Современная аналитика данных в Excel

Джордж Маунт

Modern Data Analytics in Excel

Using Power Query, Power Pivot, and More for Enhanced Data Analytics

George Mount

Джордж Маунт

Современная аналитика данных в Excel

2025

Маунт Дж.
Современная аналитика данных в Excel: - 2025. — 208 с.: ил.

Рассмотрены современные методы очистки, анализа и визуализации данных в Microsoft Excel. Описаны инструменты Power Query для создания воспроизводимых процессов подготовки данных, средства Power Pivot для построения реляци-онных моделей и настройки аналитических показателей. Приведены практические примеры использования динамических массивов, функций на базе искусственного интеллекта и интеграции с языком Python. Показано, как создавать отчеты и анали-тические материалы, ранее считавшиеся трудновыполнимыми в Excel. Книга ори-ентирована на специалистов по данным, бизнес-аналитиков и пользователей Excel, заинтересованных в расширении своих возможностей.

Для аналитиков данных

Оглавление

Предисловие **11**
Цель обучения 11
Предварительные требования 11
Технические требования 11
Необходимые навыки 12
Как я к этому пришел? 12
Что такое «современная аналитика»? Почему именно Excel? 13
Структура книги 14
Упражнения в конце глав 14
Отбор тем 15
Условные обозначения 15
Использование примеров кода 16
Контакты 16
Благодарности 16

ЧАСТЬ I. ОЧИСТКА И ПРЕОБРАЗОВАНИЕ ДАННЫХ В POWER QUERY **19**

Глава 1. Таблицы — проводники в современный Excel **21**
Создание заголовков таблицы и ссылки на них 21
Добавление строки итогов к таблице 23
Именование таблиц Excel 25
Форматирование таблиц Excel 25
Изменение диапазона таблицы 26
Упорядочивание данных для анализа 26
Заключение 27
Упражнения 28

Глава 2. Первые шаги в Power Query **29**
Что такое Power Query? 29
Power Query как «разрушитель мифов» об Excel 29
«Excel не воспроизводит результаты» 29
«В Excel нет настоящего null» 30
«Excel не может обработать более 1 048 576 строк» 31
Power Query как инструмент ETL в Excel 31
Extract (Извлечение) 31
Transform (Преобразование) 33
Load (Загрузка) 33
Обзор редактора Power Query 34
Лента 34

Запросы ... 36
Импортированные данные ... 37
Выход из редактора Power Query ... 40
Возвращение в редактор Power Query ... 41
Профилирование данных в Power Query ... 41
Что такое профилирование данных? ... 42
Опции предварительного просмотра данных ... 42
Monospaced и Show whitespace ... 42
Column quality и Column distribution ... 43
Что такое «допустимое» значение? ... 43
Отсутствующие значения ... 43
Ошибки в ячейках ... 44
Column profile (Профиль столбца) ... 45
Как убрать ограничение на тысячу строк? ... 46
Окончание профилирования данных ... 47
Заключение ... 47
Упражнения ... 47

Глава 3. Преобразование строк в Power Query ... 48
Удаление пропущенных значений ... 48
Обновление запроса ... 50
Разделение данных на строки ... 52
Заполнение заголовков и пустых ячеек ... 55
Замена заголовков столбцов ... 55
Заполнение пропущенных значений ... 55
Заключение ... 56
Упражнения ... 56

Глава 4. Преобразование столбцов в Power Query ... 57
Изменение регистра столбца ... 57
Разделение на столбцы ... 58
Изменение типов данных ... 59
Удаление столбцов ... 59
Работа с датами ... 60
Создание пользовательских столбцов ... 61
Загрузка и проверка данных ... 62
Вычисляемые столбцы и собственные расчеты ... 62
Изменение структуры данных ... 63
Заключение ... 65
Упражнения ... 65

Глава 5. Объединение и добавление данных в Power Query ... 66
Добавление нескольких источников ... 66
Подключение к внешним рабочим книгам Excel ... 66
Добавление запросов ... 69
Реляционные соединения ... 70
Левое внешнее соединение: почти то же, что и *VLOOKUP()* ... 72
Внутреннее соединение: только точное соответствие ... 75
Управление вашими запросами ... 76
Группировка запросов ... 76
Просмотр зависимостей запросов ... 77

Заключение ... 78
Упражнения ... 79

ЧАСТЬ II. МОДЕЛИРОВАНИЕ И АНАЛИЗ ДАННЫХ С ПОМОЩЬЮ POWER PIVOT ... 81

Глава 6. Знакомство с Power Pivot ... 83
Что такое Power Pivot? ... 83
Зачем нужен Power Pivot? ... 83
Power Pivot и модель данных ... 86
Подключение надстройки Power Pivot ... 87
Краткий обзор надстройки Power Pivot ... 88
Заключение ... 89
Упражнения ... 90

Глава 7. Создание реляционной модели данных в Power Pivot ... 91
Подключение данных к Power Pivot ... 91
Создание взаимосвязей между таблицами ... 92
Таблицы фактов и таблицы измерений ... 95
Упорядочивание диаграммы ... 95
Редактирование связей ... 96
Загрузка результатов в Excel ... 97
Понятие кардинальности ... 100
Связь «один к одному» ... 100
Связь «один ко многим» ... 101
Связь «многие ко многим» ... 102
Почему так важна кардинальность? ... 102
Направление фильтрации ... 103
Фильтрация *orders* через *users* ... 104
Фильтрация *users* через *orders* ... 105
Направление фильтрации и кардинальность ... 105
От теории к практике ... 106
Создание вычисляемых столбцов в Power Pivot ... 106
Вычисления в Power Query или в Power Pivot? ... 106
Пример: расчет нормы прибыли ... 107
Замена значений в столбце с помощью *SWITCH()* ... 108
Создание иерархий и работа с ними ... 110
Создание иерархии в Power Pivot ... 110
Использование иерархии в сводной таблице ... 111
Загрузка модели данных в Power BI ... 112
Power BI как третий инструмент «современного Excel» ... 112
Импорт модели данных в Power BI ... 113
Просмотр данных в Power BI ... 115
Заключение ... 116
Упражнения ... 116

Глава 8. Создание мер DAX и показателей KPI в Power Pivot ... 118
Создание мер DAX ... 118
Создание неявных мер ... 118
Создание явных мер ... 120

Создание показателей KPI ... 124
Настройка стилей значков ... 127
Добавление показателя KPI в сводную таблицу ... 127
Заключение ... 128
Упражнения ... 128

Глава 9. Функции DAX в Power Pivot ... 130
Функция *CALCULATE()* ... 130
Контекст фильтра ... 130
Функция *CALCULATE()* с одним условием ... 131
Функция *CALCULATE()* с несколькими условиями ... 132
Условие И ... 132
Условие ИЛИ ... 132
Функция *CALCULATE()* с условием *ALL()* ... 132
Функции аналитики времени ... 135
Добавление таблицы дат ... 136
Создание базовых мер для аналитики времени ... 137
Заключение ... 141
Упражнения ... 141

ЧАСТЬ III. ИНСТРУМЕНТЫ АНАЛИТИКИ В EXCEL ... 143

Глава 10. Введение в функции динамических массивов ... 145
Функции динамических массивов ... 145
Что такое массив в Excel? ... 145
Ссылки на массивы ... 146
Ссылки на статические массивы ... 146
Ссылки на динамические массивы ... 147
Формулы массива ... 147
Формулы статического массива ... 147
Функции динамического массива ... 149
Использование функций динамического массива ... 149
Поиск уникальных и неповторяющихся значений с помощью функции *UNIQUE()* ... 150
Разница между уникальными и отличающимися значениями ... 150
Использование оператора динамического диапазона ... 151
Фильтрация записей с помощью функции *FILTER()* ... 152
Добавление заголовков столбцов ... 153
Фильтрация по нескольким условиям ... 154
Условие И ... 154
Условие ИЛИ ... 154
Вложенные условия И/ИЛИ ... 154
Сортировка с помощью функции *SORTBY()* ... 154
Сортировка по нескольким диапазонам ... 155
Сортировка без включения столбца сортировки в результат ... 156
Современный поиск с помощью функции *XLOOKUP()* ... 156
Сравнение функций *XLOOKUP()* и *VLOOKUP()* ... 157
Базовые возможности функции *XLOOKUP()* ... 158
Обработка ошибок с помощью функции *XLOOKUP()* ... 158
Функция *XLOOKUP()* и столбцы слева ... 159

Другие функции динамического массива ... 159
Динамические массивы и современный Excel ... 160
Заключение ... 161
Упражнения ... 161

Глава 11. Дополненная аналитика и будущее Excel ... 162
Растущая сложность данных и аналитики ... 162
Excel и self-service BI-системы ... 163
Excel для дополненной аналитики ... 164
Использование Analyze Data для получения результатов, сгенерированных ИИ ... 164
Построение статистических моделей с помощью XLMiner ... 168
Чтение данных с изображения ... 171
Анализ настроений с помощью Azure Machine Learning ... 173
Заключение ... 177
Упражнения ... 177

Глава 12. Python и Excel ... 178
Предварительные требования ... 178
Роль Python в современном Excel ... 179
«Клей» для огромного набора инструментов ... 179
Сетевой эффект сокращает время разработки ... 180
Добавьте современные технологии к Excel ... 180
Модульное тестирование ... 180
Системы контроля версий ... 181
Разработка пакетов и их распространение ... 181
Совмещение Python и Excel с помощью пакетов *pandas* и *openpyxl* ... 182
Зачем нужен *pandas* для работы с Excel? ... 182
Ограничения при работе с *pandas* ... 182
Что умеет openpyxl? ... 182
Использование *openpyxl* вместе с *pandas* ... 183
Другие пакеты Python для Excel ... 183
Пример автоматизации Excel с помощью *pandas* и *openpyxl* ... 184
Очистка данных с помощью *pandas* ... 185
Работа с метаданными ... 186
Поиск по шаблону и регулярные выражения ... 186
Обработка отсутствующих значений ... 187
Процентильное ранжирование ... 188
Создание отчета с помощью *openpyxl* ... 189
Создание рабочего листа для отчета ... 189
Вставка диаграмм ... 190
Способ 1: создание диаграммы Excel ... 190
Способ 2: вставка изображения из Python ... 191
Диаграммы Excel и Python ... 193
Добавление стилизованной таблицы ... 194
Изменение формата на проценты ... 194
Преобразование в таблицу Excel ... 194
Применение условного форматирования ... 195
Автоподбор ширины столбцов ... 195
Заключение ... 196
Упражнения ... 196

Глава 13. Заключение и дальнейшие шаги 197
Другие функциональности Excel 197
Функции *LET()* и *LAMBDA()* 197
Power Automate, сценарии Office и Excel Online 198
Дальнейшее изучение Power Query и Power Pivot 199
Power Query и M 199
Power Pivot и DAX 200
Power BI для информационных панелей и отчетов 201
Azure и облачные вычисления 201
Программирование на Python 202
Большие языковые модели и инженерия запросов 202
Напутствие 203

Предметный указатель 204

Об авторе 206

Об изображении на обложке 207

Предисловие

Добро пожаловать в революцию Excel. Пришло время изменить ваше представление об Excel и его использовании, и тогда вы сможете значительно повысить свою производительности и более эффективно работать с данными. Эта книга познакомит вас с возможностями «современного Excel» и некоторыми другими полезными инструментами аналитики.

Цель обучения

По завершении чтения книги вы научитесь пользоваться инструментами современного Excel для очистки данных, их анализа, создания отчетов и расширенной аналитики. В частности, вы узнаете, как очищать и преобразовывать данные с помощью Power Query, создавать в Power Pivot реляционные модели для построения сложных анализов, а также познакомитесь с другими инструментами аналитики в Excel для последующей автоматизации и улучшения своей работы.

Предварительные требования

Чтобы достигнуть поставленных целей, вам надо учесть некоторые требования к своей технической оснащенности и навыкам работы в Excel.

Технические требования

Чтобы получить максимум от этой книги, рекомендуется, чтобы у вас был компьютер с операционной системой Windows и десктопное приложение Excel версии Microsoft 365. Функциональности, описанные в этой книге, относительно новые и могут быть недоступны в старых версиях Excel. Обратите внимание, что многие из этих инструментов разрабатываются также и для macOS, и совместимость с этой операционной системой может сильно различаться. Из-за быстрых темпов развития Excel сложно составить точный список того, что доступно в каждой новой версии.

В *главе 7* книги кратко рассказано о том, как загрузить модель данных из Excel в Power BI. Предполагается, что у вас, как у пользователя Microsoft 365 для ОС Windows, уже установлена бесплатная версия приложения Power BI Desktop. *Глава 12* посвящена интеграции Python и Excel, и в ней указано, где можно бесплатно скачать Python. Все приведенные в книге примеры и упражнения предназначены для выполнения исключительно в Excel, поэтому какие бы то ни было другие про-

граммы вам не понадобятся. Однако в процессе работы вам нужно будет добавить несколько надстроек Excel.

Необходимые навыки

Эта книга предназначена для пользователей со средним уровнем владения Excel, которые хотят познакомиться с его новыми возможностями, о которых они еще не знают. Чтобы получить от книги желаемый эффект, вы должны уметь:

- пользоваться абсолютными, относительными и смешанными ссылками на ячейки;
- работать с условной логикой и функциями условного агрегирования: `IF()` (`ЕСЛИ()`), `SUMIF()` (`СУММЕСЛИ()`), `SUMIFS()` (`СУММЕСЛИМН()`) и др.;
- объединять источники данных с помощью поисковых функций `VLOOKUP()` (`ВПР()`), `INDEX()` (`ИНДЕКС()`), `MATCH()` (`ПОИСКПОЗ()`) или др.;
- сортировать, фильтровать и агрегировать данные с помощью сводных таблиц;
- создавать графики (гистограммы, линейные графики и т. д.).

Если вы хотите получить больше знаний по этим темам, я могу порекомендовать книгу: Michael Alexander, Dick Kusleika «Microsoft Excel 365 Bible» (Wiley, 2022)[1].

В *части III* книги вы познакомитесь с передовыми концепциями в статистике, программировании и смежных областях. Не расстраивайтесь, если поначалу эти темы покажутся вам слишком сложными. Есть множество ресурсов, которые помогут вам овладеть этими знаниями, и по ходу повествования я буду приводить полезные ссылки. Основная цель этой книги — продемонстрировать широкие возможности, которые предоставляет Excel.

Если прежде, чем браться за эту книгу, вы предпочли бы сначала углубить свои знания, я рекомендую вам прочитать мою книгу «Погружение в аналитику данных: от Excel к Python и R» (БХВ-Петербург, 2023)[2]. В ней вы найдете исчерпывающую информацию и рекомендации по передовым методам аналитики, программированию на Python и другим темам, связанным с современной аналитикой данных в Excel.

Как я к этому пришел?

Мое знакомство с миром данных началось с Excel в начале 2010-х годов, еще до того, как наука о данных и искусственный интеллект захватили мир. В то время Excel казался закрытой системой. Если вы хотели заниматься сложной аналитикой, вам, как правило, советовали перейти на Python или R. Для работы с реляционными моделями данных рекомендовали Access. Реализация многих комплексных анализов данных и автоматизация их проведения требовали создания громадных модулей

[1] См. **https://clck.ru/3JVVEL**.

[2] См. **https://clck.ru/3JVY4U**.

VBA и тяжелых формул массивов, что делало работу пользователя не слишком комфортной.

Какое-то время казалось, что Excel скоро морально устареет. Однако современный Excel, дополненный различными функциональностями и приложениями, претерпел значительную трансформацию, о которой я расскажу далее.

Что такое «современная аналитика»? Почему именно Excel?

Современная аналитика подразумевает использование новейших инструментов и методов для подготовки и анализа данных, начиная от простого ретроспективного анализа и заканчивая прогностическим моделированием и искусственным интеллектом. В условиях изменчивой среды, в которой решения принимаются на основе данных, очень важно иметь универсальные и совместимые друг с другом инструменты, позволяющие пользователям выполнять разнообразные аналитические операции.

Ранее Excel не отвечал этим требованиям. Однако за последнее десятилетие он претерпел значительную трансформацию, превратившись в настоящий локомотив для современной аналитики.

Цель этой книги — развеять распространенные заблуждения технических специалистов об Excel и продемонстрировать его возможности в области современной аналитики. В книге показано, как работать с такими инструментами, как Power Pivot, Power Query и др., что опровергает мнение о том, что Excel ограничен базовыми формулами и функциями. В ней также подчеркивается, что современный Excel превратился в надежную платформу, помогающую решать сложные задачи анализа данных.

В итоге эта книга должна убедить вас, что Excel — это мощный и универсальный инструмент для современной аналитики. Книга призвана разрушить основные мифы об Excel и помочь техническим специалистам и менеджерам по максимуму задействовать его потенциал для эффективного анализа данных и принятия решений, ориентируясь на Excel как на важнейшую часть инструментария современной аналитики, способную формировать ее общее ви́дение и выбирать направление движения в нашем мире, управляемом данными.

Современный Excel и совместимость

В современной аналитике особое внимание уделяется совместимости, поэтому неудивительно, что многие инструменты, рассматриваемые в этой книге, также широко распространены и в других наборах инструментов для аналитиков. В частности, Power Query и Power Pivot, о которых речь пойдет в *частях I и II* соответственно, также доступны из Power BI — приложения Microsoft для бизнес-анализа и создания отчетов. В Power BI можно использовать и Python. Эти инструменты можно комбинировать друг с другом разными способами, и при освоении какого-либо одного из них вы, скорее всего, столкнетесь с упоминанием о других в том или ином контексте. В этой книге основное внимание уделено Excel, но важно понимать, как все эти приложения вписываются в более широкий набор инструментов современного аналитика.

Структура книги

Чтобы книга соответствовала целям обучения, я разделил ее материал на три части.

- *Часть I.* Очистка и преобразование данных в Power Query.

  Эта часть посвящена инструменту Power Query, выполняющему в Excel очистку данных, и использованию его для извлечения (Extract), преобразования (Transform) и загрузки (Load) данных (ETL). Вы познакомитесь со встроенным редактором Power Query, узнаете о профилировании данных и различных способах их преобразования, таких как фильтрация, разделение, агрегирование и объединение.

- *Часть II.* Моделирование и анализ данных с помощью Power Pivot.

  В этой части рассказано, как работать в Excel с инструментом Power Pivot, причем особое внимание уделено его использованию для создания отчетов. Вы узнаете, как создавать взаимосвязи и модель данных (Data Model), как дополнять ее вычисляемыми столбцами, ключевыми показателями эффективности (KPI, Key Performance Indicators) и пр., — в основном с использованием формульного языка Data Analysis Expressions (DAX).

- *Часть III.* Инструменты аналитики в Excel

  В этой части книги описаны несколько новых возможностей Excel, интересных для анализа данных. Вы узнаете о функциях динамического массива, которые позволяют выполнять быстрые и гибкие вычисления в таблицах. Кроме того, здесь рассказывается о предсказательной аналитике и искусственном интеллекте, обсуждаются возможности их применения в Excel и предлагается заглянуть в будущее этой программы. Книга завершается рассмотрением более сложной темы — автоматическим созданием рабочей книги Excel с помощью Python. Это поможет вам эффективно объединить Python и Excel для расширения своих аналитических возможностей.

Упражнения в конце глав

Когда я читаю книги, то, как правило, пропускаю практические упражнения в конце каждой главы, поскольку считаю, что важнее сохранить скорость чтения. *Не делайте так!*

В конце большинства глав я предлагаю вам возможность применить полученные знания на практике. Упражнения и их решения находятся в папке exercises GitHub-репозитория к этой книге[3], подобранные по номерам глав. Я рекомендую вам сначала попробовать самостоятельно выполнить эти упражнения, а затем сравнить свои ответы с предложенными решениями. Этим вы не только улучшите свое понимание материала, но и подадите мне положительный пример.

[3] См. **https://clck.ru/3JVeGX**.

Отбор тем

Быстрые темпы развития Excel и количество появившихся в нем новых инструментов могут ошеломить. Чтобы не потерять фокус и излишне не утяжелять книгу, я, опираясь на свой многолетний опыт работы консультантом и наставником по Excel, тщательно отобрал для включения в нее темы с самыми широкими возможностями и максимальной полезностью для читателей со средним уровнем владения Excel.

Если ваша самая любимая или наиболее значимая функциональность Excel для современной аналитики не описана в этой книге, пожалуйста, поделитесь своим мнением с Excel-сообществом. Область аналитики данных в Excel выходит за рамки одной книги, и наше сообщество с радостью ждет ваших идей и опыта.

Вы готовы приступить к изучению современного Excel? Встречаемся в *главе 1*.

Условные обозначения

В этой книге используются следующие типографские обозначения:

- *Курсивный шрифт.*

  Курсивом выделены новые или важные термины, на которые нужно обратить внимание.

- **Полужирный шрифт.**

  Им выделяются элементы интерфейса, интернет-адреса (URL) и адреса электронной почты.

- `Моноширинный шрифт.`

  Им выделяются листинги программного кода, а также встречающиеся внутри абзацев текста ссылки на элементы программ — такие как имена переменных или функций, базы данных, типы данных, переменные среды, операторы и ключевые слова.

- Рубленый шрифт.

  Им выделяются имена файлов, расширения имен файлов и пути.

Этот значок обозначает совет или подсказку.

Этот значок обозначает примечание общего характера.

Этот значок обозначает предупреждение или предостережение.

Использование примеров кода

Вспомогательные материалы (примеры кода, упражнения и т. д.) доступны для скачивания со страницы этой книги на сайте ресурса GitHub[4].

Предназначение этой книги — помочь вам в решении ваших задач. Вы можете использовать любой пример кода, содержащийся в ней, в своих программах и документации. При этом вам не требуется обращаться к нам за разрешением, если только вы не воспроизводите существенную часть кода. Это, например, касается ситуаций, когда вы включаете в свою программу несколько фрагментов кода, приведенного в книге. Однако продажа или распространение примеров из книг издательства O'Reilly требует отдельного разрешения. Вы можете свободно цитировать эту книгу, включая примеры, при ответе на вопрос, но если хотите включить существенную часть приведенного здесь кода в документацию своего продукта, то вам следует связаться с нами.

Ссылка на оригинал приветствуется, но не является обязательной. Указание авторства обычно включает название книги, автора (авторов), издателя и ISBN. Например: «Modern Data Analytics in Excel by George Mount (O'Reilly). Copyright 2024 Candid World Consulting, LLC, 978-1-098-14882-9».

Если вы сочтете, что ваше обращение с примерами кода выходит за рамки добросовестного использования или условий, упомянутых ранее, можете обратиться к нам по адресу: **permissions@oreilly.com**.

Контакты

Мы создали для этой книги веб-страницу, на которой публикуются исправления, примеры и дополнительная информация[5].

Благодарности

Один из самых захватывающих моментов при написании книги и, в частности, благодарностей заключается в том, что в книге отражен определенный период моей жизни, и можно особо отметить людей, которые были значимы для меня в это время.

Многие из этих имен уже упоминались в благодарностях к моей предыдущей книге. Я очень признателен сотрудникам издательства O'Reilly Мишель Смит и Джону Хасселлу за то, что они дали мне зеленый свет на написание еще одной книги. Мой друг и тоже автор O'Reilly Тобиас Цвингман, чьи работы я рецензировал на протяжении нескольких лет, написал исключительно полезную техническую рецензию на мою книгу. К тому же мои родители Джонатан и Анджела Маунт безоговорочно

[4] См. **https://clck.ru/3JViRb**.

[5] См. **https://clck.ru/3JVjXB**.

меня поддерживали, о чем я и мечтать не мог. Не знаю, сколько матерей грезят о том, чтобы их дети написали книгу об Excel, но моя мать оказывала мне всю возможную поддержку.

Благодаря этому проекту у меня появилась возможность поближе познакомиться с некоторыми людьми. Я выражаю благодарность Алану Мюррею, Джозефу Стеку и Меган Финли за их бесценные технические рецензии. Меган, в частности, не только помогала мне, опираясь на свой значительный опыт технического редактирования, но и, как моя девушка, всячески поддерживала меня на протяжении всего процесса создания книги. (Как скажет вам любой автор, написание книги неизбежно превращается в семейное дело.) Кроме того, я благодарен Джеффу Стивенсу, Лауре Сепеси и Марку Депоу за их отзывы на рукопись.

Я также должен поблагодарить редакцию издательства O'Reilly, которая помогла мне пройти через весь процесс написания этой книги. Особая благодарность Саре Хантер за ее глубокие редакторские советы, которыми я руководствовался, приступая к написанию своей второй книги.

И наконец, я хотел бы выразить свою признательность всему сообществу Excel за то, что оно было ко мне таким доброжелательным и вдохновляющим. Эта программа для работы с электронными таблицами открыла для меня столько возможностей и помогла познакомиться с таким количеством интересных людей, что я не мог себе и представить. Я надеюсь, что с помощью этой книги я смог внести свой посильный вклад в ваше личное знакомство со средой Excel.

ЧАСТЬ I

Очистка и преобразование данных в Power Query

Таблицы — проводники в современный Excel

Excel может похвастаться широким набором инструментов для анализа, и из-за этого иногда сложно выбрать, с чего именно следует начать его изучение. Однако важнейшим моментом во взаимодействии с Excel является умение работать с его таблицами, так что в этой главе мы рассмотрим базовые компоненты таблиц Excel, которые служат основой для работы с Power Query, Power Pivot и другими инструментами, упомянутыми в этой книге, и отметим важность четкой организации данных в таблицах.

Чтобы работать с примерами этой главы, откройте из папки ch_01 сопроводительного репозитория к этой книге файл ch_01.xlsx[1].

Создание заголовков таблицы и ссылки на них

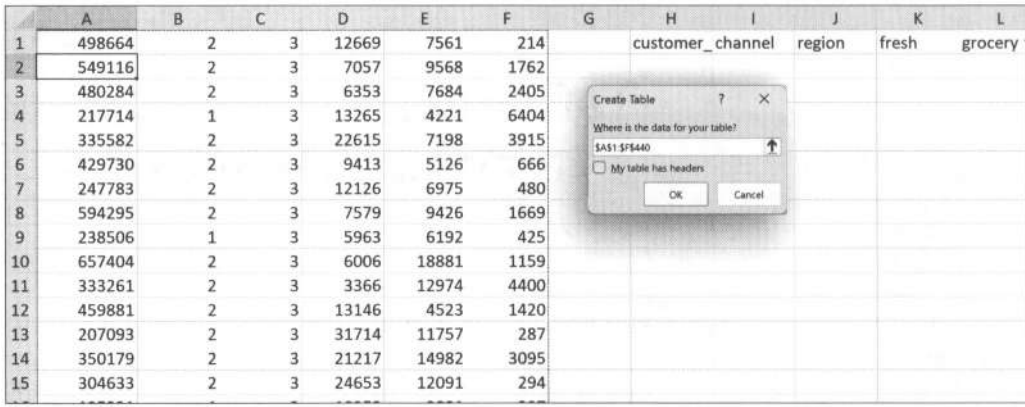
Набор данных, не имеющий заголовков столбцов, практически бесполезен, поскольку в нем отсутствует смысловой контекст, пригодный для интерпретации того, что содержится в каждом столбце. К сожалению, очень часто встречаются наборы данных, которые нарушают это основное правило. Таблицы Excel предоставляют все возможности для оснащения наборов данных четкими и информативными заголовками, подтверждая тем самым тот факт, что от их наличия напрямую зависит качество набора данных.

На рабочем листе `start` в файле ch_01.xlsx вы увидите, что данные в столбцах `A:F` не имеют соответствующих заголовков, которые пока находятся в столбцах `H:M`. Такое расположение данных и заголовков мало что поясняет. Чтобы это скорректировать, щелкните в любом месте исходных данных и перейдите на ленте на вкладку **Insert | Table** (Вставка | Таблица) — чтобы открыть диалоговое окно **Create Table** (Создание таблицы), показанное на рис. 1.1. Или вместо этого вы можете с тем же результатом нажать комбинацию клавиш <Ctrl>+<T> или <Ctrl>+<L> в любом месте исходных данных.

Диалоговое окно **Create Table** автоматически определяет, включают ли ваши данные заголовки, и позволяет изменить это. Сейчас заголовков нет. В этом случае столбцам набора данных автоматически присваиваются заголовки `Column1`, `Column2` и т. д. (`Столбец1`, `Столбец2`, ...).

---

[1] См. **https://clck.ru/3JhTPV**.



	A	B	C	D	E	F	G	H	I	J	K	L
1	498664	2	3	12669	7561	214		customer_channel		region	fresh	grocery
2	549116	2	3	7057	9568	1762						
3	480284	2	3	6353	7684	2405						
4	217714	1	3	13265	4221	6404						
5	335582	2	3	22615	7198	3915						
6	429730	2	3	9413	5126	666						
7	247783	2	3	12126	6975	480						
8	594295	2	3	7579	9426	1669						
9	238506	1	3	5963	6192	425						
10	657404	2	3	6006	18881	1159						
11	333261	2	3	3366	12974	4400						
12	459881	2	3	13146	4523	1420						
13	207093	2	3	31714	11757	287						
14	350179	2	3	21217	14982	3095						
15	304633	2	3	24653	12091	294						

Рис. 1.1. Преобразование исходных данных в таблицу

Теперь вы можете вырезать заголовки из столбцов `H:M` и вставить их в основную таблицу, чтобы было понятно, что находится в каждом столбце (рис. 1.2).

	A	B	C	D	E	F
1	customer_id	channel	region	fresh	grocery	frozen
2	498664	2	3	12669	7561	214
3	549116	2	3	7057	9568	1762
4	480284	2	3	6353	7684	2405
5	217714	1	3	13265	4221	6404
6	335582	2	3	22615	7198	3915
7	429730	2	3	9413	5126	666
8	247783	2	3	12126	6975	480
9	594295	2	3	7579	9426	1669
10	238506	1	3	5963	6192	425
11	657404	2	3	6006	18881	1159
12	333261	2	3	3366	12974	4400

Рис. 1.2. Таблица Excel с заголовками

Заголовки столбцов в таблицах Excel играют особую роль в наборе данных. Являясь частью таблицы, они работают как метаданные, а не сами данные. В отличие от обычных формул Excel, в таблицах Excel есть возможность программно различать заголовки и данные.

Чтобы проверить это различие на практике, перейдите на любую пустую ячейку рабочего листа и введите знак равенства (=). Выделите ячейки `A1:F1` для формирования ссылки, и вы увидите, что формула превратилась в `Table1[#Headers]` (`Таблица1[#Заголовки]`).

Вы можете использовать такие ссылки и в других функциях. Например, для динамического преобразования всех заголовков в верхний регистр можно использовать функцию `UPPER()` (`ПРОПИСН()`), как показано на рис. 1.3.

	A	B	C	D	E	F	G
1							Formula used:
2	customer_id	channel	region	fresh	grocery	frozen	=Table1[#Headers]
3	CUSTOMER_ID	CHANNEL	REGION	FRESH	GROCERY	FROZEN	=UPPER(Table1[#Headers])
4							
5	customer_id	channel	region	fresh	grocery	frozen	
6	498664	2	3	12669	7561	214	
7	549116	2	3	7057	9568	1762	
8	480284	2	3	6353	7684	2405	
9	217714	1	3	13265	4221	6404	
10	335582	2	3	22615	7198	3915	
11	429730	2	3	9413	5126	666	
12	247783	2	3	12126	6975	480	

Рис. 1.3. Формулы со ссылками на заголовки таблицы Excel

Добавление строки итогов к таблице

Как в каждой истории есть начало, середина и конец, так и в каждой таблице Excel есть строка заголовков, данные и итоговая строка (футер, подвал). Однако итоговую строку нужно добавлять вручную. Для этого щелкните в любом месте таблицы, перейдите в ленте на вкладку **Table Design** (Конструктор таблиц) и установите флажок **Total Row** (Строка итогов) в группе **Table Style Options** (Параметры стилей таблиц), как показано на рис. 1.4.

По умолчанию в строке итогов **Total** (Итог) вычисляется сумма последнего столбца данных — в нашем случае это столбец `frozen`. Однако вы можете настроить итог для

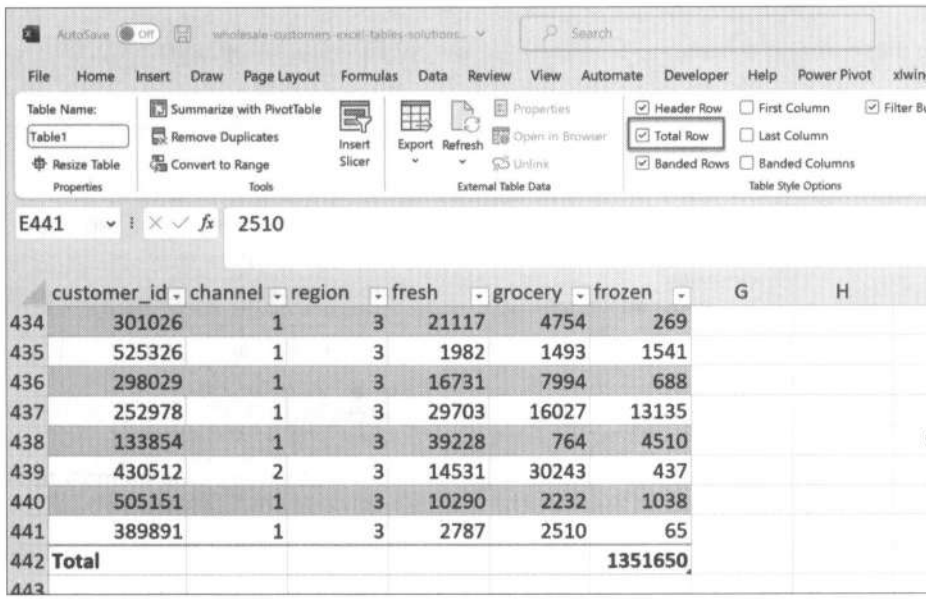


Рис. 1.4. Добавление строки итогов к таблице Excel

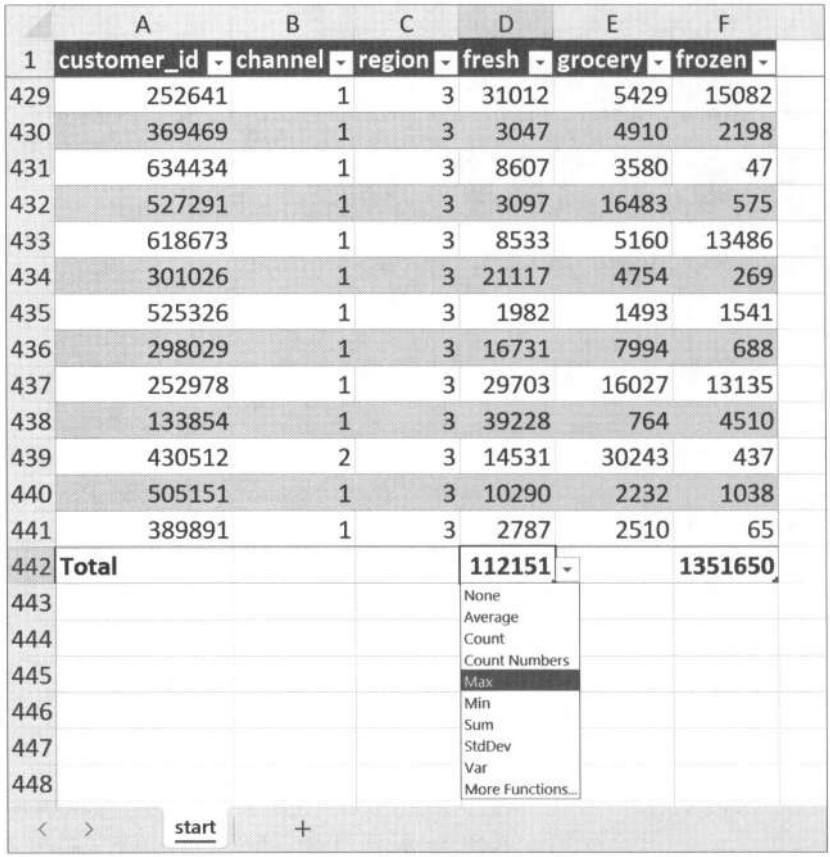


Рис. 1.5. Настройка строки итогов таблицы Excel

каждого столбца с помощью выпадающего списка. Например, можно найти максимальную сумму продажи для категории `fresh` (рис. 1.5).

В табл. 1.1 приведены ссылки на основные компоненты таблицы Excel для использования в формулах.

Таблица 1.1. Ссылки на компоненты таблицы Excel

Формула	Формула для русской версии Excel	На что ссылается
`=Table1[#Headers]`	`=Таблица1[#Заголовки]`	Заголовки таблицы
`=Table1`	`=Таблица1`	Данные таблицы
`=Table1[#Totals]`	`=Таблица1[#Итоги]`	Строка итогов таблицы
`=Table1[#All]`	`=Таблица1[#Все]`	Заголовки, данные и строка итогов таблицы

По мере улучшения своих навыков по работе с таблицами Excel вы узнаете и о других полезных ссылках, опирающихся на эти базовые компоненты таблицы: заголовки столбцов, тело таблицы и строка итогов.

Именование таблиц Excel

Преимуществом таблиц Excel является то, что они используют именованные диапазоны, а это способствует более структурированному подходу при работе с данными. Хотя ссылаться на таблицу как на `Table1` (`Таблица1`) уже удобнее, чем использовать адреса ячеек типа `A1:F22`, но лучше задать для таблицы описательное имя, которое будет отражать то, что собой представляют ее данные.

Для этого на ленте перейдите на вкладку **Formulas** (Формулы), в группе **Defined Names** (Определенные имена) выберите **Name Manager** (Диспетчер имен) и нажмите кнопку **Edit** (Изменить) для `Table1` (`Таблица1`). Измените название таблицы на `sales` и нажмите **OK**. На рис. 1.6 показано, как должен выглядеть ваш диспетчер имен после внесения этих изменений.

Рис. 1.6. Окно **Name Manager** в Excel

Обратите внимание, что как только вы закроете диспетчер имен, все ссылки на `Table1` (`Таблица1`) автоматически изменятся на новое имя `sales`.

Форматирование таблиц Excel

Будучи успешным пользователем Excel, вы должны знать, как важно представлять данные в красивом виде. Оформление таблиц может сразу повысить визуальную привлекательность вашего рабочего листа и повлиять на принятие решений. В Excel легко добавить чередование строк таблицы, цветные заголовки и многое другое. Чтобы настроить стиль таблицы, щелкните на любой ее ячейке и откройте на ленте вкладку **Table Design** (Конструктор таблиц). На рис. 1.7 показаны различные настройки, с помощью которых можно, например, изменить цвет таблицы, а также включить или выключить чередование строк с помощью флажка **Banded Rows** (Чередующиеся строки).

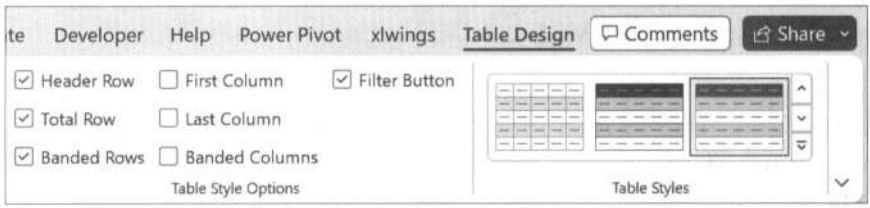


Рис. 1.7. Настройки стиля таблицы

Изменение диапазона таблицы

С помощью таблиц Excel очень просто решается проблема неправильных итогов при добавлении или удалении данных. Вы можете легко адаптировать формулы к изменениям в данных, используя структурированные ссылки, что обеспечит нужную точность. Кроме того, строка итогов в нижней части таблицы автоматически обновляется с учетом изменений в данных, и ее можно легко исключить из внешних ссылок, сохранив целостность ваших вычислений.

Вычислите сумму столбца `fresh` с помощью структурированной формулы `=SUM(sales[fresh])` (`=СУММ(sales[fresh])`). Автодополнение IntelliSense от Microsoft упрощает нам этот процесс, предлагая подходящие имена по мере ввода. Поэкспериментируйте с добавлением или удалением строк в таблице `sales`, а также с изменением данных в столбце `fresh`. Вы увидите, что общий объем продаж `fresh` динамически пересчитается и сохранит требуемую точность.

Обращение к данным с использованием именных ссылок, а не адресов ячеек, сводит к минимуму возможные ошибки с формулами, которые могли бы возникнуть из-за изменения размера таблицы или ее расположения. Таблицы также играют важную роль в предотвращении таких проблем, как отсутствие данных в сводных таблицах при добавлении новых строк.

Упорядочивание данных для анализа

Несмотря на очевидные преимущества использования таблиц, для выполнения быстрого и точного анализа данных более важным аспектом является хранение данных в надлежащем виде.

Рассмотрим в качестве примера таблицу `sales`. При попытке создать сводную таблицу PivotTable для расчета общего объема продаж по регионам возникнет проблема с форматом, в котором представлены данные. В идеале все продажи должны быть сведены в один столбец. Однако в текущей конфигурации продажи каждого отдела выделены в отдельные столбцы: `fresh` (свежие фрукты/овощи), `grocery` (бакалея) и `frozen` (заморозка). Excel не сможет понять, что все эти столбцы представляют одну и ту же метрику, а именно — продажи.

Этот и многие другие наборы данных трудно анализировать, потому что они хранятся в виде, неудобном для анализа. В этом случае могут помочь *правила упорядо-*

ченных данных (tidy data). Несмотря на то что Хэдли Уикхем (Hadley Wickham) в своей статье «Tidy Data» от 2014 года[2] приводит три таких правила, в нашей книге основное внимание будет уделяться только первому правилу: *каждая переменная формирует отдельный столбец*.

Набор данных `sales` нарушает это правило упорядоченных данных, поскольку в каждой строке есть несколько значений для одной и той же переменной для разных отделов. Эмпирическое правило гласит, что, если в нескольких столбцах записана одна и та же измеряемая величина, то данные, скорее всего, не упорядочены. Преобразование таких данных к упорядоченному виду значительно упростит анализ.

На рис. 1.8 вы можете сравнить набор данных до и после преобразования и увидеть, насколько более упорядоченными и удобными для анализа стали данные. В *главе 4* вы узнаете, как выполнить такое преобразование набора данных с помощью всего нескольких щелчков мышью. А пока рекомендую вам обратить свое внимание на рабочий лист `sales-tidy` из файла \ch_01\ch_01_solutions.xlsx, который уже преобразован, — самостоятельно убедитесь, насколько проще теперь получить сводную таблицу с общим объемом продаж по регионам.

	A	B	C	D	E	F	G	H	I	J	K	L
1	До:							После:				
3	customer_id	channel	region	fresh	grocery	frozen		customer_id	channel	region	department	sales
4	498664	2	3	12669	7561	214		498664	2	3	fresh	12669
5	549116	2	3	7057	9568	1762		498664	2	3	grocery	7561
6	480284	2	3	6353	7684	2405		498664	2	3	frozen	214
7	217714	1	3	13265	4221	6404		549116	2	3	fresh	7057
8	335582	2	3	22615	7198	3915		549116	2	3	grocery	9568
9	429730	2	3	9413	5126	666		549116	2	3	frozen	1762
10	247783	2	3	12126	6975	480		480284	2	3	fresh	6353
11	594295	2	3	7579	9426	1669		480284	2	3	grocery	7684
12	238506	1	3	5963	6192	425		480284	2	3	frozen	2405
13	657404	2	3	6006	18881	1159		217714	1	3	fresh	13265
14	333261	2	3	3366	12974	4400		217714	1	3	grocery	4221
15	459881	2	3	13146	4523	1420		217714	1	3	frozen	6404
16	207093	2	3	31714	11757	287		335582	2	3	fresh	22615
17	350179	2	3	21217	14982	3095		335582	2	3	grocery	7198
18	304633	2	3	24653	12091	294		335582	2	3	frozen	3915
19	125231	1	3	10253	3821	397		429730	2	3	fresh	9413
20	126155	2	3	1020	12121	134		429730	2	3	grocery	5126

Рис. 1.8. Список продаж до (*слева*) и после упорядочивания (*справа*)

Заключение

В этой главе мы заложили основу для эффективной работы с таблицами Excel. Более подробно о том, как по максимуму использовать все возможности таблиц, включая применение структурированных ссылок в формулах вычисляемых столбцов, читайте в книге Зака Барресса и Кевина Джонса «Таблицы Excel: Полное

[2] См. **https://clck.ru/3JbVCv**.

руководство по созданию, использованию и автоматизации списков и таблиц»[3]. Мы также рассмотрели здесь правильное упорядочивание данных — важнейший аспект любого успешного проекта по анализу данных в Excel. *Глава 2* посвящена быстрым преобразованиям данных с помощью Power Query.

Упражнения

Для закрепления знаний по созданию, анализу и обработке данных в таблицах Excel выполните следующие упражнения, используя набор данных `penguins` (пингвины) из файла ch_01_exercises.xlsx, расположенного в папке exercises\ch_01_exercises сопроводительного репозитория к этой книге[4]:

1. Преобразуйте данные в таблицу с именем `penguins`.
2. Используйте формулу со ссылкой, чтобы получить заголовки столбцов в верхнем регистре.
3. Создайте новый столбец `bill_ratio`, разделив `bill_length_mm` на `bill_depth_mm`.
4. Включите строку итогов для расчета среднего `body_mass_g`.
5. Уберите чередование цветных строк в таблице.

Готовое решение можно посмотреть в файле ch_01_exercise_solutions.xlsx, расположенном в той же папке репозитория.

[3] Zack Barresse, Kevin Jones. «Excel Tables: A Complete Guide for Creating, Using, and Automating Lists and Tables» (Holy Macro! Books, 2014). **https://clck.ru/3JbZtq**.

[4] См. **https://clck.ru/3JbdTg**.

Первые шаги в Power Query

В *главе 1* мы познакомились с таблицами Excel — проводниками в современную аналитику. В этой и следующих главах *части I* мы более подробно рассмотрим современный инструментарий Excel и сосредоточимся, в частности, на Power Query. Этот инструмент устраняет многие привычные ограничения Excel и предлагает удобную среду с низким уровнем кода.

Что такое Power Query?

Power Query — это инструмент для работы с данными, который позволяет пользователям легко подключаться к самым разным источникам данных, объединять и уточнять их с помощью Excel. Изначально это была надстройка Excel, но сейчас Power Query стал базовой функциональностью современного Excel, значительно упростив процесс импорта и очистки данных. Power Query обладает удобным интерфейсом для выполнения сложных действий с данными — такими как объединение таблиц, изменение формата данных и агрегирование, не требующим особых навыков в программировании.

Power Query как «разрушитель мифов» об Excel

Вокруг Excel, популярного бизнес-инструмента, за годы его существования возникло множество разных мифов. Однако многие из них больше не соответствуют действительности. В роли «разрушителей мифов» зачастую выступают сами аналитики, опровергая гипотезы и открывая истину. Большую часть предъявляемых к Excel претензий успешно снял и Power Query, позиционирующий себя как окончательный «разрушитель мифов» об Excel. В этом разделе с помощью Power Query мы опровергнем основные претензии к Excel.

«Excel не воспроизводит результаты»

Это распространенный сценарий: вы пытаетесь отредактировать отчет за прошлую неделю, и на вас давят приближающиеся сроки и назойливый руководитель. Автор отчета недоступен, и вы теряетесь в догадках, как этот отчет был сформирован. Рабочая книга выглядит как беспорядочный хаос из удаленных столбцов и измененных значений, что затрудняет расшифровку выполненных вычислений.

Воспроизводимость вычислений позволяет пользователю стабильно получать одни и те же результаты, используя идентичные входные данные и одни и те же действия. Но рабочая книга с отчетом не позволяет вам достичь этой цели, потому что при каждом открытии файла небольшие ошибки, возникающие на каждом шаге, сложные вычисления и прочие мелочи, вносящие неопределенность, приводят к несогласованным результатам.

Проблема с воспроизводимостью в обычном Excel стала серьезным поводом для критики этой программы. В результате многие технические специалисты стали опасаться использовать Excel, поскольку один удаленный столбец или жестко закодированная ячейка могут нарушить целостность всех результатов.

Тем не менее принятие решения о полном отказе от Excel, основываясь на предубеждениях о его старых ограничениях, было бы ошибкой. Современный Excel для обеспечения воспроизводимости предлагает использовать Power Query. С помощью Power Query пользователи могут создать копию исходных данных, выполнить последовательное преобразование данных, а каждое выполненное ими действие записывается в списке **Applied Steps** (Примененные шаги). Такой подход обеспечивает воспроизводимость и позволяет не запоминать действия по очистке данных, устраняя проблемы, которые ранее связывали с плохой воспроизводимостью в Excel.

VBA и воспроизводимость

Опытные пользователи Excel могут аргументировать свои возражения против утверждения о том, что классический Excel не имеет возможностей для стабильного воспроизведения, тем, что в состав Excel включен язык Visual Basic for Applications (VBA). Однако, хотя VBA действительно является комплексным скриптовым языком, позволяющим выполнять проверку и отладку результатов Excel, его трудно освоить, что усложняет его использование для большинства пользователей Excel. Кроме того, компания Microsoft выпускает минимальное количество обновлений VBA, вместо этого внедряя альтернативные языки, такие как Python, который мы рассмотрим в *главе 12*.

Независимо от продолжающихся споров о лучшем скриптовом языке для Excel, обеспечение воспроизводимого рабочего процесса не должно быть доступно только тем, кто умеет писать код. Эта возможность должна быть доступна пользователям с разными техническими навыками, и именно для этого и предназначен Power Query.

«В Excel нет настоящего null»

В реляционных базах данных широко используется концепция отсутствующего значения null, которое представляет собой неизвестные или неопределенные данные. Но в Excel нет зарезервированного ключевого слова для значений null, что приводит к проблемам с хранением и обработкой ряда данных. Пользователи применяют различные подходы для обозначения отсутствующих значений в Excel — например, оставляют такие значения пустыми или используют для них фиксированные значения типа NA. Такие допущения затрудняют выявление действительно неизвестных значений от тех, которые на самом деле равны нулю или намеренно оставлены пустыми.

Чтобы преодолеть это ограничение, в Power Query введено специальное значение `null` для обозначения отсутствующих данных. Это облегчает выполнение аккурат-

ного профилирования данных, удаление или замену отсутствующих значений и обеспечивает точность и воспроизводимость результатов.

«Excel не может обработать более 1 048 576 строк»

Другим часто используемым аргументом против Excel является его кажущаяся ограниченность при работе с большими данными. Критики Excel утверждают, что раз максимальный размер рабочего листа составляет примерно миллион строк, то Excel не подходит для современных задач в эпоху массивных наборов данных.

Решить эту проблему можно опять же за счет использования Power Query, который способен легко импортировать и обрабатывать миллионы строк и более. Хотя Excel сам по себе не может обрабатывать более миллиона строк, в редакторе Power Query пользователи могут агрегировать и обобщать данные перед загрузкой результатов в рабочий лист Excel.

Чтобы наглядно увидеть, как преодолевается ограничение Excel в миллион строк, прочитайте статью консультанта по аналитике Орландо Мескита, посвященную анализу 50 миллионов строк с помощью Excel Power Query[1].

Power Query как инструмент ETL в Excel

В мире технологий многие термины на первый взгляд могут показаться очень сложными. Иногда они даже представляют собой хитрые аббревиатуры. Однако при ближайшем рассмотрении эти понятия оказываются простыми и понятными.

Один из таких терминов — ETL, что расшифровывается как «Extract, Transform, Load» (извлечение, преобразование, загрузка). Очень часто администраторы баз данных и дата-инженеры горячо спорят о своих «конвейерах ETL» и «программном обеспечении ETL». Может сложиться впечатление, что только сертифицированные специалисты по работе с данными могут заниматься такими задачами.

Power Query упростил для нас процесс ETL, интегрировав его непосредственно в рабочий лист Excel. Не позволяйте техническим гикам запугивать вас! ETL представляет собой именно то, что заложено в его названии, и может быть реализован с помощью Excel.

В этом разделе далее дан пошаговый разбор этого процесса на простом примере. Чтобы продолжить его рассмотрение вместе с нами, откройте из папки ch_02 сопроводительного репозитория к этой книге файл ch_02.xlsx[2].

Extract (Извлечение)

Первым шагом в ETL является «извлечение» (extract) данных из внешнего источника. В Power Query есть возможность подключаться к различным источникам

[1] См. **https://clck.ru/3JbmM4**.

[2] См. **https://clck.ru/3JhTZg**.

данных, не ограничиваясь только рабочими книгами Excel. Вот несколько примеров источников данных, к которым он может подключаться:

- текстовые и CSV-файлы;
- реляционные базы данных, такие как Oracle, Microsoft SQL Server или SQLite;
- SharePoint;
- XML, HTML и веб-данные.

Однако для наглядного примера возьмем данные просто из рабочей книги Excel.

Для начала извлеките данные из таблицы `sales` с рабочего листа `sales` в файле ch_02.xlsx. Щелкните для этого в любом месте таблицы, затем на ленте перейдите на вкладку **Data | Get & Transform Data | From Table/Range** (Данные | Получить и преобразовать данные | Из таблицы/диапазона), как показано на рис. 2.1.

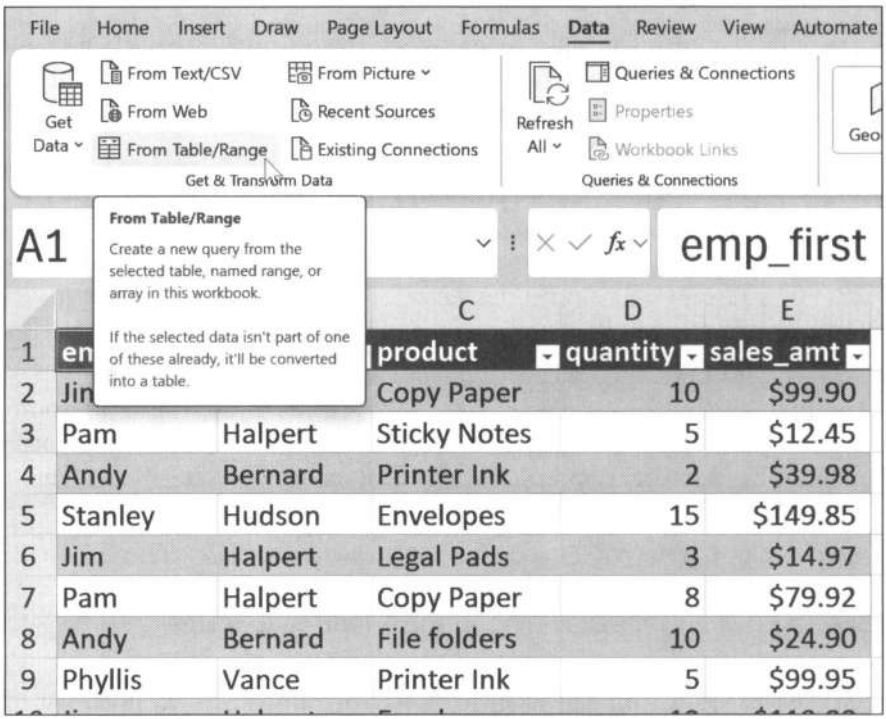


Рис. 2.1. Извлечение данных из таблицы

При этом Power Query требует, чтобы данные были оформлены в виде таблицы Excel, поэтому мы и посвятили *главу 1* обсуждению таблиц. Таблицы являются важными объектами при использовании современных инструментов Excel.

Хотя может показаться странным «подключение и извлечение» данных, которые уже находятся в вашей рабочей книге, но такой подход обоснован тем, что Power Query сохраняет исходные данные в их первоначальном виде. Даже если исходные данные находятся в той же рабочей книге, что и сам анализ, рекомендуется извлекать из нее подмножество данных для продолжения анализа.

Transform (Преобразование)

Следующий шаг — подключение к этим данным и выполнение необходимых преобразований (буква «T» в ETL — от *англ.* transform).

Преобразование данных включает в себя различные действия, необходимые для того, чтобы сделать данные удобными для дальнейшего использования, например:

- сортировку или фильтрацию строк;
- добавление, удаление, переименование или вычисление столбцов;
- объединение данных из различных источников или изменение структуры данных.

Когда вы загружаете свою таблицу в Power Query, открывается редактор Power Query, предлагающий множество опций для очистки и преобразования данных. Он может показаться слишком сложным, особенно если вы привыкли к классической среде Excel. Впрочем, озабочиваться этим не стоит, поскольку мы будем разбираться с этим редактором постепенно, шаг за шагом, на протяжении всей *части I*.

Далее в книге вы узнаете, как выполнять некоторые задачи по очистке данных. Но сначала давайте выполним простое преобразование данных, добавив столбец с порядковым номером строки. Для этого на ленте Power Query выберите пункт **Add Column | Index Column | From 1** (Добавление столбца | Столбец индекса | От 1), как показано на рис. 2.2.

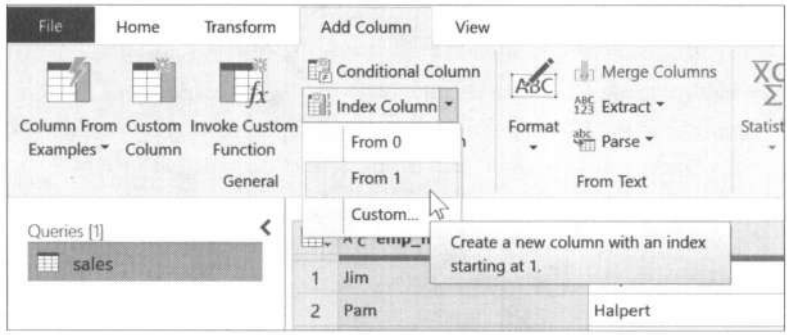


Рис. 2.2. Добавление столбца индекса в Power Query

Load (Загрузка)

И наконец, в редакторе Power Query перейдите на вкладку **Home** (Главная) и выберите **Close & Load** (Закрыть и загрузить). Это действие загрузит наш немного измененный набор данных в другую таблицу Excel и на новый рабочий лист (рис. 2.3).

Поздравляем вас с завершением задания по ETL:

- вы извлекли (extract) исходные данные из таблицы Excel;
- преобразовали (transform) данные с помощью редактора Power Query;
- загрузили (load) результаты обратно в Excel.

	A	B	C	D	E	F
1	emp_first	emp_last	product	quantity	sales_amt	Index
2	Jim	Halpert	Copy Paper	10	99.9	1
3	Pam	Halpert	Sticky Notes	5	12.45	2
4	Andy	Bernard	Printer Ink	2	39.98	3
5	Stanley	Hudson	Envelopes	15	149.85	4
6	Jim	Halpert	Legal Pads	3	14.97	5
7	Pam	Halpert	Copy Paper	8	79.92	6
8	Andy	Bernard	File folders	10	24.9	7
9	Phyllis	Vance	Printer Ink	5	99.95	8
10	Jim	Halpert	Envelopes	12	119.88	9
11	Pam	Halpert	Legal Pads	7	17.43	10
12	Andy	Bernard	Copy Paper	4	39.96	11
13	Jim	Halpert	Printer Ink	8	79.92	12
14	Phyllis	Vance	Envelopes	15	74.85	13
15	Andy	Bernard	Legal Pads	3	59.97	14
16	Stanley	Hudson	Rubber Bands	60	14.94	15
17						

Рис. 2.3. Данные, загруженные из Power Query в таблицу Excel

Обзор редактора Power Query

Теперь, когда мы рассмотрели очень простой пример ETL-процесса в Power Query, давайте поближе познакомимся с этим редактором. Для этого откройте рабочий лист `penguins`, который также находится в файле ch_02.xlsx.

Чтобы начать работу, загрузите таблицу `penguins` в Power Query. Если вы не помните, как это сделать, вернитесь к предыдущему разделу. После загрузки таблицы редактор Power Query должен выглядеть так, как показано на рис. 2.4.

Давайте уделим немного времени тому, чтобы глубже изучить и оценить уникальную среду, в которой вы находитесь. Редактор Power Query похож на интерфейс Excel с лентой, но при этом работает как самостоятельная программа. В этом разделе далее мы рассмотрим его элементы.

Лента

В верхней части окна вы можете видеть ленту Power Query с опциями, очень похожую на привычный интерфейс Excel (рис. 2.5).

На ленте есть четыре вкладки: **Home** (Главная), **Transform** (Преобразование), **Add Column** (Добавление столбца) и **View** (Просмотр):

- **Home** (Главная).

  Как и в обычном Excel, на вкладке **Home** собраны основные команды Power Query — например, выбор строк, удаление столбцов и т. д. Но, в отличие от главной вкладки Excel, где расположены команды для форматирования данных, в Power Query здесь находятся команды для преобразования и очистки данных.

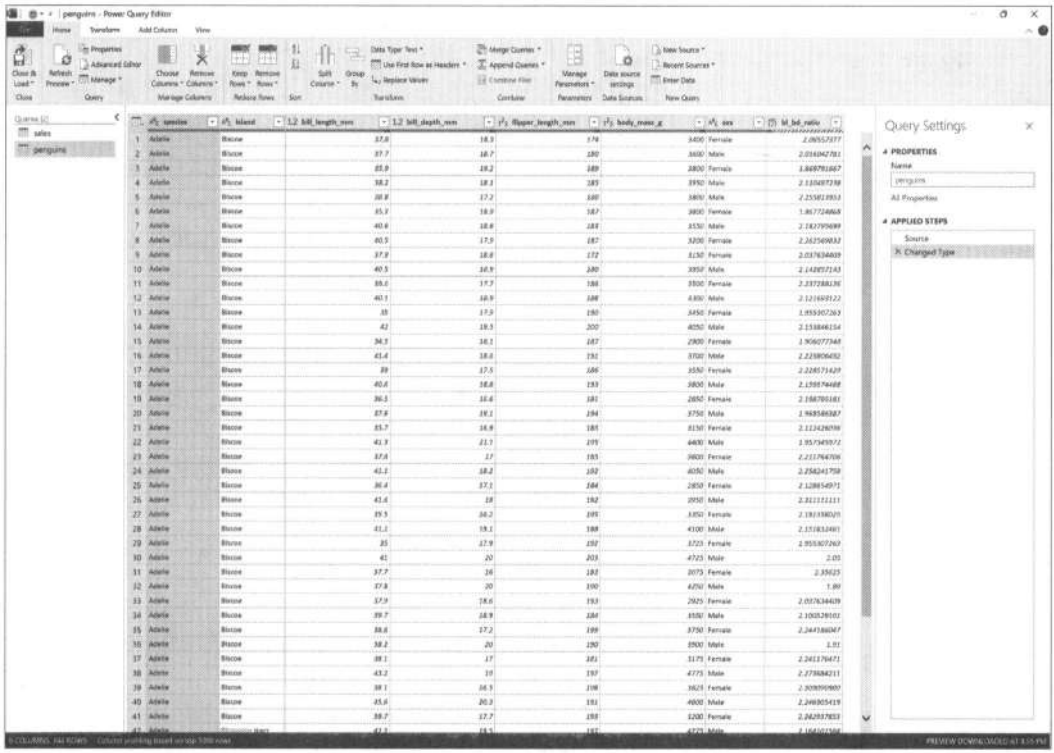


Рис. 2.4. Редактор Power Query

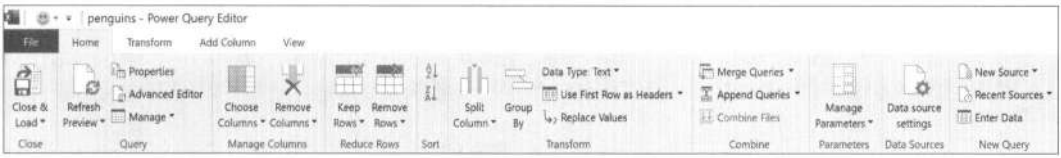


Рис. 2.5. Лента Power Query

- **Transform** (Преобразование).

  На этой вкладке находятся дополнительные команды для очистки и преобразования данных. В следующих главах у вас будет возможность познакомиться со многими из них.

- **Add Column** (Добавление столбца).

  Вкладка **Add Column** предназначена для создания новых столбцов различными способами. Ранее, в *разд. «Transform (Преобразование)»* вы уже применили команду с этой вкладки для добавления столбца индекса к данным. В *главе 4* мы будем использовать эту вкладку для создания вычисляемых столбцов.

- **View** (Просмотр).

