автоматическими и инкрементальными обновлениями.

В следующей главе мы сосредоточимся на создании дашбордов на основании подключений к данным, выборе типов визуализации, а также публикации и настройке дашбордов. Кроме того, мы затронем тему построения многомерных отчетов.

Глава 8

Создание дашбордов при помощи Power Query

Дашборды (dashboard) представляют собой страницы с набором визуализаций, позволяющие рассказать историю на основе данных из табличного источника для акцентирования внимания на важных аспектах с точки зрения ведения бизнеса.

Важнейшим преимуществом дашбордов является их полная интерактивность. Просматривать дашборды можно, находясь в любой точке мира, а если у вас достаточно прав, вы можете даже редактировать данные, лежащие в их основе. Разумеется, вся информация при этом будет актуальной.

Эта глава будет сугубо практической, и, читая ее, вы научитесь строить собственные дашборды, которые сможете публиковать онлайн. Попутно вы также освоите следующие навыки:

- выгрузку файлов специфического формата при наличии в папке файлов других форматов;
- преобразование данных таким образом, чтобы файлы, которые вы выгружаете, работали корректно;
- создание связей типа «один ко многим» для возможности использования других таблиц;
- создание различных мер для размещения расчетных величин на дашборде;
- разработку дашбордов на основе подключенных данных, выбор типов визуализации, а также публикацию и настройку дашбордов.

Основные темы данной главы:

- создание базовой сводной таблицы и сводной диаграммы;
- использование Power BI для сбора и подключения к данным;
- использование Power BI для добавления таблиц в модель данных;
- выбор типа визуализации данных и подходящей диаграммы;
- публикация дашборда;
- распространение дашборда.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Вам понадобится интернет-соединение для загрузки соответствующих файлов с GitHub. Файлы с кодом для этой главы можно скачать по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

Также для этой главы на сайте содержится дополнительная папка folder со всеми необходимыми файлами для импорта.

При написании главы мы опирались на то, что вы уже умеете загружать данные из различных источников в Power BI и Power Query.

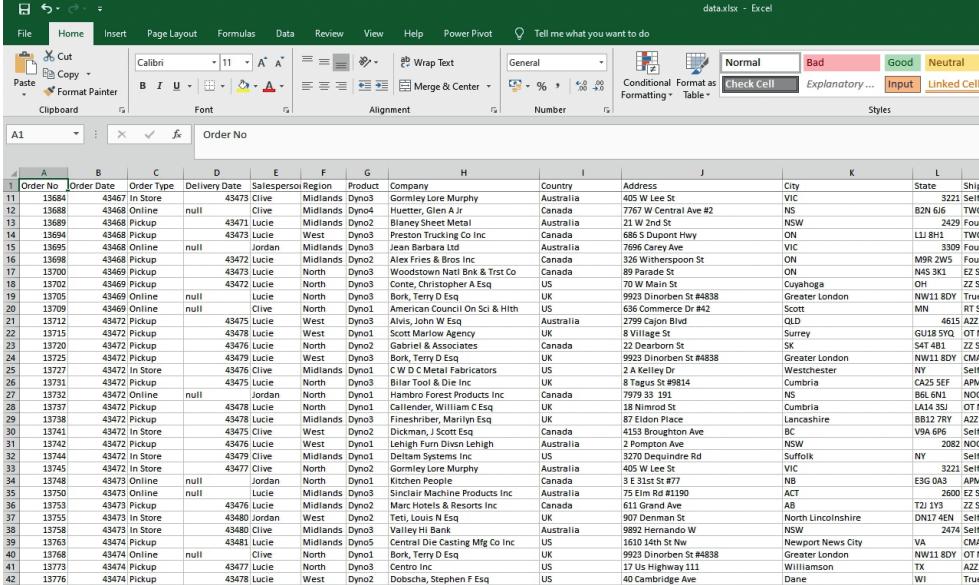
Видеофрагмент, посвященный данной главе, располагается по адресу: https://www.youtube.com/watch?v=MJ04RbLxxXk&list=PLeLcvrwLe1860_GEZs47WaZXwZjwTN83&index=9&t=0s.

СОЗДАНИЕ БАЗОВОЙ СВОДНОЙ ТАБЛИЦЫ И СВОДНОЙ ДИАГРАММЫ

Использование дашбордов позволяет представлять данные в визуальном виде вместо сухих цифр, что, в свою очередь, помогает более плодотворно анализировать информацию и отслеживать тенденции.

Зачастую вы получаете данные от третьих лиц или скачиваете файлы CSV, на основании которых вам необходимо построить отчет. Начнем с создания элементарной сводной таблицы и сводной диаграммы.

Для этого мы используем данные о продажах, покупателях и продавцах. Взгляните на исходные данные, представленные на рис. 8.1.



A	B	C	D	E	F	G	H	I	J	K	L	
Order No	Order Date	Order Type	Delivery Date	Salesperson	Region	Product	Company	Country	Address	City	State	
11	13684	43467	In Store	43473	Clive	Midlands	Dyno3	Gormley,Lore,Murphy	Australia	405 W Lee St	VIC	3221 Self
12	13688	43468	Online	null	Clive	Midlands	Dyno4	Huetter,Glen,Air	Canada	7767 W Central Ave #2	NS	B2N 6IG TWO
13	13689	43468	Pickup	43471	Lucie	Midlands	Dyno2	Blaney Sheet Metal	Australia	21 W 2nd St	NSW	2429 Four
14	13694	43468	Pickup	43475	Lucie	West	Dyno3	Preston Trucking Co Inc	Canada	686 S Dupont Hwy	ON	L1J 8H1
15	13695	43468	Online	null	Jordan	Midlands	Dyno3	Jean Barbara Ltd	Australia	7696 Carey Ave	VIC	3309 Four
16	13696	43469	Online	43472	Lucie	Midlands	Dyno3	Alpha One Manufacturing Inc	Canada	500 N 1st Street	ON	M9R 2W5 Four
17	13700	43469	Pickup	43473	Lucie	North	Dyno3	Woodstock Natl Bnk & Trst Co	Canada	89 Paradise St	ON	N4S 3K1 ZZ Sh
18	13702	43469	Pickup	43472	Lucie	North	Dyno3	Centex,Christopher A Esq	US	70 W Main St	Cuyahoga	OH 44236 ZZ Sh
19	13705	43469	Online	null	Clive	North	Dyno3	American Council On Sci & Hlth	US	9923 Diorben St #4388	Greater London	NW11 8DY True
20	13709	43469	Online	null	Clive	North	Dyno1	American Council On Sci & Hlth	US	636 Commerce Dr #42	Scott	MN 55311 RT Sh
21	13712	43472	Pickup	43475	Lucie	West	Dyno1	Alvis, John W Esq	Australia	2799 Cajon Blvd	QLD	4615 A22 S
22	13715	43472	Pickup	43476	Lucie	West	Dyno1	Scott Marlow Agency	UK	8 Village St	Surrey	GU18 5YQ OT N
23	13720	43472	Pickup	43476	Lucie	North	Dyno2	22 Diorben St	Canada	9923 Diorben St #4388	SK	S4T 4B1 ZZ Sh
24	13725	43472	Pickup	43476	Lucie	North	Dyno2	Gardner, B&B	US	2 A Valley Pkwy	Greater London	NW11 8DY CMA
25	13727	43471	In Store	43476	Clive	Midlands	Dyno3	W V D C Metal Fabricators	US	2 A Valley Pkwy	Westchester	NY 10598 FF
26	13731	43472	Pickup	43475	Lucie	North	Dyno3	Bitar Tool & Die Inc	UK	8 Tagus St #9814	Cumbria	CA25 5EF APM
27	13732	43472	Online	null	Jordan	North	Dyno3	Hambro Forest Products Inc	Canada	7973 33 191	NS	B61 8NU NOG
28	13737	43472	Pickup	43478	Lucie	North	Dyno1	Callender, William C Esq	UK	18 Nimrod St	Cumbria	LA14 3SJ OT N
29	13738	43472	Pickup	43478	Lucie	Midlands	Dyno3	Finishter, Marilyn Esq	UK	87 Edion Place	Lancashire	BB12 7RY A22 S
30	13741	43471	In Store	43475	Clive	West	Dyno2	Dickman, J Scott Esq	Canada	4153 Broughton Ave	BC	V9A 6P6 Self
31	13742	43472	Pickup	43476	Lucie	North	Dyno1	Leigh Burn Furn Leigh	Australia	2 Pompton Ave	NSW	2082 NOG
32	13744	43472	Pickup	43479	Clive	Midlands	Dyno3	Leighburn Systems Inc	US	5270 Denbigh Rd	Suffolk	NY 11701 Self
33	13745	43471	In Store	43477	Lucie	North	Dyno2	Gormley,Lore,Murphy	Australia	405 W Lee St	VIC	3221 Self
34	13748	43473	Online	null	Jordan	North	Dyno1	Kitchen People	Canada	8 E 31st St #77	NB	E3G 0A3 APM
35	13750	43473	Online	null	Lucie	Midlands	Dyno3	Sinclair Machine Products Inc	Australia	75 Elm Rd #1190	ACT	E600 EZ Sh
36	13753	43473	Pickup	43476	Lucie	Midlands	Dyno2	Marc Hotels & Resorts Inc	Canada	611 Grand Ave	AB	T2J 1Y3 ZZ Sh
37	13755	43473	In Store	43480	Jordan	West	Dyno2	Teti, Louis N Esq	UK	907 Denman St	North Lincolnshire	DN17 4EN Self
38	13758	43473	In Store	43480	Clive	Midlands	Dyno3	Valley Hi Bank	Australia	9892 Hernando W	NSW	2474 Self
39	13763	43474	Pickup	43481	Lucie	Midlands	Dyno5	Central Die Casting Mfg Co Inc	US	1610 14th St Nw	Newport News City	VA CMA
40	13768	43474	Online	null	Clive	North	Dyno1	Bork, Terry D Esq	UK	9923 Diorben St #4388	Greater London	NW11 8DY OT N
41	13773	43474	Pickup	43477	Lucie	North	Dyno3	Centro Inc	US	17 Us Highway 111	Williamson	TX A2Z S
42	13776	43474	Pickup	43478	Lucie	West	Dyno2	Dobscha, Stephen F Esq	US	40 Avenue Ave	Dane	WI True

Рис. 8.1. Исходная таблица

Давайте построим сводную таблицу и сводную диаграмму по продавцам с информацией о том, на какую сумму они продали товаров.

Примечание

Если вы уже знакомы с созданием сводной таблицы и сводной диаграммы, сразу переходите к следующему разделу.

1. Преобразуйте диапазон в умную таблицу при помощи сочетания клавиш **Ctrl+T**, убедившись в том, что в выделение попала вся таблица.
2. Создайте сводную таблицу, нажав на вкладке **Insert** (Вставка) кнопку **PivotTable** (Сводная таблица).
3. В появившемся диалоговом окне установите переключатель в положение **New Worksheet** (На новый лист) и нажмите на кнопку **OK**.

Примечание

Перед созданием сводной таблицы убедитесь, что в исходных данных нет пустых столбцов, пустых строк или объединенных ячеек.

После выполнения этих шагов взгляните на рис. 8.2.

	B	C	D	E	F	G
1						
2						
3	Row Labels ↴	Sum of Net Sales				
4	Lucie	£1,111,961.00				
5	John	£513,650.00				
6	Jordan	£384,341.00				
7	Clive					
8	Grand Total					
9						
10						
11						
12						
13						
14						
15						
16		Sum				
17		Count				
18		Average				
19		Max				
20		Min				
21		Product				
22						
23						

Рис. 8.2. Настройка полей со значениями в сводной таблице

Здесь не хватает нескольких вещей.

1. Перенесите поле **Salesperson** на строки, а **Sum of Net Sales** – в область значений.
2. Значения валюты в сводной таблице отформатированы неправильно. Щелкните на любом значении в таблице правой кнопкой мыши и выберите пункт **Value Field Settings** (Параметры полей значений).
3. Нажмите на кнопку **Number Format** (Числовой формат), выберите формат **Currency** (Денежный) и закройте окно по кнопке **OK**.
4. Выполните ранжирование продавцов, нажав правой кнопкой на денежное поле и выбрав из меню **Sort** (Сортировка) пункт **Sort Largest to Smallest** (Сортировка по убыванию).

Теперь, когда мы привели наши данные в правильный формат, можно перейти к созданию сводной диаграммы, на которой будет визуально отражено соотношение продаж по нашим продавцам.

Выполните следующие шаги.

1. Нажмите на кнопку **PivotChart** (Сводная диаграмма) в меню **Analyze** (Анализ) и выберите тип диаграммы **Bar** (Линейчатая), как показано на рис. 8.3.

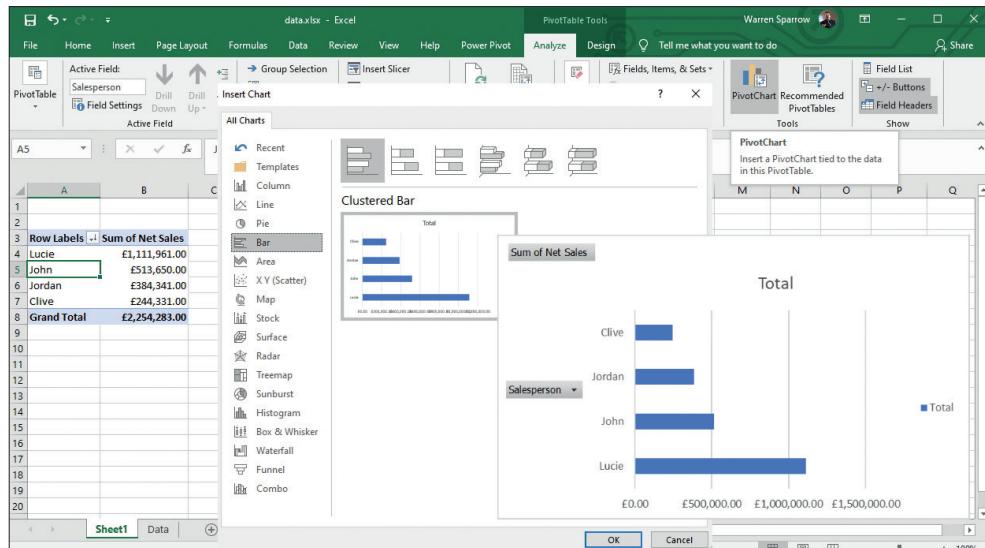


Рис. 8.3. Вставка сводной диаграммы

2. Чтобы привести диаграмму в порядок, нажмите правой кнопкой мыши на кнопку **Salesperson** и выберите в контекстном меню пункт **Hide All Field Buttons on Chart** (Скрыть все кнопки полей на диаграмме).
3. Порядок следования продавцов на нашей диаграмме не совпадает со сводной таблицей. Давайте это исправим. Щелкните правой кнопкой мыши на любом продавце, выберите пункт **Format Axis** (Формат оси) и в

появившейся справа вкладке установите флажок **Categories in reverse order** (Обратный порядок категорий).

4. Удалите с диаграммы метки, легенду и вертикальные линии. Также переименуйте заголовок диаграммы, щелкнув правой кнопкой мыши по тексту **Total** (Итог) и выбрав пункт **Edit Text** (Изменить текст).
5. Нам хотелось бы сделать столбчики на диаграмме потолще, чтобы их было лучше видно. Для этого нажмите правой кнопкой мыши по любому столбчику и выберите в меню пункт **Format Data Series** (Формат ряда данных). В появившейся панели справа уменьшите значение в поле бокового зазора, что приведет к утолщению столбиков. Установите комфортное значение в зависимости от количества продавцов на диаграмме.

На рис. 8.4 показан итоговый вид диаграммы.

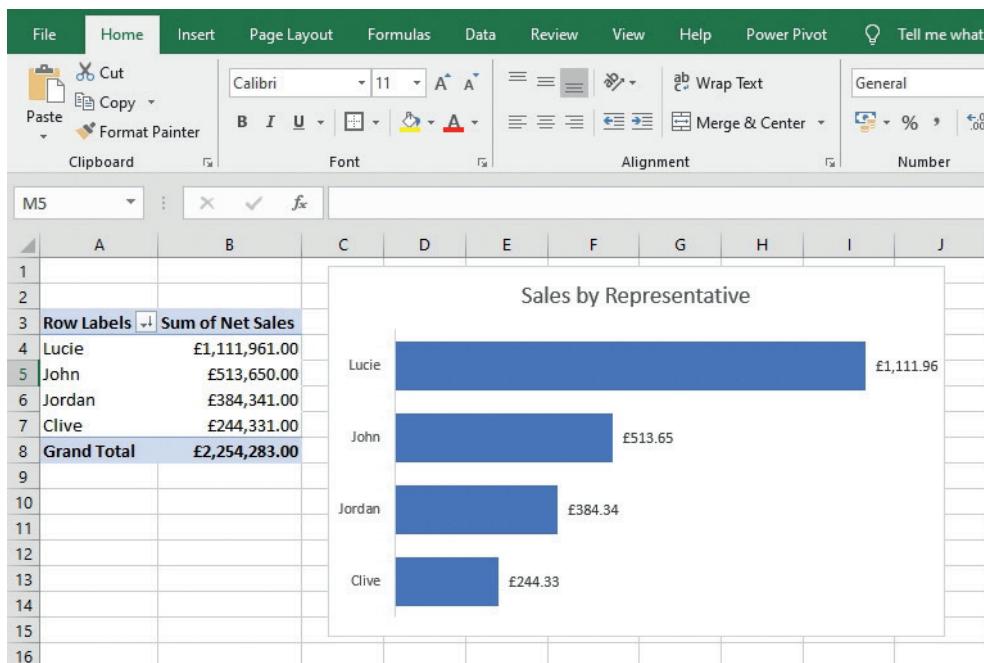


Рис. 8.4. Сводная диаграмма по продавцам

Как видите, базовые сводные таблицы и сводные диаграммы создавать очень легко и просто, а выглядят они весьма неплохо.

В следующем разделе мы посмотрим на процесс построения интерактивного дашборда при помощи Power BI Desktop.

Использование Power BI для сбора и подключения к данным

Для создания полноценного дашборда необходимо выполнить несколько шагов. Первое, что нужно сделать, – это собрать данные и подключить их к нашей модели данных для дальнейших преобразований.

Мы будем использовать данные о продажах в магазинах, базирующихся в Великобритании. При этом информация может храниться онлайн, в SQL Server, на сайте, а может быть загружена и локально, как в нашем примере. Единственным отличием будет способ получения данных. При необходимости вы всегда можете вернуться к главе 4 и вспомнить, как выполняется подключение из Power BI Desktop к различным источникам. Исходные файлы для нашего примера можно найти в репозитории GitHub.

На рис. 8.5 представлены исходные данные в виде таблицы. Здесь есть индекс магазина (**Store ZIP Code**), наименование проданного товара (**Product**), вырученная сумма (**Amount of Sale**) и себестоимость товара (**COGS**).

	A	B	C	D	E	F	G	H	
1	Date	Store Zip Code	Product	Payment Type	Units	Amount of Sale	Revenue	Discount	COGS
2	01/10/2019	WC2E 9DD	Dyno1	Cash	1	4.95	0	3.66	
3	01/10/2019	N18 3HF	Dyno1	Mastercard	1	4.95	0	4.03	
4	01/10/2019	N18 3HF	Dyno1	Visa	1	4.95	0	2.99	
5	01/10/2019	SE1 7ND	Dyno1	Cash	1	4.95	0	4.15	
6	01/10/2019	WC2E 9DD	Dyno1	Mastercard	2	9.9	0	7.34	
7	01/10/2019	N18 3HF	Dyno1	American Express	1	4.95	0	2.77	
8	01/10/2019	SE1 7ND	Dyno1	Paypal	6	29.7	0.1	22.32	
9	01/10/2019	WC2E 9DD	Dyno1	Mastercard	6	29.7	0.1	18.54	
10	01/10/2019	WC2E 9DD	Dyno1	Mastercard	1	4.95	0	2.58	
11	01/10/2019	N18 3HF	Dyno1	Visa	1	4.95	0	3.04	
12	01/10/2019	SE1 7ND	Dyno1	Debit Visa	1	4.95	0	3.37	
13	01/10/2019	SE1 7ND	Dyno1	Debit Visa	1	4.95	0	3.5	
14	01/10/2019	WC2E 9DD	Dyno1	Visa	2	9.9	0	7.22	
15	01/10/2019	WC2E 9DD	Dyno1	Debit Visa	1	4.95	0	3.6	
16	01/10/2019	SE1 7ND	Dyno1	Mastercard	1	4.95	0	2.91	
17	01/10/2019	WC2E 9DD	Dyno1	Cash	1	4.95	0	2.61	
18	01/10/2019	WC2E 9DD	Dyno1	Paypal	1	4.95	0	2.87	
19	01/10/2019	SE1 7ND	Dyno1	In Store	1	4.95	0	3.96	
20	01/10/2019	N18 3HF	Dyno1	Visa	1	4.95	0	3.73	
21	01/10/2019	WC2E 9DD	Dyno1	Mastercard	1	4.95	0	3.4	
22	01/10/2019	WC2E 9DD	Dyno1	Mastercard	1	4.95	0	4.14	
23	01/10/2019	SE1 7ND	Dyno1	American Express	1	4.95	0	3.51	
24	01/10/2019	WC2E 9DD	Dyno1	Mastercard	1	4.95	0	2.75	

Рис. 8.5. Исходные данные о продажах

Чтобы подключиться к данным из Power BI, выполните следующие действия.

1. Откройте Power BI Desktop, в группе **Data** (Данные) раскройте выпадающую кнопку **Get Data** (Получить данные) и выберите пункт **More...** (Другие...). Как мы уже говорили, если ваши данные скопированы на другой источник, обратитесь к главе 4, чтобы узнать способы подключения к ним.
2. Выберите в предложенных вариантах папку и нажмите на кнопку **Connect** (Подключить).
3. В диалоговом окне **Folder** (Папка) нажмите на кнопку **Browse** (Обзор) и укажите папку с данными.
4. После выбора папки нажмите на кнопку **OK**.

В результате будет отображен список файлов для загрузки, показанный на рис. 8.6.

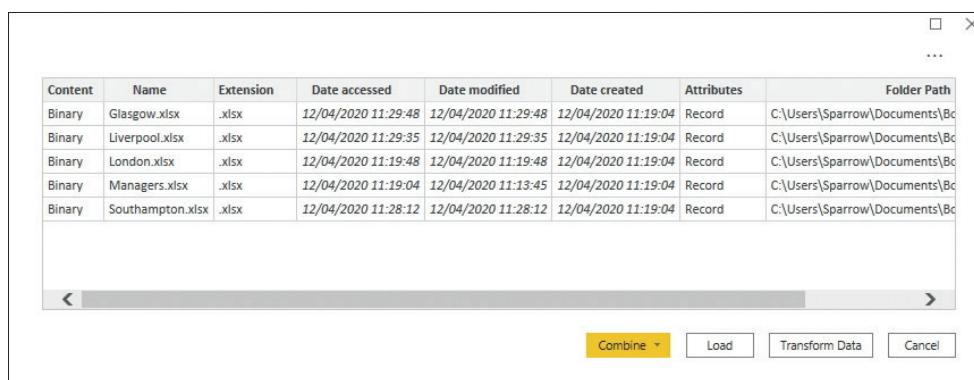


Рис. 8.6. Доступные файлы в папке

Нажмите на кнопку **Transform Data** (Преобразовать данные), чтобы выполнить необходимое форматирование информации перед загрузкой.

Откроется редактор запросов Power Query, показанный на рис. 8.7, – очень похожий на тот, который открывался при обращении из Excel.

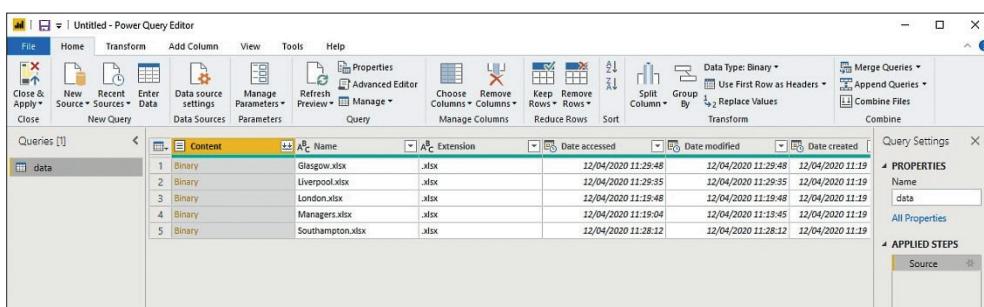


Рис. 8.7. Редактор запросов Power Query

Мы изменим имя набора данных, чтобы нам было легче работать с ним в дальнейшем. Поменяйте содержимое поля **Name** (Имя) на левой панели **PROPERTIES** (Свойства) на SalesData. Это имя станет именем запроса, а также названием набора данных, который мы позже импортируем.

Если взглянуть на список импортированных файлов, легко заметить, что все они являются рабочими книгами Excel. Позже, если мы добавим в эту папку файлы других типов, у нас могут возникнуть проблемы. К тому же хорошим тоном считается принудительное приведение расширений файлов к нижнему регистру. Причина заключается в регистрозависимости формул в Power Query. Это можно легко сделать, щелкнув правой кнопкой мыши на столбце **Extension** и выбрав в подменю **Transform** (Преобразование) пункт **lowercase** (нижний регистр), как показано на рис. 8.8. После этого все расширения импортируемых файлов будут автоматически приводиться к нижнему регистру, даже если пользователь при сохранении файлов введет их в верхнем.

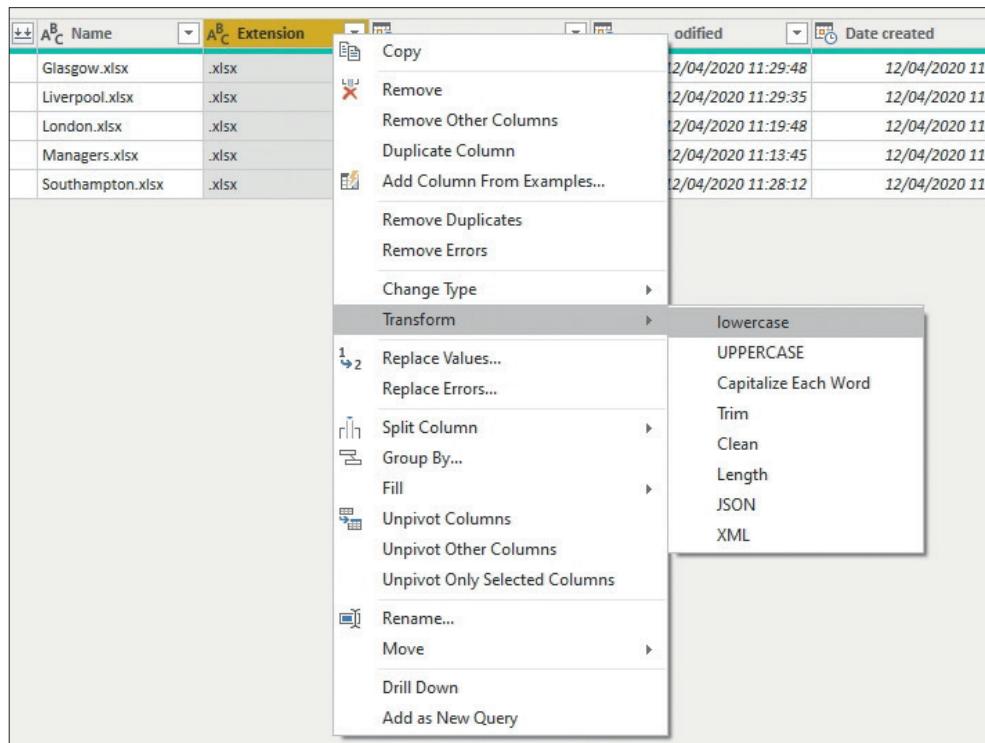


Рис. 8.8. Приведение расширений файлов к нижнему регистру

Второе действие, которое желательно сделать, – это убедиться в том, что обрабатываться будут исключительно файлы Excel. Щелкните по столбцу **Extension** и в подменю **Text Filters** (Текстовые фильтры) выберите пункт **Equals** (Равно), как показано на рис. 8.9.

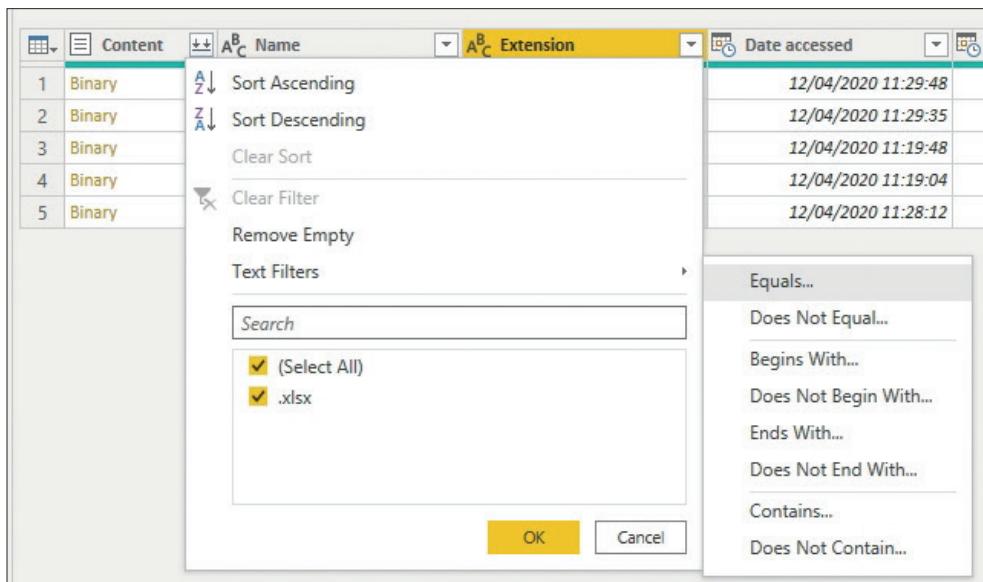


Рис. 8.9. Установка текстового фильтра

Мы не хотим, чтобы импортировались файлы других типов, так что можем ограничиться только рабочими книгами Excel. Для этого введите в открывшемся диалоговом окне значение .xlsx, как на рис. 8.10, что также отсечет загрузку файлов Excel старого формата.

Рис. 8.10. Диалоговое окно фильтрации строк

Поскольку нам не нужны никакие колонки, кроме Content, мы свободно можем их удалить. Для этого щелкните правой кнопкой мыши на единственном нужном нам столбце **Content** и выберите пункт меню **Remove Other Columns** (Удалить другие столбцы), как показано на рис. 8.11.

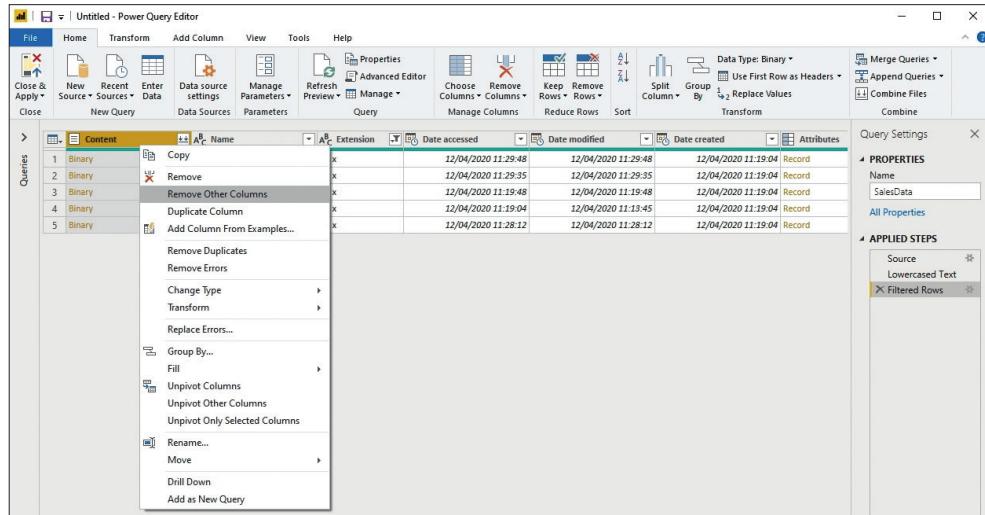


Рис. 8.11. Удаление ненужных столбцов

Если щелкнуть на любой строке правее слова **Binary**, внизу будет показан ярлык файла Excel, на который ссылается строка, что видно по рис. 8.12.

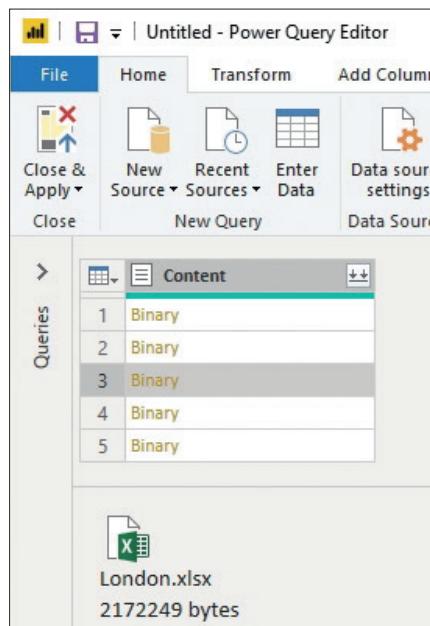


Рис. 8.12. Двоичные файлы Excel

Теперь посмотрим, как можно объединить наши файлы.

Объединение файлов

Обычно для объединения файлов Excel достаточно воспользоваться иконкой **Combine Files** (Объединить файлы) в виде двух стрелок вниз рядом с именем столбца Content. При этом все данные из файлов будут извлечены и объединены. К сожалению, в нашем случае такой способ не подойдет из-за различий в заголовках столбцов в файлах.

Нам придется создать дополнительный столбец и использовать функцию рабочей книги Excel для извлечения нужных нам данных из разных файлов Excel. Посмотрим, как это можно сделать.

1. На вкладке **Add Column** (Добавление столбца) нажмите на кнопку **Custom Column** (Настраиваемый столбец), как показано на рис. 8.13, что приведет к открытию соответствующего диалогового окна.

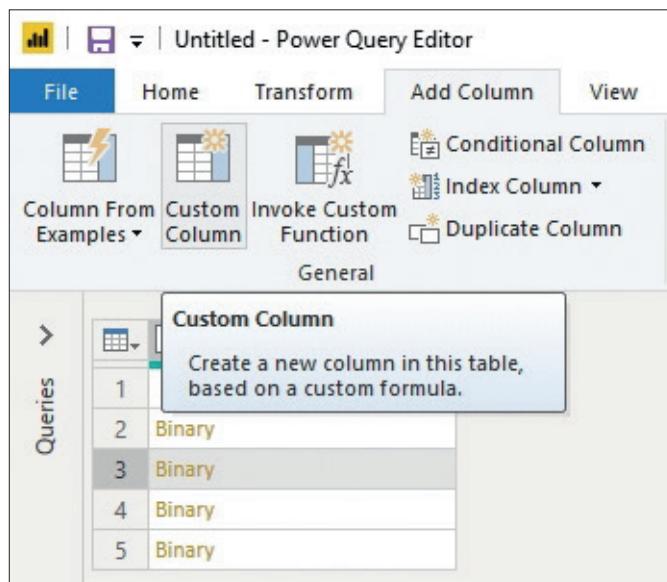


Рис. 8.13. Добавление настраиваемого столбца

2. В поле для имени столбца напишите `GetExcelData`, в поле **Custom column formula** (Настраиваемая формула столбца) введите следующую формулу: `Excel.Workbook([Content], true)`, как показано на рис. 8.14, после чего нажмите на кнопку **OK**.

Примечание

Чтобы добавить в код ссылку на столбец [Content], можно дважды щелкнуть на соответствующем поле в правом списке **Available columns** (Доступные столбцы).

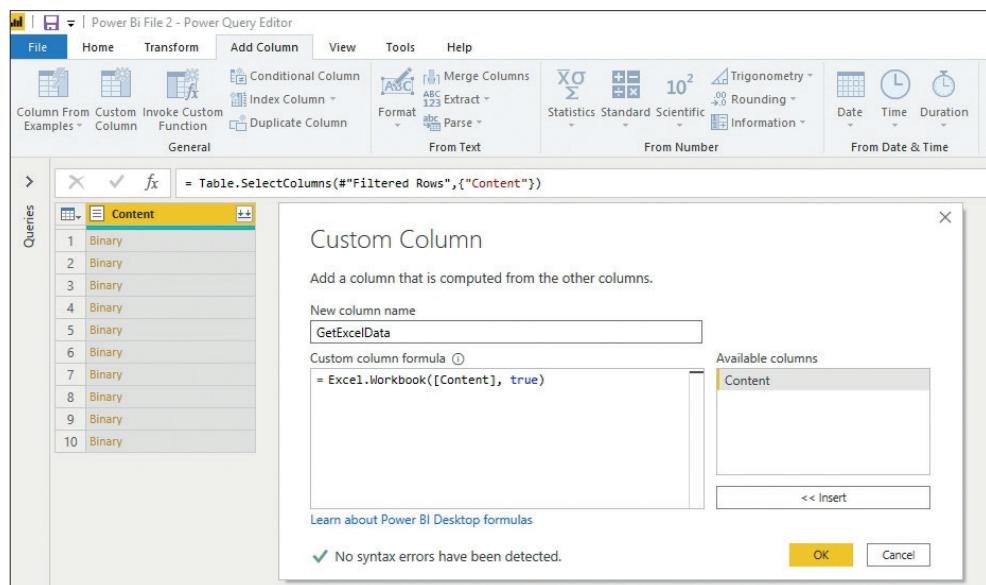


Рис. 8.14. Формула нового столбца

Первая часть формулы (до запятой) указывает Power BI открыть содержимое каждого файла Excel. Второй параметр `true` сообщает о необходимости предоставить заголовки.

Далее нам предстоит сохранить объекты, которые нам нужны, и избавиться от ненужной информации.

Щелкнув по ячейке в созданном столбце **GetExcelData**, можно обнаружить, что Power BI извлек разные объекты из файлов, как видно на рис. 8.15.

Name	Data	Item	Kind	Hidden
London	Table	London	Sheet	FALSE
Sheet1	Table	Sheet1	Sheet	FALSE

Рис. 8.15. Столбец GetExcelData

Нам больше не нужна колонка **Content**, поскольку вся необходимая нам информация теперь находится в столбце **GetExcelData**. Удалим **Content**. Теперь в нашем распоряжении оказались только файлы Excel с заголовками, так что можно воспользоваться иконкой **Combine Files** справа от заголовка столбца **GetExcelData**.

Единственное, что нужно сделать, – это снять флагок **Use original column name as prefix** (Использовать исходное имя столбца как префикс) и нажать на кнопку **OK**.

После этого в запрос будут загружены самые разные объекты из исходных файлов, что видно по рис. 8.16, и не все они нам нужны.

The screenshot shows the Power BI ribbon with the 'File', 'Home', 'Transform', 'Add Column', 'View', 'Tools', and 'Help' tabs. Below the ribbon, there are several icons for data manipulation: 'Column From Examples', 'Custom Column', 'Invoke Custom Function', 'Conditional Column', 'Index Column', 'Duplicate Column', 'Format', 'Parse', 'Merge Columns', 'Extract', 'Statistics', 'Standard', 'Scientific', 'Trigonometry', 'Rounding', 'Information', 'From Text', 'From Number', 'From Date & Time', and 'From Date & Time'. The 'Queries' section is expanded, displaying a table with columns: 'Name' (containing 'ABC 123'), 'Data' (containing 'Table'), 'Item' (containing 'Glasgow'), 'Kind' (containing 'Sheet'), and 'Hidden' (containing 'FALSE'). The table lists various objects from multiple Excel files, such as 'London', 'Sheet1', 'Sheet2', 'PhoneTable', etc. Below this table is another smaller table with columns: 'Date', 'Store', 'Zip Code', 'Product', 'Payment Type', 'Units', 'Amount of Sale', 'Revenue', 'Discount', and 'COGS'. This table contains specific transaction data like '11/12/2019 G3 8AG Dino1 Visa 1 24.95 0 13.54'.

Рис. 8.16. Загруженные объекты

На рис. 8.16 можно видеть разные столбцы, и вот что они означают:

- в столбце **Name** содержится название объекта;
- в столбце **Data** находятся данные из файла Excel, и если щелкнуть по нему, внизу будет показан предварительный просмотр этой информации;
- на колонку **Kind** необходимо обратить особое внимание, поскольку нам из нее понадобится только тип **Sheet**.

Щелкните на кнопке со стрелкой справа от заголовка столбца **Kind**, в появившемся диалоговом окне выберите в меню **Text Filters** (Текстовые фильтры) пункт **Equals** (Равно) и введите текст **Sheet** с первой заглавной буквой, поскольку именно так данные выглядят в этой колонке.

При беглом просмотре содержимого столбца **Name** становится понятно, что среди нужных нам листов здесь есть также листы по умолчанию с названиями вроде **Sheet1**, которые не содержат никакой важной информации для нас.

	ABC 123 Name	ABC 123 Data	ABC 123 Item	ABC 123 Kind	ABC 123 Hidden
1	Glasgow	Table	Glasgow	Sheet	FALSE
2	Liverpool	Table	Liverpool	Sheet	FALSE
3	London	Table	London	Sheet	FALSE
4	Sheet1	Table	Sheet1	Sheet	FALSE
5	Sheet2	Table	Sheet2	Sheet	FALSE
6	ManagerTable	Table	ManagerTable	Sheet	FALSE
7	Southampton	Table	Southampton	Sheet	FALSE

Рис. 8.17. Листы с именами по умолчанию

Щелкните на кнопке фильтрации справа от заголовка столбца **Name**, в диалоговом окне выберите в меню **Text Filters** (Текстовые фильтры) пункт **Does Not Contain...** (Не содержит...), введите **Sheet** и нажмите на кнопку **OK**.

Теперь в списке остались только строки с нужными нам таблицами, за исключением таблицы **ManagerTable**, от которой мы скоро избавимся.

	ABC 123 Name	ABC 123 Data
1	Glasgow	Table
2	Liverpool	Table
3	London	Table
4	ManagerTable	Table
5	Southampton	Table

Рис. 8.18. Кнопка разворачивания данных

Нам нужно сохранить только столбцы **Name** и **Data**, так что все остальные колонки можно удалить.

Щелкните по иконке расширения в правой части заголовка столбца **Data**, как показано на рис. 8.18. Теперь вы можете выбрать столбцы, которые вам нужны. Выберем их все, за исключением **Manager of Store**, и нажмем на кнопку **OK**.

The screenshot shows a Power BI data editor window with a table titled "Expanded Data". The table has columns: Name, Date, Store Zip Code, Product, Payment Type, Units, Amount of Sale, and Revenue Discount. The data consists of 25 rows, each representing a transaction from Glasgow. The "Name" column contains "Glasgow" repeated 25 times. The "Date" column shows dates from 11/12/2019 to 28/11/2019. The "Store Zip Code" column shows codes like G3 8AG, G62 8EP, etc. The "Product" column lists products like Dyno1, Dyno2, Dyno3, Cash, Paypal, Visa, Mastercard, Debit Visa, etc. The "Payment Type" column shows various payment methods. The "Units" column shows unit counts (e.g., 1, 9, 12). The "Amount of Sale" column shows sales amounts (e.g., 24.95, 258.75, 307.72). The "Revenue Discount" column shows discount percentages (e.g., 0, 0.1, 0.1875).

	Name	Date	Store Zip Code	Product	Payment Type	Units	Amount of Sale	Revenue Discount
1	Glasgow	11/12/2019	G3 8AG	Dyno1	Visa	1	24.95	0
2	Glasgow	20/12/2019	G3 8AG	Dyno2	Paypal	1	28.75	0
3	Glasgow	21/10/2019	ML12 6HD	Dyno2	Visa	1	28.75	0
4	Glasgow	13/11/2019	G3 8AG	Dyno3	Cash	1	43.96	0
5	Glasgow	25/11/2019	G62 8EP	Dyno2	Paypal	9	258.75	0.1
6	Glasgow	07/12/2019	ML12 6HD	Dyno3	Visa	1	43.96	0
7	Glasgow	04/11/2019	G62 8EP	Dyno1	Mastercard	1	24.95	0
8	Glasgow	05/11/2019	G62 8EP	Dyno4	Debit Mastercard	12	59.4	0.1875
9	Glasgow	30/10/2019	G3 8AG	Dyno3	Cash	1	43.96	0
10	Glasgow	15/11/2019	G3 8AG	Dyno3	Paypal	7	307.72	0.1
11	Glasgow	29/10/2019	ML12 6HD	Dyno5	Cash	6	167.7	0.1
12	Glasgow	01/10/2019	G3 8AG	Dyno3	American Express	6	263.76	0.1
13	Glasgow	07/11/2019	G3 8AG	Dyno1	Mastercard	6	149.7	0.1
14	Glasgow	17/11/2019	G3 8AG	Dyno1	Visa	1	24.95	0
15	Glasgow	28/10/2019	G3 8AG	Dyno5	Debit Mastercard	1	27.95	0
16	Glasgow	17/10/2019	G3 8AG	Dyno2	In Store	1	28.75	0
17	Glasgow	01/11/2019	ML12 6HD	Dyno3	Cash	1	43.96	0
18	Glasgow	01/10/2019	ML12 6HD	Dyno5	Visa	12	335.4	0.1875
19	Glasgow	06/10/2019	ML12 6HD	Dyno5	Cash	12	335.4	0.1875
20	Glasgow	26/10/2019	G3 8AG	Dyno2	Cash	4	115	0.1
21	Glasgow	04/10/2019	G62 8EP	Dyno3	Cash	1	43.96	0
22	Glasgow	21/11/2019	G3 8AG	Dyno3	Cash	1	43.96	0
23	Glasgow	07/11/2019	G3 8AG	Dyno1	Debit Visa	6	149.7	0.1
24	Glasgow	20/11/2019	G3 8AG	Dyno3	Debit Visa	12	527.52	0.1875
25	Glasgow	20/12/2019	G62 8EP	Dyno1	Mastercard	4	99.8	0.1

Рис. 8.19. Окончательный список импорта

Финальный список импорта, показанный на рис. 8.19, можно еще немного улучшить.