  На этой вкладке можно настроить внешний вид редактора Power Query. Для начала установите в группе **Layout** (Структура) флажок **Formula Bar** (Строка

формул). Это действие добавит строку формул над вашим набором данных — такую же как в Excel (рис. 2.6).

Формула, отображаемая в строке формул Power Query, отличается от обычных функций Excel, поскольку она написана на языке программирования M, который был разработан специально для Power Query. По мере того как вы изменяете ваш запрос с помощью простого интерфейса редактора Power Query, соответствующим образом меняется и M-код. Это позволяет выполнять отладку, кастомизировать итоговый набор данных и делиться результатами.

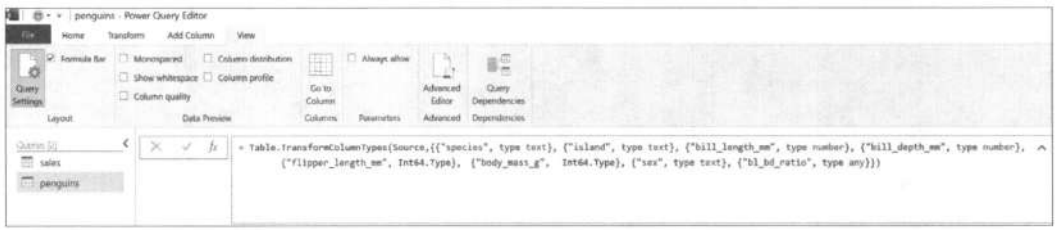


Рис. 2.6. Строка формул Power Query

Хотя наличие строки формул может означать, что для использования возможностей Power Query нужно хорошо знать сложный язык программирования, это необязательно. Большинство обычных задач можно успешно выполнять с помощью вкладки **Home** и разных опций, избегая написания M-кода. В этой книге мы сосредоточимся исключительно на таких примерах, поэтому строка формул будет нам не нужна.

В этой книге мы не будем использовать строку формул редактора Power Query. Вы можете скрыть ее, сняв флажок **Formula Bar** (Строка формул) в группе **Layout** (Структура) на вкладке **View** (Просмотр).

Если вы хотите освоить написание M-кода в Power Query, начните с изучения окна **Advanced Editor** (Расширенный редактор). Чтобы его открыть, перейдите в редакторе Power Query на вкладку **View** и в группе **Advanced** (Подробнее) выберите **Advanced Editor** (Расширенный редактор). В этом окне выводится код всего вашего запроса целиком.

Запросы

Переведите свое внимание с ленты на панель **Queries** (Запросы) в верхнем левом углу редактора. Здесь приведены импортированные источники данных, и между ними можно переключаться. Хотя сейчас оба наших источника находятся в одной рабочей книге, имейте в виду, что Power Query поддерживает разнообразные источники данных, в том числе и файлы формата CSV, базы данных, веб-страницы и многое другое.

Чтобы выполнить действия с каким-либо определенным запросом, щелкните правой кнопкой мыши на его названии (например, на `penguins`) — появится контекстное

меню с различными опциями (рис. 2.7). Среди этих опций вы увидите переименование запроса (**Rename**), удаление (**Delete**) и пр.

В Power Query имеется огромное количество опций, которые можно выполнить с помощью правой кнопки мыши, поэтому не стесняйтесь самостоятельно находить и запускать их из контекстного меню.

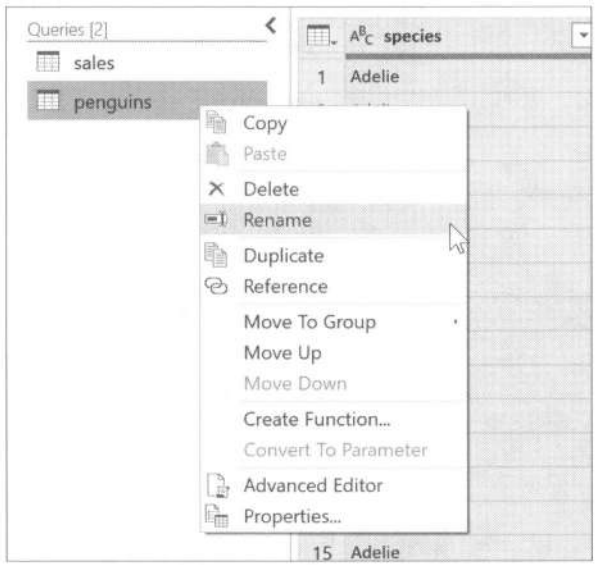


Рис. 2.7. Панель **Queries** в Power Query

Импортированные данные

Теперь давайте обратим внимание на область, занимающую бóльшую часть окна редактора — сами данные. В отличие от Excel, где вы можете свободно манипулировать данными, например скрывать столбцы или вставлять формулы, Power Query накладывает ограничения на такое редактирование.

Power Query тщательно отслеживает каждый ваш шаг и действия, выполняемые с запросом. Произвольное использование формул или скрытие столбцов запрещено. Все действия в среде Power Query выполняются программно.

Рассмотрим простую задачу по удалению столбца. Чтобы удалить столбец `island` из набора данных `penguins`, просто щелкните правой кнопкой мыши на заголовке столбца и выберите **Remove** (Удалить), как показано на рис. 2.8.

Столбец будет безвозвратно удален из набора данных... или нет? Чтобы понять, как происходит изменение данных в Power Query, посмотрите на список **Applied Steps** (Примененные шаги), расположенный справа от столбцов данных (рис. 2.9).

Power Query старательно записывает каждое ваше выполненное действие, включая удаления, в список **Applied Steps**, который отображается рядом с вашими данными. Это обеспечивает прозрачность и прослеживаемость всех операций.

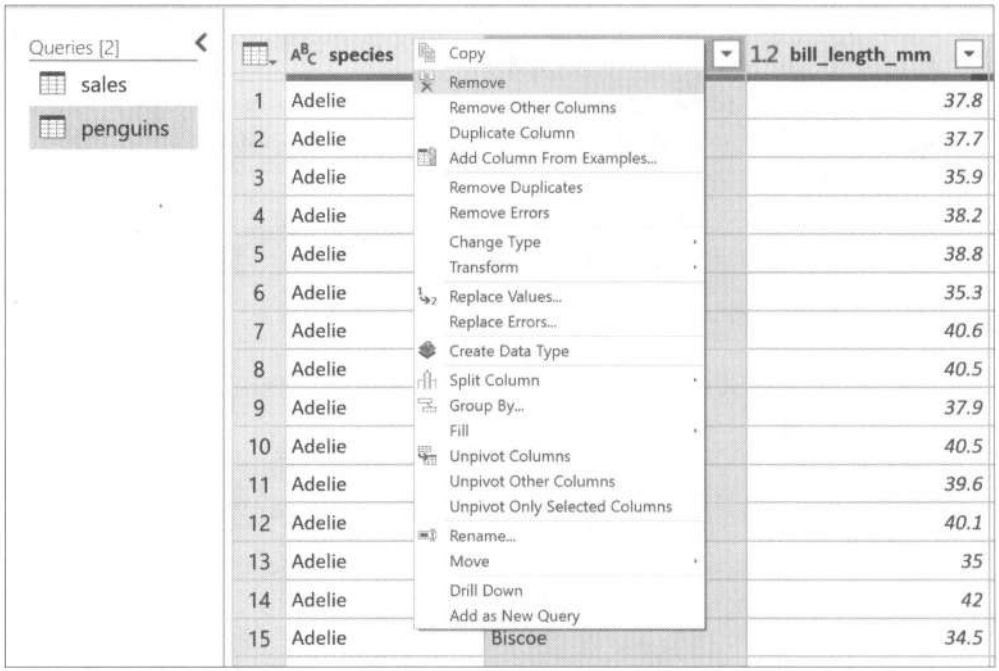


Рис. 2.8. Удаление столбца в Power Query

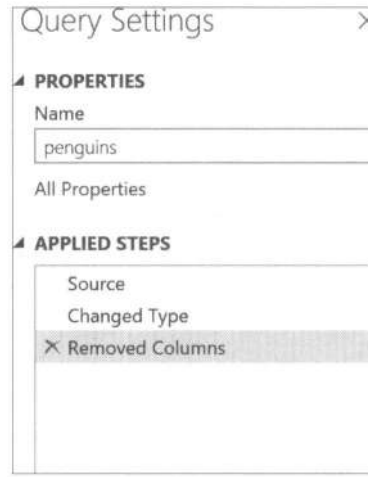


Рис. 2.9. Список **Applied Steps** в Power Query

Если быть более точным, это удаление столбца задокументировано в списке **Applied Steps** как наш третий шаг и называется **Removed Columns** (Удаленные столбцы). Первым шагом было подключение к данным, которое называется **Source** (Источник). Второй шаг — **Changed Type** (Измененный тип) — заключался в установке типов данных для таблицы. В отличие от Excel, Power Query требует, чтобы все значения в столбце были одного и того же типа. В этой книге мы будем в основном опираться на автоматическое определение типов данных в Power Query.

Подробнее о типах данных Power Query можно узнать из официальной документации Microsoft[1].

Выбрав любой шаг в списке **Applied Steps**, вы можете вновь открыть данные в том виде, в котором они отображались в конкретный момент времени. Например, если вы щелкните на **Changed Type** (шаг, предшествующий удалению столбца), столбец `island` снова появится в вашем редакторе.

Чтобы закрепить знания, полученные в этой главе, давайте при всё еще выделенном шаге **Changed Type** попробуем добавить в набор данных столбец индекса, начинающийся с 1. При этом появится предложение подтвердить, действительно ли вы хотите вставить в запрос промежуточный шаг (рис. 2.10).

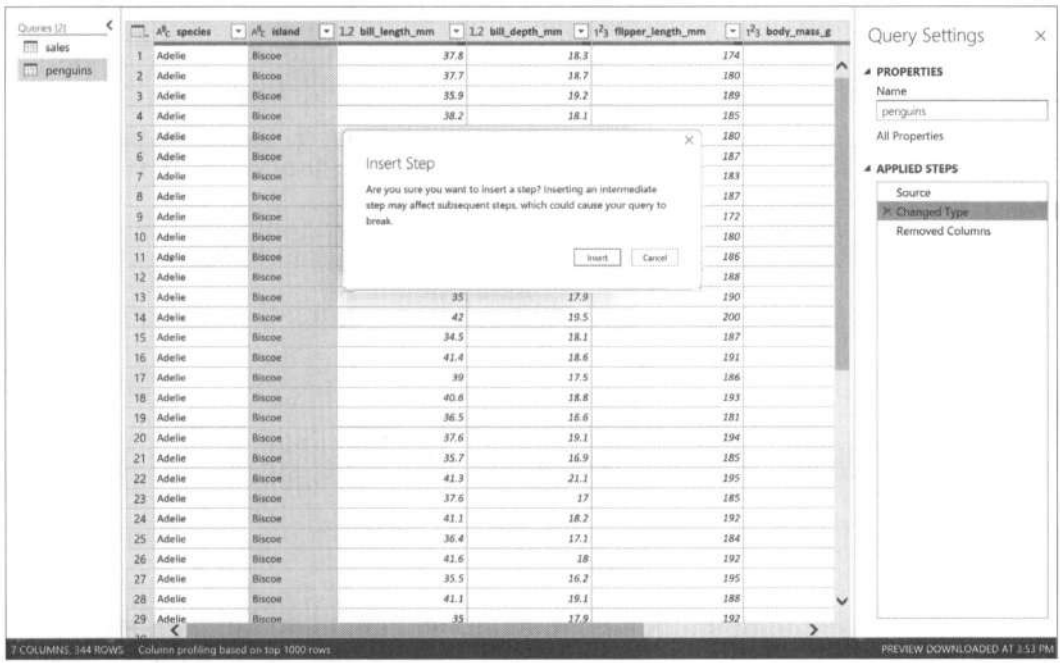


Рис. 2.10. Вставка промежуточного шага в Power Query

После нажатия кнопки **Insert** (Вставить) вы увидите, что в списке **Applied Steps** новый шаг **Added Index** (Добавлен индекс) встал перед **Removed Columns**, несмотря на то, что он был выполнен позднее. Это позволяет легко вносить изменения в запрос при изменении требований или добавлении новых шагов в рабочий процесс.

С шагами списка **Applied Steps** можно выполнять различные действия — например, удалять их или переименовывать. Предположим, вы хотите вернуть в запрос столбец `island`, удаленный ранее. Это можно сделать двумя способами: либо нажать на значок  , расположенный слева от шага **Removed Columns**, либо щелкнуть

[1] См. **https://clck.ru/3Jbtny**.

правой кнопкой мыши на этом же шаге, чтобы открылось контекстное меню, с помощью которого можно удалять и переименовывать шаги, менять порядок их следования, а также выполнять другие действия.

Хотя список **Applied Steps** и обеспечивает гибкость при работе в Power Query, но в нем отсутствует такая любимая всеми функциональность обычного Excel, как отмена действия. После удаления шага из списка нет возможности отменить это удаление. Поскольку большинство шагов можно легко воспроизвести, то при отсутствии кнопки отмены иногда проще повторить всю обработку вручную.

Выход из редактора Power Query

После создания в Power Query нужного запроса вы можете выйти из этого редактора и вернуться в обычную рабочую книгу Excel. Ранее в этой главе вы уже видели, что при простом выборе опции **Close & Load** (Закрыть и загрузить) на вкладке **Home** редактора Power Query результат запроса загрузится в таблицу Excel.

Есть и другие варианты загрузки, которые можно увидеть, нажав на раскрывающуюся кнопку рядом с **Close & Load**, а затем выбрав **Close & Load To** (Закрыть и загрузить в), — откроется диалоговое окно, показанное на рис. 2.11.

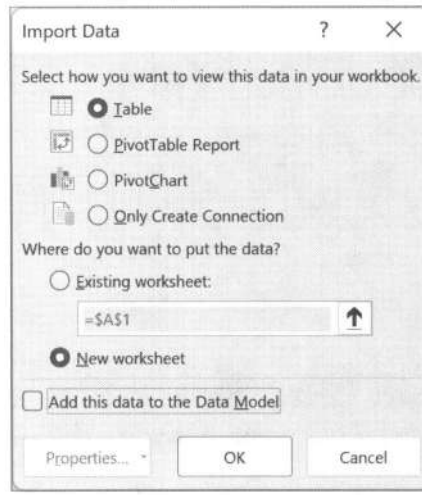


Рис. 2.11. Параметры загрузки из Power Query

Первое, что вы должны решить, — загрузить данные в таблицу, в сводную таблицу (здесь она называется **PivotTable Report**), в PivotChart или выбрать вариант **Only Create Connection** (Только создать подключение). При выборе последнего варианта результаты запроса не будут загружены в Excel, но сам запрос останется доступным в редакторе Power Query.

Если вы решили загрузить данные в рабочую книгу, вы можете поместить их на новый или уже существующий рабочий лист.

Кроме того, вы можете добавить данные в модель данных. Это позволит вам построить реляционную модель данных и использовать расширенные возможности

для создания отчетов в вашей рабочей книге. С моделями данных, Power Pivot и DAX вы познакомитесь в *части II*.

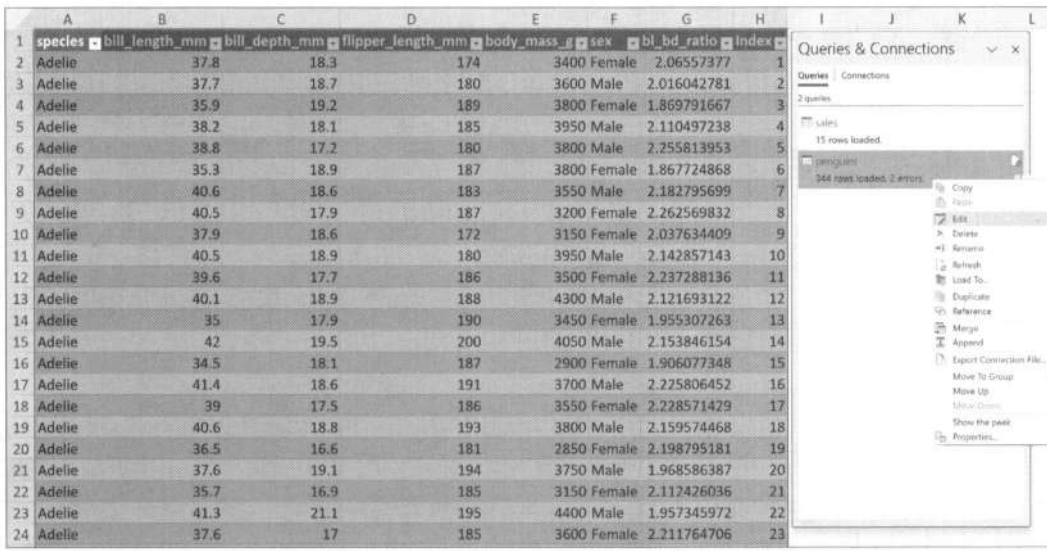
Выберите сейчас любой вариант и нажмите кнопку **OK**.

Возвращение в редактор Power Query

Чтобы вернуться к преобразованию данных в Power Query или изменить способ загрузки вашего запроса, перейдите на ленте Excel на вкладку **Data** (Данные), выберите **Queries & Connections** (Запросы и подключения) и найдите запрос `penguins` на панели, которая появилась в правой части рабочего окна.

Обратите внимание, что Power Query сообщает о двух ошибках в этом файле — мы найдем и исправим их в скором времени.

Щелкните правой кнопкой мыши на запросе `penguins`, и вы увидите меню с разными опциями, показывающими, что можно сделать с этим запросом (рис. 2.12). Чтобы изменить способ загрузки запроса из Power Query, выберите **Load To** (Загрузить в), а для возвращения в редактор Power Query — **Edit** (Изменить), как и показано на рис. 2.12.



	A	B	C	D	E	F	G	H
1	species	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	bl_bd_ratio	Index
2	Adelie	37.8	18.3	174	3400	Female	2.06557377	1
3	Adelie	37.7	18.7	180	3600	Male	2.016042781	2
4	Adelie	35.9	19.2	189	3800	Female	1.869791667	3
5	Adelie	38.2	18.1	185	3950	Male	2.110497238	4
6	Adelie	38.8	17.2	180	3800	Male	2.255813953	5
7	Adelie	35.3	18.9	187	3800	Female	1.867724868	6
8	Adelie	40.6	18.6	183	3550	Male	2.182795699	7
9	Adelie	40.5	17.9	187	3200	Female	2.262569832	8
10	Adelie	37.9	18.6	172	3150	Female	2.037634409	9
11	Adelie	40.5	18.9	180	3950	Male	2.142857143	10
12	Adelie	39.6	17.7	186	3500	Female	2.237288136	11
13	Adelie	40.1	18.9	188	4300	Male	2.121693122	12
14	Adelie	35	17.9	190	3450	Female	1.955307263	13
15	Adelie	42	19.5	200	4050	Male	2.153846154	14
16	Adelie	34.5	18.1	187	2900	Female	1.906077348	15
17	Adelie	41.4	18.6	191	3700	Male	2.225806452	16
18	Adelie	39	17.5	186	3550	Female	2.228571429	17
19	Adelie	40.6	18.8	193	3800	Male	2.159574468	18
20	Adelie	36.5	16.6	181	2850	Female	2.198795181	19
21	Adelie	37.6	19.1	194	3750	Male	1.968586387	20
22	Adelie	35.7	16.9	185	3150	Female	2.112426036	21
23	Adelie	41.3	21.1	195	4400	Male	1.957345972	22
24	Adelie	37.6	17	185	3600	Female	2.211764706	23

Рис. 2.12. Контекстное меню панели **Queries & Connections**

Профилирование данных в Power Query

До сих пор мы рассказывали о Power Query как о надежном инструменте ETL, предназначенном для оптимизации процесса очистки данных. Однако приступать к очистке набора данных без четкого понимания того, что именно его «загрязняет», нецелесообразно.

Для решения этой проблемы в Power Query реализован целый ряд методов профилирования данных. В этом разделе мы рассмотрим роль профилирования данных в Power Query и подчеркнем его важность для улучшения качества данных.

Что такое профилирование данных?

Профилирование данных позволяет получить представление о таких характеристиках данных, как пропущенные значения, частота и сводная статистика. Это помогает принимать обоснованные решения и эффективно преобразовывать данные. В процессе профилирования изучаются следующие вопросы:

- Насколько точны данные?
- Есть ли очевидные проблемы?
- Понятны ли назначение и единицы измерения каждой переменной в каждом наблюдении?
- Доступны ли все необходимые данные? Есть ли пропуски?
- Есть ли в данных из Excel ошибки в формулах, которые влияют на результаты?
- Корректно ли были внесены данные?

Ответы на эти вопросы позволяют аналитикам оценивать состояние и надежность данных, выявлять потенциальные проблемы и выбирать стратегии по очистке, преобразованию и анализу данных.

Опции предварительного просмотра данных

Профилирование данных в редакторе Power Query немного спрятано. Чтобы найти эти опции, перейдите на ленте на вкладку **View** (Просмотр) и найдите группу **Data Preview** (Предварительный просмотр данных), как показано на рис. 2.13. Рассмотрим функциональность каждого из этих пяти флажков, включая и выключая их поочередно.

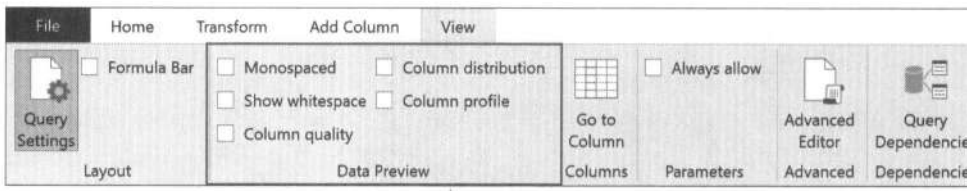


Рис. 2.13. Опции предварительного просмотра данных в Power Query

Monospaced и Show whitespace

Первые две опции изменяют отображение данных в редакторе Power Query:

- **Monospaced** (Моноширинный) — выводит данные с использованием моноширинного шрифта;
- **Show whitespace** (Показать пробелы) — показывает все ведущие и завершающие пробелы в данных.

Хотя эти опции могут вам пригодиться, особенно для нахождения текста, требующего обрезки, настоящие возможности профилирования данных заложены в остальных опциях.

Column quality и Column distribution

Теперь включите следующие два флажка: **Column quality** (Качество столбца) и **Column distribution** (Распределение столбцов). Над каждым столбцом появится поле, в котором будет указана полезная информация о данных, — например, процент допустимых, ошибочных и пустых значений. В нем также будет выведено распределение значений в столбце. Эти опции позволяют получить общее представление о качестве и распределении данных, что способствует эффективному анализу и принятию решений.

Что такое «допустимое» значение?

Когда Excel говорит о «допустимых» данных, это означает, что значение не пустое и не содержит ошибок. Важно отметить, что такое определение «допустимых» данных не учитывает правильности с точки зрения логики и содержательности данных. Из-за этого Power Query может счесть «допустимыми» даже бессмысленные значения. Например, значение столбца `sex` (пол) в строке 71 (рис. 2.14).

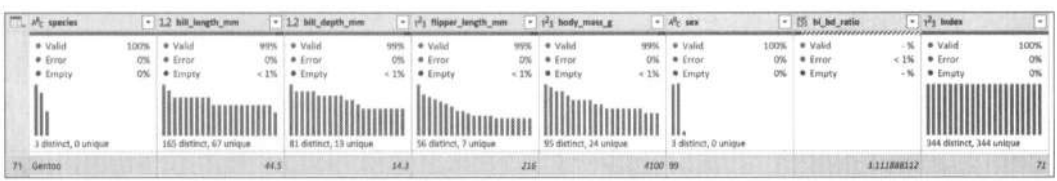


Рис. 2.14. «Допустимое» в Power Query значение `sex` в строке 71

Отсутствующие значения

Очевидно, что значение `99` не является допустимым для столбца `sex`. Скорее всего, ошибка возникла при внесении данных, когда вместо того, чтобы оставить их пустыми или `null`, в ячейки ошибочно записывали значение `99`. Увидеть действительно отсутствующие значения в Power Query можно в строке 296 (рис. 2.15).

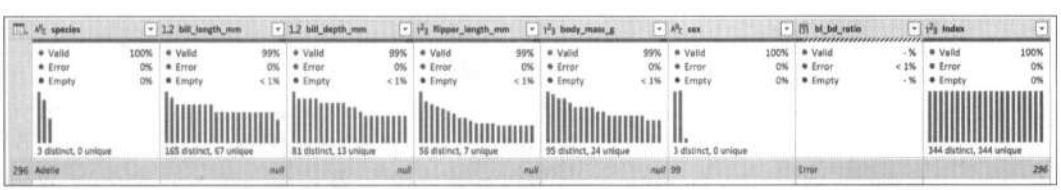


Рис. 2.15. Отсутствующие значения `null` в Power Query

В этой строке есть несколько значений, помеченных как `null`, что является корректным способом обозначения пустого значения в Power Query. Сейчас в поле с информацией о качестве столбцов в категории **Empty** (Пустой) для всех столбцов

указано менее 1% ячеек. Однако имейте в виду, что эти цифры не учитывают неправильно внесенное отсутствие значений.

Ошибки в ячейках

Чтобы понять, что относится к категории **Error** (Ошибка), посмотрите опять на строку 296 и обратите внимание на столбец `bl_bd_ratio`. Этот столбец был вычислен в Excel путем деления значений в столбце `bill_length_mm` на значения в столбце `bill_depth_mm`. Однако в этой конкретной строке знаменатель в формуле оказался пустым, что и привело к ошибке. Щелкнув по пустому полю ячейки с `Error`, вы можете убедиться, что причиной является ошибка `#DIV/0` (рис. 2.16).

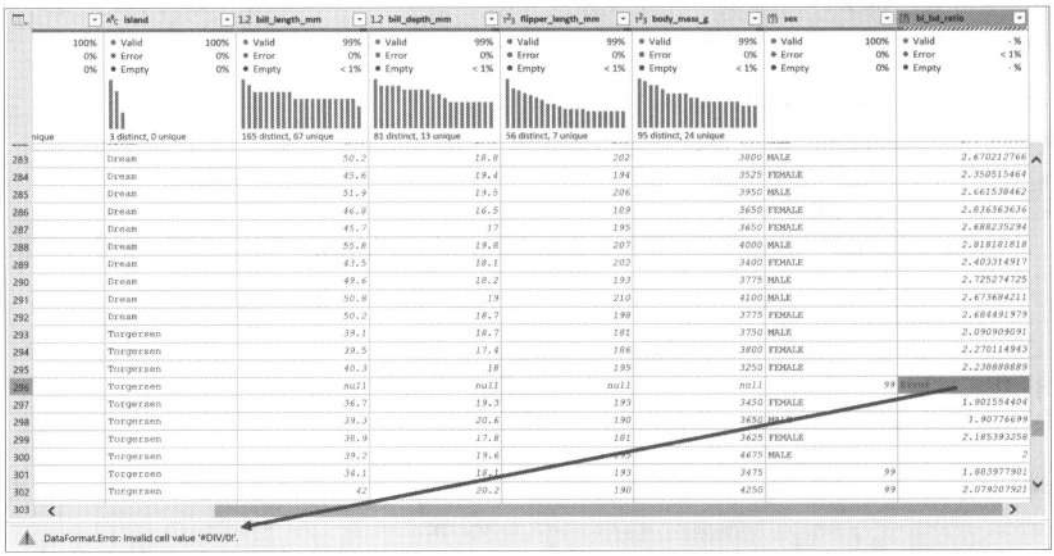


Рис. 2.16. Ошибка в ячейке при профилировании данных в Power Query

В этом столбце есть две такие ошибки вычислений, которые отображаются в ячейках в виде текста `Error`, как показано на рис. 2.16. Эти ошибки можно устранить разными способами — например, переписать оригинальную формулу Excel, чтобы уйти от деления на ноль или исключить из выборки строки, в которых возникают ошибки. О том, как фильтровать строки и выполнять другие действия со строками, вы узнаете в *главе 3*.

* * *

Вернемся к изучению группы опций предварительного просмотра данных **Column quality** и **Column distribution**. Флажок **Column distribution** (Распределение столбцов) выводит для каждого столбца небольшое изображение с распределением данных, а также некоторую полезную информацию (рис. 2.17).

Однако все эти характеристики можно сразу вывести на экран с помощью флажка **Column profile** (Профиль столбца), который мы рассмотрим далее.

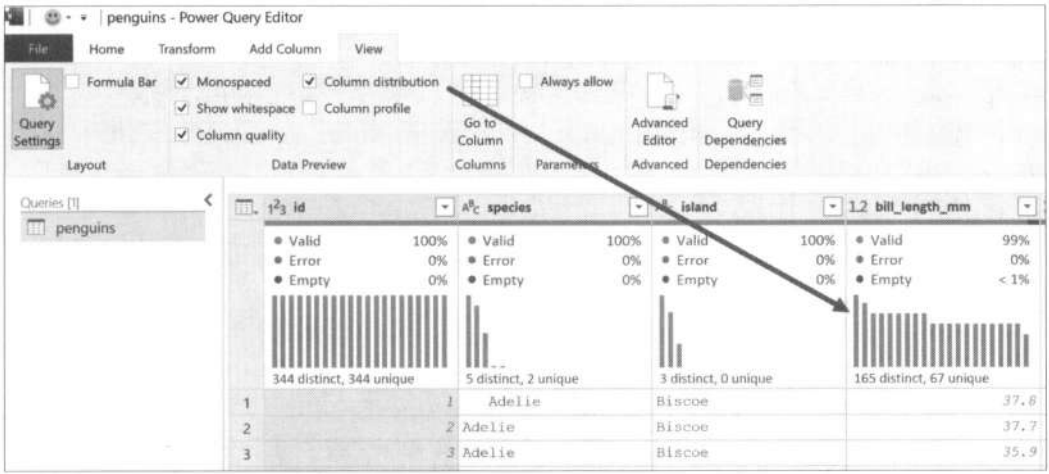


Рис. 2.17. Распределение столбцов в Power Query

Column profile (Профиль столбца)

И наконец, установите флажок **Column profile** (Профиль столбца) и выделите какой-то определенный столбец. Под набором данных откроется панель с подробной информацией об этом столбце. Давайте для примера рассмотрим столбец `species`, который представляет собой качественную переменную. При включенном флажке

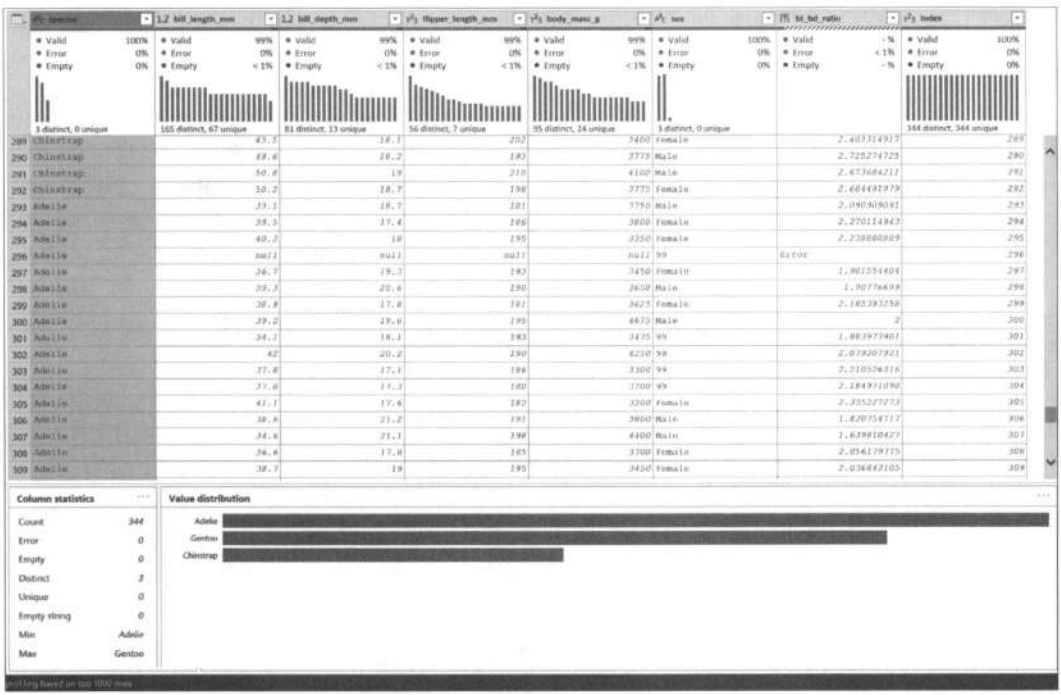


Рис. 2.18. Профилирование качественной переменной в Power Query

будет выведен подробный разбор значений в этом столбце, включая изображение с частотой наблюдения каждого значения (рис. 2.18).

Для количественных переменных — например, `bill_depth_mm`, в статистике по столбцу будут выводиться дополнительные показатели, такие как среднее значение и стандартное отклонение (рис. 2.19).

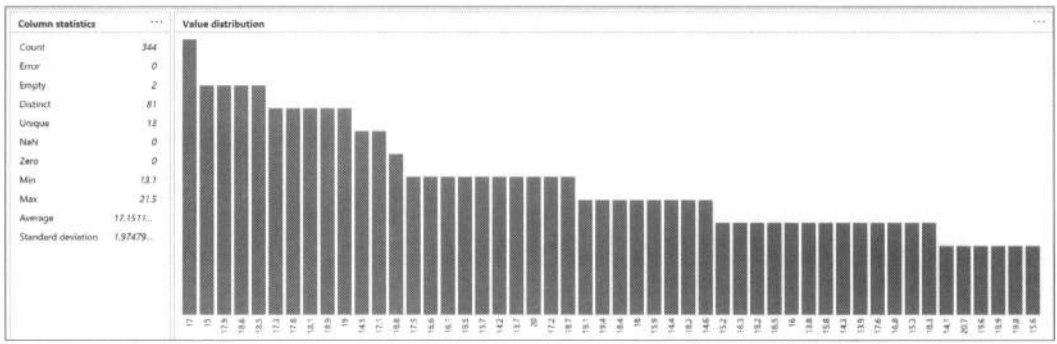


Рис. 2.19. Профилирование количественной переменной в Power Query

Если вы не знакомы с понятиями качественных и количественных переменных, считайте, что *качественные переменные* отвечают на вопрос «Какого вида?», а *количественные переменные* — на вопросы «Сколько?» или «Какое количество?». Более подробно об этих типах переменных рассказано в моей книге «Погружение в аналитику данных: от Excel к Python и R»[4]. А в *части II* книги, которую вы сейчас читаете, вы узнаете об измерениях и мерах — понятиях, схожих с качественными и количественными переменными.

Как убрать ограничение на тысячу строк?

Если вы работаете в Power Query с набором данных, содержащим более тысячи строк, обязательно включите все строки в профилирование данных. Для этого в нижней части редактора щелкните на строке состояния и выберите **Column profiling based on entire data set** (Профилирование столбца на основе всего набора данных), как показано на рис. 2.20.

Итак, благодаря возможностям Power Query по профилированию данных вы смогли:

- быстро обнаружить неправильно отформатированные ячейки;
- определить, в каких столбцах отсутствуют значения;
- визуализировать распределение каждой переменной.

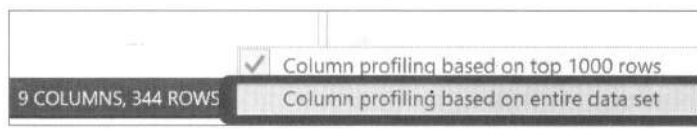


Рис. 2.20. Снятие ограничения в тысячу строк для профилирования данных

[4] См. **https://clck.ru/3JVY4U**.

Окончание профилирования данных

Закончив профилирование данных в Power Query, вы можете просто нажать **Close & Load**, чтобы вернуться в Excel без внесения каких-либо изменений в запрос. Помните, что профилирование данных — это предварительная проверка данных *до* внесения в них какие-либо изменений. В следующих главах книги мы не будем включать эти опции предварительного просмотра данных, чтобы избежать избыточной визуальной информации.

Заключение

В этой главе, опираясь на такой инструмент ETL, как Power Query, мы развеяли распространенные мифы, связанные с Excel. Мы также рассмотрели редактор Power Query и освоили процесс профилирования данных. Теперь вы готовы приступить к преобразованию данных с помощью Power Query, чему и будут посвящены остальные главы *части I*.

Упражнения

В упражнениях к этой главе вам нужно будет исследовать набор данных с ценами на компьютеры с помощью Power Query.

Откройте файл ch_02_exercises.xlsx, расположенный в папке exercises\ch_02_exercises сопроводительного репозитория к этой книге[5]. Выполните следующее:

1. Загрузите данные в Power Query как таблицу. Назовите запрос `computers`.
2. Добавьте к данным столбец индекса, начинающийся с 1.
3. Переименуйте предыдущий шаг в списке **Applied Steps** в **Added unique identifier** (Добавление уникального идентификатора).
4. Перетащите столбец `Index` так, чтобы он был первым столбцом в наборе данных.
5. Используйте возможности Power Query для профилирования данных, чтобы ответить на следующие вопросы. Обязательно настройте профилирование столбцов по всему набору данных.
   - какой диапазон цен (`price`) на компьютеры в этом наборе данных?
   - какой средний объем оперативной памяти (`ram`) в наборе данных?
   - есть ли в этом наборе данных отсутствующие значения? Если да, то где именно?
6. Загрузите результаты запроса в сводную таблицу в Excel.

Готовое решение можно посмотреть в файле ch_02_exercise_solutions.xlsx, расположенном в той же папке репозитория.

[5] См. **https://clck.ru/3JhRSp**.

Преобразование строк в Power Query

В *главе 2* вы познакомились с возможностями использования Power Query в качестве инструмента ETL для Excel. В этой и следующих главах *части I* вы сможете попрактиковаться в решении распространенных задач по преобразованию данных. Основное внимание в этой главе уделено строкам.

Очистка данных часто включает в себя задачи по обработке строк — такие как сортировка, фильтрация и удаление дубликатов. В классическом Excel есть способы решать такие задачи с помощью интерфейса, но они могут показаться громоздкими и трудновоспроизводимыми. Power Query предлагает свое решение, позволяющее выполнять проверяемую и повторяемую очистку данных без написания кода. Чтобы работать с примерами этой главы, откройте из папки ch_03 сопроводительного репозитория к этой книге файл ch_03.xlsx[1].

На рабочем листе `signups` этой рабочей книги группа по планированию вечеринок в вашей организации записывает участвующих и хочет, чтобы итоговый список был отсортирован по алфавиту с исключенными дубликатами, пробелами и опечатками. Коллегам надоело вручную сортировать и удалять ненужные строки при добавлении новых данных. Им нужна рабочая книга, которую можно было бы легко обновлять и использовать повторно при внесении новых участников или планировании новых вечеринок.

Загрузите эти данные в Power Query, назвав запрос `signups`. Импортируйте все заполненные строки из столбца `A` и не забудьте указать, что ваша таблица включает заголовки.

Удаление пропущенных значений

Как уже упоминалось в *главе 2*, в Power Query есть специальное значение `null` для обозначения отсутствующих значений. Набор данных `signups` содержит три пустых значения, что может привести к путанице. Чтобы убрать их, перейдите на ленте на вкладку **Home** и выберите **Remove Rows | Remove Blank Rows** (Удалить строки | Удалить пустые строки), как показано на рис. 3.1.

Теперь давайте отсортируем список в алфавитном порядке. Для этого рядом со столбцом `Sign-up` нажмите на стрелочку с выпадающим меню, содержащим опции для сортировки и фильтрации, похожие на обычный Excel (рис. 3.2). Чтобы отсор-

[1] См. **https://clck.ru/3JhUSV**.

тировать список по алфавиту, выберите **Sort Ascending** (Сортировка по возрастанию).

Вы могли заметить, что `Phyllis` в этом наборе данных встречается несколько раз. Чтобы удалить дубликаты, вернитесь на вкладку **Home** и выберите **Remove Rows | Remove Duplicates** (Удалить строки | Удалить дубликаты).

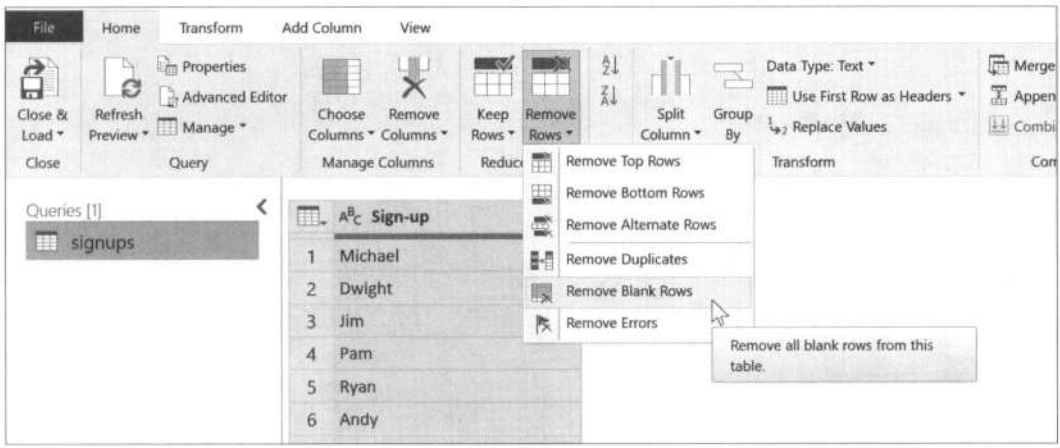


Рис. 3.1. Удаление пустых строк в Power Query

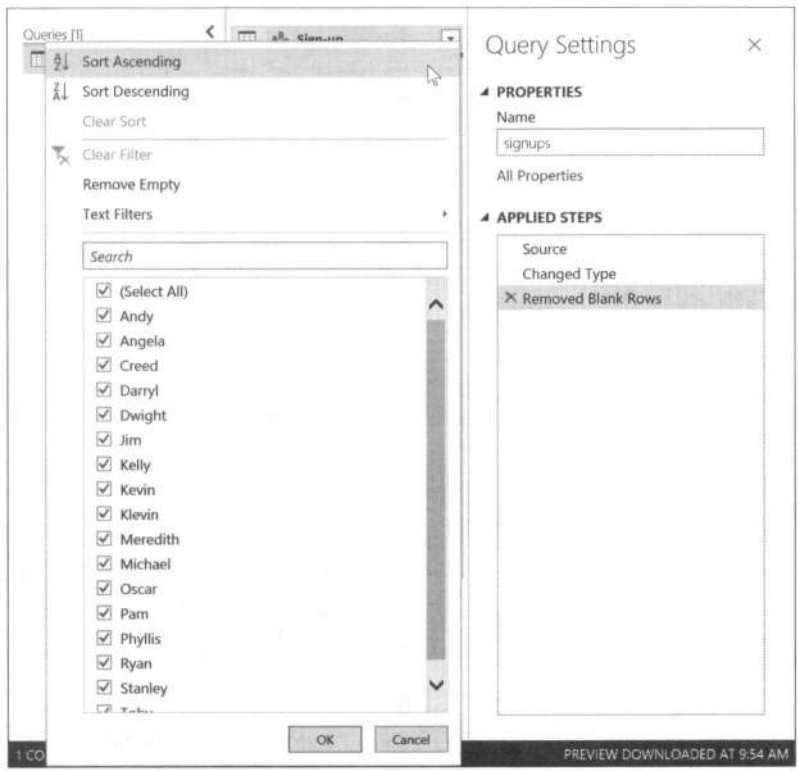


Рис. 3.2. Сортировка строк в Power Query

Данные в списке в основном чистые, за исключением опечатки в имени `Klevin` в строке 9. Эту ошибку нельзя обнаружить с помощью простого удаления пробелов или дубликатов, что подчеркивает важность знания предметной области при работе с данными. Power Query помогает выполнять стандартные задачи по очистке данных, но некоторые случаи требуют более глубокого понимания данных. И наш последний шаг — исключение опечатки из набора данных (рис. 3.3).

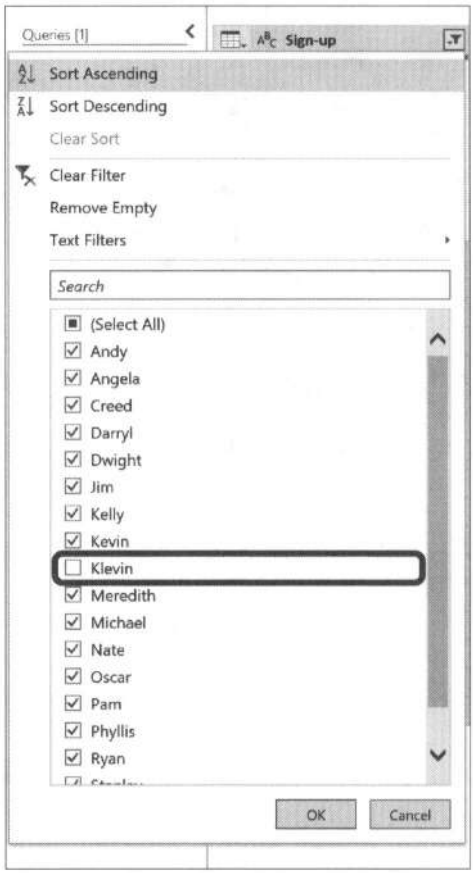


Рис. 3.3. Исключение опечатки в Power Query

Обновление запроса

Вы проделали отличную работу по очистке данных о планирующейся вечеринке! Чтобы сделать результаты общедоступными, загрузите очищенный набор данных в Excel, для чего на вкладке **Home** выберите **Close & Load**.

Организация исходных и преобразованных рабочих листов

Загрузка преобразованных данных в ту же рабочую книгу, где находятся и исходные данные, имеет существенный недостаток: их становится сложно отличить друг от друга. Хотя Excel пытается помочь с этим, раскрашивая исходную таблицу по умолчанию в синий цвет, а пре-

образованную — в зеленый, всё равно может возникнуть путаница из-за схожих имен рабочих листов. Чтобы избежать этой путаницы, можно всегда добавлять суффикс «-out» к названию выходного рабочего листа, загруженного из Power Query, или придерживаться другого аналогичного соглашения об именовании.

Power Query предлагает вам больше, чем простое избавление от утомительных и чреватых ошибками процессов очистки данных в Excel. Настоящее преимущество от его использования заключается в возможности обновлять данные одним щелчком мыши. Чтобы увидеть, как это работает, добавьте две строки в исходную таблицу `signups`. Например, я вставлю одну пустую строку и добавлю `Nate`.

Чтобы повторно выполнить ваш запрос, перейдите к выходной таблице Power Query, щелкните на ней правой кнопкой мыши и выберите **Refresh** (Обновить), как показано на рис. 3.4.

Рис. 3.4. Обновление результатов Power Query в Excel

Все те же шаги будут применены к измененным данным, и таблица автоматически обновится. Теперь наша рабочая книга содержит воспроизводимый в один щелчок процесс очистки данных, который можно будет использовать для любых будущих списков участников вечеринок.

Разделение данных на строки

Приходилось ли вам сталкиваться с ситуацией, когда в Excel у вас есть список элементов, разделенных запятыми, и вам нужно разнести их по отдельным ячейкам? Рассмотрим пример, показанный на рис. 3.5. Эти данные находятся на рабочем листе `roster` в файле ch_03.xlsx.

	A	B
1	Department	Signups
2	Sales	Jim, Dwight, Phyllis, Andy
3	Accounting	Kevin, Oscar
4	Warehouse	Daryl, Nate
5	Annex	Toby, Ryan, Kelly
6		

Рис. 3.5. Пример списков имен

В этом наборе данных приведены имена участников проекта, сгруппированные по отделам. Наша задача — удобно отсортировать и отфильтровать эти данные по имени и отделу. В классическом Excel можно было бы попробовать использовать функцию **Text to Columns** (Текст по столбцам), что привело бы к запутанному и неудовлетворительному результату (рис. 3.6).

	A	B	C	D	E
1	Department	Signups	Column1	Column2	Column3
2	Sales	Jim	Dwight	Phyllis	Andy
3	Accounting	Kevin	Oscar		
4	Warehouse	Daryl	Nate		
5	Annex	Toby	Ryan	Kelly	
6					

Рис. 3.6. Имена участников разделены по столбцам с помощью функции **Text to Columns**

В Power Query есть удобное решение для разделения таких данных, позволяющее получить требуемый нам результат.

Сначала импортируйте данные с рабочего листа `roster` в Power Query и назовите запрос `roster`. Выделите столбец `Signups` в наборе данных. В редакторе Power Query перейдите на вкладку **Home**, выберите опцию **Split Column** (Разделить столбец) и из раскрывающегося меню — пункт **By Delimiter** (По разделителю), как показано на рис. 3.7.

Термин *разделитель* (delimiter) означает символ, отделяющий элементы данных друг от друга. В нашем случае разделителем является запятая, и, скорее всего, Power Query определит это автоматически. Если разделитель определится неправильно, выберите запятую (**Comma**) в выпадающем списке. Затем щелкните на **Advanced options** (Расширенные параметры). В этой секции вы найдете скрытую опцию для разделения текста не на столбцы, а на строки (рис. 3.8), — выберите **Rows** (Строки) и нажмите **OK**.

Рис. 3.7. Разделение столбца по разделителю в Power Query

Рис. 3.8. Разделение текста на строки

Этот запрос почти готов к загрузке, но осталось выполнить последнее действие. Для этого перейдите в ленте редактора Power Query на вкладку **View** (Вид) и в группе **Data Preview** (Предварительный просмотр данных) убедитесь, что установлен флажок **Show whitespace** (Показать пробелы). Чтобы освежить в памяти опции Power Query для предварительного просмотра и профилирования данных, вернитесь к *главе 2*.

Теперь в столбце `Signups` стали видны лишние пробелы, оставшиеся после разделения списка по запятым. Чтобы удалить их, щелкните правой кнопкой мыши на

заголовке столбца и выберите **Transform | Trim** (Преобразование | Усечь), как показано на рис. 3.9.

Теперь вы можете нажать **Close & Load**, чтобы загрузить результаты в таблицу (рис. 3.10).

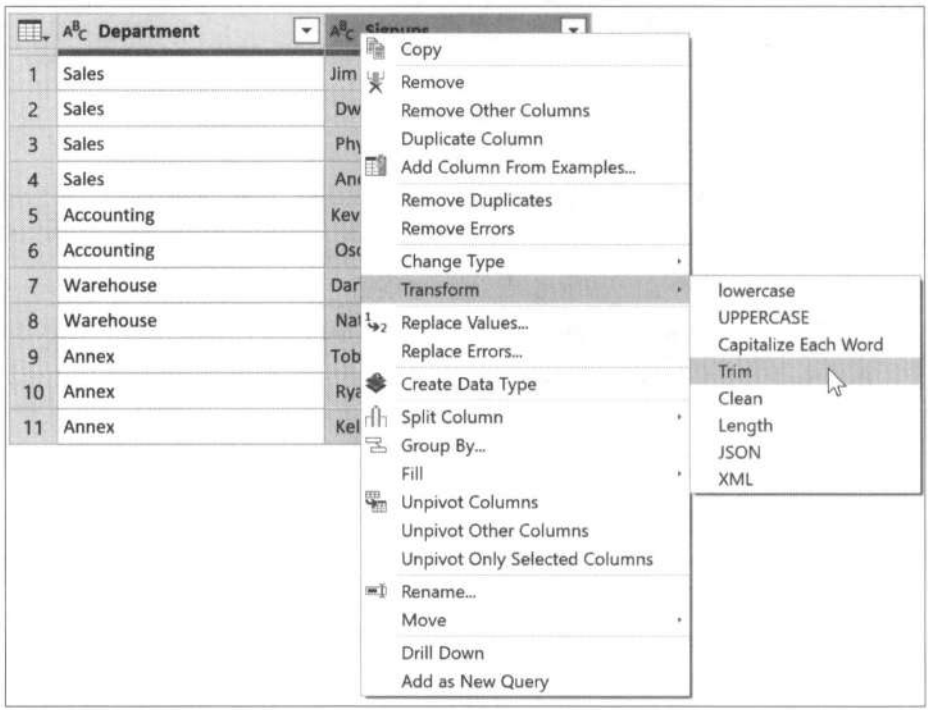


Рис. 3.9. Удаление лишних пробелов в Power Query

	A	B
1	Department	Signups
2	Sales	Jim
3	Sales	Dwight
4	Sales	Phyllis
5	Sales	Andy
6	Accounting	Kevin
7	Accounting	Oscar
8	Warehouse	Daryl
9	Warehouse	Nate
10	Annex	Toby
11	Annex	Ryan
12	Annex	Kelly
13		

Рис. 3.10. Данные разделены на строки

Заполнение заголовков и пустых ячеек

Вы можете столкнуться с ситуацией, когда некоторые фрагменты набора данных ошибочно помечены как `null` или по какой-либо причине вовсе отсутствуют. Это может быть связано с проблемами форматирования во внешней системе или неправильными способами хранения данных.

В этом разделе мы разберемся, как в Power Query можно исправить и отсутствующие заголовки, и пропущенные значения. Для следующих примеров будет использоваться рабочий лист `sales` из файла ch_03.xlsx.

Замена заголовков столбцов

Выгрузки из *систем управления предприятием* (ERP, Enterprise Resource Planning) часто содержат дополнительную строку, заполненную нерелевантной информацией. В нашем наборе данных первая строка в каждом столбце заполнена символами `####`, а правильные заголовки столбцов находятся в строке 2. Вместо того чтобы каждую неделю вручную исправлять это, удаляя ненужную строку, можно автоматизировать очистку с помощью Power Query.

После загрузки данных в Power Query и переименования запроса в `sales` перейдите на вкладку **Home** и в группе **Transform** (Преобразование) выберите **Use First Row as Headers** (Использовать первую строку в качестве заголовков), как показано на рис. 3.11.

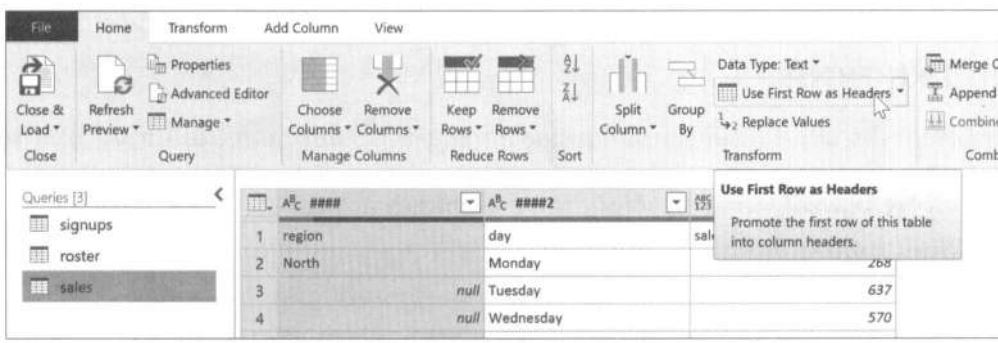


Рис. 3.11. Использование первой строки в качестве заголовков в Power Query

Заполнение пропущенных значений

Теперь, когда заголовки столбцов исправлены, нужно решить проблему с ошибочно пустыми ячейками. Похоже, что ERP-система не умеет повторять значение `region` на каждой строке в рамках одной недели, что может привести к трудностям при использовании сводных таблиц или других методов анализа данных. Чтобы это исправить, выделите столбец `region`, перейдите на вкладку **Transform** и в группе **Any Column** (Любой столбец) выберите **Fill | Down** (Заполнить | Вниз), как показано на рис. 3.12.

Теперь данные очищены. И вы можете «закрыть и загрузить» результаты в Excel.

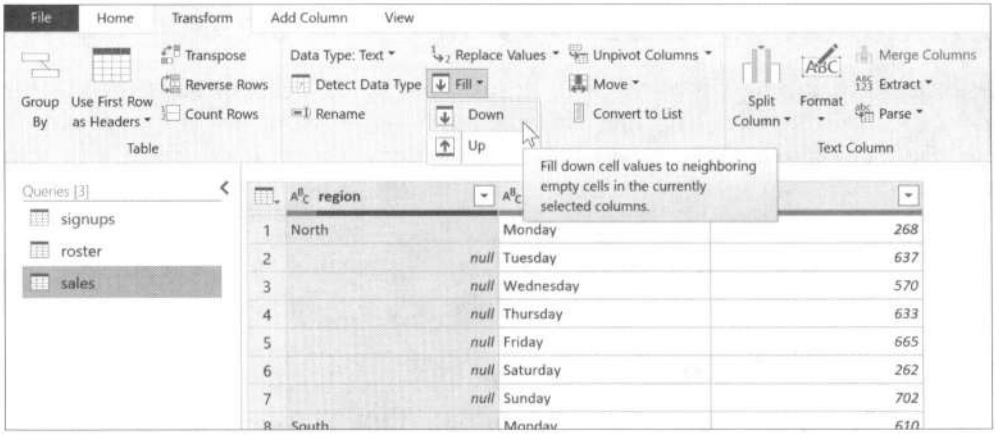


Рис. 3.12. Заполнение пропущенных значений

Заключение

Power Query отлично подходит в качестве инструмента для очистки строк данных, предлагая простые и эффективные способы выполнения таких задач, как сортировка, фильтрация, удаление дубликатов и исправление отсутствующих значений. В *главе 4* мы расширим этот перечень задач, уделив особое внимание преобразованию столбцов.

Упражнения

Откройте файл ch_03_exercises.xlsx, расположенный в папке exercises\ch_03_exercises сопроводительного репозитория к этой книге[2]. Он содержит два рабочих листа. Используйте Power Query для обработки и анализа данных.

На рабочем листе `states`:

1. Удалите из данных строку `United States`.
2. Заполните пробелы в столбцах `region` и `division`.
3. Отсортируйте по `population` от высокого значения к низкому.
4. Загрузите результаты в сводную таблицу.

Для рабочего листа `midwest_cities` загрузите данные в таблицу, чтобы каждый город (`Cities`) располагался на отдельной строке.

Вы можете найти вариант решения в файле ch_03_exercise_solutions.xlsx, расположенном в той же папке репозитория.

[2] См. **https://clck.ru/3JhZMt**.

Преобразование столбцов в Power Query

Глава 3 была посвящена работе со строками, а в этой главе основное внимание мы уделим столбцам и рассмотрим различные действия со столбцами, такие как изменение регистра текста, переформатирование столбцов, создание вычисляемых столбцов и многое другое. Чтобы работать с примерами этой главы, откройте из папки ch_04 сопроводительного репозитория к этой книге файл ch_04.xlsx[1] и загрузите в Power Query таблицу `rentals`.

Изменение регистра столбца

Power Query упрощает для нас преобразование текстов в столбцах в нижний и верхний регистры, а также в регистр, в котором каждое слово пишется с заглавной буквы. Чтобы проверить эту функциональность, удерживая клавишу <Ctrl>, выделите сразу два столбца: `Title` и `Artist Name`. Затем щелкните правой кнопкой мыши на заголовке любого из этих столбцов и выберите **Transform | Capitalize Each Word** (Преобразование | Каждое Слово С Прописной), как показано на рис. 4.1.

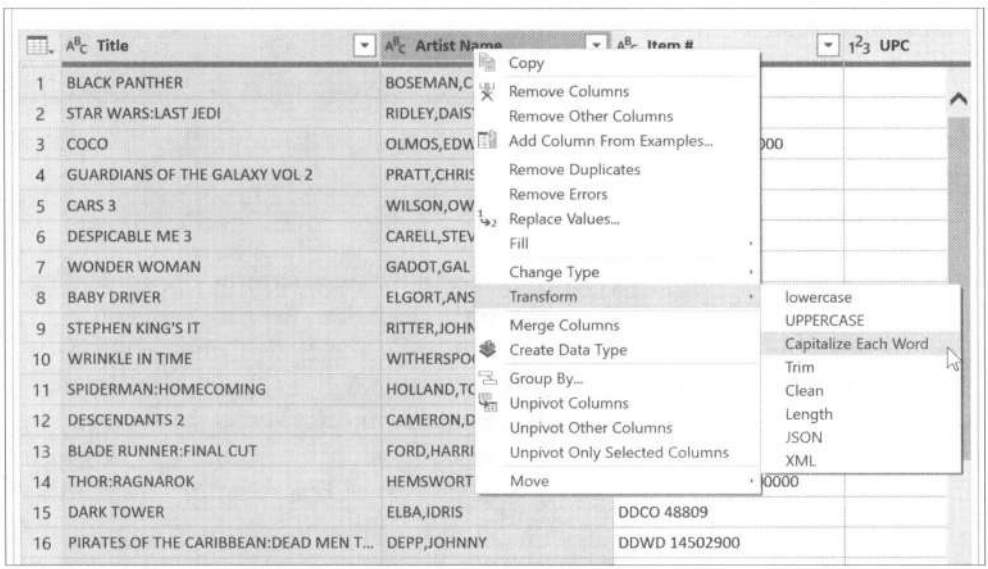


Рис. 4.1. Изменение регистра текста в Power Query

[1] См. **https://clck.ru/3Jhc3b**.

Обратите внимание, что в столбцах `Title` и `Artist Name` отсутствуют пробелы после двоеточий и запятых. Чтобы исправить это, выделите оба столбца, щелкните правой кнопкой мыши на любом из них, выберите **Replace Values** (Замена значений) и в открывшемся диалоговом окне **Replace Values** укажите в качестве значения для поиска двоеточие, а в качестве значения для замены — двоеточие с пробелом (рис. 4.2).

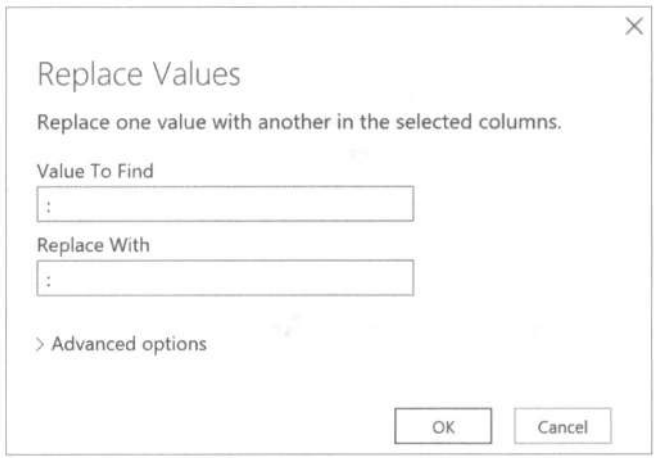


Рис. 4.2. Замена значений в Power Query

После этого выполните то же самое и для запятых: замените запятую на запятую, за которой следует пробел.

Как уже отмечалось в *главе 2*, Power Query записывает каждое ваше действие с данными в список **Applied Steps** (Примененные шаги). Это существенно облегчает проверку изменений, выполненных над текстом, по сравнению с обычным действием с поиском и заменой в Excel.

Разделение на столбцы

В *главе 3* вы узнали, как разделить текст с запятыми на строки. Теперь пришло время сделать то же самое со столбцами. Щелкните правой кнопкой мыши на столбце `Item #` и разделите его на два, выбрав опцию **Split Column | By Delimiter** (Разделить столбец | По разделителю). В открывшемся диалоговом окне выберите **Space** (Пробел) из выпадающего списка и нажмите **OK**. И еще раз подчеркну, что эта опция удобнее в использовании и предоставляет более широкие возможности по сравнению с обычной функцией **Text to Columns** (Текст по столбцам) в Excel.

Получившиеся разделенные столбцы по умолчанию называются `Item #.1` и `Item #.2`. Чтобы переименовать их, просто выполните на заголовках столбцов двойной щелчок мышью. Как и все действия в Power Query, эти преобразования записываются в **Applied Steps**, что позволяет при необходимости легко отменить или скорректировать их.

Изменение типов данных

В Power Query каждому столбцу присваивается конкретный тип данных, что сразу ограничивает действия, которые можно выполнять с этим столбцом. При импорте набора данных Power Query автоматически пытается для каждого столбца определить наиболее подходящий тип данных. Однако бывают ситуации, когда это автоматическое определение типов нужно уточнить или подкорректировать.

Рассмотрим, например, столбец `UPC`. По умолчанию ему присвоен тип данных **Whole Number** (Целое число). Но так как мы не предполагаем выполнять над данными этого столбца какие-либо математические операции, их лучше хранить в виде текста. Для этого щелкните на значке 1^2_3 рядом с именем столбца `UPC` и измените его тип данных на **Text** (Текст), как показано на рис. 4.3. Выполните также следующие изменения типов данных:

- столбец `ISBN 13` — в **Text**;
- столбец `Retail` — в **Currency** (Валюта).

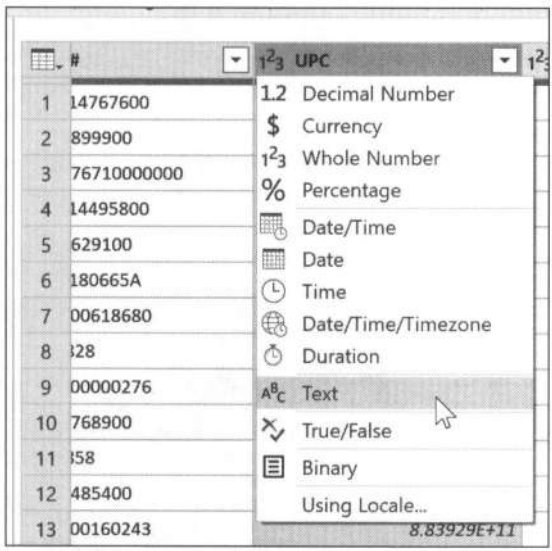


Рис. 4.3. Изменение типов данных столбцов в Power Query

Удаление столбцов

Удаление ненужных столбцов из набора данных упрощает его обработку и анализ. Выделите столбец `BTkey` и удалите его из запроса, нажав клавишу <Delete>. Если позднее вы решите все-таки оставить этот столбец, то его можно легко вернуть с помощью списка **Applied Steps**, как уже рассказывалось в *главе 2*.

Работа с датами

Power Query предлагает широкий набор сложных функций для преобразования и форматирования дат. Он также упрощает изменение типов для столбцов с датами, позволяет пользователям извлекать такие составляющие даты, как номер месяца или день, и указывать для них более подходящие типы данных.

Чтобы освоить эту функциональность, воспользуемся столбцом `Release Date` и применим к нему несколько разных опций. Для начала создайте копию этого столбца: щелкните правой кнопкой мыши на столбце и выберите **Duplicate Column** (Создать дубликат столбца). Повторите это действие еще два раза, чтобы у вас появились четыре одинаковых столбца с датами.

Щелкните правой кнопкой мыши на первой копии `Release Date` и выберите в меню **Transform | Year | Year** (Преобразование | Год | Год), как показано на рис. 4.4. Столбец будет переформатирован, и его тип соответствующим образом изменится для отображения только года, а не полной даты.

Рис. 4.4. Преобразование столбца с датами в Power Query

Извлеките из даты номер месяца и день для следующих двух столбцов, для чего выполните на заголовках столбцов двойной щелчок мышью и переименуйте их соответственно в `Year`, `Month` и `Day`, чтобы правильно интерпретировать переформатированные данные. Закройте и загрузите результаты в таблицу Excel.

Мы успешно выполнили несколько преобразований со столбцами данных в Power Query. И теперь можно загрузить этот запрос в Excel.

Создание пользовательских столбцов

Добавление *вычисляемого столбца* — обычная задача при очистке данных, будь то размер прибыли, временной период или что-либо еще, Power Query выполняет эту операцию с помощью языка программирования M.

Для следующего примера перейдите в файле ch_04.xlsx на рабочий лист `teams`. Включенный в него набор данных содержит сезонную статистику для команд из Главной лиги бейсбола (Major League Baseball, MLB) начиная с 2000 года. Наша задача — создать новый столбец, в котором будет вычисляться процент выигрыша для каждой команды в сезоне. Он вычисляется путем деления количества побед (`W`) на суммарное количество побед (`W`) и поражений (`L`), вместе взятых.

Сначала, конечно, надо загрузить данные в Power Query. Затем на ленте редактора выберите опцию **Add Column | Custom Column** (Добавление столбца | Настраиваемый столбец). Назовите свой пользовательский столбец `Wpct` и настройте для него следующую формулу:

```
[W] / ([W] + [L])
```

Язык программирования M в Power Query имеет синтаксис, напоминающий таблицы Excel, когда ссылки на столбцы заключаются в одинарные квадратные скобки. Пользуйтесь преимуществами IntelliSense от Microsoft и по мере ввода этих ссылок нажимайте клавишу <Tab> для автоматического заполнения кода. Кроме того, вы можете выполнить двойной щелчок на нужном столбце в списке **Available columns** (Доступные столбцы), чтобы вставить его в формулу.

Если всё сделано правильно, в нижнем левом углу диалогового окна (рис. 4.5) появятся зеленая галочка и надпись, свидетельствующие, что синтаксических ошибок не обнаружено.

У созданного столбца измените в Power Query тип данных на **Percentage** (Процент).

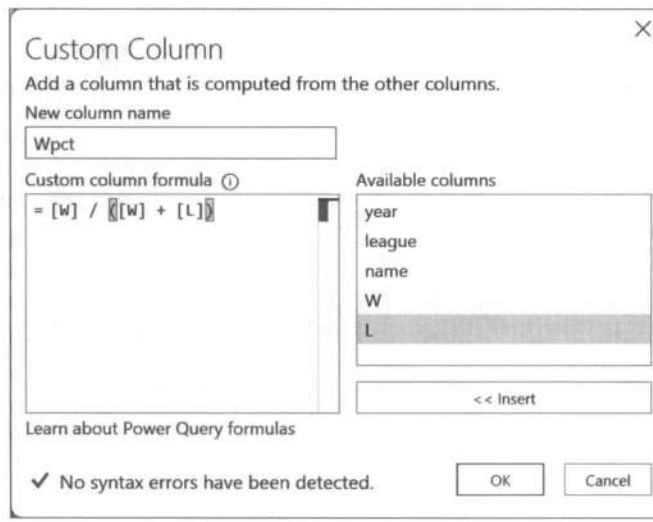
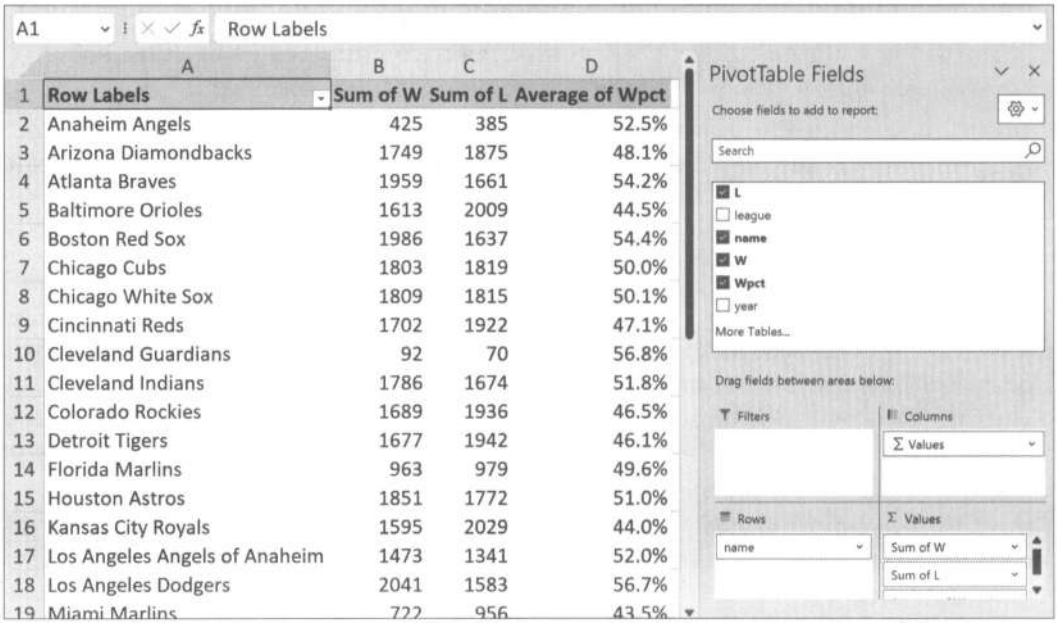


Рис. 4.5. Создание вычисляемого столбца для процента выигрыша

Загрузка и проверка данных

Наш новый столбец вычислен и готов к дальнейшей работе. На ленте редактора Power Query выберите **Home | Close & Load | Close & Load To** (Закрыть и загрузить в), затем **PivotTable Report** (Отчет сводной таблицы) и нажмите **OK**. Теперь вы можете проанализировать данные — например, вычислить среднее значение `Wpct` для каждой команды (рис. 4.6).



	A	B	C	D
1	Row Labels	Sum of W	Sum of L	Average of Wpct
2	Anaheim Angels	425	385	52.5%
3	Arizona Diamondbacks	1749	1875	48.1%
4	Atlanta Braves	1959	1661	54.2%
5	Baltimore Orioles	1613	2009	44.5%
6	Boston Red Sox	1986	1637	54.4%
7	Chicago Cubs	1803	1819	50.0%
8	Chicago White Sox	1809	1815	50.1%
9	Cincinnati Reds	1702	1922	47.1%
10	Cleveland Guardians	92	70	56.8%
11	Cleveland Indians	1786	1674	51.8%
12	Colorado Rockies	1689	1936	46.5%
13	Detroit Tigers	1677	1942	46.1%
14	Florida Marlins	963	979	49.6%
15	Houston Astros	1851	1772	51.0%
16	Kansas City Royals	1595	2029	44.0%
17	Los Angeles Angels of Anaheim	1473	1341	52.0%
18	Los Angeles Dodgers	2041	1583	56.7%
19	Miami Marlins	722	956	43.5%

Рис. 4.6. Загрузка результатов в сводную таблицу

Типы данных Power Query и форматирование ячеек в Excel

Имейте в виду, что изменение типа данных для столбца в Power Query влияет только на то, как он отображается в самом Power Query, и не влияет на формат данных в Excel. То есть, если вы изменили тип столбца на проценты в Power Query (как было сделано для столбца `Wpct` в приведенном примере), вам все равно придется задавать процентный формат еще раз в Excel. В *части III* мы рассмотрим, как системно менять представление данных с помощью форматирования в Power Pivot.

Вычисляемые столбцы и собственные расчеты

Важно отметить, что среднее значение `Wpct`, отображаемое в сводной таблице, представляет собой простое (невзвешенное) среднее от сезонных процентов выигрышей. Это означает, что сезоны с небольшим количеством игр (например, в пандемийном 2020 году) оказали непропорционально большое влияние на это среднее значение. Чтобы убедиться в этом, сравните значение `Average of Wpct` из сводной таблицы с нашим собственным расчетом по формуле в Excel (рис. 4.7).

	A	B	C	D	E	F	G
1	Row Labels	Sum of W	Sum of L	Average of Wpct	Average Wpct (Measure)		
2	Anaheim Angels	425	385	52.47%	52.47%	=B2 / (B2 + C2)	
3	Arizona Diamondbacks	1749	1875	48.08%	48.26%		
4	Atlanta Braves	1959	1661	54.23%	54.12%		
5	Baltimore Orioles	1613	2009	44.45%	44.53%		
6	Boston Red Sox	1986	1637	54.41%	54.82%		
7	Chicago Cubs	1803	1819	49.97%	49.78%		
8	Chicago White Sox	1809	1815	50.14%	49.92%		
9	Cincinnati Reds	1702	1922	47.09%	46.96%		
10	Cleveland Guardians	92	70	56.79%	56.79%		
11	Cleveland Indians	1786	1674	51.81%	51.62%		
12	Colorado Rockies	1689	1936	46.50%	46.59%		
13	Detroit Tigers	1677	1942	46.14%	46.34%		
14	Florida Marlins	963	979	49.59%	49.59%		
15	Houston Astros	1851	1772	51.02%	51.09%		
16	Kansas City Royals	1595	2029	43.99%	44.01%		
17	Los Angeles Angels of Anaheim	1473	1341	52.03%	52.35%		
18	Los Angeles Dodgers	2041	1583	56.74%	56.32%		

Рис. 4.7. Очевидное расхождение в расчетах сводной таблицы

Чтобы решить эту проблему, один из имеющихся способов предполагает использование динамических показателей для агрегирования по времени и для вычислений с учетом контекста анализа. Это можно сделать с помощью таких инструментов, как модель данных Power Pivot и язык DAX, о которых будет рассказано в *части II* этой книги.

Однако это не означает, что в Power Query вообще следует избегать использования вычисляемых столбцов. Их очень легко создавать, и они удобны для вычислений. Тем не менее, если есть вероятность, что эти столбцы могут привести к ошибочным статистическим данным, вместо них рекомендуется использовать меры DAX.

Изменение структуры данных

В *главе 1* вы познакомились с понятием «упорядоченных» данных, когда каждая переменная величина хранится в отдельном столбце, и, может быть, еще помните данные из рабочего листа `sales`, которые мы рассматривали в качестве примера неупорядоченных данных. К счастью, Power Query решает эту проблему с хранением данных. Для начала перейдите к уже знакомому рабочему листу `sales` в файле ch_04.xlsx и загрузите эту таблицу в Power Query, чтобы мы могли начать преобразование данных.

Задача состоит в том, чтобы «развернуть» или «слить» все столбцы с продажами в один столбец с именем `sales`, а названия этих столбцов с продажами поместить в столбец `department`. Для этого, удерживая клавишу <Ctrl>, выделите первые три переменные: `customer_id`, `channel` и `region`, после чего щелкните на них правой кнопкой мыши и выберите из контекстного меню **Unpivot Other Columns** (Отменить свертывание других столбцов), как показано на рис. 4.8.

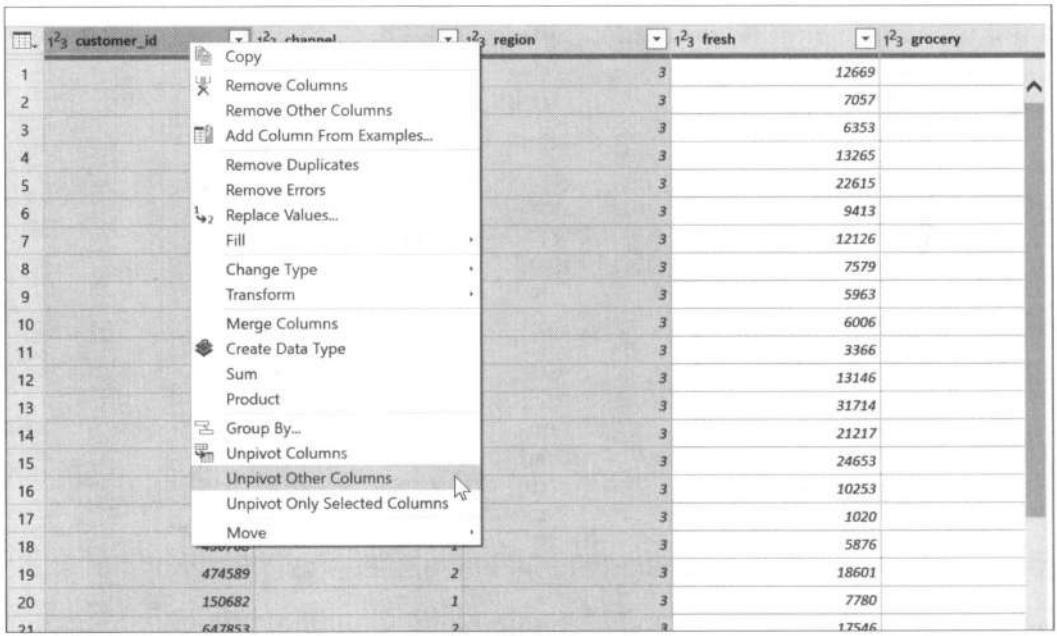
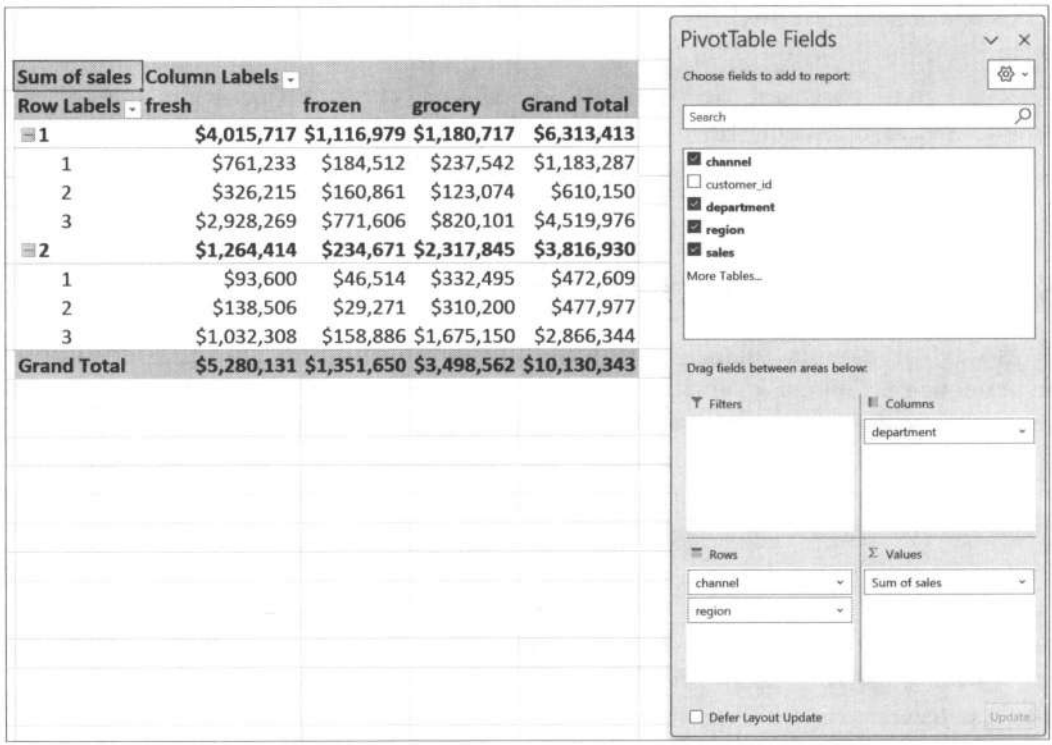


Рис. 4.8. Развертывание набора данных в Power Query



Sum of sales	Column Labels			
Row Labels	**fresh**	**frozen**	**grocery**	**Grand Total**
1	**$4,015,717**	**$1,116,979**	**$1,180,717**	**$6,313,413**
1	$761,233	$184,512	$237,542	$1,183,287
2	$326,215	$160,861	$123,074	$610,150
3	$2,928,269	$771,606	$820,101	$4,519,976
2	**$1,264,414**	**$234,671**	**$2,317,845**	**$3,816,930**
1	$93,600	$46,514	$332,495	$472,609
2	$138,506	$29,271	$310,200	$477,977
3	$1,032,308	$158,886	$1,675,150	$2,866,344
Grand Total	**$5,280,131**	**$1,351,650**	**$3,498,562**	**$10,130,343**

Рис. 4.9. Настройка сводной таблицы для развернутого набора данных

По умолчанию два полученных развернутых столбца будут называться `Attribute` (Атрибут) и `Value` (Значение). Переименуйте их соответственно в `department` и `sales`. Теперь вы можете загрузить эту выборку в сводную таблицу и проанализировать продажи по `channel` и `region`. Результаты и преимущества создания сводной таблицы на основе этих измененных данных показаны на рис. 4.9.

Заключение

В этой главе мы рассмотрели различные способы преобразования столбцов в Power Query. В *главе 5* мы сделаем еще один шаг вперед и узнаем, как работать с несколькими наборами данных в одном запросе. Вы научитесь объединять и добавлять источники данных, а также подключаться к внешним источникам — например, к файлам *.csv.

Упражнения

Потренируйтесь выполнять различные преобразования со столбцами в Power Query, используя файл ch_04_exercises.xlsx, расположенный в папке exercises\ch_04_exercises сопроводительного репозитория к этой книге[2]. Выполните следующие преобразования этого набора данных с информацией о заказах:

1. Переведите столбец `date` в формат месяца, чтобы, например, `1/1/2023` изменилось на `Январь`.
2. Преобразуйте столбец `owner` в правильный регистр (каждое слово с заглавной буквы).
3. Разделите столбец `location` на два столбца: `zip` и `state`.
4. Измените структуру набора данных так, чтобы столбцы `subscription_cost`, `support_cost` и `services_cost` преобразовались в два столбца: `category` и `cost`.
5. Добавьте новый столбец `tax`, который рассчитывается как 7% от значений столбца `cost`.
6. Преобразуйте переменную `zip` в тип данных **Text**, а столбцы `cost` и `tax` — в **Currency**.
7. Загрузите результаты в таблицу.

Готовое решение с выполненными преобразованиями можно посмотреть в файле ch_04_exercise_solutions.xlsx, расположенном в той же папке репозитория.

[2] См. **https://clck.ru/3Jbnk2**.

Объединение и добавление данных в Power Query

До сих пор в *части I* мы рассматривали различные операции Power Query по преобразованию строк и столбцов в единственной таблице. Но, как правило, данные поступают в нескольких таблицах или даже из разных источников не из среды Excel. В этой главе вы узнаете, как объединить несколько файлов в один набор данных.

Поскольку в этой главе будут использоваться подключения к внешним файлам, а не к таблицам внутри рабочей книги, создайте новую рабочую книгу.

Добавление нескольких источников

Очень часто данные поступают в нескольких файлах, и нужно соединить их вертикально друг с другом. Для примера на рис. 5.1 показана типичная ситуация, когда данные о продажах за январь, февраль и март пришли в отдельных таблицах. В таких случаях их удобно объединить в одну. Это позволит вычислить общий объем продаж, например за первый квартал.

Операция добавления (append) в Power Query упрощает нам этот действие.

month	region_numbe	sales
January	1	$1,182
January	2	$946
January	3	$1,034

month	region_numbe	sales
Feburary	1	$983
Feburary	2	$1,166
Feburary	3	$805

month	region_numbe	sales
March	1	$1,052
March	2	$1,162
March	3	$1,250

month	region_numbe	sales
January	1	$1,182
January	2	$946
January	3	$1,034
Feburary	1	$983
Feburary	2	$1,166
Feburary	3	$805
March	1	$1,052
March	2	$1,162
March	3	$1,250

Рис. 5.1. Простой пример наборов данных для объединения

Подключение к внешним рабочим книгам Excel

Как уже было отмечено ранее, до сих пор в этой книге мы использовали Power Query для работы с источниками данных внутри одной рабочей книги. Однако воз-

можности инструмента Power Query значительно шире. Он позволяет подключить множество источников данных, в частности внешние файлы Excel и файлы формата CSV, которым и будет уделено основное внимание в этой главе. В папке ch_05 сопроводительного репозитория к этой книге[1] содержатся наборы данных, взятые из базы данных спортивного журналиста Шона Лахмана (Sean Lahman) по играм Главной лиги бейсбола сезона 2022 г.

Файлы people_born_in_usa.xlsx и people_born_outside_usa.xlsx включают информацию об игроках Главной лиги бейсбола, родившихся в США и за их пределами соответственно. Задача состоит том, чтобы с помощью Power Query вертикально соединить эти два файла в одну таблицу.

Чтобы начать работу, перейдите на ленте на вкладку **Data** (Данные) и выберите **Get Data | From File | From Excel Workbook** (Получить данные | Из файла | Из книги), как показано на рис. 5.2.

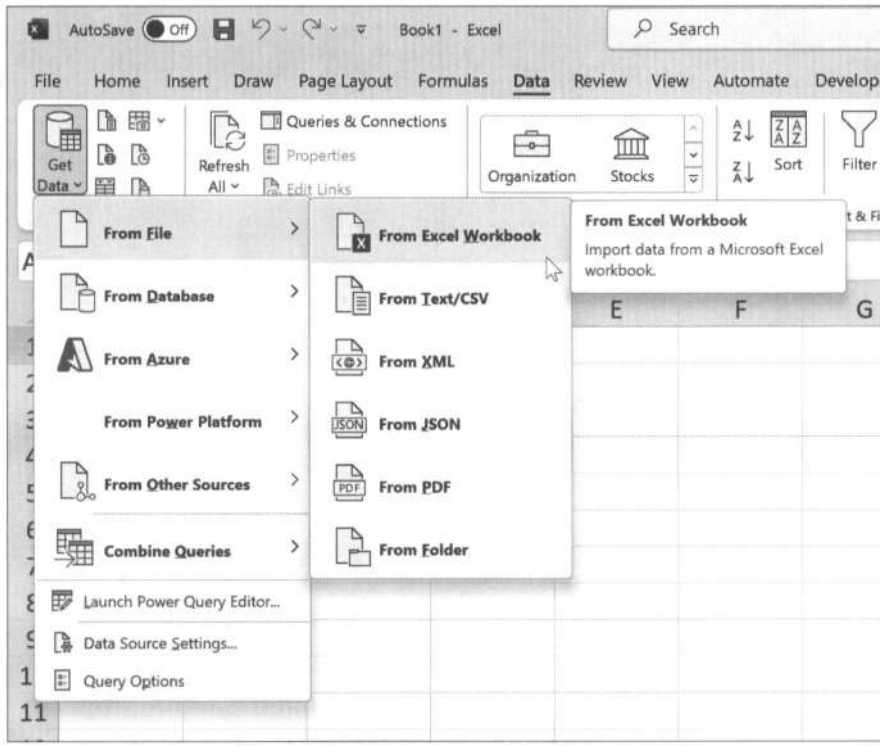


Рис. 5.2. Подключение к внешнему файлу Excel для Power Query

Давайте сначала подключим файл people_born_in_usa.xlsx. Учтите, что рабочая книга Excel может содержать несколько листов, именованные диапазоны, таблицы и многое другое. Поэтому вам нужно указать, какой именно элемент рабочей книги вы хотите загрузить в Power Query. В нашем случае нам нужна таблица people_born_

[1] См. **https://clck.ru/3JhrZ4**.

in_usa, поэтому в диалоговом окне **Navigator** (Навигатор) выделите эту строчку в списке под строкой поиска (рис. 5.3).

Перед загрузкой данных в рабочую книгу вы можете очистить или преобразовать их в редакторе Power Query, для чего надо нажать кнопку **Transform Data** (Преобразовать данные). Однако сейчас мы сразу загрузим данные в рабочую книгу, а если позднее нам понадобится как-либо преобразовать данные, всегда можно будет открыть Power Query и внести необходимые изменения.

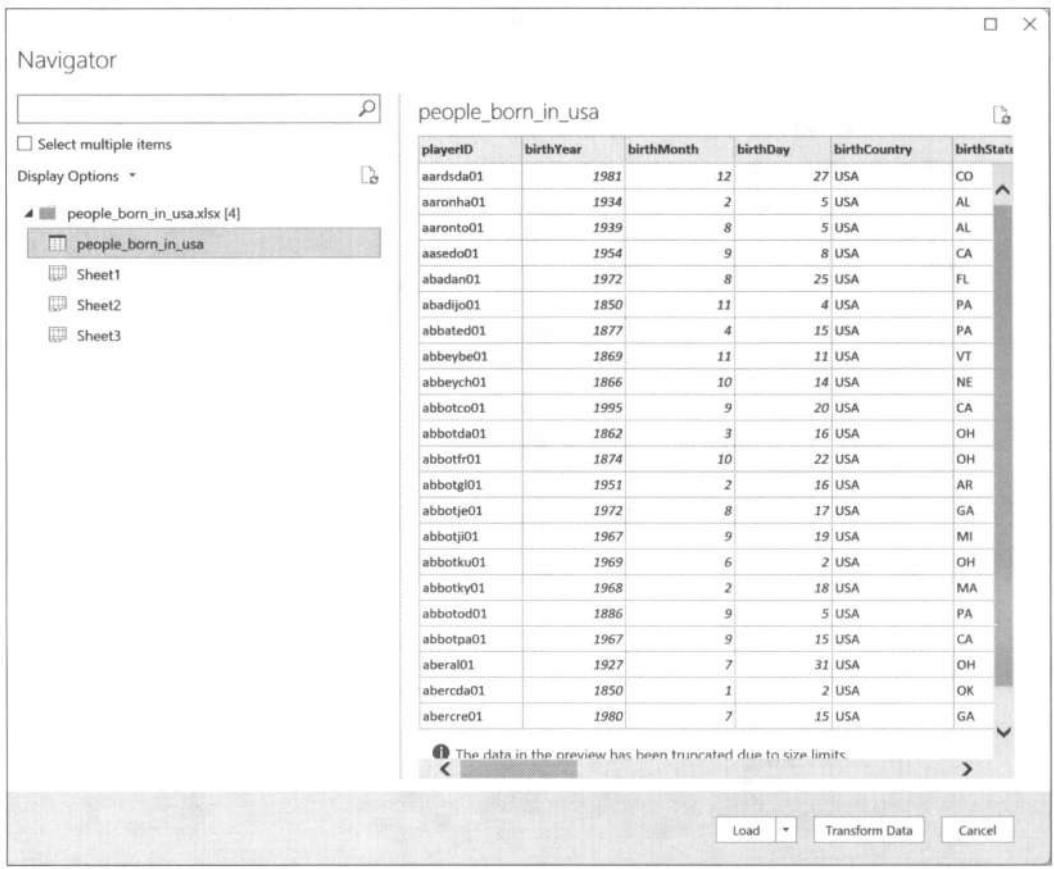


Рис. 5.3. Загрузка внешней рабочей книги Excel в Power Query

Нажмите в диалоговом окне **Navigator** на раскрывающуюся кнопку рядом с пунктом **Load** (Загрузить), выберите **Load To** (Загрузить в) и установите опцию **Only Create Connection** (Только создать подключение). Поскольку наша задача заключается в том, чтобы соединить этот файл с другим для последующего совместного анализа, на этом шаге нет необходимости загружать данные в отдельную таблицу Excel.

Повторите эти же действия для загрузки файла people_born_outside_usa.xlsx, так же загрузив данные с опцией **Only Create Connection**. Теперь в Power Query у нас оба файла подключены только как соединения.

На вкладке **Data** ленты откройте панель **Queries & Connections** (Запросы и подключения) — здесь вы найдете запросы `people_born_in_usa` и `people_born_outside_usa` с пометкой **Connection only** (Только подключение). Щелкните правой кнопкой мыши на `people_born_in_usa` и выберите **Edit** (Изменить), чтобы открыть редактор Power Query (рис. 5.4).

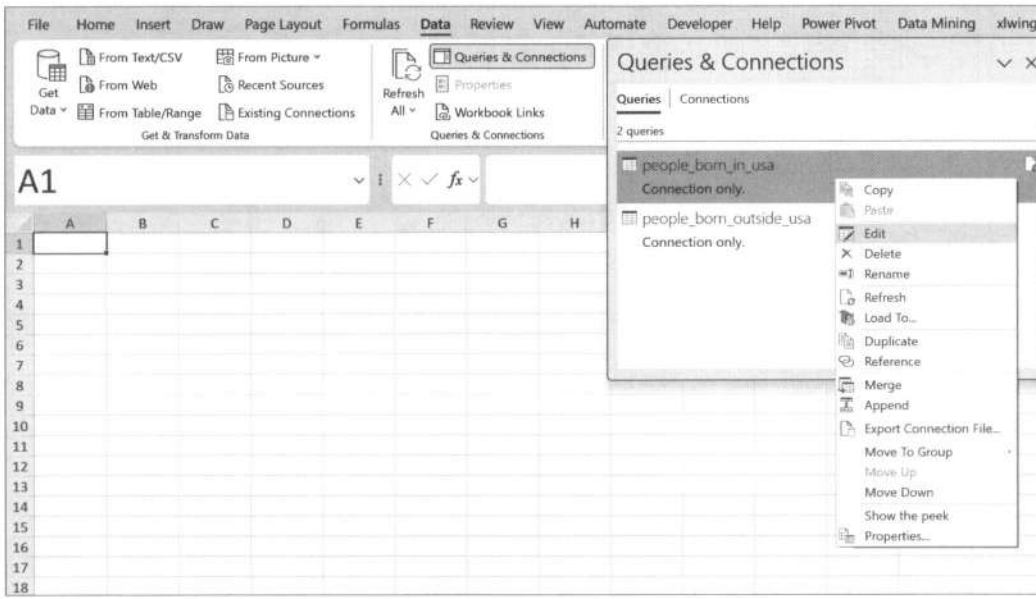


Рис. 5.4. Просмотр списка запросов на панели Queries & Connections

Добавление запросов

На ленте редактора Power Query перейдите на вкладку **Home**, в группе **Combine** (Объединить) раскройте выпадающий список **Append Queries** (Добавить запросы) и выберите **Append Queries as New** (Добавить запросы в новый), как показано на рис. 5.5.

Опция **Append Queries** добавляет данные из таблиц к существующему запросу, увеличивая его размер, в то время как **Append Queries as New** объединяет их в новый запрос, сохраняя исходные таблицы неизменными.

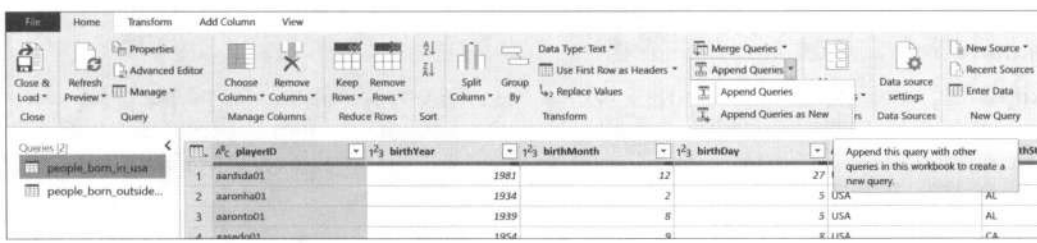


Рис. 5.5. Добавление запросов в новый запрос в Power Query

Запросы, которые вы хотите объединить, должны иметь согласованную структуру данных, т. е. одно и то же количество столбцов, одинаковые имена столбцов и типы данных. В противном случае сначала вам нужно будет выполнить некоторые действия по преобразованию данных, чтобы выровнять структуру данных перед добавлением.

После выбора опции **Append Queries as New** откроется диалоговое окно **Append** (Добавление), предлагающее уточнить, какие таблицы нужно объединить в одну. Выберите здесь `people_born_in_usa` и `people_born_outside_usa` (рис. 5.6).

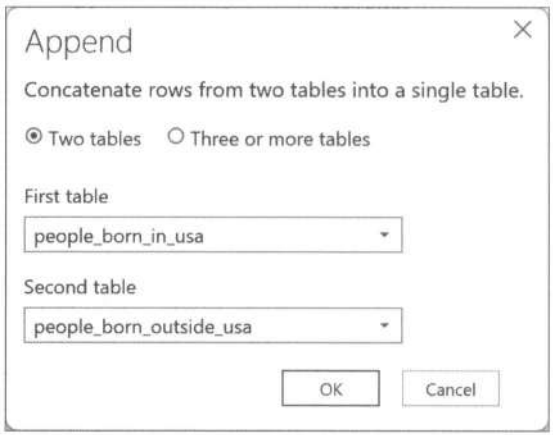


Рис. 5.6. Добавление двух таблиц в Power Query

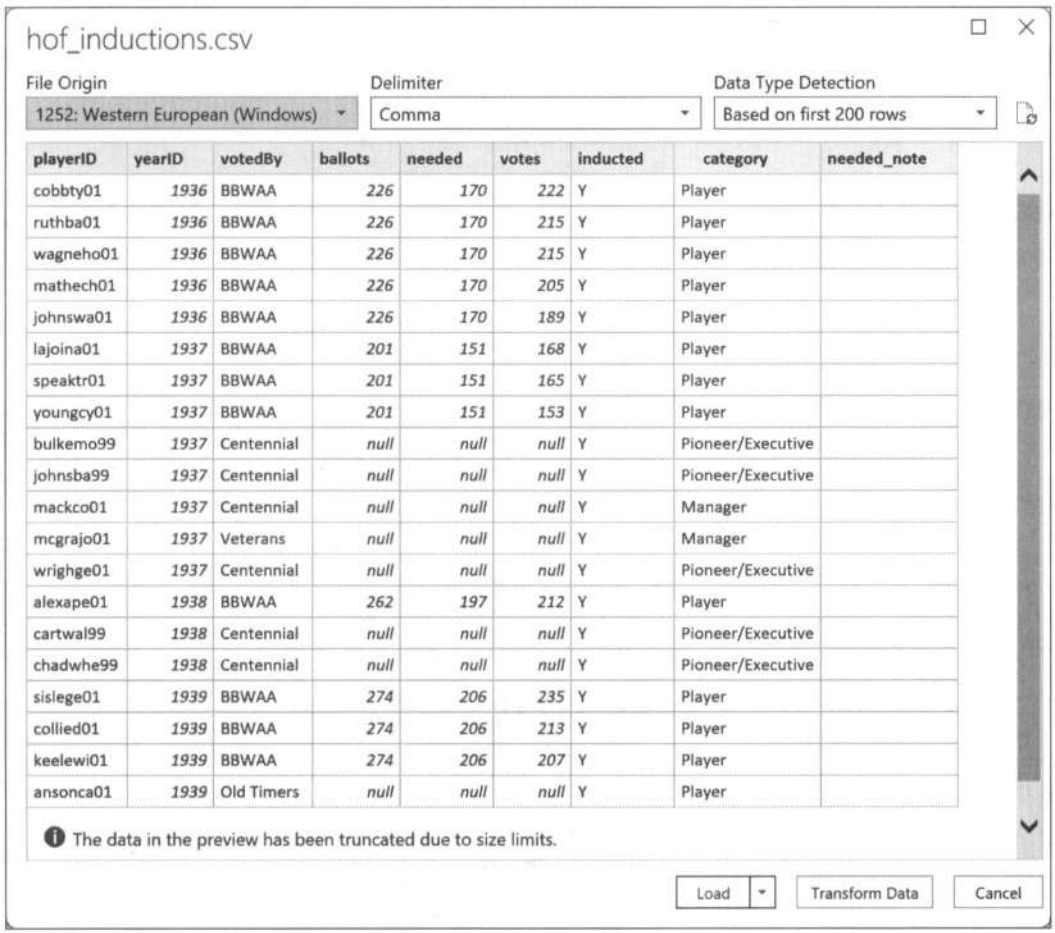
Поздравляю! Вы объединили две таблицы, чтобы создать новый запрос с именем `Append1` (`Добавить1`). Для наглядности переименуйте его в `people_append`. Закройте и загрузите результаты в таблицу Excel. В итоговом запросе будет 20 370 строк, что соответствует суммарному количеству строк в обеих таблицах. Вы можете проверить это, используя возможности Power Query по профилированию данных, о которых говорилось в *главе 2*.

Реляционные соединения

После объединения всех персональных данных в одну таблицу на следующем шаге необходимо связать ее с другими таблицами для получения дополнительной информации. Исходная база данных Лахмана включает в себя различные таблицы с персональными данными, в том числе записи об отбиваниях, появлениях на Играх всех звезд (All-Star game) и пр. Используя столбец `playerID`, можно эффективно соединить эти таблицы друг с другом.

Давайте попробуем это сделать в вашей текущей рабочей книге, подключив с помощью Power Query из файла hof_inductions.csv, расположенного в той же папке, набор данных, содержащий информацию о тех, кто был введен в Зал славы бейсбола (Baseball Hall of Fame). Для этого на ленте Excel выберите **Data | Get Data | From File | From Text/CSV** (Из текстового/CSV-файла), найдите и выберите файл hof_inductions.csv. В отличие от Excel, формат CSV не поддерживает нескольких ра-

бочих листов или диапазонов, поэтому сразу отобразится окно с данными из файла (рис. 5.7). Загрузите содержимое файла hof_inductions.csv в рабочую книгу только как соединение.



playerID	yearID	votedBy	ballots	needed	votes	inducted	category	needed_note
cobbty01	1936	BBWAA	226	170	222	Y	Player	
ruthba01	1936	BBWAA	226	170	215	Y	Player	
wagneho01	1936	BBWAA	226	170	215	Y	Player	
mathech01	1936	BBWAA	226	170	205	Y	Player	
johnswa01	1936	BBWAA	226	170	189	Y	Player	
lajoina01	1937	BBWAA	201	151	168	Y	Player	
speaktr01	1937	BBWAA	201	151	165	Y	Player	
youngcy01	1937	BBWAA	201	151	153	Y	Player	
bulkemo99	1937	Centennial	null	null	null	Y	Pioneer/Executive	
johnsba99	1937	Centennial	null	null	null	Y	Pioneer/Executive	
mackco01	1937	Centennial	null	null	null	Y	Manager	
mcgrajo01	1937	Veterans	null	null	null	Y	Manager	
wrighge01	1937	Centennial	null	null	null	Y	Pioneer/Executive	
alexape01	1938	BBWAA	262	197	212	Y	Player	
cartwal99	1938	Centennial	null	null	null	Y	Pioneer/Executive	
chadwhe99	1938	Centennial	null	null	null	Y	Pioneer/Executive	
sislege01	1939	BBWAA	274	206	235	Y	Player	
collied01	1939	BBWAA	274	206	213	Y	Player	
keelewi01	1939	BBWAA	274	206	207	Y	Player	
ansonca01	1939	Old Timers	null	null	null	Y	Player	

Рис. 5.7. Загрузка файла формата CSV в Power Query

Когда все необходимые данные загружены в Power Query, мы можем соединить информацию, содержащуюся в таблице `people_append`, с информацией, содержащейся в таблице `hof_inductions`, с помощью общего столбца `playerID`.

Один из способов выполнить это предлагает поисковая функция Excel — например, `VLOOKUP()`, которую можно использовать для получения соответствующих имен игроков для каждого значения `playerID`. Мне нравится называть `VLOOKUP()` «клейкой лентой в Excel» из-за ее способности добавлять дополнительные столбцы к набору данных.

Но `VLOOKUP()` по сравнению с реляционным соединением это как клейкая лента по сравнению с полноценной сваркой, поскольку `VLOOKUP()` предназначен в первую очередь для поиска в среде Excel по одному условию. Кроме того, в этой функции нет

системного подхода к обработке пропущенных значений, что может привести к несоответствию данных. Она также может сильно замедлить работу рабочих книг, т. к. каждая формула `VLOOKUP()` повторно вычисляется каждый раз, когда пересчитывается рабочая книга.

Разработанная в качестве современной замены `VLOOKUP()` новая функция в Excel `XLOOKUP()` устраняет некоторые из этих недостатков. Однако и она не решает всех проблем, в отличие от реляционных соединений в Power Query. Дополнительная информацию о функции `XLOOKUP()` приведена в *главе 10*.

Power Query предоставляет более комплексное решение. Он позволяет соединять данные на основе нескольких условий, более эффективно справляется с большими наборами данных, дает возможность системно обрабатывать недостающие значения, запоминает выполненные шаги по преобразованию данных для обеспечения их целостности, а также может собирать данные из различных источников.

Этот способ более эффективен и с вычислительной точки зрения, поскольку соединение создается только один раз и повторно вычисляется лишь при обновлении запроса. В результате получается плоская таблица без формул, с которой гораздо проще работать. Всё это делает Power Query более перспективным инструментом для сложной интеграции и преобразования данных.

В следующих разделах мы рассмотрим два самых распространенных типа реляционных соединений: левое внешнее (left outer join) и внутреннее (inner join).

Левое внешнее соединение: почти то же, что и *VLOOKUP()*

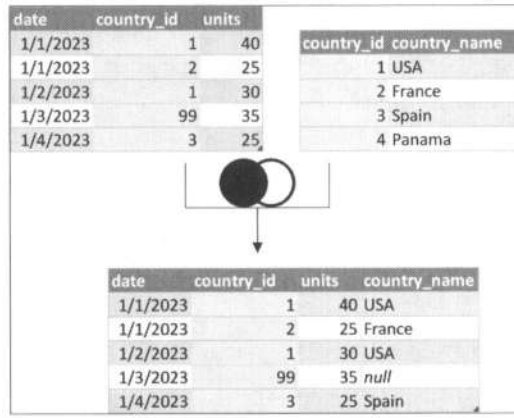
Левое внешнее соединение (left outer join) сохраняет все записи из первой объединяемой таблицы и ищет соответствующие им записи во второй таблице. Если совпадений не найдено, возвращается `null`. Этот тип соединения очень похож на `VLOOKUP()`, но с одним значительным отличием — он использует `null` для отсутствующих значений, в то время как `VLOOKUP()` вернуло бы `#N/A`.

Пример левого внешнего соединения на небольшом наборе данных показан на рис. 5.8.

Чтобы приступить к объединению, вернитесь в редактор Power Query, выберите таблицу `people_append`, перейдите на ленте в группу **Combine** (Объединить) и выберите **Merge Queries as New** (Объединить запросы в новый), как показано на рис. 5.9.

В диалоговом окне **Merge** (Слияние) выберите во втором выпадающем списке `hof_inductions` и щелкните на столбце `playerID` в обеих таблицах, чтобы определить его в качестве столбца, по которому будет выполняться соединение. Затем проверьте, что в третьем выпадающем списке выбрано **Left Outer** (Внешнее соединение слева) в качестве требуемого типа соединения (рис. 5.10).

Нажмите **OK**, и вы увидите результат соединения в запросе с именем `Merge1` (`Слияние1`). Выполните двойной щелчок на этом имени и переименуйте его в `people_left`.



date	country_id	units
1/1/2023	1	40
1/1/2023	2	25
1/2/2023	1	30
1/3/2023	99	35
1/4/2023	3	25

country_id	country_name
1	USA
2	France
3	Spain
4	Panama

date	country_id	units	country_name
1/1/2023	1	40	USA
1/1/2023	2	25	France
1/2/2023	1	30	USA
1/3/2023	99	35	*null*
1/4/2023	3	25	Spain

Рис. 5.8. Наглядный пример левого внешнего соединения

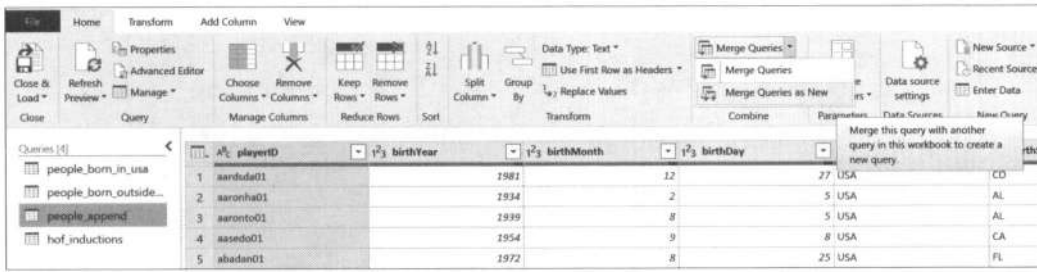


Рис. 5.9. Объединение запросов в новый запрос в Power Query

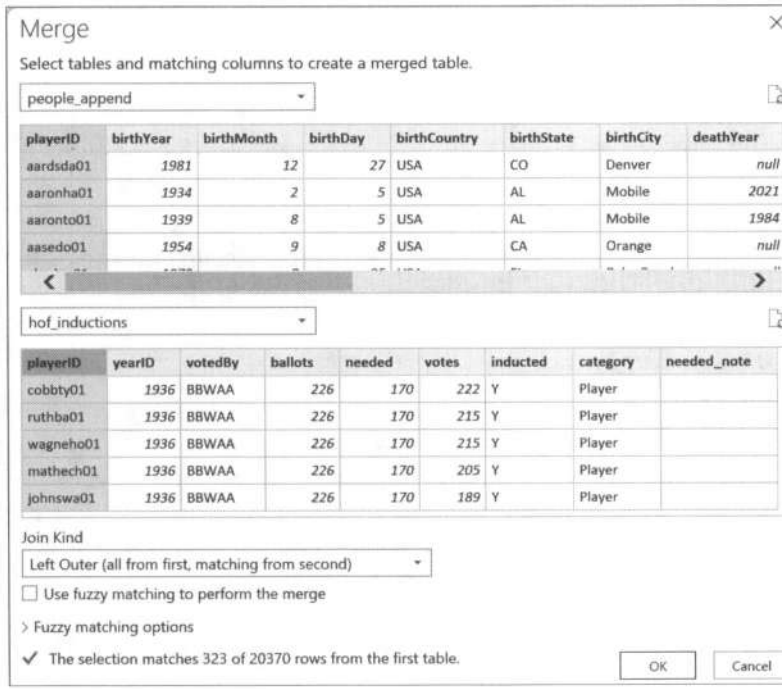


Рис. 5.10. Левое внешнее соединение в Power Query

Прокрутите вправо набор данных `people_left`. Данные в нашем запросе сейчас выглядят немного необычно — особенно столбец `hof_inductions`, который имеет значение `Table` в каждой строке данных. Это значение представляет собой вложенную таблицу, содержащую все совпадающие строки из второй таблицы для соответствующей строки из первой таблицы.

Нажмите на значок  , расположенный справа от заголовка столбца `hof_inductions`, а затем на кнопку **OK**, чтобы развернуть вложенные данные (рис. 5.11).

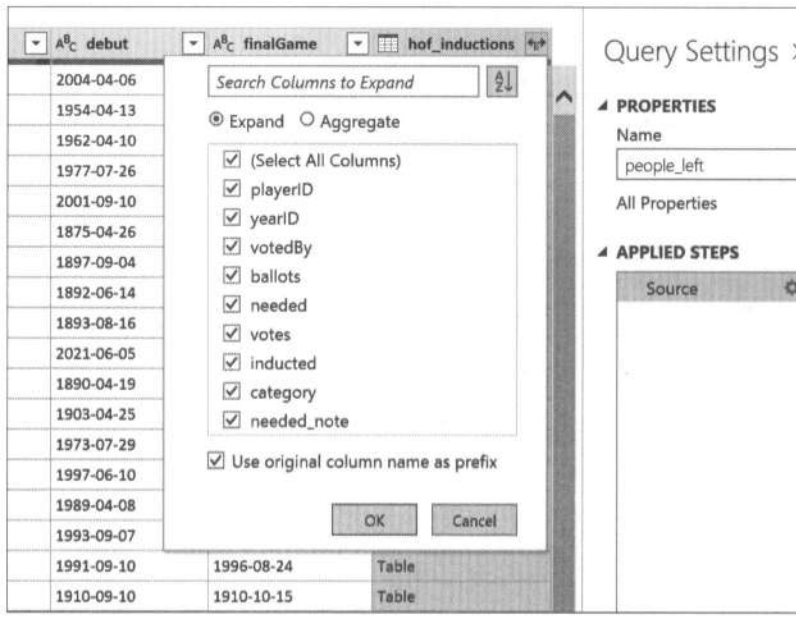


Рис. 5.11. Разворачивание результатов левого внешнего соединения

В этом списке вы можете выбрать любое количество столбцов из соответствующих записей в таблице `hof_inductions`. Вы также можете добавить к этим столбцам префикс с названием их таблицы. Для простоты давайте примем все настройки по умолчанию и загрузим все столбцы с префиксом. Однако для оптимизации ваших запросов в реальных проектах вы, скорее всего, захотите уменьшить количество разворачиваемых столбцов.

Загрузите результаты в таблицу в Excel. Таблица `people_left`, как и исходная таблица `people_append`, содержит 20 370 записей. Это связано с тем, что левое внешнее соединение сохраняет все записи из таблицы `people`, независимо от того, есть ли соответствующая запись слева или нет. Результаты соединения похожи на результаты `VLOOKUP()` — для каждого игрока подтягиваются связанные записи в Зале славы бейсбола. В чем же преимущество? Соединение сопоставляет все записи из таблицы `hof_inductions` за один проход и не выдает ошибок при несовпадении записей.

Внутреннее соединение: только точное соответствие

В отличие от левого внешнего, *внутреннее соединение* (inner join) сохраняет в результирующей таблице только те записи, для которых в обеих таблицах найдено соответствие (рис. 5.12).

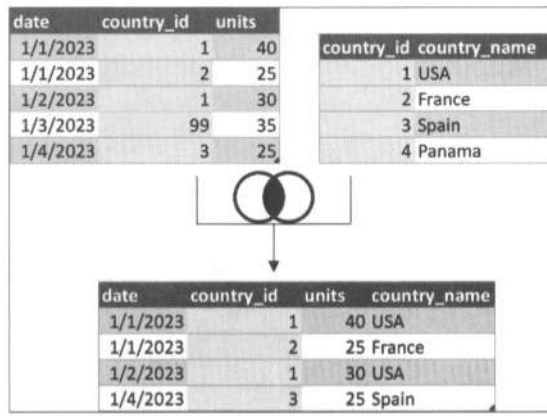


Рис. 5.12. Наглядный пример внутреннего соединения

Следуя этой логике, запись с `country_id = 4` из правой таблицы не появится в результирующей таблице, поскольку в левой таблице нет такого значения. Аналогично запись с `country_id = 99` из левой таблицы также будет исключена по той же причине — нет совпадения с правой таблицей. То есть запись будет включена в результирующую таблицу, только если для нее есть совпадение в обеих таблицах.

Такой подход нужен для сохранения только полных записей и исключения данных с возможными нарушениями целостности. С учетом этого внутреннее соединение может вернуть меньше строк, чем левое внешнее соединение.

Повторив те же действия, что и ранее, выполним внутреннее соединение в Power Query. Выберите в редакторе `people_append` и на ленте **Home | Merge Queries | Merge Queries as New**. Диалоговое окно для параметров соединения должно выглядеть так, как показано на рис. 5.13.

Вы можете развернуть нужные столбцы из вложенной таблицы так же, как и в случае с левым внешним соединением, и переименовать запрос в `people_inner`. Загрузите полученный запрос в таблицу — он будет содержать всего 323 записи.

Такая разница имеет довольно простое объяснение: внутреннее соединение возвращает только те записи, которые имеют соответствующее совпадение в обеих таблицах. Не все значения `playerID` встречаются в таблице `hof_inductions`, поскольку не все игроки были введены в Зал славы бейсбола, что и приводит к отсутствию многих `playerID` в результирующей таблице.

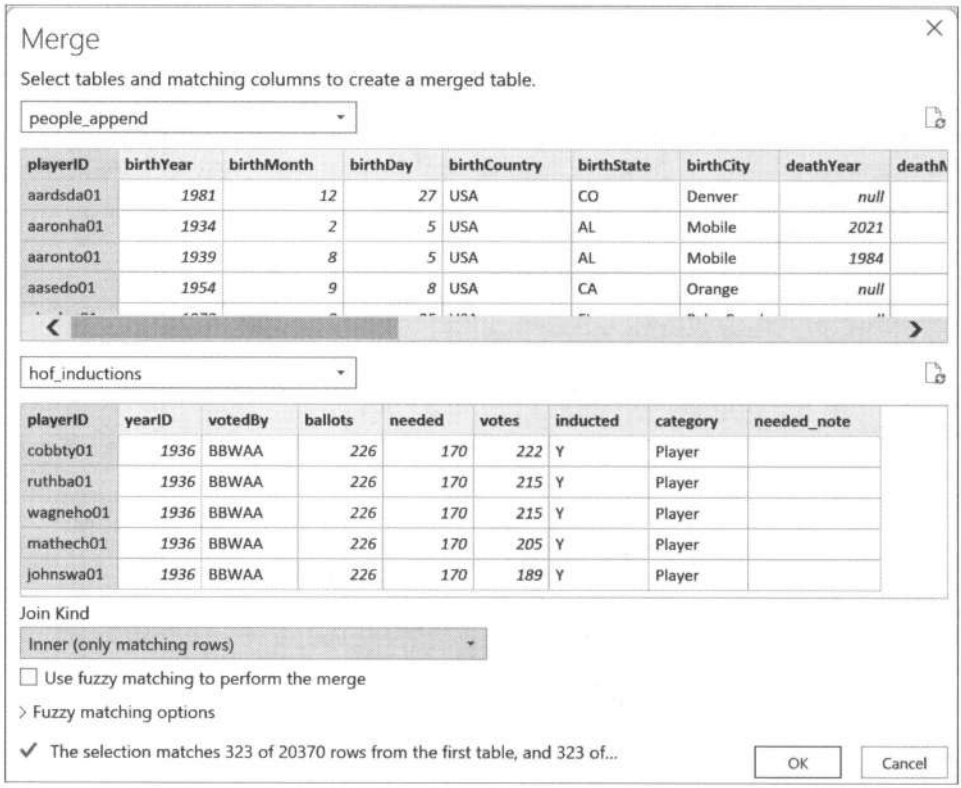


Рис. 5.13. Параметры для выполнения внутреннего соединения

Управление вашими запросами

Проделана отличная работа по загрузке и объединению данных из различных источников и форматов! По мере погружения в Power Query вы, вероятно, будете создавать множество запросов в своей рабочей книге. Управление этими запросами и понимание того, как они взаимодействуют между собой, приобретет решающее значение.

Группировка запросов

Группировка запросов в Power Query улучшает организацию процесса и упрощает его сопровождение за счет категоризации связанных запросов. Такой подход делает управление сложными проектами Excel более легким. Группируя запросы соответствующим образом, вы можете быстро отличить базовые запросы от их зависимостей, таких как добавления и объединения, созданных на основе этих базовых запросов.

Чтобы проверить, как этой работает, вернитесь в редактор Power Query.

В списке **Queries**, расположенном в окне редактора слева, удерживая клавишу <Ctrl>, выберите исходные запросы: `people_born_in_usa`, `people_born_outside_usa` и

`hof_inductions`. Щелкните правой кнопкой мыши и выберите **Move To Group | New Group** (Переместить в группу | Новая группа), как показано на рис. 5.14.

Во всплывающем окне **New Group** (Новая группа) введите название группы `Sources`. Нажмите **OK**. Вы увидите, что в списке **Queries** эти три набора данных теперь выделены в новую папку. Остальные запросы: `hof_append`, `hof_left` и `hof_inner` — также автоматически перемещены в группу **Other Queries** (Другие запросы), как показано на рис. 5.15.

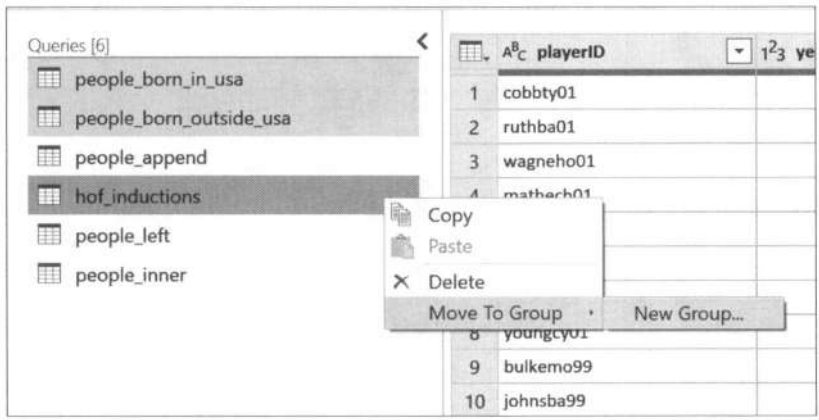


Рис. 5.14. Создание новой группы запросов в Power Query

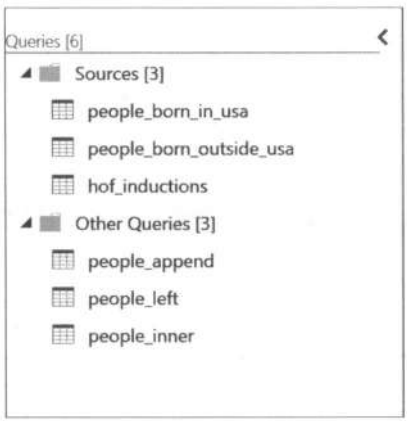


Рис. 5.15. Сгруппированные запросы

Просмотр зависимостей запросов

Просмотр *зависимостей запросов* позволяет наглядно увидеть, как запросы взаимосвязаны, а это помогает оценивать влияние изменений и эффективно управлять зависимостями в сложных проектах, чтобы обеспечивать целостность данных и уменьшать количество ошибок. Чтобы посмотреть, как это выглядит, на ленте редактора перейдите в раздел **View** (Просмотр), а затем выберите **Query Depen-**

dencies (Зависимости запроса). Открывшееся окно должно выглядеть примерно так, как показано на рис. 5.16.

Здесь показано, какие запросы получены непосредственно из первичных источников данных (например, из файлов *.csv, где расположены эти файлы, какие источники участвуют в объединениях или добавлениях, какие из них загружены в рабочую книгу и многое другое.

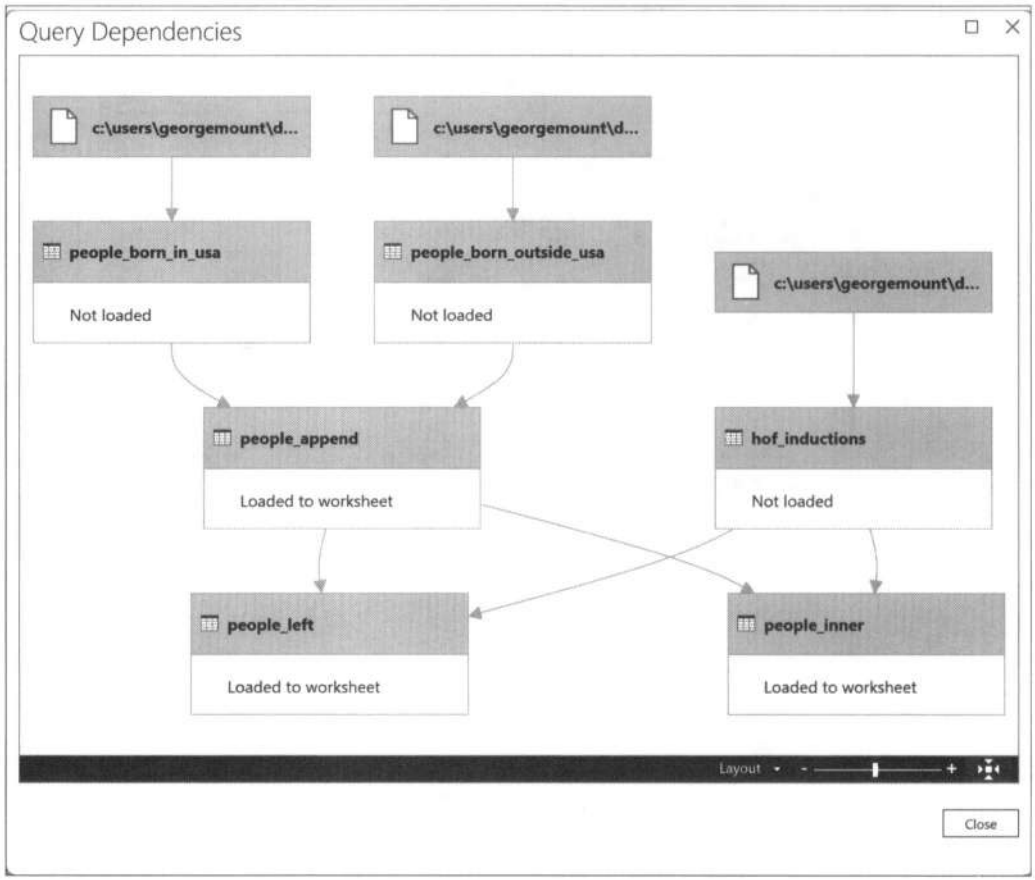


Рис. 5.16. Просмотр зависимостей запросов

Закончив просмотр диаграммы с зависимостями запросов, нажмите кнопку **Close** (Закрыть).

Заключение

В этой главе мы рассмотрели процессы объединения и добавления данных в Power Query, а также важность понимания получаемых результатов. Возможность объединять различные источники данных, например рабочие книги Excel и файлы формата CSV, в единые наборы помогает легко производить эффективный анализ данных.

В Power Query есть и другие типы соединений — например, внешнее соединение нескольких таблиц, которое возвращает все их строки. Подробное описание соединений можно найти в статье на Microsoft Learn[2].

Главная тема *части I* заключалась в рассказе о способах очистки данных для эффективного их анализа. Создав прочную основу в виде чистых данных, можно переходить к следующему этапу их анализа — моделированию и созданию отчетов с помощью Power Pivot. Этому и посвящена *часть II*.

Упражнения

Потренируйтесь объединять источники данных в один запрос, используя файлы, расположенные в папке exercises\ch_05_exercises сопроводительного репозитория к этой книге[3]. В этих файлах содержится информация обо всех вылетающих рейсах из трех основных аэропортов Нью-Йорка за 2013 г.