Дважды щелкните на заголовке первого столбца и переименуйте его из Name в City. Теперь нужно пройти по колонкам и убедиться, что их типы данных соответствуют нашим требованиям. Столбцы City, Store Zip Code, Product и Payment Type содержат текстовую информацию. Чтобы изменить тип данных сразу в нескольких колонках, необходимо выбрать их мышью с зажатой клавишей **Ctrl**, после чего щелкнуть правой кнопкой мыши по любому из выбранных столбцов и выбрать пункт Text (Текст) в подменю Change Type (Тип изменения), как показано на рис. 8.20. В результате тип данных всех выбранных столбцов будет изменен.

The screenshot shows the Power BI data editor with a context menu open over several columns. The menu is titled "Change Type" and includes options like Decimal Number, Fixed decimal number, Whole Number, Percentage, Date/Time, Date, Time, Duration, Text, True/False, Binary, and Using Locale... . The "Change Type" option is highlighted. The columns selected for this change are City, Date, Store Zip Code, Product, and Payment Type, which are part of the "Expanded Data" table.

	City	Date	Store Zip Code	Product	Payment Type	Units	Amount of Sale	Revenue
1	Glasgow	11/12/2019	G3 8AG	Dyno1		1	24.95	
2	Glasgow	20/12/2019	G3 8AG	Dyno2		1	28.75	
3	Glasgow	21/10/2019	ML12 6HD	Dyno2		1	28.75	
4	Glasgow	13/11/2019	G3 8AG	Dyno3		1	43.96	
5	Glasgow	25/11/2019	G62 8EP	Dyno2		9	258.75	
6	Glasgow	07/12/2019	ML12 6HD	Dyno3		1	43.96	
7	Glasgow	04/11/2019	G62 8EP	Dyno1		12	59.4	
8	Glasgow	05/11/2019	G62 8EP	Dyno4		1	43.96	
9	Glasgow	30/10/2019	G3 8AG	Dyno3		1	43.96	
10	Glasgow	15/11/2019	G3 8AG	Dyno3		1	43.96	
11	Glasgow	29/10/2019	ML12 6HD	Dyno5		6	167.7	
12	Glasgow	01/10/2019	G3 8AG	Dyno3		6	263.76	
13	Glasgow	07/11/2019	G3 8AG	Dyno1		6	149.7	
14	Glasgow	17/11/2019	G3 8AG	Dyno1		1	27.95	
15	Glasgow	28/10/2019	G3 8AG	Dyno5		12	335.4	
16	Glasgow	17/10/2019	G3 8AG	Dyno2		4	115	
17	Glasgow	01/11/2019	ML12 6HD	Dyno3		1	43.96	
18	Glasgow	01/10/2019	ML12 6HD	Dyno5		1	24.95	
19	Glasgow	06/10/2019	ML12 6HD	Dyno5		1	28.75	
20	Glasgow	26/10/2019	G3 8AG	Dyno2		1	43.96	
21	Glasgow	04/10/2019	G62 8EP	Dyno3		7	307.72	
22	Glasgow	21/11/2019	G3 8AG	Dyno3		7	307.72	
23	Glasgow	07/11/2019	G3 8AG	Dyno1		12	527.52	
24	Glasgow	20/11/2019	G3 8AG	Dyno3		12	527.52	
25	Glasgow	20/12/2019	G62 8EP	Dyno1		4	99.8	

Рис. 8.20. Изменение типа данных столбцов

Второму столбцу необходимо аналогичным образом дать тип Date (Дата), а колонке Units присвоим тип Whole Number (Целое число).

The screenshot shows the Microsoft Power BI Data Editor interface. On the left, there's a table with columns labeled 'Units' (containing values like 1, 1, 1, 1, 9, 1, 1, 1, 12, 1, 7, 6, 6, 1, 1, 1, 1, 1, 12, 12, 4) and 'COGS' (containing values like 0, 0, 0, 0, 0.1, 0, 0, 0.1875, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0). A context menu is open over the 'COGS' column, specifically over the second row. The menu items include: Copy, Remove Columns, Remove Other Columns, Add Column From Examples..., Remove Duplicates, Remove Errors, Replace Values..., Fill, Change Type, Transform, Merge Columns, Sum, Product, Group By..., Unpivot Columns, Unpivot Other Columns, Unpivot Only Selected Columns, Move, Decimal Number, Fixed decimal number (which is highlighted in grey), Whole Number, Percentage, Date/Time, Date, Time, Date/Time/Timezone, Duration, Text, True/False, Binary, and Using Locale... The 'Fixed decimal number' option is currently selected.

Units	COGS
1	0
1	0
1	0
1	0
9	13.54
1	14.82
1	15.54
1	22.93
12	127.71
1	21.46
1	13.14
1	39.24
7	21.4
6	50.86
6	77.52
1	33.56
1	32.62
1	15.32
1	13.69
1	14.87
12	22.14
12	49.04
4	57.04
1	51.72
1	22.79
6	22.58
12	36.52
12	56.68

Рис. 8.21. Установка типа данных десятичное число с фиксированной запятой

Последние три столбца в нашей таблице хранят финансовую информацию, но в Power BI этот тип соответствует десятичному числу с фиксированной запятой, установить который можно так, как показано на рис. 8.21. Одно из главных различий между типами данных **Decimal Number** (Десятичное число) и **Fixed decimal number** (Десятичное число с фиксированной запятой) состоит в том, что во втором варианте хранится только четыре знака после запятой, тогда как в первом – до 15 знаков. Чтобы быстро выделить последние три столбца, можно выбрать первый из них, зажать клавишу **Shift** и выбрать последний.

Теперь нам необходимо дополнить нашу таблицу новым столбцом, в котором будет рассчитываться цена товара со скидкой.

Для этого на вкладке **Add Column** (Добавление столбца) нажмите на кнопку **Custom Column** (Настраиваемый столбец). Назовите новый столбец **NetSales**, а в поле для формулы введите следующую формулу: `Number.Round([Amount of Sale]*(1-[Revenue Discount]), 2)`, как показано на рис. 8.22.

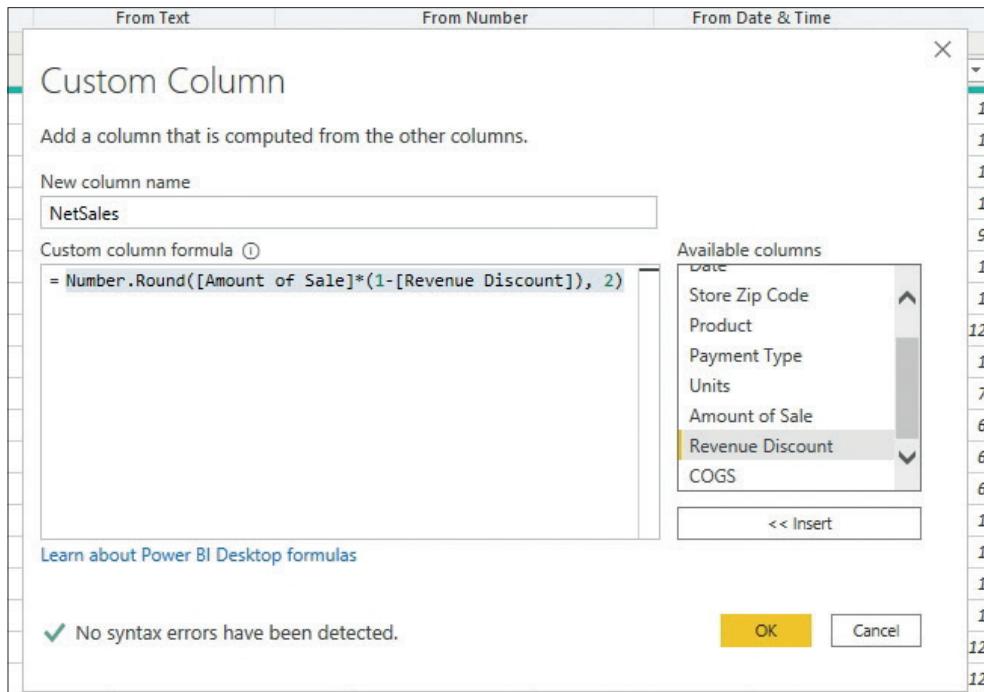


Рис. 8.22. Добавление настраиваемого столбца **NetSales**

Здесь мы умножаем значение из поля **Amount of Sale** на скидку, после чего округляем полученный результат до двух знаков после запятой.

После этого нажмите на кнопку **OK** и измените тип поля на **Fixed Decimal Number** (Десятичное число с фиксированной запятой).

Нам больше не нужны столбцы **Amount of Sale** и **Revenue Discount**, так что от них можно избавиться. Итак, таблица готова, и мы можем загружать ее в модель данных, выбрав пункт **Close and Apply** (Закрыть и применить) на вкладке **Home** (Главная страница).

На данный момент мы полностью подготовили нашу таблицу, удалили ненужные столбцы и добавили новые – с полезной информацией. Как мы уже говорили, несмотря на то что в нашем примере мы использовали файлы из одной папки, они также могут располагаться онлайн, в SQL Server или на сайте.

Теперь мы можем использовать наш запрос в модели данных, а значит, первый шаг завершен. В следующем разделе мы приступим ко второму шагу процесса.

Использование Power BI для добавления таблиц в модель данных

В данном разделе мы затронем вопросы создания связей между таблицами и добавления расчетных данных, в том числе мер и вычисляемых столбцов, при помощи языка DAX, позволяющего создавать и модифицировать дополнительные колонки, которые в дальнейшем понадобятся нам на дашборде.

Если посмотреть на созданный нами в предыдущем разделе запрос в Power BI, мы увидим в правой части таблицу SalesData с полным набором полей, как показано на рис. 8.23.

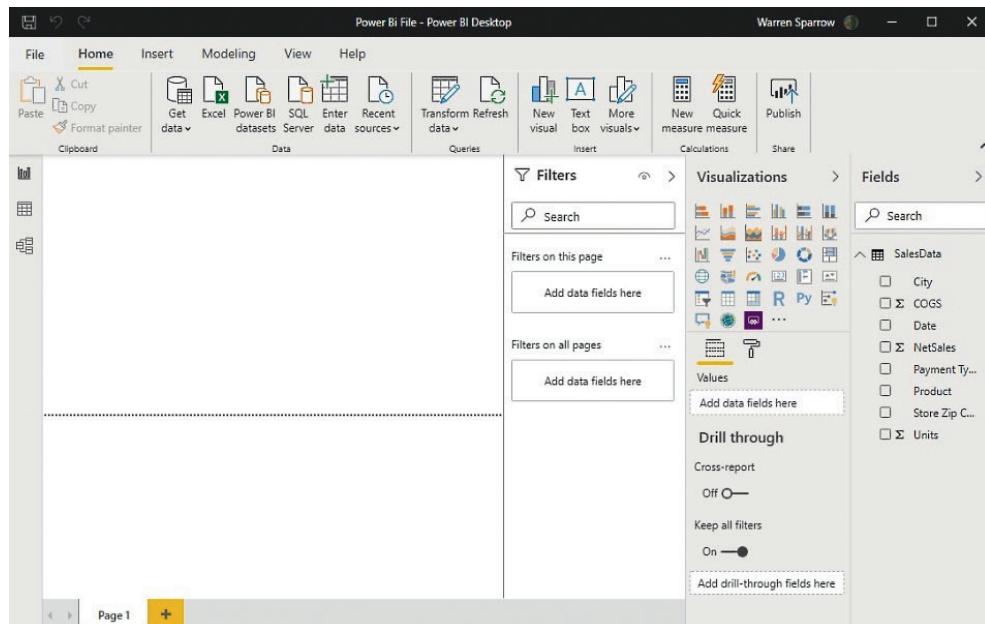


Рис. 8.23. Таблица SalesData

При ближайшем рассмотрении можно обнаружить значки Σ слева от нескольких полей, которые говорят о том, что это числовые поля. В левой части окна располагаются кнопки-иконки Report (Отчет), Data (Данные) и Model (Модель), что видно по рис. 8.24.

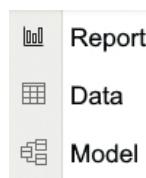


Рис. 8.24. Варианты представлений в Power BI

Открывая Power BI, мы по умолчанию оказываемся на вкладке **Report** (Отчет).

Щелкнув на вкладке **Model** (Модель), мы попадем в представление модели данных со всеми связями, настроенными между таблицами. В данный момент в нашей модели данных связей нет, но мы собираемся добавить еще одну табличку и установить связь с файлом с сервера SharePoint, хотя те же шаги вам понадобится выполнить и для SQL Server или сервера Azure. Для начала перейдите на вкладку **Home** (Главная страница), нажмите на кнопку **Get Data** (Получить данные), выберите пункт **More...** (Другие...), найдите источник данных **SharePoint folder** (Папка SharePoint) и нажмите на кнопку **Connect** (Подключить).

Далее вам необходимо будет ввести данные аутентификации и нажать на кнопку **Connect** (Подключить). В главе 2 мы подробно рассказывали о подключении к разным источникам данных.

После выбора файла укажите нужную вам таблицу и нажмите на кнопку **Transform Data** (Преобразовать). Возможно, вам понадобится очистить файл, удалив ненужные столбцы, перед его загрузкой. По завершении работ нажмите **Close and Apply** (Закрыть и применить).

Теперь, перейдя на вкладку **Model** (Модель), мы увидим уже две таблицы, одна из которых является таблицей фактов со всеми транзакциями по продаже, а вторая – измерением с менеджерами магазинов.

Создадим связь между таблицами по полям **Store Zip Code** и **Zip Codes**. Нажмите левой кнопкой мыши на поле **Store Zip Code** и, удерживая кнопку, перенесите его на поле **Zip Codes** в таблице **ManagerTable**.

В результате будет создана связь типа «один ко многим» между таблицами, что видно по рис. 8.25. Цифра 1 на связи указывает на сторону «один», а звездочка – на сторону «многие». Это означает, что одной строке из таблицы **ManagerTable** может соответствовать сколько угодно строк в таблице **SalesData** по полю индекса.

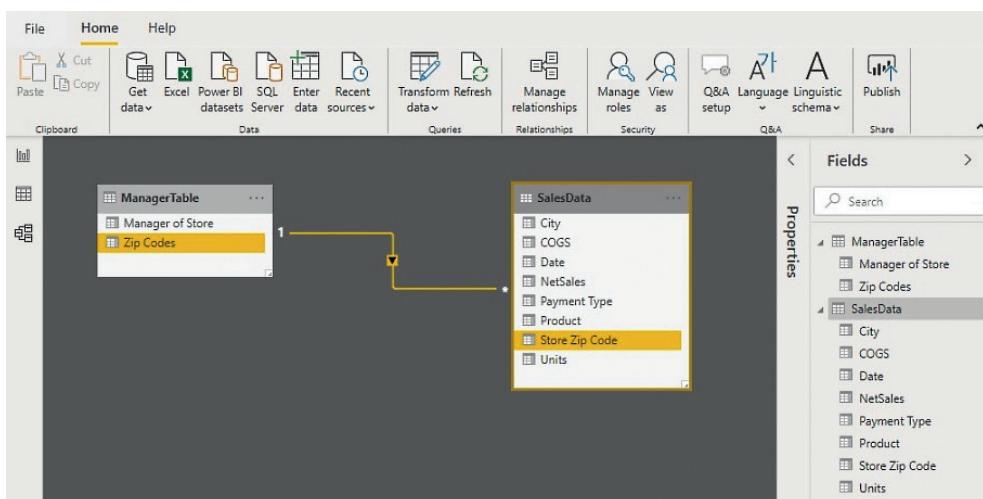


Рис. 8.25. Связь «один ко многим»

Если перейти на вкладку **Data** (Данные), мы увидим таблицы и поля в нашей модели данных. Давайте попробуем добавить столбец к нашей таблице **SalesData**. Значение этого столбца будет устанавливаться в **Retail** или **Wholesale** в зависимости от объема продажи в штуках.

Существует два способа создания столбца. Вы можете нажать правой кнопкой мыши на таблице **SalesData** в правой панели и выбрать пункт **New column** (Создать столбец), как показано на рис. 8.26. Также можно нажать кнопку с таким же названием на панели инструментов на вкладке **Table Tools** (Средства работы с таблицами).

City	Date	Store Zip Code	Product	Payment Type	Units	COGS	NetSales
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£3.07	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£2.89	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£3.25	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£2.28	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£3.22	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£3.53	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£3.99	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£4.09	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£3.01	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£2.29	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£3.93	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£3.64	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£3.93	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£2.77	£4.95
Liverpool	01/10/2019	WA11 7SR	Dyno2	Visa	1	£2.57	£4.95

Рис. 8.26. Добавление столбца

Примечание

При написании книги мы пользовались последней версией Power BI Desktop, выпущенной в апреле 2020 года. Если вы используете более старую версию программы, вместо вкладки **Table Tools** (Средства работы с таблицами) может быть вкладка **Modeling** (Моделирование).

В строке формул автоматически появится выражение **Column =**. Мы используем функцию **IF**, которая будет проверять значение в поле **Units** и в зависимости от него возвращать текст **Retail** или **Wholesale**.

Итак, удалите начало формулы **Column =** и впишите на его место следующее выражение:

```
RetailWholesale = IF(SalesData[Units]<5, "Retail", "Wholesale")
```

При вводе формулы с клавиатуры после нажатия на клавишу **S** вслед за открывающейся скобкой вам будут предложены на выбор поля из таблицы

SalesData. Вы можете выбрать из списка поле **Units**, ведь именно его мы собираемся использовать.

Исходя из нашего условия понятно, что в новом поле **RetailWholesale** появится значение **Retail**, если мы продаем меньше пяти штук, и значение **Wholesale** – если пять и более. Эта формула базируется на контексте строки, а значит, в расчетах будет учитываться значение столбца **Units** только для текущей строки.

На рис. 8.27 можно увидеть слева от названия поля **RetailWholesale** значок *fx*, говорящий о том, что в формуле столбца присутствует функция.

The screenshot shows the Power BI Desktop interface with the 'Column tools' tab selected in the ribbon. A table named 'SalesData' is open, displaying several rows of data. A new column, 'RetailWholesale', has been added, containing values 'Retail' or 'Wholesale' based on the value in the 'Units' column. The formula for this column is visible in the formula bar: `1 RetailWholesale = IF(SalesData[Units]<5, "Retail", "Wholesale")`. The 'Fields' pane on the right side of the interface lists the columns in the table, including the newly created 'RetailWholesale' measure.

Рис. 8.27. Вычисляемый столбец

Теперь у нас есть все колонки, которые нужны. В следующем разделе мы создадим дополнительные меры, в которых также будем производить математические расчеты.

Сейчас же мы добавим специфические меры в таблицу SalesData. Меры обычно используются для расчета агрегированных значений – например, для вычисления чистого дохода, валовой прибыли, суммы или среднего значения по столбцу и т. д. Меры, в отличие от вычисляемых столбцов, рассчитываются лишь в момент выполнения запроса, таким образом не являясь частью сохраненной базы данных. Меры также допустимо использовать в вычислениях и визуализациях в модели данных.

Меры, которые мы создадим применительно к этой таблице, – GrossProfit, GrossProfit%, TotalCOGSold и TotalRevenue.

На вкладке **Table Tools** (Средства работы с таблицами) нажмите на кнопку **New Measure** (Создать меру) и введите в строке формул следующее: `TotalRevenue = SUM(SalesData[NetSales])`, как показано на рис. 8.28. Перед тем как нажать на клавишу **Enter**, измените тип меры на валютный и установите количество знаков после запятой в 0. Так же точно мы создадим меру `TotalCOGSold` с формулой `TotalCOGSold = sum(SalesData[COGS])`.

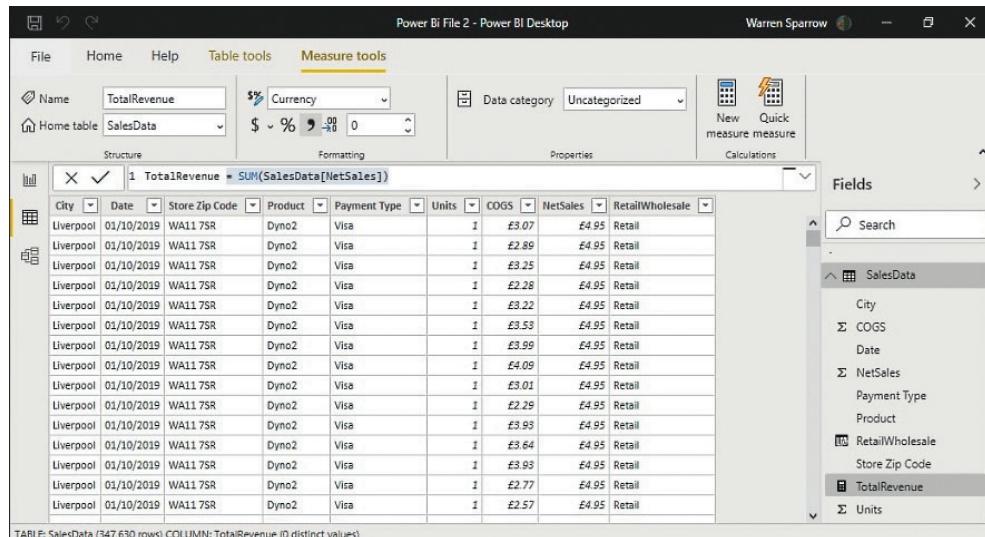


Рис. 8.28. Создание меры TotalRevenue

Создайте еще одну меру GrossProfit с формулой $GrossProfit = [TotalRevenue] - [TotalCOGSold]$. И последнюю – GrossProfit% с формулой $GrossProfit\% = [GrossProfit] / [TotalRevenue]$. Для нее нужно выбрать процентный тип и установить два знака после запятой.

На данный момент у нас в модели данных есть две таблицы ManagerTable и SalesData, вычисляемый столбец RetailWholesale и четыре созданные меры: GrossProfit, GrossProfit%, TotalCOGSold и TotalRevenue.

Таким образом, у нас есть все необходимое для создания дашборда. В следующем разделе мы именно этим и займемся.

ВЫБОР ТИПА ВИЗУАЛИЗАЦИИ ДАННЫХ И ПОДХОДЯЩЕЙ ДИАГРАММЫ

В предыдущем разделе мы импортировали всю нужную нам информацию в модель данных Power BI. Здесь мы сосредоточимся на построении интерактивного дашборда, на котором визуально отобразим данные для дальнейшего анализа.

Перейдите в Power BI на вкладку **Report** (Отчет) в левом меню, после чего перед вами откроется пустое пространство, доступное для создания визуализации. В правой панели выстроены иконки типов визуализаций, которые вы можете использовать при проектировании дашборда. Эта панель показана на рис. 8.29.

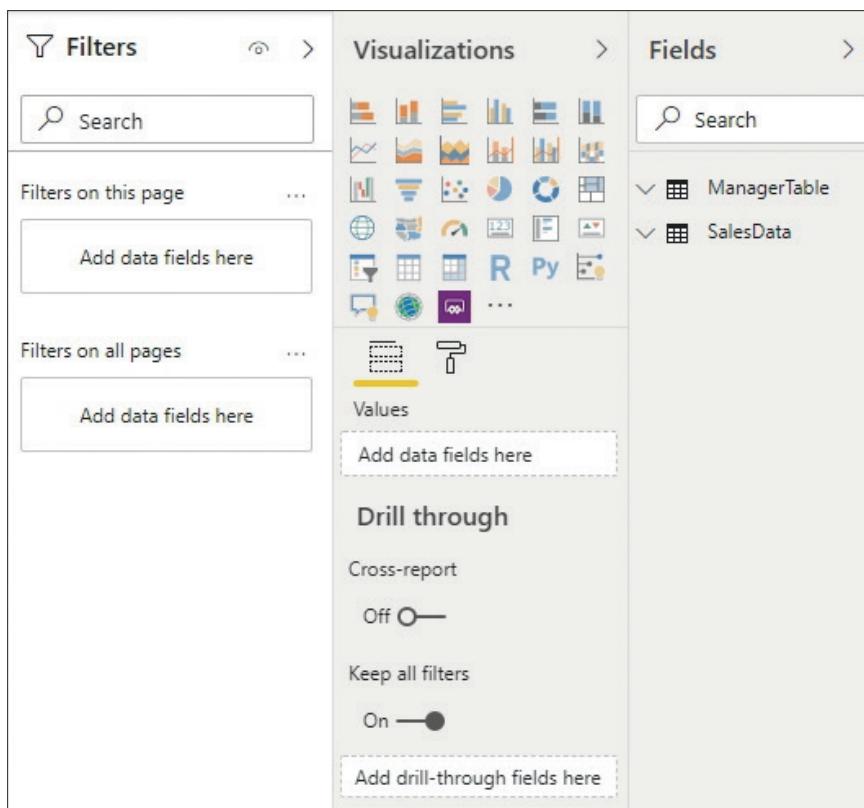


Рис. 8.29. Панель визуализаций

Многие из предложенных иконок говорят сами за себя и не нуждаются в дополнительных пояснениях. Мы будем использовать разные типы визуализаций и вместе узнаем, как они работают.

Примечание

Наполнение окна с иконками визуализаций зависит от версии используемой вами программы Power BI Desktop. Кроме того, некоторые иконки могут у вас выглядеть иначе, чем на представленных рисунках. Мы при написании книги используем последнюю версию Power BI Desktop от апреля 2020 года.

Начнем с одного из наиболее потрясающих типов визуализаций, представляющих собой географическую карту с точным указанием мест происхождения ваших данных.

1. Щелкните на иконке **Map** (Карта) в виде глобуса, и заготовка карты появится на странице в верхнем левом углу.
2. Чтобы привязать к карте данные, выберите поле **Zip Codes** в таблице **ManagerTable** и **GrossProfit** в таблице **SalesData**, как показано на рис. 8.30.

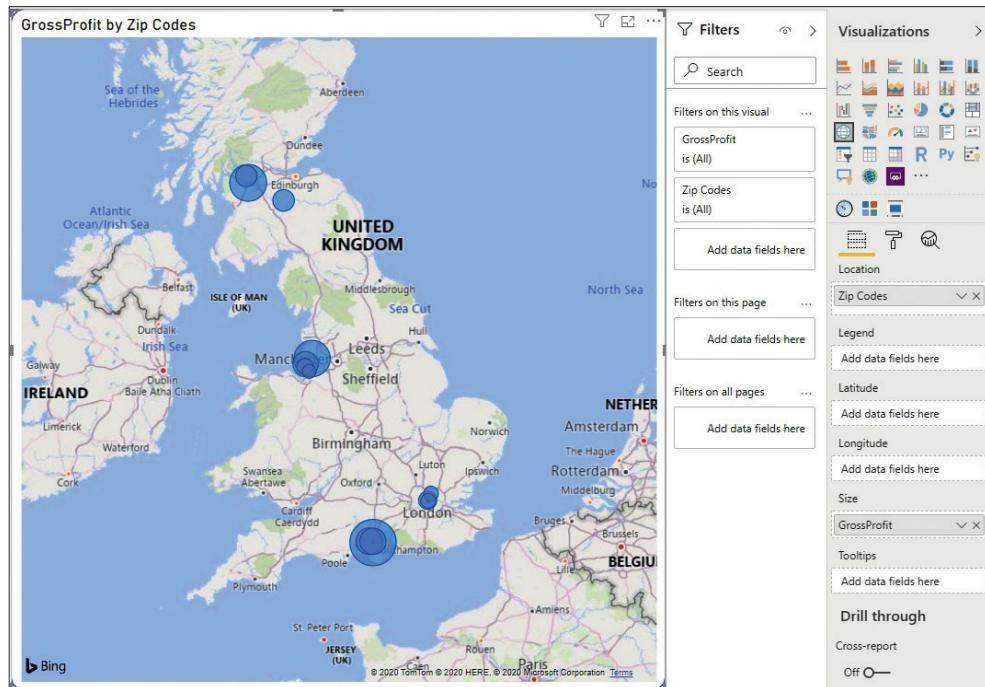


Рис. 8.30. Визуализация в виде географической карты

В результате мы увидим валовую прибыль по разным точкам в зависимости от индексов. При этом чем больше будет отображаемый круг на диаграмме, тем выше валовая прибыль. При беглом взгляде на отчет становится понятно, что есть области, в которых с продажами полный порядок, а именно Саутгемптон, Манчестер и Глазго, в то время как в Лондоне дела идут нелучшим образом.

В выборе формата отображения диаграммы вы можете испытывать абсолютную свободу, простор для творчества здесь невероятно велик:

- если щелкнуть по иконке **Formatting** (Формат) в виде малярного валика на панели **Visualizations** (Визуализации), можно, к примеру, изменить размеры кружков на диаграмме или добавить границы;
- нажмите на панели **Visualizations** (Визуализации) на иконку **Clustered bar chart** (Линейчатая диаграмма с группировкой) и выберите поле **Manager of Store** из таблицы **ManagerTable** и **GrossProfit** из таблицы **SalesData**. Появится соответствующий элемент визуализации, как показано на рис. 8.31;



Рис. 8.31. Линейчатая диаграмма с группировкой

- в уже знакомом нам разделе **Formatting** (Формат) вы можете изменить цветовую палитру диаграммы. Это поможет вам лучше отличать подразделения друг от друга или, к примеру, делать различия по размеру комиссии. Интерактивность дашборда заключается в том, что если щелкнуть по конкретному продавцу, карта преобразуется таким образом, чтобы отобразить его расположение;
- к примеру, если выбрать продавца **Sallim Simmonds**, географическая карта изменит масштаб и покажет Лондон, что видно по рис. 8.32.

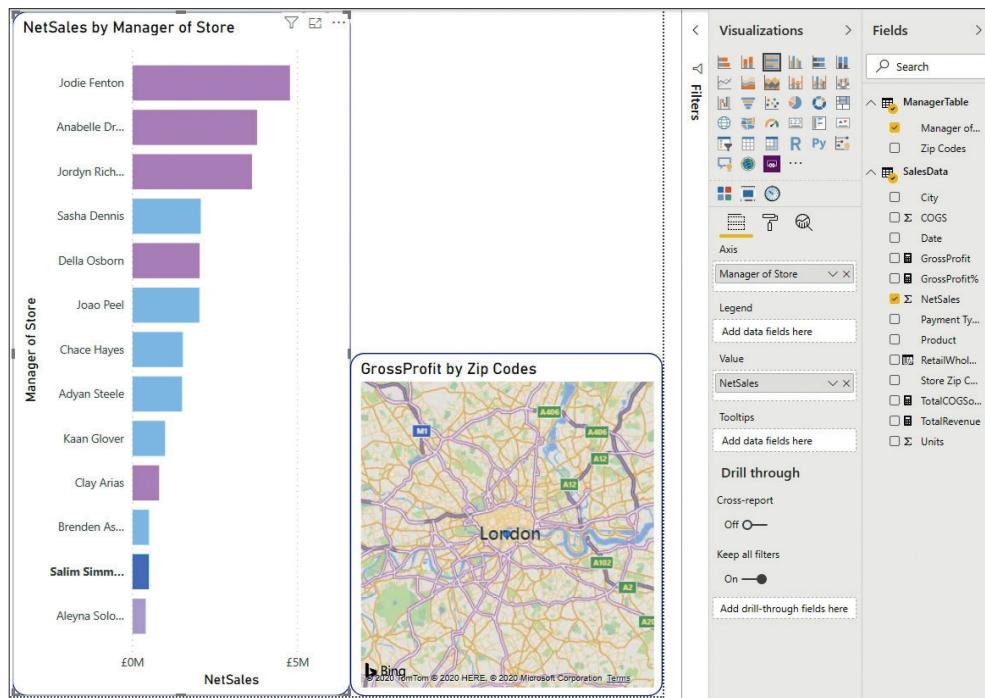


Рис. 8.32. Почтовый индекс продавца Sallim Simmonds

Нажмите на типе визуализации **Line and clustered column chart** (График и гистограмма с группировкой) и выберите поля **Product**, **GrossProfit** и **GrossProfit%** из таблицы **SalesData**.

Вы заметите, что в данный момент диаграмма отображается не так, как предыдущие, поскольку оба поля находятся в разделе **Column values** (Значения столбцов). Перенесите мышью поле **GrossProfit%** в раздел **Line values** (Значения строк), после чего будут показаны две разные диаграммы.

Создайте еще один элемент визуализации с графиком и гистограммой с группировкой, но на этот раз выберите поля **Payment Type**, **GrossProfit** и **GrossProfit%** из таблицы **SalesData**. Снова перенесите поле **GrossProfit%** в раздел **Line values** (Значения строк). Можно даже переносить поля непосредственно из списка на нужную позицию. Измените размеры и расположение элементов на дашборде, как показано на рис. 8.34.

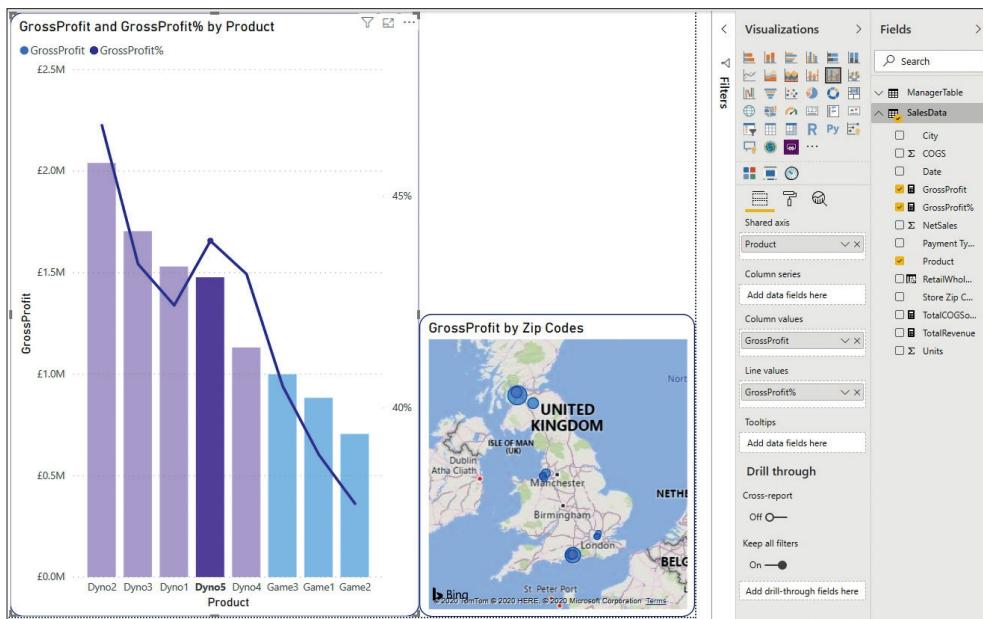


Рис. 8.33. Карта и график с гистограммой

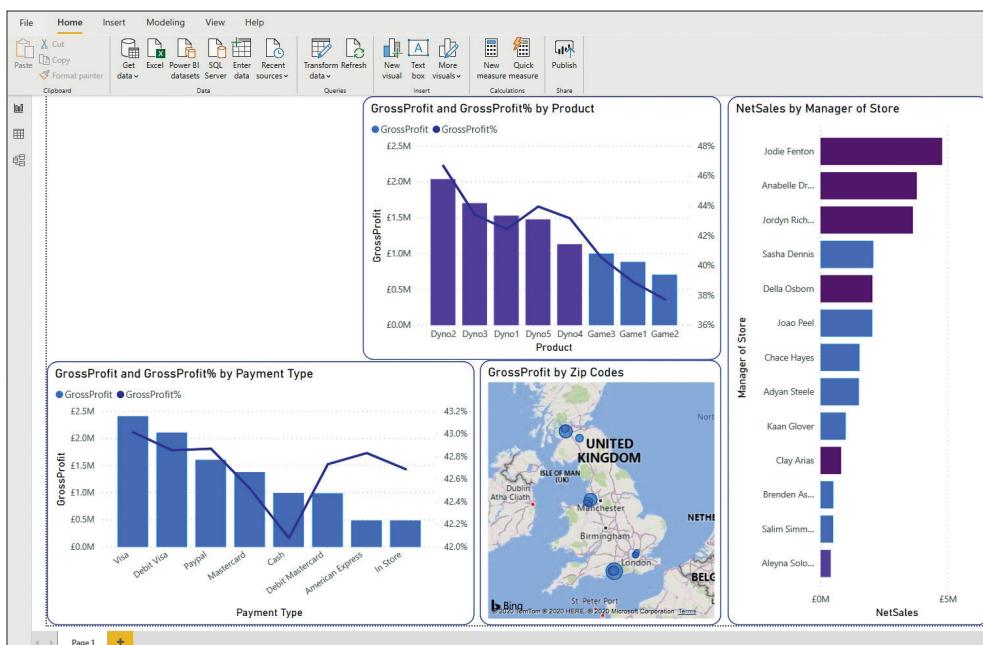


Рис. 8.34. Дашборд с двумя графиками с гистограммами

Теперь давайте создадим несколько срезов, которые работают по тому же принципу, что и в сводных таблицах в Excel. Для начала добавим срез по городам.

Щелкните на иконке **Slicer** (Срез) на панели **Visualizations** (Визуализации) и выберите поле **City** из таблицы **SalesData**. Измените размер и расположение нового элемента на дашборде. Создайте еще один срез – на этот раз по полю **Payment Types** из таблицы **SalesData**, как показано на рис. 8.35.

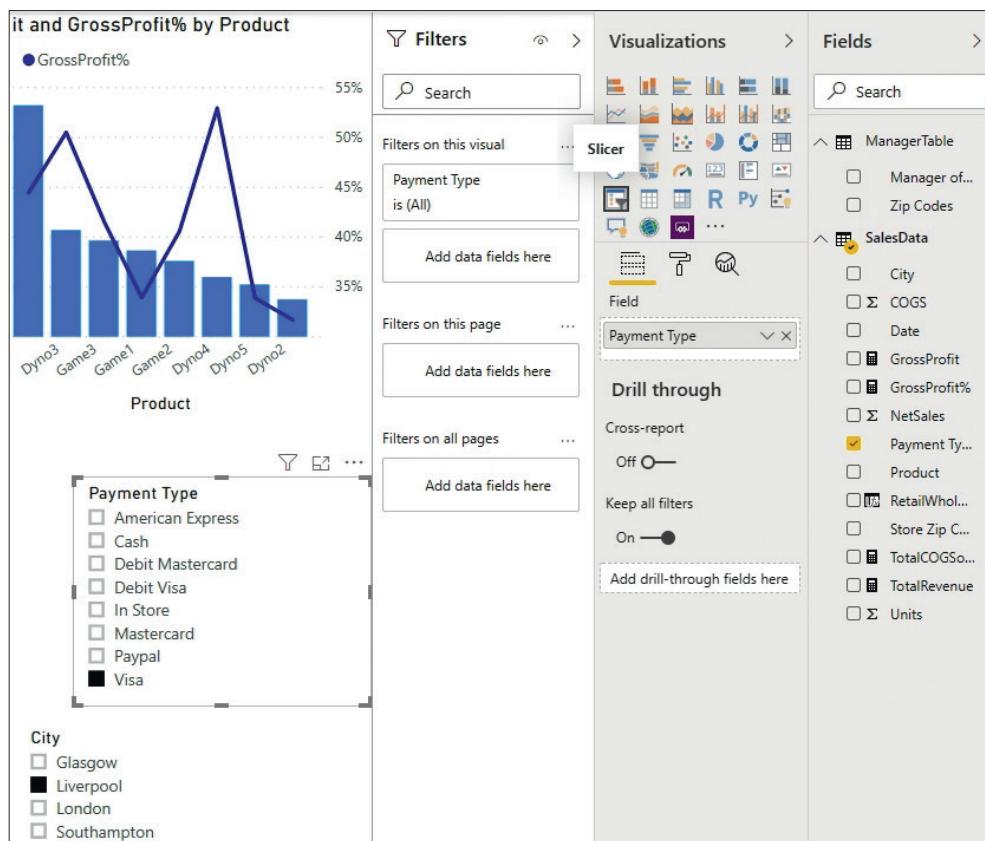


Рис. 8.35. Срезы на дашборде

Давайте еще немного улучшим наш дашборд за счет многострочной карточки и элемента «Вопросы и ответы»:

- щелкните на элементе визуализации **Card** (Карточка) и выберите поле **TotalRevenue**. Поменяйте шрифт, цвет и размер таким образом, чтобы карточка выделялась из элементов на дашборде;
- многострочная карточка представляет из себя обычную карточку, за исключением того, что может отображать одновременно несколько полей;

- найдите на панели визуализаций элемент *Multi-row card* (Многострочная карточка) и выберите три меры, созданные нами ранее. На вкладке форматирования вы можете изменить цвет фона и меток данных, а также другие опции отображения.

Последним элементом визуализации, который мы добавим на дашборд, будет *Q&A* (Вопросы и ответы). Мне этот элемент кажется исключительно любопытным, поскольку он позволяет вам задать любой вопрос о ваших таблицах и полях и получить ответ. Сразу отметим, что по умолчанию значение ответа будет округлено в большую или меньшую сторону для более комфорtnого отображения. Например, если задать вопрос «*What is the total revenue?*» (Каков совокупный доход?), ответ будет 24 млн, хотя точное значение равно 24,489472 млн.

Выберите на панели визуализаций элемент **Q&A** (Вопросы и ответы) и введите вопрос «*What is Netsales by city?*», как показано на рис. 8.36.

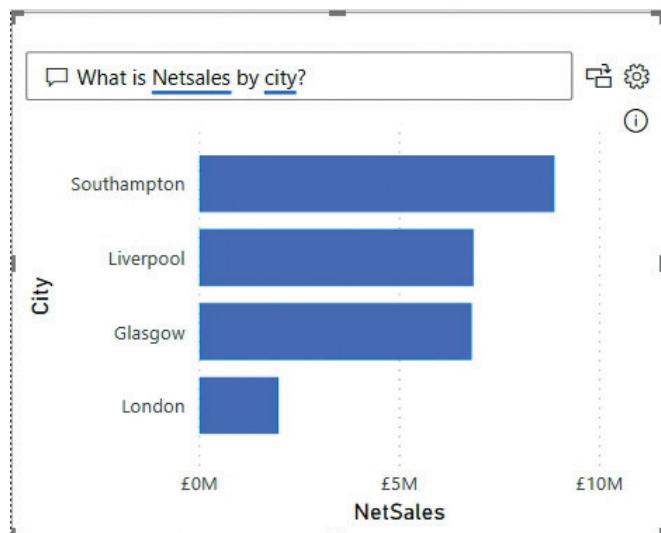


Рис. 8.36. Элемент визуализации с вопросом и ответом

Теперь можно нажать на иконку слева от вопроса с двумя окошками и стрелкой, чтобы преобразовать наш элемент в стандартный тип визуализации по выбору. Взгляните на наш первый дашборд, показанный на рис. 8.37.

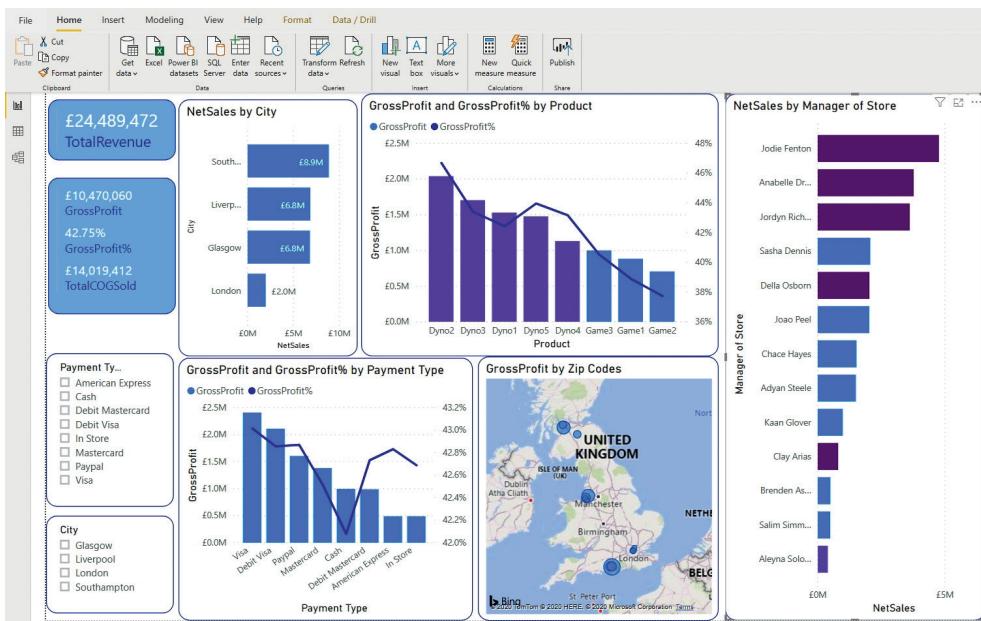


Рис. 8.37. Готовый дашборд

Мы завершили работу над первым дашбордом. Пощелкайте по нему, и вы почувствуете всю его мощь. Каждый раз при выделении элемента для него будет создаваться фильтр. Например, вы можете выбрать товар и посмотреть по нему совокупный доход, сумму продаж по городам и процент валовой прибыли, все в режиме онлайн.

В следующем разделе мы рассмотрим различные варианты сохранения дашбордов.

СОХРАНЕНИЕ, ПУБЛИКАЦИЯ И РАСПРОСТРАНЕНИЕ ДАШБОРДОВ

В данном разделе мы затронем вопросы *сохранения и публикации интерактивных дашбордов*. Перед началом вам необходимо понимать, что после публикации дашборда в сети люди смогут видеть ваши данные и запускать отчеты без какой-либо аутентификации, так что заранее убедитесь, что политика компании допускает это.

Чтобы опубликовать дашборд, необходимо выполнить следующие действия.

- На вкладке **File** (Файл) выберите подменю **Publish** (Опубликовать) и нажмите на пункт **Publish to Power BI** (Публикация в Power BI).
- Выберите расположение, как показано на рис. 8.38. На вкладке **Home** (Главная страница) также есть кнопка публикации, предоставляющая самый короткий путь к выполнению этого действия.

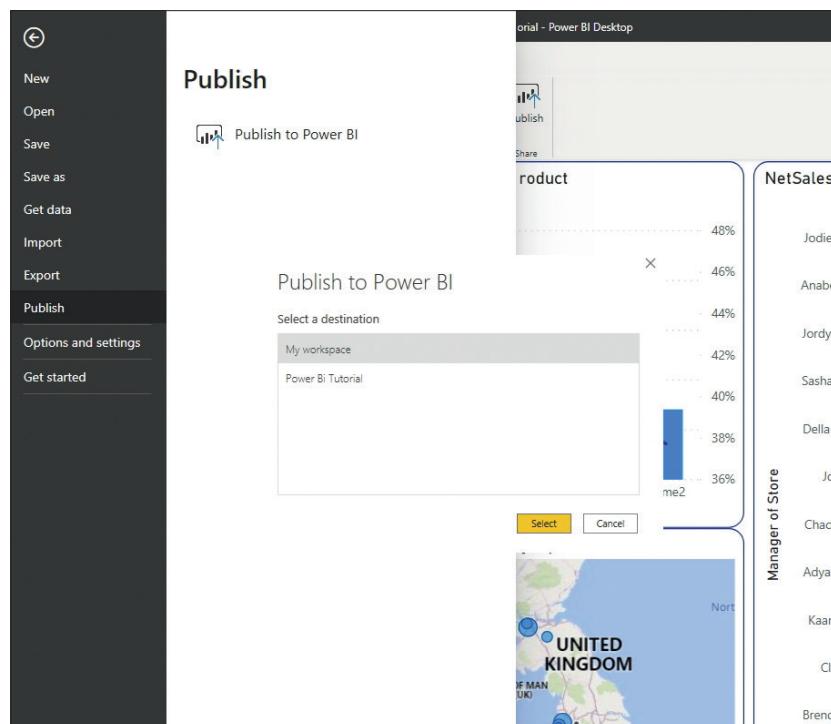


Рис. 8.38. Публикация дашборда в Power BI

3. По умолчанию у вас будет отображаться только вариант **My workspace** (Моя рабочая область).

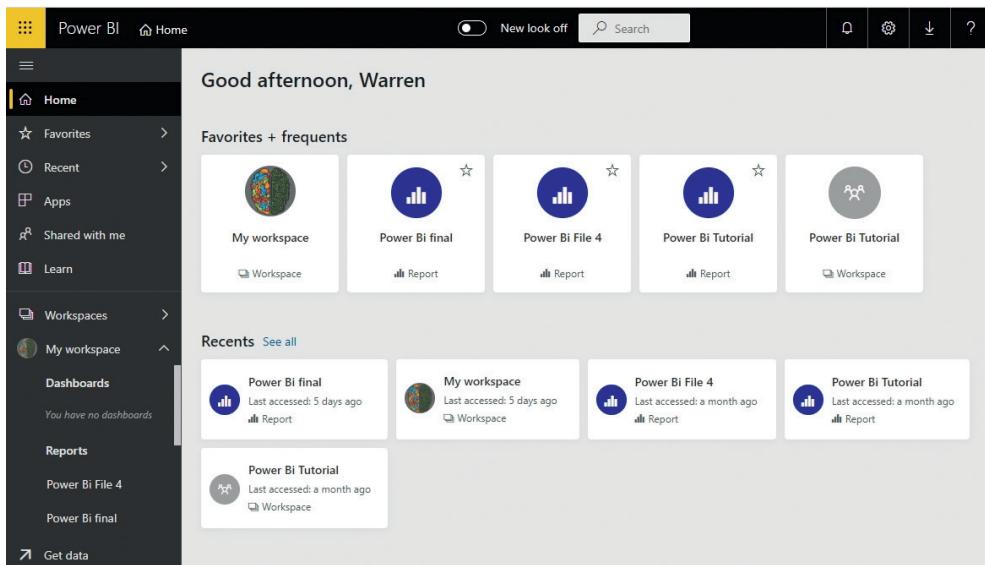


Рис. 8.39. Портал Power BI Online

4. Нажмите на кнопку **Select** (Выбрать), и через несколько секунд будет выдано сообщение об успешном завершении публикации.
5. Закройте окно кнопкой **OK**.

Теперь ваш дашборд опубликован в сети, но пока не очень понятно, где именно.

У вас есть два варианта в зависимости от подписки. Откройте адрес <http://office.com> или <https://powerbi.microsoft.com/en-us/landing/signin> и авторизуйтесь под своим аккаунтом Microsoft. Это должна быть та же учетная запись, под которой вы авторизовались в Power Query.

Войдя в систему, вы сразу попадете на страницу своей рабочей области, где хранятся все ваши данные и отчеты. Свои опубликованные файлы вы можете обнаружить в двух разных местах, как видно на рис. 8.40.

The screenshot shows the Microsoft Power BI workspace interface. On the left, there is a dark sidebar with navigation links: Home, Favorites, Recent, Apps, Shared with me, Learn, Workspaces, and My workspace. Under 'My workspace', there are sections for Dashboards, Reports, Workbooks, and Datasets. The 'Datasets' section shows one item: 'Power Bi File 4'. The main area displays two tables of published files.

Dashboards					
	NAME ↑	ENDORSEMENT	ACTIONS	REFRESHED	NEXT REFRESH
	Power Bi File 4			02/05/2020, 11:07:31	N/A
	Power Bi final			25/06/2020, 17:13:35	N/A
	Sample file1			11/04/2020, 16:13:59	N/A

Reports					
	NAME ↑		ACTIONS	OWNER	
	☆ Power Bi File 4			Warren Sparrow	
	☆ Power Bi final			Warren Sparrow	

Рис. 8.40. Файлы Power BI

Если щелкнуть на пункт меню **Datasets** (Наборы данных), вы перейдете к вашим актуальным наборам, хранящим информацию для анализа. Здесь вы в зависимости от прав доступа можете выполнить различные действия с источниками данных, как показано на рис. 8.41.

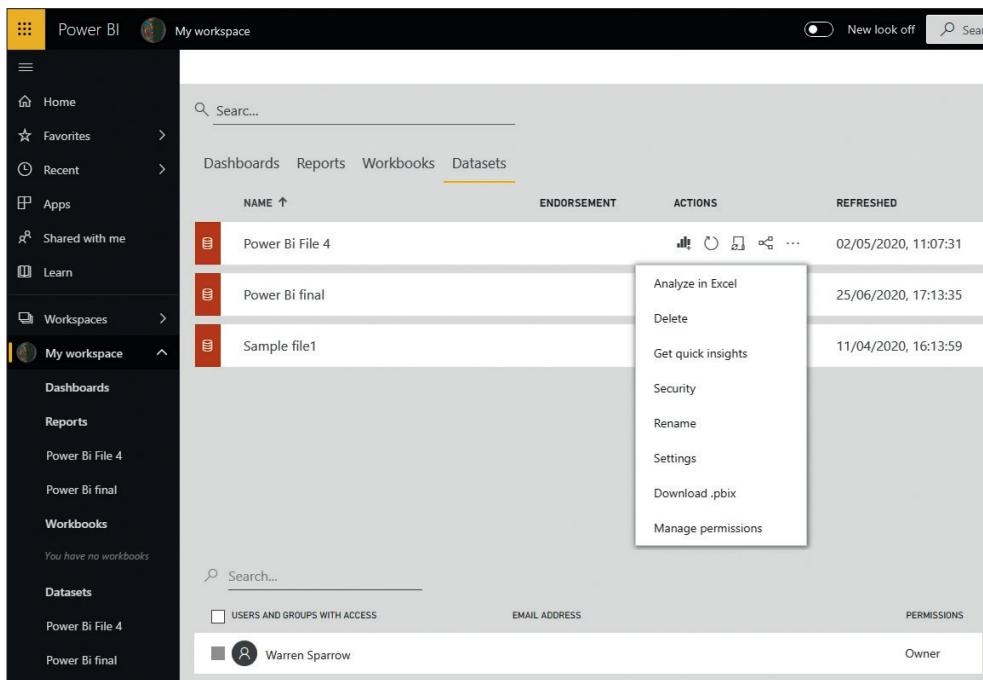


Рис. 8.41. Действия с наборами данных

И хотя вы можете впоследствии послать эти файлы другим людям или дать к ним доступ, такой способ распространения данных нельзя назвать идеальным. Одной из причин является то, что при любом обновлении исходных данных вам придется заново посыпать их коллеге.

У вас может быть возможность давать доступ к своим источникам данных всем желающим. Но этот вариант допустим только при наличии лицензии Pro.

На вкладке **Reports** (Отчеты) вам доступно больше действий, которые вы можете выполнить применительно к вашим отчетам, что видно по рис. 8.42.

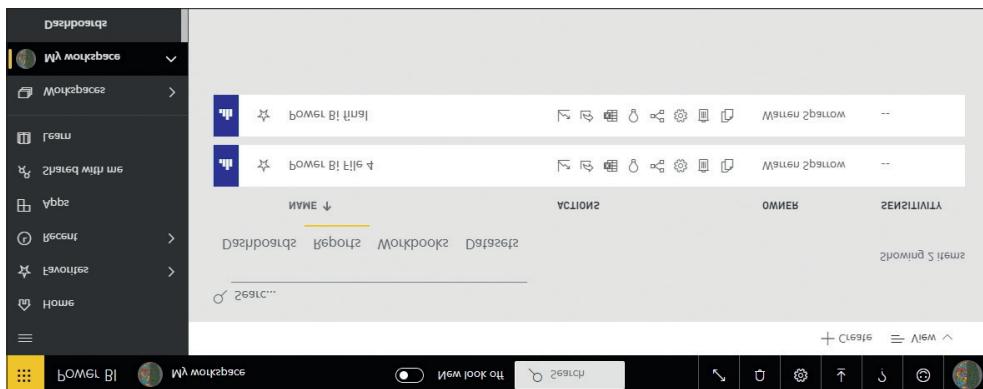


Рис. 8.42. Вкладка отчетов

В разделе действий можно предоставлять доступ к отчетам, анализировать, просматривать разные версии, менять настройки, удалять и создавать дубликаты отчетов. Набор доступных для вас действий зависит, опять же, от типа лицензии.

В следующем разделе мы поговорим о предоставлении доступа к вашим отчетам людям внутри и вне вашей организации.

Распространение дашбордов

Делиться можно много чем: отчетами, дашбордами и данными. Также может меняться уровень распространения данных. Базовая версия позволяет обмениваться начальным уровнем информации, тогда как профессиональная обеспечивает дополнительную функциональность.

Если нажать на кнопку **Pro trial** (Попробовать версию Pro бесплатно), вы получите возможность пользоваться полной версией программы в течение 60 дней без ввода данных о банковской карте. Переход на версию Pro позволит вам значительно расширить функционал. Одним из главных преимуществ этой версии является возможность делиться отчетами сразу с несколькими людьми. Опции распространения отчетов показаны на рис. 8.43.

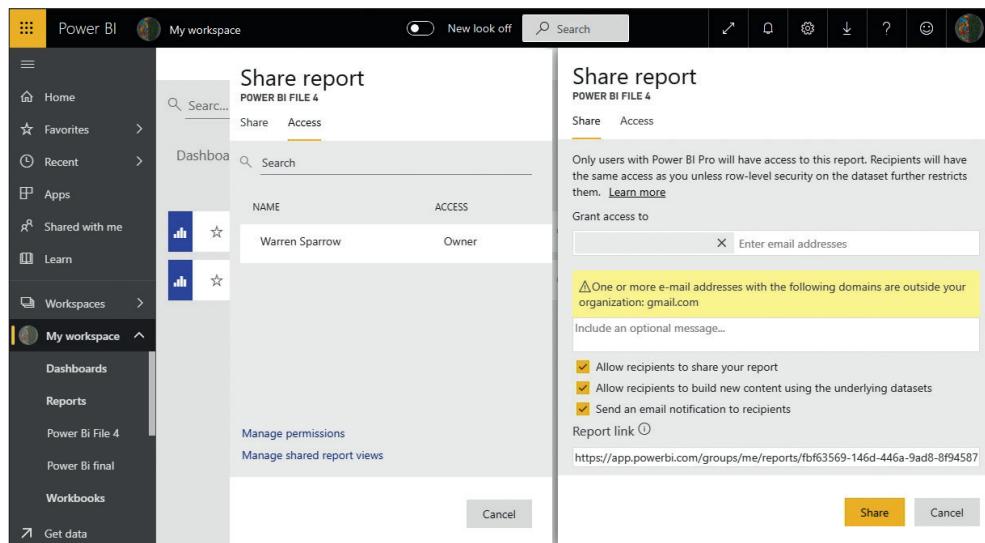


Рис. 8.43. Опции распространения данных

Если нажать на кнопку **Share** (Поделиться) в верхнем правом углу окна, можно настроить опции распространения таким образом, чтобы те, с кем вы поделились отчетом, могли или не могли дальше распространять его и позволять другим людям использовать ваши наборы данных при построении новых отчетов.

Примечание

Учтите, что другие люди могут использовать ваши данные в своих наборах, – и если вы этого хотите, то должны предоставить им такую возможность.

После того как доступ к отчету был дан, можно настроить уровень доступа, перейдя на вкладку **Access** (Доступ) и нажав на троеточие справа от имени. Вы можете как удалить доступ для пользователя, так и выбрать между уровнями только на чтение и на чтение и распространение, как показано на рис. 8.44.

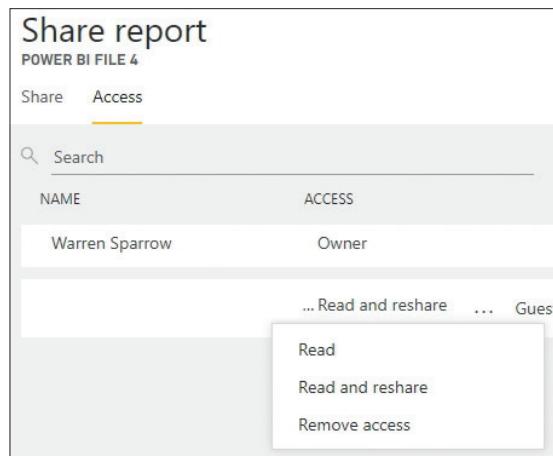


Рис. 8.44. Опции доступа к данным

Еще одним очень быстрым способом поделиться отчетом с другими людьми является создание QR-кода. Если развернуть кнопку **Share** (Поделиться), можно выбрать пункт **Generate QR Code** (Создать QR-код), как показано на рис. 8.45.

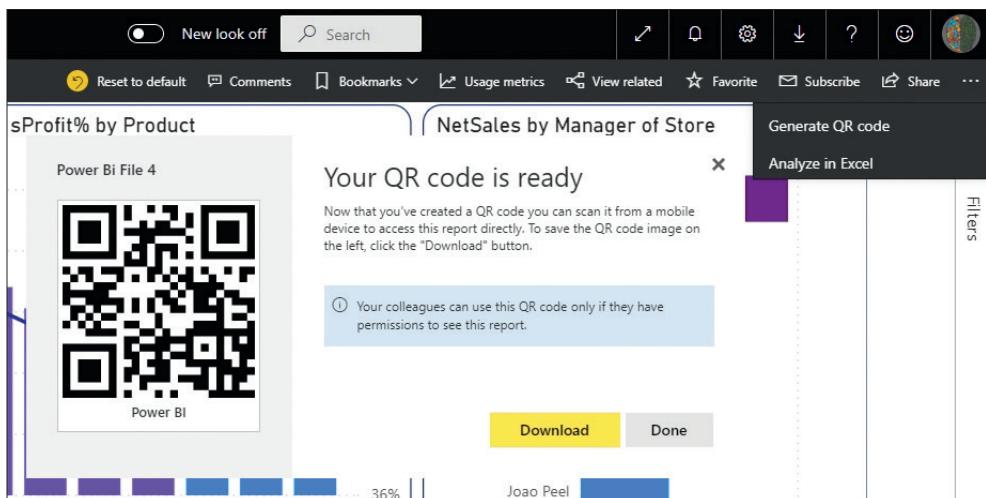


Рис. 8.45. Создание QR-кода

Вы можете отправить готовый QR-код коллеге лично или опубликовать его в социальных сетях для массового распространения.

Оба метода распространения отчетов и дашбордов являются базовыми и не займут много времени.

Существенным недостатком этих методов является то, что люди, которым вы открываете доступ к своим отчетам, не могут вносить изменения в данные. Также неудобно то, что вы можете давать доступ лишь к одному дашборду за раз, и если у вас этих дашбордов много, может возникнуть дискомфорт.

В меню **File** (Файл) можно найти дополнительные опции для распространения и встраивания своих дашбордов, показанные на рис. 8.46.

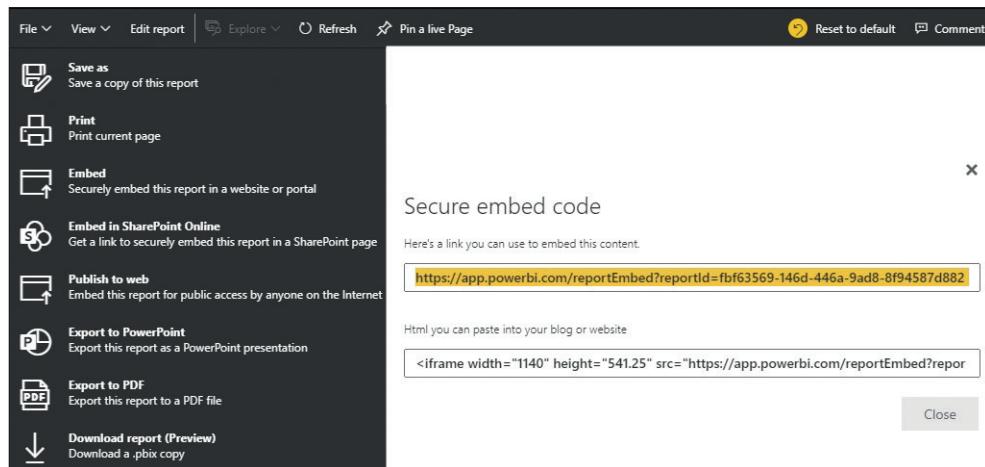


Рис. 8.46. Опции распространения информации в меню **Файл**

Посредством этого меню можно делиться как отчетами, так и дашбордами. Есть здесь пункты, не нуждающиеся в дополнительных пояснениях, такие как **Save As** (Сохранить как), **Print** (Печать), **Export to PDF** (Экспортировать в PDF) и **Download report** (Скачать отчет). Все они не поддерживают интерактивность и, по сути, позволяют сохранить копию отчетов как есть. Даже опция **Export to PowerPoint** (Экспортировать в PowerPoint) предлагает сделать лишь графический снимок дашборда, без сохранения интерактивной функциональности, как показано на рис. 8.47.

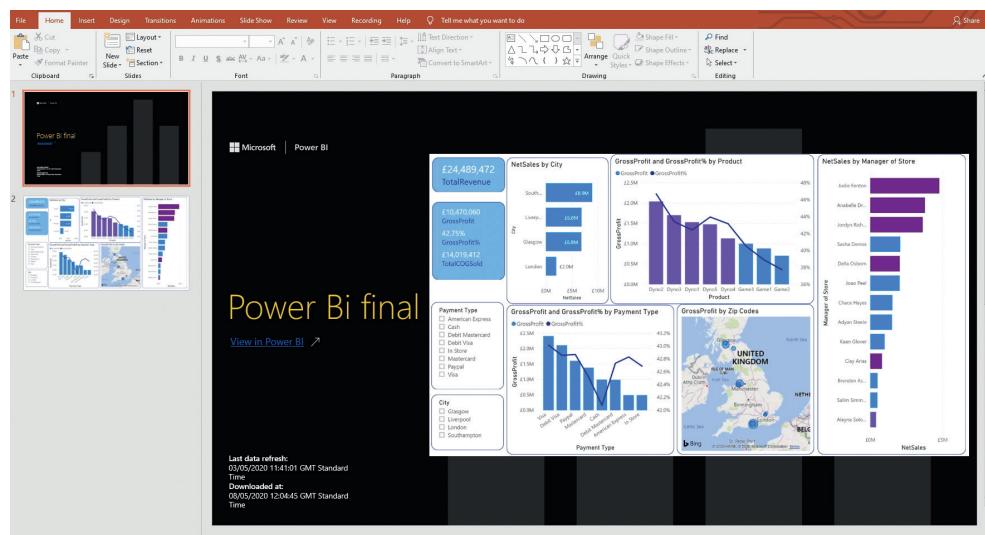


Рис. 8.47. Экспорт дашборда в PowerPoint

Другие опции в меню **Файл** предполагают гораздо большую степень интерактивности. Действия **Embed** (Встроить), **Embed in SharePoint Online** (Встроить в SharePoint Online) и **Publish to web** (Опубликовать в интернете) позволяют экспортierenовать полностью интерактивную версию дашборда.

Перечисленные опции немного отличаются друг от друга, но суть у них одна – они предоставляют вам код для встраивания в веб-страницу или готовый адрес в интернете.

- Нажав на кнопку **Publish to web** (Опубликовать в интернете), вы получите ссылку, которой сможете поделиться с коллегами по электронной почте (<https://app.powerbi.com/reportEmbed?reportId=fbf63569-146d-446a-9ad8-8f94587d8824&autoAuth=true&ctid=85931486-6860-4c35-8c73-06af6716f695&config=eyJjbHVzdGVyVXJsIjoiaHR0cHM6Ly93YWJpLXVrLXNvdXRoLWItcHJpbWFyeS1yZWRCmVjdC5hbmfSeXNpcy53aW5kb3dzLm5ldC8iifQ%3D%3D>), и код для встраивания в блог или веб-страницу (<iframe width=>1140>height=>541.25>src=><https://app.powerbi.com/reportEmbed?reportId=fbf63569-146d-446a-9ad8-8f94587d8824&autoAuth=true&ctid=85931486-6860-4c35-8c73-06af6716f695&config=eyJjbHVzdGVyVXJsIjoiaHR0cHM6Ly93YWJpLXVrLXNvdXRoLWItcHJpbWFyeS1yZWRCmVjdC5hbmfSeXNpcy53aW5kb3dzLm5ldC8iifQ%3D%3D>>frameborder=>0>allowFullScreen=>true></iframe>).
- Как видите, в коде для встраивания в блог по умолчанию прописана ширина элемента, равная 1140 пикселей, но при размещении его на своем сайте вам может понадобиться уменьшить это число. Однако при уменьшении размера дашборд может хуже восприниматься, так что учтите эти моменты заранее.

- Публикация дашбордов в интернете – легкий и быстрый способ распространения данных, но при этом стоит помнить, что в этом случае любой при наличии ссылки сможет увидеть опубликованную информацию, поскольку для этого не нужно авторизовываться и проходить проверку безопасности.
- Пункт **Embed in SharePoint Online** (Встроить в SharePoint Online) работает аналогично публикации в интернете в том смысле, что вы также получаете ссылку на страницу. Разница заключается в том, что коллегам вы должны давать доступ к файлу посредством SharePoint, как показано на рис. 8.48.

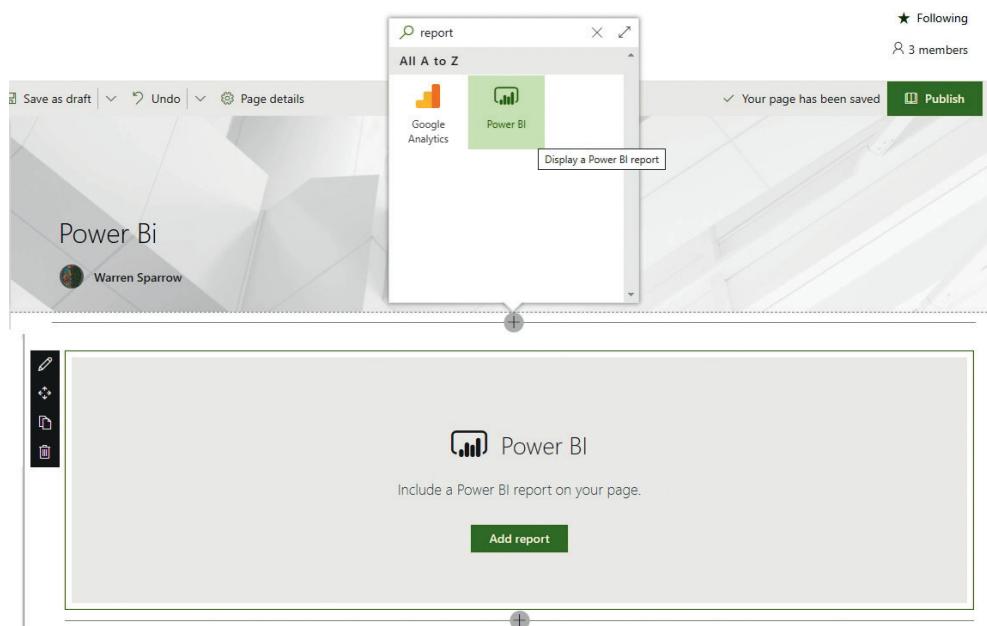


Рис. 8.48. Использование SharePoint Online для распространения дашбордов

В SharePoint Online можно создать страницу, после чего добавить *веб-часть* (web part) в первую колонку и выбрать опцию **Power BI**. Нажмите на кнопку добавления отчета и вставьте ссылку, приведенную выше, как показано на рис. 8.49.



Рис. 8.49. Привязанный отчет

После этого вы сможете настроить соотношение сторон: **16:9** или **4:3**. По завершении отчет появится автоматически, после чего вы можете его опубликовать. Взгляните на рис. 8.50.

The screenshot shows the SharePoint Online Site Pages library. A link settings dialog is open for a file named "Power Bi Dashboard.url". The dialog includes the following options:

- Who would you like this link to work for?** (Learn more)
 - Anyone with the link (radio button selected)
 - People in [group] with the link
 - People with existing access
 - Specific people
- Other settings**
 - Allow editing (checkbox checked)
- Buttons:** Apply, Cancel

Рис. 8.50. Опции распространения отчетов в SharePoint Online

Как видите, прямо в рамках SharePoint Online вы можете настраивать права доступа для конкретных людей и целой организации, как и в случае с любым другим файлом. Использовать этот метод довольно просто, поскольку вам не будет необходимости писать какой-то код: все, что вам будет нужно, – это ссылка.

Опция **Embed** (Встроить) позволяет распространять свои отчеты через веб-портал, при этом лишь авторизованные пользователи будут иметь доступ к данным.



Рис. 8.51. Встраивание отчета

Подытожим все вышесказанное следующим образом:

- базовые опции распространения отчетов позволяют легко и быстро делиться своими работами с другими людьми. Также этот вариант хорошо подходит для тестового запуска отчетов и дашбордов;
- опция **Publish to web** (Опубликовать в интернете) предлагает бесплатный вариант распространения отчетов и наборов данных и может пригодиться, если информация не несет конфиденциального характера;
- вариант **Embed in SharePoint Online** (Встроить в SharePoint Online) предоставляет возможность разграничения доступа к информации путем дополнительной авторизации. Это идеальный инструмент для распространения отчетов в рамках организации;
- опция **Embed** (Встроить) позволяет использовать локальное окружение SharePoint или веб-приложение с настройкой безопасности, но без необходимости для разработчика писать дополнительный код.

Итак, мы создали дашборд и поделились им с другими людьми и целыми отделами. В результате все, у кого есть доступ, могут использовать ваши наборы данных в полностью интерактивном режиме. При добавлении и публикации вами новых данных все адресаты получат доступ к обновленной информации.

После распространения отчета можно посмотреть, как часто пользователи используют ваши отчеты в своей работе.

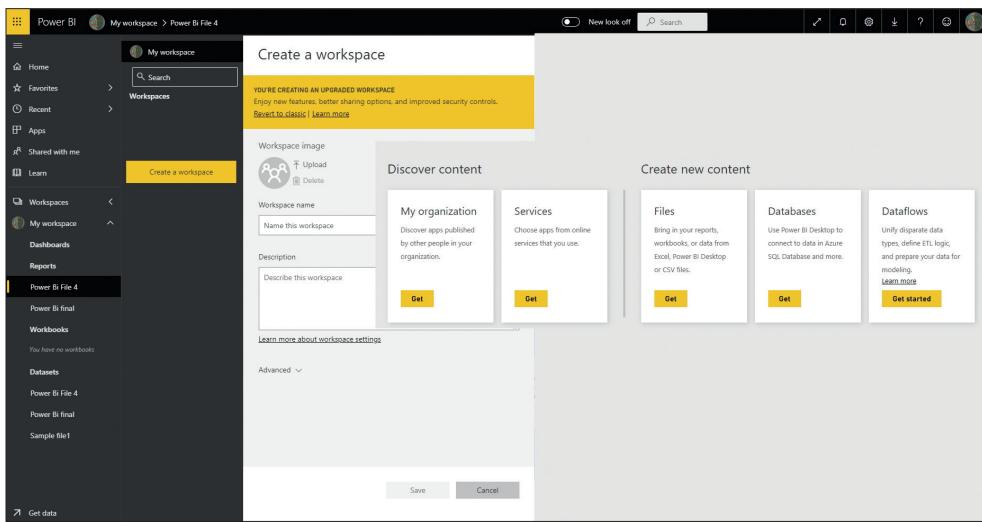


Рис. 8.52. Метрики использования

Чтобы посмотреть, кто и с какой целью использует ваши отчеты и дашборды, достаточно открыть раздел **Usage metrics** (метрики использования) в Power BI. Отчеты с этими **метриками** допустимо использовать только для чтения, но вы всегда можете скопировать отчет и затем изменить его. Применение этих отчетов позволяет понять, насколько интенсивно пользователи или отделы используют ваши дашборды.

Создавать дашборды умеют многие, но немногим удается разработать дашборд, выделяющийся среди прочих. В следующем разделе мы дадим пару ценных советов по поводу того, что стоит, а чего не стоит делать при проектировании дашбордов.

Практические рекомендации

Не существует единого мнения о том, что из себя представляет идеальный дашборд. Все зависит от огромного количества факторов.

Аудитория

Кто будет пользоваться вашим дашбордом? Будет ли это обычный человек, просматривающий государственную статистику по своей стране, или это будет опытный аналитик данных, отслеживающий финансовые тенденции? От ответа на этот вопрос, в частности, зависит, сколько данных включать в дашборд и до какой степени должна быть детализирована информация. Также очень важно учитывать, на каких устройствах пользователи будут открывать вашу визуализацию. Будет ли это ноутбук, планшет или большой экран? От этого напрямую зависит, сколько информации вы можете разместить на дашборде.

Хаос и беспорядок

Одной из самых распространенных ошибок разработчиков является попытка уместить на дашборде чересчур много информации. Это очень плохо. Во-

первых, чем больше визуальных элементов размещается на дашборде, тем ниже будет его производительность. Во-вторых, пользователь просто не способен быстро проанализировать такое большое количество данных. Попробуйте оставить на экране лишь нужные элементы при компоновке своего дашборда.

Размер и цвет

Большинство людей читают дашборды слева направо и сверху вниз. Поэтому стремитесь размещать наиболее важные элементы в верхней левой части полотна. С этой целью можно использовать элемент визуализации **Card** (Карточка) – он поможет вам вывести наиболее важные показатели, выделенные также шрифтом, цветом и размером. Если все цифры на вашем дашборде будут одной величины, пользователю будет очень трудно расставить правильные акценты. Выделяйте размером наиболее важные показатели. Что касается цветового наполнения дашборда, существует немало цветовых палитр для использования в визуальных отчетах, но при этом вы также можете менять цвет тех или иных элементов на диаграммах для привлечения к ним особого внимания.

Существует масса советов по проектированию полезных и приятных на вид дашбордов, но для начала хватит перечисленных здесь, а дальше только практика, практика и еще раз практика. Здесь, как и в любом другом деле, чем больше практикуешься, тем лучше и легче все получается.

ЗАКЛЮЧЕНИЕ

Данную главу мы начали с выполнения конкретных шагов. Сперва извлекли нужные нам данные, затем отредактировали их и импортировали в модель данных, чтобы можно было создать необходимые связи между таблицами. После этого создали несколько вычисляемых столбцов и мер, а далее перешли к разработке нашего дашборда. Опубликовав дашборд, мы рассмотрели различные варианты его распространения среди потенциальных пользователей. И хотя звучит это все замысловато, при следовании определенным рекомендациям весь процесс не займет много времени.

В этой главе мы загружали исходные данные из папки, но с таким же успехом могли бы брать их из сети или из базы данных SQL Server. При этом мы отфильтровали данные таким образом, чтобы импортировались только файлы нужного нам формата, не забыв перевести расширения файлов в нижний регистр. Затем мы объединили таблицы связью, чтобы можно было определять менеджеров магазинов по индексу, после чего создали вычисляемые столбцы и меры с использованием математических операций. На заключительном шаге мы вынесли все нужные нам данные на один дашборд и опубликовали его.

Мне лично очень импонируют дашборды своей наглядностью. При помощи дашборда можно огромное количество информации отобразить графически с использованием небольшого пространства. При этом исходные данные могут собираться из самых разных источников, включая веб-страницы. Одного щелчка достаточно, чтобы все элементы визуализации обновились и показывали актуальную информацию.

В следующей главе мы приступим к изучению языка программирования M.

Часть III

Изучаем язык M

В этой части мы познакомимся с языком программирования, использующимся в Power Query. Для начала узнаем основы синтаксиса языка M, после чего продолжим его изучение на примерах, таких как сведение и отмена свертывания таблицы, а также познакомимся с ключевыми словами и научимся писать пользовательские функции. В заключительной главе книги мы сравним языки M и DAX и посмотрим, как в M генерируются результаты запроса.

Данная часть состоит из следующих глав:

- глава 9 «Работаем с языком M»;
- глава 10 «Примеры использования языка M»;
- глава 11 «Создание базовых пользовательских функций»;
- глава 12 «Различия между языками DAX и M».

Глава 9

Работаем с языком M

В этой главе мы познакомимся с языком программирования M и его синтаксисом, структурой и функциями.

У каждого языка есть свои четко обозначенные синтаксис и структура, и как только вы усвоите базовое устройство языка M, дальше все будет намного проще. Здесь мы познакомимся с основными типами данных и функциями языка, а также приведем примеры использования каждого типа. В конце главы мы решим задачу загрузки файла CSV посредством языка M.

Основные темы этой главы:

- знакомство с языком M;
- освоение синтаксиса языка M и способов написания сценариев;
- использование ключевого слова `#shared` для просмотра всех функций из библиотеки, включая их определения, без подключения к интернету или любому другому источнику данных.

К концу главы вы должны научиться анализировать код на языке M, выявлять в нем синтаксические и структурные ошибки и исправлять их.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Вам понадобится интернет-соединение для загрузки соответствующих файлов с GitHub. Файлы с кодом для этой главы можно скачать по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

При написании главы мы предполагали, что вы знаете, как открывать редактор запросов Power Query, и умеете обращаться с его основными командами. Также мы исходили из предположения о том, что у вас есть опыт использования как минимум одного языка программирования, – это поможет вам быстрее понять синтаксис и логику языка M.

Видеофрагмент, посвященный данной главе, располагается по адресу: https://www.youtube.com/watch?v=pQ6lyh7mxZE&list=PLeLcvrwLe1860_GIEZs47WaZXwZjwTN83&index=10&t=0s.

Знакомство с языком M

Как мы уже знаем, Power Query позволяет подключаться к различным источникам данных, импортировать информацию, очищать, фильтровать и преобразовывать ее, после чего публиковать. Кроме того, с помощью этого инструмента легко можно объединять и сравнивать данные из различных источников. Язык программирования, использующийся в Power Query, получил имя *M*, поскольку его рабочее название было **Mashup** (*смешивание, англ.*).

Power Query обладает полноценным интерфейсом, что позволяет использовать множество функций без необходимости уметь программировать. Все без исключения действия, которые производите в среде Power Query, в дальнейшем транслируются в инструкции на языке *M*. И хотя ранее в этой книге мы написали пару строчек кода на *M*, по большей части мы обходились выполнением нужных нам действий посредством интерфейса Power Query. В предыдущей главе мы редактировали код *M* и писали свой для создания дополнительных столбцов. Без языка *M* бывает трудно создавать сложные запросы и проводить полноценные преобразования и вычисления. Кроме того, есть вещи – такие как подключение к веб-службам, – реализовать которые без написания кода на языке *M* просто невозможно.

Первое, что стоит отметить, – это то, что язык программирования *M* является функциональным и, как и все остальные языки, обладает четко выраженным синтаксисом и структурой. Прежде чем перейти к описанию синтаксиса, хотелось бы напомнить о том, что язык *M* является чувствительным к регистру символов как в отношении имен переменных, так и в плане функций. Этим *M* отличается от Excel, где можно писать код в любом регистре, – анализатор сам приведет его в правильный вид. Например, вы можете писать функции VBA в нижнем регистре, первая буква автоматически будет преобразована в прописную. Что касается языка DAX, то он позволяет использовать как верхний, так и нижний регистр при написании кода. Вы даже можете смешивать стилистику при программировании, хотя это и не рекомендуется.

Конечно, язык *M* можно использовать не только в Power BI Desktop. Он может применяться в Power Query и Excel, а также в более поздних версиях Excel при использовании инструмента **Get & Transform** (Получить и преобразовать).

Теперь мы можем смело приступить к синтаксису языка *M* и начнем с отличий между ключевыми словами `let` и `in`, являющимися основой конструкции языка.

Основы синтаксиса языка M

В данном разделе мы познакомимся с основными составляющими синтаксиса языка *M*, после чего перейдем непосредственно к примерам. Как мы уже упоминали ранее, язык *M* состоит из отдельных элементов, включающих функции, переменные, выражения и значения, все из которых участвуют в процессе преобразования исходных данных.

Посмотрим, как выглядит фрагмент кода на языке *M* при первом открытии редактора.

Откройте редактор запросов Power Query из Excel, после чего нажмите на кнопку **Advanced Editor** (Расширенный редактор) на вкладке **View** (Просмотр), как показано на рис. 9.1.

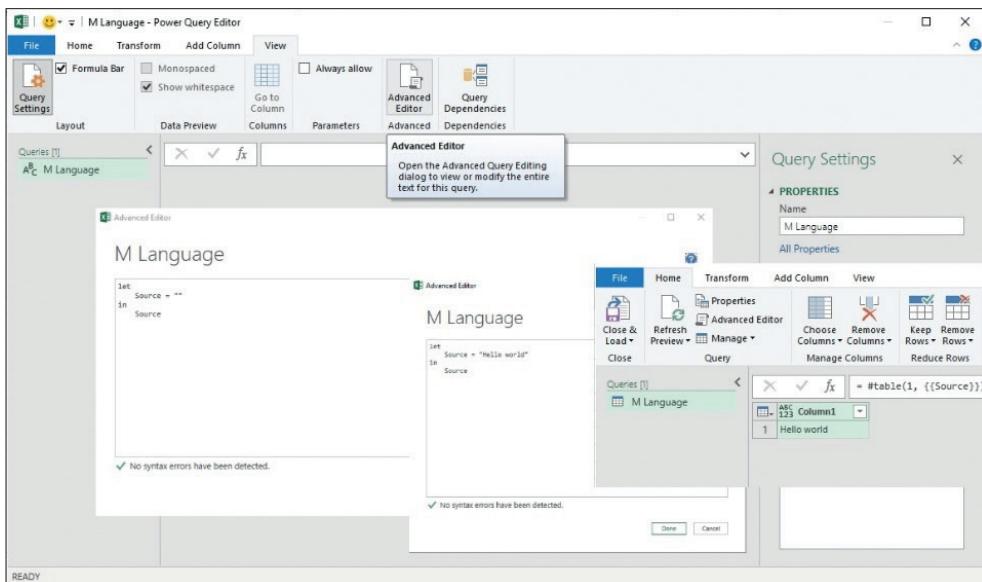


Рис. 9.1. Расширенный редактор Power Query

По умолчанию открывшийся фрагмент кода будет иметь следующий вид:

```
let
    Source = ''
in
    Source
```

Здесь мы видим два явно выраженных блока: блок выражений *let* и блок выражений *in*. Давайте отдельно поговорим о каждом из них.

Блок выражений let содержит программный код – или, как его еще можно называть, набор процедурных шагов, – используемый для определения запроса. На каждом из этих шагов мы присваиваем переменной определенное выражение, заключающее в себе логику преобразования исходных данных. Таких шагов может быть несколько, и обычно каждый последующий шаг выводится из предыдущего, хотя это не всегда так. Процедурные шаги могут не соответствовать физическому или логическому порядку следования, а могут быть перемешаны. В этом случае, однако, вас или любого, кто попытается разобраться в логике кода, будет ждать настоящий кошмар. Поэтому шаги всегда желательно упорядочивать согласно логике. По сути, каждый запрос в языке M представляет собой одно выражение *let*. Это код, определяющий наборы данных, а каждый процедурный шаг соответствует одной переменной.

Блок выражений in представляет собой вывод. Я знаю, что этоозвучит неправильно, но фактически здесь возвращается значение переменной, основываясь на блоке *let*.

Как видно по рис. 9.1, основа для блоков *let* и *in* уже готова в базовом представлении кода, остается только наполнить их. Если вставить текст Hello World

в блок `let`, то в преобразованном в таблицу виде мы получим соответствующий текст в столбце **Column 1**. И хотя это не слишком впечатляет, зато показывает, как работает синтаксис языка в целом.

Важно

При заполнении блока выражений `let` необходимо помнить о некоторых очень важных вещах. Например, каждая строка в этом блоке должна заканчиваться запятой `(,)`, за исключением последней строки, стоящей непосредственно перед началом блока `in`. Имена переменных пишутся без пробелов в одно слово с использованием прописных букв или символа подчеркивания `(_)` в начале каждого слова. Если же вам необходимо в название переменной вставить пробелы, придется использовать символ `#` с последующим написанием имени переменной в кавычках, например так: `#'This is the long variable name'`. Имена переменных могут также включать специальные символы, такие как `%` или `&`. Кроме того, имя переменной будет показываться в панели примененных шагов справа в качестве названия шага. Для размещения в коде комментариев можно использовать сочетание символов `//` в начале строки (так же, как в JavaScript). Круглые скобки `(())` используются для передачи параметров функциям, квадратные `(([]))` служат для представления набора записей, а фигурные `(({}))` применяются для списков.

Большинство функций в языке M являются частью класса `object`, а общий синтаксис их вызова будет таким:

```
ObjectClass.Function(),
ObjectClass.Function(Parameter),
ObjectClass.Function(Parameter1, parameter2)
```

При этом есть несколько функций, которые изначально в Power Query не принадлежат базовому классу `object`, и среди них, например, функция `date`. Чтобы использовать ее, следует писать просто `#date()`, а в качестве параметров передавать год, месяц и день. Подробнее мы об этом поговорим в главе 10.

Базовый синтаксис языка M довольно прост и заключается в том, чтобы четко соблюдать структуру блоков выражений `let` и `in`. Чтобы легче было ориентироваться в коде на языке M, можно посоветовать обращать особое внимание на используемые скобки, дабы понимать, с чем мы имеем дело – с переменными, списками или инкапсуляцией набора записей.

В следующем разделе мы рассмотрим несколько примеров кода на языке M. Некоторым фрагментам при этом не понадобится источник данных, поскольку значения будут генерироваться сами по себе. Используя одну и ту же технику, мы сможем в дальнейшем применять одинаковые функции к разным источникам данных.

КЛЮЧЕВОЕ СЛОВО #SHARED ДЛЯ ПОЛУЧЕНИЯ ФУНКЦИЙ ИЗ БИБЛИОТЕКИ

В этом разделе мы познакомимся с *ключевым словом #shared*, которое загружает функции и константы *перечисляемого типа* (enumerator) в результирующий набор данных. Для этого нам не понадобится никакой источник данных, поскольку код, который мы создадим, будет автоматически использовать библиотеки `#shared`. Мы поговорим о работе с различными типами данных, а также импортируем файл CSV. Для каждого типа данных мы напишем небольшой пример, чтобы вы могли усвоить работу с ними и применять в своих проектах. Кроме того, мы будем ассоциировать данные с конкретными типами, чтобы анализатор языка точно знал, имеет он дело с текстовым типом, числовым или каким-то еще. Мы коротко пройдемся по всем типам данных и посмотрим на их сходства и различия.

Текстовый тип данных

В начале данной главы мы уже создали один крохотный пример с использованием текстового типа данных в виде строки «Hello world». Вы увидите, что нет никакой необходимости явно указывать тип данных, поскольку он может быть выведен автоматически. Снова обращаясь к рис. 9.1, можно заметить, что слева от заголовка столбца с выводом текста «Hello world» располагаются буквы **ABC**, что говорит о том, что мы имеем дело с *текстовыми данными* (*Text*). И хотя текстовый тип данных в нашем распоряжении один, использовать его можно совершенно по-разному.