1. Добавьте файлы ewr-flights.csv, jfk-flights.csv и lga-flights.csv, включающие записи о полетах из аэропортов Ньюарк Либерти, им. Джона Ф. Кеннеди и Ла Гуардия соответственно, в один запрос. Назовите этот запрос `flights`. (Подсказка: выберите в окне **Append** Power Query опцию **Three or more tables** (Три таблицы и более), чтобы упростить себе эту задачу).
2. Объедините этот запрос с данными файла planes.xlsx, используя левое внешнее соединение, а затем и внутреннее соединение. Назовите эти запросы `flights_left` и `flights_inner` соответственно. Сколько записей будет в каждом из них? (Подсказка: соединяйте таблицы по столбцу `tailnum`).

Готовое решение можно посмотреть в файле ch_05_exercise_solutions.xlsx, расположенном в той же папке репозитория.

[2] См. **https://clck.ru/3JoWoF**.

[3] См. **https://clck.ru/3JoaQK**.

ЧАСТЬ II

Моделирование и анализ данных с помощью Power Pivot

Знакомство с Power Pivot

Первая часть этой книги была посвящена использованию Power Query для извлечения данных из различных источников и преобразования их в наборы данных, пригодные для дальнейшего применения. При этом Power Query не используется как самостоятельный инструмент анализа данных, а является промежуточным звеном для улучшения данных перед анализом.

Чтобы продолжить знакомство с аналитикой в Excel, в *части II* книги мы рассмотрим Power Pivot — инструмент, разработанный специально для анализа данных. С помощью Power Pivot пользователи могут настраивать взаимосвязи между источниками данных и генерировать расширенные показатели, что позволяет ускорить анализ данных и создание отчетов.

Что такое Power Pivot?

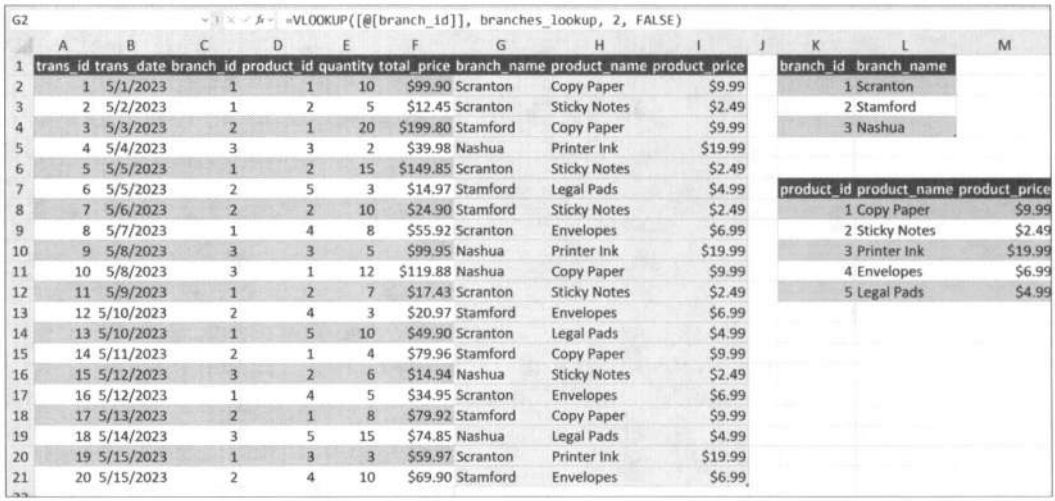
Power Pivot — это инструмент для моделирования и анализа реляционных данных, встроенный непосредственно в Excel. Он позволяет настраивать связи между несколькими таблицами и с помощью сводных таблиц создавать информационные панели (дашборды) и отчеты на основе построенной модели данных. Power Pivot предлагает широкий набор инструментов для выполнения качественного анализа, что значительно расширяет возможности Excel в области бизнес-аналитики и отчетности.

Зачем нужен Power Pivot?

Чтобы оценить эффективность Power Pivot для анализа данных в Excel, откройте из папки ch_06 сопроводительного репозитория к этой книге файл ch_06.xlsx[1]. Обратите внимание, что в этой папке нет файла с решением, поскольку все необходимые шаги уже выполнены за вас. На рабочем листе `sales` расположены три таблицы, содержащие данные о продажах, точках продаж и товарах. Предположим, вы хотите — для большей наглядности — к каждой продаже добавить соответствующие названия магазина и проданного товара.

В Excel это можно сделать несколькими способами. Самый популярный способ — использовать функцию `VLOOKUP()` (`ВПР()`) для переноса нужных значений из одной таблицы в другую (рис. 6.1).

[1] См. **https://clck.ru/3JofYV**.



G2 =VLOOKUP([@[branch_id]], branches_lookup, 2, FALSE)

trans_id	trans_date	branch_id	product_id	quantity	total_price	branch_name	product_name	product_price
1	5/1/2023	1	1	10	$99.90	Scranton	Copy Paper	$9.99
2	5/2/2023	1	2	5	$12.45	Scranton	Sticky Notes	$2.49
3	5/3/2023	2	1	20	$199.80	Stamford	Copy Paper	$9.99
4	5/4/2023	3	3	2	$39.98	Nashua	Printer Ink	$19.99
5	5/5/2023	1	2	15	$149.85	Scranton	Sticky Notes	$2.49
6	5/5/2023	2	5	3	$14.97	Stamford	Legal Pads	$4.99
7	5/6/2023	2	2	10	$24.90	Stamford	Sticky Notes	$2.49
8	5/7/2023	1	4	8	$55.92	Scranton	Envelopes	$6.99
9	5/8/2023	3	3	5	$99.95	Nashua	Printer Ink	$19.99
10	5/8/2023	3	1	12	$119.88	Nashua	Copy Paper	$9.99
11	5/9/2023	1	2	7	$17.43	Scranton	Sticky Notes	$2.49
12	5/10/2023	2	4	3	$20.97	Stamford	Envelopes	$6.99
13	5/10/2023	1	5	10	$49.90	Scranton	Legal Pads	$4.99
14	5/11/2023	2	1	4	$79.96	Stamford	Copy Paper	$9.99
15	5/12/2023	3	2	6	$14.94	Nashua	Sticky Notes	$2.49
16	5/12/2023	1	4	5	$34.95	Scranton	Envelopes	$6.99
17	5/13/2023	2	1	8	$79.92	Stamford	Copy Paper	$9.99
18	5/14/2023	3	5	15	$74.85	Nashua	Legal Pads	$4.99
19	5/15/2023	1	3	3	$59.97	Scranton	Printer Ink	$19.99
20	5/15/2023	2	4	10	$69.90	Stamford	Envelopes	$6.99

branch_id	branch_name
1	Scranton
2	Stamford
3	Nashua

product_id	product_name	product_price
1	Copy Paper	$9.99
2	Sticky Notes	$2.49
3	Printer Ink	$19.99
4	Envelopes	$6.99
5	Legal Pads	$4.99

Рис. 6.1. Объединение источников данных с помощью VLOOKUP()

Впрочем, функция VLOOKUP(), хотя и задействуется достаточно часто, имеет тем не менее свои ограничения. Как уже отмечалось в *главе 5*, результат этой поисковой функции статичен — существующая таблица просто дополняется новым столбцом, а новый источник данных не создается. Масштабирование таблиц при присоединении к ним с помощью этой функции нескольких столбцов очень трудоемкое.

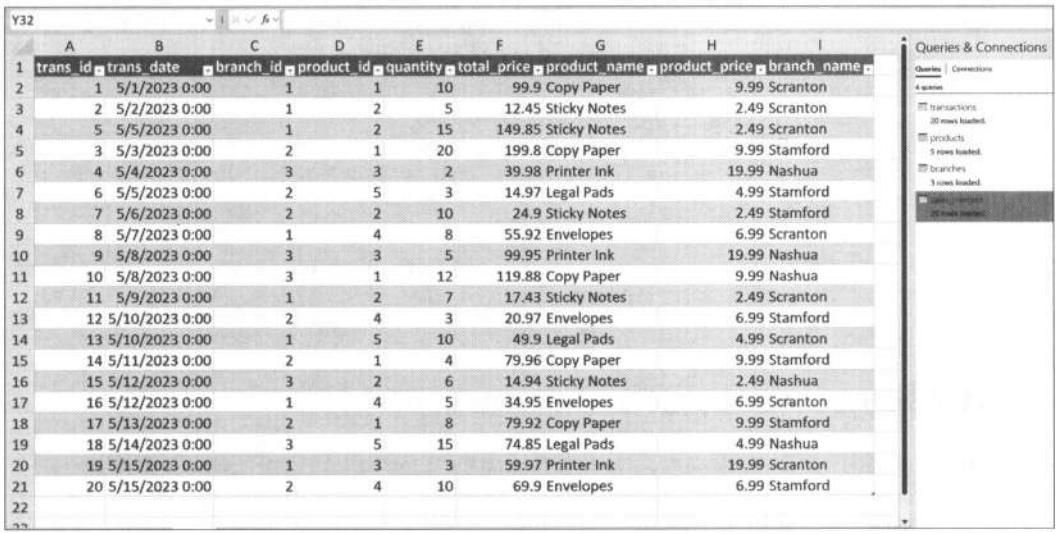
При этом использование поисковой функции означает, что Excel должен держать в памяти все данные для поиска по ним. По мере роста объема данных и увеличения количества операций поиска рабочие книги могут сильно замедлять свою работу или даже зависать. Я называю эти огромные и тяжелые наборы данных Excel «франкен-таблицами» (Frankentables).

В *главе 5* вы познакомились с более эффективным способом объединения источников данных — с помощью Power Query. Если пользоваться этим способом, то, как показано на рис. 6.2, вы получите новую таблицу без формул и с такими же размерами, как и в предыдущем способе с поисковой функцией (при условии, что применено левое внешнее соединение).

Power Query более универсален и эффективен по сравнению с поисковой функцией, но не для всех задач и он будет оптимальным выбором. Как и функция VLOOKUP(), Power Query объединяет все данные в плоскую таблицу, что приводит к увеличению размера файла и дублированию записей. Не забывайте, что основная задача Power Query — это *очистка данных*, а не их анализ. В нем нет функциональности для создания расширенных показателей, таких как расчеты с начала года (YTD, Year-To-Date) или динамические агрегации.

Для более надежного и эффективного анализа лучше объединять эти источники данных путем создания реляционной модели данных с помощью Power Pivot.

В табл. 6.1 сведены плюсы и минусы этих способов объединения источников данных.



	A	B	C	D	E	F	G	H	I
1	trans_id	trans_date	branch_id	product_id	quantity	total_price	product_name	product_price	branch_name
2	1	5/1/2023 0:00	1	1	10	99.9	Copy Paper	9.99	Scranton
3	2	5/2/2023 0:00	1	2	5	12.45	Sticky Notes	2.49	Scranton
4	5	5/5/2023 0:00	1	2	15	149.85	Sticky Notes	2.49	Scranton
5	3	5/3/2023 0:00	2	1	20	199.8	Copy Paper	9.99	Stamford
6	4	5/4/2023 0:00	3	3	2	39.98	Printer Ink	19.99	Nashua
7	6	5/5/2023 0:00	2	5	3	14.97	Legal Pads	4.99	Stamford
8	7	5/6/2023 0:00	2	2	10	24.9	Sticky Notes	2.49	Stamford
9	8	5/7/2023 0:00	1	4	8	55.92	Envelopes	6.99	Scranton
10	9	5/8/2023 0:00	3	3	5	99.95	Printer Ink	19.99	Nashua
11	10	5/8/2023 0:00	3	1	12	119.88	Copy Paper	9.99	Nashua
12	11	5/9/2023 0:00	1	2	7	17.43	Sticky Notes	2.49	Scranton
13	12	5/10/2023 0:00	2	4	3	20.97	Envelopes	6.99	Stamford
14	13	5/10/2023 0:00	1	5	10	49.9	Legal Pads	4.99	Scranton
15	14	5/11/2023 0:00	2	1	4	79.96	Copy Paper	9.99	Stamford
16	15	5/12/2023 0:00	3	2	6	14.94	Sticky Notes	2.49	Nashua
17	16	5/12/2023 0:00	1	4	5	34.95	Envelopes	6.99	Scranton
18	17	5/13/2023 0:00	2	1	8	79.92	Copy Paper	9.99	Stamford
19	18	5/14/2023 0:00	3	5	15	74.85	Legal Pads	4.99	Nashua
20	19	5/15/2023 0:00	1	3	3	59.97	Printer Ink	19.99	Scranton
21	20	5/15/2023 0:00	2	4	10	69.9	Envelopes	6.99	Stamford

Рис. 6.2. Объединение источников данных с помощью Power Query

Таблица 6.1. Сравнение способов объединения источников данных

Способ	Плюсы	Минусы
XLOOKUP()	• Легко разобраться. • Доступно в обычном Excel	• Ограниченная гибкость. • Столбцы просматриваются по одному. • Интенсивное использование памяти
Power Query	• Больше контроля над результатом. • Легче проверять и сопровождать	• Реляционные соединения могут сбивать с толку. • Дополнительная нагрузка на компьютер при загрузке данных в Power Query.
Power Pivot	• Можно создавать сложные модели данных. • Есть встроенные функции для вычислений и агрегирования	• Сложно настроить модель данных. • Тяжело освоить. • Требуется построение реляционной модели, с которой не знакомы многие пользователи Excel

Зачем тогда нужно объединение данных в Power Query, если есть Power Pivot?

Если вас смущает то, что в *главе 5* рассказывалось о превосходстве Power Query над поисковыми функциями при объединении данных, а сейчас уже Power Query отодвигается на второй план из-за Power Pivot, то для этого кажущегося избыточным количества альтернативных решений есть свои причины.

Power Query позволяет легко и эффективно решать специфические задачи, такие как беглый анализ или работа с нереляционными источниками данных. Кроме того, Power Query отлично справляется с различной степенью детализации данных. Например, если вы работаете с ежемесячными показателями продаж и ежедневными данными о трафике, Power Query

может агрегировать ежедневные показатели в итоговые за месяц, обеспечивая согласованность в степени детализации этих данных из разных источников.

Однако, хотя Power Query незаменим для решения некоторых предварительных задач и упрощений, для более сложных анализов его способностей уже не хватает. Именно здесь нам на помощь приходит Power Pivot, предлагающий расширенные возможности для выполнения реляционного моделирования. Возможность создавать в Power Pivot связи между таблицами повышает эффективность и гибкость модели данных, особенно для больших их наборов.

Оба инструмента обладают неоспоримыми преимуществами, а их совместное использование позволяет закрыть широкий спектр потребностей при работе с данными.

Power Pivot и модель данных

Power Pivot работает с *моделью данных*, в которой устанавливаются и настраиваются взаимосвязи. Такой подход позволяет создавать сводную таблицу из нескольких источников без их физического объединения.

Используя формульный язык DAX, с помощью Power Pivot можно выполнять сложные вычисления на основе модели данных, включая временной анализ, ранжирование, определение процентилей и многое другое.

Основное преимущество Power Pivot заключается в способности эффективно управлять многочисленными источниками данных. Он не хранит тяжелые «франкен-таблицы», занимающие много памяти, и вычисляет меры DAX по мере необходимости. Однако Power Pivot может показаться сложным для освоения из-за «крутой кривой обучения» пользования им, особенно при работе с источниками данных, не объединенными в одну таблицу.

В файле ch_06.xlsx я создал модель данных из трех источников данных о продажах и загрузил результаты в сводную таблицу на рабочем листе `sales_pp` (рис. 6.3). Теперь я могу анализировать данные и проводить вычисления на основе этих связанных таблиц.

Рис. 6.3. Объединение источников данных с помощью связей в Power Pivot

Дублирование имен таблиц в сводной таблице Power Pivot

В примерах с Power Pivot, приведенных в этой книге, в настройках итоговой сводной таблицы название каждой таблицы выводится дважды: один раз со значком в виде оранжевого цилиндра, а второй раз — без него. Всегда выбирайте таблицы с этим значком, поскольку они напрямую связаны с моделью данных и включают в себя все созданные меры. Если бы вы импортировали таблицу в модель данных из внешнего источника, а не из текущей рабочей книги, такого дублирования таблиц не было бы.

Чтобы настраивать связи между таблицами или добавлять такие функциональности, как вычисляемые столбцы или меры, сначала необходимо в Excel подключить надстройку Power Pivot.

Подключение надстройки Power Pivot

Чтобы получить доступ к Power Pivot, перейдите на ленте на вкладку **File** и выберите **Options | Add-ins** (Параметры | Надстройки). На панели **Add-ins** (Надстройки) в раскрывающемся списке **Manage** (Управление) выберите **COM Add-ins** (Надстройки COM) и нажмите кнопку **Go** (Перейти), как показано на рис. 6.4.

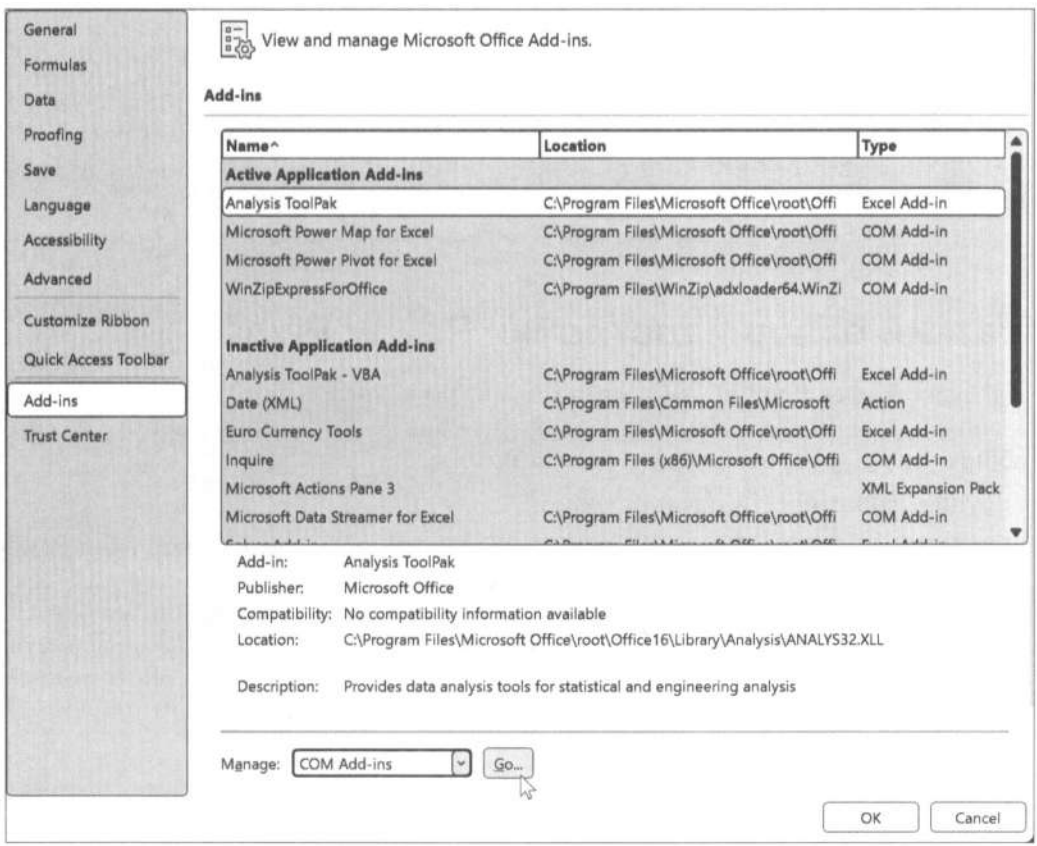


Рис. 6.4. Подключение надстройки Power Pivot

В открывшемся диалоговом окне **COM Add-ins** (Надстройки COM), показанном на рис. 6.5, установите флажок **Microsoft Power Pivot for Excel** и нажмите кнопку **OK**. В результате на ленте Excel должна появиться новая вкладка **Power Pivot** (рис. 6.6).

Теперь вы готовы к работе с Power Pivot.

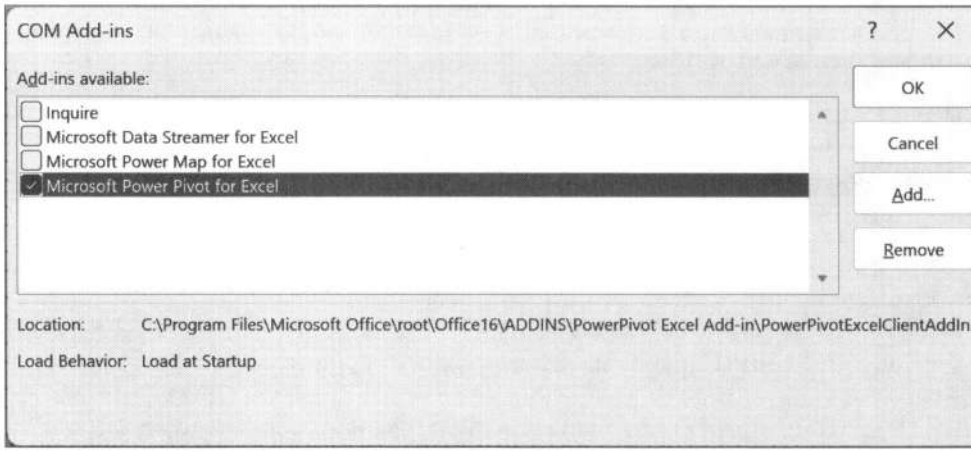


Рис. 6.5. Выбор надстройки Power Pivot

Рис. 6.6. Надстройка Power Pivot на ленте Excel

Краткий обзор надстройки Power Pivot

На вкладке **Power Pivot** ленты сосредоточены различные опции для создания и изменения вашей модели данных, а также связанные функциональности. Давайте в общих чертах рассмотрим каждую из этих опций.

- Группа **Data Model** (Модель данных)

  При выборе опции **Manage** (Управление) открывается специальное окно Power Pivot, отображающее таблицы в вашей модели данных. Этот редактор позволяет визуализировать связи между показанными таблицами и предоставляет другие функциональные возможности. Потратьте немного времени на знакомство с этим окном и просто закройте его по завершении. По мере изучения следующих глав *части II* мы еще встретимся с этим интерфейсом.

- Группа **Calculations** (Вычисления)

  С помощью опций группы **Calculations** (Вычисления) вкладки **Power Pivot** можно создавать вычисляемые меры и ключевые показатели эффективности (KPI), о которых мы более подробно расскажем в других главах *части II*:

- **Measures** (Меры).

  В мерах Power Pivot используется язык DAX для выполнения вычислений, агрегирования данных, проведения сложной обработки данных и глубокого анализа. С их помощью можно агрегировать значения, вычислять итоговые суммы, средние и процентные значения, они также имеют важнейшее значение для расширенной аналитики в Excel.

- **KPIs** (Ключевые показатели эффективности).

  KPI (Key Performance Indicators), или *ключевые показатели эффективности* — это измеряемые величины, показывающие, насколько эффективно компания или организация достигает своих основных бизнес-целей. Показатели KPI необходимы для оценки успешности в достижении целей, они играют важную роль в мониторинге прогресса и в управленческом процессе принятия решений. Power Pivot позволяет пользователям создавать свои ключевые показатели эффективности и отображать их в сводных таблицах и отчетах.

♦ Группа **Tables** (Таблицы).

  Этот инструмент вкладки **Power Pivot** позволяет импортировать таблицу из рабочей книги в модель данных. Впрочем, импортировать данные всё же рекомендуется с помощью Power Query, о чем будет рассказано в *главе 7*. Power Query предоставляет возможность подключаться к более широкому спектру источников данных, таких как внешние рабочие книги и файлы формата CSV, о которых говорилось в *части I*, а также позволяет выполнять очистку данных перед созданием модели данных.

♦ Группа **Relationships** (Связи).

  Этот инструмент позволяет автоматически определять и создавать связи между таблицами в рамках модели данных. И хотя он весьма полезен, и с ним вам стоит разобраться самостоятельно после изучения основ, приведенных в этой книге, очень важно самим уметь адекватно оценивать, насколько точно была построена модель данных. Именно поэтому мы будем создавать связи вручную, а не полагаться на эту автоматическую опцию.

♦ Опция **Settings** (Параметры).

  Параметры этой опции позволяют повысить производительность расчета модели данных и найти потенциальные проблемы. Их использование выходит за рамки книги.

Заключение

В этой главе рассказано о возможностях инструмента Power Pivot управлять данными из нескольких источников, не объединяя их в одну таблицу, при этом он считается отличным решением для борьбы с огромными «франкен-таблицами», а также, как и Power Query, развеивает распространенные мифы об Excel. И хотя работа

с Power Pivot может пугать своей сложностью, особенно обычных пользователей Excel, его возможности не имеют себе равных.

Power Pivot упрощает нам процесс поиска информации, принятия обоснованных решений и проведения сложных анализов в Excel. В последующих главах *части II* мы подробнее познакомимся с Power Pivot и разберемся в тонкостях создания и анализа модели данных.

Упражнения

Чтобы проверить свое понимание темы, рассмотренной в этой главе, ответьте на следующие вопросы:

1. Зачем подключать надстройку Power Pivot и что она позволяет нам делать?
2. Объясните роль модели данных в Power Pivot и ее значение для анализа данных.
3. В чем основная роль мер DAX и ключевых показателей эффективности в Power Pivot?
4. В чем разница между соединениями Power Query и взаимосвязями Power Pivot с точки зрения объединения источников данных?
5. Какие недостатки есть у использования поисковых функций, таких как `VLOOKUP()` или `XLOOKUP()`, для объединения таблиц в Excel?

Примеры ответов на эти вопросы можно найти в папке exercises\ch_06_exercises сопроводительного репозитория к этой книге[2].

[2] См. **https://clck.ru/3Jouhg**.

Создание реляционной модели данных в Power Pivot

В *главе 6* вы познакомились с основами Power Pivot — эффективного инструмента для анализа данных и создания отчетов, особенно при работе с несколькими источниками данных. В этой главе мы рассмотрим, как использовать Power Pivot для построения реляционной модели данных.

Подключение данных к Power Pivot

Как уже отмечалось в *главе 6*, модель данных служит основой для Power Pivot, помогая создавать и управлять связями между таблицами для выполнения эффективных расчетов и анализа данных. Power Pivot упрощает эту задачу с помощью интуитивно понятного интерфейса с возможностью перетаскивания объектов. В этой главе мы более подробно рассмотрим модель данных, используя файл ch_07.xlsx, расположенный в папке ch_07 сопроводительного репозитория к этой книге[1] и содержащий набор данных о розничных продажах, на который часто ссылаются в аналитическом сообществе.

В примере, приведенном в *главе 6*, модель данных была определена заранее. В этой главе нам нужно будет определить ее вручную.

Несмотря на то что в Power Pivot можно создавать прямые подключения к источникам данных, рекомендуется сначала загрузить данные в Power Query. Такой подход обеспечивает удобную рабочую платформу, позволяющую вам запускать любые повторяющиеся процессы по очистке данных в таблицах, когда это необходимо.

Чтобы начать работу, импортируйте таблицу `orders` в Power Query с помощью опции **Data | From Table/Range**. Пропуская все действия по преобразованию данных, сразу выберите на вкладке **Home** опцию **Close & Load | Close & Load To**.

Чтобы загрузить этот запрос из Power Query в Power Pivot, выберите вариант **Only Create Connection** (Только создать подключение), а затем установите флажок **Add this data to the Data Model** (Добавить эти данные в модель данных) и нажмите кнопку **OK** (рис. 7.1).

После этого запрос станет доступен для моделирования данных в Power Pivot, но отдельный рабочий лист создан не будет. Важно отметить, что основная цель Power Pivot — связать загруженную таблицу с другими таблицами, создать меры DAX

[1] См. **https://clck.ru/3JoxJA**.

и т. д. Привычная загрузка данных в рабочую книгу не подходит для выполнения этих действий.

Повторите указанные шаги для таблиц `returns` и `users` и убедитесь, что у вас есть три запроса, загруженные в рабочую книгу только как подключение и добавленные в модель данных.

Рис. 7.1. Загрузка запроса из Power Query в Power Pivot

Создание взаимосвязей между таблицами

В Power Pivot связи между таблицами упрощают анализ данных, позволяя создавать сложные модели без обычного трудоемкого объединения данных. Такой подход повышает согласованность, сокращает избыточность данных и упрощает работу с набором данных. Благодаря установленным связям пользователи могут выполнять динамический и интерактивный анализ данных, пользуясь современными аналитическими возможностями Excel.

Чтобы задать связи между таблицами `orders`, `returns` и `users`, на вкладке **Power Pivot** ленты Excel выберите **Manage** (Управление) и в группе **View** (Просмотр) — **Diagram View** (Представление диаграммы). Три наши таблицы и названия их столбцов будут отображены в виде диаграммы (рис. 7.2).

Ничего страшного, если ваши таблицы отобразятся не в таком порядке, как показано на рис. 7.2, — мы зададим связи между этими таблицами, которые будут работать независимо от их расположения на диаграмме. Как только с помощью связей нам удастся разобраться с содержанием наших таблиц, мы визуально преобразуем эту диаграмму в более логичную и эффективную схему.

Чтобы создать первую связь, выделите таблицу `orders`. Затем на ленте Power Pivot перейдите на вкладку **Design** (Конструктор) и в группе **Relationships** (Связи) выберите **Create Relationship** (Создание связи), как показано на рис. 7.3.

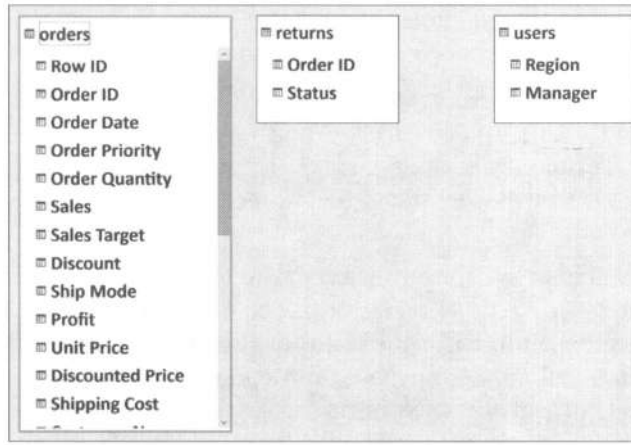


Рис. 7.2. Вид диаграммы в редакторе Power Pivot

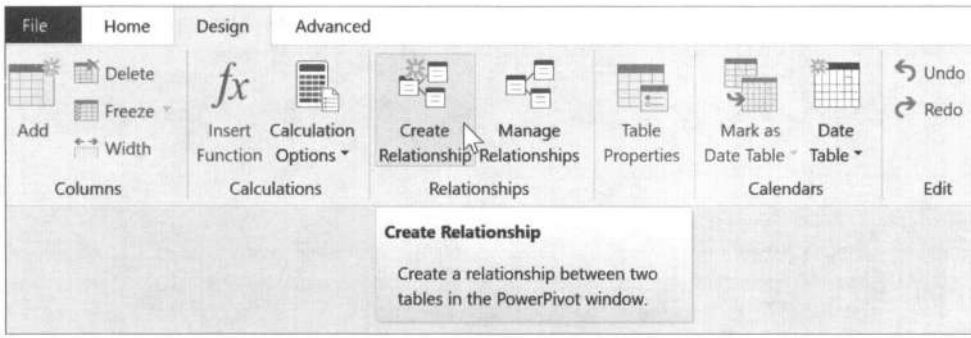


Рис. 7.3. Создание связи в Power Pivot

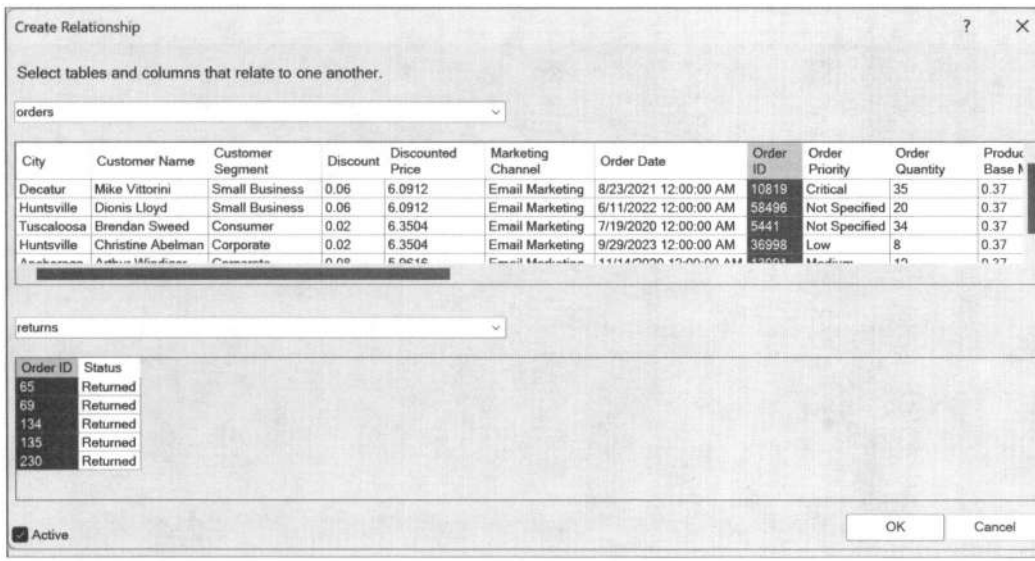


Рис. 7.4. Создание связи между таблицами `orders` и `returns`

Чтобы задать связь между таблицами `orders` и `returns`, во втором раскрывающемся списке открывшегося диалогового окна выберите таблицу `returns` и выделите в обеих таблицах столбец `Order ID` (рис. 7.4). Завершите процесс, нажав кнопку **OK**.

Как и в функции `VLOOKUP()`, взаимосвязь строится на общих столбцах таблиц. В нашем случае общим столбцом является `Order ID`. После установки этой связи и нажатия кнопки **OK** на диаграмме появится линия, соединяющая эти две таблицы (рис. 7.5).

Чтобы установить последнюю взаимосвязь в модели данных и связать все три таблицы, можно использовать поле `Region`, которое есть и в `orders`, и в `users`. При этом вместо использования опции **Create Relationship** нам быстрее будет перетащить поле `Region` от одной таблицы к другой с помощью курсора мыши (рис. 7.6) — это простое действие и создаст нужную связь.

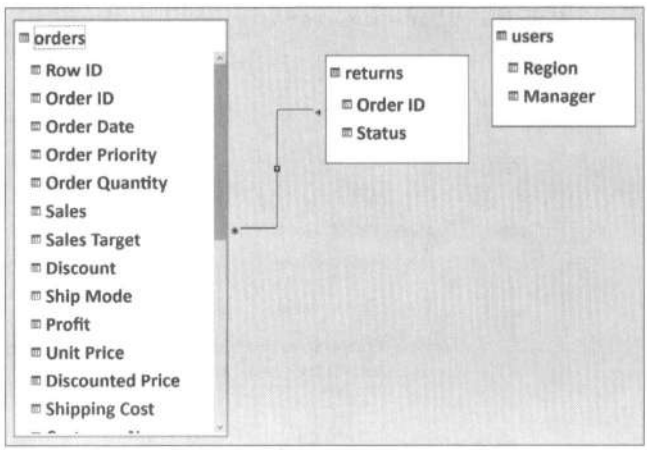


Рис. 7.5. Созданная связь между таблицами `orders` и `returns` на диаграмме

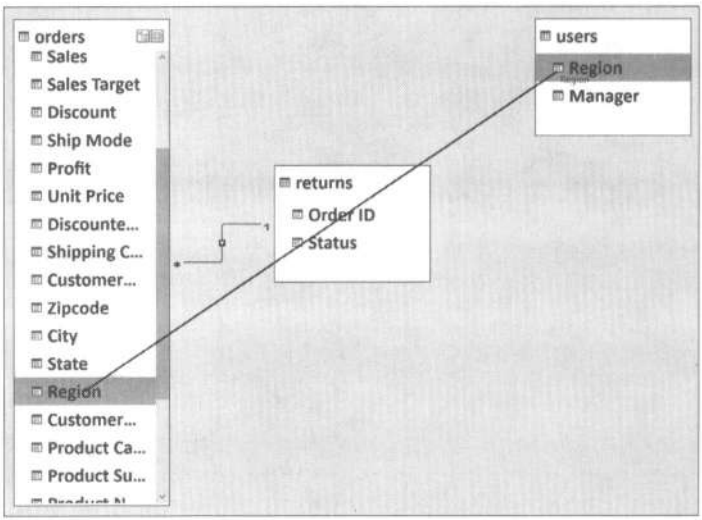


Рис. 7.6. Создание связи между таблицами `orders` и `users` с помощью перетаскивания

Таблицы фактов и таблицы измерений

Создав целостную модель данных, на следующем шаге нам нужно определить, какие таблицы являются таблицами фактов, а какие — таблицами измерений. *Таблицы фактов* (fact tables) обычно содержат количественные данные, которые можно использовать для вычислений, — например, среднего, минимального и максимального значений. *Таблицы измерений* (dimension tables), наоборот, содержат описательные данные, которые обеспечивают контекст для значений из таблиц фактов.

Например, таблица `orders` состоит из нескольких измеряемых величин, таких как продажи, прибыль и количество проданного товара, которые можно суммировать, усреднять и пр. Эти величины представляют собой основные показатели бизнес-процесса, который вы анализируете. Наличие таких количественных данных указывает на то, что это таблица фактов.

В таблицах фактов часто отсутствует описательная информация, которая может иметь критическое значение для интерпретации данных. Так, для таблицы `orders` полезно будет знать, какой менеджер компании за какой регион отвечает. Таблица `users`, таким образом, является таблицей измерений, поскольку она представляет собой описательный контекст, указывая, кто из менеджеров работает по конкретному региону. Таблицы измерений играют важнейшую роль в получении срезов данных и в более глубоком анализе.

Упорядочивание диаграммы

В реальных проектах часто приходится сталкиваться с моделями данных, состоящими из десяти и более таблиц. Правильное упорядочивание диаграммы имеет критическое значение для того, чтобы пользователи могли эффективно разобраться в ней.

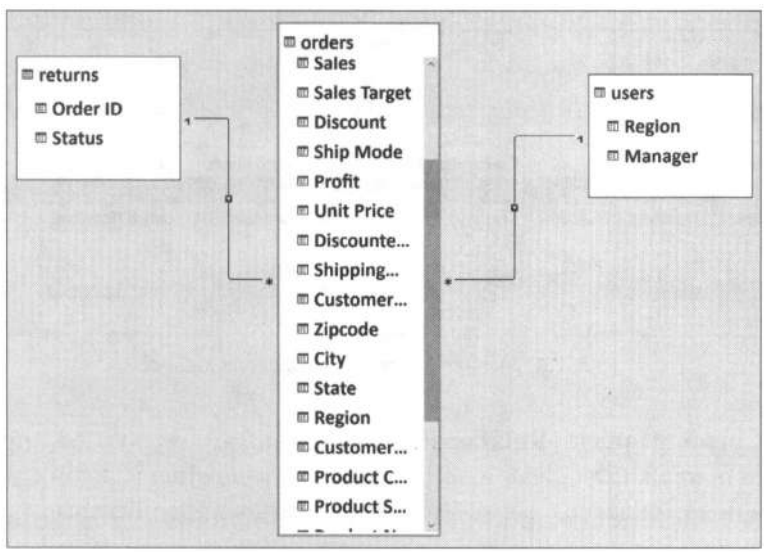


Рис. 7.7. Модель данных в виде упорядоченной диаграммы

Один из полезных приемов — расположить таблицу фактов в центре диаграммы, а вокруг нее поместить таблицы измерений. Такое визуальное расположение помогает понять взаимосвязи и зависимости между таблицами. Для этого перетащите таблицы `returns` и `users` так, чтобы они оказались по разные стороны от таблицы `orders` (рис. 7.7).

Когда в модели данных центральное место занимает таблица фактов, а вокруг нее размещены таблицы измерений, как показано на рис. 7.7, такое расположение называется схемой «*звезда*». Схема «звезда» — это базовая концепция в проектировании моделей данных. Она получила свое название из-за визуального сходства, при котором таблица фактов является центром «звезды», а таблицы измерений расходятся в разные стороны, имитируя ее лучи.

Редактирование связей

В Power Pivot вы можете отредактировать любую установленную взаимосвязь между таблицами несколькими способами. Прежде всего, можно щелкнуть правой кнопкой мыши на любой линии связи в диаграмме и из контекстного меню выбрать **Edit Relationship** (Изменить связь). Откроется знакомое диалоговое окно, в котором можно изменить связи таблиц и столбцов. Кроме того, с помощью этого же контекстного меню можно временно отключить или удалить связь, выбрав соответствующую опцию.

Вы также можете централизованно управлять всеми связями в модели данных из одного окна. Для этого на ленте перейдите на вкладку **Design** (Конструктор) и выберите **Manage Relationships** (Управление связями). В открывшемся диалоговом окне (рис. 7.8) будет выведен полный список всех связей в вашей модели данных с возможностью вносить изменения в любую связь.

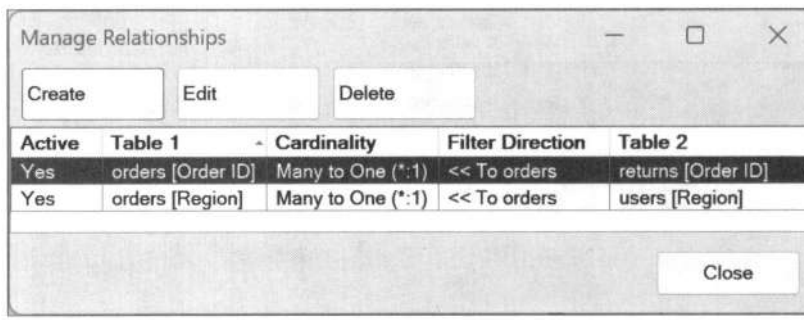


Рис. 7.8. Управление связями в модели данных

Диалоговое окно **Manage Relationships**, показанное на рис. 7.8, предоставляет информацию о *кардинальности* и *направлении фильтрации* каждой связи. Мы рассмотрим эти понятия далее в соответствующих разделах этой главы.

Загрузка результатов в Excel

После того как модель данных создана, следующим нашим шагом будет перенос результатов в Excel. Для этого в редакторе Power Pivot перейдите на вкладку **Home** (Главная) и нажмите на раскрывающееся меню **PivotTable** (Сводная таблица), как показано на рис. 7.9.

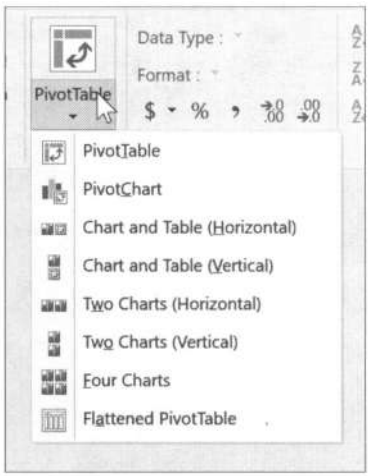


Рис. 7.9. Варианты выгрузки из Power Pivot

Power Pivot предлагает для загрузки сводной таблицы в рабочую книгу различные конфигурации. Из этих конфигураций чаще всего используются сводная таблица и сводная диаграмма, поскольку Power Pivot обычно задействуется именно для создания базовых информационных панелей и отчетов. Последний вариант — **Flattened PivotTable** (Плоская сводная таблица) — удаляет все промежуточные итоги и выводит данные в простом табличном, не древовидном формате.

Выберите в меню, показанном на рис. 7.9, опцию **PivotTable** (Сводная таблица) и в открывшемся диалоговом окне **Create PivotTable** (Создать сводную таблицу) нажмите кнопку **OK**, чтобы добавить сводную таблицу на новый рабочий лист. У вас должно получиться что-то похожее на рис. 7.10.

Теперь в окне **PivotTable Fields** (Поля сводной таблицы) перетащите поле `Region` из таблицы `users` в область **Rows** (Строки), а поле `Sales` из таблицы `orders` в область **Values** (Значения). Для выполнения точных вычислений сразу же будет применена модель данных с заданными связями между этими таблицами и, в частности, организована связь по общему полю `Region`. Результаты вычислений вы можете увидеть на рис. 7.11.

В сводной таблице результаты по полю `Sales` непривычно округлены до трех десятичных знаков, и формат не выводит символ валюты, что делает эти результаты трудночитаемыми. Отображение этих чисел можно исправить в рабочем листе или в сводной таблице, но надежнее будет задать формат в модели данных Power Pivot. Для этого на ленте Excel вернитесь на вкладку **Power Pivot** и выберите **Manage**.

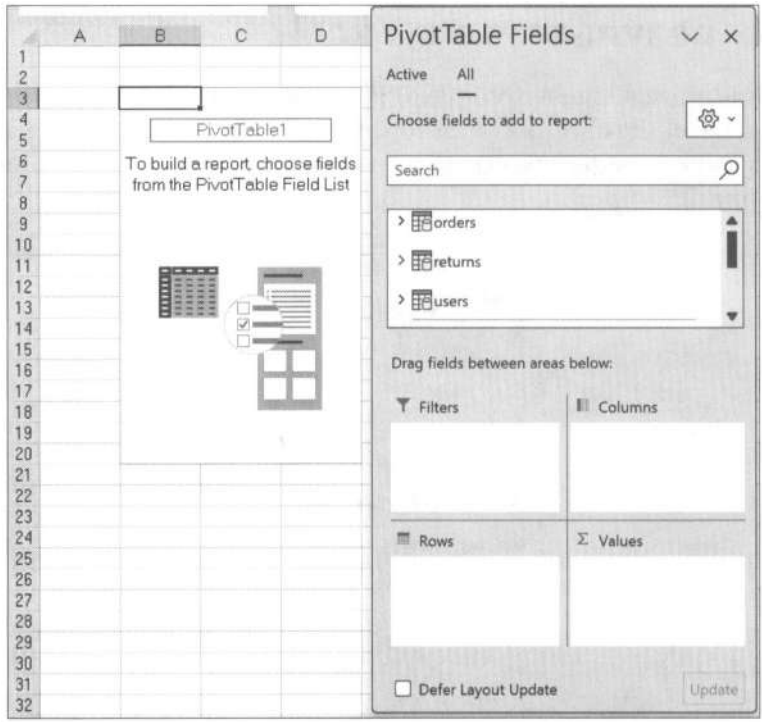


Рис. 7.10. Сводная таблица, сгенерированная Power Pivot

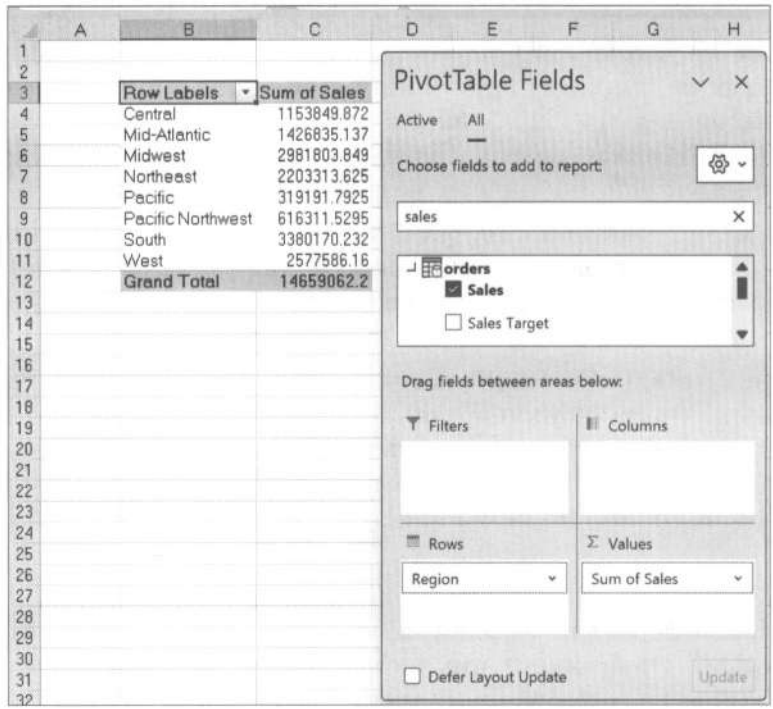
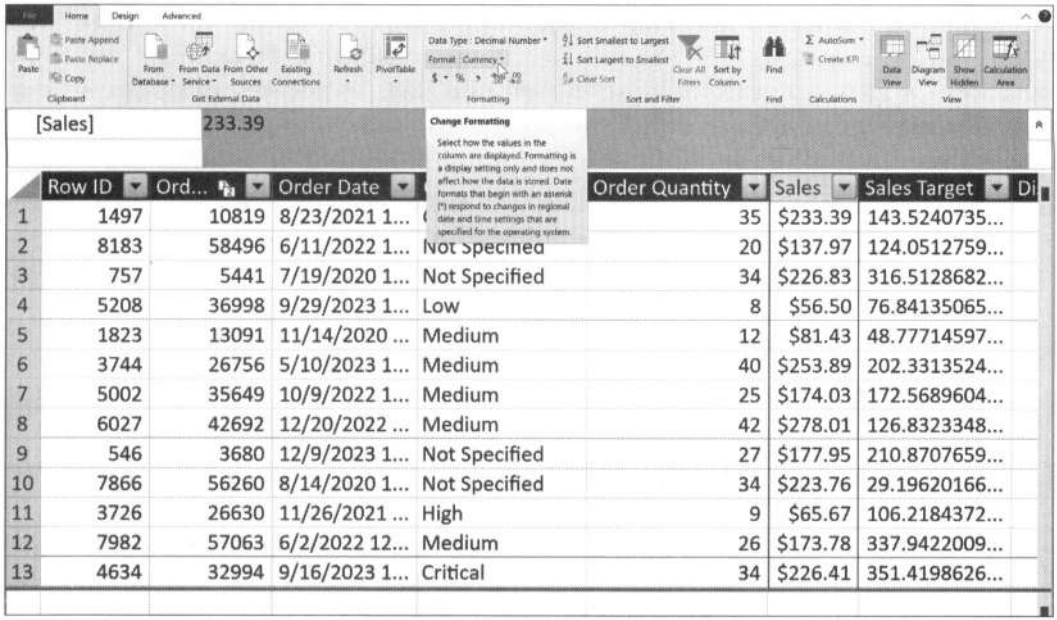


Рис. 7.11. Сводная таблица, созданная на основе нескольких таблиц



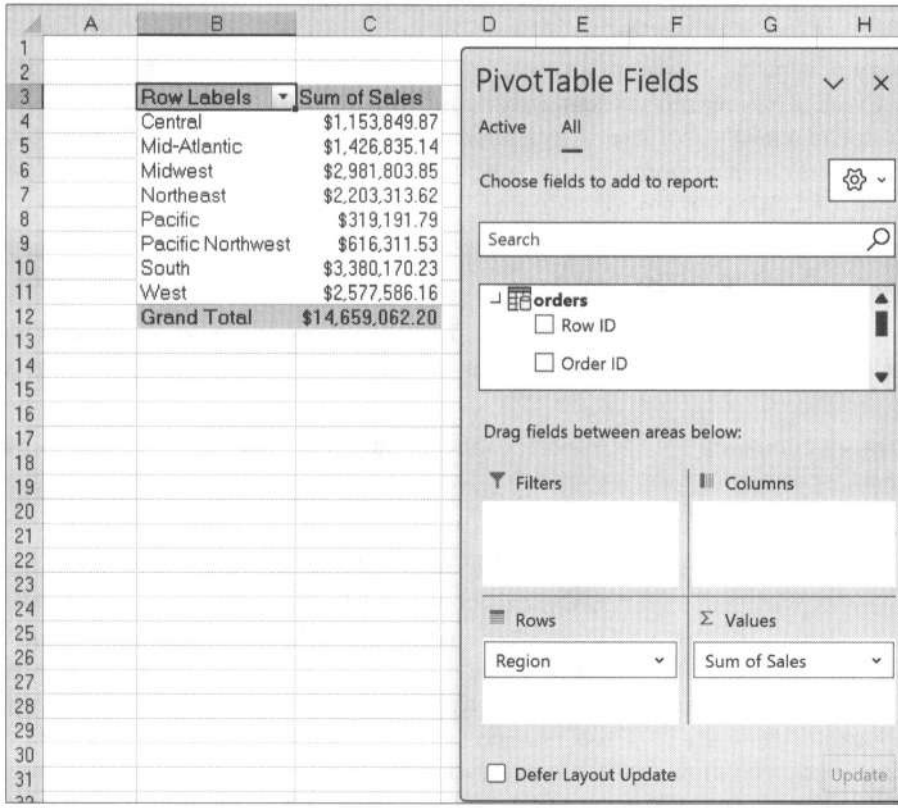
[Sales] 233.39

Change Formatting

Select how the values in the column are displayed. Formatting is a display setting only and does not affect how the data is stored. Date formats that begin with an asterisk (*) respond to changes in regional date and time settings that are specified for the operating system.

	Row ID	Ord...	Order Date		Order Quantity	Sales	Sales Target	Di
1	1497	10819	8/23/2021 1...	(	35	$233.39	143.5240735...	
2	8183	58496	6/11/2022 1...	Not Specified	20	$137.97	124.0512759...	
3	757	5441	7/19/2020 1...	Not Specified	34	$226.83	316.5128682...	
4	5208	36998	9/29/2023 1...	Low	8	$56.50	76.84135065...	
5	1823	13091	11/14/2020 ...	Medium	12	$81.43	48.77714597...	
6	3744	26756	5/10/2023 1...	Medium	40	$253.89	202.3313524...	
7	5002	35649	10/9/2022 1...	Medium	25	$174.03	172.5689604...	
8	6027	42692	12/20/2022 ...	Medium	42	$278.01	126.8323348...	
9	546	3680	12/9/2023 1...	Not Specified	27	$177.95	210.8707659...	
10	7866	56260	8/14/2020 1...	Not Specified	34	$223.76	29.19620166...	
11	3726	26630	11/26/2021 ...	High	9	$65.67	106.2184372...	
12	7982	57063	6/2/2022 12...	Medium	26	$173.78	337.9422009...	
13	4634	32994	9/16/2023 1...	Critical	34	$226.41	351.4198626...	

Рис. 7.12. Форматирование столбца в Power Pivot



Row Labels	Sum of Sales
Central	$1,153,849.87
Mid-Atlantic	$1,426,835.14
Midwest	$2,981,803.85
Northeast	$2,203,313.62
Pacific	$319,191.79
Pacific Northwest	$616,311.53
South	$3,380,170.23
West	$2,577,586.16
Grand Total	$14,659,062.20

PivotTable Fields

Active All

Choose fields to add to report:

Search

orders

Row ID

Order ID

Drag fields between areas below:

Filters

Columns

Rows

Region

Values

Sum of Sales

Defer Layout Update

Update

Рис. 7.13. Результат форматирования столбца в Power Pivot

Затем в Power Pivot в группе **View** (Просмотр) вкладки **Home** выберите **Data View** (Представление данных). В нижней части представления **Data View** выберите вкладку `orders`, чтобы далее работать с этой таблицей, выделите столбец `Sales`, укажите для него формат **Currency** (Валюта), а также включите разделитель разрядов (тысяч), как показано на рис. 7.12.

Здесь же вы можете внести в форматирование ваших данных и любые другие изменения.

После выхода из Power Pivot эти изменения в формате сразу применятся к сводной таблице (рис. 7.13).

Далее в этой книге мы будем вносить изменения в форматирование столбцов в Power Pivot, не останавливаясь на подробном описании выполненных действий.

Понятие кардинальности

С результатами Power Pivot могут возникать проблемы, когда агрегирование чисел не работает должным образом или когда из-за добавленной связи некоторые поля в модели данных становятся непригодными для использования. Такие проблемы, как правило, возникают из-за неполного понимания структуры модели данных и ее *кардинальности* (от *англ.* cardinality). Давайте подробнее остановимся на этом понятии.

В предыдущем разделе говорилось о значимости общих полей для создания связей в Power Pivot. Количество уникальных записей в каждой таблице играет ключевую роль в определении того, как будут работать связи в модели данных. *Кардинальность* в нашем случае определяется тем, какое количество записей в одной таблице соответствует записям в другой таблице.

Связь «один к одному»

Связь *«один к одному»* представляет собой самый простой вид кардинальности, когда каждая запись в таблице однозначно соответствует одной записи в другой таблице.

Рассмотрим сценарий, в котором модель данных состоит из двух таблиц: `product_details` и `supplier_details` (рис. 7.14).

Как можно здесь видеть, каждая запись имеет уникальный `Product ID`, по которому и строится связь между двумя таблицами.

Хотя такая структура может казаться удобной, но она, как правило, не самая эффективная. Таблицы, связанные по схеме «один к одному», можно объединить, что позволит минимизировать дублирование данных, снизить затраты на сопровождение и улучшить производительность. Кроме того, Power Pivot как инструмент моделирования данных в Excel не позволяет настроить кардинальность «один к одному», что ограничивает его использование при работе с реальными моделями данных. Однако Power Pivot оптимизирован для работы со связями «один ко многим».

product_details

Product ID	Product Name	Category	Sub-Category	Price ($)
P001	Adjustable Desk	Furniture	Tables	200.00
P002	Executive Chair	Furniture	Chairs	120.00
P003	Ballpoint Pen (Blue)	Office Supplies	Pens	1.00
P004	Printer Paper (500 sheets)	Office Supplies	Paper	5.00
P005	Monitor 24"	Technology	Monitors	150.00

supplier_details

Product ID	Supplier ID	Supplier Name	Address	Contact Number
P001	S001	FurniFix Inc.	123 Furniture St, NY	(123) 456-7890
P002	S002	ChairCrafters Ltd.	456 Chair Lane, LA	(234) 567-8901
P003	S003	PenMasters Corp.	789 Pen Ave, SF	(345) 678-9012
P004	S004	PaperStacks	101 Paper Rd, TX	(456) 789-0123
P005	S005	TechBrite	202 Tech Blvd, MI	(567) 890-1234

Рис. 7.14. Пример связи «один к одному»

Связь «один ко многим»

Связь «*один ко многим*» означает, что нескольким записям в таблице соответствует только одна запись в другой таблице.

customers

Customer ID	Customer Name	Location
C001	Alice Smith	New York, NY
C002	Bob Johnson	Los Angeles, CA
C003	Charlie Brown	Chicago, IL

orders

Order ID	Product	Order Date	Amount ($)	Customer ID
O001	Adjustable Desk	2023-01-05	200.00	C001
O002	Ballpoint Pen (Blue)	2023-01-10	1.00	C001
O003	Executive Chair	2023-01-15	120.00	C002
O004	Monitor 24"	2023-01-20	150.00	C002
O005	Printer Paper (500 sheets)	2023-01-25	5.00	C003

Рис. 7.15. Пример связи «один ко многим»

В примере, приведенном на рис. 7.15, покупатель из первой таблицы может иметь несколько связанных с ним записей в другой таблице, например в таблице заказов. Храня такие связанные записи в отдельной таблице и присоединяя их с помощью одного столбца, можно уменьшить избыточность данных, оптимизировать выполнение запросов и обеспечить целостность данных. Такой подход является очень эффективным при создании масштабируемых и поддерживаемых баз данных, которые смогут точно отразить всю сложность бизнес-процессов.

Связь «многие ко многим»

В случаях, когда сущности из двух разных таблиц могут образовывать множество соединений, реализуется связь *«многие ко многим»*. Однако инструменты типа Power Pivot напрямую не поддерживают этот вид связи, и общим подходом к управлению такими связями является использование *таблицы-моста* или *таблицы связей*.

Давайте рассмотрим возможность отслеживания заказов покупателей в рамках нескольких рекламных акций в розничных магазинах (рис. 7.16).

customers

Customer ID	Customer Name
C101	Emily White
C102	Daniel Green
C103	Laura Blue

promotions

Promotion ID	Promotion Name	Date
P101	Summer Sale	2023-06-15
P102	Black Friday	2023-11-24
P103	New Year Bonanza	2024-01-01

Рис. 7.16. Пример связи «многие ко многим»

В этом примере пока только приведены покупатели и рекламные акции, и очевидно, что каждый покупатель может сделать несколько заказов в рамках одной акции. Чтобы справиться с этой сложностью, мы введем промежуточную таблицу, в которой будет указано, какие покупатели в каких акциях участвовали (рис. 7.17).

customer-promotion (Bridge table)

Association ID	Customer ID	Promotion ID
A201	C101	P101
A202	C101	P102
A203	C102	P102
A204	C103	P103

Рис. 7.17. Пример таблицы-моста для связи «многие ко многим»

Такая таблица упрощает реализацию связи «многие ко многим», храня записи об участии каждого покупателя в конкретной рекламной акции.

Почему так важна кардинальность?

Кардинальность играет ключевую роль в моделировании данных, обеспечивая их точность и согласованность. При связи «один ко многим» важно убедиться, что каждая «одна» сущность из первой таблицы однозначно соответствует «многим» сущностям из второй таблицы, и наоборот.

Хотя Power Pivot не проводит различий между связями «один к одному» и «один ко многим», желательно, чтобы вы имели об этом определенное представление — хотя бы для повышения производительности модели данных при работе с инстру-

ментом Power BI, который умеет их различать. За более подробной информацией об использовании связей в Power BI обратитесь к документации Microsoft[2].

Понимание разных видов кардинальности: «один к одному», «один ко многим» и «многие ко многим» — очень важно для работы со всеми инструментами моделирования данных, а не только с Power Pivot. Хотя Power Pivot акцентируется на связи «один ко многим», нужно знать обо всех видах кардинальности, чтобы обеспечивать упорядоченность данных, сохранять их целостность и безболезненно внедрять новые инструменты. Это знание будет особенно ценно при устранении ошибок и эффективном взаимодействии с коллегами по работе с данными. Короче говоря, глубокое понимание принципов кардинальности позволяет легко адаптироваться к совершенно разнообразным данным.

Направление фильтрации

Работая с реляционной моделью данных, Power Pivot упрощает анализ данных из нескольких таблиц за счет использования их общих полей. Фильтрация по этим полям влияет на связанные таблицы, что и заложено в понятии *направление фильтрации*, которое неразрывно связано с кардинальностью.

На той же диаграмме, которую мы использовали ранее, отображена связь между таблицами `users` и `orders` через поле `Region`. Если внимательно рассмотреть линию, которая обозначает эту связь, можно заметить на ней маленькую стрелку, указывающую направление от `users` к `orders` (рис. 7.18).

Стрелка эта определяет направление передачи фильтров из одной таблицы в другую. Так, применение фильтра к правой таблице повлияет на левую таблицу, но не наоборот.

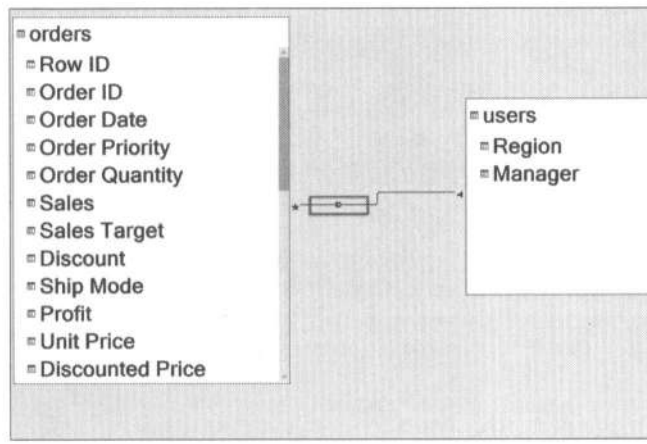


Рис. 7.18. Направление фильтрации от `users` к `orders`

[2] См. **https://clck.ru/3JuU6A**.

Символ звездочка (*) на рис. 7.18 указывает на сторону «многие» для связи «один ко многим» между таблицами. Это визуальное обозначение позволяет быстро понять характер и кардинальность связей между таблицами.

Фильтрация *orders* через *users*

Чтобы понять, как фильтрация таблицы `users` влияет на таблицу `orders`, вставьте в рабочую книгу сводную таблицу из модели данных. Добавьте поле `Region` из таблицы `users` в область **Filters** (Фильтры) и поле `Sum of Sales` (`Сумма по столбцу Sales`) из таблицы `orders` в область **Values** (Значения). При выборе значения в поле `Region` — например, `Central`, в сводной таблице будет выведена сумма продаж только для центрального региона (рис. 7.19).