Одна из самых популярных функций из этого раздела – `Text.format()`. Она позволяет вставлять значения внутрь текстовой строки. Взгляните на следующий пример:

```
Text.Format('#[StudentName] has chosen #[Subject]',  
          [StudentName ='Alice', Subject ='Maths'])
```

Результирующей строкой этого выражения будет следующая: `Alice has chosen Maths`. Эту же функцию можно использовать и для передачи в качестве параметра не записи, а целого списка, как показано ниже:

```
Text.Format('The first number is #{0}, the second number is  
#{1}, the third number is #{2}',  
          {15,9,29})
```

В результате получим такой текст:

```
The first number is 15, the second number is 9, the third  
number is 29.
```

Следующим мы рассмотрим числовой тип данных, который, как ясно из названия, предназначен для хранения чисел и числовых форматов.

Числовой тип данных

Анализатор кода языка M автоматически присваивает *числовой тип данных* (*Number*) вычисляемым переменным, когда это возможно. Посмотрим, как это работает.

Создайте пустой запрос и вставьте в него следующий код:

```
let
    variable1 = 5,
    variable2 = 10,
    variable3 = variable1 + variable2
in
    variable3
```

В результате вычисления этого нехитрого выражения из трех переменных мы получим ответ **15**. Как видите, первым двум переменным были присвоены числовые значения, тогда как третья ссылается на сумму первых двух. На панели **APPLIED STEPS** (Примененные шаги) в правой части экрана наши переменные преобразились в полноценные шаги запроса. Взгляните на рис. 9.2.

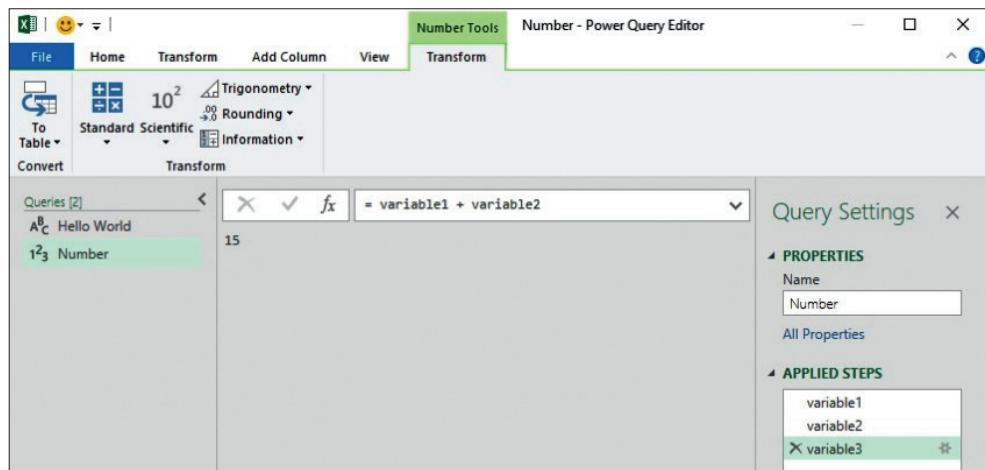


Рис. 9.2. Шаги внутри запроса

Если бы мы хотели сохранить пробел внутри имени переменной, нам пришлось бы записать его так: #'variable 3'. Посмотрите на приведенный выше фрагмент кода. Он напоминает обычный процедурный язык программирования, в котором выполнение идет последовательно сверху вниз. Движок Power Query начинает работу с поиска всей необходимой информации для блока выражений *let*. В данном случае мы ищем значение шага 3. Анализатор кода посмотрит, какие данные нужны для выполнения третьего шага, в результате

чего выполнит первые два. Если результат вычисления первой строки кода для получения итогового значения не понадобится, Power Query не будет его выполнять вовсе, несмотря на его первую позицию в списке.

Также из полезных возможностей можно отметить то, что мы можем ссылаться на вывод других запросов. Если у нас есть запрос с названием Number и нам необходимо обратиться к его результату, нам даже не придется писать разделы выражений let и in, достаточно будет все стереть и написать просто Number. Результат выполнения одноименного запроса будет выведен тут же.

Числовой тип данных используется очень часто, ведь в большинстве случаев нам необходимо подсчитывать итоги, средние значения и выполнять другие математические вычисления. Числовой тип данных включает в себя целые значения, числа с десятичными знаками и валюту.

В программировании мы постоянно используем списки, храня в них как числа, так и любые другие объекты. В следующем разделе мы посмотрим, как в языке M можно работать со списками, включая их вложенные версии.

Списки

Списки (List) представляют собой отсортированный набор значений. Чтобы создать список в языке M, откройте редактор запросов Power Query и введите следующую формулу:

```
let
    fruit={'Apple','Grapes','Pear'}
in
    fruit
```

Слева от названия запроса на левой панели мы увидим соответствующую списку иконку, кроме того, в меню автоматически появится вкладка **List Tools** (Средства для списков). Посмотрите на рис. 9.3.

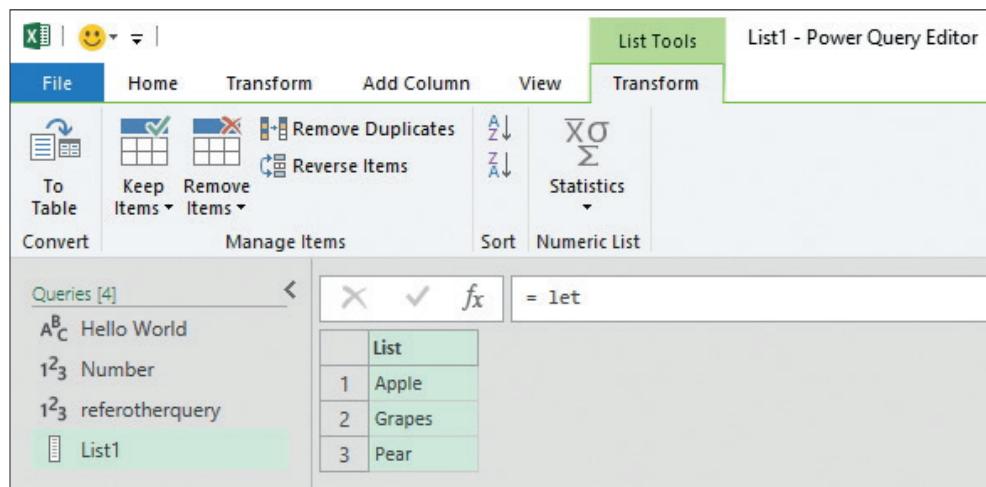


Рис. 9.3. Средства для списков в меню

При помощи этого дополнительного меню вы можете преобразовывать списки в таблицы, удалять ненужные элементы и выполнять другие полезные операции. На выходе мы получим упорядоченный список: Apple будет идти первым, а Pear – последним.

Также мы можем создавать вложенные списки с использованием конструкций вида `{[],{}}`. Рассмотрите следующий код:

```
let  
    fruitNested ={{'Apple','Grapes'},{'Limes','Lemons'}}  
in  
    fruitNested
```

В результате вы увидите в выводе два списка, нажав на один из которых, сможете раскрыть его.

С числовыми списками можно производить множество полезных операций. Например, если объявить переменную как `numbersToTen = {1..10}`, мы получим список из десяти чисел: от 1 до 10.

Теперь перейдем к еще одному типу данных в Power Query, а именно к записям, которые представляют собой списки полей.

Записи

Запись (*Record*) представляет собой разделенный запятыми список, в котором значения ассоциируются с разными полями. Объявлять записи можно при помощи квадратных скобок `([])`. Посмотрим, как это работает.

Создайте пустой запрос и введите следующую формулу в расширенном редакторе:

```
let  
    record = [firstName='Warren',surname='Sparrow',title='Mr'],  
in  
    record
```

Как видите, мы создали разделенный запятыми список полей. У нас есть три поля – `firstName`, `surname` и `title`, – а имена полей и их значения записаны с двух сторон от знака равенства.

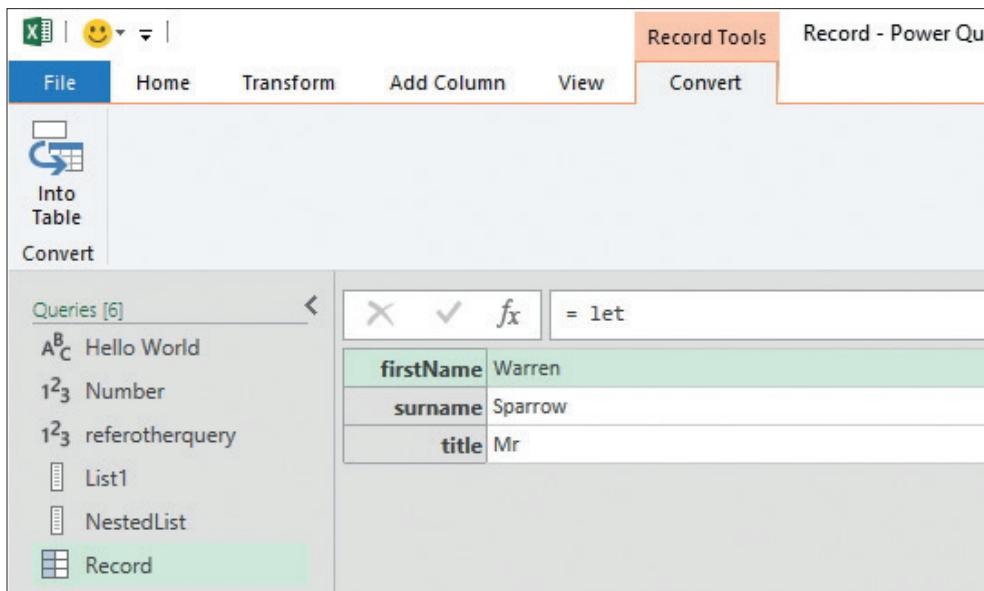


Рис. 9.4. Средства для записей в меню

После создания записи вы увидите соответствующую иконку справа от имени запроса на левой панели, а также дополнительную вкладку **Record Tools** (Средства для записей) в меню, как показано на рис. 9.4.

Теперь пришло время рассмотреть еще один тип данных в языке M, и этот тип представляет собой таблицу.

Табличный тип данных

Табличный тип данных (Table) является, пожалуй, наиболее важным из структурных типов в языке M. Можно мысленно представить его себе как комбинацию списков и записей.

Создайте пустой запрос и введите следующую формулу в расширенном редакторе:

```
let
    Source = #table({‘A’, ‘B’, ‘C’,
    ‘D’}, {{‘1’, ‘2’, ‘3’, ‘4’}, {‘10’, ‘11’, ‘12’, ‘13’}})
in
    Source
```

Функция `#table` служит для создания таблиц. При этом таблица состоит из двух частей – сначала в нее передается список наименований столбцов в текстовом формате. В нашем примере столбцы будут называться A, B, C и D. Следующие переданные списки будут формировать строки таблицы, как показано на рис. 9.5.

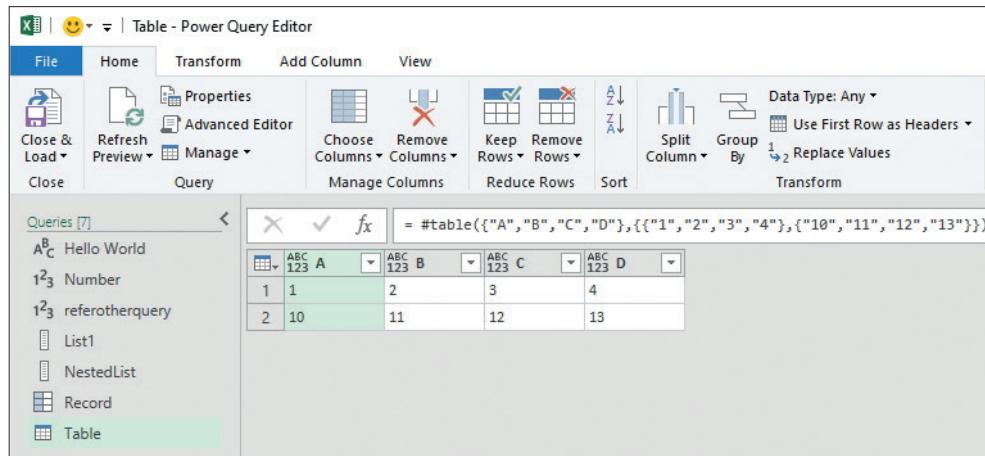


Рис. 9.5. Табличный тип данных

В списке запросов слева мы увидим новый запрос с возвращаемым типом **Table** (Таблица), чтобы мы понимали, что в нем представлена информация в табличном виде. Заметьте, что слева от имен столбцов стоит иконка **ABC123**, а это означает, что данные в этом столбце могут быть любого типа. Для устранения неоднозначности толкования можно заранее избавиться от неопределенности, жестко задав тип данных для колонок.

Попробуйте такой вариант инициализации переменной в блоке выражений `let`:

```
Source = #table(type table[A=number, B=number, C=number, D=number],{{1,2,3,4},{100,101,102,103}})
```

Здесь мы снова создали таблицу, но на этот раз четко указали тип для столбцов, так что вместо иконки **ABC123** увидим ожидаемый значок **123**, что нам и требовалось.

Зачастую после получения данных нам нужно осуществлять поиск по ним, и следующий раздел будет посвящен как раз этой теме.

Поиск данных

В предыдущем разделе мы усвоили, что таблица представляет собой комбинацию из списков и записей. Это очень важно, поскольку мы можем выполнять поиск по спискам, записям и таблицам как по позиции, так и по вхождению. Посмотрим.

Снова создайте пустой запрос и введите следующую формулу в расширенном редакторе:

```
let
    fruit={'Apple','Grapes','Pear'},
    numbersToTen = {1..10}
in
    numbersToTen[2]
```

Здесь у нас есть переменная с именем `numbersToTen`, представляющая собой список от 1 до 10. В блоке выражений `in` мы запросили значение из этого списка с индексом 2. Запуск этого кода вернет нам значение 3, поскольку индексация в языке M начинается не с 1, а с 0. Таким образом, нулевым элементом будет 1, первым – 2, а вторым – 3.

Если изменить последнюю строку на `fruit[0]`, мы получим на выходе значение `Apple`. Запрос `fruit[6]` вернет ошибку за неимением такого количества элементов в списке. При этом если написать `fruit[6]?`, ответом будет `null`, поскольку вопросительный знак будет указывать на требование вернуть пустое значение при ошибочном обращении.

Использование фигурных скобок `({})` позволяет осуществлять поиск в списке по индексу, и тот же самый подход мы можем применить к записям.

Можно применить одну и ту же функцию для поиска в списках и записях. Создайте пустой запрос и введите такую формулу в расширенном редакторе:

```
let
    Source = #table({{'A','B','C',
    'D'},{{'1','2','3','4'},{'10','11','12','13'}}}),
    Output = Source[A]
in
    Output
```

Здесь у нас есть таблица с заголовками столбцов A, B, C и D. Этот код похож на предыдущий, за исключением строки `Output = Source[A]`. При помощи квадратных скобок `([])` мы получаем конкретное поле из записи, но в источнике у нас таблица, так что итогом будет список всех значений из колонки A.

Если изменить `[A]` на `{A}`, поиск выдаст первую строку из таблицы, а именно 1, 2, 3, 4.

Допустимо комбинировать эти методы. Например, при вводе формулы `= Source[A]{A}` мы получим на выходе значение из первого столбца и первой строки.

Язык M включает в себя великое множество функций, и здесь мы коснулись лишь нескольких из них. Если в пустом запросе ввести формулу `= #shared`, на экран будет выведен полный список доступных функций в языке M. Информация будет показана в виде записи, где в каждом поле будет указана конкретная функция. Щелчок мыши по любой функции приведет к открытию документации по выбранной функции.

До этого момента мы создавали данные непосредственно в коде. В следующем разделе попробуем импортировать информацию из файла CSV при помощи языка M.

ИМПОРТ ФАЙЛА CSV ПОСРЕДСТВОМ ЯЗЫКА M

Каждому из нас довольно часто приходится сталкиваться с необходимостью анализа файлов CSV и Excel. Можно сделать это при помощи Power Query, но если мы хотим загрузить данные посредством языка M, можно прибегнуть к двум способам. Первый заключается в использовании расширенного редак-

тора Power Query, а второй предполагает написание кода на языке M непосредственно в строке формул.

Откройте расширенный редактор и введите следующую формулу:

```
let
    salesData = Csv.Document(File.Contents('C:\DataFiles\sales.csv'),
    [Delimiter=',', Encoding=1252])
in
    salesData
```

Если вы желаете все сделать без открытия расширенного редактора, то можете ввести в редакторе формул следующий код: = Csv.Document(File.Contents('C:\DataFiles\sales.csv'), [Delimiter=',', Encoding=1252]).

В результате будет открыт файл CSV, в котором есть две проблемы. Первая заключается в наличии пустого столбца, а вторая – в том, что данные из первой строки необходимо воспринимать в качестве заголовков.

Чтобы избавиться от лишней колонки, откройте расширенный редактор и добавьте такую строку к блоку выражений let, как показано на рис. 9.6:

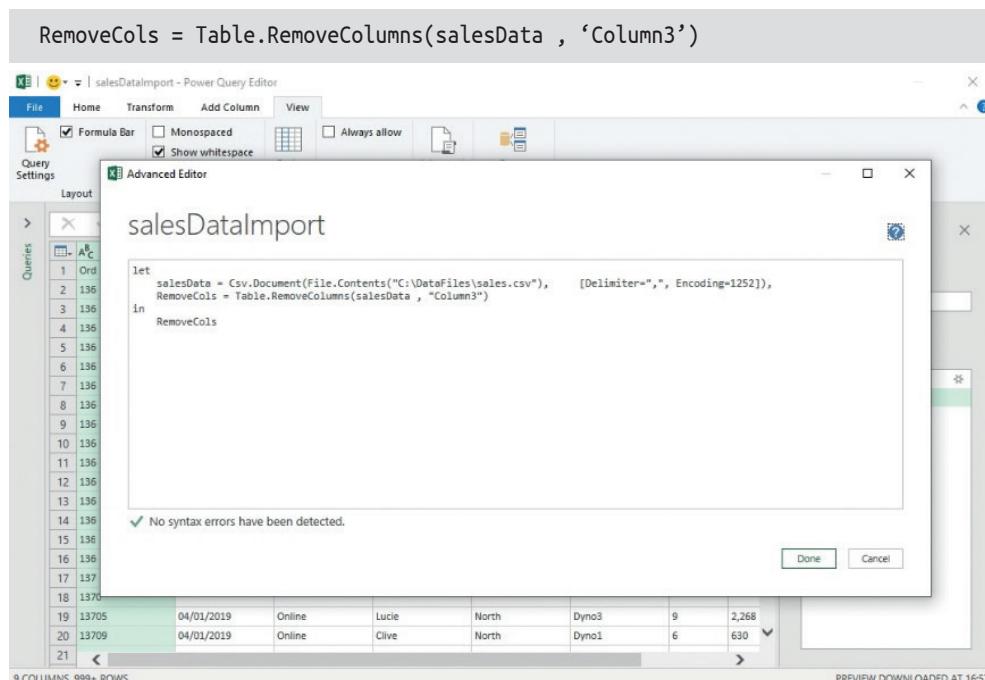


Рис. 9.6. Удаление колонки

После этого перейдем к решению задачи по указанию правильных заголовков таблицы. Для этого подойдет следующая команда:

```
PromoteNames = Table.PromoteHeaders(RemoveCols, [PromoteAllScalars=true])
```

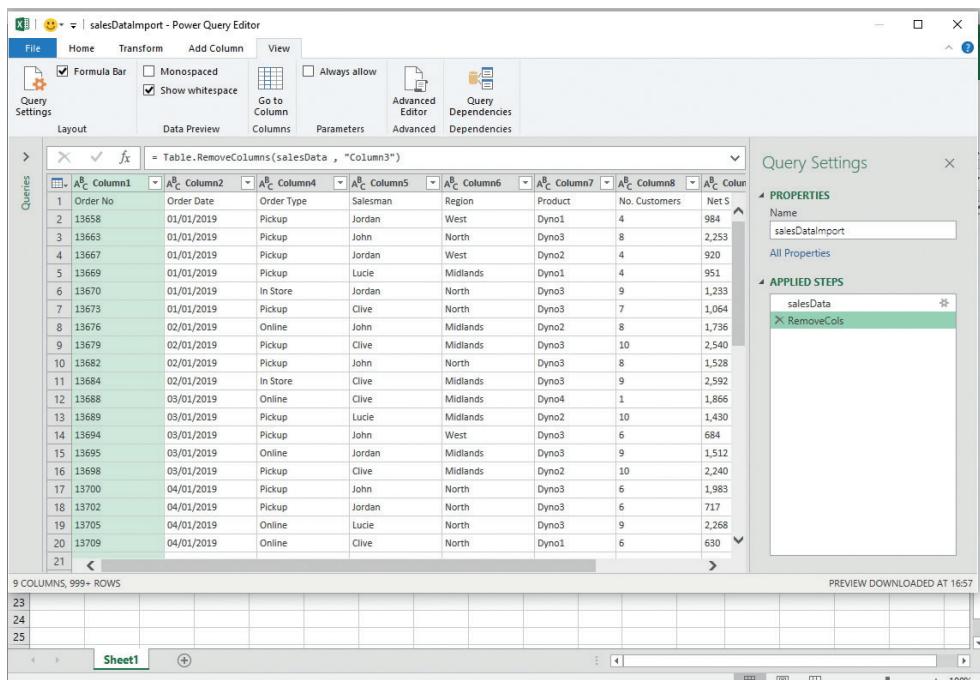


Рис. 9.7. Восстановление заголовков таблицы

В данной главе мы освоили основы синтаксиса и структуры языка M, а также рассмотрели примеры использования типов данных и некоторых распространенных функций языка.

ЗАКЛЮЧЕНИЕ

Читая эту главу, вы сделали свои первые шаги в мире языка M. Вы обрели достаточно знаний, чтобы понимать, как устроена внутренняя структура языка и как работает его синтаксис. На данном этапе этого хватит, чтобы анализировать несложные сценарии в Power Query и при необходимости исправлять их.

Мы провели беглое сравнение языков M, DAX и VBA в отношении их требовательности к регистру, а затем приступили к написанию простых формул на языке M. Мы больше пользовались расширенным редактором Power Query, но те же формулы вы можете писать и в строке формул. При этом предпочтительно все же использовать расширенный редактор, поскольку найти ошибку в длинных многострочных выражениях, введенных в строку формул, будет очень затруднительно.

В этой главе мы рассмотрели множество примеров по работе со всеми известными нам типами данных в языке M, включая текст, числа, списки, записи и таблицы, а также осуществили импорт файла CSV при помощи M. Теперь вам не должно составлять труда определить по скобкам в коде, о каком именно типе данных идет речь.

В следующей главе мы посмотрим, как можно посредством языка M решать куда более сложные сценарии.

Глава 10

Примеры использования языка M

В данной главе мы рассмотрим практические примеры применения языка M, включая выполнение операции конкатенации. Сначала мы проведем сравнение формул в Excel и Power BI, после чего перейдем к оператору **амперсанд** (&) и посмотрим, как можно использовать его на практике.

Мы также узнаем, как функции `Text.From` и `Text.Combine` могут быть использованы для объединения и сцепки различных строк, дат и столбцов. Кроме того, мы установим и настроим полнофункциональную бесплатную версию SQL Server для некоммерческого использования. Попутно узнаем, как загрузить базу данных AdventureWorks и использовать ее в наших примерах.

В заключение мы вновь коснемся темы **параметров** и посмотрим, как можно их эффективно использовать для выполнения фильтрации источника данных по дате. Далее мы узнаем о применении параметров для сортировки объектов и столбцов в возрастающем и убывающем порядке и посмотрим, как производить эти изменения на вкладке **Data** (Данные) в Power BI. Мы пройдем по всем стадиям работы с параметрами, включая их объявление, создание и переименование.

Основные темы этой главы:

- объединение данных с использованием конкатенации;
- преобразование типов данных;
- установка и настройка SQL Server;
- использование параметров.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Вам понадобится интернет-соединение для загрузки соответствующих файлов с GitHub. Файлы с кодом для этой главы можно скачать по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

В этой главе мы исходим из предположения о том, что вы уже понимаете синтаксис и структуру языка M, а также знаете, как управляться с различными типами данных, включая текст, числа, списки, записи и таблицы. Кроме того, вы должны уметь выполнять поиск и пользоваться библиотекой `#shared`. Если вы не уверены в том, что усвоили эти темы, перечитайте главу 9.

Вам также понадобится для работы SQL Server. Если он у вас не установлен, мы покажем, где скачать бесплатную версию и установить совершенно легально. После установки SQL Server вам нужно будет скачать базу данных AdventureWorks по адресу: <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15>.

Видеофрагмент, посвященный данной главе, располагается по адресу: https://www.youtube.com/watch?v=mzRQLQCmvg&list=PLLeLcvrwLe1860_GJEZs47WaZXwZjwTN83&index=11&t=1s.

Объединение данных с использованием конкатенации

Я всегда удивляюсь, как по-разному можно хранить одни и те же данные в файлах CSV в зависимости от источника. В студенческих программах и пакетах управления проектами информация также зачастую хранится в разных форматах. Типичным вариантом отображения информации о полном имени человека является следующий: SMITH, John. В Excel для разделения имени и фамилии по столбцам можно воспользоваться кнопкой **Text to Columns** (Текст по столбцам) на панели **Data** (Данные), указав в качестве разделителя запятую, как показано на рис. 10.1.

	A	B	C	D
1				
2	SMITH	John	Together John SMITH	Proper John Smith
3				
4			=CONCATENATE(B2, " ", A2)	=PROPER(\$C2)
5				
6				
7				

Рис. 10.1. Разделенные столбцы А и В

В дальнейшем вы можете вновь объединить значения двух колонок нужным вам образом, попутно воспользовавшись функцией **Proper** (ПРОПНАЧ) для корректного отображения первых букв в словах. Конечно, это займет какое-то время, и автоматизировать этот первый шаг вам не удастся. А если вам необходимо выполнять похожие операции в разных документах, вас ждет настоящий кошмар.

Сначала мы посмотрим, какие действия для приведения данных в порядок необходимо выполнить в Excel, и сравним эту последовательность с аналогичными действиями в Power BI. Вы увидите, насколько проще это делается в Power BI. К тому же дополнительным преимуществом использования Power BI является то, что созданные шаги вы можете в будущем применить к другим источникам данных – Power BI Desktop сделает это за вас автоматически.

Итак, что вам нужно сделать:

1. Запустите Power BI Desktop и выберите пункт **Excel** из выпадающей кнопки **Get Data** (Получить данные).
2. Выберите файл **names.xls** и нажмите на кнопку **Open** (Открыть).
3. Выберите лист **SurnameNames** и затем нажмите на кнопку **Transform Data** (Преобразовать данные).

В зависимости от имеющейся у вас версии Power BI Desktop вкладка **Transform** (Преобразовать) может находиться в разных местах рабочего экрана. В последних версиях Power BI Desktop (июнь 2020) кнопка **Split Column** (Разделить столбец) находится на вкладке **Transform** (Преобразовать), как показано на рис. 10.2. Если у вас более ранняя версия Power BI, вам необходимо будет нажать на кнопку **Data source settings** (Настройки источника данных) в группе **Transform data** (Преобразовать данные) на вкладке **Home** (Главная).

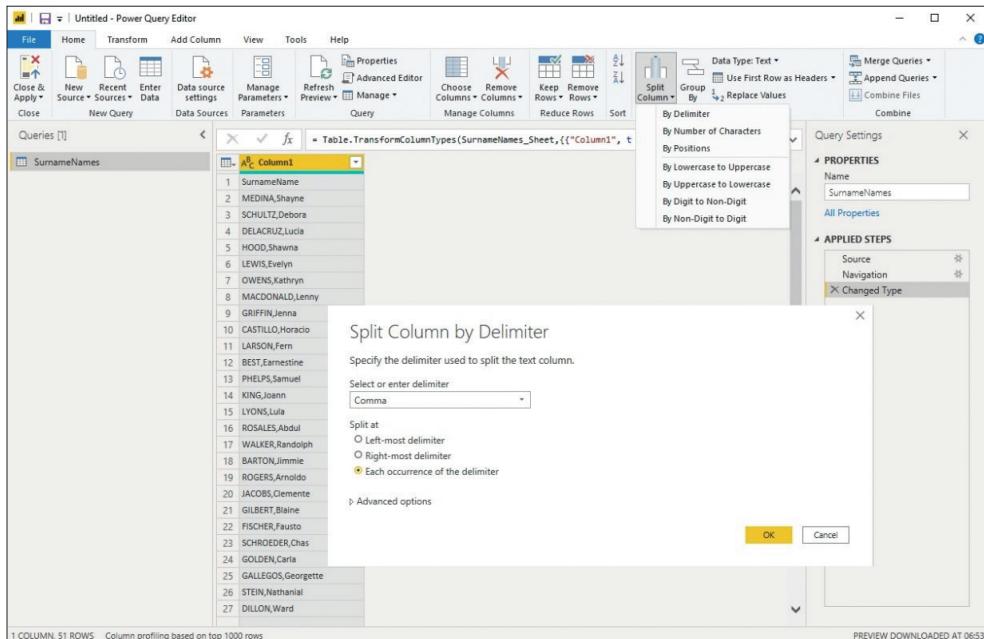


Рис. 10.2. Пункт меню Разделить столбец

В нашем случае разделителем является запятая, но если у вас иная ситуация, вы всегда можете выбрать собственные настройки. Также в диалоговом окне разделения столбца вы можете открыть область расширенных параметров и выбрать тип разделения данных на столбцы или строки, но в данный момент мы все оставим по умолчанию.

Итак, мы разделили наш столбец на два, но при этом фамилии написаны в верхнем регистре. Мы бы хотели, чтобы прописной была только начальная буква. Есть два подхода к решению этой задачи:

- первый состоит в написании следующего кода на языке M: `Table.TransformColumns(#"Changed Type1",{{"Column1.1", Text. Rgoreg, type text}})`. Как видите, здесь мы попросили перевести содержимое столбца `Column1.1` в режим `Rgoreg`, то есть с начальной прописной буквой. Примечательно, что здесь мы указываем то же ключевое слово, что и в Excel, а именно `Rgoreg`;
- второй способ – интерфейсный, и для его реализации нужно нажать правой кнопкой мыши на столбце **Column 1.1** и в подменю **Transform** (Преобразование) выбрать пункт **Capitalize Each Word** (Каждое слово с прописной), как показано на рис. 10.3.

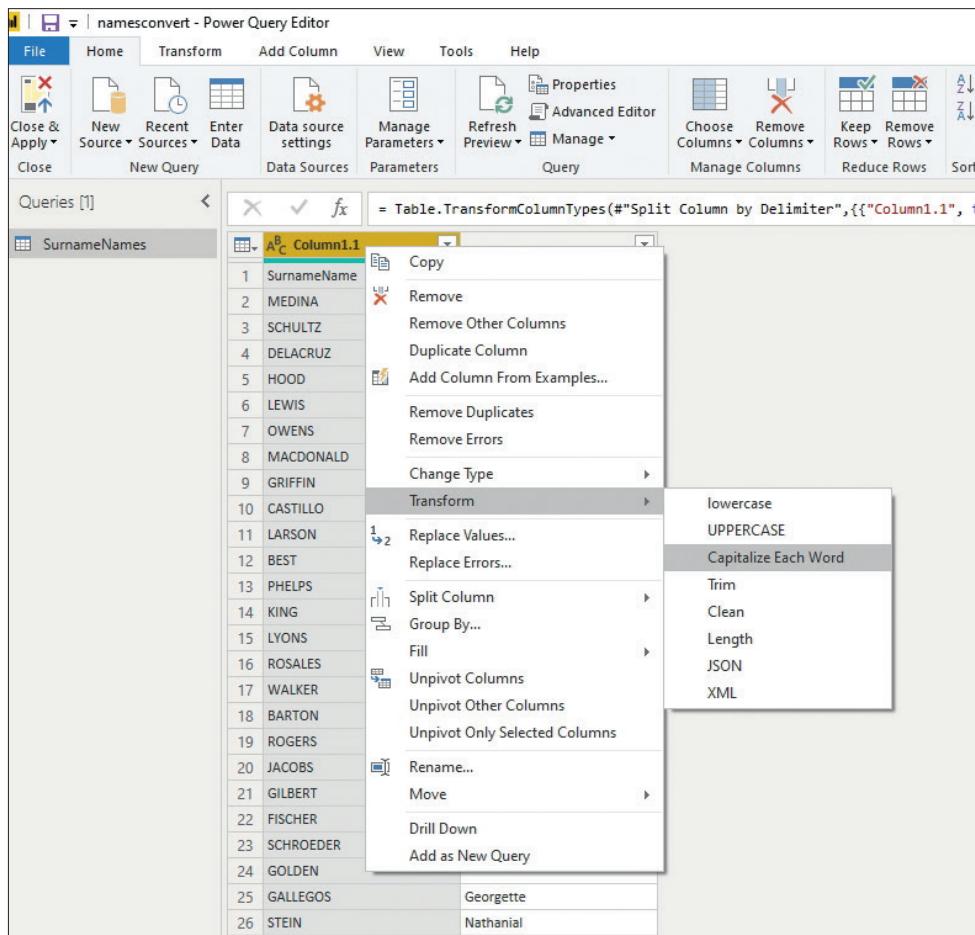


Рис. 10.3. Каждое слово с прописной

На заключительном шаге нам нужно собрать воедино разрозненные столбцы. С этой целью мы создадим настраиваемый столбец с формулой `[Name]&" "&[Surname]`. Здесь мы просто извлекаем значения из двух столбцов – `Name` и `Surname` – и объединяя их. Символ пробела « « используется для того, чтобы не превратить имя и фамилию в одно слово. Также необходимо помнить, что имя должно следовать перед фамилией, так что склейке столбцов нужно сначала указать столбец `Column1.2`, а затем `Column1.1`.

Нажмите на кнопку **Custom column** (Настраиваемый столбец) на вкладке **Add Column** (Добавить столбец) и введите следующую формулу: `[Column1.2]&" "&[Column1.1]`. Нужные столбцы вы можете выбирать из предложенного списка справа, как показано на рис. 10.4.

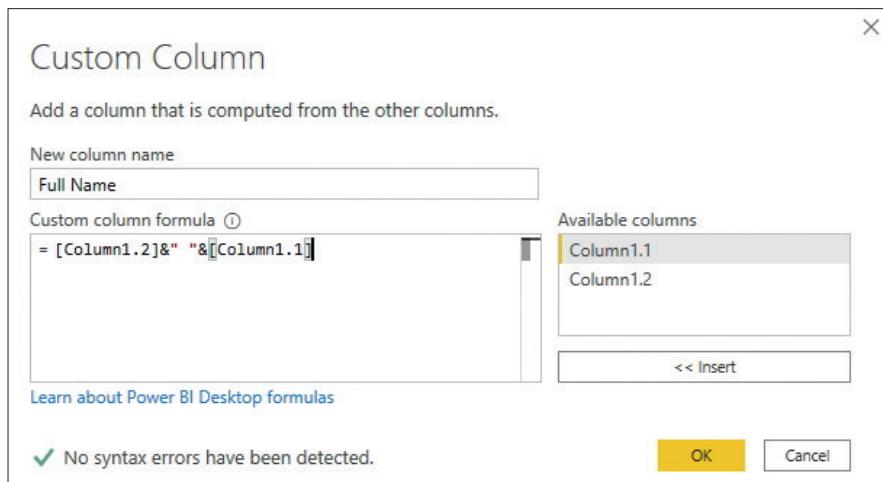


Рис. 10.4. Доступные столбцы

Альтернативой этому методу может служить объединение двух столбцов. Нажмите на кнопку **Merge Columns** (Объединить столбцы) на вкладке **Add Column** (Добавить столбец). Выберите в выпадающем списке **Separator** (Разделитель) пункт **Custom** (Пользовательский) и введите в качестве разделителя знак ««»». Нажмите на кнопку **OK**.

The screenshot shows the Power Query Editor interface with the 'Merge Columns' dialog box open. The 'Separator' dropdown is set to '-Custom-' and the 'New column name (optional)' field is set to 'Merged'. The main area displays a table with two columns: 'SurnameName' and 'Name'. The 'SurnameName' column contains names like MEDINA, SCHULTZ, DELACRUZ, etc., and the 'Name' column contains names like Shayne, Debora, Lucia, etc.

Рис. 10.5. Метод объединения столбцов

На рис. 10.6 видно, что мы оставили имена и при этом удалили первую строку, чтобы данные выглядели адекватно. На самом деле мне стоило оставить только третий столбец, удалив первые два.

Surname	Name	Full Name
Medina	Shayne	Shayne Medina
Schultz	Debora	Debora Schultz
Delacruz	Lucia	Lucia Delacruz
Hood	Shawna	Shawna Hood
Lewis	Evelyn	Evelyn Lewis
Owens	Kathryn	Kathryn Owens
Macdonald	Lenny	Lenny Macdonald
Griffin	Jenna	Jenna Griffin
Castillo	Horacio	Horacio Castillo
Larson	Fern	Fern Larson
Best	Ernestine	Ernestine Best
Phelps	Samuel	Samuel Phelps
King	Joann	Joann King
Lyons	Lula	Lula Lyons
Rosales	Abdul	Abdul Rosales
Walter	Randolph	Randolph Walter
Barton	Jimmie	Jimmie Barton
Rogers	Arnoldo	Arnoldo Rogers
Jacobs	Clemente	Clemente Jacobs
Gilbert	Blaine	Blaine Gilbert

Рис. 10.6. Завершенная конкатенация столбцов

Все преобразование заняло у меня ровно четыре минуты, включая написание формул вручную. При этом разработка подобного сценария способна сэкономить человеку уйму времени на обработку различных файлов CSV с именами и фамилиями, результаты которой он может использовать в других документах. Прелесть этого метода заключается еще и в том, что нам не важно, в каком регистре написаны фамилии, поскольку мы сами преобразуем их в нормальный вид. Можно также добавить подобное преобразование и к именам, чтобы не беспокоиться о том, как они выглядели в исходном файле.

Бывает, что перед вами стоит прямо противоположная задача – объединить имя и фамилию человека из разных столбцов, чтобы можно было использовать результат, например, в качестве учетной записи для входа в систему.

Продолжим использовать тот же исходный файл, но на этот раз обратимся к листу **Names**. Вы можете выполнить все те же действия, но на этапе создания настраиваемого столбца нужно будет ввести следующую формулу без использования пробелов: `= [Column2] & [Column1] & [Column3]`.

На рис. 10.7 видно, что вместе с именем и фамилией в новый столбец попал год, когда сотрудник начал работать в компании, что вполне уместно для названия учетной записи.

Если использовать эту формулу в Power BI, вы получите ошибку по причине попытки применить конкатенацию к текстовым и числовым данным. В следующем разделе мы расскажем, как обойти эту ошибку. Логично предположить, что для выполнения этой задачи лучше всего преобразовать числовые значения в текстовые перед выполнением объединения.

В данном разделе мы познакомились с методами объединения и разделения полей по определенным условиям для использования результатов в других приложениях.

The screenshot shows the Power Query Editor interface with two tables selected: 'SurnameNames' and 'Names'. The 'Applied Steps' pane on the right shows a step named 'Lowercased Text'.

	ABC 123 Name	ABC 123 Surname	B Year	A usernames
1	Shayne	MEDINA	2020	medinashayne2020
2	Debora	SCHULTZ	2020	schultzdebona2020
3	Lucia	DELACRUZ	2020	delacruzluicia2020
4	Shawna	HOOD	2020	hoodshawna2020
5	Evelyn	LEWIS	2020	lewisevelyn2020
6	Kathryn	OWENS	2020	owenskathryn2020
7	Lenny	MACDONALD	2020	macdonaldlenny2020
8	Jenna	GRIFFIN	2020	griffinjenna2020
9	Horacio	CASTILLO	2020	castillohoracio2020
10	Fern	LARSON	2020	larsonfern2020
11	Earnestine	BEST	2019	bestearnestine2019
12	Samuel	PHELPS	2019	phelpssamuel2019
13	Joann	KING	2019	kingjoann2019
14	Lula	LYONS	2019	lyonslula2019
15	Abdul	ROSALES	2019	rosalesabdul2019
16	Randolph	WALKER	2019	walkerrandolph2019
17	Jimmie	BARTON	2019	bartonjimmie2019
18	Arnoldo	ROGERS	2019	rogersarnoldo2019
19	Clemente	JACOBS	2019	jacobsclemente2019
20	Blaine	GILBERT	2019	gilbertblaine2019
21	Fausto	FISCHER	2019	fischerfausto2019
22	Chas	SCHROEDER	2019	schoederchas2019
23	Carla	GOLDEN	2019	goldencarla2019
24	Georgette	GALLEGO	2019	gallegoogeorgette2019
25	Nathaniel	STEIN	2018	steinnathaniel2018
26	Ward	DILLON	2018	dillonward2018
27	Eloise	PEARSON	2018	pearsoneloise2018
28	Rowena	JACOBSON	2018	jacobsonrowena2018

Рис. 10.7. Создание имен учетных записей

В следующем разделе мы поговорим о важной теме преобразования типов данных.

ПРЕОБРАЗОВАНИЕ ТИПОВ ДАННЫХ

В предыдущем разделе мы попытались выполнить конкатенацию тестового значения с числовым, что обернулось ошибкой. Та же участь ждет нас и при попытке объединить поля с участием даты и некоторых других типов данных. В данном разделе мы рассмотрим несколько способов конкатенации значений разных типов данных.

Для объединения полей несовместимых типов данных можно использовать функцию *Text.From*. Эта функция принимает в качестве параметра число, дату, время или даже двоичные данные и возвращает их текстовое представление. Например, на выходе формулы *Text.From(7)* будет «7», но не число 7, а его текстовое представление. Если же функции *Text.From* передать значение *null*, она также вернет *null*. Функцию *Text.From* можно использовать для преобразования в текст не только чисел, но и даты со временем.

Можно разными способами использовать оператор амперсанд (&) для выполнения операции конкатенации данных. В Excel простейшим вариантом выполнить сцепку значений является использование оператора &. Например, если вам нужно объединить вместе поля **Title**, **Name**, **Street**, **City**, **Zip Code**

и **Country**, показанные на рис. 10.8, можно использовать следующую формулу: `=C2&" "&A2&CHAR(10)&D2&" "&E2&CHAR(10)&F2&" "&G2`.

	A	B	C	D	E	F	G	H
1	Name	Gender	Title	Street	City	Zip Code	Country	
2	Faith C Schofield	female	Mrs	19 St Denys Road	POYSTON CROSS	SA62 3NX	USA	Mrs Faith C Schofield 19 St Denys Road POYSTON CROSS SA62 3NX USA
3	Liam E Pollard	male	Mr	123 Bootham Crescent	RIPE	BN8 4FX	USA	Mr Liam E Pollard 123 Bootham Crescent RIPE BN8 4FX USA
4	Benjamin M Bentley	male	Mr	38 Bootham Crescent	RICKLING GREEN	CB11 5HF	USA	Mr Benjamin M Bentley 38 Bootham Crescent RICKLING GREEN CB11 5HF USA
5	Bradley J Hayes	male	Mr	27 Consett Rd	HILGAY	PE38 5JJ	USA	Mr Bradley J Hayes 27 Consett Rd HILGAY PE38 5JJ USA
								Mr Ewan M Nash

Рис. 10.8. Конкатенация с использованием оператора &

Мы специально разместили колонки в неправильном порядке, чтобы показать, что обращаться к ним в формуле можно произвольно. Еще я использовал выражение `CHAR(10)`, представляющее собой перенос строки, чтобы выходные данные выглядели более структурированными.

Мы также можем сделать это в Power Query и Power BI, поскольку здесь используется операция преобразования данных.

1. Откройте файл `Address.xlsx` и перейдите в редактор запросов Power Query, выбрав пункт **Table/Range** (Из таблицы/диапазона) на вкладке **Data** (Данные). Мы будем использовать функцию `Text.Combine` для преобразования всех нетекстовых полей в текстовые значения. Сюда включаются поля с *обобщенным типом данных* (generic data type).
2. Для добавления шага можно либо нажать правой кнопкой мыши на примененном шаге и выбрать пункт **Insert Step After** (Вставить шаг после), либо щелкнуть на иконке `fx` слева от строки формул.

Вставьте следующий код в строку формул:

```
= Table.AddColumn(#"Changed Type", "Address Labels", each Text.Combine(Record.ToList(_), "#(lf)"))
```

Эта формула создает новый столбец с именем `Address Labels`, в который собирает данные из всех полей слева направо. Выражение `#(lf)` означает переход строки в Power Query и работает по примеру `CHAR(10)` в Excel.

Объединяя строки в Excel, мы можем использовать функцию `CONCATENATE` (СЦЕПИТЬ) или оператор `&`. Эти методы очень похожи, при этом у функции `CONCATENATE` есть единственное ограничение в виде максимального количества объединяемых строк, равного 255. В реальности же вам очень редко может понадобиться склеивать более 255 элементов, так что можете выбирать способ по вкусу. Мне лично импонирует использование оператора `&`.

Далее мы рассмотрим примеры использования базовых операторов, но для этого у вас должен быть установлен SQL Server. Если он у вас уже установлен и настроен, можете пропустить следующий раздел и сразу приступать к загрузке базы данных. Если же нет, читайте дальше и узнаете, как скачать и установить бесплатную версию этой СУБД на своем компьютере.

УСТАНОВКА И НАСТРОЙКА SQL SERVER

В данном разделе мы расскажем о том, как загрузить и установить *SQL Server*. Понятно, что с *SQL Server* на постоянной основе работают далеко не все читатели, но установить *SQL Server* версии *Developer* можно абсолютно бесплатно. Главным отличием этой версии от *Enterprise* является невозможность использования ее в коммерческих целях. В остальном же эти версии имеют сопоставимый функционал. Если вы хотите ознакомиться с полным списком отличий между разными версиями *SQL Server*, обратитесь по адресу: <https://www.microsoft.com/en-au/sql-server/sql-server-2017-editions>. Также для загрузки файлов вам понадобится подписка Microsoft или MSDN, которые являются бесплатными. Лично я использую версию *SQL Server 2017*, но вы можете загрузить по желанию любую версию *Developer SQL Server: 2016, 2017 или 2019*. Между ними есть определенные различия, но в целом они похожи. Заметим, что в зависимости от устанавливаемой версии вам может потребоваться от 2,9 до 4 Гб свободного места на диске.

Существуют разные способы установить версию *Developer SQL Server*. Если у вас достаточно старый компьютер, вы можете остановиться на версии *SQL Server 2016 Developer*, не поддерживающей продвинутые запросы на языке *Transact-SQL (T-SQL)*. Но мы бы не рекомендовали этого делать.

Есть два основных способа загрузки и установки *SQL Server* версии *Developer*. Первый из них заключается в загрузке программы по ссылке <https://my.visualstudio.com/Downloads?q=SQL%20Server>, как показано на рис. 10.9.

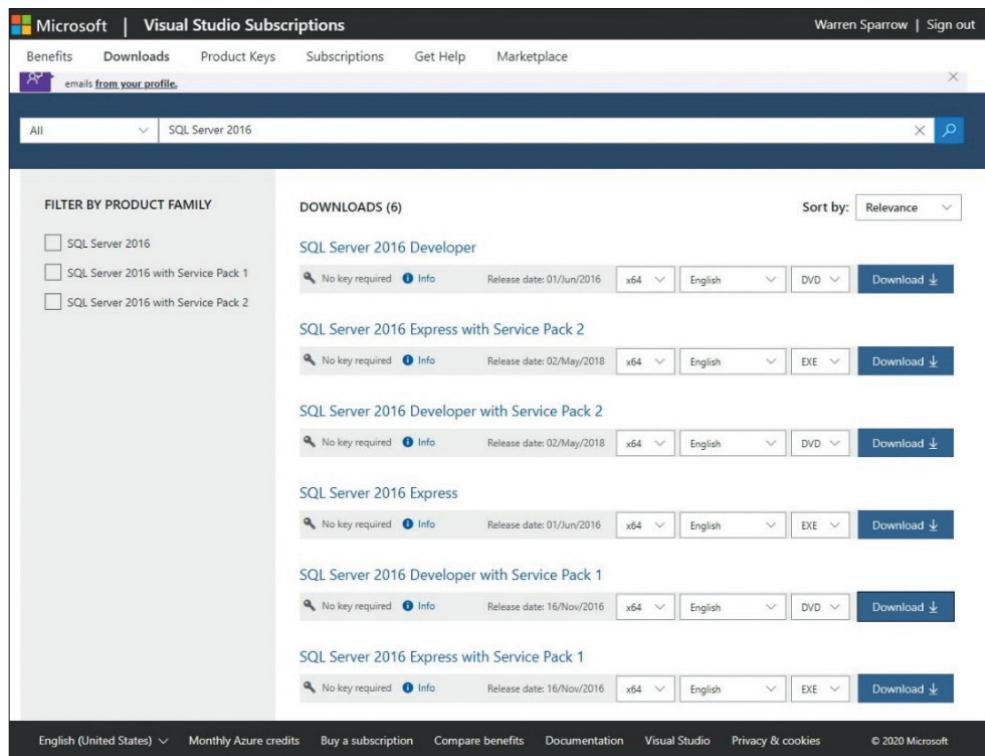


Рис. 10.9. Опции загрузки SQL Server

Здесь вы можете выбрать, какую версию скачать. Установка любой из них выполняется похожим образом.

После загрузки файла вам необходимо произвести ряд действий для установки программы. Некоторые шаги могут быть пропущены, другие являются обязательными. Пройдемся по алгоритму.

1. Первый шаг установки включает в себя планирование дальнейших шагов, как показано на рис. 10.10. Здесь можно только читать. Вы можете пропустить этот шаг и сразу переходить к установке.

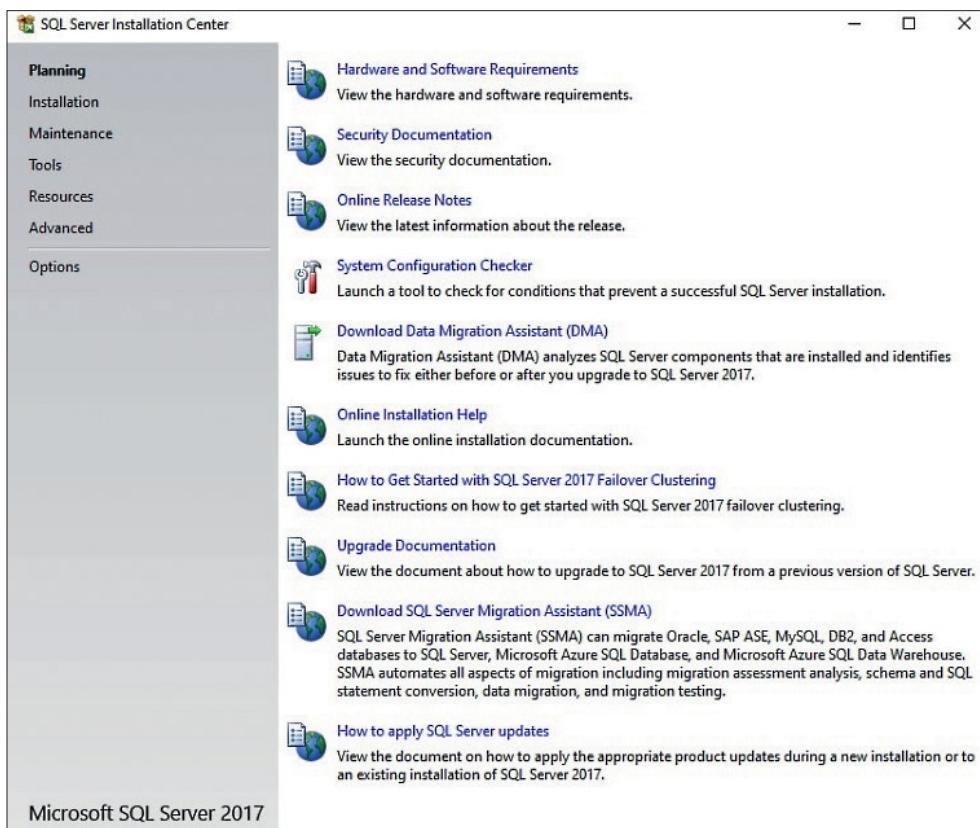


Рис. 10.10. Шаг планирования установки

2. На шаге установки вам необходимо выбрать, что именно вы хотите установить. Выберите верхний пункт «Новая установка изолированного экземпляра SQL Server или добавление компонентов к существующей установке» и дождитесь появления диалогового окна.

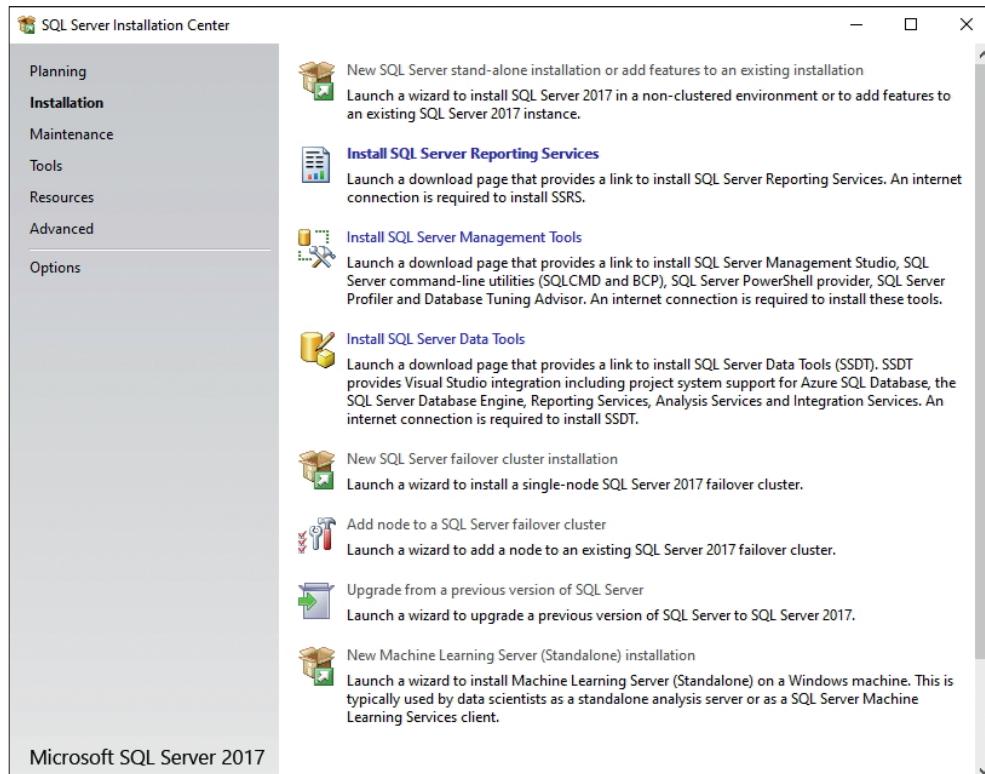


Рис. 10.11. Шаг установки

Иногда диалоговое окно открывается за основным, так что вы его можете не увидеть. Перемещайте время от времени основное окно по экрану, чтобы не пропустить появления второго окна.

3. В окне ввода ключа продукта необходимо установить переключатель в положение **Specify a free edition** (Укажите бесплатный выпуск) и в выпадающем списке выбрать пункт **Developer**.
4. Нажимайте на кнопку продолжения, попутно согласившись с условиями лицензии.
5. После этого появится диалоговое окно с правилами установки, в котором будет выполнена проверка библиотек, ключей реестра и т. п. Не беспокойтесь, если на этом этапе вы получите предупреждение от своего брандмауэра.
6. Далее вам необходимо будет выбрать компоненты для установки, как показано на рис. 10.12.

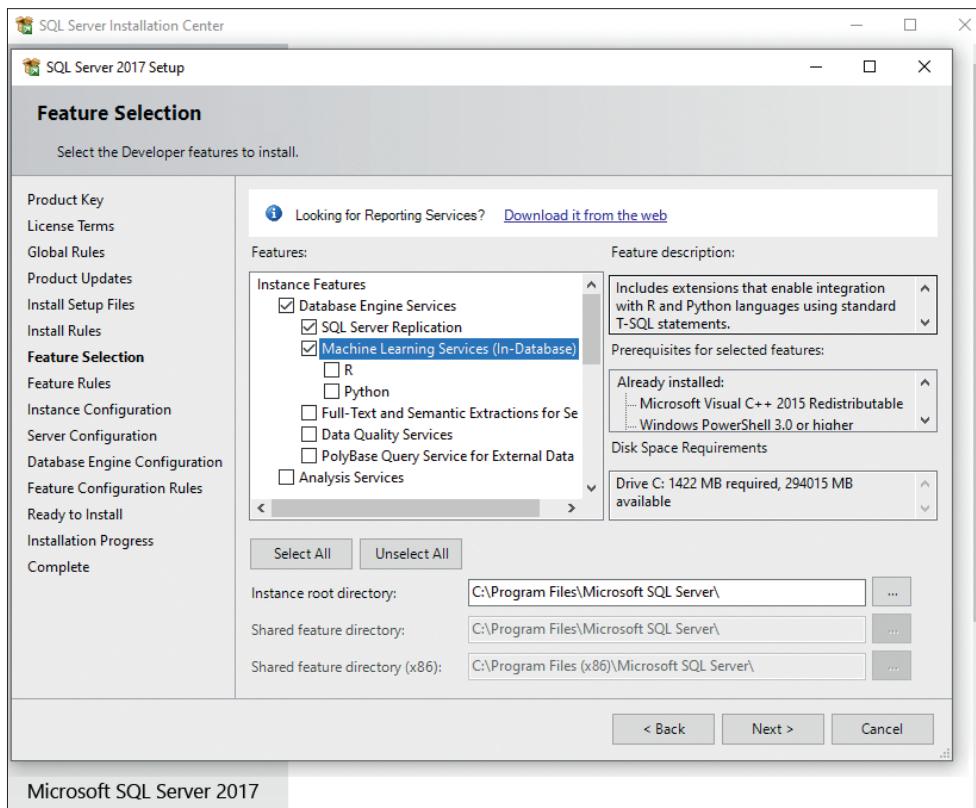


Рис. 10.12. Выбор компонент для установки

Выберите пункт **Database Engine Services** (Службы ядра СУБД). Остальные компоненты для выполнения заданий из этой главы не обязательны, но вы можете установить их, если собираетесь пользоваться SQL Server и после чтения главы.

7. После этого откроется окно настройки экземпляра SQL Server. На этом шаге вы создаете и даете имя своему экземпляру SQL Server. Диалоговое окно будет выглядеть примерно так, как на рис. 10.13, за исключением отличий в установленных экземплярах.

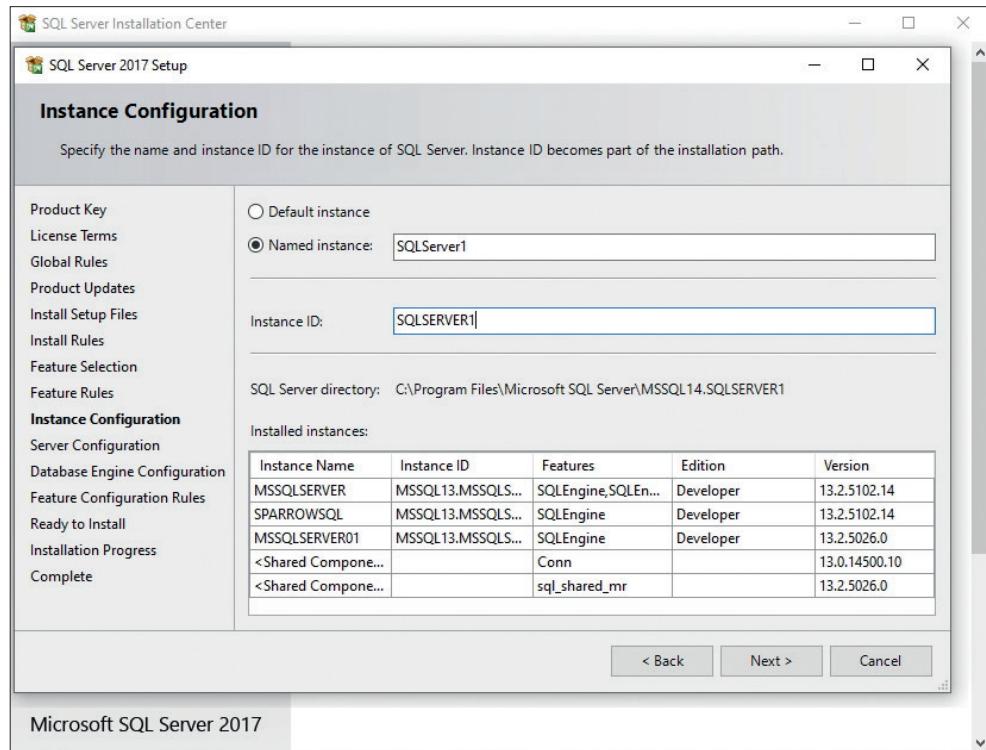


Рис. 10.13. Настройка экземпляров

8. Если у вас нет ни одного установленного экземпляра, лучше всего использовать экземпляр по умолчанию с именем **MSSQLSERVER**. И хотя обычно достаточно иметь в наличии один экземпляр СУБД, в наших примерах с параметрами мы будем использовать два экземпляра, так что вы сразу можете установить и второй.
9. На шаге **Database Engine Configuration** (Настройка ядра СУБД) вы должны выбрать метод аутентификации. Лично я предпочитаю использовать **Mixed Mode** (Смешанный режим). Нажмите на кнопку **Далее**, чтобы завершить установку.

Второй способ установки SQL Server предполагает использование портала Microsoft Visual Studio Dev Essentials, показанного на рис. 10.14.

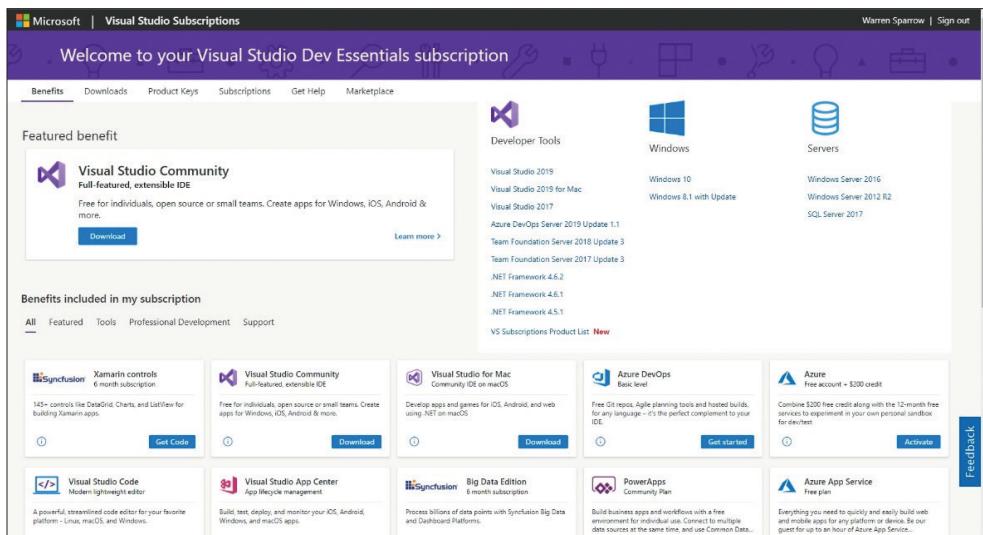


Рис. 10.14. Visual Studio Dev Essentials

Вы должны будете согласиться с условиями лицензии и авторизоваться при помощи учетной записи Microsoft или MSDN. При этом вы можете скачать инструмент **Visual Studio Community**, если желаете, но это не обязательно. Стоит отметить, что сайт поменял свой внешний вид – если раньше версии SQL Server располагались на главной странице, то теперь они перекочевали в раздел **Downloads**.

После выбора версии SQL Server можете переходить к ее загрузке. В результате будет загружен файл ISO, который вы можете открыть или записать на DVD. Первый способ будет более быстрым и легким. После распаковки необходимо запустить процесс установки и следовать всем тем же шагам, что и в предыдущей инструкции. Понадобится определенное время, но в результате у вас появится рабочая версия SQL Server на вашем компьютере.

УСТАНОВКА SQL SERVER MANAGEMENT STUDIO

Следующим шагом вам необходимо будет подключиться к SQL Server при помощи **Microsoft SQL Server Management Studio (SSMS)**. Хотя способов для подключения и взаимодействия с базами данных есть масса, этот является одним из наиболее простых. Вы можете либо щелкнуть по ссылке **Install SQL Server Management Tools** (Установить SQL Server Management Tools) в диалоговом окне, которое мы показывали на рис. 10.11, либо скачать этот пакет по адресу: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>. Загрузите на компьютер установочный файл Microsoft SQL Server Management Studio. Его объем составляет порядка 550 Мб. Устанавливаться приложение может довольно долго, так что запаситесь десятком свободных минут.

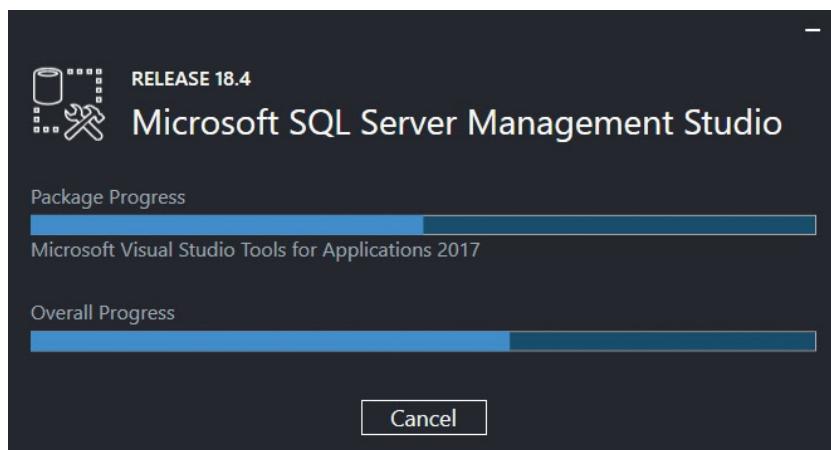


Рис. 10.15. Установка SSMS

После окончания процесса инсталляции программного пакета необходимо перезагрузить компьютер.

Откройте SSMS и установите связь со службой SQL Server, нажав в диалоговом окне, показанном на рис. 10.16, на кнопку **Connect** (Подключить).

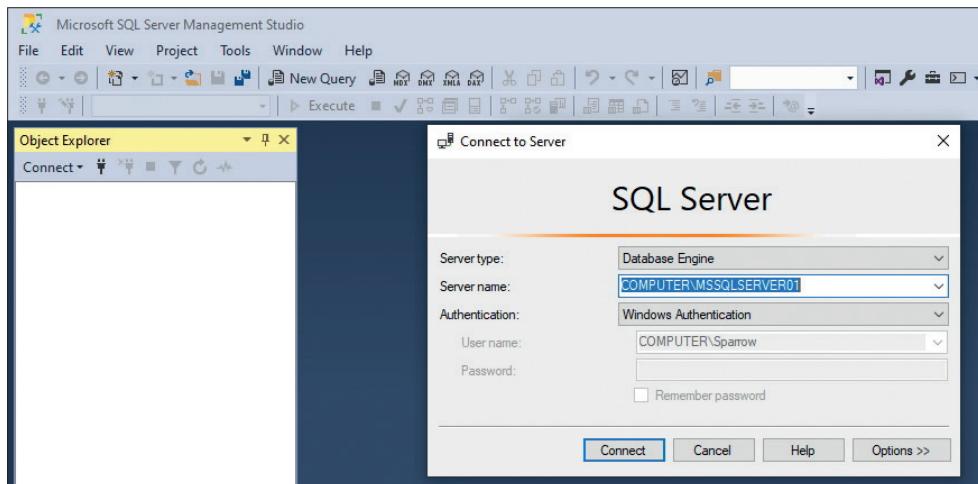


Рис. 10.16. Подключение к SQL Server

При успешном подключении к серверу вы увидите слева панель **Object Explorer** (Обозреватель объектов). Теперь нам нужно обновить базу данных, которую мы будем использовать для наших примеров. Соответствующие файлы резервных копий базы данных Microsoft AdventureWorks находятся по ссылке: <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15>.

Здесь вы можете загрузить различные версии базы данных. У нас установлены четыре версии до 2017 DW включительно. Для выполнения примеров

из этой главы лучше будет скачать более одной базы данных, чтобы иметь возможность при помощи параметров подключаться к разным источникам. Но можно обойтись и одной установленной базой данных.

После загрузки файлов резервных копий баз данных вам необходимо развернуть их на сервере.

Нажмите правой кнопкой мыши на папке **Databases** (Базы данных) и выберите пункт **Restore Database...** (Восстановить базу данных...), как показано на рис. 10.17.

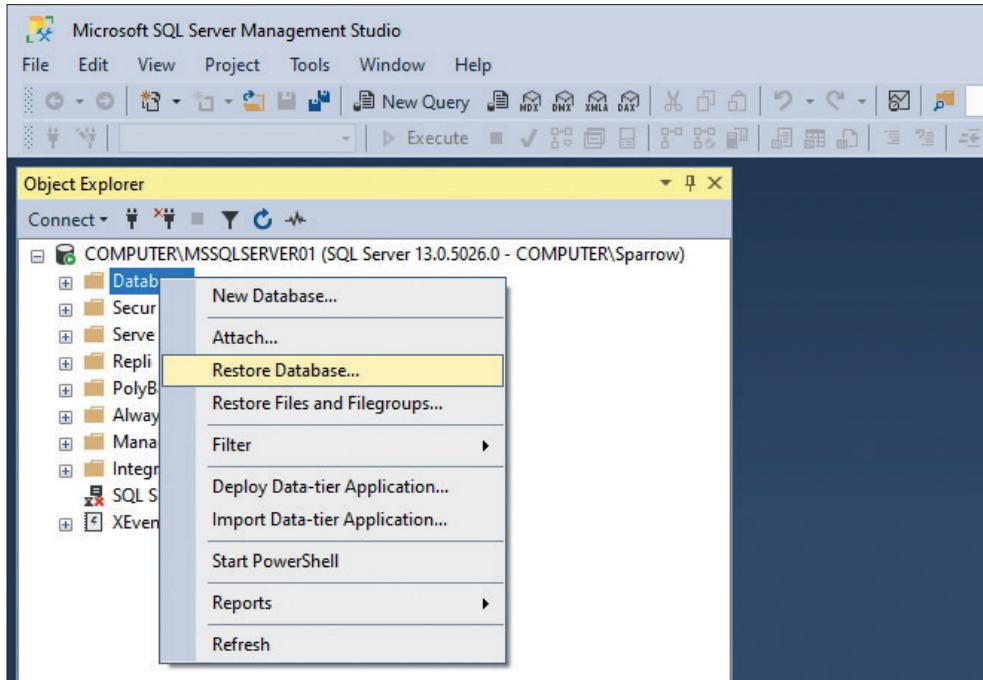


Рис. 10.17. Восстановление базы данных

Выберите пункт **Device** (Устройство) в разделе **Source** (Источник) и нажмите на кнопку с тремя точками. Затем выберите файл с резервной копией базы на диске и восстановите его, как показано на рис. 10.18.

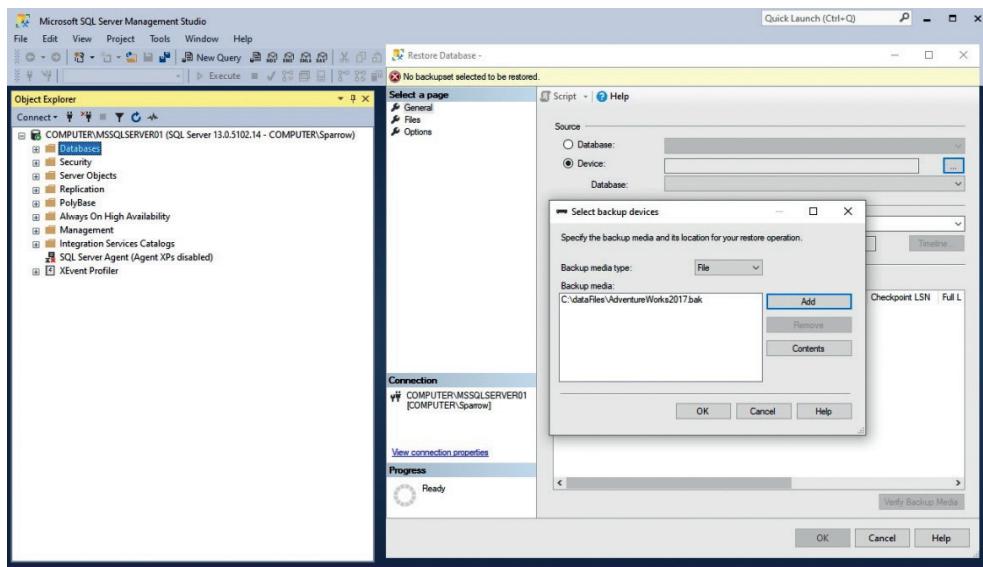


Рис. 10.18. Восстановление базы данных AdventureWorks

База данных появится в списке слева с соответствующим именем, что видно на рис. 10.19.

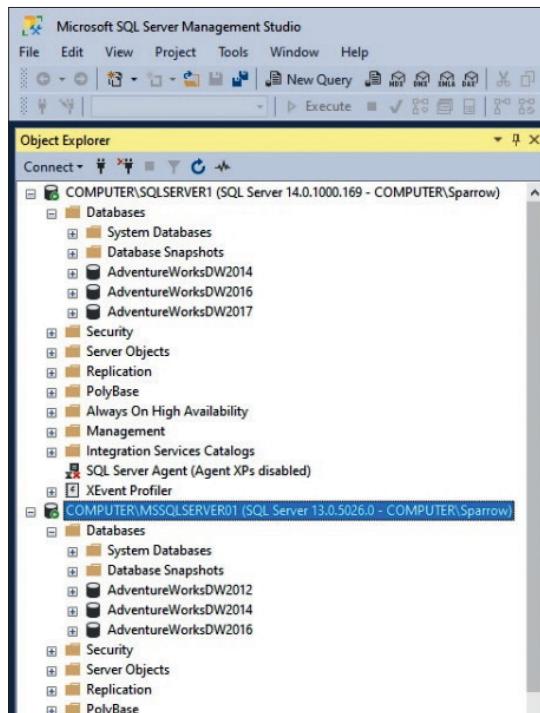


Рис. 10.19. Базы данных AdventureWorks DW

Мы повторили перечисленные шаги для восстановления всех необходимых нам версий базы данных AdventureWorks. Если вы хотите, чтобы другие пользователи получили доступ к вашим базам данных, откройте папку **Security** (Безопасность), нажмите правой кнопкой на пункт **Users** (Пользователи) и добавьте пользователя. Теперь все наши базы данных готовы к использованию.

Следующим шагом необходимо получить доступ к нужной базе данных из Power BI Desktop. Чтобы это сделать, откройте Power BI Desktop и выберите **SQL Server**, нажав на выпадающую кнопку **Get Data** (Получить данные). Здесь важно помнить, как вы называли свой экземпляр сервера. В окне навигации в Power BI введите `./servername` и нажмите на кнопку **OK**.

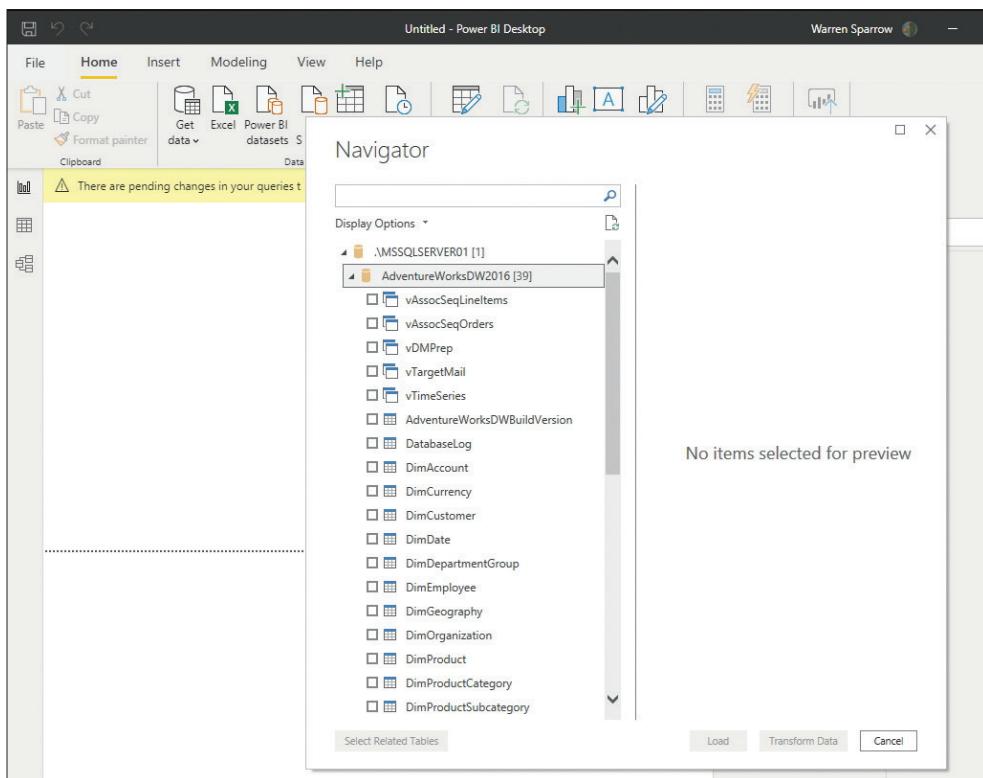


Рис. 10.20. Подключение к SQL Server

Мы создали два экземпляра SQL Server, чтобы можно было продемонстрировать переключение между ними и базами данных в наших примерах. Если у вас лишь один экземпляр, вы можете подключиться к нему.

Обнаружив нужную базу данных AdventureWorks, вы можете подключиться к ней, чтобы узнать, как применять различные параметры запросов.

ИСПОЛЬЗОВАНИЕ ПАРАМЕТРОВ

В данном разделе мы посмотрим, как можно применять параметры в запросах для осуществления фильтрации данных в Power BI Desktop. В 2016 году Power BI Desktop претерпел серьезные изменения, в том числе в отношении создания и применения параметров. В основном параметры используются при обращении к источникам данных, фильтрации, удалении и замене строк. Также можно загрузить параметры непосредственно в модель данных, чтобы можно было обращаться к ним из мер, вычисляемых столбцов, таблиц и отчетов.

Параметризация источника данных

В этом разделе мы узнаем, как можно подключаться к различным источникам данных, определенным в параметрах запроса, и загружать разные столбцы. Хорошой новостью является то, что различные приложения, такие как Salesforce, SharePoint и Power BI Desktop, позволяют использовать параметры при определении свойств подключения. Это означает, что у вас может быть определен один параметр для подключения к экземпляру SQL Server, а другой – для целевой базы данных.

Параметры не зависят от наборов данных, а значит, вы можете создавать их до или после добавления набора данных. Когда мы создавали параметры в главе 4, мы, если помните, определяли их и присваивали им исходное значение в редакторе запросов Power Query. После этого параметры появлялись на вкладке запросов, где мы могли впоследствии изменять и настраивать их.

Предположим, что у вас есть несколько клиентов, использующих одну и ту же структуру данных, но на разных экземплярах SQL Server и в базах с разными именами. Используя параметры запроса, мы можем переключаться между различными источниками данных и публиковать отчеты в разных службах Power BI. Посмотрим, как это можно сделать.

1. Откройте редактор запросов Power Query и нажмите на кнопку **Manage Parameters** (Управление параметрами) на вкладке **Home** (Главная страница).
2. В поле **Name** (Имя) введите `SqlSrvInstance`, а в поле **Description** (Описание) – примечание к параметру. Дальнейшая настройка параметра может отличаться в зависимости от того, сколько экземпляров SQL Server у вас установлено, при этом вы должны убедиться, что используемые имена экземпляров написаны правильно и адресуют существующие подключения. Также заметим, что перед названием экземпляра нужно ставить символы `\`, как показано на рис. 10.21.

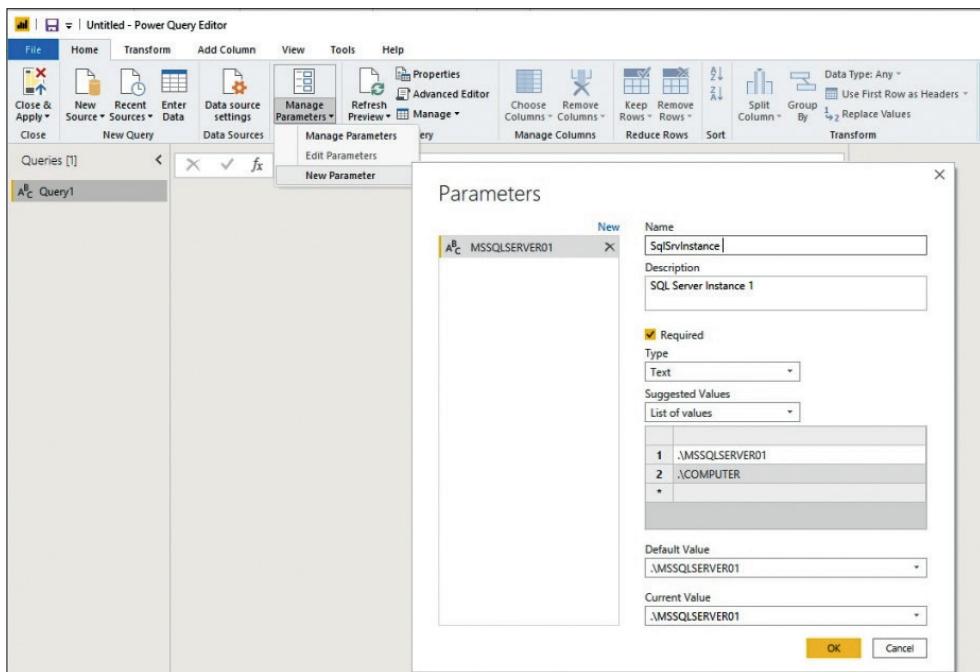


Рис. 10.21. Настройка параметра

3. В выпадающем списке **Type** (Тип) выберите **Text** (Текст), а в списке **Suggested values** (Предлагаемые значения) – вариант **List of values** (Список значений). Ниже появится табличка, в которой будет создаваться новая строка каждый раз, когда вы будете нажимать на клавишу **Enter**.
4. Выберите нужное имя экземпляра SQL Server в списках **Default value** (Значение по умолчанию) и **Current value** (Текущее значение).
5. По щелчку на кнопке **OK** диалоговое окно **Parameters** (Параметры) закроется, и параметр **SqlSrvInstance** появится на панели запросов, как видно на рис. 10.22.

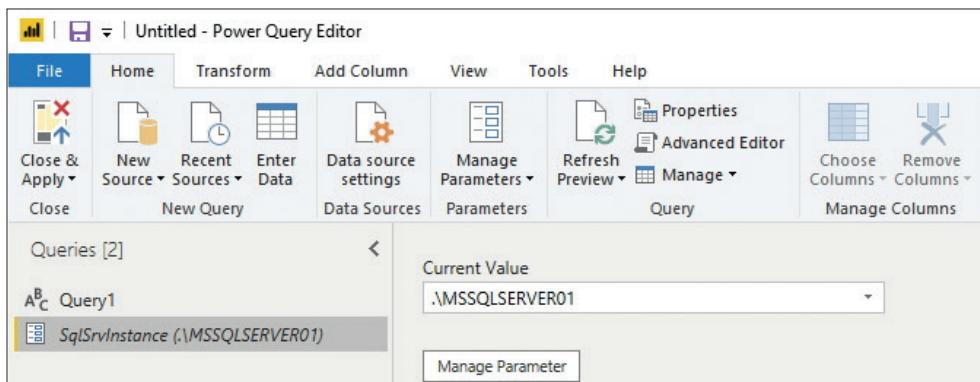


Рис. 10.22. Параметр на панели запросов

Теперь повторим все описанные выше действия, создав параметр для указания имени базы данных, к которой будет выполняться подключение. Настройте новый параметр согласно рис. 10.23 и назовите его **Database**.

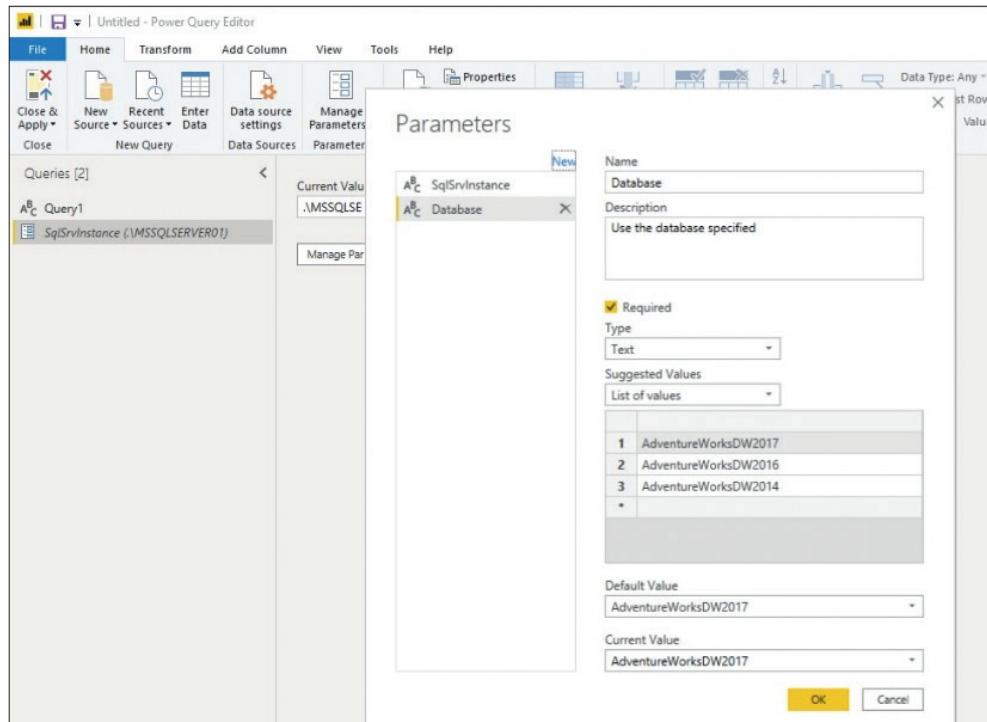


Рис. 10.23. Параметр Database

6. Нажмите на кнопку **OK**, а затем на **Close and Apply** (Закрыть и применить). Итак, мы создали параметры для подключения к источникам данных, и теперь самое время воспользоваться ими для извлечения данных из нужной нам базы данных в нужном экземпляре.
7. Мы будем использовать язык T-SQL для выполнения подключения. Но для этого необходимо убедиться в том, что флагок **Require user approval for new native database queries** (Требовать утверждения пользователя для новых машинных запросов баз данных) в настройках Power BI снят, как показано на рис. 10.24.

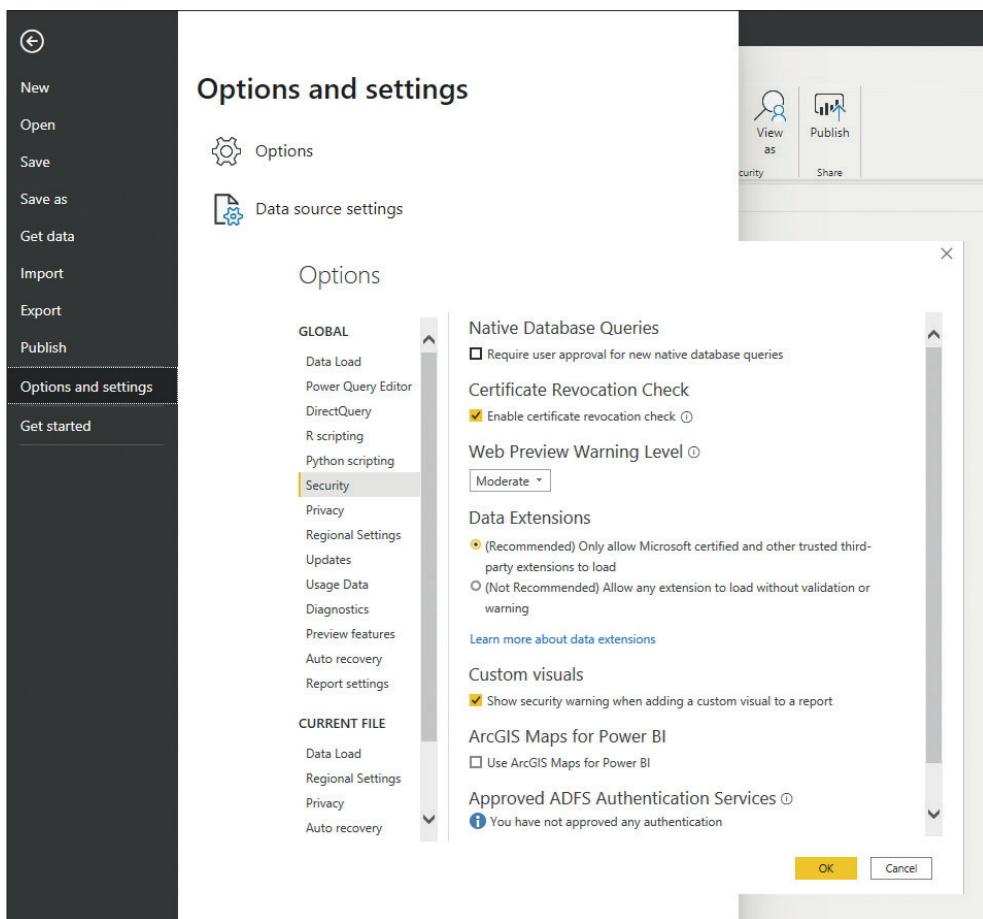


Рис. 10.24. Настройка требований утверждений пользователя

8. Найти этот флажок можно, перейдя в Power BI Desktop на вкладку **File** (Файл), далее выбрав пункт **Options and Settings** (Параметры и настройки), **Options** (Параметры) и открыв закладку **Security** (Безопасность).
9. Вернитесь в Power BI Desktop и выберите **SQL Server**, нажав на выпадающую кнопку **Get Data** (Получить данные).
10. Установите переключатели слева от полей **Server** (Сервер) и **Database** (База данных) в положение **Parameter** (Параметр) и выберите в списках значения **SqlSrvInstance** и **Database** соответственно, как показано на рис. 10.25.