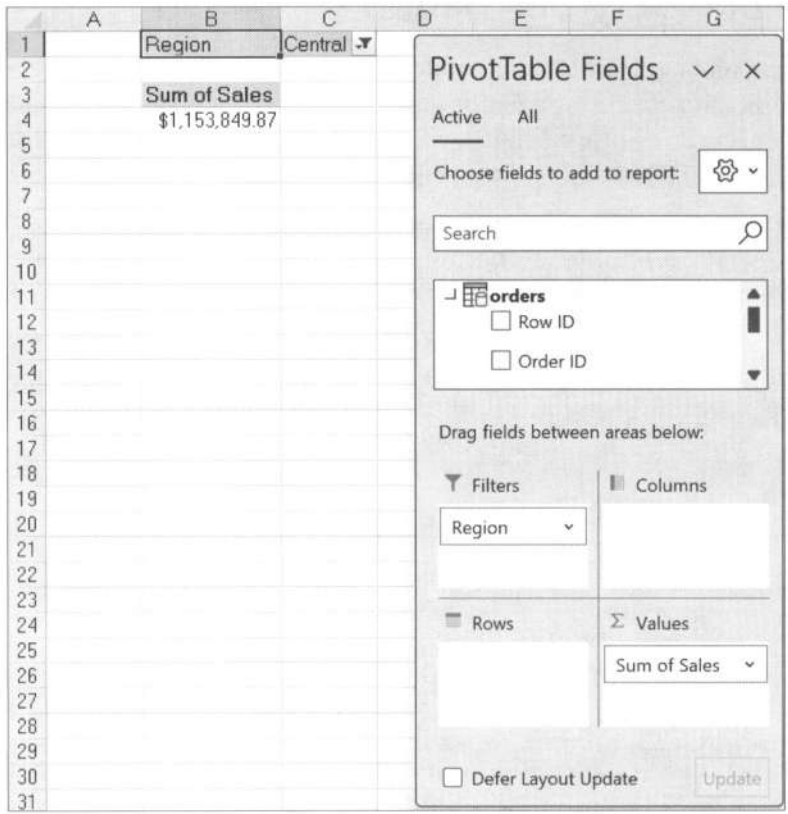


Рис. 7.19. Фильтрация `orders` через таблицу `users`

Технически это означает, что фильтр «спускается» из таблицы `users` в таблицу `orders`. Это ожидаемое поведение при использовании фильтра, с которым вы уже, вероятно, знакомы.

Фильтрация *users* через *orders*

Теперь рассмотрим другую сводную таблицу, в которой поле `Region` из таблицы `orders` помещено в область **Filters**, а поле `Manager` из таблицы `users` — в область **Rows**.

При фильтрации по центральному региону происходит что-то странное: данные не меняются, ни одна запись не пропала (рис. 7.20).

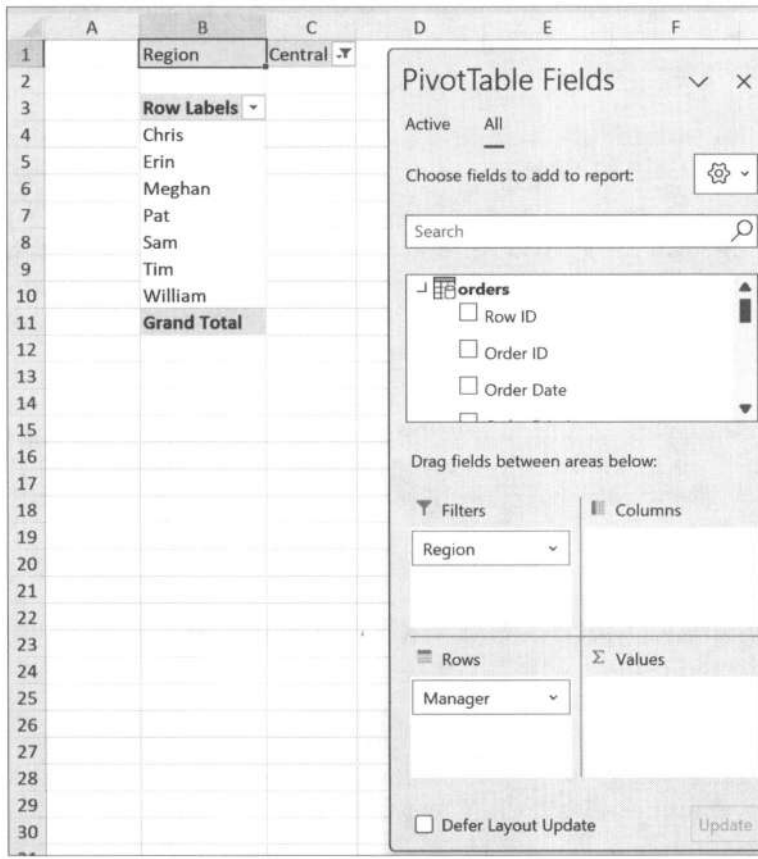


Рис. 7.20. Фильтрация `users` через `orders`

В связи с этим возникает закономерный вопрос: почему фильтр, примененный к таблице `orders`, никак не повлиял на таблицу `users`? Разве в сводной таблице не должен был остаться один менеджер `Chris`, с учетом того, что он является единственным менеджером по центральному региону? Ответ кроется в направлении фильтрации.

Направление фильтрации и кардинальность

В Power Pivot направление фильтрации зависит от типа связи. В связи «один ко многим» фильтр передается от «одного» ко «многим». Например, фильтрация таб-

лицы `users` (см. рис. 7.18) может влиять на таблицу `orders`, но не наоборот. Такой подход повышает производительность, поскольку передача фильтрации от таблицы с меньшим количеством записей к таблице с большим количеством записей более эффективна.

Изменение направления фильтра в Power Pivot

В Power Pivot вы не можете сами изменить направление фильтрации, поскольку оно определяется кардинальностью табличной связи. Но если вам необходимо принудительно изменить направление фильтрации в силу каких-то особых требований, используйте функцию `CROSSFILTER()` в DAX, описание которой выходит за рамки этой книги. Подробнее о ней можно прочитать на сайте Microsoft Learn[3].

От теории к практике

Джазовый гитарист Ирвинг Эшби (Irving Ashby) однажды сравнил ритм-гитару с ванилью в пирожном: «Вы можете не ощущать ее вкуса, но вы сразу поймете, когда ее забыли добавить». И с направлением фильтрации в модели данных так же. Как правило, оно работает незаметно, в фоновом режиме, но когда что-то не так, его отсутствие становится слишком очевидным.

Разобравшись с такими базовыми понятиями модели данных, как кардинальность и направление фильтрации, теперь мы можем перейти к рассмотрению расширенных возможностей Power Pivot. Создание вычисляемых столбцов и иерархий позволит еще больше усовершенствовать вашу модель данных, добавить ей гибкости и улучшить ее функциональность.

Создание вычисляемых столбцов в Power Pivot

В *главе 4* вы уже познакомились с созданием вычисляемых столбцов в Excel Power Query. Теперь давайте узнаем, как можно выполнить эту же задачу в Power Pivot, а также рассмотрим преимущества и недостатки обоих подходов.

Вычисления в Power Query или в Power Pivot?

Power Query и Power Pivot — это разные инструменты с взаимодополняющими функциями, и в обоих можно создавать вычисляемые столбцы. Чтобы определиться, какой из инструментов задействовать, учитывайте следующие рекомендации:

- используйте Power Query для очистки и преобразования данных на этапе подготовки. Он идеально подходит для выполнения разовых задач, таких как объединение таблиц или изменение типов данных, а также для оптимизации модели за счет упрощения данных перед загрузкой в Power Pivot;

[3] См. **https://clck.ru/3JuUXA**.

- используйте Power Pivot для расширенного анализа — например, для динамических вычислений или построения связей между таблицами. Эти операции выполняются после загрузки данных и позволяют настроить отчеты и информационные панели. Однако чрезмерная ориентация на них может привести к увеличению размера файла и снижению производительности.

Придерживаясь этих рекомендаций, вы сможете максимально эффективно использовать возможности Power Query и Power Pivot, создавая оптимальные вычисляемые столбцы с учетом состояния данных и требований к их обработке.

Хотя эти эмпирические правила полезны и применимы на практике, лучший способ выбрать, где создавать вычисляемые столбцы: в Power Query или в Power Pivot, — это поэкспериментировать с обоими инструментами и посмотреть, какой из них лучше подходит для вашего случая.

Пример: расчет нормы прибыли

Вернитесь в редактор Power Pivot и в **Data View** (Представление данных) откройте вкладку с таблицей `orders`.

Давайте создадим вычисляемый столбец с названием `Profit margin`. Прокрутите страницу вправо, пока не дойдете до конца таблицы — до столбца с заголовком **Add Column** (Добавление столбца). Щелкните на заголовке **Add Column** и измените название столбца на `Profit margin`, а затем добавьте формулу для расчета нормы прибыли (`=orders[Profit]/orders[Sales]`), как показано на рис. 7.21.

[Profit mar... ✓ =orders[Profit] / orders[Sales]

	tegory	Product Sub-Category	Product Name	Supplier	Product Container	Marketing Channel	Product Base Margin	Ship Date	Profit margin	Add Column
1	olies	Paper	Xerox 1905	Xerox	Small Box	Email Marketing	0.37	8/23/2021 ...	-0.845751746...	
2	olies	Paper	Xerox 1997	Xerox	Small Box	Email Marketing	0.37	6/13/2022 ...	-0.898311227...	
3	olies	Paper	Xerox 21	Xerox	Small Box	Email Marketing	0.37	7/21/2020 ...	-0.405810518...	
4	olies	Paper	Xerox 1995	Xerox	Small Box	Email Marketing	0.37	10/1/2023 ...	-0.303716814...	
5	olies	Paper	Xerox 214	Xerox	Small Box	Email Marketing	0.37	11/15/202...	-0.548446518...	
6	olies	Paper	Xerox 1894	Xerox	Small Box	Email Marketing	0.37	5/12/2023 ...	-0.408720311...	
7	olies	Paper	Xerox 1994	Xerox	Small Box	Email Marketing	0.37	10/11/202...	-0.243463770...	
8	olies	Paper	Xerox 227	Xerox	Small Box	Email Marketing	0.37	12/21/202...	-0.727707636...	
9	olies	Paper	Xerox 2	Xerox	Small Box	Email Marketing	0.37	12/11/202...	-0.347007586...	
10	olies	Paper	Xerox 216	Xerox	Small Box	Email Marketing	0.37	8/15/2020 ...	-0.624150875...	
11	olies	Paper	Xerox 210	Xerox	Small Box	Email Marketing	0.37	11/28/202...	-0.536013400...	
12	olies	Paper	Xerox 220	Xerox	Small Box	Email Marketing	0.37	6/2/2022 1...	-0.553918747...	
13	olies	Paper	Xerox 227	Xerox	Small Box	Email Marketing	0.37	9/16/2023 ...	-0.728280552...	
14	olies	Paper	Xerox 224	Xerox	Small Box	Email Marketing	0.37	9/16/2021 ...	-0.716479200...	
15	olies	Paper	Xerox 213	Xerox	Small Box	Email Marketing	0.37	8/11/2023 ...	-0.660714285...	
16	olies	Paper	Xerox 207	Xerox	Small Box	Email Marketing	0.37	9/15/2023 ...	-0.407959356...	
17	olies	Paper	Xerox 226	Xerox	Small Box	Email Marketing	0.37	11/24/202...	-0.338816940...	
18	olies	Paper	Xerox 226	Xerox	Small Box	Email Marketing	0.37	3/7/2022 1...	-0.329385001...	
19	olies	Paper	Xerox 210	Xerox	Small Box	Email Marketing	0.37	9/10/2023 ...	-0.501557273...	
20	olies	Paper	Xerox 212	Xerox	Small Box	Email Marketing	0.37	11/25/202...	-0.749319812...	
21	olies	Paper	Xerox 1905	Xerox	Small Box	Email Marketing	0.37	5/25/2023 ...	-0.582962492...	

Рис. 7.21. Создание вычисляемого столбца для нормы прибыли

Ваш вычисляемый столбец должен рассчитываться так:

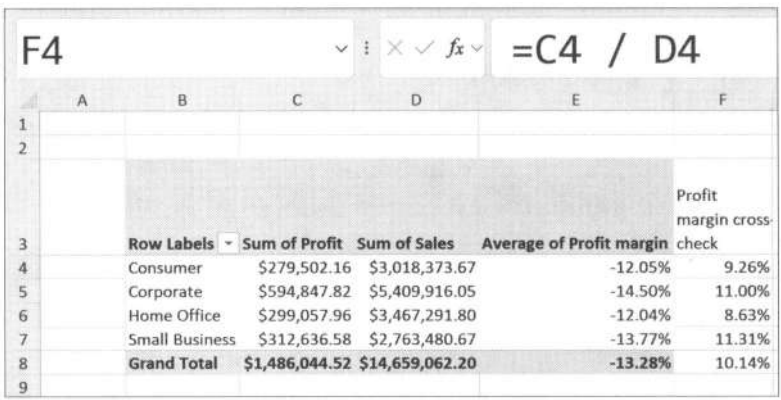
```
orders[Profit] / orders[Sales]
```

Обратите внимание, что здесь, в отличие от таблиц Excel, ссылки на столбцы нужно вводить вручную, а не выделять их с помощью курсора мыши или нажатия клавиши.

На этом мы завершим ваш первый опыт использования языка программирования DAX для управления моделью данных в Power Pivot. Заметьте, что ссылки на отдельные столбцы очень похожи на структурированные ссылки на столбцы в таблицах Excel. Также в режиме представления данных вы можете указать для нового столбца формат процентов.

Чтобы проверить правильность вычислений, загрузите модель данных в новую сводную таблицу. Перетащите поле `Customer Segment` в область **Rows** и поле `Average of Profit Margin` (`Среднее по столбцу Profit Margin`) в область **Values**. Для перекрестной проверки точности добавьте `Sum of Profit` и `Sum of Sales` в область **Values**.

При ручном расчете нормы прибыли по формуле можно заметить расхождение со значениями из сводной таблицы (рис. 7.22).



	A	B	C	D	E	F
1						
2						
3		Row Labels	Sum of Profit	Sum of Sales	Average of Profit margin	Profit margin cross-check
4		Consumer	$279,502.16	$3,018,373.67	-12.05%	9.26%
5		Corporate	$594,847.82	$5,409,916.05	-14.50%	11.00%
6		Home Office	$299,057.96	$3,467,291.80	-12.04%	8.63%
7		Small Business	$312,636.58	$2,763,480.67	-13.77%	11.31%
8		**Grand Total**	**$1,486,044.52**	**$14,659,062.20**	**-13.28%**	10.14%
9						

Рис. 7.22. Проверка точности расчета нормы прибыли

Проблема со сводной таблицей возникает из-за того, что среднее значение для столбца `Profit margin` рассчитывается по промежуточным нормам прибыли, без учета общей прибыли и общего объема продаж. Для точного расчета нормы прибыли необходимы динамические вычисления на лету, которые нельзя реализовать с помощью одних только вычисляемых столбцов. В таких случаях необходимо использовать меры DAX, которые мы подробно рассмотрим в *главах 8* и *9*.

Сейчас же важно запомнить, что вычисляемые столбцы в Power Pivot не следует использовать, если можно обойтись простым агрегированием. Эта проблема аналогична проблеме с вычисляемыми столбцами в Power Query, которые тоже могут искажать результаты агрегирования.

Впрочем, бывают ситуации, когда вычисляемые столбцы в модели данных действительно могут стать подходящим выбором. Одной из таких ситуаций является использование функции `SWITCH()`, которое мы рассмотрим в следующем разделе.

Замена значений в столбце с помощью *SWITCH()*

Функция `SWITCH()` очень удобна для замены значений по определенным условиям. Учитывая, что при этом каждая строка вычисляется независимо, и результаты, как

правило, не агрегируются, целесообразнее сохранять результат выполнения SWITCH() как вычисляемый столбец, а не как меру.

Для примера допустим, что нам нужно присвоить номера 1, 2, 3 и 4 сегментам "Consumer", "Corporate", "Home Office" и "Small Business" соответственно. В случаях, когда ни одно совпадение не найдено, пусть значение будет "Unknown". Для начала добавьте новый вычисляемый столбец с именем Segment number в таблицу orders в Power Pivot (рис. 7.23).

```
[Seg...  = SWITCH(
             orders[Customer Segment],
             "Consumer", "1",
             "Corporate", "2",
             "Home Office", "3",
             "Small Business", "4",
             "Unknown"
           )
```

	Su...	Product Co...	Marketing C...	Product Base...	Ship ...	Profit margin	Segment number	Add
1	Xerox	Small Box	Email Marketing	0.37	8/23/20...	-84.58%	4	
2	Xerox	Small Box	Email Marketing	0.37	6/13/20...	-89.83%	4	
3	Xerox	Small Box	Email Marketing	0.37	7/21/20...	-40.58%	1	
4	Xerox	Small Box	Email Marketing	0.37	10/1/20...	-30.37%	2	
5	Xerox	Small Box	Email Marketing	0.37	11/15/2...	-54.84%	2	
6	Xerox	Small Box	Email Marketing	0.37	5/12/20...	-40.87%	3	
7	Xerox	Small Box	Email Marketing	0.37	10/11/2...	-24.35%	1	
8	Xerox	Small Box	Email Marketing	0.37	12/21/2...	-72.77%	4	
9	Xerox	Small Box	Email Marketing	0.37	12/11/2...	-34.70%	2	
10	Xerox	Small Box	Email Marketing	0.37	8/15/20...	-62.42%	2	
11	Xerox	Small Box	Email Marketing	0.37	11/28/2...	-53.60%	2	
12	Xerox	Small Box	Email Marketing	0.37	6/2/202...	-55.39%	4	

Рис. 7.23. Создание столбца Segment number с помощью функции SWITCH()

Имейте в виду, что все значения в столбце таблицы в модели данных должны иметь один и тот же тип данных. Поскольку в значениях Segment number может встречаться строка "Unknown", для сохранения согласованности необходимо преобразовать в строки и остальные значения: 1, 2, 3, 4.

Загрузите обновленную модель данных в новую сводную таблицу или обновите существующую сводную таблицу, чтобы можно было использовать этот новый столбец в анализе. Например, на рис. 7.24 продажи суммируются по номерам сегментов, а не по исходным названиям сегментов.

Функция SWITCH() и условные столбцы

Функция SWITCH(), доступная в Power Pivot, предлагает более эффективный и понятный способ выполнения множественных сравнений в рамках одной формулы, чем условные столбцы. Хотя с помощью условных столбцов в Power Query можно сделать то же самое, функция SWITCH() упрощает формулу со сложными условиями, анализируя только одно проверяемое выражение, — она возвращает результат, соответствующий первому совпадению. Такой подход помогает избежать составления многочисленных вложенных операторов IF(), которые могут привести к громоздкой и плохо читаемой формуле, особенно при работе с большими наборами данных и сложными условиями.

Кроме того, функция SWITCH() оптимизирована для обеспечения максимальной производительности в Power Pivot. Это позволяет сократить время обработки данных по сравнению

с вычислением вложенных операторов `IF()`. Хотя производительность может сильно варьироваться в каждом конкретном случае, функция `SWITCH()` в целом представляет собой оптимизированное и эффективное решение для выполнения множественных сравнений в Power Pivot.

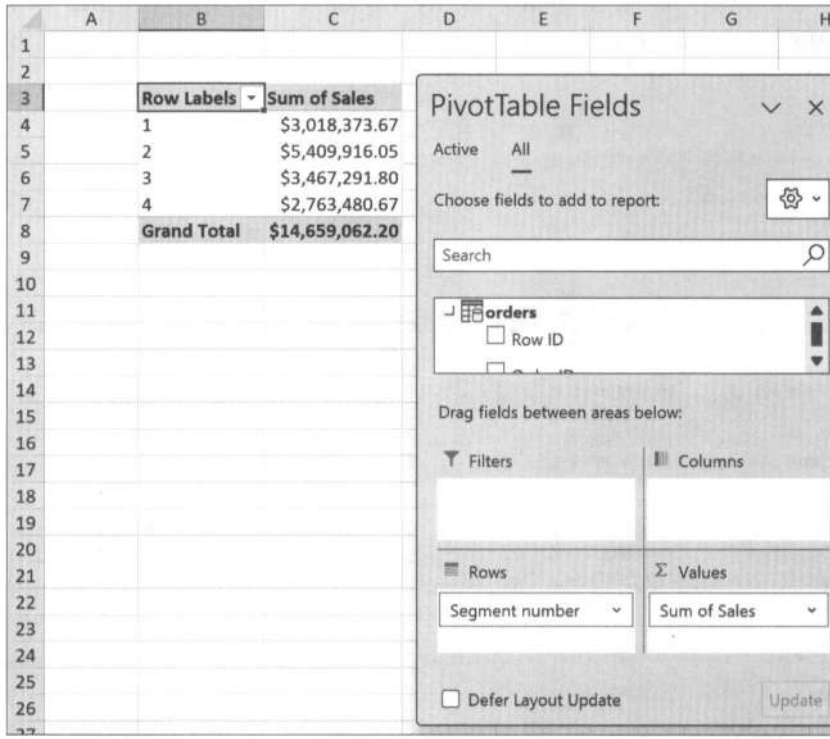


Рис. 7.24. Результаты выполнения функции `SWITCH()`, используемые в сводной таблице

Создание иерархий и работа с ними

Иерархия играет важную роль во многих сферах нашей жизни. В качестве примера можно взять мое местоположение при написании этой книги: Кливленд, Огайо, США. Это можно организовать в иерархическую структуру, начиная с самой широкой категории (страна: США), после которой будет следовать более узкая категория (штат: Огайо), и заканчивая конкретным местоположением (город: Кливленд). Внедрение таких иерархических структур в модель данных упрощает исследование и анализ данных, позволяя использовать различные уровни детализации.

Создание иерархии в Power Pivot

Давайте в нашей модели данных создадим иерархию на основе продукта, которая будет состоять из `Product Category`, `Product Sub-Category` и `Product Name`. Для этого перейдите в режим диаграммы. Удерживая клавишу <Ctrl>, выделите нужные из-

мерения в требуемом для иерархии порядке (например, первым должен быть `Product Category`). После этого щелкните правой кнопкой мыши и выберите пункт **Create Hierarchy** (Создать иерархию). Задайте для иерархии новое имя, например `Product Hierarchy` (рис. 7.25).

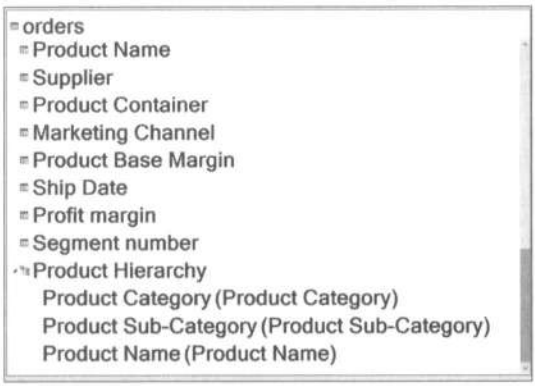


Рис. 7.25. Созданная иерархия в режиме диаграммы

В режиме диаграммы вы можете легко добавлять, изменять или удалять иерархии из модели данных при необходимости. А пока загрузите данные в новую сводную таблицу, чтобы увидеть иерархию в действии.

Использование иерархии в сводной таблице

После закрытия Power Pivot вернитесь к своей сводной таблице. Поместите `Product Hierarchy` в область **Rows** и `Sum of Sales` в область **Values**. Заметьте, что три измерения, объединенные в иерархию, нельзя использовать по отдельности, — они могут существовать только в рамках своей иерархии.

Теперь в сводной таблице вы можете щелкнуть на маленьком значке ⊞ около любой категории товаров, чтобы перейти на уровень подкатегорий и еще глубже по иерархии, до уровня наименований отдельных товаров (рис. 7.26).

Точно так же, щелкая на значке ⊟, вы сможете подняться обратно вверх по иерархии. В группе **Active Field** (Активное поле) вкладки ленты **PivotTable Analyze** (Анализ сводной таблицы) имеются дополнительные опции для работы с иерархиями — например, чтобы одновременно развернуть или свернуть всю иерархию.

Прежде чем добавлять иерархии в свою модель данных, необходимо проверить качество данных и возможные несоответствия. В случае, когда одна и та же подкатегория ошибочно относится к нескольким категориям, иерархия может потерять свой смысл для анализа. Важно также отметить, что не очень опытные пользователи Excel на первых порах могут столкнуться с трудностями при работе с иерархиями.

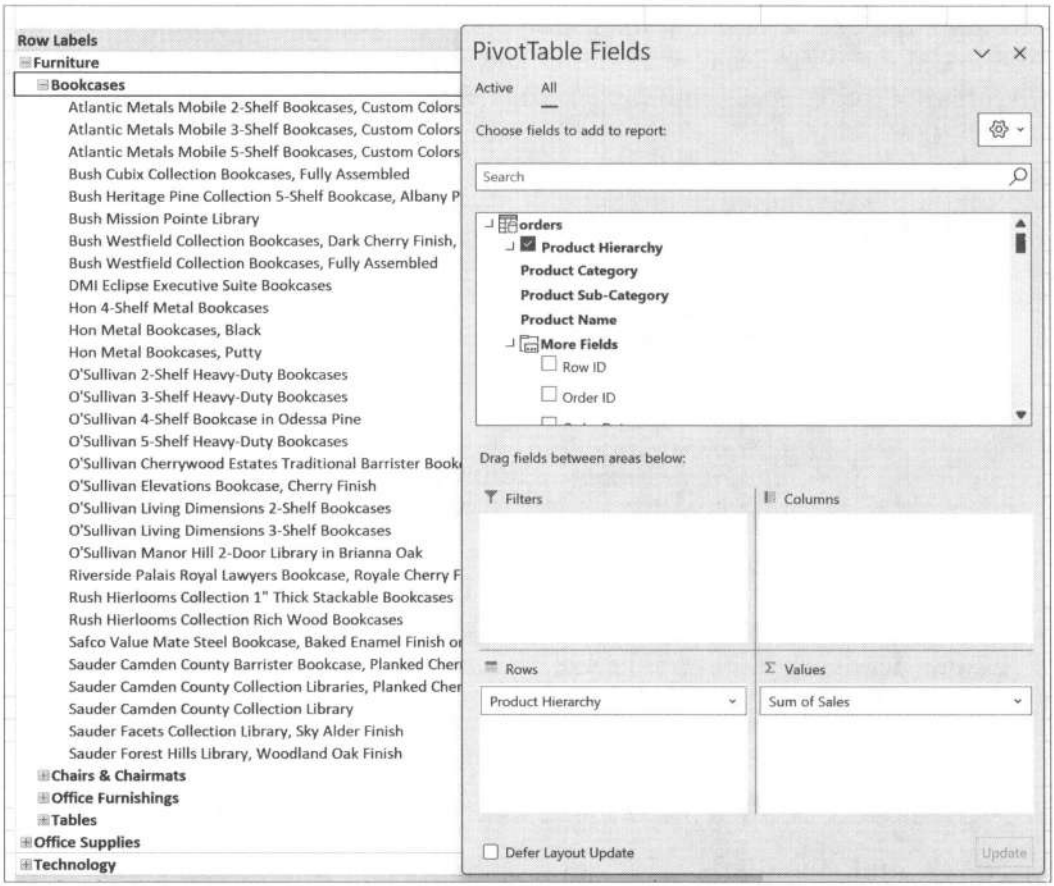


Рис. 7.26. Уровни иерархии в сводной таблице

Загрузка модели данных в Power BI

К этому моменту вы уже освоили создание моделей данных, включая такие полезные элементы, как вычисляемые столбцы и иерархии. В *главах 8* и *9* мы рассмотрим создание мер DAX и использование показателей KPI для улучшения анализа данных и отчетности. Но, прежде чем мы продолжим двигаться дальше, давайте бегло рассмотрим альтернативный способ анализа и визуализации модели данных — инструмент Power BI — и разберемся с основными принципами его работы и преимуществами от его использования.

Power BI как третий инструмент «современного Excel»

До сих пор в этой книге основное внимание уделялось Power Query и Power Pivot для очистки и анализа данных соответственно. Сначала третьим в этом наборе инструментов был Power View, который использовался для визуализации данных, но он больше не поддерживается. Изначально разработанный для Excel, Power View

позволял создавать интерактивные информационные панели и отчеты. Однако со временем его концепция была полностью интегрирована в Power BI, и последние версии Excel уже не содержат Power View или включают его с очень ограниченной функциональностью.

Решение Microsoft переключить внимание в Excel с Power View на Power BI было обусловлено несколькими факторами. Power BI предлагает расширенные возможности для визуализации данных, которые позволяют пользователям создавать интерактивные информационные панели и отчеты на основе различных источников данных. Эта замена также соответствует новой стратегии Microsoft, ориентированной на облачные вычисления, поскольку Power BI работает в основном как облачная платформа, обеспечивающая совместную работу и доступ к данным из любого местоположения. Фокусируясь на Power BI, Microsoft предоставляет более современный, комплексный и интегрированный инструмент для бизнес-аналитики, который лучше удовлетворяет постоянно меняющиеся потребности пользователей.

Хотя Power BI стал весьма популярным благодаря своим возможностям по построению интерактивных информационных панелей, некоторым аналитикам он на первый взгляд может показаться слишком сложным и привести к проблемам при создании и распространении результатов их работы. Использовать Excel для первичного построения модели данных по-прежнему целесообразно, поскольку он широко распространен среди специалистов. Однако по мере увеличения проектов и появления требований к усложнению информационных панелей переход от Excel к Power BI станет жизненно необходимым. В этом разделе мы сделаем первые шаги в направлении такого плавного перехода.

Импорт модели данных в Power BI

Поскольку эта книга не про Power BI, наша задача сейчас — просто загрузить модель данных в Power BI для просмотра. Проверьте, что у вас установлено бесплатное приложение Power BI Desktop. Инструкции по его установке можно найти в официальной документации Microsoft[4]. Если вы захотите изучить Power BI более подробно, могу порекомендовать книгу Jeremey Arnold, «Learning Microsoft Power BI: Transforming Data into Insights» (O'Reilly, 2022)[5].

Чтобы убедиться в том, насколько легко перенести свою работу из Power Pivot в Power BI, вы можете взять из сопроводительного репозитория к этой книге уже готовый файл ch_07_solutions.xlsx или попробовать загрузить рабочую книгу, над которой вы работали на протяжении этой главы.

Итак, закройте эту рабочую книгу в Excel. Откройте приложение Power BI Desktop и создайте новый отчет. На ленте Power BI Desktop выполните команду **File | Import | Power Query, Power Pivot, Power View** (Файл | Импортировать | Power Query, Power Pivot, Power View), как показано на рис. 7.27.

---

[4] См. **https://clck.ru/3JuW8Z**.

[5] См. **https://clck.ru/3JuWL4**.

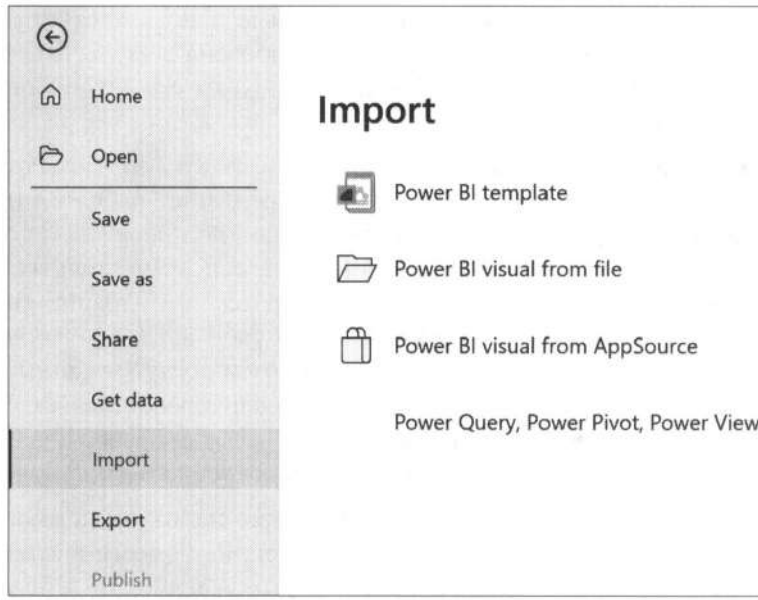


Рис. 7.27. Импорт рабочей книги из Power Pivot в Power BI

Затем найдите файл ch_07_solutions.xlsx и выберите его. Может появиться предупреждение, сообщающее о том, что Power BI приложит максимум усилий для импорта ваших данных (рис. 7.28). Нажмите кнопку **Start** (Запуск), чтобы продолжить импорт.

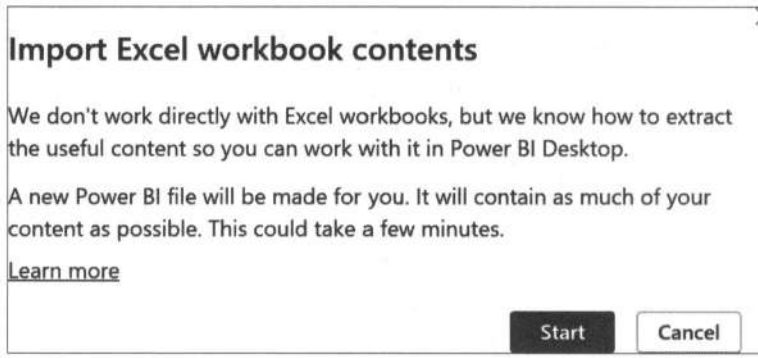


Рис. 7.28. Предупреждение об импорте рабочей книги Excel

Вы можете выбрать копирование данных из Excel или сохранить постоянное подключение. Подключение к рабочей книге Excel удобно при внесении новых изменений в данные, но это может привести к снижению производительности Power BI. Для простоты я сделаю копию данных вместо удерживания постоянного подключения.

Далее вы должны увидеть сообщение, подтверждающее, что Power BI успешно импортировал вашу рабочую книгу, включая запросы, взаимосвязи в модели данных и все добавленные меры и показатели KPI. Иногда вы можете получить сообщение

о том, что из-за большого размера одного из импортируемых объектов вместо копирования было использовано постоянное подключение.

Просмотр данных в Power BI

Чтобы убедиться в том, что модель данных была правильно импортирована в Power BI, перейдите в **Model View** (Представление модели), выбрав на панели в левой части экрана значок  . В этом представлении, которое аналогично диаграмме в Power Pivot, мы можем проверить, правильно ли восстановлены связи между таблицами (рис. 7.29).

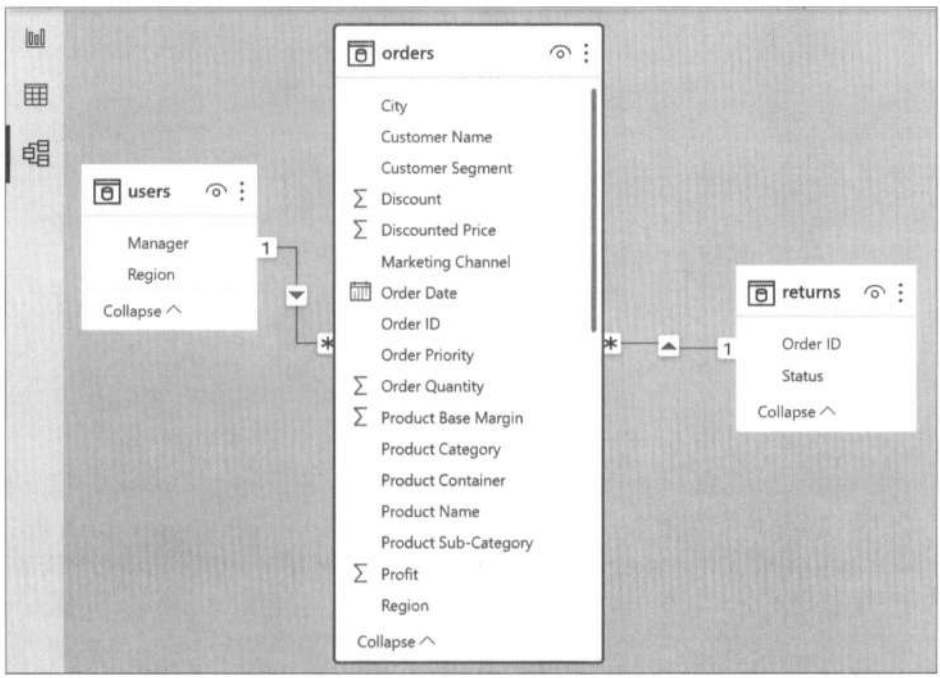


Рис. 7.29. Представление модели в Power BI

Если вы прокрутите таблицу `orders` вниз, то увидите, что иерархия и вычисляемые столбцы, созданные в Power Pivot, также перенесены в Power BI.

Эти вычисляемые столбцы вы можете просмотреть в режиме **Table View** (Представление таблицы), который открывается по нажатию на панели в левой части экрана на значке  . Этот режим просмотра похож на **Data View** (Представление данных) в Power Pivot и тоже позволяет переключаться между источниками данных. Как можно видеть на рис. 7.30, вычисляемые столбцы `Profit Margin` и `Segment Number` были успешно импортированы вместе со своими формулами.

Редактор формул в Power BI — значительно более продвинутый по сравнению с редактором формул в Power Pivot. Это одно из множества серьезных улучшений Power BI, которые позволяют создавать актуальные информационные панели и отчеты, что было бы крайне сложно сделать в Excel.

Впрочем, несмотря на то, что Power BI является современной платформой Microsoft для разработки информационных панелей и отчетов, классический Excel всё еще сохраняет свою популярность как быстрый и удобный инструмент, позволяющий выполнять гибкое моделирование и беглый анализ данных. В конечном счете Power BI и Excel дополняют друг друга и служат разным целям, являясь частью одной дружной команды.

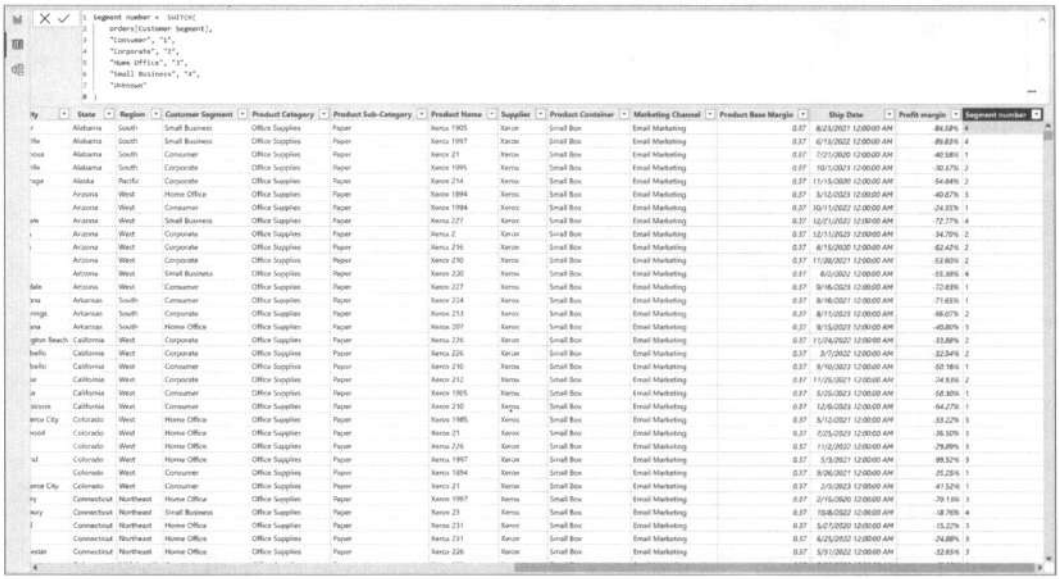


Рис. 7.30. Представление таблицы в Power BI

Теперь ваш отчет Power BI можно сохранить. А я уже сохранил этот файл Power BI для вас в папке ch_07 сопроводительного репозитория к этой книге под именем ch_07_solutions.pbix.

Заключение

В этой главе мы на практике освоили способ построения базовой модели данных и рассмотрели основные функциональности Power Pivot. В оставшихся главах *части II* мы более подробно познакомимся с возможностями Power Pivot для анализа данных и создания отчетов.

Упражнения

Для выполнения этих упражнений откройте файл ch_07_exercises.xlsx, расположенный в папке exercises\ch_07_exercises сопроводительного репозитория к этой книге[6]. Включенная в файл рабочая книга состоит из трех таблиц: `batting`, `people` и `hof`. Выполните следующее:

[6] См. **https://clck.ru/3JuXPK**.

1. Загрузите таблицы в Power Pivot через Power Query и создайте связи в модели данных в Power Pivot.
2. Определите таблицы фактов и таблицы измерений в модели данных и соответствующим образом упорядочьте модель на диаграмме.
3. Разберитесь, какая кардинальность у связей между этими таблицами?
4. Используйте функцию `SWITCH()` для создания столбца `is_player` в таблице `hof`. Присвойте новому столбцу значение `"Yes"`, если в столбце `category` указано `"Player"`, и значение `"No"` в противном случае.
5. Создайте иерархию с полями `birthCountry`, `birthState` и `birthCity` в таблице `people`.
6. Загрузите результаты модели данных в сводную таблицу Excel. Вычислите количество игроков. Это можно сделать, подсчитав количество `playerID`, у которых в столбце `is_player` стоит `"Yes"`.

Готовое решение можно посмотреть в файле ch_07_exercise_solutions.xlsx, расположенном в той же папке репозитория.

Создание мер DAX и показателей KPI в Power Pivot

В *главе 7* вы познакомились с основами Power Pivot и модели данных (Data Model), включая связи, иерархии и вычисляемые столбцы. Когда модель данных готова, можно перейти к созданию мер DAX и показателей KPI, которые помогают конечным пользователям интерпретировать данные, чему и посвящена эта глава.

Чтобы работать с примерами этой главы, откройте из папки ch_08 сопроводительного репозитория к этой книге файл ch_08.xlsx[1]. Мы воспользуемся тем же набором данных о розничных продажах, с которым встретились в *главе 7*, для работы с моделью данных, уже определенной в этом файле.

Создание мер DAX

В *главе 7* попытка добавить в таблицу `orders` вычисляемый столбец `Profit margin` привела к неудовлетворительным результатам. Для выполнения агрегирования и вычислений по различным категориям и временны́м периодам необходимо использовать меры DAX. В Power Pivot эти меры можно создавать двумя способами: явно и неявно. Чтобы проверить это на практике, вставьте новую сводную таблицу из модели данных.

Создание неявных мер

Для агрегирования данных — например, чтобы найти общее количество заказов по регионам, можно просто перетащить соответствующие поля в сводную таблицу (рис. 8.1).

Чтобы изменить агрегирование в сводной таблице для вычисления *среднего* количества проданных товаров по регионам, раскройте выпадающий список в поле `Sum of Order Quantity` (`Сумма по столбцу Order Quantity`), выберите пункт **Value Field Settings** (Параметры поля значений), а в области **Summarize value field by** (Суммировать поле значений по) переключите значение с **Sum** (Сумма) на **Average** (Среднее).

Посмотреть, как модель данных работает с этими вычислениями в сводной таблице, вы сможете, перейдя на вкладку **Power Pivot** на ленте и выбрав опцию **Manage**. На вкладке **Home** в группе **View** переключитесь на **Diagram View**, а на вкладке **Advanced** (Дополнительно) включите опцию **Show Implicit Measures** (Отображение неявных мер) — в самом низу таблицы `orders` появятся две меры (рис. 8.2).

---

[1] См. **https://clck.ru/3JuY6r**.

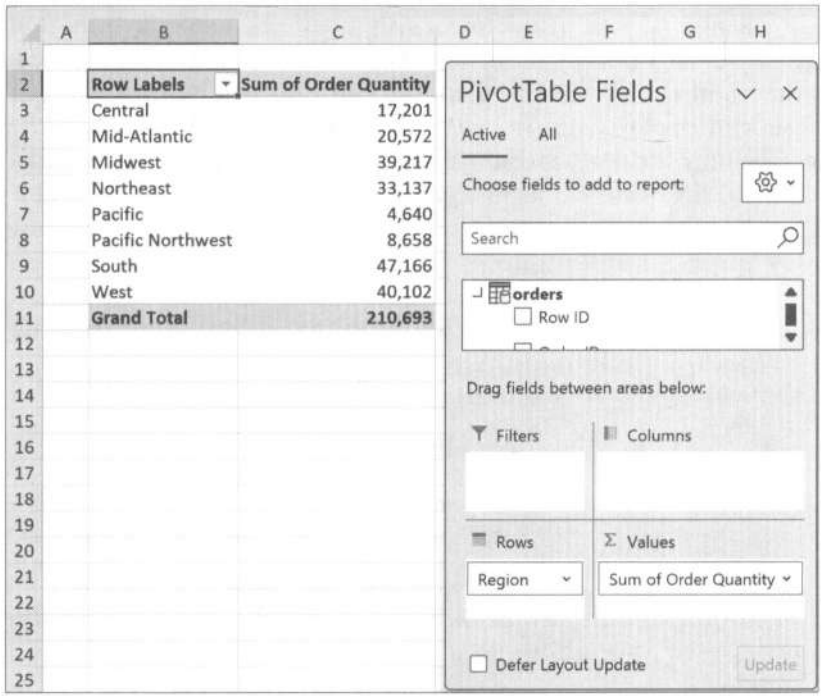


Рис. 8.1. Обычная агрегация в сводной таблице

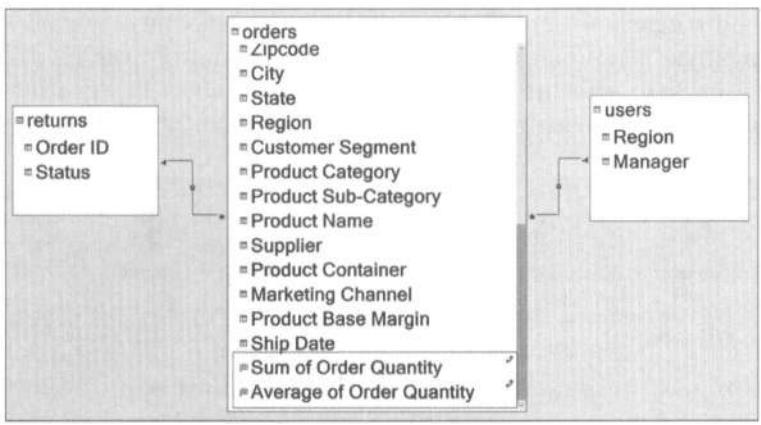


Рис. 8.2. Неявные меры в режиме диаграммы

Эти меры, созданные ранее в сводной таблице, называются *неявными мерами*. Power Pivot генерирует и сохраняет их автоматически. Они позволяют выполнять быстрый и удобный анализ данных без необходимости самим вносить эти сложные вычисления в модель данных.

Тем не менее неявные меры создают проблемы с их настройкой и повторным использованием в рамках модели данных. При добавления новой меры на их основе, например среднего объема продаж на единицу продукции, будет недостаточно про-

стого агрегирования по уже существующему полю. В таких случаях придется создавать отдельную явную меру на основе двух полей (`Sales` и `Order Quantity`). Кроме того, скрытность неявных мер затрудняет работу с ними. Чтобы устранить все эти проблемы, можно просто удалить неявные меры. Для этого выделите обе меры, удерживая клавишу <Ctrl>, затем щелкните правой кнопкой мыши и нажмите **Delete** (Удалить), как показано на рис. 8.3.

Рис. 8.3. Удаление неявных мер в Power Pivot

Создание явных мер

Вместо создания неявной меры DAX через сводную таблицу, можно добавить новую меру вручную с помощью соответствующей опции Power Pivot. Выйдите из редактора Power Pivot, на ленте Excel перейдите на вкладку **Power Pivot** и выберите опцию **Measures | New Measure** (Меры | Создать меру), как показано на рис. 8.4.

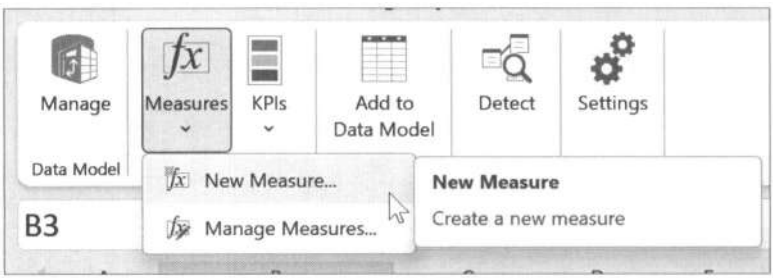


Рис. 8.4. Создание новой меры на вкладке **Power Pivot**

Давайте начнем с создания меры `Total sales`, которая агрегирует столбец `sales` из таблицы `orders`. Формула DAX для ее создания будет содержать структурированные ссылки на таблицы, которые уже встречались нам в *главе 1*. Здесь вы тоже можете пользоваться преимуществами автодополнения IntelliSense от Microsoft при вводе названий функций, таблиц и других элементов, используемых в мере. Свяжите эту меру с таблицей `orders`, указав ее в поле **Table name** (Имя таблицы), а также задайте формат валюты с двумя десятичными знаками.

Завершив ввод формулы, нажмите кнопку **Check formula** (Проверить формулу), чтобы подтвердить правильность ввода. Если всё хорошо, появится сообщение **No errors in formula** (Формула не содержит ошибок), как показано на рис. 8.5.

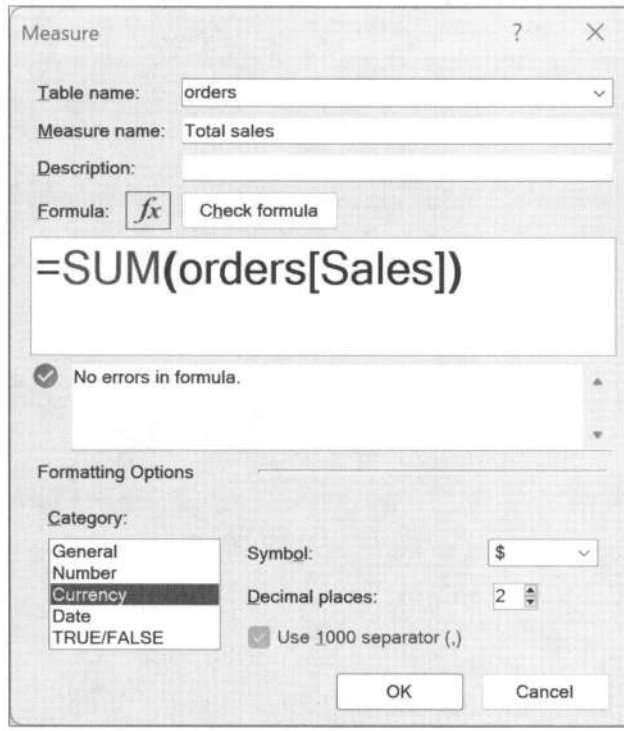


Рис. 8.5. Создание явной меры `Total sales`

Теперь нажмите кнопку **OK** — мера станет доступна для использования в сводной таблице и появится в списке полей таблицы `orders` с обозначением *fx*. Перенесите поле `Region` в **Rows**, а новую меру `Total sales` — в **Values** (рис. 8.6).

В сводной таблице нельзя изменить тип агрегации явной меры, в отличие от неявных мер. Для того чтобы изменить агрегирование явной меры, перейдите опять на вкладку **Power Pivot**, выберите **Measures | Manage Measures** (Меры | Управление мерами), а затем выделите меру `Total sales` и нажмите кнопку **Edit** (Изменить).

Создайте еще одну явную меру с названием `Total profits`. Эта мера должна выглядеть так, как показано на рис. 8.7.

Вычисленные меры могут использоваться в качестве входных данных для других производных мер, что позволяет выполнять сложные расчеты, выходящие за рамки неявных мер. Например, норма прибыли может быть определена с помощью только что созданных мер `Total profits` и `Total sales` (рис. 8.8).

Для этого в сводной таблице добавьте `Total sales`, `Total profits` и `Profit margin` в область **Values** и перенесите `Region` в **Rows**. Перепроверьте правильность расчета нормы прибыли в отдельном столбце по формуле Excel. В отличие от результатов из

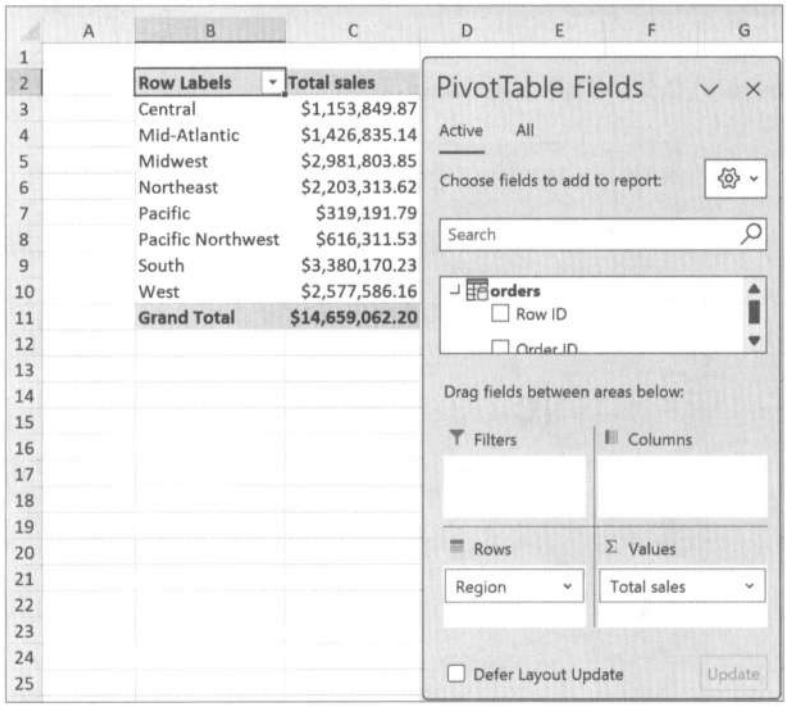


Рис. 8.6. Использование меры DAX в сводной таблице

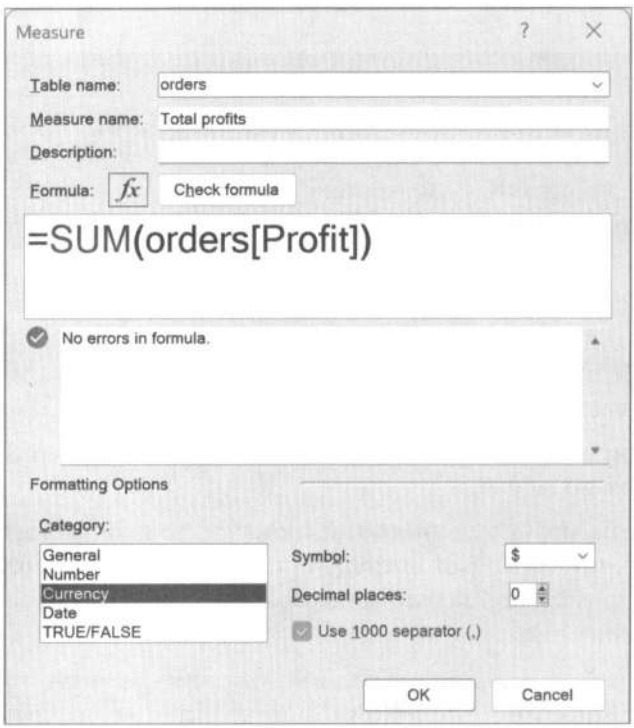


Рис. 8.7. Создание явной меры `Total profits`

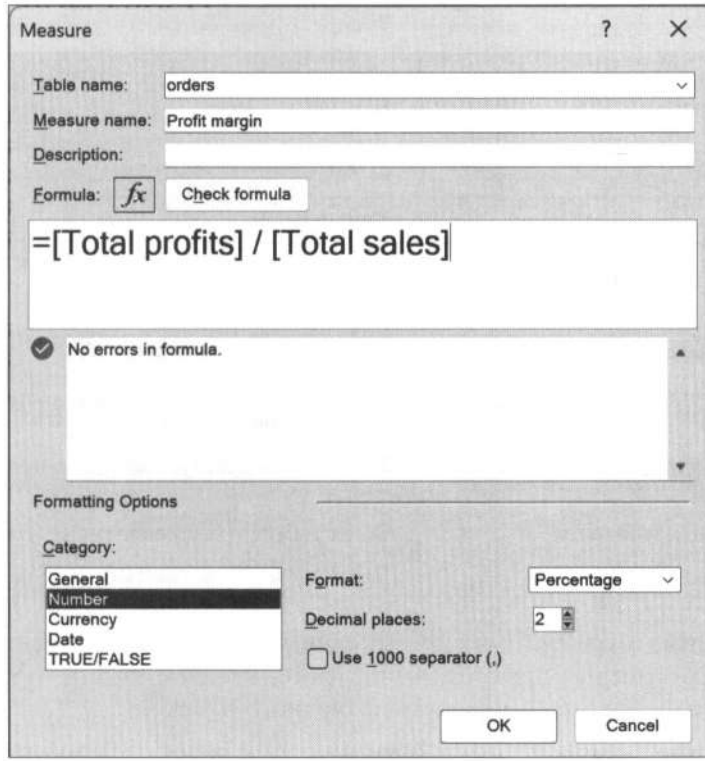
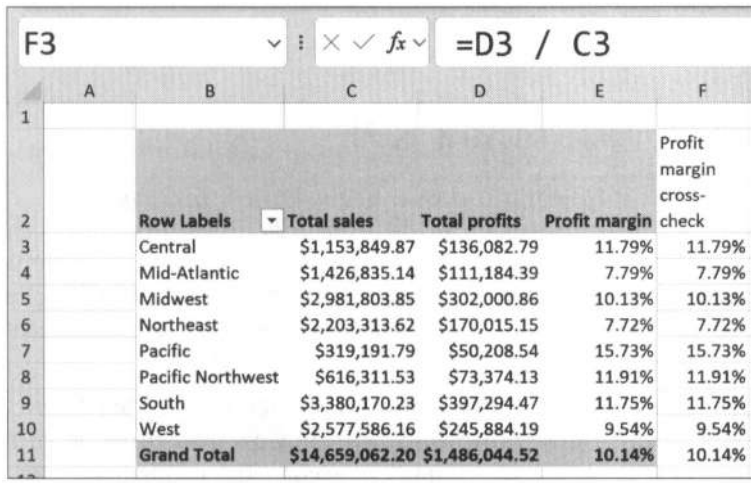


Рис. 8.8. Создание производной меры `Profit margin` на основе других явных мер



	A	B	C	D	E	F
1						
2		Row Labels	Total sales	Total profits	Profit margin	Profit margin cross-check
3		Central	$1,153,849.87	$136,082.79	11.79%	11.79%
4		Mid-Atlantic	$1,426,835.14	$111,184.39	7.79%	7.79%
5		Midwest	$2,981,803.85	$302,000.86	10.13%	10.13%
6		Northeast	$2,203,313.62	$170,015.15	7.72%	7.72%
7		Pacific	$319,191.79	$50,208.54	15.73%	15.73%
8		Pacific Northwest	$616,311.53	$73,374.13	11.91%	11.91%
9		South	$3,380,170.23	$397,294.47	11.75%	11.75%
10		West	$2,577,586.16	$245,884.19	9.54%	9.54%
11		**Grand Total**	**$14,659,062.20**	**$1,486,044.52**	**10.14%**	10.14%

Рис. 8.9. Дополнительная проверка меры `Profit margin`

главы 7, полученных с помощью вычисляемого столбца, теперь наши вычисления точны, в чем можно убедиться, посмотрев на рис. 8.9.

Неявные меры могут быть более удобными, но зато явные меры предоставляют прозрачность, гибкую настройку и возможность выполнения сложных вычислений.

Мы рекомендуем создавать все меры Power Pivot явными, даже самые простые, потому что дополнительные усилия по их созданию не будут потрачены зря.

В табл. 8.1 приведено сравнение неявных и явных мер.

Таблица 8.1. Сравнение явных и неявных мер

Неявные меры	Явные меры
Автоматически генерируются Power Pivot на основе полей сводной таблицы	Вычисления, определяемые самим пользователем
Создаются быстро и очень просто, не требуя никаких усилий	Для создания требуется больше времени и технических навыков
Идеально подходят для беглого анализа данных	Можно адаптировать к конкретным бизнес-потребностям
Могут неточно подсчитать меру или показатель KPI	Точные и предсказуемые
Не гибкие и хуже настраивать	Более гибкие и настраиваемые
Подходят для простого анализа	Подходят для комплексного анализа

С помощью явных мер DAX можно выполнять различные комплексные анализы. В *главе 9* мы рассмотрим примеры вычислений, которые крайне трудно или даже невозможно реализовать с помощью только одного Excel.

Однако перед тем, как приступить к изучению этой более сложной темы, нам необходимо закрепить только что пройденный материал, чтобы вы лучше понимали данные и могли эффективнее работать с ними в Power Pivot. Цель анализа заключается в том, чтобы упростить интерпретацию данных и принятие решений. Поэтому нам не обойтись здесь без рассмотрения показателей KPI.

Создание показателей KPI

KPI (Key Performance Indicators) — *ключевые показатели эффективности* — необходимы для оценки результативности бизнеса и достижения поставленных целей. В Excel Power Pivot показатели KPI могут дать дополнительную ценную информацию для анализа ваших данных. В этом разделе в качестве примера мы создадим показатель KPI, сравнивающий общий объем продаж (overall sales) с целевым объемом продаж (sales target).

Согласно требованиям Power Pivot оба этих значения должны быть созданы как явные меры. Мы уже создали меру `Total sales` ранее. Повторите те же действия для создания меры `Total sales target` (Общий целевой объем продаж), как показано на рис. 8.10.

Чтобы создать показатель KPI, на ленте Excel откройте вкладку **PowerPivot** и выберите **KPIs | New KPI** (Ключевые показатели эффективности | Создать ключевой показатель эффективности). Для базового поля KPI выберите из списка `Total sales`, а для целевого значения — `Total sales target` (рис. 8.11). Это позволит сравнить фактические продажи с целевым показателем.

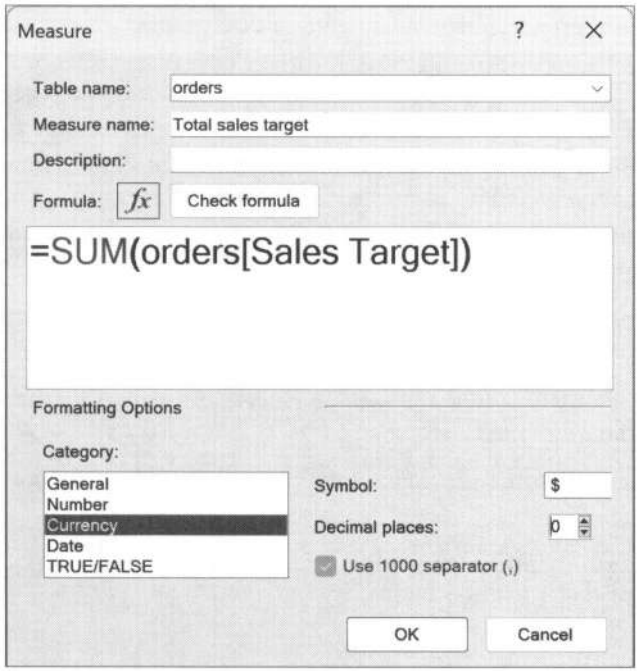


Рис. 8.10. Создание меры `Total sales target`

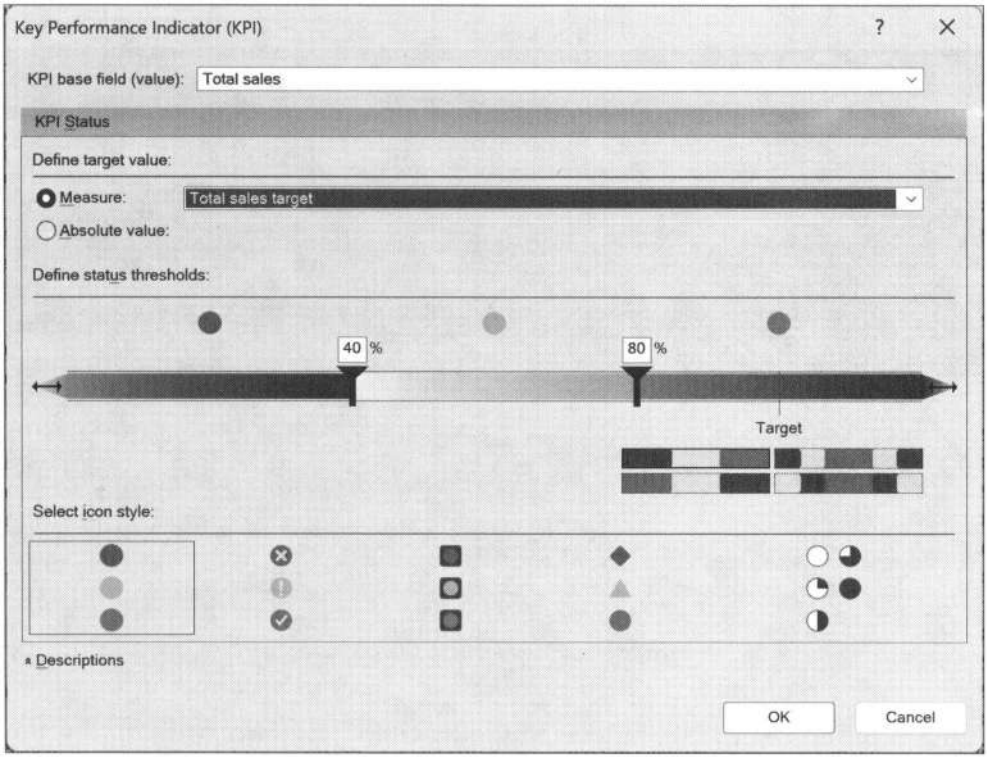


Рис. 8.11. Определение базового поля и целевого значения KPI

Кроме этого, установите пороговые значения состояния, чтобы определить приемлемый диапазон для значений показателя. Эти пороговые значения делят результаты на категории: «хорошие», «удовлетворительные» и «плохие», что позволяет пользователям быстро оценить результативность относительно целевых значений с чуть бóльшим количеством нюансов, чем простая бинарная оценка «попали или не попали». Определим теперь пороговые значения для такого трехуровнего показателя, чтобы было понятно, какие значения превышают ожидания, какие соответствуют им и какие от них отстают:

- если процентное отношение базового значения к целевому будет меньше 90%, оно будет обозначено красным значком;
- если процентное отношение попадет в диапазон между 90 и 100%, оно будет отмечено желтым значком;
- если процентное отношение окажется равным 100% или более, будет стоять зеленый значок.