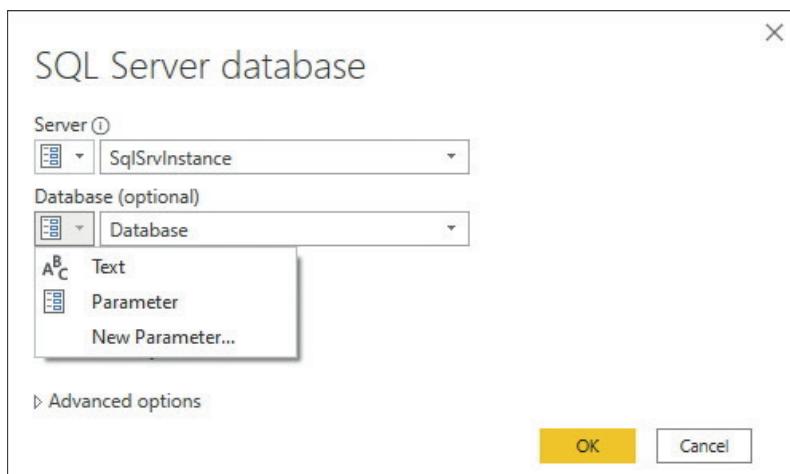


Рис. 10.25. Выбор параметров при подключении

11. Раскройте область **Advanced options** (Расширенные параметры) и вставьте следующий запрос T-SQL в поле **SQL statement** (Инструкция SQL):

```
SELECT h.SalesPersonID AS RepID,
CONCAT(p.LastName, ' ', p.FirstName) AS FullName,
CAST(SUM(h.SubTotal) AS INT) AS SalesAmounts
FROM Sales.SalesOrderHeader h INNER JOIN Person.Person p
ON h.SalesPersonID = p.BusinessEntityID
WHERE h.SalesPersonID IS NOT NULL
AND YEAR(h.OrderDate) = 2012
GROUP BY h.SalesPersonID, p.FirstName, p.LastName
ORDER BY FullName ASC;
```

12. Нажмите на кнопку **OK**, и если вы все сделали правильно, то окно предварительного просмотра будет выглядеть примерно как на рис. 10.26.

The screenshot shows a software interface titled 'SqlSrvInstance: Database'. It displays a table with three columns: 'RepID', 'FullName', and 'SalesAmounts'. The data consists of 29 rows of sales records. At the bottom of the window are three buttons: 'Load' (highlighted in yellow), 'Transform Data', and 'Cancel'.

RepID	FullName	SalesAmounts
285	Abbas, Syed	151257
287	Alberts, Amy	560092
280	Anzman-Wolfe, Pamela	963421
275	Blythe, Michael	3985375
283	Campbell, David	1351422
277	Carson, Jillian	3396776
281	Ito, Shu	2387256
274	Jiang, Stephen	431089
284	Mensa-Annan, Tete	1269909
276	Mitchell, Linda	4111295
289	Pak, Jae	4106064
279	Reiter, Tsvi	2188083
282	Saraiva, José	1870884
286	Tsoflias, Lynn	836055
288	Valdez, Rachel	1245459
278	Vargas, Garrett	1389837
290	Varkey Chudukatil, Ranjit	2646078

Рис. 10.26. Предварительный просмотр выполнения скрипта

Если взглянуть на заголовок окна предпросмотра, можно увидеть названия наших параметров: `SqlSrvInstance` и `Database`. Значит, подключение было успешно выполнено с использованием значений параметров по умолчанию.

Нажмите на кнопку **Load** (Загрузить), и информация из источника загрузится в набор данных. Перед продолжением вам наверняка захочется переименовать запрос подходящим образом. Это можно сделать в правой панели **Fields** (Поля), как показано на рис. 10.27.

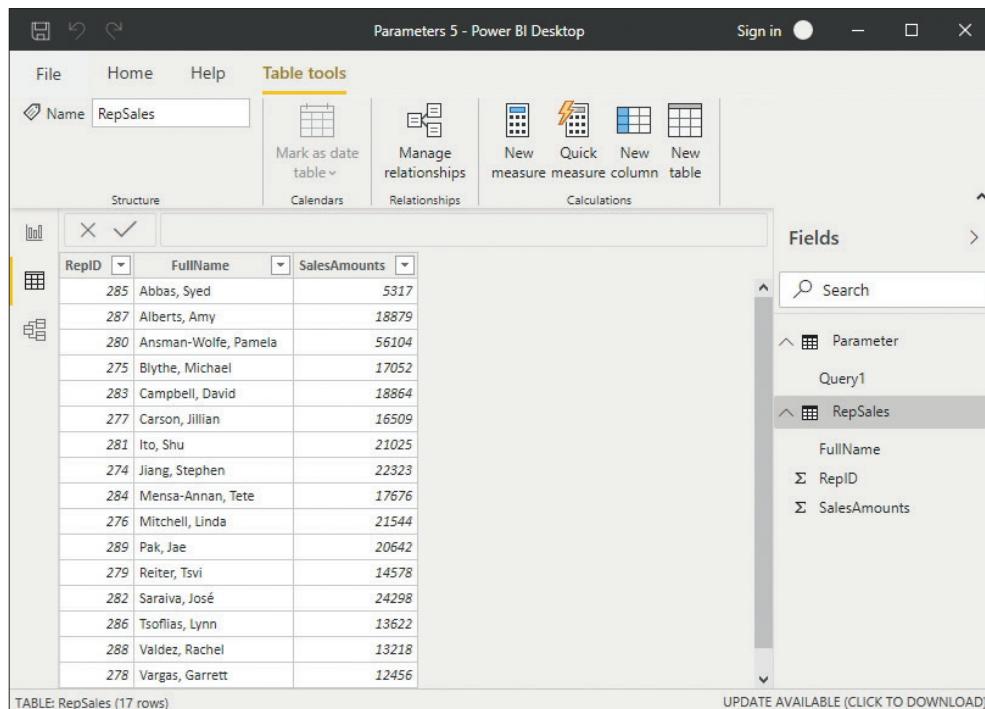


Рис. 10.27. Параметризованный запрос

Итак, мы создали два разных параметра для указания имени экземпляра SQL Server и базы данных. В результате мы можем использовать их для разных наборов данных путем настройки.

Если взглянуть на код на языке M, сгенерированный после применения выражения T-SQL, можно заметить, как изящно были встроены в итоговую конструкцию наши параметры:

```
= Sql.Database(SqlSrvInstance, Database)
```

Несмотря на созданные нами параметры, на панели **APPLIED STEPS** (Приимененные шаги) в правой части экрана по-прежнему будет присутствовать шаг **Filtered Rows** (Строки с примененным фильтром). В следующем разделе мы посмотрим, как можно хранить в параметре дату.

Использование параметров для фильтрации данных

Параметры допустимо использовать для наложения *фильтрации* на набор данных. В нашей инструкции на языке T-SQL из предыдущего раздела был такой фрагмент: `AND YEAR(h.OrderDate) = 2012`. Мы можем заменить жестко прописанный 2012 год на параметр, что позволит нам менять его значение динамически. Создайте новый параметр с именем `YearSales`, воспользовавшись инструкциями из предыдущего раздела, и введите в список возможных значений годы с 2011-го по 2015-й, как показано на рис. 10.28.

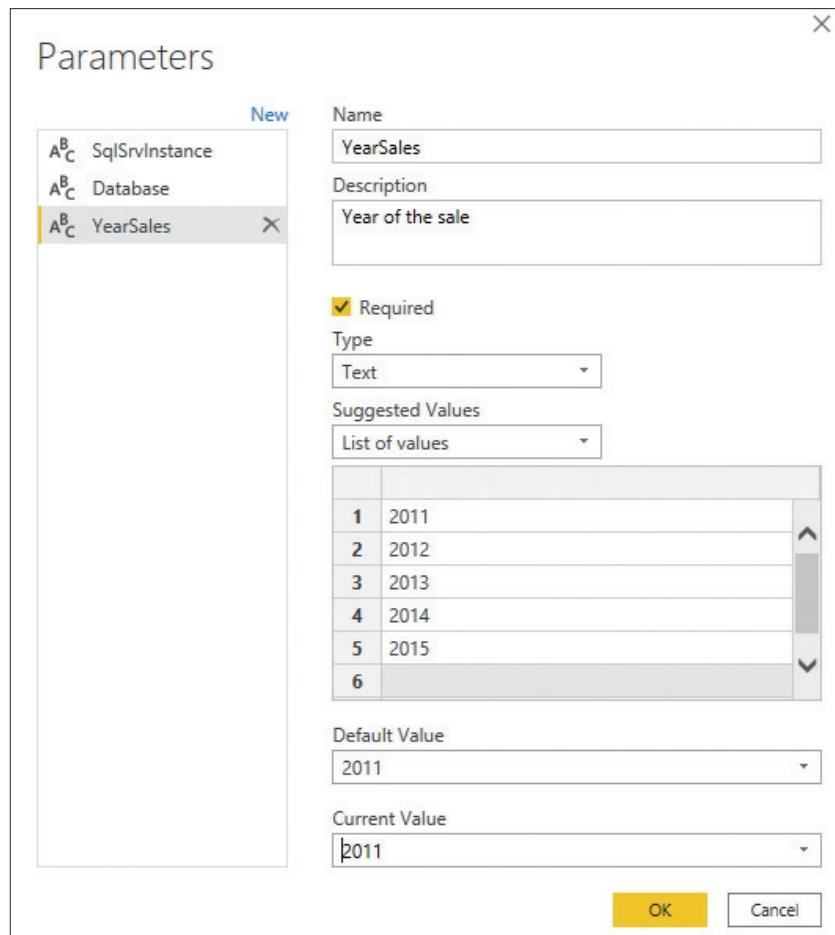


Рис. 10.28. Параметр YearSales

Теперь, когда у нас есть новый параметр, можно в коде на языке M поменять 2012 на « & YearSales & » (включая кавычки), как показано на рис. 10.29.

The screenshot shows the Power Query Editor with the M code. The code includes a parameter reference &YearSales& which is highlighted in yellow. The code is as follows:

```
= Sql.Database(SqlServerInstance, Database, [Query="SELECT n.SalesPersonID AS pId, p.FirstName AS pFirstName, p.LastName AS pLastName, h.Total AS pTotalSales, h.SubTotal AS hSubTotal FROM Sales.SalesOrderHeader h INNER JOIN Person.Person p ON h.SalesPersonID = p.BusinessEntityID AND h.SalesPersonID = p.BusinessEntityID WHERE h.OrderDate >= #1/1/2012 AND h.OrderDate < #1/1/2013 AND h.Total >= &YearSales& ORDER BY p.LastName, p.FirstName, h.Total DESC"], [ConnectionString="Data Source=SQLServer2017;Initial Catalog=AdventureworksLT2017;Provider=SQLNCLI11.1;Integrated Security=SSPI;"], [DefaultContainer=1], [Cache=true], [MaxRows=0], [ImpersonateUser=true], [QueryType="Table"])]
```

Рис. 10.29. Внедрение параметра в код на языке M

Смотрите, как здорово работает новый параметр, – вы можете выбирать разные годы, после чего возвращаться в запрос RepSales и видеть разные цифры, соответствующие выбранному году, как показано на рис. 10.30. Также заметьте, что при выборе 2015 года окно предварительного просмотра останется пустым ввиду отсутствия данных.

	RepID	FullName	SalesAmounts
1	280	Ansmann-Wolfe, Pamela	648486
2	275	Blythe, Michael	875824
3	283	Campbell, David	599988
4	277	Carson, Jillian	1311627
5			
6			
7			
8			
9			
10			

	RepID	FullName	SalesAmounts
1	285	Abbas, Syed	21267
2	287	Alberts, Amy	56638
3	280	Ansmann-Wolfe, Pamela	504932
4	275	Blythe, Michael	1057247
5	283	Campbell, David	490466
6	277	Carson, Jillian	1040093
7	281	Ito, Shu	777942
8	274	Jiang, Stephen	178584
9	284	Menss-Annan, Tete	600997
10	271	Mitchell, Linda	1271089
11	289	Pak, Jae	1382997
12	279	Reiter, Tsvi	787204
13	282	Saravina, José	1044811
14	286	Tschlias, Lynn	585756
15	288	Valdez, Rachel	581608
16	278	Vargas, Garrett	435949
17	290	Varkey Chudukatil, Ranjith	867519

Рис. 10.30. Выбор разных лет в параметре

Можно пойти еще дальше и изменять не только текст, но и математические формулы, о чем мы поговорим далее.

Использование параметров для изменения логики выражений

До сих пор мы применяли параметры, если нам нужно было изменить текст в выражении. Но текст может выступать и в другом качестве, например участвовать в формировании математических формул. Допустим, с помощью параметров мы легко можем подменить функцию SUM на AVG, тем самым влияя на агрегацию в выражении. Хотя этот подход можно использовать применительно к любым формулам, я покажу, как это работает, на примере самых распространенных функций SUM, AVG, MIN и MAX.

Создайте новый параметр и назовите его MathsAgg. В качестве возможных значений задайте ему следующие: SUM(h.SubTotal), AVG(h.SubTotal), MAX(h.SubTotal) и MIN(h.SubTotal), как показано на рис. 10.31.

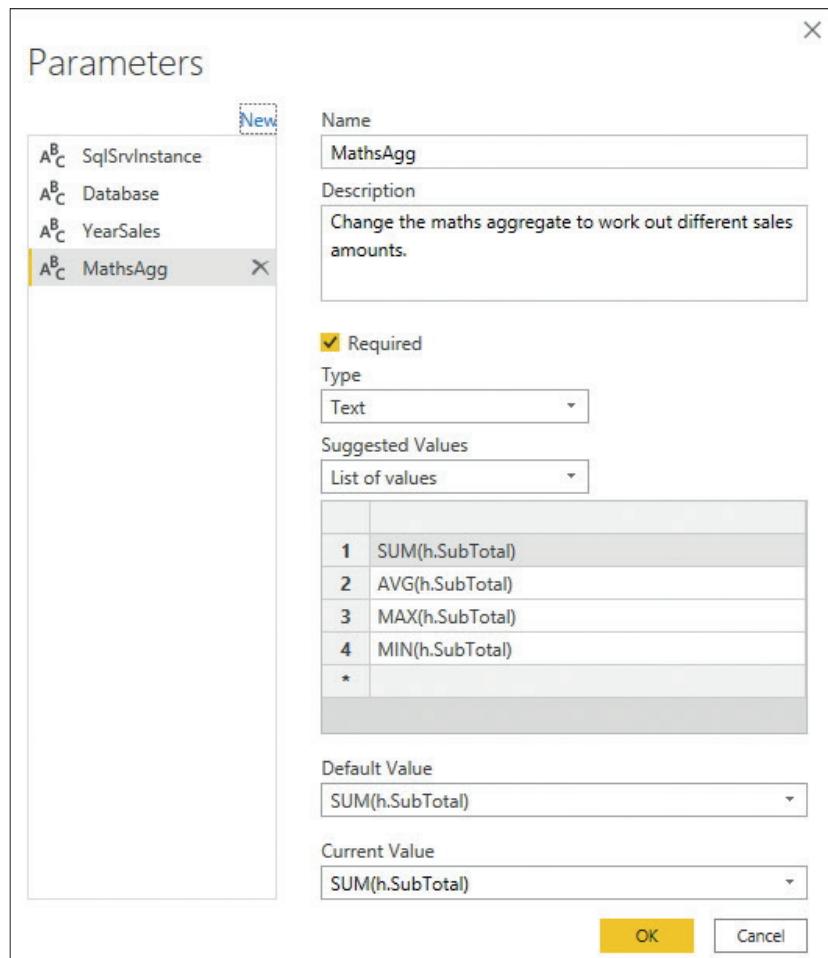


Рис. 10.31. Параметр MathsAgg

В базе данных может быть огромное количество столбцов, к которым можно применить агрегацию. Например, это может быть колонка `DiscountAmounts`, к которой можно применить агрегацию `MIN(h.DiscountAmounts)` для вычисления минимальной суммы скидки. При этом мы можем использовать разные параметры для разных столбцов, если наборы данных поддерживают такую функциональность.

Нам снова потребуется внести изменения в код на языке M. Заменим `SUM(h.SubTotal)` на « & MathsAgg & », включая кавычки, как показано на рис. 10.32.

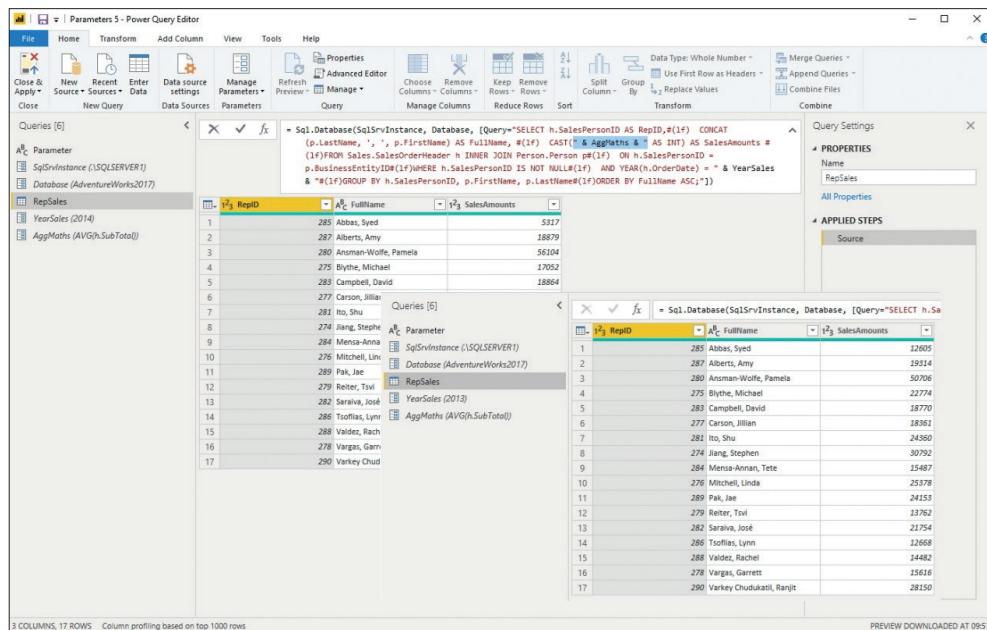


Рис. 10.32. Отредактированный код на языке M

И снова, если источник данных поддерживает такой функционал, вы можете применять различные параметры к разным колонкам.

Здесь мы использовали параметры для корректировки математических функций в коде, а в следующем разделе посмотрим, как можно применять их для сортировки данных в нужном нам порядке.

Использование параметров для изменения порядка сортировки

В Excel и Power BI мы можем сортировать данные как по возрастанию, так и по убыванию. Следовательно, можно использовать параметры для осуществления динамической сортировки столбцов. При этом мы можем управлять не только порядком сортировки, но и указывать конкретные столбцы, по которым должны быть упорядочены данные в итоговом наборе.

Как и раньше, создайте новый параметр, назовите его OrderResults и перечислите в соответствующей таблице следующие допустимые значения: FullName ASC, FullName DESC, SalesAmounts ASC и SalesAmounts DESC, как показано на рис. 10.33.

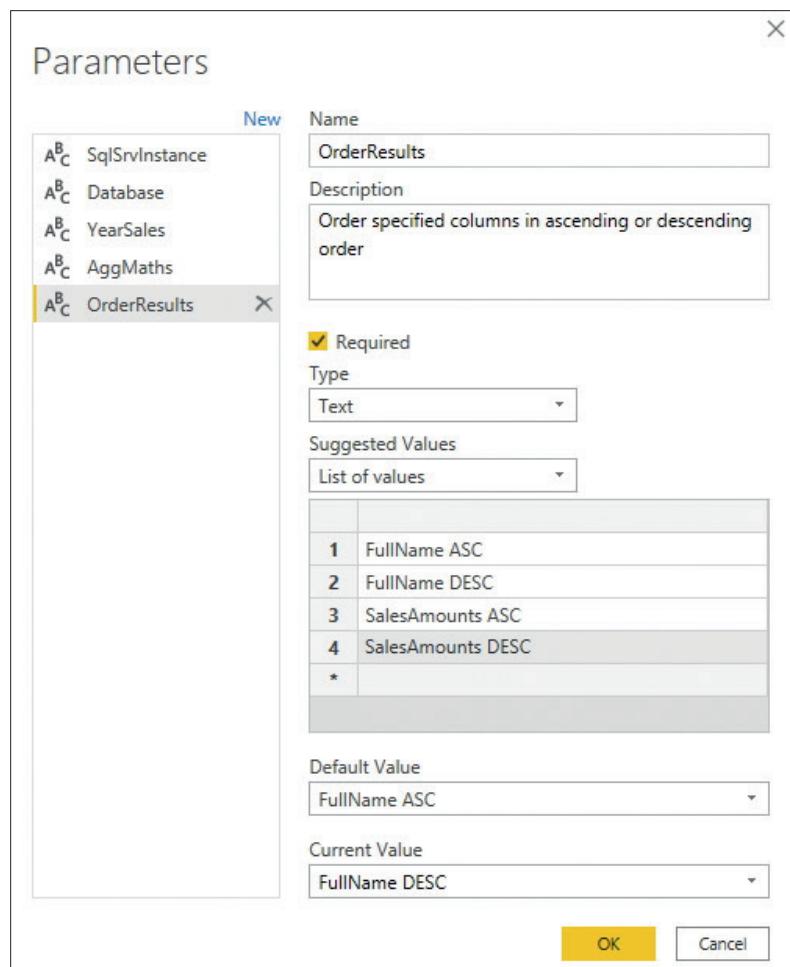


Рис. 10.33. Параметр OrderResults

Привычным уже движением изменим код на языке M, поменяв текст FullName ASC на « & OrderResults & », включая кавычки, как показано на рис. 10.34.

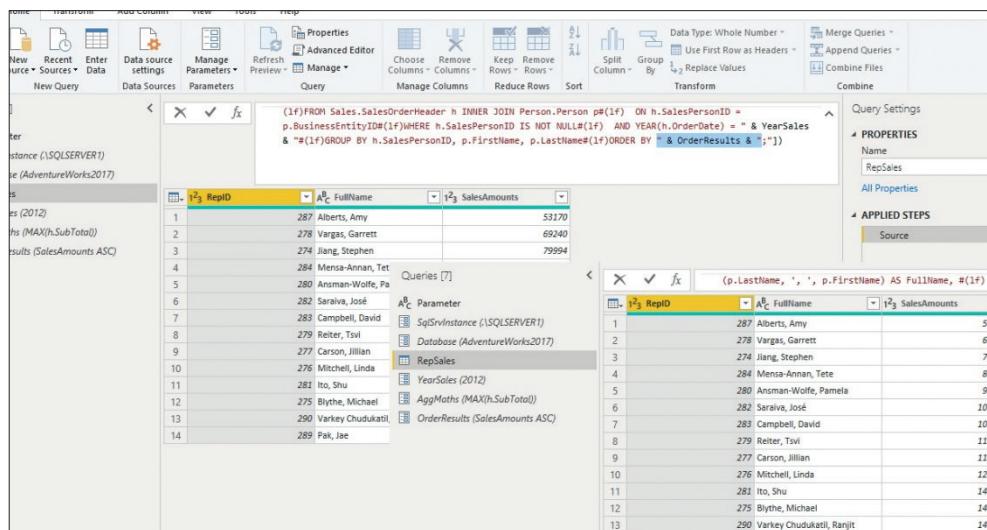


Рис. 10.34. Внедрение в код параметра OrderResults

В данной главе мы научились использовать параметры для выполнения динамической фильтрации результатов. Ранее в этой книге мы в основном применяли фильтрацию на основании панели **APPLIED STEPS** (Примененные шаги). Но проблема состоит в том, что эти шаги довольно статичны – упорядочив их однажды, бывает нелегко внести изменения. В то же время параметры позволяют вносить корректировки динамически, без изменения последовательности шагов – фильтрация в этом случае применяется непосредственно к набору данных.

Последний совет, касающийся параметров, который хотелось бы дать, заключается в том, что при создании шаблонов с их использованием желательно сохранять промежуточные версии в виде отдельных файлов. Например, при написании этой главы мы последовательно сохранили файлы под названиями Chapter10.Parameter1, Chapter10. Parameter2 и т. д. Причин для этого две:

- у меня есть отдельные файлы с каждым параметром, которые можно использовать в качестве примеров;
- если код на языке M в какой-то момент сломается, всегда можно будет откатиться к одной из версий файла.

Одним из немногих недостатков параметров является то, что для изменения их значений нам каждый раз приходилось открывать редактор запросов Power Query, что отнимает времени. Сейчас же мы посмотрим, как можно менять значения параметров непосредственно на вкладке **Data** (Данные) в Power BI.

Изменение параметров на вкладке Data (Данные)

В Power BI существует возможность задавать значения всех созданных параметров непосредственно на вкладке **Data** (Данные) – без необходимости переключаться в режим редактора запросов Power Query, – что здорово экономит времени.

Перейдите на вкладку **Home** (Главная страница) и в выпадающей кнопке **Transform data** (Преобразование данных) выберите пункт **Edit Parameters** (Изменить параметры). Откроется диалоговое окно, показанное на рис. 10.35.

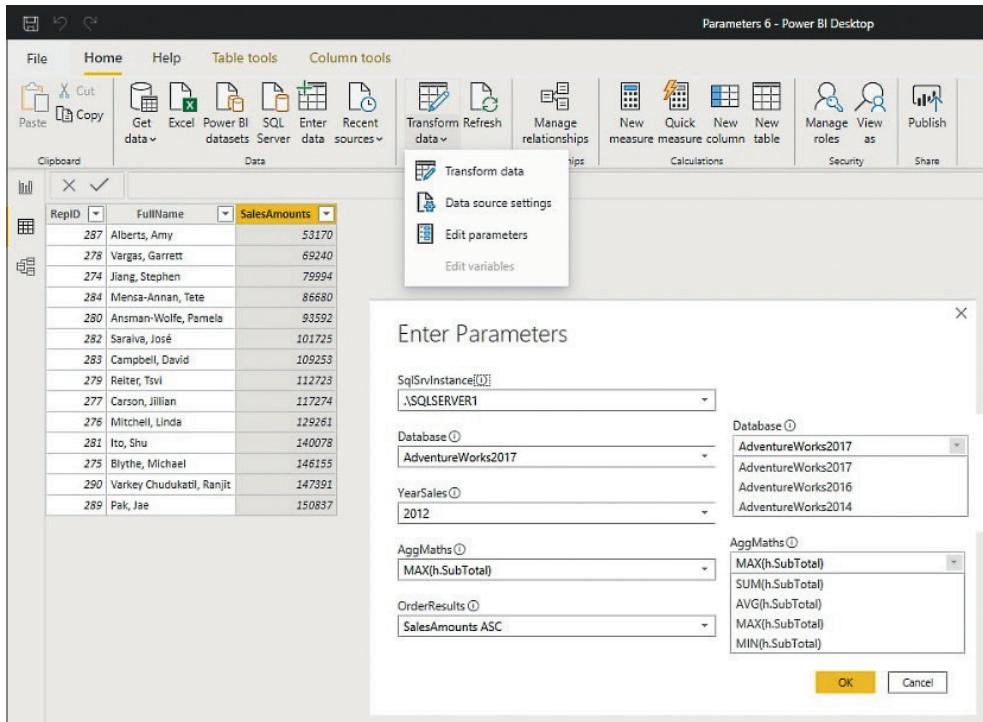


Рис. 10.35. Ввод параметров на вкладке данных

Назначение этого окна вполне понятно. Вы можете настроить нужные вам параметры, выбрав определенные значения. Нажав на кнопку **OK**, вы автоматически примените фильтрацию к набору данных на основании выбранных значений. Можно также менять значения параметров, находясь на вкладке **Report** (Отчет), что приведет к обновлению ваших элементов визуализации.

ЗАКЛЮЧЕНИЕ

В начале главы мы кратко сравнили языки, присущие Excel и Power BI, на примере выполнения конкатенации данных. Мы поговорили об использовании оператора **амперсанд** (&) применительно к объединению строк, дат и других объектов, после чего посмотрели на сходства и различия этого оператора со связкой функций **Text.From** и **Text.Combine**.

Установкой SQL Server в крупных компаниях занимаются технические специалисты, но все становится иным в меняющемся климате масштабируемых онлайн-платформ, что позволяет вам приобретать только то, что нужно, и делает решения более доступными для маленьких организаций. Внимательно следуйте всем шагам при установке системы управления базами данных, по-

скольку одно неверное движение, например на этапе настройки аутентификации, может привести к дальнейшей невозможности получить доступ к серверу со стороны пользователей.

В заключительной части главы мы более подробно рассмотрели параметры, включая их настройку и выбор значений. Мы научились использовать параметры для доступа к источникам данных, а также для выполнения динамической фильтрации и сортировки в возрастающем и убывающем порядке. Наконец, мы рассмотрели возможность изменения значений параметров непосредственно на вкладке **Data** (Данные) в Power BI без перехода в редактор запросов Power Query.

В следующей главе мы сосредоточим внимание на пользовательских функциях в языке M.

Глава 11

Создание базовых пользовательских функций

В главе 9 мы познакомились с языком M, его синтаксисом и структурой, а также узнали способ показать все функции и их определения в интерфейсе Power Query.

В данной главе мы продолжим изучать язык M, научимся писать с нуля *пользовательские функции* (custom function) и создавать столбец с датой и временем при помощи функций.

Основные темы этой главы:

- создание функций на языке M;
- создание столбца типа DateTime с использованием трех функций языка M.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Вам понадобится интернет-соединение для загрузки соответствующих файлов с GitHub. Файлы с кодом для этой главы можно скачать по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

В этой главе мы исходим из предположения о том, что вы внимательно прочитали главы 9 и 10. Также вы должны уметь загружать информацию из различных источников данных в Excel или Power BI и комфортно чувствовать себя в интерфейсе Power Query.

Видеофрагмент, посвященный данной главе, располагается по адресу: https://www.youtube.com/watch?v=arwkny-WhIk&list=PLelcvrwLe186O_GJEzs47WaZXwZjwTN83&index=12&t=0s.

СОЗДАНИЕ ФУНКЦИЙ В ЯЗЫКЕ M

Как и в Excel, в Power Query (и Power BI) присутствует множество функций, которые можно использовать для подготовки и преобразования данных. И количество этих функций увеличивается от версии к версии. Когда мы нажимаем на кнопки и применяем различные настройки в интерфейсе, под капотом Power Query неустанно трудится движок, генерируя десятки строк кода, обеспечивающего выполнение требуемого действия.

В главе 9 мы уже упоминали об очень полезной инструкции `#shared`, позволяющей при вводе в строку формул получить полный список доступных функций языка M. Все эти функции можно использовать при построении запросов к данным, но иногда разработчику требуется написать собственную функцию, которая облегчит ему дальнейшее проектирование и поддержку системы преобразования данных. Под этим имеется в виду то, что он без труда сможет повторно использовать код, созданный ранее, а не писать его заново каждый раз для каждого очередного запроса. Мы подробно рассмотрим процесс преобразования запроса в *пользовательскую функцию* при помощи расширенного редактора Power Query.

В этом разделе мы напишем пользовательскую функцию, которая может быть применена к любому файлу. Представьте, что вы работаете с ежемесячно обновляемыми данными. Скажем, раз в месяц вы получаете информацию для менеджеров и регулярно экспортируете данные для составления годового прогноза. Мы вполне можем поручить эту задачу Power Query, написав соответствующую пользовательскую функцию. Разумеется, в каждой ежемесячной книге значения при этом будут отличаться.

Здесь мы научимся редактировать запрос для изменения пути к файлам, после чего создадим копию запроса и поработаем с расширенным редактором, чтобы преобразовать код в параметр и получить на выходе пользовательскую функцию. После этого мы попробуем вызвать нашу функцию путем подключения к папке и отображения результатов в таблице, как показано на рис. 11.1.

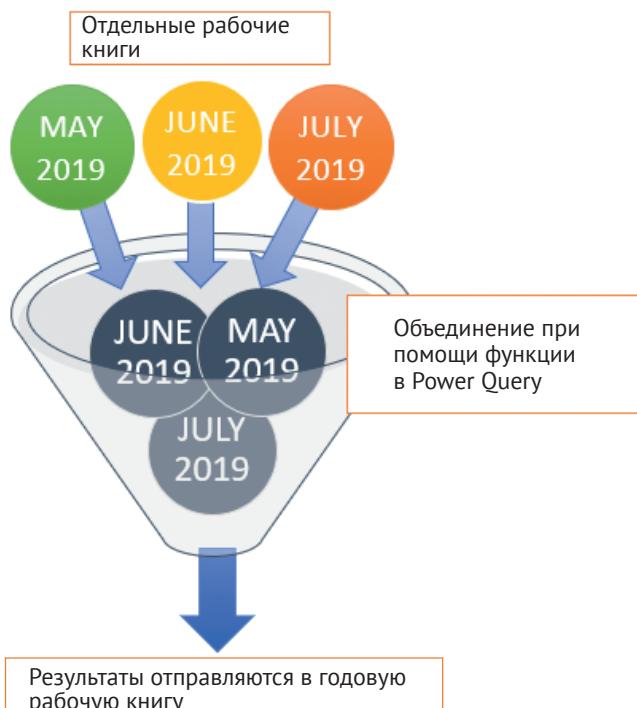


Рис. 11.1. Графическое представление преобразования разрозненных данных в одну выходную таблицу с использованием одного параметра

Для этого примера нам понадобится файл Power BI под названием ChoklatoFlakSales.pbix, включающий в себя запрос MaySales.

Если хотите, можете начать с нуля и импортировать книгу Excel с названием May-ChoklatoFlakSales в Power BI. Рабочие книги Excel имеют полностью идентичную структуру и формат. Для этого примера мы использовали книги May-ChoklatoFlakSales.xlsx, June-ChoklatoFlakSales.xlsx и July-ChoklatoFlakSales.xlsx.

Приступим, разбив наш сценарий на три главных шага.

Изменение пути к файлам в запросе на локальную папку

Поскольку вы будете использовать файлы с кодом для этого примера, для начала необходимо изменить путь к файлам таким образом, чтобы он указывал на локальную папку. Это делается для того, чтобы при открытии существующего файла Power BI (в нашем случае завершенного файла с примером), содержащего запрос, была произведена загрузка файлов из корректного места. Посмотрим, как это можно сделать.

Примечание

Если желаете, можете загрузить рабочую книгу в Power BI непосредственно с GitHub.

1. Откройте в Power BI Desktop файл ChoklatoFlakSales.pbix.
2. На панели запросов вы увидите запрос с именем MaySales.
3. Далее нам необходимо преобразовать наши данные при помощи Power Query. Запустите редактор запросов Power Query, нажав на вкладке Home (Главная страница) на кнопку **Transform Data** (Преобразование данных).
4. Нажмите на иконку с шестеренкой рядом с шагом **Source** (Источник) на панели **APPLIED STEPS** (Примененные шаги) в правой части экрана.
5. Вы увидите, что в главном окне появится ошибка, возникшая из-за того, что путь указывает на несуществующую папку, в результате чего запрос не загрузится.

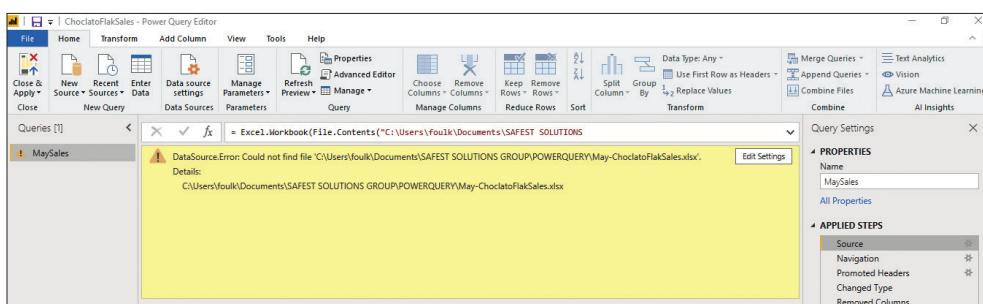


Рис. 11.2. Ошибка, возникшая по причине неправильного пути к файлу

6. Нажмите на кнопку выбора пути и укажите путь к папке, в которой у вас располагается файл May-Choklatoflaksales.xlsx. Выделите файл для обновления пути, после чего нажмите на кнопку **Open** (Открыть), как показано на рис. 11.3. Щелкните по кнопке **OK** для изменения пути.

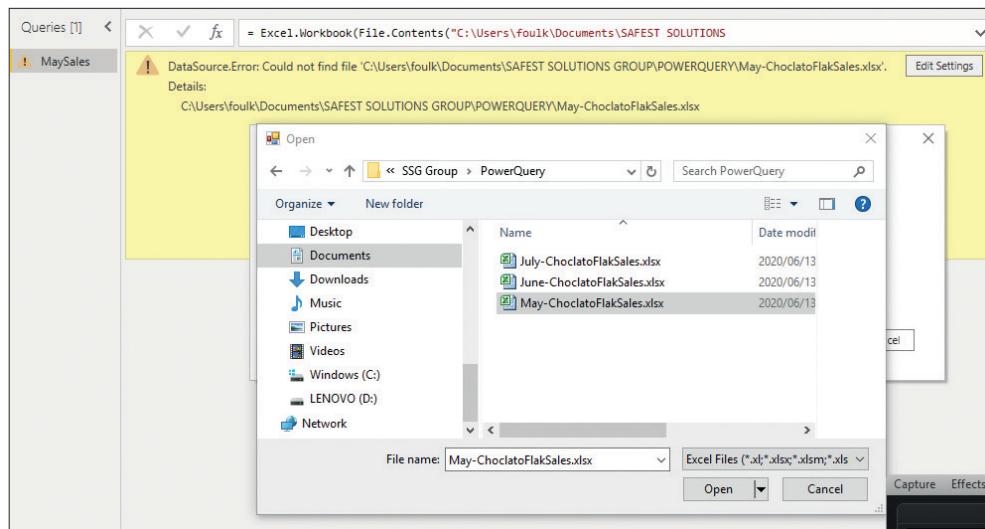


Рис. 11.3. Обновление файла в Power Query

7. Теперь путь к файлу в Power Query обновлен, и запрос загрузится. Узнаем, что нужно сделать, чтобы вручную преобразовать запрос в функцию.

Создание функции

Следующее действие, которое нам необходимо выполнить, состоит в преобразовании запроса в функцию и передаче ей параметра. Если вы не знаете, что такое параметры, прочитайте главу 10, в которой мы подробно обсуждали эту тему.

Посмотрим, что нужно сделать, чтобы преобразовать запрос в функцию.

- Для начала создадим дубликат запроса **MaySales** в Power Query.
- Для этого щелкните правой кнопкой мыши на запросе **MaySales** и из предложенного списка выберите пункт **Duplicate** (Дублировать).
- Копия исходного запроса появится с именем **MaySales(2)** под его оригиналом.
- Откройте расширенный редактор, чтобы изменить код на языке M. Для этого нажмите на кнопку **Advanced Editor** (Расширенный редактор) на вкладке **View** (Просмотр), как показано на рис. 11.4.

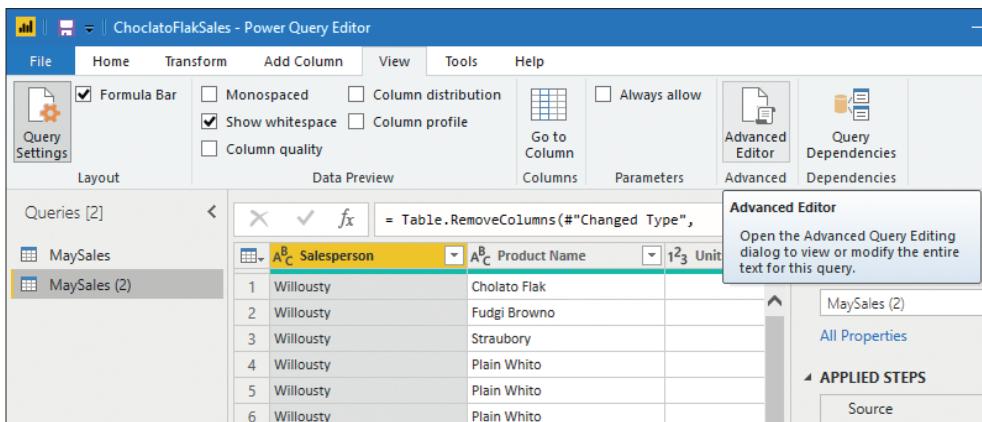


Рис. 11.4. Расширенный редактор Power Query

5. Нам необходимо интегрировать в этот код входной параметр, чтобы можно было запускать этот запрос в любом источнике данных внутри папки. На рис. 11.5 выделен фрагмент кода, который нам предстоит изменить. В данный момент этот фрагмент представляет собой путь к источнику данных, который мы импортировали в Power BI Desktop.

The screenshot shows the 'Advanced Editor' dialog for the query 'MaySales (2)'. The code is identical to the one shown in Figure 11.4. A note at the bottom left of the code area says: 'No syntax errors have been detected.'

Рис. 11.5. Код, который мы преобразуем во входной параметр при помощи расширенного редактора

6. Чтобы снабдить этот код входным параметром, напишем первой строкой (выше инструкции let) следующий код:

```
(FileBinary as binary) as table =>
```

Результат показан на рис. 11.6.

```
(FileBinary as binary) as table =>
let
    Source = Excel.Workbook(File.Contents("C:\Users\foulk\Documents\SSG Group\PowerQuery\May-ChoclateFlakSales.xlsx"), null, true),
    SSGProducts_Sheet = Source[[Item="SSGProducts", Kind="Sheet"]][Data],
    #"Promoted Headers" = Table.PromoteHeaders(SSGProducts_Sheet, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Salesperson", type text}, {"Product Name", type text}, {"Units", Int64}}, [RemoveColumns="Column7", "Column8"]),
    in
    #"Removed Columns"
```

No syntax errors have been detected.

Done Cancel

Рис. 11.6. Включение входного параметра в запрос

7. Замените подсвеченный фрагмент (за исключением первой скобки) на имя переменной `FileBinary` и нажмите на кнопку **Done** (Готово), тем самым превратив обычный запрос в *параметрический* (parameter query). Такие запросы или функции можно определять по иконке **fx** слева от названия запроса на соответствующей панели, как показано на рис. 11.7.

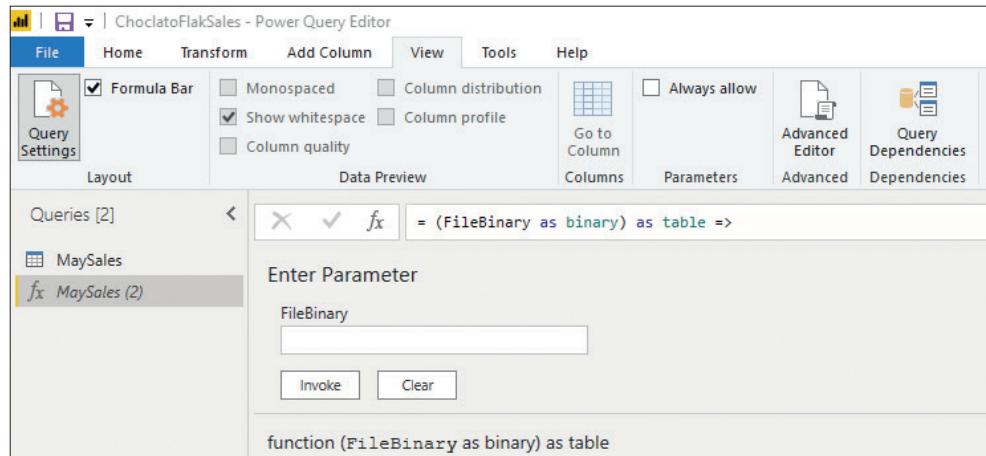


Рис. 11.7. Параметрический запрос

В этом разделе мы научились редактировать запросы при помощи расширенного редактора, снабжая их входными параметрами. Теперь пришло время проверить работу получившегося запроса.

Проверка параметрического запроса

В данном разделе мы убедимся в том, что наша функция работает так, как мы и планировали. Помните, что все файлы в папке должны быть одного типа, иначе может возникнуть ошибка. Посмотрим, как можно проверить работу функции.

- Находясь в редакторе запросов Power Query, на вкладке **Home** (Главная страница) нажмите на выпадающую кнопку **New Source** (Создать источник) и выберите пункт **More...** (Дополнительно...). Выделите пункт **Folder** (Папка), как показано на рис. 11.8, и нажмите на кнопку **Connect** (Подключить).

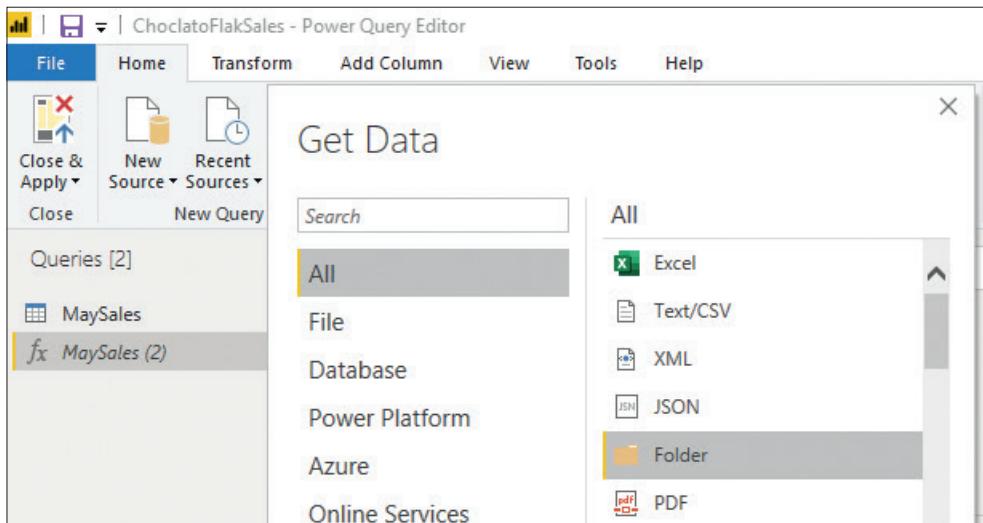


Рис. 11.8. Создание подключения к папке

- Нажмите на кнопку **Browse...** (Обзор...) и выберите папку для подключения, как видно на рис. 11.9.

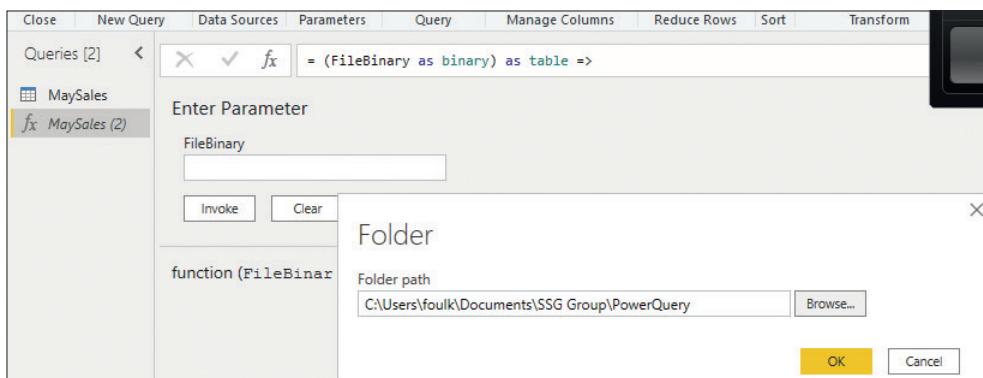


Рис. 11.9. Выбор пути к папке

- Нажмите на кнопку **OK**.
- После этого нажмите на кнопку **Transform Data** (Преобразовать данные), как на рис. 11.10, чтобы добавить новый запрос в интерфейс Power Query.

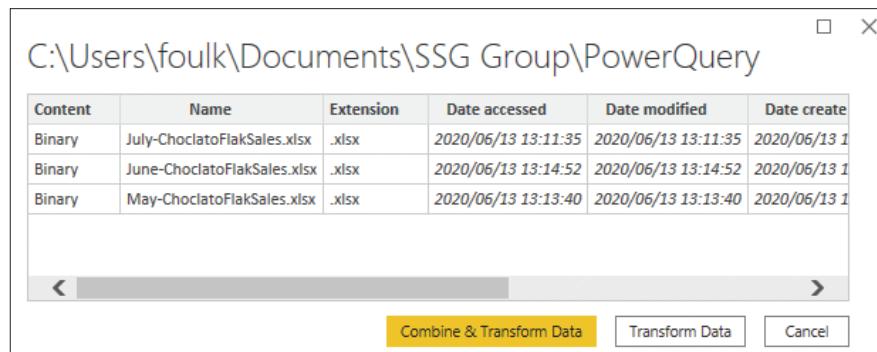


Рис. 11.10. Преобразование данных при подключении к папке

Мы будем использовать опцию вызова настраиваемой функции для ее применения ко всем файлам в папке, выбранной на предыдущем шаге. По существу, мы получим новый столбец с параметрическим запросом, вызывающим функцию.

5. Перейдите на вкладку **Add Column** (Добавление столбца) и нажмите на кнопку **Invoke Custom Function** (Вызвать настраиваемую функцию).
6. Введите имя колонки в поле **New column name** (Имя нового столбца). В нашем примере мы назовем столбец SalesInvoke. В выпадающем списке **Function query** (Запрос функции) выберите пункт **MaySales (2)**. В поле **FileBinary** оставьте **Content**, как показано на рис. 11.11.

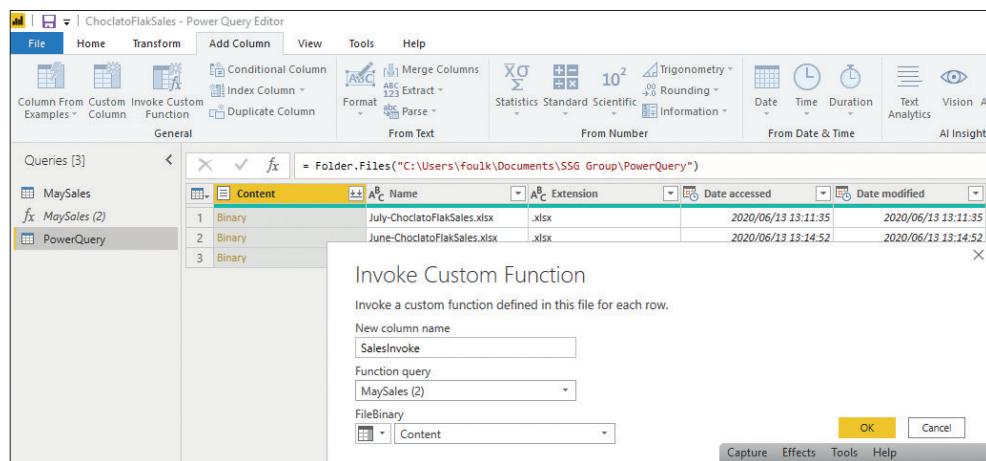


Рис. 11.11. Вызов пользовательской функции

7. Нажмите на кнопку **OK** для добавления столбца.
8. Новый столбец **SalesInvoke** появится в таблице, что видно по рис. 11.12.

The screenshot shows the Power Query Editor interface. In the main area, there is a table with three columns: 'Attributes', 'Folder Path', and 'SalesInvoke'. The 'SalesInvoke' column contains the formula = Table.AddColumn(Source, "SalesInvoke", each #"MaySales (2)"([Content])). To the right, the 'Query Settings' pane is open, showing the 'PROPERTIES' section with the name 'PowerQuery' and the 'APPLIED STEPS' section which includes the 'Invoked Custom Function' step.

Рис. 11.12. Новый столбец в запросе

9. Нажмите на столбец **SalesInvoke** для просмотра исходных данных в каждом источнике, как показано на рис. 11.13.

The screenshot shows the Power Query Editor with the 'ChoclateFlakSales' query selected. In the 'Querries' pane, there is one query named 'MaySales (2)'. In the main table area, the 'SalesInvoke' column is expanded, showing three rows of data. Each row contains a timestamp ('Date created') and a folder path ('Attributes'). The first two rows are labeled 'Record' and the third is labeled 'Table'.

	Salesperson	Product Name	Units	Price/Unit	Sales	Due Date
1	Willousty	Cholato Flak	45	11.6	522	2019/05/02
2	Willousty	Fudgi Brown	56	9.99	559.44	2019/05/02
3	Willousty	Straubory	123	10	1230	2019/05/02
	Willousty	Plain Whito	47	9.99	469.53	2019/05/02
	Willousty	Plain Whito	89	9.99	889.11	2019/05/02
	Willousty	Plain Whito	55	9.99	549.45	2019/05/02
	Smouthy	Cholato Flak	22	11	242	2019/05/02
	Smouthy	Fudgi Brown	65	16	1040	2019/05/02
	Smouthy	Straubory	89	13.6	1210.4	2019/05/02

Рис. 11.13. Щелчок мышью по пустому пространству в столбце раскрывает вложенную таблицу

10. Заметьте, что файлы в папке должны быть абсолютно одинакового типа и формата, в противном случае вы получите ошибку. Например, если мы обрабатываем файлы Excel, а в папке вместе с ними будут находиться файлы CSV, в этих строках будут выведены ошибки. Удалите лишние файлы из папки перед запуском функции.
11. Избавьтесь от столбцов, которые вам не нужны для запуска конкретной функции, как показано на рис. 11.14.

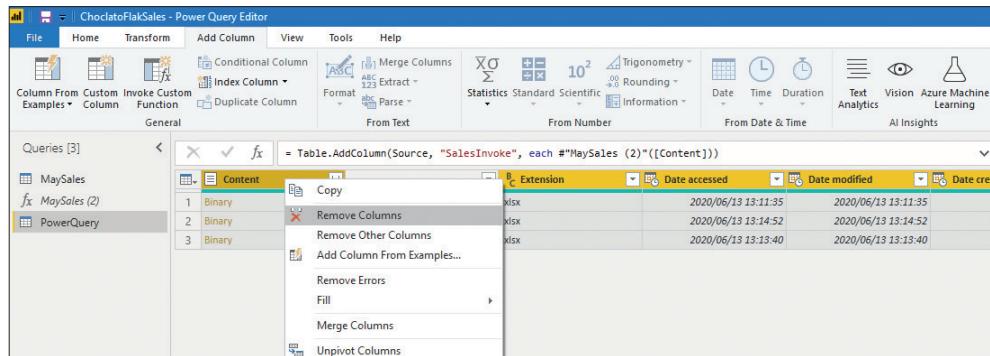


Рис. 11.14. Удаление ненужных столбцов из запроса

В данном случае мы удалим все столбцы, за исключением Name и SalesInvoke. В результате запрос приобретет вид, показанный на рис. 11.15.

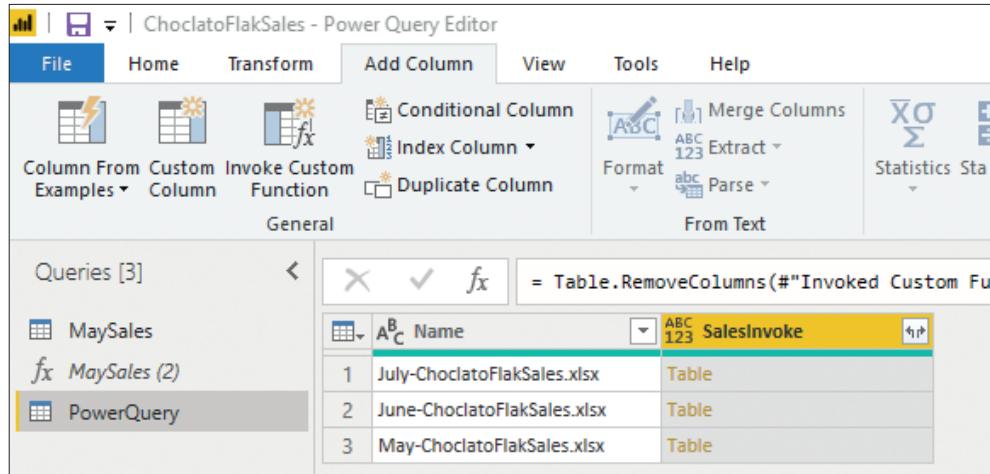


Рис. 11.15. Колонки, оставшиеся после операции удаления

12. На заключительном шаге нам нужно добавить данные из всех файлов Excel в единую рабочую книгу, чтобы мы видели май, июнь и июль вместе.

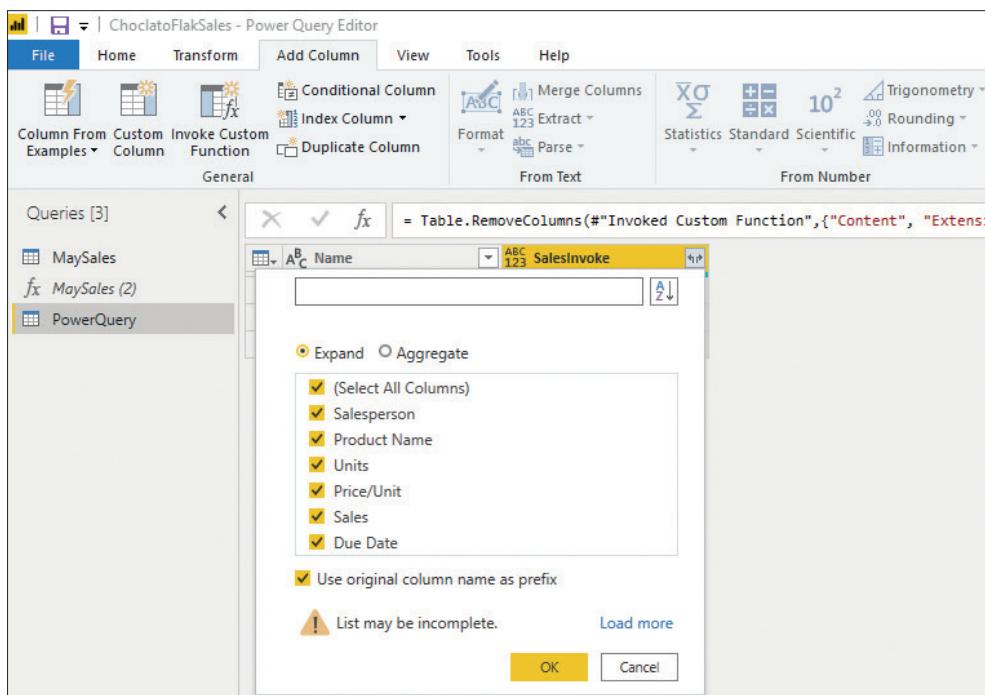


Рис. 11.16. Щелчок на иконке с двойными стрелками (Развернуть) покажет опции добавления

13. Нажмите на кнопку **OK**, чтобы посмотреть результаты.

Преимущество этого метода состоит в том, что любые файлы, появившиеся в папке, будут добавлены к таблице. Все, что вам необходимо сделать, – это обновить подключение.

14. Добавьте в нашу папку файл August-ChoklatoFlakSales.xlsx и посмотрите, что произойдет.

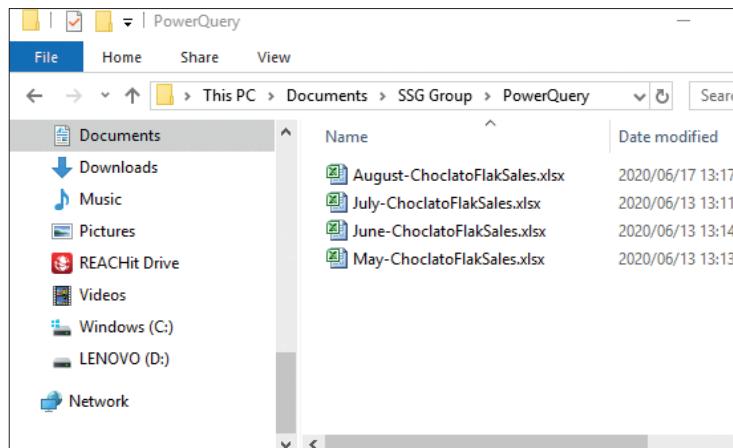


Рис. 11.17. Рабочая книга за август добавлена в общую папку

15. Нажмите на кнопку **Refresh All** (Обновить все) на вкладке **Home** (Главная) в Power Query для обновления данных. Вы увидите, что продажи за август добавились к набору данных, что видно по рис. 11.18.

The screenshot shows the Microsoft Power Query Editor window titled "ChoclatoflakSales - Power Query Editor". The ribbon at the top has tabs for File, Home, Transform, Add Column, View, Tools, and Help. The "Home" tab is selected. In the ribbon, under the "Transform" tab, there is a "Refresh" section with three buttons: Refresh Preview, Refresh All, and Cancel Refresh. The "Refresh All" button is highlighted with a red rectangle. On the left side of the editor, there is a sidebar with "Queries [3]" listed: "MaySales" (selected), "MaySales (2)", and "PowerQuery". The main area displays a table with 10 rows of data. The columns are labeled "Name" and "SalesInvoke.Salesperson". The data shows various salespeople and their names. The last row is highlighted with a red rectangle.

Рис. 11.18. Обновление набора данных

Важно отметить, что существует множество способов решить рассмотренную нами задачу. В своем методе решения мы старались продемонстрировать всю мощь Power Query при обновлении рабочих книг посредством добавления наборов данных. При этом очень важно понимать, как весь этот процесс организовать вручную, что мы и показали. Это гораздо полезнее, чем щелкать по кнопкам, не понимая, что происходит за сценой. Более короткий способ состоит в использовании кнопки **Combine Files** (Объединить файлы) на вкладке **Home** (Главная страница) в Power Query, как показано на рис. 11.19.

The screenshot shows the Power Query ribbon with the "Transform" tab selected. Under the "Combine" section, the "Combine Files" button is highlighted with a red rectangle. A tooltip window titled "Combine Files" appears, containing the text "Merge all files in a given column into a single table." Below the tooltip, there is a dropdown menu with the word "Name" and a "Name" button.

Рис. 11.19. Объединение файлов

Если у вас достаточно опыта в написании кода на языке M, вы можете воссоздать все шаги в расширенном редакторе, не пользуясь элементами интерфейса, что мы и сделали в этом примере. Я предпочитаю именно такой метод

работы, поскольку он помогает досконально все понять, а не переключаться с элементов интерфейса на код и обратно.

Вы также можете воспользоваться кнопкой **Manage Parameters** (Управление параметрами) на вкладке **Home** (Главная страница) для создания двоичного параметра, а после этого создать ссылку, щелкнув правой кнопкой мыши по двоичному файлу. На следующем шаге можно создать пользовательскую функцию, нажав правой кнопкой мыши по созданному параметрическому запросу и выбрав пункт **Create Function** (Создать функцию).

Функции, которые вы хотите выполнить с целью преобразования данных, запускаются вручную, после чего можно воспользоваться опцией **Invoke Custom Function** (Вызвать настраиваемую функцию). Так что шаги очень похожи – все зависит от наиболее предпочтительного для вас способа.

В следующем разделе мы создадим столбец с датой и временем при помощи трех функций языка M.

Создание столбца с датой и временем с использованием трех функций языка M

В данном разделе мы закрепим тему написания пользовательских функций на примере создания столбца типа `DateTime` в Power Query с применением функций языка M, таких как `List.DateTimes`, `Duration.TotalDays` и `Time.LocalNow()`.

Полный список функций языка M для работы с полями даты и времени в Power Query можно найти по адресу: <https://docs.microsoft.com/en-us/powerquery-m/date-functions>.

Удобство электронной документации от Microsoft состоит в возможности копирования целых блоков кода и вставки непосредственно в окно расширенного редактора Power Query или в строку формул. Таким образом можно сэкономить немало времени при написании кода на языке M. Просто найдите на сайте фрагмент кода, который хотите использовать в своих задачах, и в правом верхнем углу нажмите ссылку **Copy** (Копировать), как показано на рис. 11.20.

Example

Create a list of 10 values starting from 5 minutes before New Year's Day (#datetime(2011, 12, 31, 23, 55, 0)) incrementing by 1 minute (#duration(0, 0, 1, 0)).

powerquery-m

Copy

```
List.DateTimes(#datetime(2011, 12, 31, 23, 55, 0), 10, #duration(0, 0, 1, 0))
```

12/31/2011 11:55:00 PM

12/31/2011 11:56:00 PM

Рис. 11.20. Копирование кода на языке M с сайта docs.microsoft.com

Перед тем как продолжить изучать важные функции языка M, давайте посмотрим на то, как можно производить вычитание дат в Power Query без запоминания специальных языковых конструкций.

Если вам необходимо определить разницу в днях между двумя датами в Power Query, вы можете воспользоваться специальным пунктом меню для вычитания дат – **Subtract Days** (Вычесть дни). Его можно найти в интерфейсе Power Query на вкладке **Add Column** (Добавление столбца). Выполните следующие действия для подсчета количества дней между двумя датами.

1. В запросе SSGDates у нас содержится список товаров со столбцами **Date Sold** (Дата продажи) и **Date Delivered** (Дата доставки), что видно по рис. 11.21. Нам нужно вычесть одну дату из другой, чтобы получить информацию о длительности доставки в днях. Для начала выделите столбцы, с которыми будете работать. При этом важен порядок выделения столбцов. Если сначала выделить колонку **Date Sold**, а затем **Date Delivered**, на выходе мы получим отрицательные числа, поскольку дата продажи всегда будет предшествовать дате доставки. Если вы хотите, чтобы длительность в днях была положительной, выделите сначала поле **Date Delivered**, а затем **Date Sold**.

Region	Product	Sales	Date Sold	Date Delivered
West	Board5	53395.17732	2019/12/31	2020/02/01
West	Game6	116609.6948	2019/10/31	2020/04/01
South	Game1	72524.0953	2019/12/30	2020/05/01
South	Game5	22538.47873	2019/09/30	2020/06/01
North	Dyno4	45616.53282	2019/12/29	2020/06/01
North	Game5	81902.36089	2019/11/29	2020/03/01

Рис. 11.21. Запрос SSGDates

2. После выделения нужных полей в правильном порядке перейдите на вкладку **Add Column** (Добавление столбца) и в выпадающей кнопке **Date** (Дата) выберите пункт **Subtract Days** (Вычесть дни), как показано на рис. 11.22.

Region	Product	Sales	Date Sold	Date Delivered
West	Board5	53395.17732	2019/12/31	2020/02/01
West	Game6	116609.6948	2019/10/31	2020/04/01
South	Game1	72524.0953	2019/12/30	2020/05/01
South	Game5	22538.47873	2019/09/30	2020/06/01
North	Dyno4	45616.53282	2019/12/29	2020/06/01
North	Game5	81902.36089	2019/11/29	2020/03/01
7 South	Board5	72000.3749	2019/12/31	2020/02/01
North	Game10	99227.6778	2019/10/31	2020/04/01
West	Dyno1	93520.54806	2019/12/30	2020/05/01
South	Game5	77445.82665	2019/11/29	2020/04/04

Рис. 11.22. Вычисление разницы в днях между двумя датами

3. Результат будет выведен в отдельном столбце, представленном на рис. 11.23.

= Table.RenameColumns(#"Inserted Date Subtraction",{{"Subtraction", "Number Days"}})				
A ^B Product	1.2 Sales	Date Sold	Date Delivered	Number Days
Board5	53395.17732	2019/12/31	2020/02/01	32
Game6	116609.6948	2019/10/31	2020/04/01	153
Game1	72524.0953	2019/12/30	2020/05/01	123
Game5	22538.47873	2019/09/30	2020/06/01	245
Dyno4	45616.53282	2019/12/29	2020/06/01	155
Game5	81902.36089	2019/11/29	2020/03/01	93
Board5	72000.3749	2019/10/20	2020/02/08	111
Game10	99227.6778	2019/10/01	2019/12/01	61
Dyno1	93520.54806	2019/11/15	2020/03/23	129
Game5	77445.82665	2020/09/29	2020/04/04	-178

Рис. 11.23. Разница в днях между датой продажи и датой доставки

4. Давайте посмотрим, какой код на языке М был сгенерирован. Для этого перейдите на вкладку **View** (Просмотр) и нажмите на кнопку **Advanced Editor** (Расширенный редактор). Вы увидите код, представленный на рис. 11.24.



```

Advanced Editor

SSGDates
Display Options

let
    Source = Excel.Workbook(File.Contents("C:\Users\For1UK\Documents\SSG Group\SSGSalesD.xlsx"), null, true),
    Sheet1_Sheet = Source{[Item="Sheet1", Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(Sheet1_Sheet, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Region", type text}, {"Product", type text}, {"Sales", type number}, {"Date Sold", type date}, {"Date Delivered", type date}}),
    #"Inserted Date Subtraction" = Table.AddColumn(#"Changed Type", "Subtraction", each Duration.Days([Date Delivered] - [Date Sold]), Int64.Type),
    #"Renamed Columns" = Table.RenameColumns(#"Inserted Date Subtraction",{{"Subtraction", "Number Days"}})
in
    #"Renamed Columns"

```

Рис. 11.24. Код на языке М для вычитания двух дат

Это был первый способ вычесть одну дату из другой в Power Query. Теперь давайте познакомимся с важной функцией языка М под названием *List.DateTimes*. Она позволяет создать список дат или времен на основании вашего ввода.

Для начала также вычислим разницу в днях между двумя датами.

1. Создайте новый запрос с именем *NumbDays*.
2. Перейдите на вкладку **View** (Просмотр) и нажмите на кнопку **Advanced Editor** (Расширенный редактор).
3. Переименуйте в коде *Source* в *NumbDays*, как показано на рис. 11.25. Это будет имя нашей функции, которую мы будем использовать на поздних стадиях в формуле *List.DateTimes*.

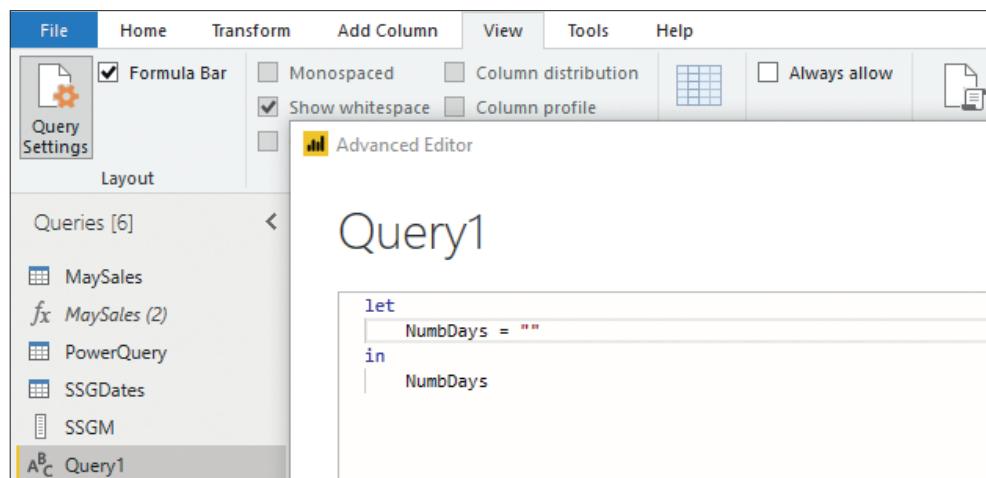


Рис. 11.25. Переименование источника в расширенном редакторе

4. Мы будем использовать следующее выражение для вычитания одной даты из другой:

```
= #datetime(2019,01,01,00,00,00)-
#datetime(2020,06,17,00,00,00)
```

5. Нажмите на кнопку **Done** (Готово), чтобы отобразить разницу между двумя датами, как показано на рис. 11.26.

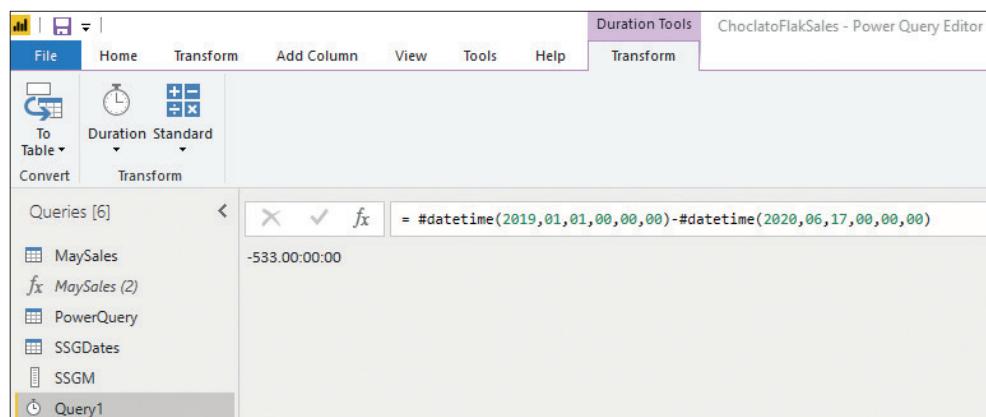


Рис. 11.26. Вычитание одной даты из другой для получения разницы в днях между ними

6. Чтобы число в результате получилось положительным, поменяйте местами элементы выражения – должно получиться следующее выражение:

```
= #datetime(2020,06,17,00,00,00)-
#datetime(2019,01,01,00,00,00)
```

7. Добавьте шаг, нажав на иконку **fx** в строке формул, как показано на рис. 11.27.

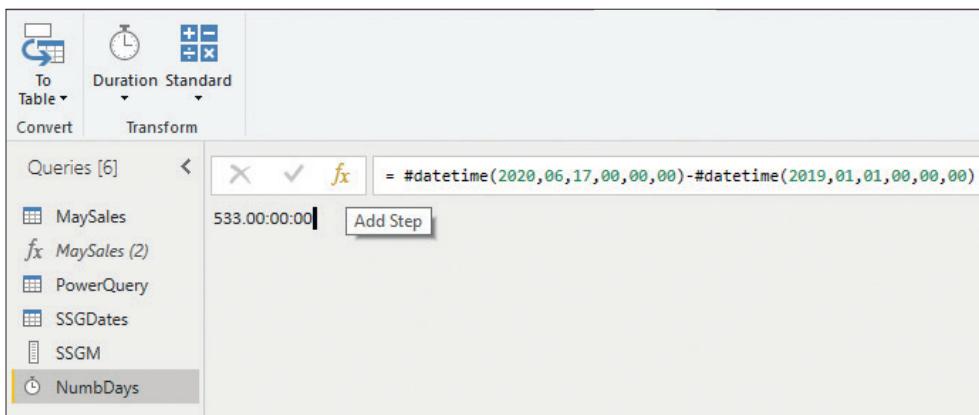


Рис. 11.27. Создание шага

8. При создании шага с кодом на языке M можно вводить текст непосредственно в расширенном редакторе или в строке формул. Введите следующую формулу в строку формул:

```
List.DateTimes(#datetime(2019, 01, 01, 00, 00, 0), 10,
#duration(0, 0, 1, 0))
```

Эта функция генерирует список с десятью значениями с датой и временем начиная с 1 января 2019 года с интервалом в одну минуту, что видно по рис. 11.28.

Sort Numeric List	
fx	= List.DateTimes(#datetime(2019, 01, 01, 00, 00, 0), 10, #duration(0, 0, 1, 0))
	19/01/01 00:00:00
	19/01/01 00:01:00
	19/01/01 00:02:00
	19/01/01 00:03:00
	19/01/01 00:04:00
	19/01/01 00:05:00

Рис. 11.28. Код на языке M с использованием функции List.DateTimes

Подобная инструкция может быть использована применительно к дням, часам, минутам и секундам.

9. На рис. 11.29 показано, как можно изменить код, чтобы интервалы между датами были не минутными, а часовыми.

A screenshot of a spreadsheet application showing a list of 10 dates. The formula bar at the top shows: = List.Datetimes(#datetime(2019, 01, 01, 00, 00, 0), 10, #duration(0, 1, 0, 0)). The table has a header row labeled "List" and contains 10 rows numbered 1 to 10, each showing a date and time from 2019/01/01 00:00:00 to 2019/01/01 09:00:00, with an interval of 1 hour.

	List
1	2019/01/01 00:00:00
2	2019/01/01 01:00:00
3	2019/01/01 02:00:00
4	2019/01/01 03:00:00
5	2019/01/01 04:00:00
6	2019/01/01 05:00:00
7	2019/01/01 06:00:00
8	2019/01/01 07:00:00
9	2019/01/01 08:00:00
10	2019/01/01 09:00:00

Рис. 11.29. Даты с интервалами в час, а не в минуту

10. Теперь увеличим количество дат в списке с 10 до 24, чтобы представить суточный интервал, как на рис. 11.30.

A screenshot of a spreadsheet application showing a list of 24 dates. The formula bar at the top shows: = List.Datetimes(#datetime(2019, 01, 01, 00, 00, 0), 24, #duration(0, 1, 0, 0)). The table has a header row labeled "List" and contains 24 rows numbered 1 to 24, each showing a date and time from 2019/01/01 00:00:00 to 2019/01/01 21:00:00, with an interval of 1 hour.

	List
1	2019/01/01 00:00:00
2	2019/01/01 01:00:00
3	2019/01/01 02:00:00
4	2019/01/01 03:00:00
5	2019/01/01 04:00:00
6	2019/01/01 05:00:00
7	2019/01/01 06:00:00
8	2019/01/01 07:00:00
9	2019/01/01 08:00:00
10	2019/01/01 09:00:00
11	2019/01/01 10:00:00
12	2019/01/01 11:00:00
13	2019/01/01 12:00:00
14	2019/01/01 13:00:00
15	2019/01/01 14:00:00
16	2019/01/01 15:00:00
17	2019/01/01 16:00:00
18	2019/01/01 17:00:00
19	2019/01/01 18:00:00
20	2019/01/01 19:00:00
21	2019/01/01 20:00:00
22	2019/01/01 21:00:00

Рис. 11.30. 24 даты с интервалами в один час

11. Вернемся к шагу **NumbDays**, на котором мы получили разницу в днях между двумя датами. Как видите, в нашем случае между представленными датами разница составляет 533 дня. Мы будем использовать переменную **NumbDays** вместо числа 24 в нашей формуле, чтобы составить список из 533 дней. Замените в тексте формулы 24 на **NumbDays**. Вы увидите ошибку, показанную на рис. 11.31. Причина ее возникновения в том, что переменная **NumbDays** не является числовой, что мы сейчас исправим.

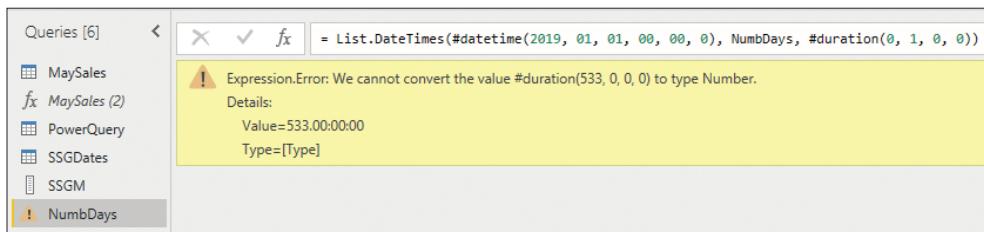


Рис. 11.31. Ошибка из-за несоответствия типов данных в выражении

12. На этом этапе в игру вступает другая функция с названием *Duration.TotalDays*. Вернитесь к шагу **NumbDays** и отредактируйте код следующим образом:

```
=Duration.TotalDays(#datetime(2020,06,17,00,00,00)-
#datetime(2019,01,01,00,00,00))
```

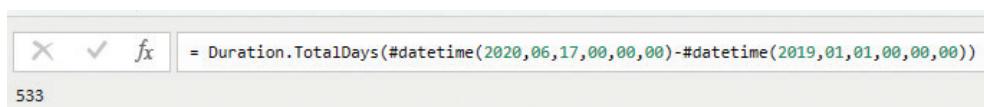


Рис. 11.32. После добавления функции *Duration.TotalDays* результат вычисления стал числовым

13. Если щелкнуть на шаге с именем **Custom1** (Пользовательский1), мы увидим, что обновились 533 записи.
14. Осталось преобразовать значения в таблицу. Для этого нажмите на кнопку **To Table** (В таблицу) в левой части ленты, как показано на рис. 11.33.

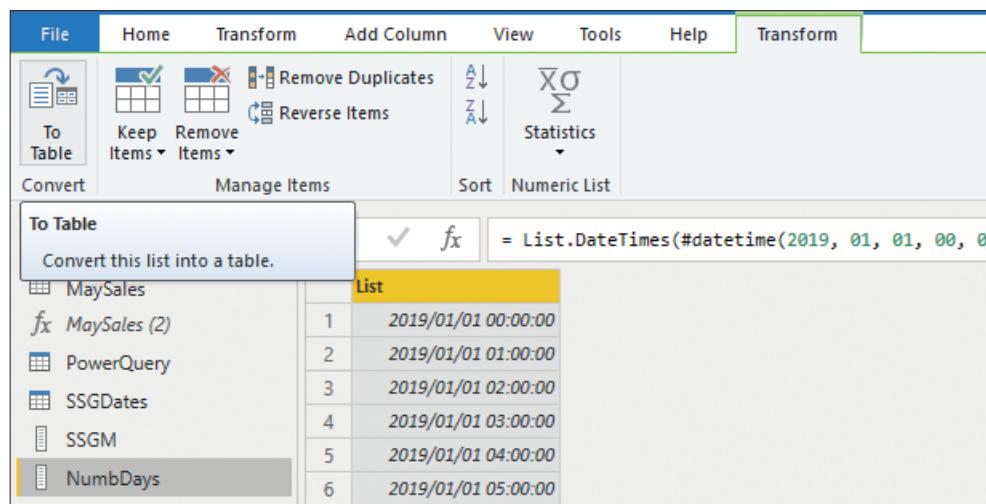


Рис. 11.33. Преобразование значений в таблицу данных

15. Нажмите на кнопку **OK**, чтобы конвертировать значения в таблицу.
16. Отсортируйте полученную таблицу в порядке возрастания, чтобы проверить, что дата 1 января 2019 года включена в список. Теперь смените порядок сортировки на убывающий. Вы обнаружите, что дни заканчиваются раньше нужной даты, а значит, нужно подкорректировать наш код на языке M. Причина этого в том, что у нас нет достаточного количества строк для хранения всех дней в интервале. Таким образом, нам необходимо умножить результат, полученный на первом шаге, на 24, как показано на рис. 11.34, чтобы уместились все часы в требуемом интервале.

The screenshot shows the Power Query Editor interface. The formula bar contains the expression: `= Duration.TotalDays(#datetime(2020,06,17,00,00,00)-#datetime(2019,01,01,00,00,00))*24`. The results table below shows a single row with the value 12792. The right pane displays 'Query Settings' and 'APPLIED STEPS' sections.

Рис. 11.34. Умножение результата первого шага на 24

17. Если вернуться к шагу **Sorted Rows** (Сортированные строки), мы увидим, что календарь по-прежнему заполнен некорректно. Давайте добавим в наш код ссылку на текущую системную дату и время (`DateTime.LocalNow`), как показано на рис. 11.35.



Рис. 11.35. Добавление ссылки на текущую дату в функцию

18. Перейдите к шагу сортировки, и вы увидите, что результат изменился на ожидаемый, как видно по рис. 11.36.

	ABC 123 Column1
1	2020/06/17 20:00:00
2	2020/06/17 19:00:00
3	2020/06/17 18:00:00
4	2020/06/17 17:00:00
5	2020/06/17 16:00:00
6	2020/06/17 15:00:00
7	2020/06/17 14:00:00

Рис. 11.36. Корректный вывод интервала до текущей даты

В этом разделе мы научились работать с датами в Power Query посредством функций `List.DateTimes`, `Duration.TotalDays` и `Time.LocalNow`. Вы можете самостоятельно пройтись по остальным функциям для работы с датой и временем в языке M, открыв ссылку, приведенную в начале раздела. Надеемся, что эти три важные функции подогреют ваш интерес к изучению темы работы с датами и вы сможете использовать их вместе с другими полезными функциями в своих сценариях.

ЗАКЛЮЧЕНИЕ

В данной главе вы освоили работу с путями к файлам и папкам и научились создавать собственные параметрические запросы посредством расширенного редактора Power Query.

Кроме того, вы узнали, как можно тестировать параметрические запросы или функции и вызывать их при создании нового столбца. Также вы научились использовать некоторые полезные функции Power Query для работы с датой и временем, такие как `List.DateTimes`, `Duration.TotalDays` и `Time.LocalNow`, для создания пользовательских столбцов.

В заключительной главе книги мы обсудим различия между языками DAX и M. Вы узнаете о сходных и различающихся особенностях этих языков на примерах, включающих создание вычисляемых мер.

Глава 12

Различия между DAX и M

Язык M считается полноценным функциональным языком среди Power Query. Его официальным названием является Mashup или Power Query Formula Language (Язык формул Power Query), и он используется для осуществления запросов к различным источникам и преобразования полученных данных. В то же время **Data Analysis eXpressions (DAX)** представляет собой набор функций для работы с данными, хранящимися в виде таблиц, подобно Excel.

В настоящей главе мы обсудим сходства и различия между этими двумя языками, исходя из характерных для них особенностей. Мы познакомимся с синтаксисом языка DAX, чтобы понять, как в нем работают формулы, а также узнаем, как использовать формулы на DAX применительно к Excel. Прочитав главу, вы научитесь создавать вычисляемые столбцы с применением формул и писать меры с нуля для расчета агрегатов.

Основные темы этой главы:

- особенности и функционал языков DAX и M;
- синтаксис языка DAX;
- создание вычисляемых столбцов;
- создание вычисляемых мер.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Вам понадобится интернет-соединение для загрузки соответствующих файлов с GitHub. Файлы с кодом для этой главы можно скачать по адресу: <https://github.com/PacktPublishing/Learn-Power-Query>.

В этой главе мы исходим из того, что вы внимательно прочитали главы 9, 10 и 11. В этих главах мы привели важные примеры, описывающие особенности языка M.

Видеофрагмент, посвященный данной главе, располагается по адресу: https://www.youtube.com/watch?v=up4CCRsNj7I&list=PLeLcvrwLe186O_GJEZs47WaZXwZjwTN83&index=13&t=0s.

ОСОБЕННОСТИ И ФУНКЦИОНАЛ ЯЗЫКОВ DAX И M

В данном разделе мы поговорим об областях применения языков DAX и M, а также обсудим их функционал.

Язык DAX главным образом используется при выполнении преобразований данных в дашбордах Power BI, помогая извлекать важную аналитическую информацию. DAX незаменим при анализе продаж с процентами прироста по товарам в рамках выбранного периода или выявлении рыночных тенденций. В рамках Power BI DAX позволяет создавать богатые интерактивные визуализации, помогающие принимать бизнес-решения. DAX – это язык формул, он не считается языком программирования в полном смысле слова, поскольку строится на основе пользовательских вычислений. Будучи языком формул, DAX строится на следующих структурных единицах:

- операторы;
- константы;
- функции.

Что касается языка M, он применяется в расширенном редакторе Power Query. По сути, каждый раз, когда вы выполняете то или иное преобразование данных, автоматически создается соответствующая инструкция на языке M. Чтобы отредактировать код на M внутри Power Query, необходимо открыть расширенный редактор или войти в строку формул, о чём мы неоднократно упоминали в предыдущих главах этой книги.

В следующей таблице мы акцентировали внимание на основных отличиях между языками M и DAX.

Таблица 12.1. Основные отличия между языками M и DAX

Свойство	M	DAX
Язык используется в...	Power Query, Power BI, Analysis Services и Excel	SQL Server Analysis Services Tabular, Power BI и Power Pivot (Excel)
Официальное название...	Mashup of Power Query formula language	Data analysis eXpression language
Разработчик...	Microsoft	Microsoft
Для чего применяется...	Подключение к источникам данных Преобразование данных	Выбор, связка, объединение и динамическая фильтрация данных, а также создание вычисляемых столбцов, мер и таблиц. Не использует блоки программирования
Как работает...	Мощная пошаговая, функциональная, регистрозависимая языковая структура	Посредством выражений, подобно Excel. DAX – язык формул, а не язык программирования
Преимущества...	Легкий в использовании с последовательными шагами преобразований. В основе лежат выражения let и in	Очень мощный язык со схожими с Excel функциями. Простой в освоении для пользователей Excel. Используется для анализа бизнес-запросов
IntelliSense	M IntelliSense	Есть система рекомендаций
Ресурсы	https://docs.microsoft.com/en-us/powerquery-m	https://docs.microsoft.com/en-us/power-bi-transform-model/desktop-quickstart-learn-dax-basics
Список формул	Встроенная библиотека #shared с поддержкой https://docs.microsoft.com/en-us/powerquery-m-function-reference	https://docs.microsoft.com/en-us/dax/dax-function-reference

Теперь, когда вы узнали об основах функционирования языков DAX и M, давайте поговорим немного подробнее о строении первого из них.

Синтаксис языка DAX

Каждый язык, с которым вы сталкиваетесь, обладает своей структурой и синтаксисом. Под синтаксисом подразумевается набор элементов, допустимых к использованию при построении формул. Формулы зачастую состоят из вызовов функций. В главе 9 вы познакомились с синтаксисом языка M. В этой главе вы узнаете, что из себя представляет синтаксис DAX в отношении именования таблиц и столбцов.

Если вы новичок в языке DAX, то вам будет полезно узнать, что структурно этот язык состоит из двух частей, как видно из табл. 12.2.

Таблица 12.2. Две части DAX

DAX	Инструмент
Запросы DAX	Analysis Services, DAX Studio и SQL Management Studio
Формулы DAX	Power BI, Excel и Power Pivot

Очень важно понимать, чем отличаются друг от друга эти составные части DAX, чтобы правильно выбирать инструмент для обработки того или иного кода. Зачастую разработчики ищут фрагменты кода в официальных справочниках, копируют их и адаптируют под свои сценарии. С DAX в этом отношении можно попасть в ловушку, поскольку корректный код может не работать, если будет неправильно выбран инструмент.

Функции DAX и Excel во многом схожи в плане поведения и семантики – к примеру, функции массивов или поиска. Основные отличия кроются в синтаксисе. В Excel используются ссылки на ячейки и диапазоны, тогда как DAX оперирует в формулах целыми столбцами и таблицами. Также стоит отметить недопустимость смешивания функций Excel и DAX в одной формуле. DAX всегда требует наличия связей между таблицами при выполнении операций поиска.

Давайте взглянем на элементы синтаксиса языка DAX. На рис. 12.1 показана простейшая формула на языке DAX в Power BI.

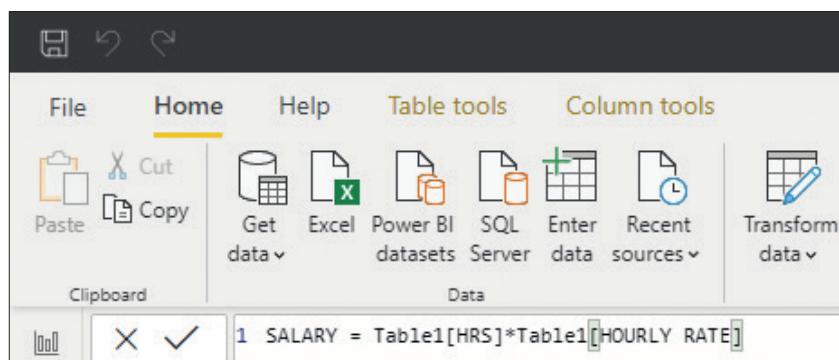


Рис. 12.1. Вычисления на языке DAX в Power BI Desktop

Заметим, что не все конструкции формул будут одинаковыми – некоторые формулы могут не содержать определенных элементов, например функций.

В табл. 12.3 описаны все составляющие синтаксиса языка DAX.