С помощью курсора мыши перетащите пороговые значения так, чтобы они соответствовали этим правилам (рис. 8.12).

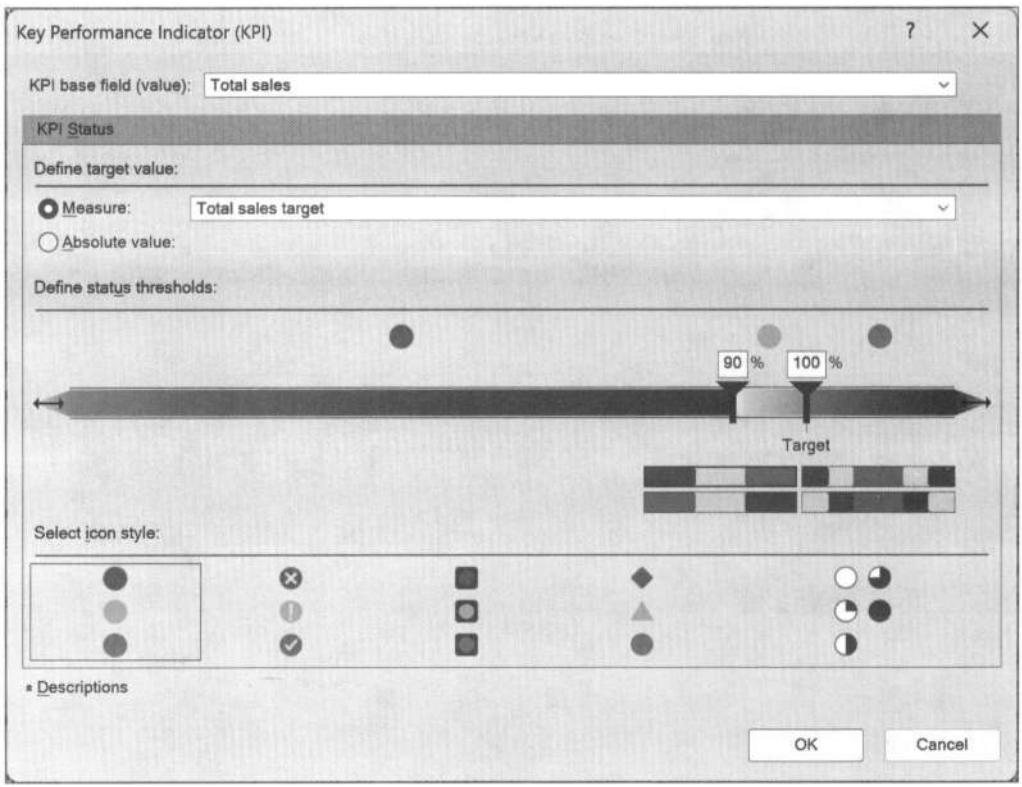


Рис. 8.12. Определение пороговых значений состояния KPI

Настройка стилей значков

Для изменения внешнего вида KPI и дизайна значков можно использовать разные настройки, что может вам пригодиться в некоторых случаях. Однако важно отметить, что в визуализации данных не рекомендуется использовать вместе красный, зеленый и желтый цвета, поскольку это может привести к путанице и неправильной интерпретации у людей с отличающимся цветовым восприятием.

К сожалению, в Power Pivot нет возможности изменять цветовую гамму, что считается значительным недостатком этого инструмента. Поэтому иногда пользователи рассматривают возможность переноса сложных информационных панелей и отчетов на более гибкие BI-платформы, такие как Tableau или Power BI. Эти платформы предоставляют больше опций для настройки цвета и различных визуализаций, что позволяет оформить представление данных более стилизованно и эффектно.

Добавление показателя KPI в сводную таблицу

Настроив показатель KPI, нажмите кнопку **OK**. Вставьте новую сводную таблицу из модели данных или используйте уже имеющуюся в рабочей книге. Перетащите `Region` в **Rows** и `Customer Segment` в **Columns**.

В самом конце таблицы `orders` в списке полей сводной таблицы вы должны увидеть значок светофора с надписью `Total sales`. Раскройте этот выпадающий список и перетащите три поля из него в **Values** (рис. 8.13).

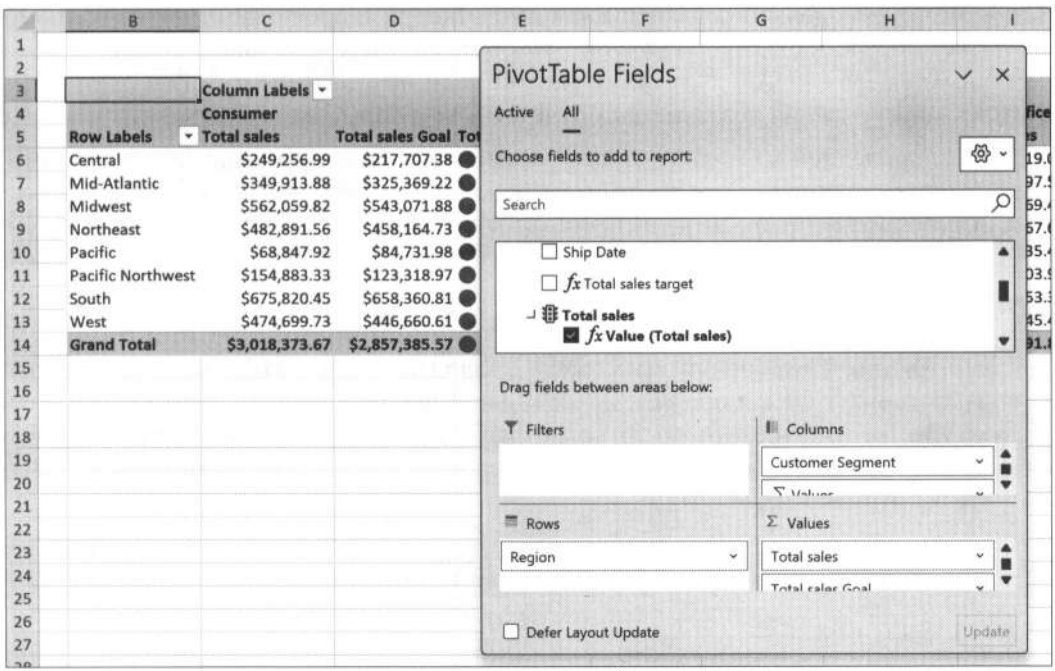


Рис. 8.13. Показатель KPI `Total sales` в сводной таблице

Если новый столбец `Total sales Goal` будет отформатирован неправильно, вы можете исправить это с помощью опции **Value Field Settings** (Параметры полей значений) сводной таблицы.

Показатели KPI структурированы для сводных таблиц следующим образом: сначала выводится столбец с фактическим объемом продаж `Total sales`, затем установленный целевой показатель, который представлен в столбце `Sales target`. Эти два столбца дополняются третьим столбцом с цветными значками, позволяющим сразу визуально определить, достиг ли объем продаж поставленной цели, превзошел ее или не дотянул до нее.

Если вам не подошли пороговые значения и вы хотите их подкорректировать, то всегда можете перейти на вкладку **Power Pivot** ленты, а затем выбрать **KPIs | Manage KPIs** (Ключевые показатели эффективности | Управление ключевыми показателями эффективности), выделить нужный KPI и нажать **Edit** (Изменить).

Показатели KPI и сводные таблицы, созданные на основе явных мер, представляют собой лишь стартовую точку для создания расширенных отчетов и визуализаций с помощью Excel и Power Pivot. Чтобы получить более полное представление о том, как можно использовать созданную модель данных для построения комплексных информационных панелей, дополненных разными функциональностями, такими как срезы, условное форматирование и пр., рекомендую вам книгу: Bernard Obeng Boateng. «Data Modeling with Microsoft Excel» (Packt, 2023)[2].

Заключение

В этой главе мы сделали первые шаги к созданию расширенных анализов и надежных отчетов с помощью Power Pivot, разобравшись, в чем разница между явными и неявными мерами DAX. Мы также рассмотрели показатели KPI в Power Pivot, их важность для создания полноценных отчетов и некоторые их ограничения.

В *главе 9*, последней в *части II*, будут рассмотрены более продвинутые возможности DAX для создания анализов на основе сводных таблиц, которые было бы сложно или невозможно построить без использования DAX.

Упражнения

Для выполнения упражнений вы можете продолжить работать с той же моделью данных, которую создали при рассмотрении примеров этой главы, и той же рабочей книгой или начать всё заново, открыв файл ch_08_exercises.xlsx, расположенный в папке exercises\ch_08_exercises сопроводительного репозитория к этой книге[3]. Выполните следующие задания:

1. Создайте сводную таблицу, чтобы подсчитать общее количество хоум-ранов (`HR`) в зависимости от штата рождения (`birthState`) с помощью неявной меры.

[2] См. **https://clck.ru/3Jui85**.

[3] См. **https://clck.ru/3JuiTT**.

2. Удалите неявную меру, созданную в пункте 1, и создайте новую явную меру с названием `hr_total`, которая будет вычислять сумму хоум-ранов, отформатированную как целое число с разделителем разрядов. Добавьте эту меру в сводную таблицу.
3. Создайте еще одну явную меру с названием `hr_pct`, которая вычисляет процентное соотношение общего количества хоум-ранов (`HR`) к общему количеству подходов к бите (`AB`) из таблицы `batting`. В качестве формата результата укажите процент. Чтобы упростить себе задачу, вы, конечно, можете создать вспомогательную меру, вычисляющую общее количество подходов к бите.
4. Создайте показатель KPI на основе меры `hr_pct`, в качестве цели укажите абсолютное значение 1. Используйте следующие пороговые значения состояния:
   - менее 2% — красный значок;
   - от 2 до 3% — желтый значок;
   - более 3% — зеленый значок.
5. Добавьте показатель KPI в сводную таблицу, в которой поле `teamID` будет в строках, а `yearID` — в столбцах.

Готовое решение можно посмотреть в файле ch_08_solutions.xlsx, расположенном в той же папке репозитория.

Функции DAX в Power Pivot

В *главе 8* мы рассмотрели базовые меры DAX, используемые для создания отчетов. В этой, заключительной главе *части II* мы чуть глубже погрузимся в DAX, чтобы научиться создавать более качественные отчеты с помощью сводных таблиц Excel.

Чтобы работать с примерами этой главы, откройте из папки ch_09 сопроводительного репозитория к этой книге файл ch_09.xlsx[1]. Мы воспользуемся здесь тем же набором данных о розничных продажах, что и в предыдущих главах.

Открытая нами рабочая книга Excel уже содержит сводную таблицу, связанную с моделью данных, и заранее созданную явную меру `Total sales`. Эта мера вычисляется как сумма столбца `Sales` из таблицы `orders`, и мы будем использовать ее далее в разных примерах.

Функция *CALCULATE()*

Контекст фильтра

В обычных сводных таблицах можно настроить общий фильтр для всех значений. Например, если, как показано на рис. 9.1, отфильтровать `Ship Mode` (тип доставки) по значению `Express Air` (авиаперевозка), то вместо общего объема продаж будет отображаться объем продаж отфильтрованных записей. Вы можете получить или общий объем продаж, или только продажи `Express Air`, но не оба результата одновременно.

	A	B	C	D
1				
2		Ship Mode	Express Air	
3				
4		Total sales		
5		$1,172,357.55		
6				

Рис. 9.1. `Total sales` теперь вычисляется в контексте фильтра

Проще говоря, каждое значение в сводной таблице соответствует *контексту фильтра* (Filter Context). Однако с помощью функции `CALCULATE()` можно освободить

[1] См. **https://clck.ru/3K3KeU**.

меры от этого ограничения, позволяя им работать в измененном контексте фильтра. Это кардинальным образом расширяет возможности сводных таблиц. Несмотря на всю мощь функции `CALCULATE()`, ее синтаксис достаточно прост (табл. 9.1).

Таблица 9.1. Параметры функции `CALCULATE()`

Параметр	Тип данных	Описание
`expression`	Любое допустимое выражение DAX	Выражение, которое необходимо вычислить. Это может быть мера, столбец или другая функция
`[filter1], [filter2], [...]`	Столбец, таблица или булево выражение	Необязательные параметры. Один или несколько фильтров, которые будут применены к выражению

Функция *CALCULATE()* с одним условием

Давайте для начала создадим меру с названием `Total express air sales`. Эта мера будет вычислять `Total sales` для отфильтрованных заказов, у которых `Ship Mode` равно `Express Air` (рис. 9.2).

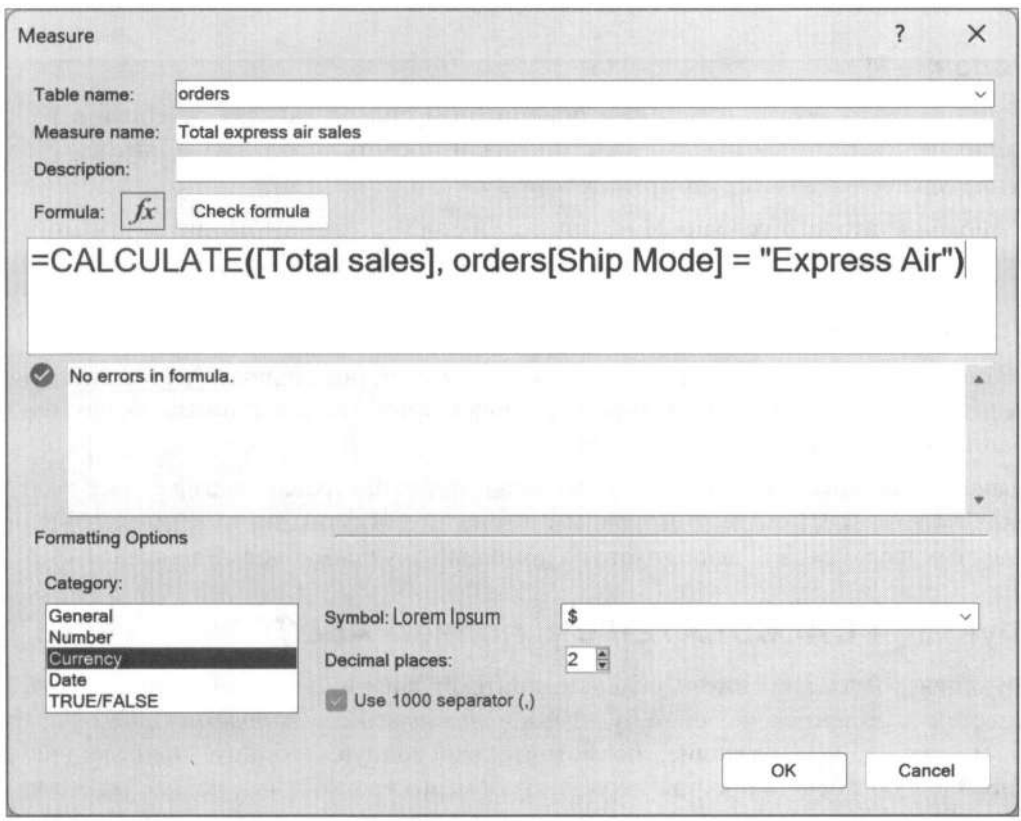


Рис. 9.2. Функция `CALCULATE()` с одним условием

Создав эту меру, добавьте ее в сводную таблицу вместе с `Total sales` и уберите из сводной таблицы фильтр по `Ship Mode`. Теперь можно одновременно видеть в сводной таблице общие продажи и продажи `Express Air` (рис. 9.3).

	A	B	C
1			
2			
3		**Total sales**	**Total express air sales**
4		$14,659,062.20	$1,172,357.55
5			

Рис. 9.3. Мера `Total express air sales`, не зависящая от контекста фильтра

Теперь вы знаете, как можно реализовать в сводной таблице анализ частных случаев. Это огромный шаг вперед в развитии функциональности сводных таблиц.

Функция *CALCULATE()* с несколькими условиями

Изменить контекст фильтра в `CALCULATE()` можно также с помощью нескольких условий. В этом разделе мы рассмотрим, как добавлять в эту функцию условия `AND` (И), `OR` (ИЛИ) и `ALL()`.

Условие И

Учитывая, что заказы с высоким приоритетом сильно зависят от сбоев в работе авиаперевозок, было бы полезно проанализировать продажи, в которых `Order Priority` равно `High` И `Ship Mode` равно `Express Air`.

Включение второго условия И в меру с функцией `CALCULATE()` сводится просто к добавлению еще одного параметра функции (рис. 9.4).

Условие ИЛИ

Работая с условной логикой, всегда помните о чувствительности результата. Даже незначительные изменения в условиях могут привести к кардинально другим результатам.

Давайте для примера проверим сумму продаж, отфильтровав заказы с `Order Priority`, равным `High`, ИЛИ с `Ship Mode`, равным `Express Air`. В функции `CALCULATE()` объединение этих двух условий выполняется с помощью двух символов `||` (рис. 9.5).

Функция *CALCULATE()* с условием *ALL()*

Функция `CALCULATE()` может добавлять контекст фильтра, но при ее использовании вместе с `ALL()` при вычислении будет *игнорироваться весь контекст фильтра*. Чтобы понять, в чем различие, посмотрите на сводную таблицу, приведенную на рис. 9.6. Обе меры: и простая мера `Total sales`, и мера `Total express air sales` с измененным контекстом фильтра — меняются в зависимости от общего фильтра сводной таблицы `Product Category`.

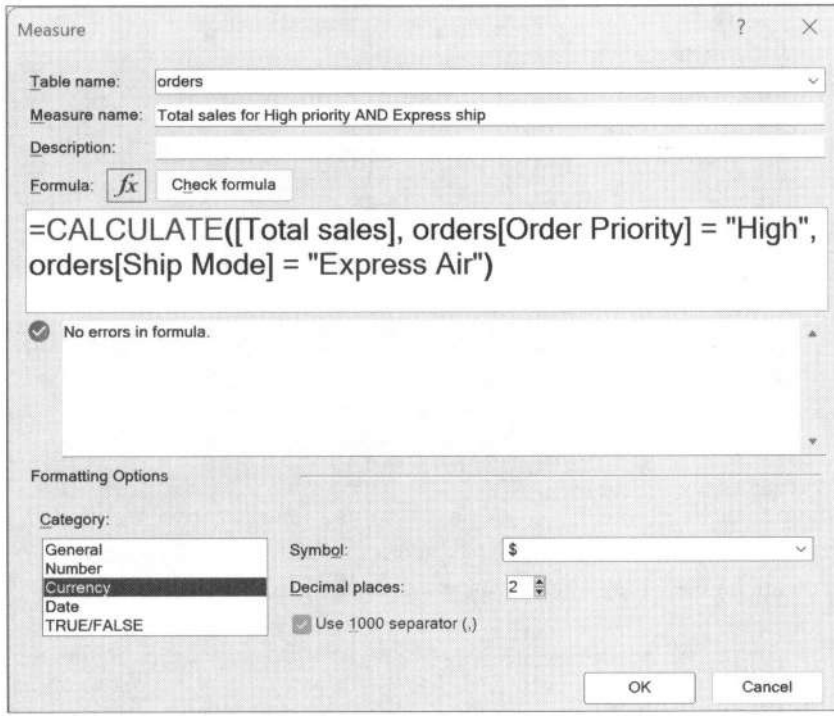


Рис. 9.4. Функция `CALCULATE()` с условием И

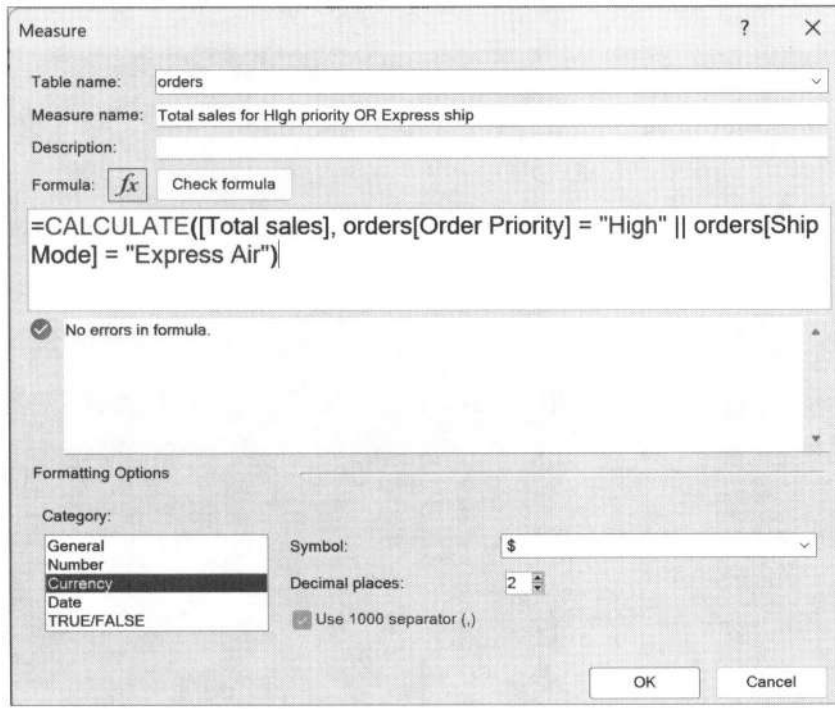


Рис. 9.5. Функция `CALCULATE()` с условием ИЛИ

	A	B	C	D	E	F
1		Product Category	Office Supplies			
2						
3		Row Labels	Total sales	Total express air sales	Total sales for High priority AND Express ship	Total sales for High priority OR Express ship
4		Central	$293,175.13	$29,472.47	$1,175.20	$85,720.11
5		Mid-Atlantic	$354,770.45	$39,211.78	$3,347.29	$103,395.10
6		Midwest	$822,744.58	$111,774.44	$32,535.80	$253,007.40
7		Northeast	$570,737.26	$69,083.95	$11,675.41	$177,060.79
8		Pacific	$64,565.18	$11,739.70		$24,596.95
9		Pacific Northwest	$161,470.81	$14,112.92	$1,941.95	$35,228.42
10		South	$786,499.83	$125,652.99	$26,255.53	$370,723.24
11		West	$628,879.49	$71,732.89	$16,345.86	$182,583.44
12		**Grand Total**	**$3,682,842.73**	**$472,781.14**	**$93,277.04**	**$1,232,315.45**
13						

Рис. 9.6. На результаты `CALCULATE()` влияет фильтр сводной таблицы

Чтобы сделать вычисление по *всем* базовым значениям, независимо от контекста общего фильтра, нужно использовать функцию `CALCULATE()` вместе с `ALL()`. Давайте для примера создадим меру с названием `All total sales` (рис. 9.7).

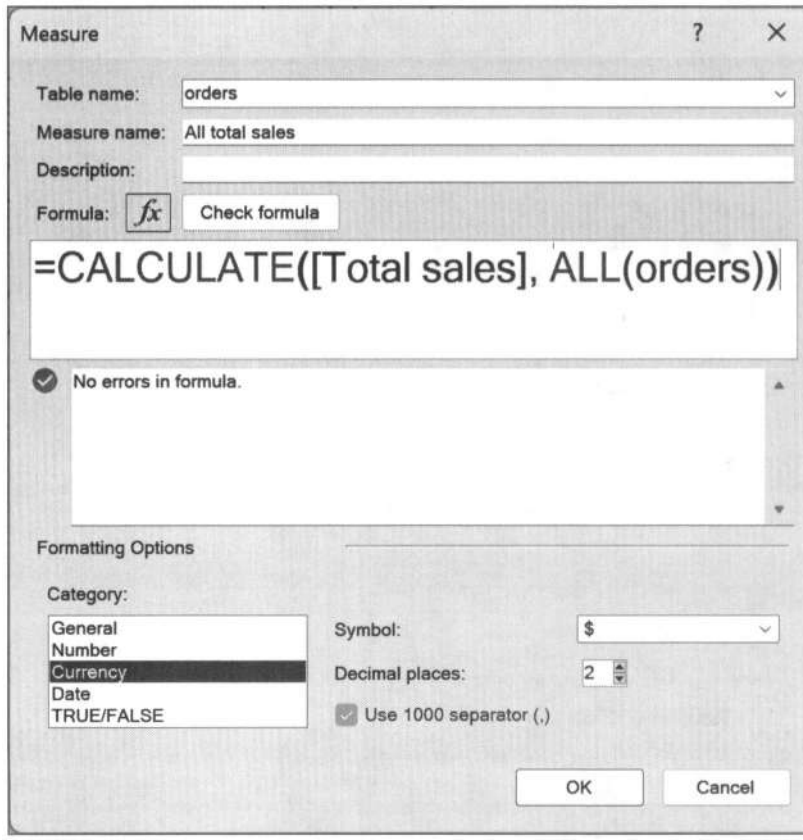


Рис. 9.7. Использование `CALCULATE()` вместе с `ALL()` для игнорирования контекста общего фильтра

Если указать `ALL(orders)` в качестве условия фильтрации, в расчете будет участвовать каждая запись таблицы, что перекроет любой другой контекст фильтра в сводной таблице. Различие показано на рис. 9.8.

Это удобно для сравнения различных частных агрегаций данных с общими объемами продаж, которые не будут зависеть от применяемых фильтров.

	A	B	C	D
1				
2		Product Category	(Multiple Items)	
3				
4		**Row Labels**	**Total sales**	**All total sales**
5		Central	$725,994.94	$14,659,062.20
6		Mid-Atlantic	$857,744.48	$14,659,062.20
7		Midwest	$1,844,183.47	$14,659,062.20
8		Northeast	$1,235,775.95	$14,659,062.20
9		Pacific	$175,674.06	$14,659,062.20
10		Pacific Northwest	$407,247.28	$14,659,062.20
11		South	$2,009,764.91	$14,659,062.20
12		West	$1,543,525.47	$14,659,062.20
13		**Grand Total**	**$8,799,910.55**	**$14,659,062.20**
14				

Рис. 9.8. Мера `All total sales` в сводной таблице

* * *

Функция `CALCULATE()` в Power Pivot играет такую же роль, как поисковые функции и сводные таблицы в классическом Excel. Она расширяет ваши возможности и помогает перейти на новый уровень вычислений. Более подробную информацию о контексте фильтра и функции `CALCULATE()` можно найти в книге Марко Руссо и Альберто Феррари «Подробное руководство по DAX» (ДМК Пресс, 2020)[2].

Функции аналитики времени

> В основе количественных доказательств лежит один-единственный вопрос: По сравнению с чем?
> – *Эдвард Тафти*

Анализ трендов имеет огромное значение для бизнеса. Аналитики сравнивают текущую производительность с историческими данными и оценивают ежемесячные и ежегодные показатели. Решение таких задач методами обычного Excel может оказаться слишком громоздким, а Power Pivot предлагает лаконичный и рациональный подход.

В Power Pivot есть *функции аналитики времени*, которые предназначены для выполнения временно́го анализа данных, например для вычисления итоговых показателей за год или динамики за месяц. Они помогают уйти от использования сложных формул и упрощают анализ трендов в Excel.

[2] См. **https://clck.ru/3K3SmE**.

Добавление таблицы дат

Чтобы эффективно работать с аналитикой времени в Power Pivot, сначала нужно добавить таблицу дат. Это создаст последовательную и целостную структуру дата-время, которая повысит точность анализа данных и позволит выполнять более сложные расчеты и сравнения на основе времени. В модели данных перейдите на вкладку **Power Pivot**, выберите **Manage** (Управление), откройте вкладку **Design** (Конструктор) и выберите **Date Table | New** (Таблица дат | Создать), как показано на рис. 9.9.

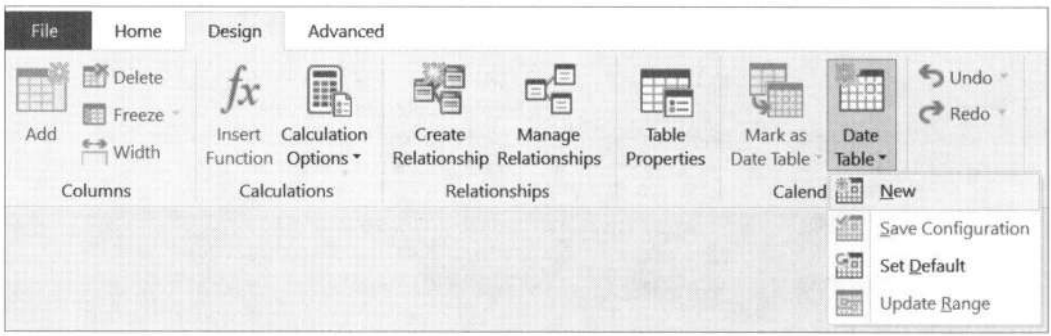


Рис. 9.9. Добавление таблицы дат в модель данных

После этого в вашей модели данных должна появиться таблица `Calendar`. Установите связь между столбцом `Date` этой таблицы и столбцом `Order Date` таблицы `orders`. Ваша модель данных станет похожа на диаграмму, показанную на рис. 9.10.

Теперь в вашей модели данных вы можете создавать различные меры с датами из таблицы дат, и все они будут связаны с полем `Order Date` в таблице `orders`.

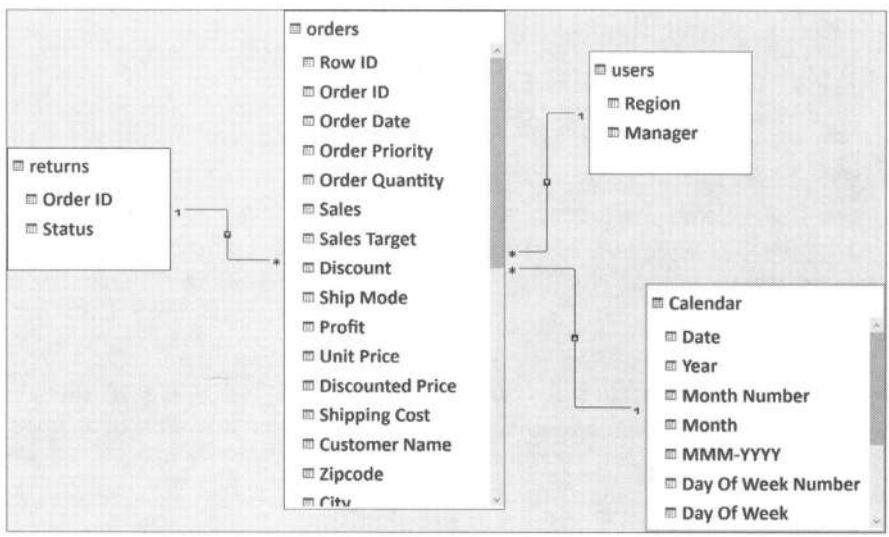


Рис. 9.10. Модель данных с таблицей `Calendar`

Изменение связи с таблицей дат с помощью `USERELATIONSHIP()`

При расчетах может быть активна только одна связь между таблицей `Calendar` и таблицей `orders`. Это означает, что если вы захотите построить свой анализ, опираясь не на `Order Date`, а на `Ship Date` (например, чтобы получить представление о логистике или удовлетворенности покупателей), вам нужно будет в `CALCULATE()` изменить эту связь с помощью функции `USERELATIONSHIP()`, чтобы указать, какие именно даты должны использоваться в этой мере.

Чтобы посмотреть, как работает таблица дат, добавьте новую сводную таблицу в рабочую книгу. Перетащите `Date Hierarchy` из таблицы `Calendar` в `Rows`, а `Total sales` из `orders` в `Values` (рис. 9.11).

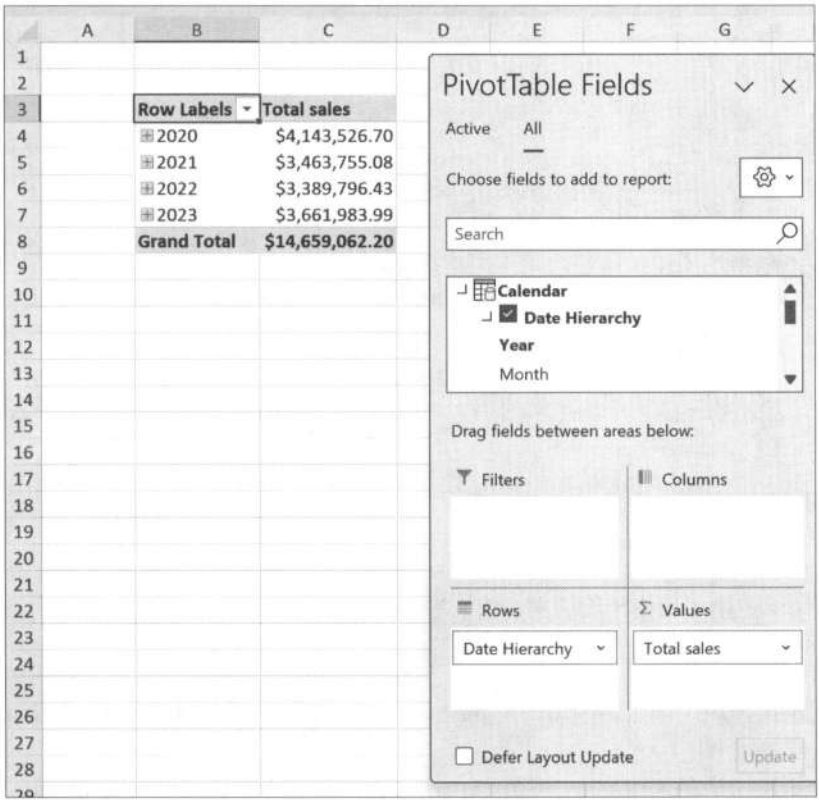


Рис. 9.11. Использование таблицы `Calendar` в сводной таблице

Создание базовых мер для аналитики времени

В DAX есть множество функций аналитики времени, позволяющих получать данные за прошлые периоды, за текущий период и т. д. Например, чтобы вычислить объем продаж с начала года (Year-To-Date, YTD), используйте формулу `TOTALYTD()`, как показано на рис. 9.12.

Чтобы проверить правильность этой меры, добавьте ее в сводную таблицу в `Rows` вместе с `Date Hierarchy`. Разверните данные за 2020 год, щелкнув на значке ⊞ рядом

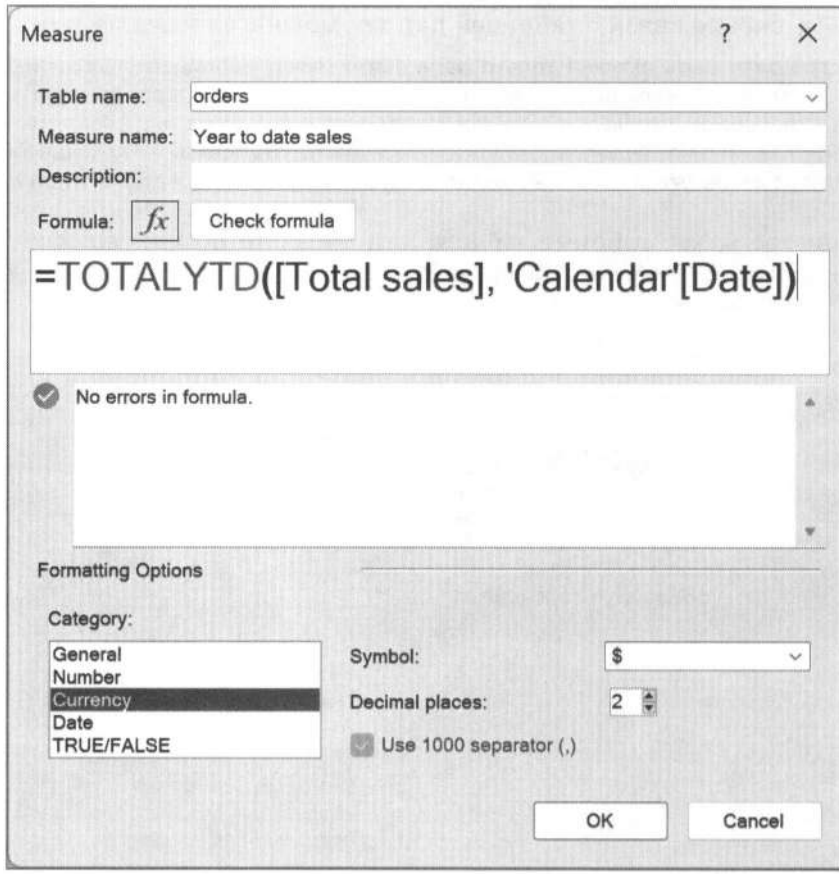


Рис. 9.12. Создание меры для расчета продаж с начала года с помощью DAX

с годом. Обратите внимание, что мера `Year to date sales` постепенно увеличивается с каждым месяцем (рис. 9.13).

Далее, чтобы вычислить объем продаж за аналогичный период прошлого года, нужно объединить уже знакомую функцию `CALCULATE()` с функцией `SAMEPERIODLASTYEAR()`, как показано на рис. 9.14.

Чтобы убедиться в правильности вычислений, добавьте эту меру в сводную таблицу вместе с `Total sales`. Вы можете заметить, что для записей за 2021 год объем месячных продаж за прошлый год `Last year sales` равен значениям `Total sales` за 2020 год (рис. 9.15).

И наконец, давайте создадим меру `Last year YTD sales`, чтобы сравнить продажи с начала года за текущий и за прошлый годы.

Для этого можно объединить функцию `CALCULATE()` с функциями `DATESYTD()` и `DATEADD()`. При этом с помощью таблицы дат будут получены все даты за текущий год, скорректированные на минус один год назад. Настройки для создания этой меры показаны на рис. 9.16.

Теперь вы можете сравнить тренды за текущий и прошлый годы (рис. 9.17).

Row Labels	Total sales	Year to date sales
2020		
January	$506,492.34	$506,492.34
February	$338,766.52	$845,258.86
March	$410,095.71	$1,255,354.57
April	$361,379.00	$1,616,733.57
May	$232,891.66	$1,849,625.23
June	$285,060.40	$2,134,685.63
July	$355,061.09	$2,489,746.71
August	$332,038.38	$2,821,785.10
September	$311,597.75	$3,133,382.85
October	$350,730.38	$3,484,113.23
November	$246,465.57	$3,730,578.80
December	$412,947.91	$4,143,526.70
2021	**$3,463,755.08**	**$3,463,755.08**
2022	**$3,389,796.43**	**$3,389,796.43**
2023	**$3,661,983.99**	**$3,661,983.99**
Grand Total	**$14,659,062.20**	**$3,661,983.99**

Рис. 9.13. Объем продаж с начала года в сводной таблице

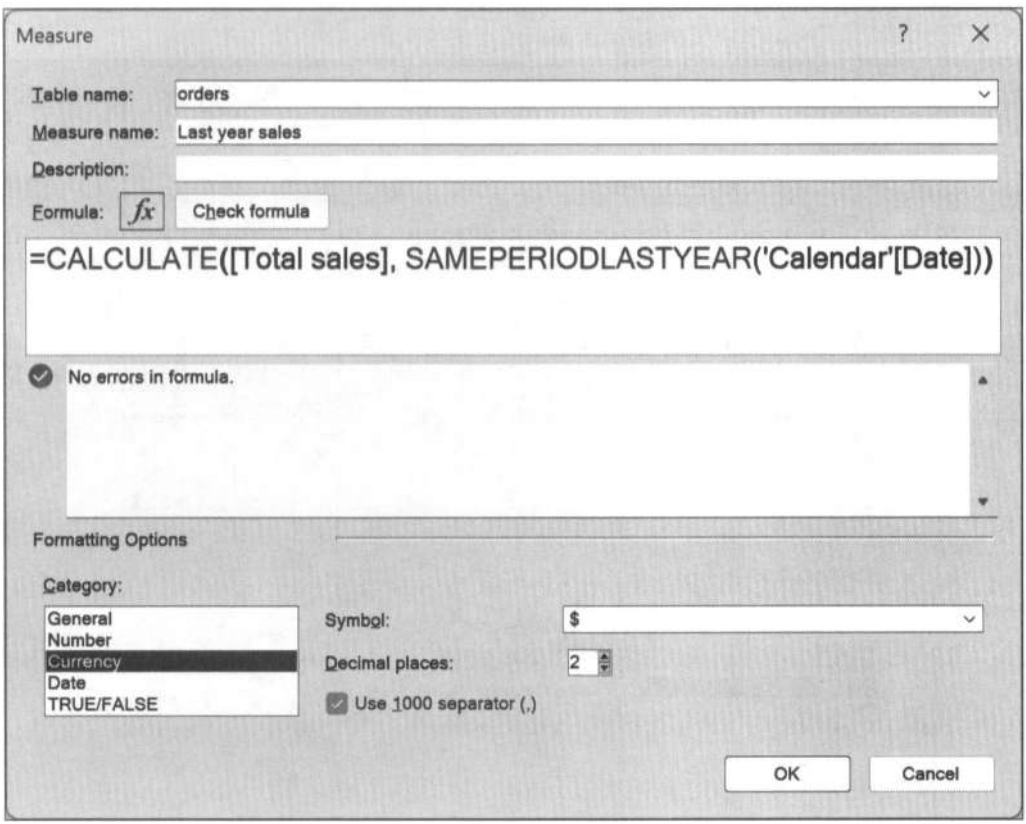


Рис. 9.14. Создание меры для расчета объема продаж за аналогичный период прошлого года

Row Labels	Total sales	Last year sales
2020		
January	$506,492.34	
February	$338,766.52	
March	$410,095.71	
April	$361,379.00	
May	$232,891.66	
June	$285,060.40	
July	$355,061.09	
August	$332,038.38	
September	$311,597.75	
October	$350,730.38	
November	$246,465.57	
December	$412,947.91	
2021		
January	$329,835.43	$506,492.34
February	$262,774.37	$338,766.52
March	$232,099.30	$410,095.71
April	$237,092.18	$361,379.00
May	$301,493.40	$232,891.66
June	$258,956.58	$285,060.40
July	$229,323.67	$355,061.09
August	$202,630.03	$332,038.38
September	$404,141.93	$311,597.75
October	$360,294.92	$350,730.38
November	$296,435.95	$246,465.57
December	$348,677.33	$412,947.91
2022	**$3,389,796.43**	**$3,463,755.08**
2023	**$3,661,983.99**	**$3,389,796.43**
Grand Total	**$14,659,062.20**	**$10,997,078.21**

Рис. 9.15. Сравнение объемов месячных продаж за текущий и прошлый годы

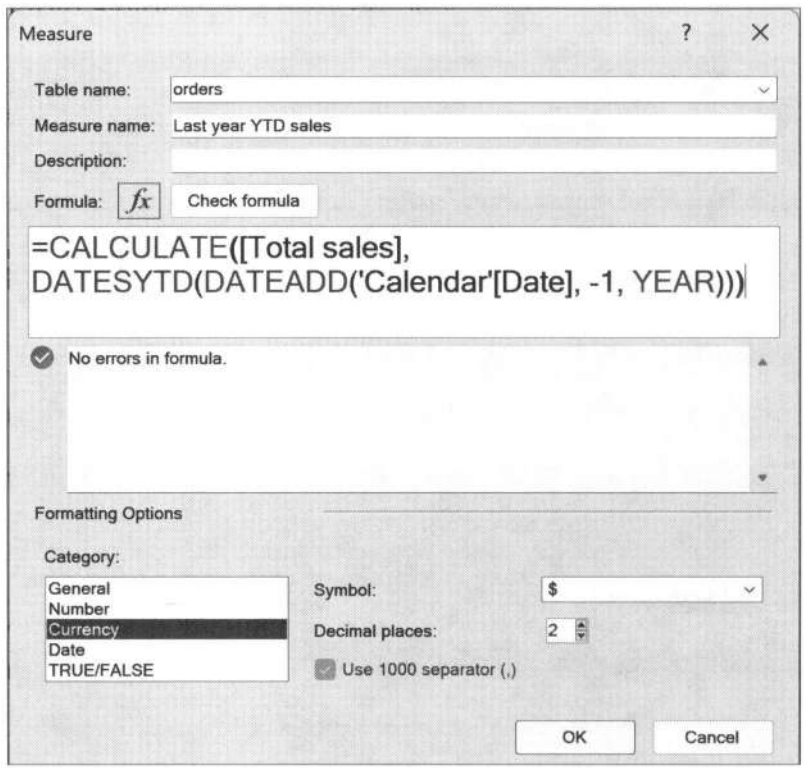


Рис. 9.16. Создание меры для расчета объема продаж за аналогичный период с начала прошлого года

Row Labels	Year to date sales	Last year YTD sales
2020		
January	$506,492.34	
February	$845,258.86	
March	$1,255,354.57	
April	$1,616,733.57	
May	$1,849,625.23	
June	$2,134,685.63	
July	$2,489,746.71	
August	$2,821,785.10	
September	$3,133,382.85	
October	$3,484,113.23	
November	$3,730,578.80	
December	$4,143,526.70	
2021		
January	$329,835.43	$506,492.34
February	$592,609.80	$845,258.86
March	$824,709.10	$1,255,354.57
April	$1,061,801.28	$1,616,733.57
May	$1,363,294.67	$1,849,625.23
June	$1,622,251.25	$2,134,685.63
July	$1,851,574.91	$2,489,746.71
August	$2,054,204.95	$2,821,785.10
September	$2,458,346.88	$3,133,382.85
October	$2,818,641.80	$3,484,113.23
November	$3,115,077.75	$3,730,578.80
December	$3,463,755.08	$4,143,526.70
2022	**$3,389,796.43**	**$3,463,755.08**
2023	**$3,661,983.99**	**$3,389,796.43**
Grand Total	**$3,661,983.99**	**$3,389,796.43**

Рис. 9.17. Сравнение объемов продаж с начала года за текущий и прошлый годы

Заключение

В DAX и Excel есть очень много разнообразных функций для дальнейшего изучения, и вы можете комбинировать их как угодно. Эта глава подошла к концу, но вас ждут новые самостоятельные открытия с Power Pivot в Excel.

До сих пор в книге мы рассматривали только Power Query и Power Pivot для очистки и моделирования данных соответственно. Но анализ данных в Excel не ограничивается этими двумя инструментами. *Часть III* книги будет посвящена другим возможностям Excel, которые помогут сделать ваши проекты более динамичными и содержательными.

Упражнения

Чтобы потренироваться в создании сложных мер DAX, воспользуйтесь набором данных о продажах в велосипедных магазинах, содержащимся в файле ch_09_exercises.csv, расположенном в папке exercises\ch_09_exercises сопроводительного ре-

позитория к этой книге[3]. Несмотря на то что в Power Pivot мы уделяли особое внимание настройке связей между несколькими таблицами из одной рабочей книги, Power Pivot можно эффективно использовать и при работе с одним файлом *.csv.

С помощью Power Query загрузите данные в Power Pivot и создайте следующие меры:

1. `accessories_rev` — общая выручка `revenue` для `product_category`, равного `Accessories`.
2. `accessories_rev_aus` — общая выручка, когда `product_category` равно `Accessories`, а `country` — `Australia`.
3. `aov_all` — вычисляется как общая выручка, деленная на общее количество заказов по всему набору данных, независимо от примененных фильтров.
4. `profit_margin_ytd` — норма прибыли с начала года.
5. `profit_margin_ly_ytd` — возвращает значение нормы прибыли с начала года за прошлый год.

Всегда создавайте промежуточные меры, которые помогают вам в создании требуемых мер. Например, если вам нужен итоговый объем продаж с начала года, удобнее сначала создать меру для общего объема продаж.

Проверьте, что ваши меры работают правильно, протестировав их в сводной таблице. Например, если вы создали меру, не зависящую от контекста фильтра, добавьте фильтр в сводную таблицу и проверьте, будут ли меняться значения. При создании меры за текущий год проверьте простую меру, просуммировав ее значения за нужный период. Кроме того, даже если мы работаем с одной таблицей, все равно полезно включать таблицу дат в модель данных для выполнения операций, связанных с датами.

Готовое решение можно посмотреть в файле ch_09_exercise_solutions.xlsx, расположенном в той же папке репозитория.

[3] См. **https://clck.ru/3K3YTK**.

ЧАСТЬ III

Инструменты аналитики в Excel

Введение в функции динамических массивов

До сих пор в этой книге рассказывалось о построении мер в Power Pivot с помощью DAX и мельком упоминался M-код в Power Query. Однако мы совсем не касались обычных формул и функций, которые долгое время были основой Excel. Казалось бы, ими можно пренебречь в пользу других, более эффектных инструментов, таких как Power Pivot и Power Query, но даже эта привычная функциональность Excel претерпела значительные улучшения, став более сильной и качественной.

В этой главе вы познакомитесь с функциями динамического массива и их возможностями и узнаете, как сортировать, фильтровать и объединять наборы данных, а также выполнять другие задачи, используя хорошо знакомую всем строку формул в Excel.

Формулы и функции в Excel

В Excel принято различать такие понятия, как формула и функция. *Формула* — это математическое выражение, которое обрабатывает данные с помощью операторов, ссылок на ячейки и констант, — например: `=B1 * B2`. А *функция* — это предопределенная формула, предоставляющая широкие возможности для анализа и обработки данных, — например: `TRIMMEAN(B1:B3)`. Формулы могут содержать в себе функции, что несколько стирает это различие. В этой главе мы будем стараться различать эти понятия, учитывая при этом их взаимозависимость.

Функции динамических массивов

Функции динамических массивов обладают очень мощными возможностями, и возникает соблазн сразу начать экспериментировать с ними. Но важно понимать, что делает эти функции особенными и чем они отличаются от подходов классического Excel. В этом разделе мы последовательно пройдем путь от массивов и ссылок на них до функций динамического массива.

Что такое массив в Excel?

Чтобы работать с примерами этой главы, откройте из папки ch_10 сопроводительного репозитория к этой книге файл ch_10.xlsx[1] и перейдите на рабочий лист `array-references`.

[1] См. **https://clck.ru/3K3cmG**.

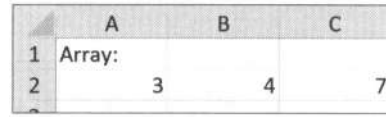


Рис. 10.1. Простой массив в Excel

Прежде всего, *массив* в Excel — это набор значений. Например, простой массив может состоять из чисел 3, 4 и 7, размещенных в ячейках `A2:C2` (рис. 10.1).

Массивы и диапазоны в Excel

`A2:C2` — это пример диапазона, но мы будем использовать его для обращения к массиву. В чем разница? *Диапазон* — это набор ячеек и их расположение, тогда как *массив* — это данные в этих ячейках. Подробнее об этом различии вы можете прочитать на форуме Microsoft Tech Community Forum[2].

Ссылки на массивы

Разобравшись, что такое массив в Excel, давайте рассмотрим различные способы написания ссылок на массивы.

Ссылки на статические массивы

Чтобы попытаться создать обычную ссылку на массив Excel, введите `=A2:C2` в ячейку `E2` и нажмите комбинацию клавиш <Ctrl>+<Shift>+<Enter>, указывающую, что вы ссылаетесь на массив значений, а не на одно значение (рис. 10.2).

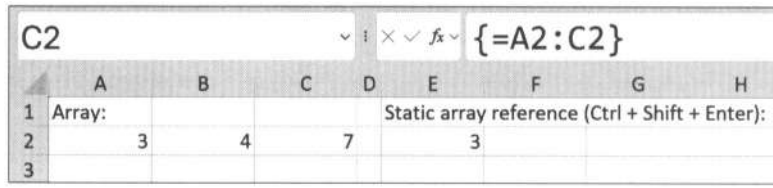


Рис. 10.2. Ссылка на простой массив Excel

В результате вы увидите, что формула оказалась заключена в фигурные скобки `{}`, но в ячейке отображаются не все значения массива (только значение **3**). Это произошло потому, что в Excel каждая ячейка предназначена для хранения только одного значения данных, а не трех, как мы пытались сделать в этом случае. Чтобы растянуть данные из массива по нескольким ячейкам, выделите диапазон `E2:G2`, введите `=A2:C2` и нажмите ту же комбинацию клавиш (рис. 10.3).

Классический подход Excel к работе с массивами имеет свои ограничения. Действия по добавлению и управлению ссылками с помощью <Ctrl>+<Shift>+<Enter> могут вызывать сложности, и такие ссылки автоматически не корректируются.

[2] См. **https://clck.ru/3K3d6Q**.

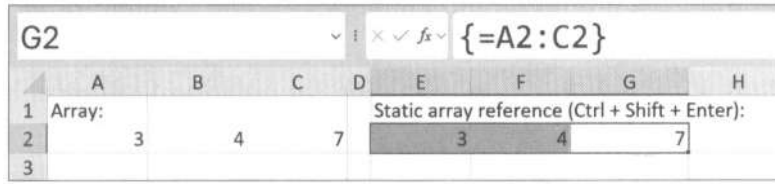


Рис. 10.3. Исправленная ссылка на массив Excel

В случаях, когда, например, между `A2:C2` вставляются или удаляются ячейки, ссылка на массив не будет автоматически перестроена с учетом нового размера массива. Такие ссылки на массивы называются *статическими*, поскольку они динамически не корректируются при изменении структуры электронной таблицы или количества ячеек.

Ссылки на динамические массивы

Динамические массивы, позволяющие обойти ограничения обычных статических массивов, появились в Excel в 2018 г. Теперь, чтобы сослаться на `A2:C2`, нужно просто ввести `=A2:C2`, например в ячейку `E5`, и нажать клавишу <Enter> (рис. 10.4).

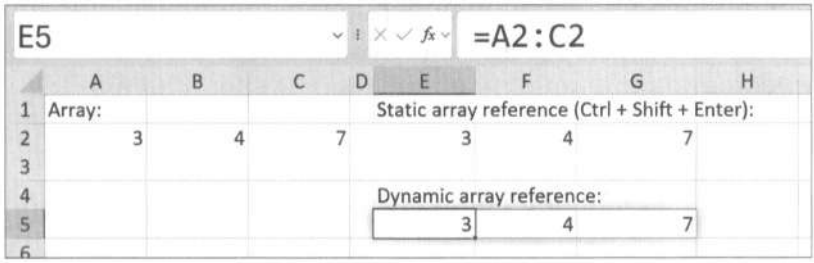


Рис. 10.4. Ссылка на динамический массив Excel

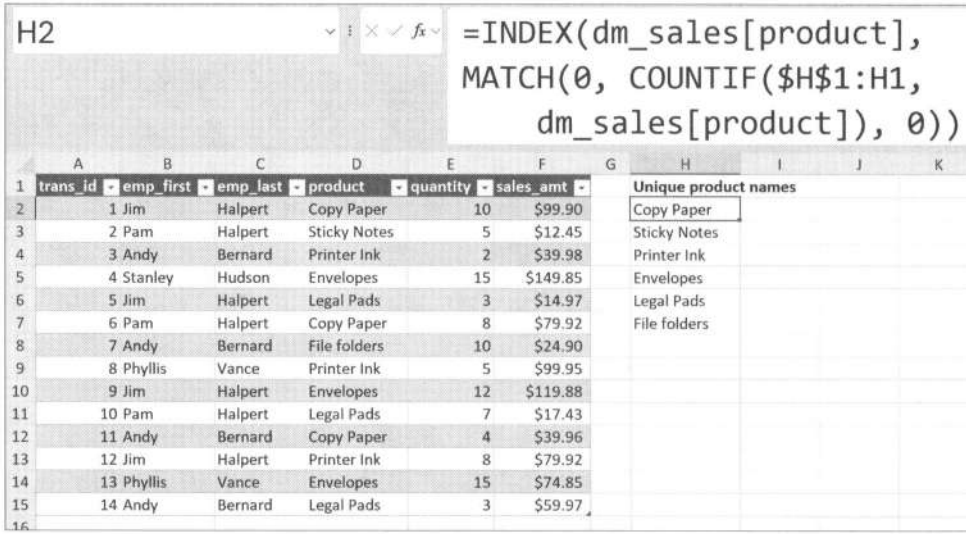
Используя эту ссылку, Excel сам определит количество ячеек в массиве. То есть если вы вставите или удалите ячейки между `A2` и `C2`, динамическая ссылка на массив автоматически изменит свой размер с учетом изменений размера массива. Такая динамическая корректировка экономит наше время и силы, избавляя от необходимости вручную обновлять ссылки при каждом изменении структуры данных.

Формулы массива

Сравнив поведение ссылок на массивы в классическом и современном Excel, далее мы рассмотрим, как работают функции, использующие эти ссылки.

Формулы статического массива

Возьмем следующий пример, в котором задействована формула статического массива для получения списка уникальных товаров в наборе данных о продажах (рис. 10.5). Вы можете найти этот пример на рабочем листе `array-functions` в файле ch_10.xlsx.



trans_id	emp_first	emp_last	product	quantity	sales_amt
1	Jim	Halpert	Copy Paper	10	$99.90
2	Pam	Halpert	Sticky Notes	5	$12.45
3	Andy	Bernard	Printer Ink	2	$39.98
4	Stanley	Hudson	Envelopes	15	$149.85
5	Jim	Halpert	Legal Pads	3	$14.97
6	Pam	Halpert	Copy Paper	8	$79.92
7	Andy	Bernard	File folders	10	$24.90
8	Phyllis	Vance	Printer Ink	5	$99.95
9	Jim	Halpert	Envelopes	12	$119.88
10	Pam	Halpert	Legal Pads	7	$17.43
11	Andy	Bernard	Copy Paper	4	$39.96
12	Jim	Halpert	Printer Ink	8	$79.92
13	Phyllis	Vance	Envelopes	15	$74.85
14	Andy	Bernard	Legal Pads	3	$59.97

Unique product names
Copy Paper
Sticky Notes
Printer Ink
Envelopes
Legal Pads
File folders

Рис. 10.5. Формула статического массива

Сейчас не нужно вникать в то, из чего состоит эта формула, потому что далее мы рассмотрим более рациональную альтернативу. Пока же заметьте, что статический массив может вызывать сложности, когда нужно заранее знать количество возвращаемых элементов.

trans_id	emp_first	emp_last	product	quantity	sales_amt
1	Jim	Halpert	Copy Paper	10	$99.90
2	Pam	Halpert	Sticky Notes	5	$12.45
3	Andy	Bernard	Printer Ink	2	$39.98
4	Stanley	Hudson	Envelopes	15	$149.85
5	Jim	Halpert	Legal Pads	3	$14.97
6	Pam	Halpert	Copy Paper	8	$79.92
7	Andy	Bernard	File folders	10	$24.90
8	Phyllis	Vance	Printer Ink	5	$99.95
9	Jim	Halpert	Envelopes	12	$119.88
10	Pam	Halpert	Legal Pads	7	$17.43
11	Andy	Bernard	Copy Paper	4	$39.96
12	Jim	Halpert	Printer Ink	8	$79.92
13	Phyllis	Vance	Envelopes	15	$74.85
14	Andy	Bernard	Legal Pads	3	$59.97
15	Stanley	Hudson	Rubber bands	60	$14.94

Unique product names
Copy Paper
Sticky Notes
Printer Ink
Envelopes
Legal Pads
File folders

Рис. 10.6. В списке уникальных товаров не появилось значение `Rubber bands` (из строки 16)

Дело в том, что, как и в случае со ссылками на массивы, при использовании формул статического массива отсутствует автоматическое обновление списка при внесении изменений в уникальные значения, что делает этот подход громоздким и неудобным. Например, если в таблицу `dm_sales` добавятся новые продажи — как, скажем,

Rubber bands на рис. 10.6, все такие новые уникальные значения автоматически не появятся в результатах функции массива.

Чтобы увидеть это дополнительное значение в списке, вам придется расширить действие формулы массива в столбце H еще на одну строку.

Функции динамического массива

А вот функции динамического массива в таких случаях очень хорошо работают. В Excel появилась функция UNIQUE() (УНИК()), специально разработанная для решения нашей задачи (рис. 10.7).

trans_id	emp_first	emp_last	product	quantity	sales_amt
1	Jim	Halpert	Copy Paper	10	99.9
2	Pam	Beesly	Sticky Notes	5	12.45
3	Andy	Bernard	Printer Ink	2	39.98
4	Stanley	Hudson	Envelopes	15	149.85
5	Jim	Halpert	Legal Pads	3	14.97
6	Pam	Beesly	Copy Paper	8	79.92
7	Andy	Bernard	Sticky Notes	10	24.9
8	Phyllis	Vance	Printer Ink	5	99.95
9	Jim	Halpert	Envelopes	12	119.88
10	Pam	Beesly	Legal Pads	7	17.43
11	Andy	Bernard	Copy Paper	4	$39.96
12	Jim	Halpert	Printer Ink	8	$79.92
13	Phyllis	Vance	Envelopes	15	$74.85
14	Andy	Bernard	Legal Pads	3	$59.97
15	Stanley	Hudson	Rubber bands	60	$14.94

Рис. 10.7. Список уникальных значений, полученный с помощью функции UNIQUE()

Если при использовании функции UNIQUE() вы видите ошибку #SPILL! (#ПЕРЕНОС!), проверьте, что под текущей ячейкой есть достаточное количество пустых ячеек. Эта ошибка возникает, когда результаты функции перекрывают непустые соседние ячейки.

Функции динамического массива значительно превосходят по своим возможностям традиционные формулы массива, поскольку мгновенно обновляют результат в ответ на изменения во входных данных. Такое динамическое поведение позволяет не пересчитывать или не обновлять формулы вручную, обеспечивая непрерывный и эффективный рабочий процесс.

Использование функций динамического массива

Давайте теперь рассмотрим несколько примеров, в которых показаны возможности функций динамического массива. И сначала мы вернемся к функции UNIQUE(), о которой уже упоминалось ранее.

Поиск уникальных и неповторяющихся значений с помощью функции *UNIQUE()*

В предыдущем примере для создания списка уникальных товаров использовалась функция динамического массива `UNIQUE()` (`УНИК()`). Для дальнейшего изучения этой функции мы продолжим работать с таблицей `dm_sales` из рабочей книги, содержащейся в файле ch_10.xlsx.

Параметры и аргументы в Excel

Термины «параметры» и «аргументы» в Excel часто используются как взаимозаменяемые понятия, но они несут разный смысл. *Параметры* — это переменные внутри функции, вместо которых потом подставляются переданные значения, в то время как *аргументы* — это фактические значения или ссылки, указанные в формуле. Например, функция `ABS()` принимает параметр с именем `number`, а при вызове функции `ABS(-10)` аргументом является число `-10`.

Функция `UNIQUE()` имеет три параметра, два из которых являются необязательными. Описания этих параметров приведены в табл. 10.1.