Таблица 12.3. Разбор синтаксиса DAX

Параметр	Описание
Имя	В синтаксисе языка DAX вы всегда именуете создаваемые элементы. Например, в случае создания вычисляемого столбца это будет его имя (заголовок)
=	Знак равенства (=) символизирует начало формулы, подобно Excel
Функции	Следом за любой функцией должны идти скобки – открывающие и закрывающие, с перечислением параметров между ними. Например, SUM(Table1[HRS])
Скобки	Квадратные скобки употребляются в выражениях, содержащих более одного аргумента
Ссылочные таблицы	В выражении Table1[HRS] таблицей, на которую мы ссылаемся, является Table1
Ссылочные столбцы	В выражении Table1[HRS] столбцом внутри Table1, на который мы ссылаемся, является HRS
Выражение	Некоторые функции, такие как CALCULATE, требуют указания выражения. Выражения могут быть мерами. Скоро вы узнаете, что такое меры

Примечание

В случае ввода некорректного синтаксиса в столбце будет выведена синтаксическая ошибка.

В предыдущей таблице мы познакомились с элементами синтаксиса языка DAX и концепцией построения формул. Посмотрим, где все это можно использовать.

Использование формул DAX в Excel

Чтобы применять формулы DAX в Excel, необходимо использовать надстройку **Power Pivot**. Мы будем отталкиваться от того, что вам уже известно, как добавить данные из Excel в Power Pivot, поскольку мы говорили об этом в главе 2.

К формулам DAX в Power Pivot можно прийти двумя путями:

- нажмите на строку формул в Power Pivot и начинайте вводить формулы, как показано на рис. 12.2;
- нажмите на кнопку **fx** (Вставить функцию) на вкладке **Design** (Конструктор) для выбора функции, которую хотите использовать в формуле.

Итак, мы узнали, где в Excel можно использовать формулы, написанные на языке DAX. В следующем разделе посмотрим, какой помощью можно воспользоваться при написании кода.

The screenshot shows the Microsoft Excel ribbon with the 'Design' tab selected. In the formula bar, the text '=Table1[HRS]*Table1[HOURLY RATE]' is entered. Below the formula bar is a table with several columns: CODE, FIRST, SURNAME, EMP NO, DIVISION, DEPT, DATE OF HIRE, HRS, HOURLY RATE, and SALARY. The 'FIRST' column is highlighted in green, indicating it is the current selection.

Рис. 12.2. Использование формул DAX в Power Pivot

Использование IntelliSense

Технология *IntelliSense* позволяет разработчику получить полный список допустимых функций и их параметров при написании кода на DAX. Вы наверняка уже сталкивались с подобным помощником при вводе формул в Excel. В DAX это работает приблизительно похожим образом. Например, при вводе открывающего парного элемента – кавычки или любого вида скобки – *IntelliSense* автоматически вставляет соответствующий закрывающий элемент.

IntelliSense в любой ситуации предлагает полный набор доступных к использованию элементов языка в виде выпадающего списка. Работая в расширенном редакторе Power Query, слева от элемента вы увидите специальную иконку, представляющую собой куб, треугольник, квадрат или круг, – и у каждой свой смысл. Куб символизирует родные функции языка M, а треугольник, квадрат и круг применяются для описания переменных, шагов, параметров, констант, имен запросов и т. д.

IntelliSense также подсвечивает ключевые слова и предлагает помочь при указании параметров функций.

The screenshot shows the Microsoft Excel ribbon with the 'Design' tab selected. In the formula bar, the text '=t' is entered. Below the formula bar is a table with columns: CODE, FIRST, SURNAME, DEPT, DATE OF HIRE, HRS, HOURLY RATE, and SALARY. The 'SURNAME' column is highlighted in green. A dropdown menu is open over the 'SURNAME' column, listing various table columns such as Table1[CODE], Table1[DATE OF HIRE], etc.

Рис. 12.3. DAX IntelliSense в действии в строке формул Power Pivot

На рис. 12.4 представлен пример использования IntelliSense в Power Query при написании кода на языке M.

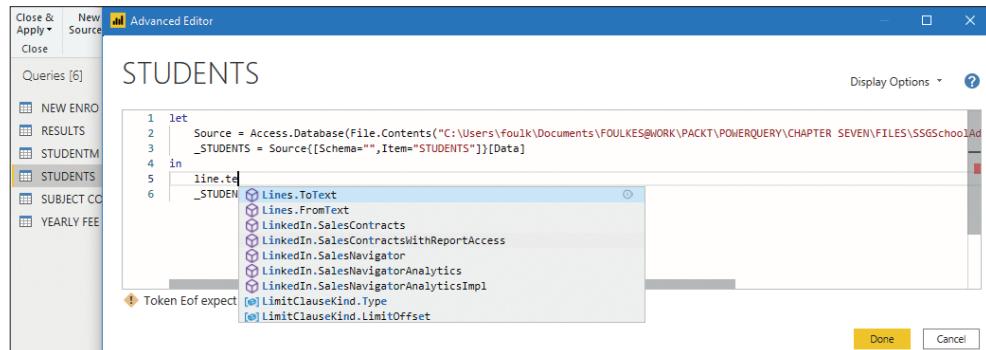


Рис. 12.4. Использование IntelliSense в расширенном редакторе Power Query

Теперь, когда вы в общих чертах понимаете синтаксис языка DAX и назначение технологии IntelliSense, давайте поближе присмотримся к формулам DAX.

Написание формул DAX

В Power BI вы можете в полной мере оперировать формулами на языке DAX. На вкладке **Home** (Главная страница) располагается группа **Calculations** (Вычисления), включающая кнопки **New measure** (Создать меру), **Quick measure** (Быстрая мера), **New column** (Создать столбец) и **New table** (Создать таблицу), как показано на рис. 12.5.

Salesperson	Product Name	Units	Price/Unit	Sales	Due Date
Willousty	Cholato Flak	45	11.6	522	Thursday, 02 May 2019
Willousty	Fudgi Browno	56	9.99	559.44	Thursday, 02 May 2019
Willousty	Straubory	123	10	1230	Thursday, 02 May 2019
Willousty	Plain Whito	47	9.99	469.53	Thursday, 02 May 2019

Рис. 12.5. Группа вычислений в ленте Power BI Desktop

Ниже мы приведем основные типы формул на языке DAX. Ко всем из них можно получить доступ через указанные выше кнопки из группы вычислений:

- **вычисляемые столбцы:** используются для выполнения построчных вычислений путем создания нового столбца в таблице;
- **быстрые меры:** позволяют воспользоваться заранее определенными шаблонами вычислений, которые можно создать без знания синтаксиса языка DAX;

- **вычисляемые меры:** меры не занимают физическую память, а вычисляются на лету исходя из контекста. Такие вычисления всегда требуют определенного типа агрегации;
- **вычисляемые таблицы:** служат для создания новых таблиц с данными. Совсем скоро мы познакомимся с этими типами вычислений поближе.

ВВЕДЕНИЕ В ДВИЖОК ФОРМУЛ И ДВИЖОК ХРАНИЛИЩА DAX

За обработку выражений на языке DAX отвечают два движка: движок формул и движок хранилища. В табл. 12.4 приведены зоны ответственности каждого из них, а также их преимущества и недостатки.

Таблица 12.4. Движки DAX

Тип движка	Зона ответственности	Преимущества и недостатки
Движок формул (FE)	<ul style="list-style-type: none"> • читает запросы; • интерпретирует запросы; • запрашивает данные у движка хранилища; • выполняет запросы 	<ul style="list-style-type: none"> • однопоточный, то есть способен выполнять по одной операции за раз. Как следствие достаточно медленный; • позволяет обнаружить ошибки в запросе
Движок хранилища (SE)	<ul style="list-style-type: none"> • сканирует и анализирует данные, а также обеспечивает быстрый доступ к данным и управление простыми запросами (с применением функций SUM, COUNT и т. д.); • управляет механизмом VertiPaq, лежащим в основе Power Query и Power BI 	<ul style="list-style-type: none"> • многопоточный, вычисления выполняются очень быстро. Может выполнять более одной операции за раз; • не способен выполнять сложные запросы. За их выполнением обращается к движку формул; • сталкивается с трудностями при обработке огромных наборов данных при необходимости выполнять построчные операции (вычисляемые столбцы). В этом случае отправляет запросы обратно движку формул для их выполнения; • очень быстро выполняет расчет мер в отфильтрованной таблице. Но для этого необходимо правильно сформулировать запрос DAX. Пример: использование мер вместо вычисляемых столбцов

Теперь вы знаете назначение, а также плюсы и минусы движка формул и движка хранилища DAX. Перейдем к более конкретным примерам формул и запросов.

СОЗДАНИЕ ВЫЧИСЛЯЕМОГО СТОЛБЦА

В данном разделе мы создадим *вычисляемый столбец* (calculated column) в Power BI Desktop, который будет служить для выполнения среза или фильтрации значений или вычислений в каждой строке таблицы. Но что из себя представляет вычисляемый столбец?

Вычисляемый столбец – это колонка, присоединенная к существующей таблице, значения которой вычислены движком DAX по указанной формуле, подобно автозаполнению, примененному к формулам в Excel. Всегда помните, что значения из вычисляемого столбца хранятся в физической памяти

компьютера, за исключением периодов их перезагрузки в моменты включения и выключения Excel / Power BI. В этих случаях таблицы обновляются, что приводит к перерасчету вычисляемых столбцов. Давайте посмотрим, как это работает.

1. Откройте Power BI. Для нашего примера мы будем использовать файл `SSGThemePark.pbix`.
2. Формулы DAX допустимо создавать, находясь в любой вкладке, но я предпочитаю делать это на вкладке **Data** (Данные), чтобы параллельно видеть, как структурированы данные в столбцах. Очень удобно бывает видеть заголовки столбцов при работе с формулами. Но, в принципе, выбор вкладки, в которой писать формулы DAX, остается за вами.
3. Нажмите на кнопку **New column** (Создать столбец) в группе **Calculations** (Вычисления) на вкладке **Home** (Главная страница) или **Table tools** (Средства работы с таблицами). Справа в открытой таблице появится новая колонка с именем `Column` (Столбец). Также будет доступна строка формул, ожидающая ввода, как показано на рис. 12.6.

CODE	FIRST	SURNAME	EMP NO	DIVISION	DEPT	DATE of HIRE	HRS	HOURLY RATE	Column
69	Shirley	Dandrow	TBV45	View Tabue	Cobrella	33311	35	12.1	
12	Seth	Rose	TBV76	View Tabue	Cobrella	32968	32	5.5	
47	Anne	Davidson	TBV23	View Tabue	Cobrella	31508	25	8.52	

Рис. 12.6. Добавление вычисляемого столбца в таблицу

4. Введите формулу на языке DAX в строку формул, находящуюся непосредственно над представлением таблицы. Мы используем для примера простое вычисление заработной платы каждого сотрудника, базируясь на количестве отработанных часов и почасовой ставке.
5. Для начала поменяйте в строке формул текст `Column` на `SALARY`, чтобы заодно дать столбцу более осмысленное имя. После знака равенства начните вводить имя существующего в таблице столбца, ссылку на который вы хотите использовать в формуле. В нашем случае это столбец `HRS`, в котором хранится количество отработанных сотрудником часов. Заметьте, что после начала ввода анализатор предлагает на выбор элементы, подходящие под ваш ввод. При этом в выпадающем списке слева от каждого элемента будет соответствующая иконка. В списке, представленном на рис. 12.7, мы видим иконки функций и таблиц. Дважды щелкните на элементе `Table1[HRS]` из списка, чтобы добавить его в формулу.

The screenshot shows the Power BI Data Editor interface. On the left, there's a table of employee data with columns: CODE, FIRST, SURNAME, and three additional columns. On the right, there's a table with columns: ON, DEPT, DATE of HIRE, HRS, HOURLY RATE, and a 'Column' header. A context menu is open over the formula '1 SALARY = h', listing options like HASONEFILTER, HASONEVALUE, HOUR, and Table[Hourly Rate]. The 'Table[Hourly Rate]' option is highlighted.

Рис. 12.7. Выпадающий список с предлагаемыми на выбор элементами для вставки в формулу

6. Теперь поставьте оператор умножения (*), после чего введите название второй колонки (HOURLY RATE), чтобы завершить создание формулы.
7. После нажатия на клавишу **Enter** содержимое столбца заполнится вычисленными значениями.

Теперь вы знаете, как создавать вычисляемые столбцы при помощи формул DAX. Посмотрим, как можно создавать меры.

СОЗДАНИЕ ВЫЧИСЛЯЕМЫХ МЕР

В данном разделе вы познакомитесь с понятием *вычисляемых мер* и научитесь создавать их путем добавления функций или выражений. Давайте перечислим факты, которые необходимо знать для оперирования вычисляемыми мерами.

Все без исключения меры должны включать в себя функцию. Мера не может работать без применения к столбцу, на который она ссылается, определенных функций (`sum`, `min`, `count` и т. д.). Эти функции называются *агрегаторами* (aggregator), и без их применения столбец будет именоваться *голым* (naked column). При написании мер очень полезно бывает пользоваться технологией дополнения формул IntelliSense.

Прелесть мер состоит в том, что они вычисляются только во время осуществления доступа к ним, а значит, не используют драгоценную память. Кроме того, меры позволяют осуществить разный вывод в зависимости от примененного к ним критерия фильтра. Меры создаются в Power BI на вкладке **Report** (Отчет) или **Data** (Данные).

Вычисляемые меры могут быть применены к разным таблицам и запросам. Они адаптируются подобно формулам в Excel при обращении к ячейкам. Таким образом, мы можем свободно перемещать меры без потери их функциональности. При создании вычисляемого столбца данные физически хранятся в движке xVelocity, что приводит к расходованию памяти. В то же время при задействовании посредством отчетов виртуальной памяти требуется гораздо меньше. Меры загружаются куда быстрее, поскольку вовсе не сохраняются в памяти.

Перед созданием полноценной вычисляемой меры с нуля давайте обратимся к упрощенной процедуре создания быстрой меры с выбором заранее подготовленных шаблонов.

Использование быстрых мер

Можно использовать меры с применением заведомо подготовленных шаблонов, а можно писать свои с использованием формул на языке DAX. Если вы новичок в мире DAX, начать лучше будет с **быстрых мер** (quick measure), при создании которых вам будет предложен список распространенных видов вычислений, которые вы можете применить к набору данных. Создание быстрых мер поможет познакомиться с формулами DAX поближе и освоить структуру языка.

Чтобы добавить быструю меру, выполните следующие шаги.

1. Откройте набор данных, в котором хотите создать меру. Для этого примера мы будем использовать файл SalesData.pbix.
2. Перейдите на вкладку **Home** (Главная страница) и нажмите на кнопку **Quick measure** (Быстрая мера), как показано на рис. 12.8. Также можно щелкнуть правой кнопкой мыши по существующей таблице и выбрать в контекстном меню пункт **Quick measure** (Новая быстрая мера).

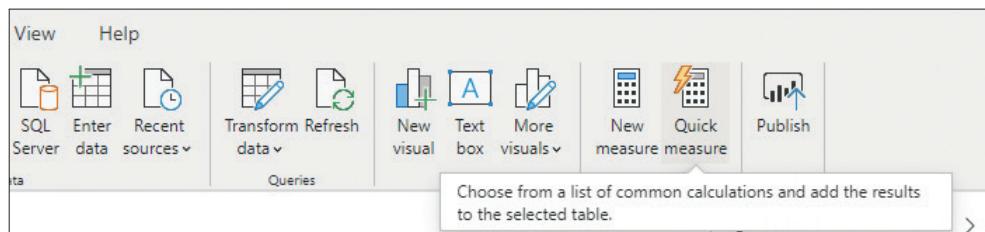


Рис. 12.8. Создание быстрой меры

3. Откроется диалоговое окно **Quick measures** (Быстрые меры), в котором вы можете выбрать из выпадающего списка **Calculation** (Вычисление) нужный тип вычисления.
4. Для примера выберите вычисление **Year-to-date total** (Итого с начала года).
5. Перенесите в вычисление поля из правого окна, к которым хотите применить операцию. В нашем примере перенесем поле **Sum of TOTAL SALES** из списка **Fields** (Поля) в **Base value** (Базовое значение), как показано на рис. 12.9.

Year	Region	Season	Year	Quarter	Month	Day	TOTAL SALES
2018	East	Autumn	2020	Qtr 1	February	13	5734
2018	East	Autumn	2020	Qtr 2	April	24	7024
2018	East	Autumn	2020	Qtr 2	June	2	7378
2018	East	Autumn	2020	Qtr 2	June	8	6319
2018	East	Spring	2020	Qtr 1	February	21	6399
2018	East	Spring	2020	Qtr 1	February	27	5728
2018	East	Spring	2020	Qtr 1	February	28	6162
2018	East	Spring	2020	Qtr 2	May	11	5144
2018	East	Summer	2020	Qtr 1	January	23	6076
2018	East	Summer	2020	Qtr 1	February	7	5327
2018	East	Summer	2020	Qtr 2	May	7	6868
2018	East	Summer	2020	Qtr 2	May	26	6254
2018	East	Winter	2020	Qtr 1	January	14	6002

Рис. 12.9. Вычисление быстрой меры

6. Обратите внимание, что вы можете выбрать тип агрегации значения из выпадающего списка, как видно на рис. 12.10.

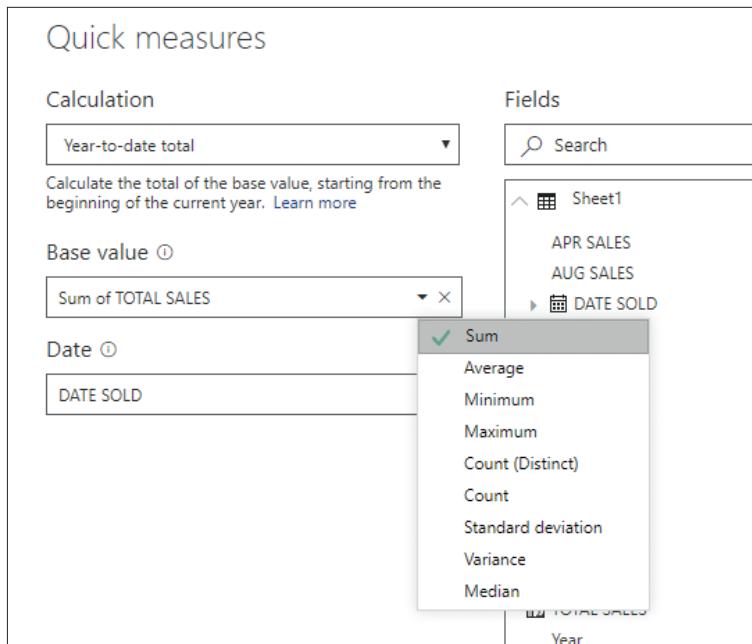


Рис. 12.10. Выбор метода агрегации для поля

7. Перетащите поле **DATE SOLD** из списка справа в поле **Date** (Дата).
8. Нажмите на кнопку **OK**, чтобы Power BI сгенерировал соответствующую формулу на языке DAX.
9. В результате новая мера будет добавлена на панель **Fields** (Поля), как показано на рис. 12.11.

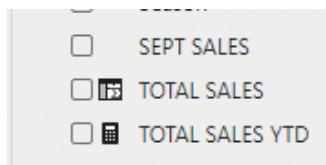


Рис. 12.11. Выбор метода агрегации для поля

10. Перетащите созданную меру **TOTAL SALES YTD** на дашборд с выведенной таблицей. Элемент визуализации обновится, и на нем появится новый столбец с мерой, как на рис. 12.12.

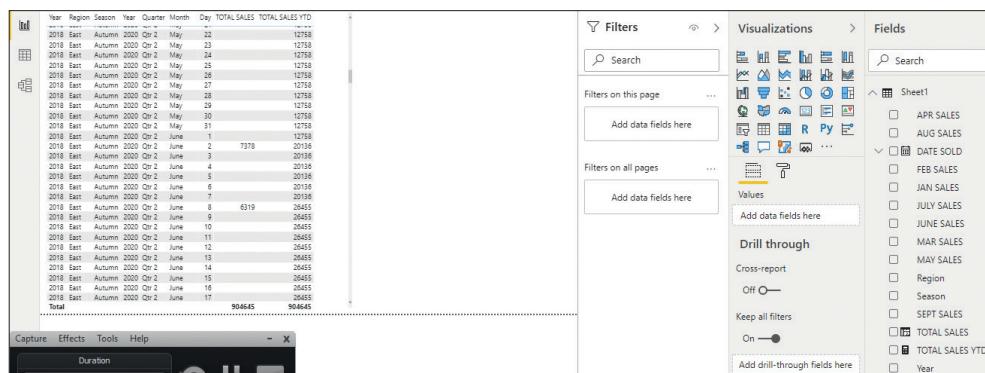


Рис. 12.12. Обновленный дашборд с выводом меры

11. Нажмите на строке формул, чтобы посмотреть, какой код на языке DAX был сгенерирован автоматически. Формула для нашего примера представлена на рис. 12.13.

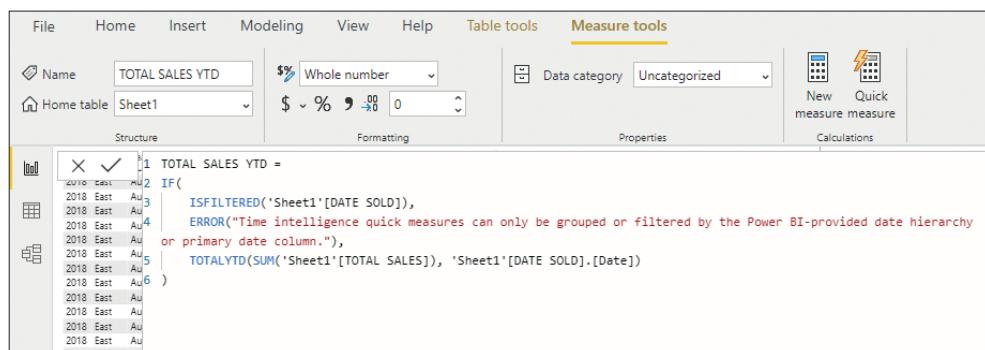


Рис. 12.13. Формула на языке DAX для меры TOTAL SALES YTD

Теперь вы готовы к созданию меры с нуля, чем мы и займемся в следующем разделе.

Создание меры на языке DAX с нуля

В данном разделе мы продолжим работать с файлом из предыдущего примера и узнаем, как и для чего применяются функции CALCULATE и FILTER в языке DAX.

Поскольку мы будем рассчитывать сумму продаж по кварталам, нам понадобится функция Filter для разбивки квартала по категориям. После этого формула подсчитает сумму продаж согласно категориям. Причина использования этой меры кроется в том, что в квартале может появляться новая информация о зарплатах в любой момент, что приведет к увеличению количества строк.

Итак, создадим меру для добавления на наш дашборд.

1. Используя файл SalesData.pbix из предыдущего примера, мы добавим поле TOTAL SALES на дашборд рядом с нашей таблицей при помощи элемента визуализации Card (Карточка), как показано на рис. 12.14.

The screenshot shows a Power BI dashboard. On the left is a table view of sales data. In the center is a large card visual displaying the value "904645" and the text "TOTAL SALES". To the right of the card are two panes: "Filters" and "Fields". The "Filters" pane contains sections for "Filters on this page" and "Filters on all pages", both with a "Add data fields here" button. The "Fields" pane lists various measures like APR SALES, AUG SALES, etc., with "TOTAL SALES" and "TOTAL SALES YTD" selected.

Year	Region	Season	Year	Quarter	Month	Day	TOTAL SALES	TOTAL SALES YTD
2018	East	Autumn	2020	Qtr 1	February	13	5734	5734
2018	East	Autumn	2020	Qtr 1	February	14	5734	5734
2018	East	Autumn	2020	Qtr 1	February	15	5734	5734
2018	East	Autumn	2020	Qtr 1	February	16	5734	5734
2018	East	Autumn	2020	Qtr 1	February	17	5734	5734
2018	East	Autumn	2020	Qtr 1	February	18	5734	5734
2018	East	Autumn	2020	Qtr 1	February	19	5734	5734
2018	East	Autumn	2020	Qtr 1	February	20	5734	5734
2018	East	Autumn	2020	Qtr 1	February	21	5734	5734
2018	East	Autumn	2020	Qtr 1	February	22	5734	5734
2018	East	Autumn	2020	Qtr 1	February	23	5734	5734
2018	East	Autumn	2020	Qtr 1	February	24	5734	5734
2018	East	Autumn	2020	Qtr 1	February	25	5734	5734
2018	East	Autumn	2020	Qtr 1	February	26	5734	5734
2018	East	Autumn	2020	Qtr 1	February	27	5734	5734
2018	East	Autumn	2020	Qtr 1	February	28	5734	5734
2018	East	Autumn	2020	Qtr 1	February	29	5734	5734
2018	East	Autumn	2020	Qtr 1	March	1	5734	5734
2018	East	Autumn	2020	Qtr 1	March	2	5734	5734
2018	East	Autumn	2020	Qtr 1	March	3	5734	5734
2018	East	Autumn	2020	Qtr 1	March	4	5734	5734
2018	East	Autumn	2020	Qtr 1	March	5	5734	5734
2018	East	Autumn	2020	Qtr 1	March	6	5734	5734
2018	East	Autumn	2020	Qtr 1	March	7	5734	5734
2018	East	Autumn	2020	Qtr 1	March	8	5734	5734
2018	East	Autumn	2020	Qtr 1	March	9	5734	5734
2018	East	Autumn	2020	Qtr 1	March	10	5734	5734
Total							904645	904645

Рис. 12.14. Поле TOTAL SALES добавлено на дашборд посредством карточки

- Измените размеры карточки так, чтобы она не занимала весь экран.
- Перенесите поля **Year**, **Region** и **TOTAL SALES** с панели **Fields** (Поля) на свободное пространство дашборда для создания фильтра, как показано на рис. 12.15. Мы это делаем, чтобы продемонстрировать вам, что фильтры можно создавать и без знания формул DAX.

The screenshot shows a Power BI dashboard. On the left is a table view of sales data. In the center is a large card visual displaying the value "904645" and the text "TOTAL SALES". To the right are two panes: "Filters" and "Fields". The "Filters" pane contains sections for "Filters on this page" and "Filters on all pages", both with a "Add data fields here" button. The "Fields" pane lists various measures like APR SALES, AUG SALES, etc., with "TOTAL SALES" selected.

Year	Region	Season	Year	Quarter	Month	Day	TOTAL SALES	TOTAL SALES YTD
2018	East	Autumn	2020	Qtr 1	February	13	5734	5734
2018	East	Autumn	2020	Qtr 1	February	14	5734	5734
2018	East	Autumn	2020	Qtr 1	February	15	5734	5734
2018	East	Autumn	2020	Qtr 1	February	16	5734	5734
2018	East	Autumn	2020	Qtr 1	February	17	5734	5734
2018	East	Autumn	2020	Qtr 1	February	18	5734	5734
2018	East	Autumn	2020	Qtr 1	February	19	5734	5734
2018	East	Autumn	2020	Qtr 1	February	20	5734	5734
2018	East	Autumn	2020	Qtr 1	February	21	5734	5734
2018	East	Autumn	2020	Qtr 1	February	22	5734	5734
2018	East	Autumn	2020	Qtr 1	February	23	5734	5734
2018	East	Autumn	2020	Qtr 1	February	24	5734	5734
2018	East	Autumn	2020	Qtr 1	February	25	5734	5734
2018	East	Autumn	2020	Qtr 1	February	26	5734	5734
2018	East	Autumn	2020	Qtr 1	February	27	5734	5734
2018	East	Autumn	2020	Qtr 1	February	28	5734	5734
2018	East	Autumn	2020	Qtr 1	February	29	5734	5734
2018	East	Autumn	2020	Qtr 1	March	1	5734	5734
2018	East	Autumn	2020	Qtr 1	March	2	5734	5734
2018	East	Autumn	2020	Qtr 1	March	3	5734	5734
2018	East	Autumn	2020	Qtr 1	March	4	5734	5734
2018	East	Autumn	2020	Qtr 1	March	5	5734	5734
2018	East	Autumn	2020	Qtr 1	March	6	5734	5734
2018	East	Autumn	2020	Qtr 1	March	7	5734	5734
2018	East	Autumn	2020	Qtr 1	March	8	5734	5734
2018	East	Autumn	2020	Qtr 1	March	9	5734	5734
2018	East	Autumn	2020	Qtr 1	March	10	5734	5734
Total							904645	904645

Рис. 12.15. Поля **Year**, **Region** и **TOTAL SALES**, добавленные на дашборд

- Теперь посмотрим, как вычислить сумму продаж по сезонам. Убедитесь, что вы работаете на вкладке **Report** (Отчет).
- Перенесите поля **Season** и **TOTAL SALES** на дашборд. Это автоматически приведет к созданию фильтра по полю **Season**, что видно по рис. 12.16. Хотя это куда проще, чем создавать меры, всегда лучше в подобных ситуациях отдавать предпочтение именно мерам, поскольку вам может понадобиться использовать меры в качестве выражений при написании формул на DAX.

Season	TOTAL SALES
Autumn	231288
Spring	224917
Summer	212421
Winter	236019
Total	904645

Рис. 12.16. Поля Season и TOTAL SALES на дашборде

Давайте сделаем то же самое, но при помощи меры.

6. Перед созданием меры выделите таблицу, в которой хотите ее сохранить. Иначе будет выбрана первая по порядку таблица. Помните, что принадлежность меры к таблице можно изменить позже без потери функциональности, но всегда лучше сразу выбрать нужную таблицу.
7. Нажмите на ленте в Power BI кнопку **New measure** (Создать меру). После этого напишите код с рис. 12.17 в строке формул, но не забудьте закрыть скобку, чтобы рассчитать сумму продаж по каждому сезону с использованием фильтра и вычисления.

Structure		Formatting
<input type="button" value="X"/>	<input checked="" type="button" value="✓"/>	1 TOTAL SALES SUM = SUM(Sheet1[TOTAL SALES])
<input type="button" value="grid"/>		
<input type="button" value="list"/>		
<input type="button" value="matrix"/>		
		Y8 21

Рис. 12.17. Формула TOTAL SALES SUM

Заметьте, что если бы в вашей формуле не присутствовала агрегирующая функция, вы бы не смогли создать меру – IntelliSense будет внимательно наблюдать за вами и не предложит вам на выбор поле без предварительного выбора функции. Если в предлагаемом списке полей нет нужного вам, возможно, вы не выполнили необходимые условия для их появления.

При вводе первых букв механизм IntelliSense будет предлагать вам на выбор поля и функции, основываясь на вашем вводе. Для навигации по списку используйте стрелки на клавиатуре, а для выбора элемента нажмите на клавишу **Tab**. В качестве альтернативы вы можете выбирать нужные вам элементы из списка двойным щелчком мыши.

Чтобы перейти на новую строку, используйте сочетание клавиш **Shift+Enter**. Теперь вы знаете еще больше о правилах написания кода.

8. Если сейчас перенести поле **TOTAL SALES SUM** из панели **Fields** (Поля) в таблицу, вы увидите, что значения в нем будут такие же. Основное от-

личие состоит в том, что значение меры останется прежним, тогда как столбец TOTAL SALES на каком-то этапе может быть переименован или удален, что в будущем повлечет проблемы в формуле.

9. Теперь мы можем убрать поле TOTAL SALES из визуализации, поскольку его роль выполняет созданная мера.
10. Используем функцию CALCULATE для подсчета сумм продаж по кварталам. Применение функции CALCULATE обязывает нас работать именно с мерой, здесь мы не можем указывать ссылку на таблицу или столбец.
11. Введите в строку формул имя меры, знак равенства, после чего напишите функцию CALCULATE, на вход которой нужно подать меру. На рис. 12.18 показано, как должна выглядеть итоговая формула.

The screenshot shows the Power BI formula editor with the following content:

```
1 CALQ = CALCULATE([TOTAL SALES SUM])
```

Below the formula, there is a table preview showing data for the years 2018 and 2020 across different regions and seasons. The table has columns: Year, Region, Season, Year, Month, Day, and Value. All rows show a value of 5734 for the 'Value' column.

Year	Region	Season	Year	Month	Day	Value
2018	East	Autumn	2020	Qtr 1	February	14
2018	East	Autumn	2020	Qtr 1	February	15
2018	East	Autumn	2020	Qtr 1	February	16
2018	East	Autumn	2020	Qtr 1	February	17
2018	East	Autumn	2020	Qtr 1	February	18
2018	East	Autumn	2020	Qtr 1	February	19
2018	East	Autumn	2020	Qtr 1	February	20

Рис. 12.18. Формула меры для расчета общей суммы

12. Перетащите меру CALQ из панели Fields (Поля) на дашборд.

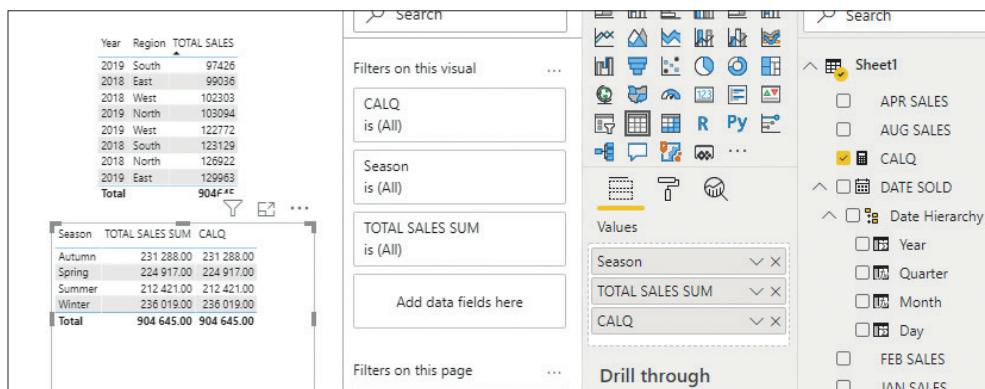


Рис. 12.19. Мера CALQ, добавленная на дашборд

13. Истинная мощь функции CALCULATE проявляется при совместном использовании с функцией FILTER. Давайте скомбинируем функции FILTER и CALCULATE, чтобы в поле Season осталось только значение Spring, как показано на рис. 12.20.

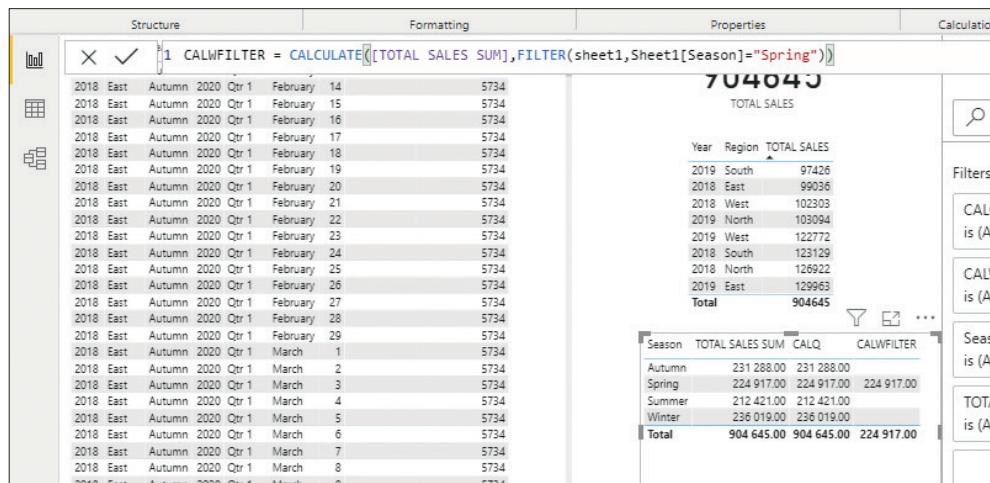


Рис. 12.20. Функция CALCULATE с фильтром по Spring

14. Теперь перетащите меру **CALWFILTER** из панели **Fields** (Поля) на дашборд. Заметьте, что значения показываются только напротив сезона **Spring**.

15. Добавьте поле **Year** на дашборд, как показано на рис. 12.21.

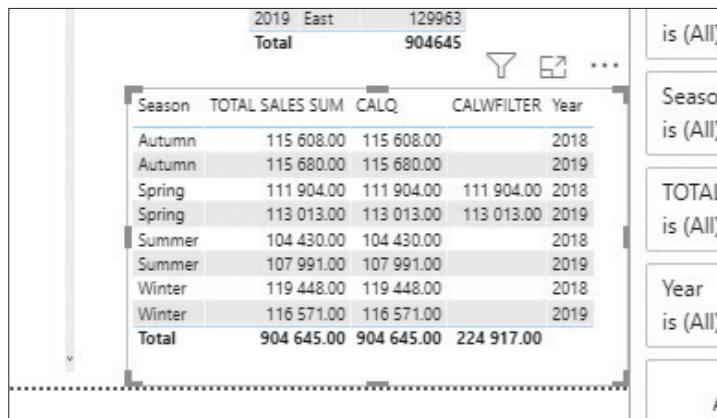


Рис. 12.21. Год, перенесенный на дашборд

16. Теперь фильтрация работает по 2018 и 2019 годам. Вы можете использовать сортировку в столбце **Year** для вывода информации по возрастанию или убыванию.

Надеюсь, описанный здесь сценарий поможет вам лучше понять, чего можно добиться с использованием мер, написанных на DAX. Сейчас же мы поговорим об организации мер.

Организация мер

Меры в списке полей можно узнать по иконке с калькулятором слева от имени, что видно по рис. 12.22.

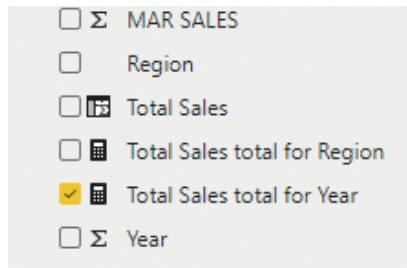


Рис. 12.22. Обозначение мер в списке полей

Можно разместить все наши меры в одной *папке отображения* (display folder), даже если они ассоциированы с разными таблицами.

1. Перейдите в левой навигационной панели в Power BI на вкладку **Model** (Модель).
2. Выделите меру, которую хотите перенести в папку.

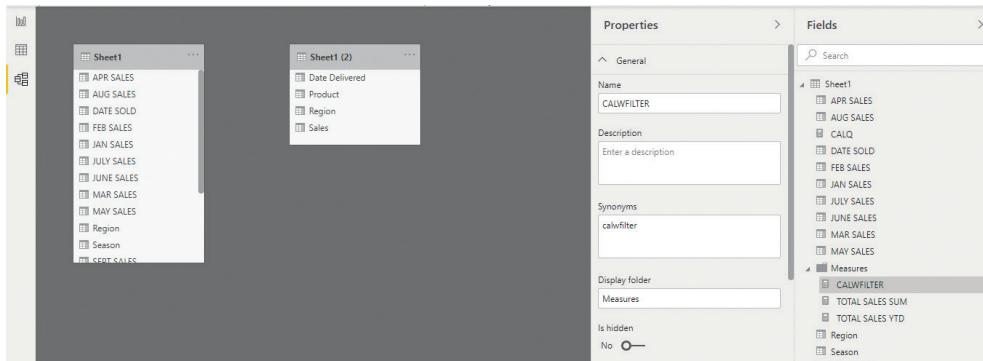


Рис. 12.23. Создание папки отображения на вкладке **Model** (Модель)

3. Слева от панели **Fields** (Поля) появится панель **Properties** (Свойства).
4. В поле **Display folder** (Папка отображения) на панели свойств введите имя папки, в которую хотите поместить меру.

Если желаете создать иерархию папок, введите название родительской и дочерней папок через обратный слеш (\). Например, **Measures\Sales**.

Из данного раздела вы узнали, как можно организовать иерархию папок отображения на панели **Fields** (Поля).

ЗАКЛЮЧЕНИЕ

В заключительной главе книги вы узнали о ключевых отличиях между языками DAX и M. Теперь вы сможете легко распознавать структуру формул на языке DAX и при желании применять их на практике, работая с Excel, Power Query и Power BI.

Также мы затронули тему технологии IntelliSense, помогающей разработчикам писать формулы на DAX для вычисляемых столбцов, мер и таблиц. При помощи языка DAX вы можете создавать с нуля вычисляемые столбцы и меры. Кроме того, написанные меры вы можете удобно организовать по папкам на панели **Fields** (Поля) в интерфейсе Power BI. Сами же меры помогут вам в написании сложных выражений для расчета результата, специфичного в рамках конкретных условий, который может быть получен в динамике или даже в реальном времени.

Надеемся, что прочтение данной книги прибавило вам знаний и зародило желание узнать еще больше о таком мощном и гибком инструменте, как Power Query.

Предметный указатель

А

Автоматическое обновление страницы [241](#)

Б

Блок выражений `in` [289](#)
Блок выражений `let` [289](#)
Быстрая мера в DAX [366](#)

В

Визуализации
вопросы и ответы [271](#)
график и гистограмма с группировкой [268](#)
карта [265](#)
карточка [270](#)
линейчатая диаграмма с группировкой [266](#)
многострочная карточка [271](#)
срез [270](#)
Вычисляемая мера в DAX [365](#)
Вычисляемое поле [60](#)
Вычисляемый столбец [55](#)
Вычисляемый столбец в DAX [363](#)

Г

Группировка данных [178](#)

Д

Добавление столбца с индексом [205](#)
Документирование и переименование шагов [73](#)

З

Зависимости запроса [77](#)
Замена пустых значений [167](#)

И

Извлечение данных [181](#)
Извлечение данных из столбцов [183](#)
Изменение настроек подключения в Excel [139](#)
Изменение шрифта [76](#)
Инкрементальное обновление [241](#)

Инструмент отмены

свертки (Unpivot) [147](#)

Использование параметров

для изменения логики [328](#)

для сортировки [330](#)

для фильтрации данных [326](#)

Использование первой строки в качестве заголовков

[167](#)

К

Ключи

естественные [107](#)

суррогатные [107](#)

Комментарии в расширенном редакторе

[85](#)

М

Метрики использования

[283](#)

Моделирование данных

[50](#)

Модель данных

[24](#)
составная модель [228](#)

Н

Настраиваемый столбец

[191](#)

Настройка запланированных обновлений

[240](#)

Недостатки Power Pivot

[62](#)

О

Обновление данных

[150](#)

Обратная загрузка данных в Excel

[90](#)

Объединение запросов

добавление [175](#)

комбинирование [171](#)

Объединение столбцов

[174](#)

Операция сведения (Pivot)

[148](#)

Очистка кеша в Power Query

[93](#)

П

Панель APPLIED STEPS (Примененные шаги)

[72](#)

Панель Queries (Запросы)

[67](#)

Панель Query Settings

(Параметры запроса) [72](#)

-
- Папка отображения 373
 Параметрический запрос 340
 Параметры 193
 создание 194
 таблица параметров 199
 Подключение к базе данных из Excel 119
 Подключение к базе данных
 из Power BI 121
 Подключение к веб-страницам 111
 Подключение к таблице/диапазону 108
 Пользовательские подключения 124
 Пользовательские функции 336
 Профилирование данных 99
 Качество столбца 102
 Профиль столбца 101
 Распределение столбцов 103
- P**
- Разделение столбцов 168
 Распространение дашбордов 276
 Расширенный редактор 79
 Режимы хранения 225
 Import 228
 DirectQuery 228
 Dual 228
 Ручная загрузка запросов 94
- C**
- Связь один ко многим 54
 Создание дашбордов 243
 Создание сводной таблицы 41
 Создание столбца с индексом 154
 Создание условного столбца 155
 Создание фильтров
 с множественными критериями 163
 с одним критерием 160
 Сохранение и публикация
 дашбордов 272
 Срезы 43
 Стока ввода формул 75
 Структурированный столбец 173
 Схема звезда 106
- T**
- Типы данных
 запись 294
- справка 293
 таблица 295
 текстовый 291
 числовой 292
 Транзакционные базы данных 105
- У**
- Удаление верхних и/или нижних строк
 в таблице 154
 Удаление дублирующихся строк 165
 Удаление столбцов 153
 Удаление шагов запроса 73
 Установка SQL Server 309
- Ф**
- Фильтрация данных
 And/Or (И/ИЛИ) 159
 Фоновая загрузка данных 93
 Функции
 DateTime.LocalNow 354
 Duration.TotalDays 353
 IF 188
 List.Dates 349
 Modulo 206
 Related 56
 Text.Combine 308
 Text.Format 291
 Text.From 307
- Ш**
- Шаги запроса 72
- Я**
- Язык DAX 359
 Язык M 288
 #shared 291, 297
- I**
- IntelliSense 361
- Р**
- Power BI 25
 Power Pivot 24
 Power Query 29

Книги издательства «ДМК Пресс» можно заказать
в торгово-издательском холдинге «Планета Альянс» наложенным платежом,
выслав открытку или письмо по почтовому адресу:
115487, г. Москва, 2-й Нагатинский пр-д, д. 6А.
При оформлении заказа следует указать адрес (полностью),
по которому должны быть высланы книги;
фамилию, имя и отчество получателя.
Желательно также указать свой телефон и электронный адрес.
Эти книги вы можете заказать и в интернет-магазине: www.a-planeta.ru.
Оптовые закупки: тел. (499) 782-38-89.
Электронный адрес: books@aliens-kniga.ru.

**Линда Фоукс
Уоррен Спарроу**

Изучаем Power Query

**Наглядный подход к подключению и преобразованию данных
из множества источников для Power BI и Excel**

Главный редактор *Мовчан Д. А.*
Зам. главного редактора *Сенченкова Е. А.*
dmkpress@gmail.com
Перевод *Гинько А. Ю.*
Корректор *Синяева Г. И.*
Верстка *Луценко С. В.*
Дизайн обложки *Мовчан А. Г.*

Формат 70×100 1/16.
Гарнитура «PT Serif». Печать цифровая.
Усл. печ. л. 27,64. Тираж 200 экз.

Веб-сайт издательства: www.dmkpress.com