Таблица 10.1. Параметры функции `UNIQUE()`

Параметр	Описание
`range`	Обязательный параметр, определяющий диапазон или массив данных, из которого нужно извлечь уникальные значения
`[by_col]`	Необязательный параметр, определяющий, в каком направлении ищутся уникальные значения: по столбцам или по строкам. По умолчанию этот параметр равен `FALSE` (`ЛОЖЬ`), и возвращаются уникальные строки. Если вы передадите в этот параметр значение `TRUE` (`ИСТИНА`), будут возвращаться уникальные столбцы
`[exactly_once]`	Необязательный параметр, определяющий, что уникальными считаются лишь те значения, которые встречаются только один раз. По умолчанию извлекаются все уникальные значения, независимо от частоты их появления в диапазоне. Если передать в этот параметр значение `TRUE`, то будут возвращаться лишь те значения, которые встречаются только один раз

Разница между уникальными и отличающимися значениями

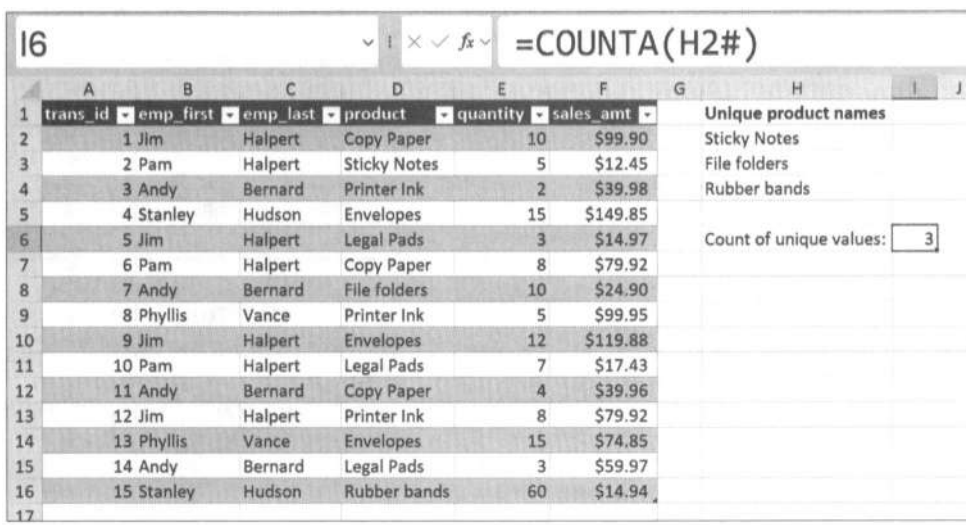
В терминологии баз данных *уникальными* (unique) значениями считаются те, которые встречаются в заданном диапазоне только один раз. Из-за этого название функции `UNIQUE()` немного вводит в заблуждение. На самом деле эта функция возвращает не строго уникальные, а *отличающиеся* (distinct) значения — т. е. те, которые могут встречаться один или более раз. Тем не менее с помощью этой функции можно получить действительно уникальные значения, передав в качестве третьего параметра `TRUE` (рис. 10.8).

	trans_id	emp_first	emp_last	product	quantity	sales_amt
2	1	Jim	Halpert	Copy Paper	10	$99.90
3	2	Pam	Halpert	Sticky Notes	5	$12.45
4	3	Andy	Bernard	Printer Ink	2	$39.98
5	4	Stanley	Hudson	Envelopes	15	$149.85
6	5	Jim	Halpert	Legal Pads	3	$14.97
7	6	Pam	Halpert	Copy Paper	8	$79.92
8	7	Andy	Bernard	File folders	10	$24.90
9	8	Phyllis	Vance	Printer Ink	5	$99.95
10	9	Jim	Halpert	Envelopes	12	$119.88
11	10	Pam	Halpert	Legal Pads	7	$17.43
12	11	Andy	Bernard	Copy Paper	4	$39.96
13	12	Jim	Halpert	Printer Ink	8	$79.92
14	13	Phyllis	Vance	Envelopes	15	$74.85
15	14	Andy	Bernard	Legal Pads	3	$59.97
16	15	Stanley	Hudson	Rubber bands	60	$14.94

Рис. 10.8. Поиск действительно уникальных значений с помощью `UNIQUE()`

Использование оператора динамического диапазона

В Excel часто приходится добавлять новые вычисления поверх уже существующих — например, агрегировать результаты вычисляемого столбца. *Оператор динамического диапазона*, обозначаемый символом решетки (#), упрощает агрегирование для функций динамического массива. Как и динамические массивы, этот оператор автоматически расширяет диапазон, чтобы в него вошли все данные, избавляя нас от необходимости вручную вводить формулы массива и изменять размеры диапазонов. Этот оператор повышает эффективность и лаконичность формул для агрегирования данных в Excel.



	trans_id	emp_first	emp_last	product	quantity	sales_amt
2	1	Jim	Halpert	Copy Paper	10	$99.90
3	2	Pam	Halpert	Sticky Notes	5	$12.45
4	3	Andy	Bernard	Printer Ink	2	$39.98
5	4	Stanley	Hudson	Envelopes	15	$149.85
6	5	Jim	Halpert	Legal Pads	3	$14.97
7	6	Pam	Halpert	Copy Paper	8	$79.92
8	7	Andy	Bernard	File folders	10	$24.90
9	8	Phyllis	Vance	Printer Ink	5	$99.95
10	9	Jim	Halpert	Envelopes	12	$119.88
11	10	Pam	Halpert	Legal Pads	7	$17.43
12	11	Andy	Bernard	Copy Paper	4	$39.96
13	12	Jim	Halpert	Printer Ink	8	$79.92
14	13	Phyllis	Vance	Envelopes	15	$74.85
15	14	Andy	Bernard	Legal Pads	3	$59.97
16	15	Stanley	Hudson	Rubber bands	60	$14.94

Рис. 10.9. Агрегирование динамического массива с помощью оператора #

Количество уникальных значений, полученных по формуле, приведенной на рис. 10.8, можно определить с помощью функции COUNTA() (СЧЁТЗ()). При выборе диапазона H2:H4 Excel автоматически добавит ссылку на него, используя оператор динамического диапазона, обозначаемый с помощью # (рис. 10.9).

Оператор динамического диапазона очень помогает при построении зависимых выпадающих списков, динамических диаграмм и в ряде других случаев, но их описание выходит на рамки этой книги.

Фильтрация записей с помощью функции *FILTER()*

Привычные выпадающие списки Excel для фильтрации данных интуитивно понятны, но имеют свои ограничения. Например, при их применении исходные данные уже нельзя посмотреть полностью. В таких случаях приходится создавать отдельную копию данных, а затем применять фильтр к этой копии, аналогично тому, как Power Query работает с очисткой данных. Кроме того, применение сложных логических условий для фильтрации по нескольким столбцам может показаться трудоемким и скучным.

Чтобы обойти эти ограничения, в Excel добавили функцию динамического массива FILTER() (ФИЛЬТР()). Эта функция имеет три параметра, которые подробно описаны в табл. 10.2.

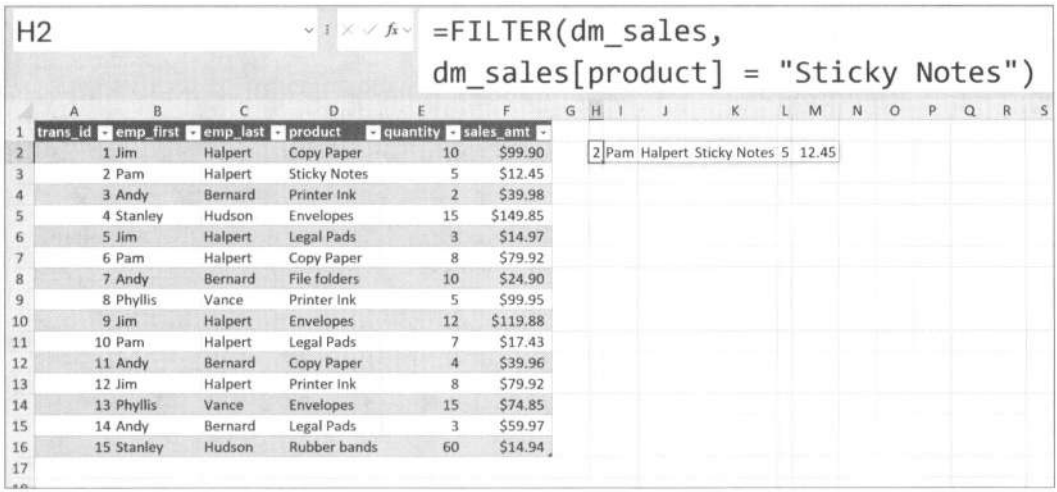
Таблица 10.2. Параметры функции FILTER()

Параметр	Описание
array	Обязательный параметр, определяющий диапазон или массив данных, который нужно отфильтровать
include	Обязательный параметр, определяющий критерий или условие фильтрации. Значения, удовлетворяющие этому условию, будут включены в отфильтрованный результат. Это может быть логическое выражение, значение для сравнения или формула, которая для каждого элемента массива возвращает TRUE или FALSE
[if_empty]	Необязательный параметр, определяющий значение, которое будет возвращаться, если отфильтрованный результат оказался пустым. По умолчанию, если ни одно значение не удовлетворяет условию фильтрации, функция возвращает массив со значениями ошибки #CALC! (#ВЫЧИСЛ!)

Предположим, нам нужно отфильтровать таблицу dm_sales так, чтобы возвращались только записи с product, равным Sticky Notes. Результат показан на рис. 10.10.

По умолчанию функция FILTER() нечувствительна к регистру. В приведенном примере фильтрация по Sticky Notes и sticky notes вернет один и тот же результат. Чтобы выполнить фильтрацию с учетом регистра, нужно объединить функции FILTER() и EXACT() (СОВПАД()), например:

```
=FILTER(dm_sales, EXACT(dm_sales[product], "Sticky Notes"))
```



trans_id	emp_first	emp_last	product	quantity	sales_amt
1	Jim	Halpert	Copy Paper	10	$99.90
2	Pam	Halpert	Sticky Notes	5	$12.45
3	Andy	Bernard	Printer Ink	2	$39.98
4	Stanley	Hudson	Envelopes	15	$149.85
5	Jim	Halpert	Legal Pads	3	$14.97
6	Pam	Halpert	Copy Paper	8	$79.92
7	Andy	Bernard	File folders	10	$24.90
8	Phyllis	Vance	Printer Ink	5	$99.95
9	Jim	Halpert	Envelopes	12	$119.88
10	Pam	Halpert	Legal Pads	7	$17.43
11	Andy	Bernard	Copy Paper	4	$39.96
12	Jim	Halpert	Printer Ink	8	$79.92
13	Phyllis	Vance	Envelopes	15	$74.85
14	Andy	Bernard	Legal Pads	3	$59.97
15	Stanley	Hudson	Rubber bands	60	$14.94

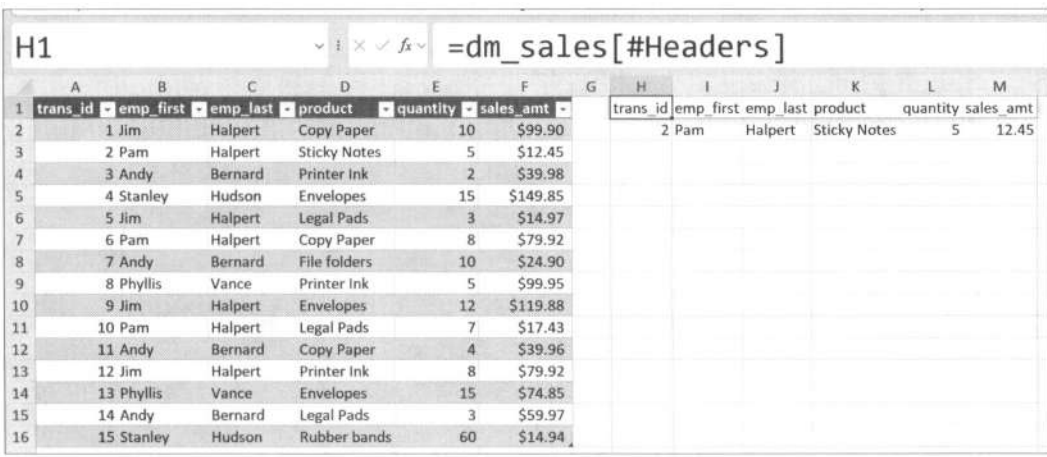
2	Pam	Halpert	Sticky Notes	5	12.45

Рис. 10.10. Простой пример использования функции `FILTER()`

Добавление заголовков столбцов

С функцией `FILTER()` уже можно работать, но ей не хватает одной важной особенности — она возвращает только совпадающие строки, но не заголовки столбцов. Чтобы вывести эти заголовки, необходимо использовать динамическую ссылку на заголовки таблицы. Краткое описание структурированных ссылок было приведено в *главе 1*. Добавьте такую ссылку над результатом фильтрации, чтобы динамически выводились заголовки таблицы, как показано на рис. 10.11.

Функция `FILTER()` и другие функций динамического массива по умолчанию не включают заголовки таблицы Excel в свои результаты.



trans_id	emp_first	emp_last	product	quantity	sales_amt
1	Jim	Halpert	Copy Paper	10	$99.90
2	Pam	Halpert	Sticky Notes	5	$12.45
3	Andy	Bernard	Printer Ink	2	$39.98
4	Stanley	Hudson	Envelopes	15	$149.85
5	Jim	Halpert	Legal Pads	3	$14.97
6	Pam	Halpert	Copy Paper	8	$79.92
7	Andy	Bernard	File folders	10	$24.90
8	Phyllis	Vance	Printer Ink	5	$99.95
9	Jim	Halpert	Envelopes	12	$119.88
10	Pam	Halpert	Legal Pads	7	$17.43
11	Andy	Bernard	Copy Paper	4	$39.96
12	Jim	Halpert	Printer Ink	8	$79.92
13	Phyllis	Vance	Envelopes	15	$74.85
14	Andy	Bernard	Legal Pads	3	$59.97
15	Stanley	Hudson	Rubber bands	60	$14.94

trans_id	emp_first	emp_last	product	quantity	sales_amt
2	Pam	Halpert	Sticky Notes	5	12.45

Рис. 10.11. Результаты функции `FILTER()` с заголовками столбцов

Фильтрация по нескольким условиям

В отличие от обычной фильтрации с помощью выпадающего списка, в функции FILTER() можно использовать формулу для фильтрации данных. Это открывает широкие возможности, сохраняя при этом интуитивно понятный и простой подход к написанию условий фильтрации.

Чтобы включить несколько условий в функцию FILTER(), используйте символ * (звездочка) вместо оператора И и символ + (плюс) вместо оператора ИЛИ.

Условие И

Чтобы найти записи, в которых product равен Copy Paper И количество проданного товара больше 5, можно скомбинировать эти условия с помощью символа звездочки (*), заключив их в круглые скобки:

```
=FILTER(dm_sales, (dm_sales[product] = "Copy Paper") *
 (dm_sales[quantity] > 5))
```

Условие ИЛИ

Если вы хотите найти записи, удовлетворяющие *любому* из этих условий, замените символ * на +, чтобы применился оператор ИЛИ:

```
=FILTER(dm_sales, (dm_sales[product] = "Copy Paper") +
(dm_sales[quantity] > 5))
```

Вложенные условия И/ИЛИ

Чтобы написать функцию фильтрации с вложенными условиями И/ИЛИ, группируйте операторы с помощью круглых скобок. Например, следующая функция фильтрации выведет записи, в которых поле sales_amt не менее 100 ИЛИ quantity не менее 10 вместе с product, равным Envelopes:

```
=FILTER(dm_sales,
 (dm_sales[sales_amt] >= 100) +
 ((dm_sales[quantity] >= 10) * (dm_sales[product] = "Envelopes")))
```

* * *

Таким образом, следуя этим простым правилам, вы можете задавать в функции FILTER() множественные условия.

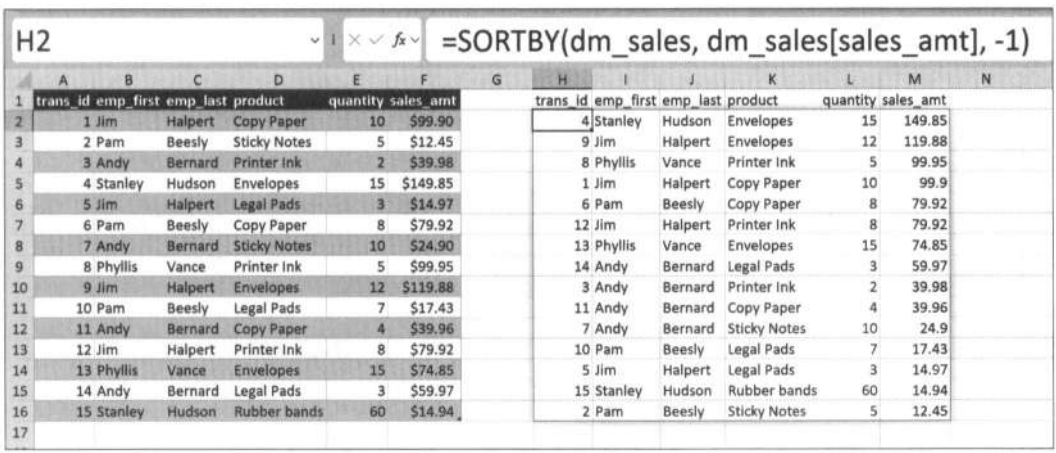
Сортировка с помощью функции *SORTBY()*

SORTBY() (СОРТПО()) — это функция динамического массива, позволяющая сортировать записи одновременно по нескольким критериям. Синтаксис аналогичен синтаксису функции SUMIFS() (СУММЕСЛИМН()). В табл. 10.3 приведены параметры этой функции.

Таблица 10.3. Параметры функции `SORTBY()`

Параметр	Описание
`array`	Обязательный параметр, определяющий массив или диапазон, который нужно отсортировать
`by_array1`	Обязательный параметр, определяющий массив или диапазон, значения которого будут использоваться для сортировки
`[sort_order1]`	Необязательный параметр, определяющий порядок сортировки: `1` — по возрастанию, `-1` — по убыванию. По умолчанию используется сортировка по возрастанию
`[by_array2]`	Необязательный параметр, определяющий второй массив или диапазон, по значениям которого будет выполнена сортировка в рамках `by_array1`
`[sort_order2]`	Необязательный параметр, определяющий порядок сортировки по `by_array2`: `1` — по возрастанию, `-1` — по убыванию. По умолчанию используется сортировка по возрастанию

Для примера на рис. 10.12 показано, как с помощью функции `SORTBY()` мы можем отсортировать набор данных по столбцу `sales_amt` в порядке убывания.



	A	B	C	D	E	F
1	trans_id	emp_first	emp_last	product	quantity	sales_amt
2	1	Jim	Halpert	Copy Paper	10	$99.90
3	2	Pam	Beesly	Sticky Notes	5	$12.45
4	3	Andy	Bernard	Printer Ink	2	$39.98
5	4	Stanley	Hudson	Envelopes	15	$149.85
6	5	Jim	Halpert	Legal Pads	3	$14.97
7	6	Pam	Beesly	Copy Paper	8	$79.92
8	7	Andy	Bernard	Sticky Notes	10	$24.90
9	8	Phyllis	Vance	Printer Ink	5	$99.95
10	9	Jim	Halpert	Envelopes	12	$119.88
11	10	Pam	Beesly	Legal Pads	7	$17.43
12	11	Andy	Bernard	Copy Paper	4	$39.96
13	12	Jim	Halpert	Printer Ink	8	$79.92
14	13	Phyllis	Vance	Envelopes	15	$74.85
15	14	Andy	Bernard	Legal Pads	3	$59.97
16	15	Stanley	Hudson	Rubber bands	60	$14.94
17						

H	I	J	K	L	M
trans_id	emp_first	emp_last	product	quantity	sales_amt
4	Stanley	Hudson	Envelopes	15	149.85
9	Jim	Halpert	Envelopes	12	119.88
8	Phyllis	Vance	Printer Ink	5	99.95
1	Jim	Halpert	Copy Paper	10	99.9
6	Pam	Beesly	Copy Paper	8	79.92
12	Jim	Halpert	Printer Ink	8	79.92
13	Phyllis	Vance	Envelopes	15	74.85
14	Andy	Bernard	Legal Pads	3	59.97
3	Andy	Bernard	Printer Ink	2	39.98
11	Andy	Bernard	Copy Paper	4	39.96
7	Andy	Bernard	Sticky Notes	10	24.9
10	Pam	Beesly	Legal Pads	7	17.43
5	Jim	Halpert	Legal Pads	3	14.97
15	Stanley	Hudson	Rubber bands	60	14.94
2	Pam	Beesly	Sticky Notes	5	12.45

Рис. 10.12. Сортировка таблицы Excel с помощью функции `SORTBY()`

Сортировка по нескольким диапазонам

Функция `SORTBY()` позволяет сортировать данные по нескольким диапазонам, и для каждого из них можно указать свой порядок сортировки. Например, так можно отсортировать данные по `emp_last` по убыванию и по `product` по возрастанию:

```
=SORTBY(dm_sales, dm_sales[emp_last], -1, dm_sales[product], 1)
```

Вы можете добавить еще параметры, чтобы отсортировать набор данных по любому количеству столбцов, указав для каждого из них нужный порядок сортировки.

Сортировка без включения столбца сортировки в результат

Функция SORTBY() может даже отсортировать диапазон по другому диапазону, не включая этот другой диапазон, использованный для сортировки, в результат.

Пусть вы хотите получить список идентификаторов продаж, отсортированных в порядке убывания по объему продаж. Вместо того чтобы передавать всю таблицу dm_sales в качестве первого аргумента, укажите только столбец trans_id. Последующие шаги должны быть вам знакомы. Результат будет состоять из одного столбца (рис. 10.13).

	A	B	C	D	E	F	G	H	I	J
1	trans_id	emp_first	emp_last	product	quantity	sales_amt				
2	1	Jim	Halpert	Copy Paper	10	$99.90		4		
3	2	Pam	Halpert	Sticky Notes	5	$12.45		9		
4	3	Andy	Bernard	Printer Ink	2	$39.98		8		
5	4	Stanley	Hudson	Envelopes	15	$149.85		1		
6	5	Jim	Halpert	Legal Pads	3	$14.97		6		
7	6	Pam	Halpert	Copy Paper	8	$79.92		12		
8	7	Andy	Bernard	File folders	10	$24.90		13		
9	8	Phyllis	Vance	Printer Ink	5	$99.95		14		
10	9	Jim	Halpert	Envelopes	12	$119.88		3		
11	10	Pam	Halpert	Legal Pads	7	$17.43		11		
12	11	Andy	Bernard	Copy Paper	4	$39.96		7		
13	12	Jim	Halpert	Printer Ink	8	$79.92		10		
14	13	Phyllis	Vance	Envelopes	15	$74.85		5		
15	14	Andy	Bernard	Legal Pads	3	$59.97		15		
16	15	Stanley	Hudson	Rubber bands	60	$14.94		2		

Рис. 10.13. Результат SORTBY() в виде одного столбца

Современный поиск с помощью функции *XLOOKUP()*

До сих пор в примерах с функциями динамического массива использовалась только одна таблица. Но, как правило, данные поступают из нескольких таблиц, а это требует их соединения. Хотя Power Query и Power Pivot предлагают свои собственные способы объединения данных из разных источников, тем не менее для решения таких задач, как построение модели на основе пользовательского ввода, выполнение беглого анализа в режиме реального времени и пр., преимущество остается за быстрыми, динамичными и интерактивными формулами Excel.

Функция XLOOKUP() (ПРОСМОТРX()) представляет собой универсальную альтернативу классической функции VLOOKUP() (ВПР()), использующую возможности динамических массивов.

Для изучения этой функции откройте в файле ch_10.xlsx рабочий лист xlookup, содержащий три отдельные таблицы, связанные с продажами канцелярских товаров.

Сравнение функций *XLOOKUP()* и *VLOOKUP()*

Функция XLOOKUP() предлагает пользователям, знакомым с VLOOKUP(), привычный способ переноса данных из одной таблицы в другую на основе какого-то общего значения. Кроме того, она дополнительно предоставляет ряд более универсальных и сложных способов поиска. Основные различия между VLOOKUP() и XLOOKUP() представлены в табл. 10.4.

Таблица 10.4. Сравнение VLOOKUP() и XLOOKUP()

Характеристика	VLOOKUP()	XLOOKUP()
Направление поиска	Поиск возможен только по вертикали	Поиск может выполняться как по вертикали, так и по горизонтали
Направление возвращаемых данных	Может возвращать только значения, расположенные справа от просматриваемого столбца	Может возвращать значения из столбцов слева и справа от просматриваемого столбца
Обработка ошибок	Возвращает #N/A, если значение не найдено	Можно определить значение по умолчанию для ненайденных значений

У функции XLOOKUP() есть шесть параметров (табл. 10.5).

Таблица 10.5. Параметры функции XLOOKUP()

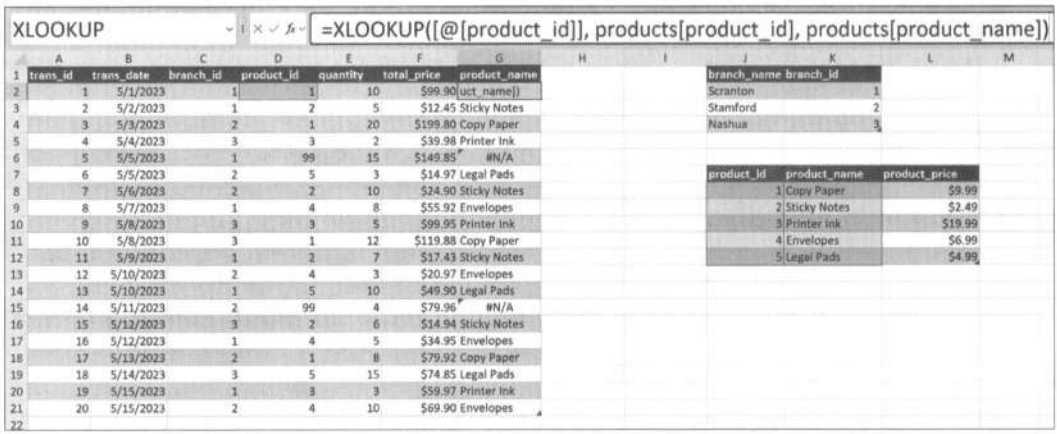
Параметр	Описание
lookup_value	Обязательный параметр, определяющий значение, которое надо найти в массиве lookup_array
lookup_array	Обязательный параметр, определяющий диапазон или массив, где будет искаться значение lookup_value
return_array	Обязательный параметр, определяющий диапазон или массив, из которого нужно вернуть данные
[if_not_found]	Необязательный параметр, определяющий значение, которое будет возвращаться, если значение lookup_value не найдено
[match_mode]	Необязательный параметр, определяющий режим сопоставления значений при поиске
[search_mode]	Необязательный параметр, определяющий режим поиска значения lookup_value

Мы рассмотрим здесь примеры только с первыми четырьмя параметрами функции XLOOKUP(). Более подробное их описание можно найти в *главе 12* книги: Alan Murray. «Advanced Excel Formulas: Unleashing Brilliance with Excel Formulas» (Apress, 2022)[3].

[3] См. **https://clck.ru/3K9uur**.

Базовые возможности функции *XLOOKUP()*

Начнем с простого примера: таблица `transactions` содержит столбец `product_id`, для которого надо найти и сопоставить соответствующие значения `product_name`. Здесь в качестве значения для поиска будет выступать столбец `product_id`, а в качестве возвращаемого значения — `product_name` (рис. 10.14).



trans_id	trans_date	branch_id	product_id	quantity	total_price	product_name
1	5/1/2023	1	1	10	$99.90	uct_name])
2	5/2/2023	1	2	5	$12.45	Sticky Notes
3	5/3/2023	2	1	20	$199.80	Copy Paper
4	5/4/2023	3	3	2	$39.98	Printer Ink
5	5/5/2023	1	99	15	$149.85	#N/A
6	5/5/2023	2	5	3	$14.97	Legal Pads
7	5/6/2023	2	2	10	$24.90	Sticky Notes
8	5/7/2023	1	4	8	$55.92	Envelopes
9	5/8/2023	3	3	5	$99.95	Printer Ink
10	5/8/2023	3	1	12	$119.88	Copy Paper
11	5/9/2023	1	2	7	$17.43	Sticky Notes
12	5/10/2023	2	4	3	$20.97	Envelopes
13	5/10/2023	1	5	10	$49.90	Legal Pads
14	5/11/2023	2	99	4	$79.96	#N/A
15	5/12/2023	3	2	6	$14.94	Sticky Notes
16	5/12/2023	1	4	5	$34.95	Envelopes
17	5/13/2023	2	1	8	$79.92	Copy Paper
18	5/14/2023	3	5	15	$74.85	Legal Pads
19	5/15/2023	1	3	3	$59.97	Printer Ink
20	5/15/2023	2	4	10	$69.90	Envelopes

branch_name	branch_id
Scranton	1
Stamford	2
Nashua	3

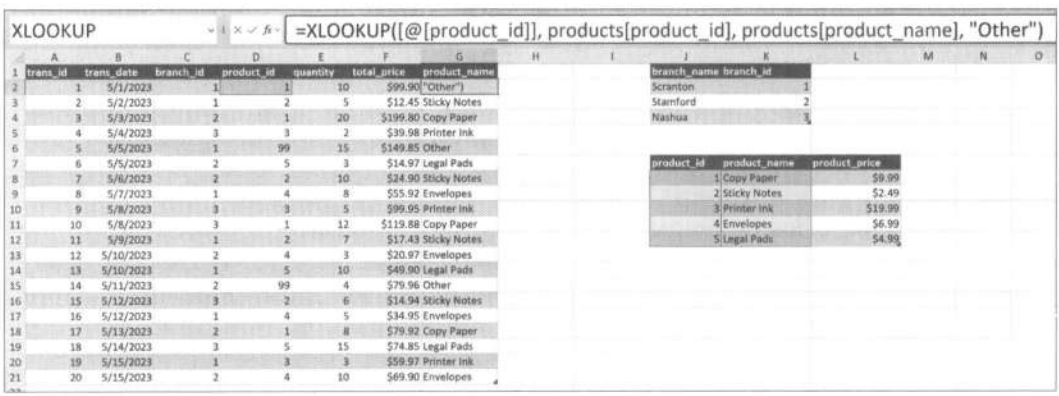
product_id	product_name	product_price
1	Copy Paper	$9.99
2	Sticky Notes	$2.49
3	Printer Ink	$19.99
4	Envelopes	$6.99
5	Legal Pads	$4.99

Рис. 10.14. Базовый вариант использования функции `XLOOKUP()`

Обработка ошибок с помощью функции *XLOOKUP()*

Поиск значения `product_id`, равного 99, возвращает ошибку. Использование `#N/A` в качестве результата при отсутствии совпадения может стать проблемой — привести к ошибкам в расчетах и запутать конечных пользователей, которые будут спрашивать, почему возвращается `#N/A`.

Чтобы заменить это сообщение об ошибке понятным результатом, передайте в функцию `XLOOKUP()` четвертый необязательный параметр, — допустим, для нашего случае вы решили, что товары с идентификатором 99 должны быть помечены как `Other` (рис. 10.15).



trans_id	trans_date	branch_id	product_id	quantity	total_price	product_name
1	5/1/2023	1	1	10	$99.90	"Other")
2	5/2/2023	1	2	5	$12.45	Sticky Notes
3	5/3/2023	2	1	20	$199.80	Copy Paper
4	5/4/2023	3	3	2	$39.98	Printer Ink
5	5/5/2023	1	99	15	$149.85	Other
6	5/5/2023	2	5	3	$14.97	Legal Pads
7	5/6/2023	2	2	10	$24.90	Sticky Notes
8	5/7/2023	1	4	8	$55.92	Envelopes
9	5/8/2023	3	3	5	$99.95	Printer Ink
10	5/8/2023	3	1	12	$119.88	Copy Paper
11	5/9/2023	1	2	7	$17.43	Sticky Notes
12	5/10/2023	2	4	3	$20.97	Envelopes
13	5/10/2023	1	5	10	$49.90	Legal Pads
14	5/11/2023	2	99	4	$79.96	Other
15	5/12/2023	3	2	6	$14.94	Sticky Notes
16	5/12/2023	1	4	5	$34.95	Envelopes
17	5/13/2023	2	1	8	$79.92	Copy Paper
18	5/14/2023	3	5	15	$74.85	Legal Pads
19	5/15/2023	1	3	3	$59.97	Printer Ink
20	5/15/2023	2	4	10	$69.90	Envelopes

branch_name	branch_id
Scranton	1
Stamford	2
Nashua	3

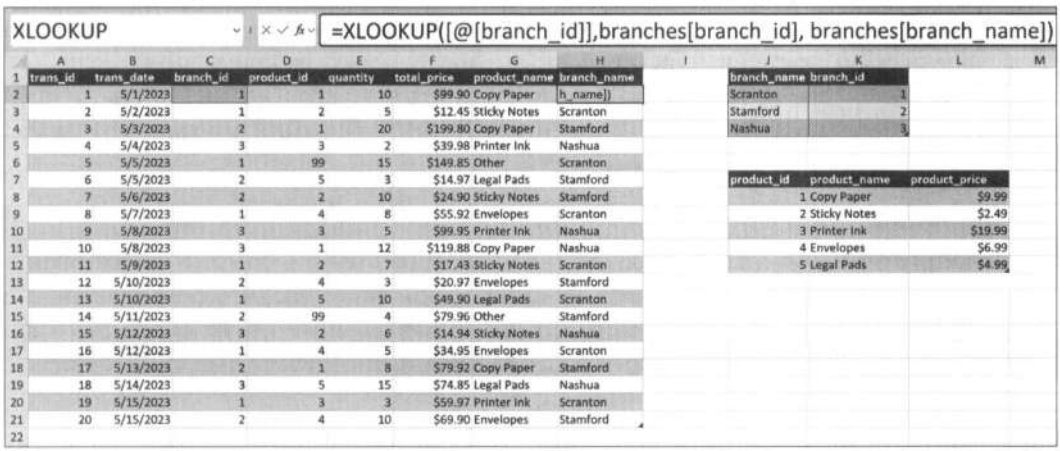
product_id	product_name	product_price
1	Copy Paper	$9.99
2	Sticky Notes	$2.49
3	Printer Ink	$19.99
4	Envelopes	$6.99
5	Legal Pads	$4.99

Рис. 10.15. Обработка ошибок с помощью функции `XLOOKUP()`

Функция *XLOOKUP()* и столбцы слева

Освоив выполнение поиска названий товаров по их идентификаторам (т. е. поиск значений, находящихся справа от просматриваемого столбца), выполним поиск названий товаров по названиям филиалов (branch), т. е. по столбцу, находящемуся слева от просматриваемого.

Достаточно часто `VLOOKUP()` критикуют за то, что эта функция не может возвращать данные слева от массива, по которому выполняется поиск, если не пользоваться вспомогательными функциями. В противоположность ей функция `XLOOKUP()` может возвращать значения из любого диапазона Excel, включая столбцы таблицы слева от просматриваемого столбца (рис. 10.16).



trans_id	trans_date	branch_id	product_id	quantity	total_price	product_name	branch_name
1	5/1/2023	1	1	10	$99.90	Copy Paper	h_name])
2	5/2/2023	1	2	5	$12.45	Sticky Notes	Scranton
3	5/3/2023	2	1	20	$199.80	Copy Paper	Stamford
4	5/4/2023	3	3	2	$39.98	Printer Ink	Nashua
5	5/5/2023	1	99	15	$149.85	Other	Scranton
6	5/5/2023	2	5	3	$14.97	Legal Pads	Stamford
7	5/6/2023	2	2	10	$24.90	Sticky Notes	Stamford
8	5/7/2023	1	4	8	$55.92	Envelopes	Scranton
9	5/8/2023	3	3	5	$99.95	Printer Ink	Nashua
10	5/8/2023	3	1	12	$119.88	Copy Paper	Nashua
11	5/9/2023	1	2	7	$17.43	Sticky Notes	Scranton
12	5/10/2023	2	4	3	$20.97	Envelopes	Stamford
13	5/10/2023	1	5	10	$49.90	Legal Pads	Scranton
14	5/11/2023	2	99	4	$79.96	Other	Stamford
15	5/12/2023	3	2	6	$14.94	Sticky Notes	Nashua
16	5/12/2023	1	4	5	$34.95	Envelopes	Scranton
17	5/13/2023	2	1	8	$79.92	Copy Paper	Stamford
18	5/14/2023	3	5	15	$74.85	Legal Pads	Nashua
19	5/15/2023	1	3	3	$59.97	Printer Ink	Scranton
20	5/15/2023	2	4	10	$69.90	Envelopes	Stamford

branch_name	branch_id
Scranton	1
Stamford	2
Nashua	3

product_id	product_name	product_price
1	Copy Paper	$9.99
2	Sticky Notes	$2.49
3	Printer Ink	$19.99
4	Envelopes	$6.99
5	Legal Pads	$4.99

Рис. 10.16. Функция `XLOOKUP()`, возвращающая столбец слева

Благодаря универсальности поиска по вертикали и горизонтали, возможности получать значения из столбцов по обе стороны от найденного значения и способности обработки ошибок функция `XLOOKUP()` стала очень популярной для поиска данных в Excel.

Другие функции динамического массива

Функции динамического массива, приведенные далее, были предложены одними из первых в Excel, а более поздние дополнения расширили их возможности.

Например, функция `RANDARRAY()` (`СЛУЧМАССИВ()`) создает массив случайных чисел с указанным количеством строк и столбцов. Это упрощает создание динамических массивов, заполненных случайными значениями, что идеально подходит для моделирования. Аналогично функция `SEQUENCE()` (`ПОСЛЕДОВ()`) создает массив с последовательностью чисел, используя заданное начальное число, шаг и размер массива. Это особенно удобно для генерации линейной последовательности или временно́го ряда в симуляциях и динамических моделях.

Многие функции динамического массива ориентированы на работу с текстом, в том числе VSTACK() (ВСТОЛБИК()) — для вертикального объединения массивов и TEXTSPLIT() (ТЕКСТРАЗД()) — для разделения текста по заданному разделителю. Чтобы ознакомиться с полным списком функций динамического массива и посмотреть дополнительный учебный материал, обратитесь к статье на **Exceljet.com**[4], посвященной этой теме.

Динамические массивы и современный Excel

Функции динамического массива могут показаться шагом назад в развитии Excel, учитывая появление таких инструментов, как Power Query и Power Pivot. Зачем возвращаться во времена чувствительных формульных рабочих книг, если есть новые расширенные возможности? Такая позиция не учитывает ценность динамических массивов для современного анализа в Excel. У функций динамического массива есть ряд своих важных преимуществ:

- *Простота.*

  Функции динамического массива упрощают работу с данными и их анализ, позволяя выполнять вычисления в рамках одной формулы, что повышает читаемость и удобство сопровождения. Это выгодно отличает их от сложного и поэтапного процесса, связанного с очисткой данных в Power Query или моделями данных в Power Pivot.

- *Легкодоступность.*

  Функции динамического массива отличаются от других инструментов, рассматриваемых в этой книге, своей интеграцией в привычную среду Excel. В отличие от надстроек, требующих установки или использования отдельных редакторов, функции динамического массива доступны в самом Excel. С ними можно сразу начинать работать, что значительно упрощает их освоение для обычного пользователя.

- *Автоматические обновления результатов.*

  Функции динамического массива автоматически обновляют результаты при изменении исходных данных. Это избавляет нас от необходимости вручную переписывать формулы или обновлять соединения, а также позволяет проводить анализ в режиме реального времени. Это преимущество особенно полезно при динамических сценариях, где данные постоянно меняются, — например, в информационных панелях в режиме реального времени или в финансовых моделях.

[4] См. **https://clck.ru/3KA2MQ**.

Заключение

В этой главе были представлены функции динамического массива, восстановившие репутацию традиционных и иногда громоздких ссылок и формул Excel. Теперь эти функции занимают важное место в аналитическом инструментарии Excel наряду с Power Query и Power Pivot.

Функции динамического массива — достаточно простые и нетрудоемкие, а в следующих главах *части III* мы рассмотрим более хитрые инструменты. Они требуют дополнительных настроек, но позволяют получить сложные аналитические данные, превосходящие те, которые можно получить с помощью одних лишь формул. В следующих главах вы узнаете, как в ваш рабочий процесс в Excel внедрить искусственный интеллект, методы машинного обучения и сложную автоматизацию с помощью языка Python.

Упражнения

Чтобы потестировать работу функций динамического массива, откройте файл ch_10_exercises.xlsx, расположенный в папке exercises\ch_10_exercises сопроводительного репозитория к этой книге[5]. Содержащаяся в нем рабочая книга включает два набора данных: `vehicles` и `common`. Выполните следующие упражнения:

1. Найдите отличающиеся (distinct) и действительно уникальные (unique) значения в столбце `make` набора данных `vehicles`. Сколько значений возвращается в каждом случае?
2. Выведите только автомобили с расходом топлива по городу `cty` более 30.
3. Выведите только те автомобили, у которых либо расход топлива по городу `cty` больше 30, либо оба цилиндра `cyl` меньше 6 и топливо `fuel` равно `Regular`.
4. Отсортируйте набор данных `vehicles` по расходу `hwy` в порядке убывания.
5. Выведите только столбец `model` набора данных `common`, отсортированный по столбцу `years` в порядке убывания.
6. К набору данных `vehicles` добавьте столбец `years` из набора данных `common`, используя сопоставление по столбцу `model`. Верните `Not available`, если значение не найдено.

Готовое решение можно посмотреть в файле ch_10_exercise_solutions.xlsx, расположенном в той же папке репозитория.

[5] См. **https://clck.ru/3KA4f6**.

Дополненная аналитика и будущее Excel

По мере того как мир аналитики становится все более сложным и широким, какую роль в нем будет играть Excel? Устареет ли он в экосистемах, основанных на искусственном интеллекте? В этой главе рассказывается о появлении дополненной аналитики, роли Excel в этом процессе, а также о некоторых современных вариантах ее применения.

Прежде чем погрузиться в захватывающий мир предсказательной аналитики, искусственного интеллекта (ИИ) и кардинальных изменений, происходящих в бизнесе, в том числе и с использованием наших электронных таблиц, важно осознать динамичность этой области. Постоянно появляется новое программное обеспечение. Даже такие хорошо зарекомендовавшие себя инструменты, как ChatGPT и Microsoft Copilot, часто претерпевают значительные обновления и изменения. Цель этой главы — сосредоточиться на понимании фундаментальных и более стабильных функциональностях Excel. Я не собираюсь делать полный обзор последних достижений, но хочу дать вам представление о том, что такое дополненная аналитика в Excel, и поделиться знаниями, необходимыми для работы в этой области, независимо от того, как будут развиваться его функциональность и инструментарий.

Растущая сложность данных и аналитики

В 2017 г. компания International Data Corporation (IDC), специализирующаяся на анализе рынка, предсказала[1], что в период с 2016 по 2025 г. объем существующих данных увеличится в десять раз и составит в общей сложности 163 зеттабайта, или триллион гигабайт.

С ростом общего объема данных растет и их разнообразие. По информации компании Taiger, разрабатывающей системы ИИ, к 2020 году 80% цифровых данных были неструктурированными[2], и эта цифра, скорее всего, выросла с появлением систем генеративной обработки естественного языка (Natural Language Processing, NLP), таких как ChatGPT. *Неструктурированные данные* — это информация, которая не имеет определенного формата или структуры, что затрудняет ее хранение и анализ в рамках традиционных баз данных или электронных таблиц, таких как

---

[1] См. **https://clck.ru/3KFmCn**.

[2] К сожалению, некоторые ссылки, присутствующие в исходном издании книги, не открываются или недоступны в настоящее время в России, поэтому далее такие ссылки приводиться не будут. — *Прим. ред.*

Excel. Примерами неструктурированных данных являются тексты, изображения, видео и сообщения в социальных сетях — к ним для извлечения полезной информации нужно применять более сложные методы обработки.

Кроме того, большое значение приобрели данные, поступающие в режиме реального времени. По оценкам компании IDC, к 2025 году потоковые данные будут составлять 30% от общего объема данных[3].

Такой взрыв количества данных, отразившийся, по мнению консалтинговой компании Gartner[4], на их объеме, скорости и разнообразии, потребовал использования новых аналитических методов. Наука о данных (Data science) помогает выявлять взаимосвязи и оценивать данные с помощью различных вычислений и статистических методов, а машинное обучение и искусственный интеллект предоставляют компьютерам возможность изучать и имитировать человеческий интеллект, а бизнес-компаниям — автоматизировать процессы принятия решений, выявлять тренды в режиме реального времени и создавать персонализированные сервисы.

И эти революционные тенденции никуда не денутся: в ходе опроса, проведенного компанией Deloitte, 94% руководителей ответили, что в ближайшие пять лет ИИ будет иметь решающее значение для успешного ведения бизнеса, а Бюро трудовой статистики США прогнозирует в течение следующего десятилетия увеличение числа специалистов по работе с данными на 36%, начиная со 113 000 человек в 2021 г.

Excel и self-service BI-системы

Self-service BI-системы (Business Intelligence)[5], создаваемые с помощью таких инструментов, как Excel, кардинально изменили процесс принятия решений в бизнес-компаниях. Однако возможности этих систем ограничены. Используемые в них данные должны быть структурированы таким образом, чтобы быть совместимыми с Excel, а это позволяет получать лишь описательную и диагностическую аналитику и означает, что Excel не приспособлен для работы со сложными алгоритмами и моделями машинного обучения, необходимыми для предсказательной или предписывающей аналитики.

Чтобы принимать более правильные стратегические решения, бизнес-компаниям необходимо дополнить свои self-service BI-системы более современными инструментами и методами аналитики, такими как интеллектуальный анализ данных (data mining), машинное обучение и искусственный интеллект.

---

[3] См. **https://clck.ru/3KFoZW**.

[4] См. **https://clck.ru/3KFpAe**.

[5] Self-service BI — это подход к анализу данных, ориентированный на конечного пользователя, который позволяет ему самостоятельно создавать аналитические отчеты, модели данных, визуализации и делиться ими с заинтересованными сторонами бизнес-процесса, не прибегая к помощи ИТ-специалистов. — *Прим. ред.*

Excel для дополненной аналитики

Дополненная аналитика (Augmented Analytics) — это подход, использующий технологии искусственного интеллекта и машинного обучения для улучшения процессов анализа данных. В рамках этого подхода с помощью просеивания больших массивов данных, выявления тенденций, закономерностей и аномалий автоматизируется сам процесс получения аналитической информации без необходимости ручного вмешательства. Это значительно повышает эффективность и точность анализа данных и работы систем self-service BI, позволяя бизнесу и частным лицам легко принимать обоснованные решения на основе полученной информации.

В следующих разделах этой главы вы познакомитесь с некоторыми практическими примерами использования дополненной аналитики в том виде, в котором она реализована в Excel на сегодняшний день. Вначале мы разберемся, как максимально эффективно использовать надстройку Analyze Data для получения аналитической информации на основе ИИ. Затем построим базовую предсказательную модель с помощью XLMiner. И в завершение используем оптическое распознавание символов и надстройку Azure Machine Learning для анализа настроений. Изучив все примеры этой главы, вы расширите свое представление о возможностях Excel и укрепите его перспективы в области дополненной аналитики.

Использование Analyze Data для получения результатов, сгенерированных ИИ

Analyze Data в Excel — это продукт дополненной аналитики, использующий ИИ для более эффективного получения значимых аналитических результатов. Тем не менее ИИ не может полностью заменить опытного специалиста. Чтобы по максимуму использовать потенциал Excel и Analyze Data для получения результатов, сгенерированных ИИ, данные должны быть правильно структурированы.

Чтобы работать с примерами этой главы, откройте из папки ch_11 сопроводительного репозитория к этой книге файл ch_11.xlsx[6].

Надстройка **Analyze Data** (Анализ данных) уже готова к использованию в Excel без необходимости ее загрузки[7]. Просто поместите курсор в таблицу `wholesale_customers`, расположенную на первом рабочем листе, и выберите на ленте **Home | Analyze Data** (Главная | Анализ данных)[8]. Вам сразу будет предложено множество интересных аналитических материалов, сгенерированных ИИ (рис. 11.1). Выберите любой из них, чтобы вставить его в вашу рабочую книгу.

---

[6] См. **https://clck.ru/3KFv6L**.

[7] Если в вашей версии Excel вы такой настройки не найдете, воспользуйтесь советами, приведенными, например, здесь: **https://clck.ru/3KFxLi**. — *Прим. ред.*

[8] В некоторых версиях Excel надстройка **Анализ данных** открывается с вкладки **Данные**. — *Прим. ред.*

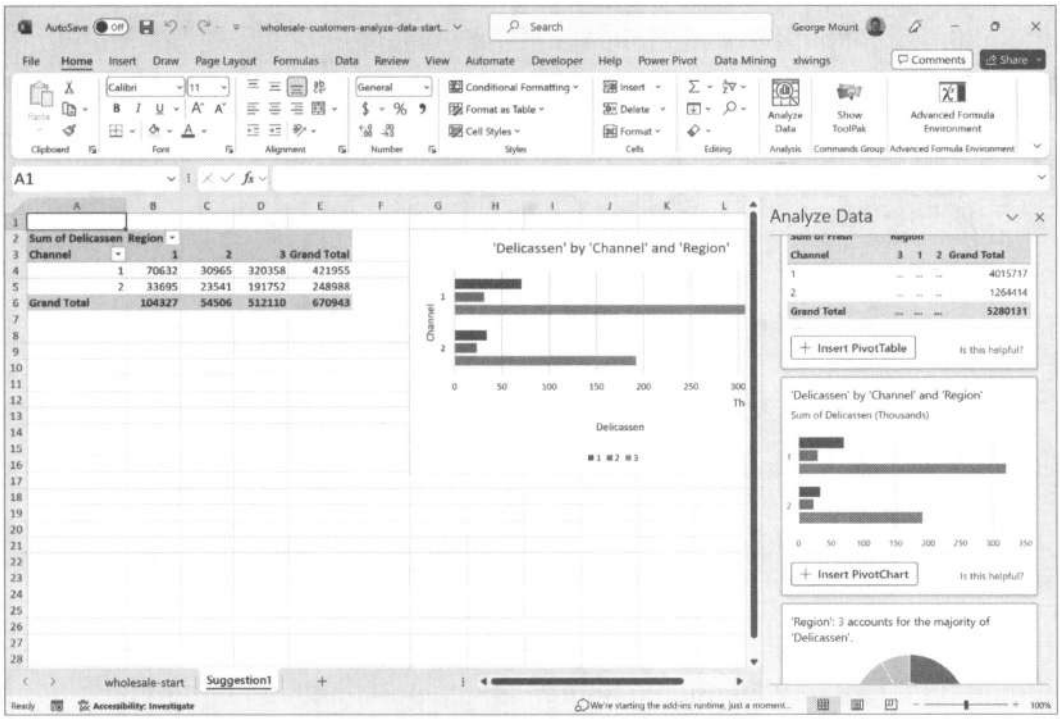


Рис. 11.1. Вставка результатов Analyze Data

Из-за вероятностного характера информации, генерируемой ИИ, в том числе и созданной с помощью Analyze Data, полученные вами результаты могут отличаться от моих. Это еще раз подчеркивает, что при использовании ИИ очень важно глубокое понимание предметной области и данных, поскольку вам нужно уметь правильно интерпретировать сложные и динамичные результаты и ориентироваться в них.

Возможности Analyze Data становятся еще более очевидными благодаря запросам на естественном языке. Представьте, например, что вы находитесь на совещании и вам нужно быстро получить данные об общем объеме продаж в продуктовом отделе. Вместо того чтобы тратить время на ручное вычисление, вы можете напрямую задать вопрос Analyze Data и сразу получить нужную информацию (рис. 11.2).

Несмотря на то что выполнение такого текстового запроса к набору данных, безусловно, впечатляет, у этой функциональности есть определенные ограничения, связанные в первую очередь со структурой данных. Например, если вы попытаетесь запросить в Analyze Data общие объемы продаж по регионам, то вместо этого получите сумму чисел в столбце `Region` (рис. 11.3).

Analyze Data не понимает, что нужно сделать, потому что данные представлены в неупорядоченном виде, — все продажи не хранятся в одном столбце, а распределены по нескольким столбцам в зависимости от отдела. В результате Analyze Data не может определить, в каком столбце находятся нужные числа для суммирования. Наглядно эта ошибка с упорядочиванием данных показана на рис. 11.4.

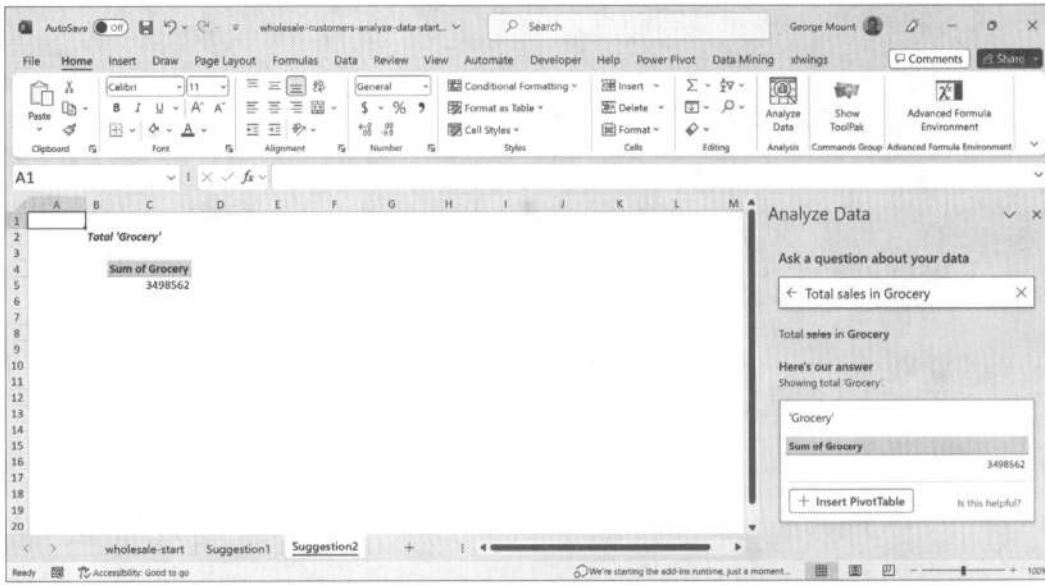


Рис. 11.2. Запрос на естественном языке в Analyze Data

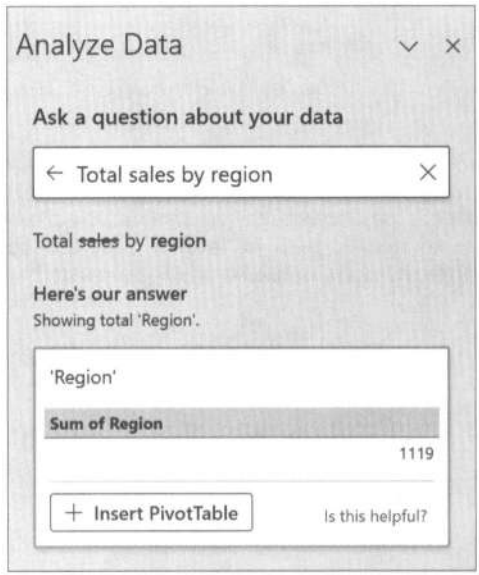
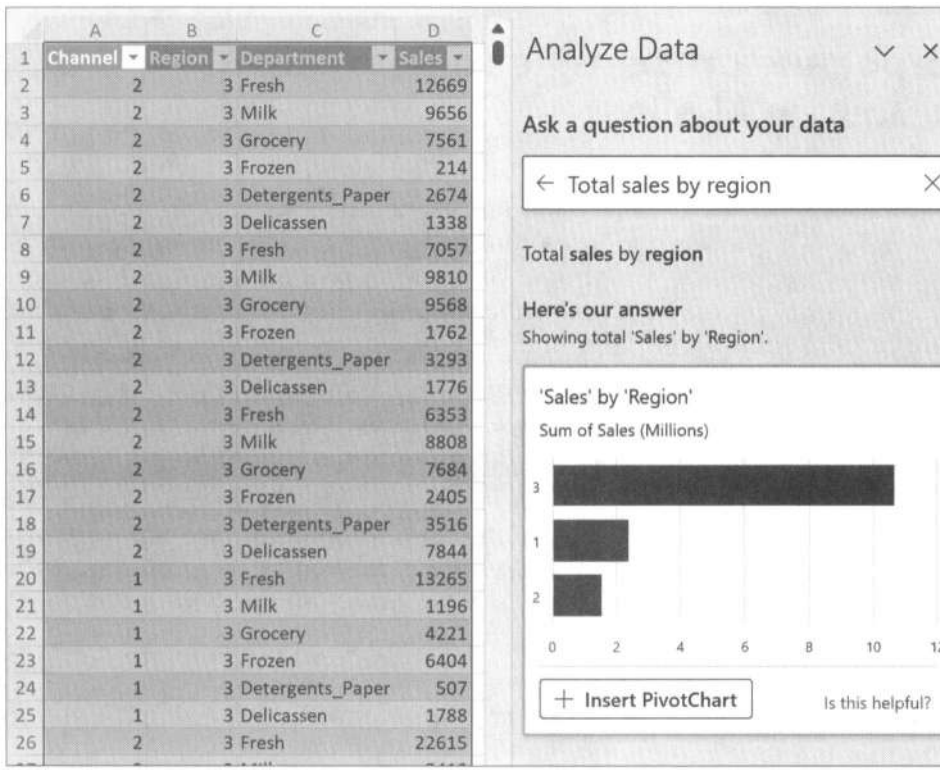


Рис. 11.3. Analyze Data пытается разобраться в неупорядоченных данных

Хранение данных в неупорядоченном или «грязном» виде может привести к серьезным нарушениям в аналитике. Вероятно, вы уже сталкивались с такой проблемой по своей работе, но не понимали, в чем именно ошибка. Понимание концепции «грязных» данных позволит вам выявлять проблемы на ранних этапах вашего проекта, что сэкономит вам много времени в дальнейшем. Если вы хотите углубиться в теорию упорядоченных данных и научиться эффективно с ними работать, вернитесь к *главе 1*.

	A	B	C	D	E	F	G	H
1	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
2	2	3	12669	9656	7561	214	2674	1338
3	2	3	7057	9810	9568	1762	3293	1776
4	2	3	6353	8808	7684	2405	3516	7844
5	1	3	13265	1196	4221	6404	507	1788
6	2	3	22615	5410	7198	3915	1777	5185
7	2	3	9413	8259	5126	666	1795	1451
8	2	3	12126	3199	6975	480	3140	545
9	2	3	7579	4956	9426	1669	3321	2566
10	1	3	5963	3648	6192	425	1716	750
11	2	3	6006	11093	18881	1159	7425	2098
12	2	3	3366	5403	12974	4400	5977	1744

Рис. 11.4. Нужно упорядочить этот набор данных для более качественного анализа



	A	B	C	D
1	Channel	Region	Department	Sales
2	2	3	Fresh	12669
3	2	3	Milk	9656
4	2	3	Grocery	7561
5	2	3	Frozen	214
6	2	3	Detergents_Paper	2674
7	2	3	Delicassen	1338
8	2	3	Fresh	7057
9	2	3	Milk	9810
10	2	3	Grocery	9568
11	2	3	Frozen	1762
12	2	3	Detergents_Paper	3293
13	2	3	Delicassen	1776
14	2	3	Fresh	6353
15	2	3	Milk	8808
16	2	3	Grocery	7684
17	2	3	Frozen	2405
18	2	3	Detergents_Paper	3516
19	2	3	Delicassen	7844
20	1	3	Fresh	13265
21	1	3	Milk	1196
22	1	3	Grocery	4221
23	1	3	Frozen	6404
24	1	3	Detergents_Paper	507
25	1	3	Delicassen	1788
26	2	3	Fresh	22615

Рис. 11.5. Получение общих объемов продаж по регионам с помощью Analyze Data

Чтобы ИИ мог в полной мере раскрыть свой потенциал, данные должны быть представлены в машиночитаемом, упорядоченном виде, когда каждая переменная хранится в своем отдельном столбце. Чтобы исправить эту проблему с нашим набором данных, давайте воспользуемся Power Query и развернем столбцы от `Fresh` до

Delicassen[9] в два столбца с именами Department и Sales. Загрузите результат запроса в таблицу Excel. Чтобы вспомнить, как в Power Query разворачивать и загружать набор данных, вернитесь к *главе 4*.

Когда данные представлены в упорядоченном виде, запрос на естественном языке для поиска общих объемов продаж по регионам выполнится без проблем (рис. 11.5).

Вы можете сами потренироваться и поискать новые удивительные результаты от Analyze Data, используя файл ch_11_solutions.xlsx с готовым решением, расположенный в той же папке репозитория.

Analyze Data — это мощный инструмент дополненной аналитики, который использует ИИ для получения ценных аналитических результатов. Но для достижения максимальных результатов очень важно иметь правильно упорядоченные данные. Только понимая концепцию упорядоченных данных и умея приводить данные к нужному виду, пользователи могут в полной мере использовать потенциал ИИ для получения значимых результатов.

Построение статистических моделей с помощью XLMiner

Надстройка XLMiner для Excel расширяет возможности аналитика, предлагая набор основных инструментов для анализа данных и моделирования. Она позволяет пользователям Excel пользоваться более сложной современной аналитикой, расширяя возможности дополненной аналитики. Набор данных для следующего примера находится в файле ch_11.xlsx на рабочем листе housing.

Чтобы начать работу, выберите на ленте опцию **File | Setting | Add-ins | Get Add-ins** (Файл | Параметры | Надстройки | Получить надстройки). В диалоговом окне надстроек Office (рис. 11.6) найдите XLMiner и нажмите кнопку **Add** (Добавить).

Согласитесь с условиями лицензии и политикой конфиденциальности, нажмите кнопку **OK**, и в правой части вашего рабочего листа откроется панель с инструментами XLMiner. Как вы можете заметить, в XLMiner представлено множество статистических инструментов и методов. Давайте остановимся на «матери всех моделей» — линейной регрессии.

В качестве зависимой (целевой) переменной возьмем столбец price, а в качестве независимых переменных: lotsize, airco и prefarea. На панели XLMiner перейдите на вкладку **Linear Regression**, заполните ее так, как показано на рис. 11.7, и нажмите кнопку **OK**.

Использование курсора мыши для задания входных диапазонов в XLMiner может вызвать некоторые сложности, поэтому проще ввести адреса ячеек вручную. Но прежде чем приступить к построению модели и прогнозированию, необходимо тщательно изучить набор данных, чтобы убедиться, что он соответствует допуще-

[9] Названия столбцов оставлены такими же, как в исходном наборе данных: **https://clck.ru/3KG5ku**.

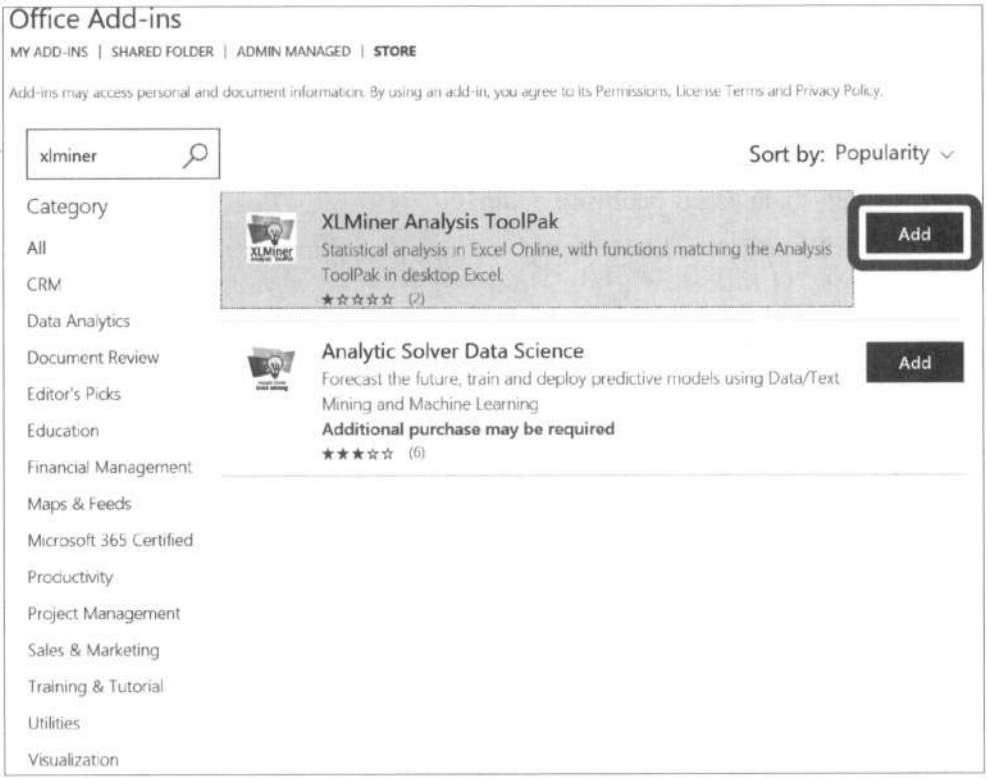


Рис. 11.6. Добавление надстройки XLMiner

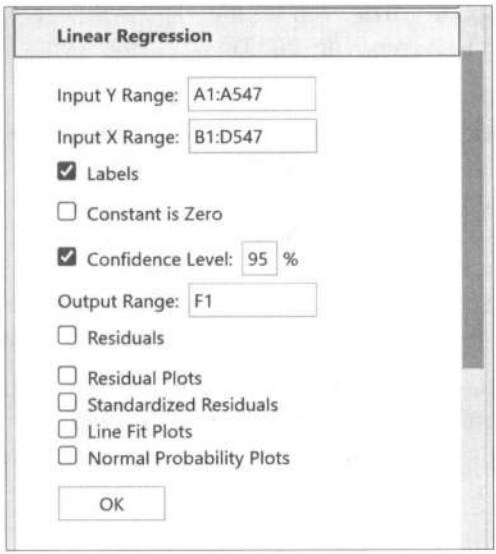


Рис. 11.7. Настройка линейной регрессии в XLMiner

ниям, принятым в выбранной модели. Хотя Python и R предлагают более широкий инструментарий для анализа и тестирования, наглядность взаимодействия с данными в Excel может оказаться очень полезной. XLMiner находится примерно посередине, объединяя в себе простоту Excel при работе с данными и фундаментальные основы сложного анализа, которые обычно встречаются только в узкоспециализированных приложениях для работы с данными.

После запуска регрессии в XLMiner вы должны увидеть результат вычислений, показанный на рис. 11.8.

	E	F	G	H	I	J	K	L	M	N
6		Adjusted R Square	0.436467707							
7		Standard Error	20045.37142							
8		Observations	546							
9										
10		ANOVA								
11			df	SS	MS	F	Significance F			
12		Regression	3	1.70818E+11	56939339259	141.7046846	0			
13		Residual	542	2.17785E+11	401816915.2					
14		Total	545	3.88603E+11						
15										
16			Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95%	Upper 95%
17		Intercept	32770.49675	2216.879895	14.78226079	8.65278E-42	28415.76783	37125.22568	28415.76783	37125.22568
18		lotsize	5.116814473	0.415952537	12.3014383	7.6259E-31	4.299737937	5.933891009	4.299737937	5.933891009
19		airco	19437.2651	1895.231911	10.25587686	1.12388E-22	15714.36553	23160.16468	15714.36553	23160.16468
20		prefarea	12112.04184	2087.892931	5.80108379	1.12101E-08	8010.688509	16213.39516	8010.688509	16213.39516

Рис. 11.8. Результаты линейной регрессии в XLMiner

Здесь приведены стандартные параметры регрессии, такие как *p*-коэффициент, *R*-квадрат и многое другое. Если вы хотите больше узнать о том, как интерпретировать эти результаты, прочитайте мою книгу «Погружение в аналитику данных: от Excel к Python и R» (БХВ-Петербург, 2023)[10].

XLMiner расширяет возможности Excel по анализу данных, предоставляя простые средства для выполнения статистического моделирования пользователям с разным уровнем подготовки. Несмотря на удобный интерфейс и полную интеграцию с Excel, XLMiner нельзя считать универсальным инструментом дополненной аналитики. К недостаткам этой надстройки можно отнести, прежде всего, неспособность использования моделей в режиме потоковых данных, отсутствие непрерывного обучения для адаптации моделей со временем, а также недостаточную поддержку современных методов моделирования, таких как нейронные сети.

Более того, ограниченная интеграция ИИ в XLMiner сокращает его возможности по комплексной автоматизации процессов анализа данных. Для решения более сложных аналитических задач пользователям, вероятно, придется использовать универсальные инструменты, доступные в экосистемах R и Python.

[10] См. **https://clck.ru/3KGAa4**.

Чтение данных с изображения

Иногда аналитики сталкиваются с ситуациями, когда данные изначально доступны только в бумажном виде или в другом аналоговом формате. Чтобы избежать медленного и ненадежного процесса ручного ввода данных, в Excel предусмотрена функциональность, позволяющая преобразовывать текст на изображении в рабочую книгу.

Преобразование отсканированных бумажных документов в редактируемые текстовые файлы используется давно и известно как *оптическое распознавание символов* (Optical Character Recognition, OCR). Технология OCR существует с 1970-х гг. и с тех пор претерпела значительные изменения. Сегодня она доступна в различных приложениях, в том числе и в Excel.

Для примера давайте возьмем отзывы покупателей, которые есть только в печатном виде. Наша задача — импортировать их в Excel для анализа настроений. Содержащий эти отзывы файл scanned_reviews.png также находится в папке ch_11 сопроводительного репозитория к этой книге.

Чтобы начать работу, создайте новую рабочую книгу Excel и выберите на ленте опцию **Data | Get & Transform Data | From Picture | Picture from File** (Данные | Получить и преобразовать данные | ...), найдите упомянутый ранее файл с изображением scanned_reviews.png и выберите его. В правой части рабочей книги появится панель **Data from Picture** (рис. 11.9).

Опция OCR в Excel преобразует изображение в текст — казалось бы, замечательная функциональность, но, к сожалению, она допускает ошибки. С помощью ИИ Excel может предположить, где могли возникнуть ошибки распознавания.

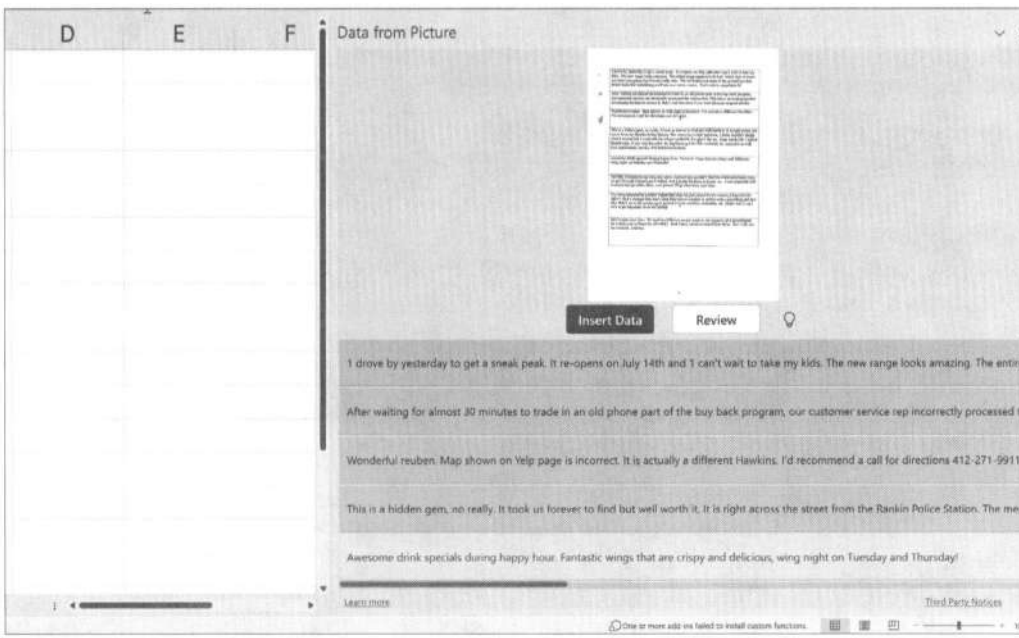


Рис. 11.9. Панель Data from Picture

В нашем случае Excel выделил красным цветом все записи как вероятно содержащие ошибку, кроме одной. Вы можете нажать кнопку **Review**, чтобы пройтись по всем таким записям, проверить каждую из них и внести какие-либо корректировки в данные. Например, первая запись начинается с числа 1, вместо которой должно быть английское местоимение I (рис. 11.10).

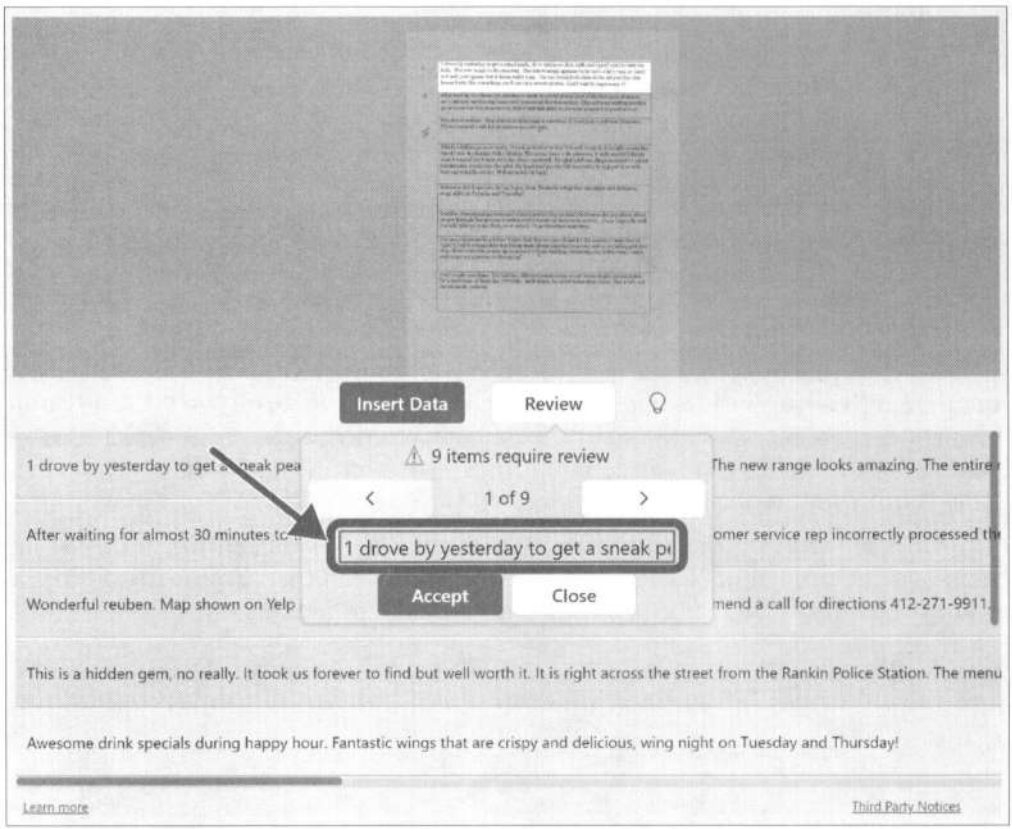


Рис. 11.10. Ошибка распознавания текста, обнаруженная ИИ

Проверив записи и исправив ошибки, нажмите кнопку **Insert Data**, чтобы вставить результаты в Excel.

ИИ в Excel неплохо справляется с предсказанием возможных ошибок в тексте, но и он несовершенен. Например, он может обнаружить ошибку в записи, когда ее нет — в статистике это называется *ложноположительный результат (или ложное срабатывание)*. Или, наоборот, он может одобрить запись, в которой есть явная ошибка, — это уже *ложноотрицательный результат*.

Поиск баланса при выявлении возможных ложноположительных и ложноотрицательных результатов представляет собой серьезную проблему в статистике и в машинном обучении. На начальном этапе вы можете полагаться на оценку Excel, но со временем в своей работе аналитиком вы можете столкнуться с ситуациями, когда надежнее будет принимать решения самостоятельно.

Вставка неструктурированных данных — например, текста, в Excel может вызвать некоторые трудности, поскольку Excel изначально не предназначен для работы с такими данными. Чтобы придерживаться структуры Excel, рекомендуется вставить каждый отзыв покупателя в отдельную ячейку и вручную внести нужные исправления. Так, строки 6 и 7 можно объединить в один отзыв.

Несмотря на то что технология распознавания текста существует уже давно, ее интеграция в Excel оказалась необыкновенно удобной. Эта функциональность очень помогает при работе с финансовыми отчетами или аналогичными цифровыми документами, которые могут понадобиться для анализа в Excel.

Результаты этого примера пригодятся нам в следующем разделе.

Анализ настроений с помощью Azure Machine Learning

Несмотря на то что Excel традиционно считается инструментом для работы лишь с небольшими структурированными наборами данных, появление функциональностей, связанных с ИИ и машинным обучением, снимает эти привычные ограничения. Это еще раз подчеркивает огромный потенциал дополненной аналитики в Excel. Ярким примером того является возможность использования Excel для *анализа настроений* (Sentiment Analysis), позволяющего оценить эмоциональную окраску в текстовых отзывах.

Анализ настроений — это инструмент анализа данных, который использует алгоритмы машинного обучения для определения эмоций и мнений в неструктурированных данных. При этом, как правило, текст категоризируется как положительный, отрицательный или нейтральный, что позволяет бизнес-компаниям улучшать работу с потребителями и решать проблемы, опираясь на общее отношение к бренду, продукту или услуге.

Самим просмотреть несколько отзывов — не проблема, но это становится проблематичным, если их около тысячи и более. Давайте продолжим анализировать отзывы, полученные из изображения в предыдущем разделе. Наша текущая задача — категоризировать настроение каждого отзыва как положительное, отрицательное или нейтральное. Чтобы автоматизировать эту задачу, воспользуемся возможностями текстового анализа от Azure.

Сначала нам необходимо установить надстройку Azure Machine Learning в Excel. Для этого на ленте выберите **File | Setting | Add-ins | Get Add-ins** (Файл | Параметры | Надстройки | Получить надстройки). В диалоговом окне **Office Add-ins** (Надстройки Office) найдите **Azure Machine Learning**, нажмите кнопку **Add** (Добавить) и **Continue** (Продолжить). После завершения установки панель **Azure Machine Learning** должна появиться в правой части окна Excel. Нам нужен второй пункт: **Text Sentiment Analysis (Excel Add-in Solver)**, как показано на рис. 11.11.

Azure требует, чтобы входные данные для анализа настроений соответствовали определенному формату или *схеме*. В следующем примере будет показано, что

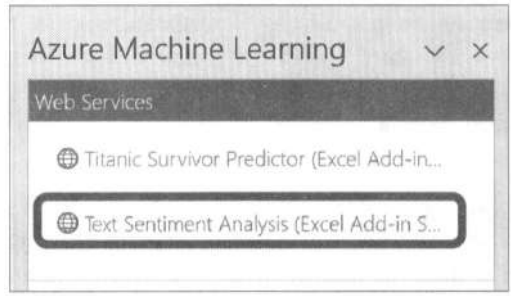


Рис. 11.11. Выбор анализа настроений в Azure Machine Learning

структурирование данных в требуемый формат имеет решающее значение для эффективной работы ИИ.

Для нашей задачи нам нужно создать три столбца в рабочей книге: `tweet_text`, `Sentiment` и `Score`. Заголовки этих столбцов должны точно соответствовать тем, которые указаны в разделе **View Schema** надстройки Azure Machine Learning (рис. 11.12).

В первый столбец: `tweet_text` — мы должны поместить отзывы о ресторане, которые были импортированы на предыдущем шаге. Несмотря на заголовок столбца, это могут быть любые текстовые отзывы, а не только сообщения из социальных сетей. Столбцы `Sentiment` (настроение) и `Score` (оценка) нужны надстройке Azure, чтобы записать результаты анализа настроений.

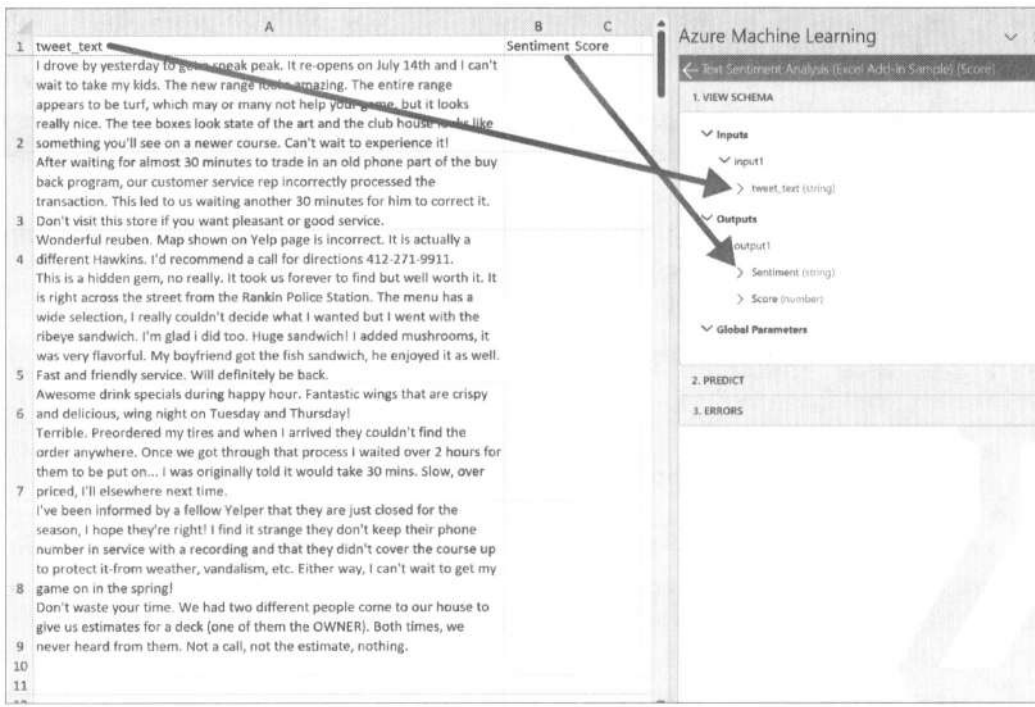


Рис. 11.12. Соответствие схеме для анализа настроений

Чтобы настроить входные данные для анализа настроений, перейдите в раздел **Predict** и в поле **Input** укажите диапазон для входных данных. Он должен охватывать ячейки `A1:A9`, включая заголовок. Обязательно установите флажок **My data has headers** (В моих данных есть строка заголовков).

Чтобы результаты анализа настроений выводились в ячейки, начиная с `B1`, укажите эту ячейку в поле **Output**. Убедившись, что ваши параметры для анализа настроений соответствуют показанным рис. 11.13, нажмите кнопку **Predict**, чтобы сгенерировать анализ настроений.

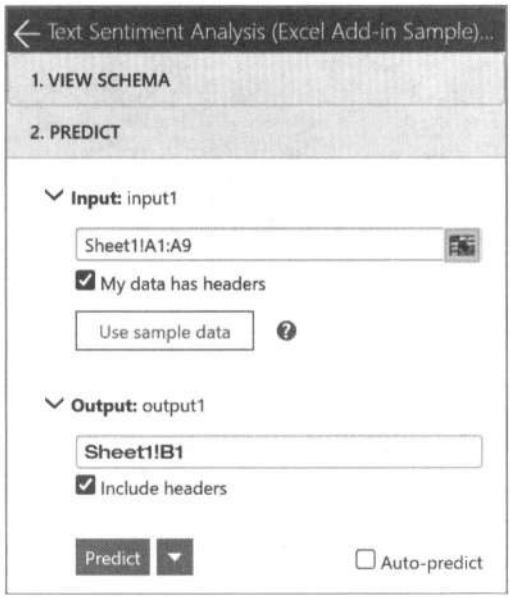


Рис. 11.13. Определение входных и выходных данных для анализа настроения

После нажатия кнопки **Predict** в столбцах `B` и `C` должны появиться результаты анализа. К сожалению, иногда этот процесс может давать сбои. Если у вас возникли проблемы, проверьте еще раз схему и входные данные или перезапустить Excel.

Как и ожидалось, Azure категоризировал каждый отзыв и записал результаты в столбец `Sentiment`. Столбец `Score` содержит числовые значения от 0 до 1, представляющие собой оценку настроения, рассчитанную Azure. Чем выше оценка, тем более положительный отзыв. Позднее по этим оценкам можно отфильтровать строки с отрицательным, нейтральным и положительным настроением.

Пояснить, как вычисляются полученные оценки, сложно — производится это с помощью специальной модели машинного обучения, и нюансы знает только Azure. Такие автоматизированные сервисы удобны в использовании, но им часто не хватает прозрачности и информации о том, как они работают. Несмотря на их бесспорную эффективность, эти инструменты не очень надежные. Например, в результатах анализа настроений, показанных на рис. 11.14, строки 3 и 5 (см. перевод под рисунком) отмечены как нейтральная и отрицательная, несмотря на то, что они содержат отрицательный и положительный отзывы соответственно.

	A	B	C
1	tweet_text	Sentiment	Score
2	I drove by yesterday to get a sneak peak. It re-opens on July 14th and I can't wait to take my kids. The new range looks amazing. The entire range appears to be turf, which may or many not help your game, but it looks really nice. The tee boxes look state of the art and the club house looks like something you'll see on a newer course. Can't wait to experience it!	positive	9.05E-01
3	After waiting for almost 30 minutes to trade in an old phone part of the buy back program, our customer service rep incorrectly processed the transaction. This led to us waiting another 30 minutes for him to correct it. Don't visit this store if you want pleasant or good service.	neutral	0.57314
4	Wonderful reuben. Map shown on Yelp page is incorrect. It is actually a different Hawkins. I'd recommend a call for directions 412-271-9911.	positive	8.55E-01
5	This is a hidden gem, no really. It took us forever to find but well worth it. It is right across the street from the Rankin Police Station. The menu has a wide selection, I really couldn't decide what I wanted but I went with the ribeye sandwich. I'm glad i did too. Huge sandwich! I added mushrooms, it was very flavorful. My boyfriend got the fish sandwich, he enjoyed it as well. Fast and friendly service. Will definitely be back.	negative	2.97E-02
6	Awesome drink specials during happy hour. Fantastic wings that are crispy and delicious, wing night on Tuesday and Thursday!	positive	0.953895

Рис. 11.14. Неправильно оцененные отзывы при анализе настроений.
СТРОКА 3: *После почти 30-минутного ожидания обмена старого телефона в рамках программы trade in наш представитель службы поддержки клиентов неправильно обработал транзакцию. Это привело к тому, что нам пришлось ждать еще 30 минут, пока он исправит ошибку. Не посещайте этот магазин, если вы хотите приятного или хорошего обслуживания.*
СТРОКА 5: *Это скрытая жемчужина, нет, правда. Нам потребовалась целая вечность, чтобы найти его, но оно того стоило. Оно находится прямо через дорогу от полицейского участка Ранкин. В меню большой выбор, я правда не могла решить, что хочу, но выбрала сэндвич с рибаем. Я рада, что так сделала. Огромный сэндвич! Я добавила грибы, получилось очень вкусно. Мой парень взял сэндвич с рыбой, ему тоже понравилось. Быстрое и дружелюбное обслуживание. Обязательно вернусь.*

Мораль этой истории заключается в том, что нужно, конечно же, пользоваться огромным потенциалом ИИ, но включать при этом критическое мышление и не полагаться только на ИИ. Само собой, что у ИИ есть определенные интеллектуальные навыки, но вы обладаете человеческой рассудительностью и интуицией. Объединив сильные стороны обоих интеллектов, вы можете принимать более взвешенные решения.

Всем известно, что очень сложно обрабатывать неструктурированные данные, но ИИ хорошо помогает при работе с ними. Несмотря на то что Excel в первую очередь ориентирован на структурированные данные, наблюдается растущая тенденция использовать его для работы с неструктурированными данными, включая тексты и изображения. Но точно так же, как Analyze Data оптимальнее всего работает с определенной структурой данных, надстройка Azure для анализа настроений зависит от точности схемы, позволяющей эффективно интерпретировать неструктурированные входные данные.

Анализ настроений — это всего лишь начало. Готовящаяся в Excel интеграция языковой модели на основе GPT, такой как Copilot, представляет собой значительный шаг вперед в этом направлении. Эта интеграция значительно расширит функцио-

нальность Excel и позволить пользователям задействовать огромные возможности языковых моделей в своих проектах.

Заключение

В завершение следует отметить, что предсказательная аналитика и ИИ — это мощные инструменты, которые помогут вам лучше изучить данные и сделать прогнозы на будущее. Современный Excel сильно эволюционировал и умеет подключать эти инструменты, расширяя возможности пользователей при решении различных задач: от распознавания изображений до создания отчетов и анализов. Использование надстройки Analyze Data для получения аналитических результатов на основе ИИ, создание предсказательных моделей с помощью XLMiner и интеграция Excel с Azure Machine Learning позволят вам раскрыть весь потенциал предсказательной аналитики и ИИ в Excel.

Упражнения

Чтобы потренироваться в использовании возможностей дополненной аналитики и искусственного интеллекта в Excel, откройте файл ch_11_exercises.xlsx, расположенный в папке exercises\ch_11_exercises сопроводительного репозитория к этой книге[1], и выполните следующие упражнения:

1. С помощью надстройки Azure Machine Learning произведите анализ настроений на основе набора данных с рецензиями на фильмы, расположенного на рабочем листе `imdb`. После этого используйте надстройку XLMiner для создания описательной статистики (вкладка **Descriptive Statistics**) полученных оценок.
2. Импортируйте в Excel данные с изображения, содержащегося в файле life_expectancy.png. С помощью опции **Analyze Data** постройте линейный график зависимости средней продолжительности жизни от времени. Для этого вам сначала нужно будет преобразовать данные к нужному формату.

Готовые решения можно посмотреть в файле ch_11_exercise_solutions.xlsx, расположенном в той же папке репозитория.

[1] См. https://clck.ru/3KLNSz.

ГЛАВА 12
Python и Excel

До сих пор основное внимание в этой книге уделялось инструментам, созданным специально для экосистемы Microsoft, таким как Power Pivot и Power Query. Но в завершение мы рассмотрим ставший в последнее время очень популярным важнейший язык программирования, совместимый практически со всеми возможными программами, в том числе и с Excel. Добро пожаловать на ознакомительный урок о том, как Python может расширить ваши возможности при работе с Excel.

Эта глава специально помещена в конец книги, поскольку я понимаю, какие опасения она может вызвать у обычных пользователей Excel. Тем не менее, если вы дочитали до этой главы и заинтересованы продолжить обучение в области современной аналитики, я настоятельно рекомендую вам попробовать освоить Python.

И это не только мое личное мнение. Компания Microsoft одобрила взаимодействие Python и Excel, официально интегрировав язык Python внутрь Excel, что значительно расширяет возможности аналитиков при совместном использовании этих двух мощнейших инструментов.

Но пока что Python встроен в Excel лишь в виде новой специфичной функциональности, и она не отражает полностью все те возможности, которые язык Python может предложить пользователям Excel. В этой главе мы рассмотрим более широкое концептуальное взаимодействие этих двух инструментов и приведем несколько примеров. Если эта глава окажется для вас познавательной, я рекомендую заняться также изучением интеграции Python в Excel.

Примеры, приведенные в этой главе, не используют встроенную интеграцию Python в Excel. Вместо этого в ней приведены альтернативные способы соединения этих инструментов для реализации более сложных процессов, чем позволяют возможности встроенной интеграции Python в Excel.

Предварительные требования

Хотя эту главу можно читать и без знания Python, предварительное знакомство с такими понятиями, как списки, индексация, циклы, а также с пакетами pandas и seaborn значительно ускорит освоение материала.

Если вы хотите познакомиться с Python перед чтением этой главы, рекомендую начать с моей книги «Погружение в аналитику данных: от Excel к Python и R» (БХВ-Петербург, 2023)[1], в которой дано введение в Python для пользователей Excel. Что-

[1] См. **https://clck.ru/3KLRyH**.

бы еще глубже погрузиться в тему, почитайте книгу Феликса Зумштейна «Python для Excel: Современная среда для автоматизации и анализа данных» (БХВ-Петербург, 2024)[2].

Эта глава, прежде всего, должна показать на практике, как программировать на Python. Чтобы извлечь из примеров максимальную пользу, я рекомендую вам активно выполнять все примеры, используя свой собственный компьютер. Все, что вам при этом требуется, — совершенно бесплатная версия Python, которую можно установить, например, с его официальной страницы[3].

Роль Python в современном Excel

Новичков часто пугает упоминание Python рядом с Excel. Многие пользователи Excel считают, что этот инструмент должен быть последним в списке для изучения, поскольку он не является продуктом Microsoft и требует освоения нового языка программирования.

Скорее всего, Python пригодится не всем пользователям Excel, но к нему стоит серьезно присмотреться тем, кто хочет создавать сложные автоматизированные системы, проекты с системами контроля версий и другие сложные программные продукты. Давайте разберемся с ролью Python в современной аналитике и в его взаимосвязи с современным Excel.

«Клей» для огромного набора инструментов

Когда я только начинал работать аналитиком, мой набор инструментов состоял из одного Excel. Управление данными, отчетность, информационные панели — все это размещалось в его привычном бело-зеленом интерфейсе.

Спустя несколько лет картина кардинально изменилась: появились Power BI, сценарии Office, Jupyter Notebook и даже интеграция Python в Excel. Эта экспансия в Excel внешнего инструментария стала следствием более масштабного технологического сдвига — перехода от одного универсального приложения к распределенной сети специализированных и взаимосвязанных инструментов.

Для управления этой разнообразной экосистемой необходимо что-то вроде «проводника» или «клеевого пистолета», способствующего легкому подключению к базовому инструменту различных компонентов. Неважно, что именно вам может понадобиться: передача данных между платформами, новая визуализация или развертывание облачной модели машинного обучения на информационной панели — Python подойдет идеально. Его универсальность помогает справляться практически с любыми задачами — от создания простых скриптов до разработки сложных корпоративных решений. Кроме того, он совместим с различными операционными системами и языками программирования.

---

[2] См. **https://clck.ru/3KLS9W**.

[3] См. **https://clck.ru/3KLSJk**.

Компания Microsoft высоко оценила роль языка Python как универсального «клея» и включила его использование в Azure, Power BI, SQL Server и др. Популярность Python среди разработчиков и организаций привела к появлению огромного сообщества его почитателей и изобилию разных ресурсов.

Сетевой эффект сокращает время разработки

Как правило, аргумент «потому что все так делают» не является веской причиной для того, чтобы начать заниматься чем-то, но в случае с языками программирования это может быть оправдано.

Сетевой эффект — концепция, согласно которой ценность чего-либо увеличивается с ростом числа пользователей. Она применима и к языкам программирования. По мере присоединения к сообществу всё большего числа программистов растет объем совместно используемого кода, что приводит к появлению большой кодовой базы для дальнейшей разработки и развития, и это все эффективно развивается в цикле.

Универсальность Python как нейтрального языка-«клея» привела к тому, что его стали использовать в различных сферах, включая управление базами данных, веб-разработку и анализ. Это также означает, что, независимо от направления вашего проекта в Excel или используемых инструментов, вероятно, у вас есть коллеги, которые уже «говорят» на Python.

Представьте, например, что вы начали разрабатывать систему инвентаризации или похожий инструмент с помощью Excel, но вскоре обнаруживаете, что эта система стала слишком сложной для простой рабочей книги или оказалась настолько популярной, что возникла необходимость превратить ее в отдельное веб-приложение. Такое преобразование, как правило, представляет собой серьезную проблему. Но если бы изначально программирование выполнялось на языке Python, этот переход был бы значительно более быстрым и эффективным.

Универсальность Python и его широкая поддержка в сообществе веб-разработчиков способствуют более легкой интеграции языка с различными веб-технологиями и платформами. Следовательно, время, необходимое для преобразования вашего решения на основе Excel в полноценное веб-приложение, может значительно сократиться. По сути, начав писать на Python, вы получаете значительное преимущество и закладываете прочную основу для будущего развития или адаптации вашего проекта по мере его роста.

Добавьте современные технологии к Excel

Python позволяет специалистам, работающим с современным Excel, внедрить в свои проекты лучшие практики из разработки программного обеспечения, в том числе модульное тестирование, контроль версий и разработку пакетов.

Модульное тестирование

Модульное тестирование (Unit Testing) включает в себя тестирование отдельных компонентов или модулей программного обеспечения, позволяющее убедиться, что

каждый из них правильно работает сам по себе. Такое тестирование помогает разработчикам выявлять и исправлять ошибки на ранних стадиях процесса разработки, обеспечивая надежность и результативность конечного продукта.

Во многих языках программирования реализована функциональность для выполнения модульного тестирования, дающая возможность гарантировать, что код работает так, как надо. Однако в Excel изначально такой возможности нет. И хотя существуют разные альтернативные инструменты для выполнения модульного тестирования, Python считается отличным выбором благодаря своему масштабному сетевому эффекту и огромному многообразию существующих пакетов. Автоматизация модульного тестирования повышает надежность проекта и снижает вероятность ошибок, что особенно полезно для рабочих книг Excel, с которыми работают люди с разным уровнем технической подготовки.

Системы контроля версий

Система контроля версий отслеживает изменения в репозитории, позволяя пользователям просматривать последние изменения, возвращаться к предыдущим версиям и т. д. Если вы когда-нибудь сталкивались с трудностями при поиске различий между несколькими рабочими книгами — такими как, например, budget-model-final.xlsx и budget-modelFINAL-final.xlsx, вы сможете оценить важность и полезность контроля версий.

Хотя в Excel есть ограниченные возможности для контроля версий, такие как просмотр истории в OneDrive или надстройка Inquire для сопоставления рабочих книг, это не идет ни в какое сравнение с широкими возможностями, доступными при переносе разработки на Python.

Разработка пакетов и их распространение

Если вам нужна конкретная причина, поясняющая, зачем использовать Python для решения своих повседневных аналитических задач, могу выделить его главное преимущество — *пакеты*.

Несмотря на то что мне, безусловно, нравится разрабатывать собственные инструменты под свои нужды, я также использую и функциональность уже существующих решений, когда они отвечают моим требованиям. Широкие возможности Python по созданию и распространению пакетов, особенно через каталог Python Package Index (PyPI), привели к появлению целой Вселенной разнообразных инструментов, с которой не могут сравниться ни надстройки Excel, ни модули VBA. Почти все эти инструменты имеют открытый исходный код и находятся в свободном доступе.

Независимо от того, что является вашей целью: загрузка данных по API, анализ изображений или просто получение описательной статистики, — доступность огромного количества пакетов Python является убедительным аргумент в пользу того, чтобы потратить время на изучение Python.

И особенно потому, что некоторые из этих пакетов разработаны для плавной интеграции с Excel.

Совмещение Python и Excel с помощью пакетов *pandas* и *openpyxl*

Учитывая роли Python и современного Excel, давайте рассмотрим, как они могут работать вместе. Вот два ключевых пакета Python, помогающих реализовывать эту интеграцию: `pandas` и `openpyxl`.

Давайте поближе познакомимся с обоими.

Зачем нужен *pandas* для работы с Excel?

Если вы работаете с любыми табличными данными в Python, вам не обойтись без `pandas`. Этот пакет позволяет выполнять в том числе и следующие операции:

- сортировать и фильтровать строки;
- добавлять, удалять и преобразовать столбцы;
- агрегировать и реструктурировать таблицы;
- объединять или соединять несколько таблиц.

Это аналог Power Query, но на языке Python, позволяющий создавать процессы для очистки и преобразования данных, которые можно многократно запускать. Как и Power Query, с помощью `pandas` можно легко импортировать данные из различных источников, в том числе и из Excel, после чего экспортировать результаты анализа, например, обратно в Excel.

Ограничения при работе с *pandas*

Тем не менее возможности `pandas` при взаимодействии с рабочими книгами Excel ограничены. Например, этот пакет не может помочь в решении следующих задач:

- расширенные возможности форматирования ячеек — например, применение стилей или условное форматирование;
- выполнение макросов Excel или кода VBA в рабочих книгах;
- непосредственный доступ к специфичным для Excel функциональностям — таким как проверка данных, диаграммы, сводные таблицы и формулы;
- управление рабочими листами — например, изменение или удаление данных.

Но, к счастью, есть несколько других пакетов, позволяющих выполнять такие сложные действия в Python/Excel, и самый популярный из них — `openpyxl`.

Что умеет openpyxl?

Пакет `openpyxl` (произносится как «open pie Excel») предоставляет функциональность для работы с файлами Excel, в частности с файлами формата *.xlsx. Он позволяет пользователям программно читать, записывать и изменять файлы Excel. Пакет `openpyxl` легко интегрируется с `pandas`, поэтому пользователи могут очистить данные с помощью `pandas` и добавить дополнительную функциональность в рабочую книгу с помощью `openpyxl`.

И хотя openpyxl также имеет свои ограничения и не охватывает все возможные случаи использования Excel, он всё еще остается лучшим пакетом Python, с которого стоит начинать автоматизацию задач Excel.

Использование *openpyxl* вместе с *pandas*

Давайте рассмотрим стандартный пример автоматизации обычного отчета для бизнеса, когда аналитику нужно формировать ежемесячные отчеты о продажах на основе нескольких рабочих листов Excel. Для решения этой и аналогичных задач с помощью pandas и openpyxl основной рабочий процесс будет выглядеть следующим образом:

1. Чтение данных — с помощью pandas извлечь данные из разных источников и сохранить в табличные объекты DataFrame (датафрейм).
2. Очистка и анализ данных — с помощью pandas выполнить очистку и преобразование данных, в том числе добавить вычисления, фильтры, обработку пропущенных значений и сделать первичный анализ.
3. Создание отчета — с помощью openpyxl создать новую рабочую книгу Excel или открыть уже существующую, заполнить эту рабочую книгу консолидированными данными, применить условное форматирование, создать диаграммы и вставить необходимые визуальные элементы.
4. Сохранение отчета — с помощью openpyxl сохранить заполненную рабочую книгу Excel, указав имя файла и его местоположение.
5. Рассылка отчета и автоматизация процесса — отправить по электронной почте сгенерированный отчет всем адресатам, поделиться им через файлообменник или любым другим удобным способом.

Другие пакеты Python для Excel

Тем не менее, как бы ни был эффективен openpyxl для решения задач Excel, особенно в сочетании с pandas, у него есть свои ограничения. К счастью, существуют и другие пакеты для разных специфичных случаев. Упомянем некоторые пакеты, которые могут вам пригодиться:

- XlsxWriter[4].

  Как и openpyxl, пакет XlsxWriter можно использовать для работы с файлами Excel с расширением xlsx, включая запись данных, форматирование и создание диаграмм. Этот пакет оптимизирован для улучшения производительности, особенно при работе с большими наборами данных. Но при этом, как следует из его названия, XlsxWriter может только записывать данные в файлы Excel, в то время как openpyxl может и читать, и записывать.

[4] См. **https://clck.ru/3KLXSF**.

- `xlwings`[5].

  С помощью этого пакета можно автоматизировать задачи Excel, включая взаимодействие с рабочими книгами, запуск макросов VBA и работу с COM (Component Object Model) API Excel в Windows. Пакет `xlwings` обеспечивает полноценное двустороннее взаимодействие Excel и Python, чего не умеет делать `openpyxl`. С другой стороны, этот пакет требует более сложной среды разработки, и многие функции доступны только в Windows.

- `PyXLL`[6].

  Это платная библиотека, которая позволяет с помощью Python создавать свои надстройки для Excel. Вместо автоматизации рабочих книг Excel пакет `PyXLL` предлагает разработчикам создавать отдельные приложения для анализа данных, обработки финансовых операций и многого другого. Таким образом, пользователи могут работать с приложениями, разработанными на Python, непосредственно из Excel, без необходимости самим писать или понимать логику кода на Python.

Для решения задач, связанных с Excel, существует множество других пакетов Python, каждый из которых обладает своими уникальными достоинствами и недостатками.

Пример автоматизации Excel с помощью *pandas* и *openpyxl*

Теперь давайте перейдем от слов к делу! В этом разделе мы автоматизируем создание небольшого отчета на Python, используя пакеты `pandas`, `openpyxl` и др.

Сначала мы задействуем `pandas` для решения сложных задач по очистке и анализу данных, которые трудно выполнить в Excel. Затем создадим рабочий лист, состоящий из краткого отчета о данных и двух диаграмм: одна будет родная для Excel, а вторая — полностью из Python. Наконец, мы загрузим весь набор данных на новый рабочий лист и отформатируем результаты.

Полная версия этого скрипта содержится в файле ch_12.ipynb, расположенном в папке ch_12 сопроводительного репозитория к этой книге[7]. Если вы не знаете, как открыть, редактировать или запускать этот файл, обратитесь к *части 3* моей книги «Погружение в аналитику данных: От Excel к Python и R»[8], в которой дано краткое руководство по работе с Python и Jupyter Notebook.

Чтобы начать работу, нам нужно импортировать соответствующие модули и набор данных:

---

[5] См. **https://clck.ru/3KLXnz**.

[6] См. **https://clck.ru/3KLY4J**.

[7] См. **https://clck.ru/3KLaZv**.

[8] См. **https://clck.ru/3KLRyH**.

```
In [1]: # Обработка данных и визуализация
        import pandas as pd
        import seaborn as sns

        # Работа с файлами Excel
        from openpyxl import Workbook
        from openpyxl.styles import PatternFill
        from openpyxl.chart import BarChart, Reference
        from openpyxl.drawing.image import Image
        from openpyxl.utils import get_column_letter
        from openpyxl.utils.dataframe import dataframe_to_rows
        from openpyxl.worksheet.table import Table, TableStyleInfo
```

Библиотека `pandas` может импортировать данные различных форматов, в том числе и из рабочих книг Excel, — с помощью функции `read_excel()`. Давайте импортируем из папки ch_12 сопроводительного репозитория к этой книге файл data\contestants.xlsx и назовем полученный датафрейм `contestants`:

```
In [2]: contestants = pd.read_excel('data/contestants.xlsx')
```

Очистка данных с помощью *pandas*

Датафрейм может содержать тысячи или даже миллионы строк, поэтому непрактично и неэффективно при выполнении вычислений на каждом этапе анализа выводить все строки на экран. Тем не менее возможность визуального просмотра данных необходима для их понимания, чем и привлекает пользователей Excel. Чтобы быстро просмотреть данные и убедиться, что они соответствуют нашим ожиданиям, мы можем использовать метод `head()`, который выводит на экран первые пять строк:

```
In [3]: contestants.head()
```

```
Out[3]:
```

	EMAIL	PRE	POST	SEX	EDUCATION	STUDY_HOURS
0	smehaffey0@creativecommons.org	485	494	Male	Bachelor's	20.0
1	dbateman1@hao12@.com	462	458	Female	Bachelor's	14.8
2	bbenham2@xrea.com	477	483	Female	Bachelor's	22.2
3	mwison@@g.co	480	488	Female	Bachelor's	21.3
4	jagostini4@wordpress.org	495	494	Female	NaN	26.2

На основе этого предварительного просмотра мы уже можем выявить несколько проблем, которые нужно исправить. Во-первых, некоторые адреса электронной почты, как оказалось, имеют неправильный формат. Во-вторых, в столбце `EDUCATION` встречается значение `NaN`, которому здесь не место. Для нашего набора данных мы можем решить эти и другие проблемы, но это было бы трудно или невозможно сделать с помощью Excel.

Работа с метаданными

Правильный процесс очистки данных должен в равной степени уметь работать как с данными, так и с метаданными, — например, с заголовками столбцов. И здесь pandas является особенно удобным инструментом.

Пока что в нашем датафрейме имена столбцов записаны в верхнем регистре. Чтобы упростить себе ввод с клавиатуры, я предпочитаю использовать имена столбцов в нижнем регистре. К счастью, в pandas это можно сделать с помощью одной строчки кода:

```
In [4]: contestants.columns = contestants.columns.str.lower()
        contestants.head()

Out[4]:

   email                          pre post sex    education  study_hours
0  smehaffey0@creativecommons.org 485 494  Male   Bachelor's 20.0
1  dbateman1@hao12@.com           462 458  Female Bachelor's 14.8
2  bbenham2@xrea.com              477 483  Female Bachelor's 22.2
3  mwison@@g.co                   480 488  Female Bachelor's 21.3
4  jagostini4@wordpress.org       495 494  Female NaN        26.2
```

Поиск по шаблону и регулярные выражения

В столбце email нашего датафрейма перечислены адреса электронной почты каждого участника конкурса. Наша задача — удалить все строки, содержащие адреса в недопустимом формате.

Для этого мы можем использовать сопоставление текстовых шаблонов, которое выполняется с помощью *регулярных выражений* (Regular Expression). Хотя в Power Query есть базовая функциональность для обработки текста — например, для изменения регистра, в нем отсутствует текстовый поиск по шаблону — возможность, которую предоставляет Python.

Составление регулярных выражений может оказаться сложной задачей, но с этим вам всегда могут помочь многочисленные интернет-ресурсы, такие как ChatGPT. Мы применим следующее регулярное выражение:

```
In [5]: # Определение регулярного выражения для корректных email-адресов
        email_pattern = r'^[a-z0-9]+[\._]?[a-z0-9]+[@]\w+[.]\w{2,3}$'
```

Далее мы можем использовать метод str.contains(), чтобы оставить только те записи, которые соответствуют шаблону:

```
In [6]: full_emails = contestants[contestants['email'].str.contains(email_pattern)]
```

Чтобы проверить, сколько строк было исключено, можно сравнить атрибут shape двух датафреймов:

```
In [7]: # Размеры исходного DataFrame
        contestants.shape
Out[7]: (100, 6)
```

```
In [8]: # Размеры DataFrame, содержащего только корректные email
        full_emails.shape
Out[8]: (82, 6)
```

Из-за того, что мы уточнили нашу выборку и оставили в ней только участников с правильно введенными адресами электронной почты, число участников сократилось со 100 до 82.

Обработка отсутствующих значений

Метод `info()` возвращает подробную информацию о размерах датафрейма и некоторые его свойства:

```
In [9]: full_emails.info()

    <class 'pandas.core.frame.DataFrame'>
    Index: 82 entries, 0 to 99
    Data columns (total 6 columns):
     #   Column       Non-Null Count  Dtype
    ---  ------       --------------  -----
     0   email        82 non-null     object
     1   pre          82 non-null     int64
     2   post         82 non-null     int64
     3   sex          82 non-null     object
     4   education    81 non-null     object
     5   study_hours  82 non-null     float64
    dtypes: float64(1), int64(2), object(3)
    memory usage: 4.5+ KB
```

Во многих компьютерных программах, в том числе в Power Query, есть ключевое слово `null`, обозначающее отсутствующее или неопределенное значение. В датафреймах `pandas` для таких значений используется ключевое слово `NaN` (от *англ.* Not a Number).

Хотя в классическом Excel нет строгого аналога значению `null`, в Power Query было введено это значение (о чем уже рассказывалось в *главе 2*), и это существенно улучшило обработку и анализ данных. Тем не менее работа в Power Query с такими отсутствующими значениями — например, удаление их из всех столбцов — может представлять определенные сложности. Но эта задача легко решается с помощью `pandas`.

Так, если нужно определить, в каких столбцах самый большой процент отсутствующих значений, в `pandas` это можно сделать с помощью одной строки кода:

```
In [10]: full_emails.isnull().mean().sort_values(ascending=False)

Out[10]:

    education      0.012195
    email          0.000000
    pre            0.000000
```

```
post          0.000000
sex           0.000000
study_hours   0.000000
dtype: float64
```

Поскольку отсутствующих значений очень мало и они встречаются только в одном столбце, мы можем просто исключить все строки, в которых есть отсутствующие значения в любом столбце:

```
In [11]: complete_cases = full_emails.dropna()
```

Чтобы убедиться, что все отсутствующие значения были удалены из датафрейма, мы можем снова вызвать метод `info()`:

```
In [12]: complete_cases.info()

<class 'pandas.core.frame.DataFrame'>
Index: 81 entries, 0 to 99
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   email        81 non-null     object
 1   pre          81 non-null     int64
 2   post         81 non-null     int64
 3   sex          81 non-null     object
 4   education    81 non-null     object
 5   study_hours  81 non-null     float64
dtypes: float64(1), int64(2), object(3)
memory usage: 4.4+ KB
```

Процентильное ранжирование

Используя `pandas`, сделаем процентильное ранжирование для столбца `post` и проверим его с помощью функции `describe()`, которая генерирует описательную статистику:

```
In [13]: complete_cases['post_pct'] = complete_cases['post'].rank(pct=True)
         complete_cases['post_pct'].describe()

Out[13]:
         count    81.000000
         mean      0.506173
         std       0.290352
         min       0.012346
         25%       0.265432
         50%       0.506173
         75%       0.759259
         max       1.000000
         Name: post_pct, dtype: float64
```

Сделать процентильное ранжирование в Excel достаточно просто, но в pandas проще выполнить проверку результатов с помощью его статистических функций, методов для обработки отсутствующих значений и многого другого.

Проверим теперь, что с помощью pandas мы очистили и преобразовали наш набор данных:

```
In [14]: complete_cases.describe()

Out[14]:

             pre        post       study_hours post_pct
count  81.000000  81.000000  81.000000   81.000000
mean  480.506173 481.012346 23.445679   0.506173
std    20.626514  23.037737  8.178142    0.290352
min   409.000000 398.000000 0.000000    0.012346
25%   470.000000 467.000000 18.700000   0.265432
50%   484.000000 483.000000 22.600000   0.506173
75%   494.000000 497.000000 29.000000   0.759259
max   521.000000 540.000000 42.800000   1.000000
```

И далее воспользуемся openpyxl для создания отформатированного итогового отчета.

Создание отчета с помощью *openpyxl*

Итак, подготовив данные в pandas, создадим сводный отчет в Excel с помощью пакета openpyxl. Отчет будет включать в себя как цифры и текст, так и визуализацию данных.

Создание рабочего листа для отчета

Чтобы начать построение рабочей книги Excel с помощью openpyxl, нужно объявить переменные для объектов рабочей книги и рабочего листа:

```
In [14]: # Создание новой рабочей книги
         wb = Workbook()

         # Присвоение переменной ws активного рабочего листа
         ws = wb.active
```

После этого мы можем заполнить любую ячейку активного рабочего листа, используя буквенно-цифровую ссылку на нее. Я вставлю текст и средние значения для столбцов pre и post в ячейки A1:B2:

```
In [16]: ws['A1'] = "Average pre score"

         # Округление вычисленного значения до двух десятичных знаков
         ws['B1'] = round(complete_cases['pre'].mean(), 2)
         ws['A2'] = "Average post score"
         ws['B2'] = round(complete_cases['post'].mean(), 2)
```

Такая вставка данных в рабочую книгу представляет собой простое заполнение ячеек без какого-либо форматирования в Excel. По своему опыту работы с данными могу предположить, что тексту в столбце A будет не хватать ширины столбца по умолчанию. Настроить ширину столбца можно с помощью свойства width:

```
In [17]: ws.column_dimensions['A'].width = 16
```

Позже в этой главе мы узнаем о том, как добиться автоматического подбора ширины столбцов. А сейчас переключим свое внимание на добавление диаграмм в наш отчет.

Вставка диаграмм

Существуют два способа создания диаграмм Excel с помощью Python. Первый способ заключается в создании диаграммы Excel непосредственно из кода Python, а во втором способе на Python создается произвольный график, а в рабочую книгу Excel вставляется его статичное изображение. Оба способа имеют свои плюсы и минусы, с которыми мы разберемся далее.

Способ 1: создание диаграммы Excel

Визуализация данных в Excel очень популярна, потому что диаграммы легко создавать, и они помогают решать основные задачи по визуализации. Рассмотрим здесь, как в Python создавать родные диаграммы Excel с помощью openpyxl.

Для начала нам нужно указать тип диаграммы Excel, которую мы хотим создать, и определить диапазон данных для диаграммы на рабочем листе:

```
In [18]: # Создание объекта столбчатой диаграммы
         chart = BarChart()

         # Определение диапазона данных
         data = Reference(ws, min_col=2, min_row=1, max_col=2, max_row=2)
```

Далее добавить этот источник данных в диаграмму, определить заголовок диаграммы и подписи к осям:

```
In [19]: # Добавление в диаграмму данных
         chart.add_data(data)

         # Определение заголовка диаграммы и меток осей
         chart.title = "Score Comparison"
         chart.x_axis.title = "Score Type"
         chart.y_axis.title = "Score Value"
```

Немного изменим настройки диаграммы: зададим в качестве подписей к горизонтальной оси (к категориям) текст из первого столбца диапазона, а также скроем легенду:

```
In [20]: # Определение названий категорий
         categories = Reference(ws, min_col=1, min_row=1, max_row=2)
         chart.set_categories(categories)
```

```
# Скрыть легенду
chart.legend = None
```

Когда диаграмма полностью определена и настроена, можно вставить ее в рабочий лист:

```
In [21]: # Вставка диаграммы в определенную ячейку рабочего листа
         ws.add_chart(chart, "D1")
```

Способ 2: вставка изображения из Python

По сравнению с Excel у Python есть свои преимущества при визуализации данных, поскольку у него существует множество разнообразных вариантов визуализации, и в нем легче настраивать графики. Так, например, в Excel нет встроенной функциональности для одновременного анализа взаимосвязей между несколькими переменными. А в пакете визуализации данных `seaborn` есть функция `pairplot()`, которая позволяет быстро и удобно исследовать такие взаимосвязи с помощью парного графика.

Следующий фрагмент кода визуализирует эти взаимосвязи по выбранным столбцам `contestants`. Результат можно увидеть на рис. 12.1:

```
In [22]: sns.pairplot(contestants[['pre', 'post', 'study_hours']])
```

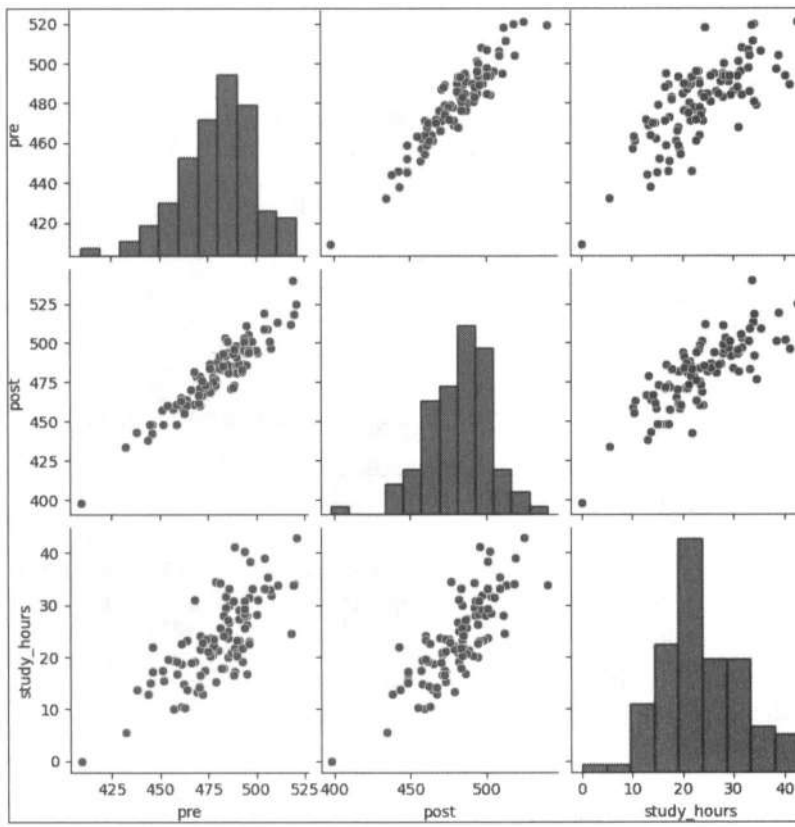


Рис. 12.1. Парный график, созданный в `seaborn`

Python хорош не только тем, что позволяет строить графики таких типов, которые сложно создать в Excel, но графики в Python еще и очень легко настраивать. Например, я хочу добавить на мой парный график разделение по столбцу sex, что можно сделать с помощью параметра hue (рис. 12.2). Я сохраню созданный график как sns_plot, чтобы иметь возможность работать с ним дальше:

```
In [23]: sns_plot = sns.pairplot(contestants[['pre', 'post',
             'study_hours', 'sex']], hue='sex')
```

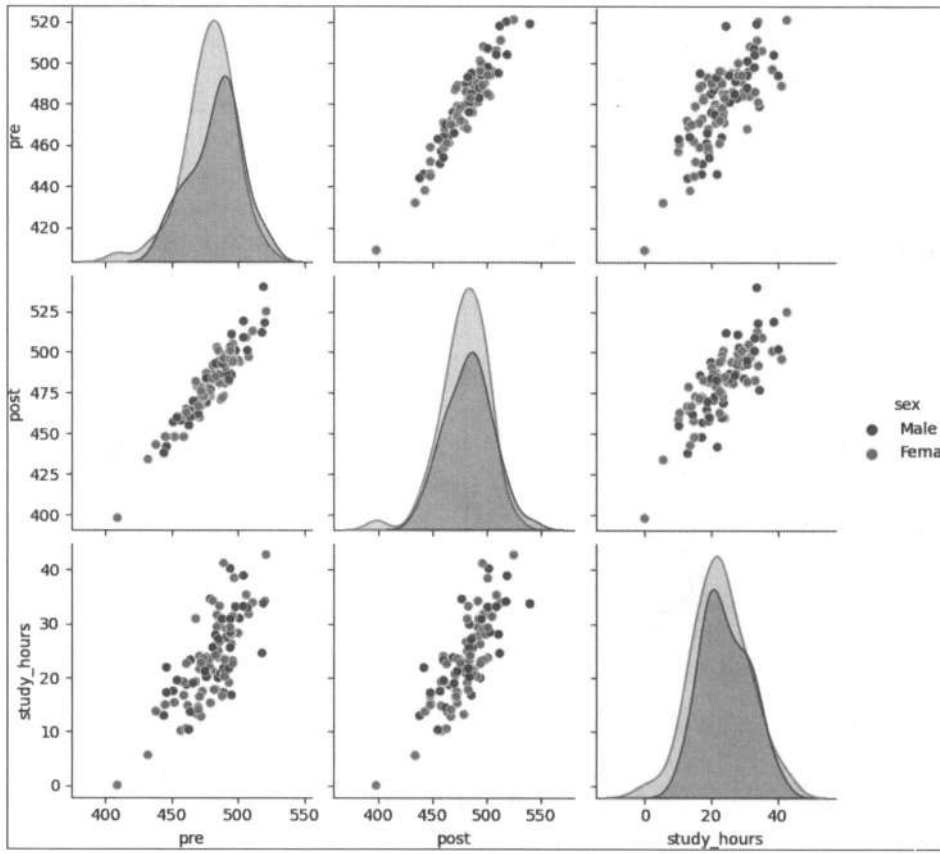


Рис. 12.2. Парный график с разделением по полу

Вставим теперь статичное изображение этого парного графика в рабочую книгу. Для этого сначала нужно сохранить изображение в файл, а затем указать адрес ячейки, куда его нужно вставить:

```
In [21]: # Сохранение изображения парного графика в файл
         sns_plot.savefig('pairplot.png')

         # Вставка сохраненного изображения в рабочий лист
         image = Image('pairplot.png')
         ws.add_image(image, 'A20')
```

Преимущества, за которые я особенно ценю графики Python, — это легкая настройка и простота, с которой можно переключаться между различными типами графиков. В Excel такой метод проб и ошибок сложнее применить из-за ограниченного количества типов диаграмм и трудностей с настройкой привлекательной визуализации.

Тем не менее важно отметить, что графики Python, импортированные в Excel таким образом, по факту являются статичными изображениями. При изменении исходных данных эти графики не будут обновляться автоматически, как это происходит с родными диаграммами Excel. Кроме того, на таких импортированных графиках Python не будет никаких интерактивных опций типа всплывающих подсказок, которые появляются при наведении курсора на элементы привычной диаграммы Excel.

Новая интеграция Python с Excel позволяет создавать графики Python, которые будут обладать некоторой интерактивностью и смогут обновляться в ответ на изменения исходных данных. Отличный пример такой функциональности можно найти в блоге Excel MVP Минды Трейси[9].

Диаграммы Excel и Python

Краткое описание плюсов и минусов этих двух способов приведено в табл. 12.1.

Таблица 12.1. Плюсы и минусы диаграмм Excel и графиков Python

Способ построения	Плюсы	Минусы
Построение родной диаграммы Excel	• Диаграмма будет автоматически обновляться при изменении данных Excel • Диаграмма интерактивная, и пользователь может настроить ее произвольным образом. • Диаграммы могут быть интегрированы с другими опциями Excel — например, с формулами или сводными таблицами	• Очень ограниченное количество типов диаграмм в Excel. • Иногда сложно настроить диаграмму в Excel или повторить ее построение
Вставка изображения с графиком Python	• Возможность использовать разные мощные библиотеки для построения графиков, такие как `matplotlib` и `seaborn`. • График легко проверить и повторить с помощью исходного кода	• График представляет собой статичное изображение, без интерактивности. • Нельзя изменить или обновить график из Excel

Выбор способа создания графика зависит от различных факторов — например, от необходимости обновления данных и наличия определенных типов диаграмм в Excel. Тем не менее универсальность и широкий выбор типов графиков еще раз подчеркивают огромные возможности Python при работе с Excel.

[9] См. https://clck.ru/3KLfLk.

Добавление стилизованной таблицы

Теперь, когда рабочий лист с нашим сводным отчетом создан, добавим второй рабочий лист, на который поместим стилизованную таблицу с датафреймом `complete_cases`. Сначала нужно создать новый рабочий лист:

```
In [25]: ws2 = wb.create_sheet(title='data')
```

Теперь перебрать все строки `complete_cases` и каждую из них вставить в рабочий лист:

```
In [26]: for row in dataframe_to_row(complete_cases, index=False, header=True):
             ws2.append(row)
```

Вставка датафрейма в рабочий лист — это только первый шаг, но в таком простом виде данные могут быть неудобны для чтения и работы с ними. Давайте немного улучшим нашу таблицу.

Изменение формата на проценты

По умолчанию столбец `post_pct` будет иметь формат десятичных цифр, а не процентов, что было бы удобнее для чтения данных. Чтобы изменить формат, нам нужно узнать положение этого столбца в рабочем листе и отформатировать его.

Я здесь применю метод `get_loc()` — чтобы найти индекс столбца `post_pct` в датафрейме, и обязательно прибавлю к результату 1, потому что индексация столбцов в Python начинается с 0, а в Excel — с 1. Функция `get_column_letter()` преобразует этот числовой индекс в буквенный адрес столбца Excel:

```
In [27]: post_pct_loc = complete_cases.columns.get_loc('post_pct') + 1
         post_pct_col = get_column_letter(post_pct_loc)
         post_pct_col

Out[27]: 'J'
```

Определив нужный столбец, я задам требуемый формат каждой ячейке:

```
In [28]: number_format = '0.0%'

         for cell in ws2[post_pct_col]:
             cell.number_format = number_format
```

Преобразование в таблицу Excel

Как уже говорилось в *главе 1*, таблицы Excel обладают рядом преимуществ для хранения данных и анализа. Мы можем преобразовать наш набор данных в таблицу Excel с помощью следующего кода:

```
In [29]: # Задание нужного стиля таблицы
         style = TableStyleInfo(name='TableStyleMedium9', showRowStripes=True)

         # Определение названия таблицы и ее диапазона
         table = Table(displayName='contestants',
```

```
        ref='A1:' + get_column_letter(ws2.max_column) +
            str(ws2.max_row))

    # Применение стилей и вставка в рабочий лист
    table.tableStyleInfo = style
    ws2.add_table(table)
```

Применение условного форматирования

Чтобы улучшить читаемость нашего отчета для конечных пользователей, мы можем применить к рабочему листу условное форматирование. Следующий код применит зеленую заливку фона к участникам, ранг которых выше 90-го процентиля, и желтую заливку к участникам, ранг которых выше 70-го процентиля:

```
In [30]: # Определение стиля условного форматирования
         green_fill = PatternFill(start_color="B9E8A2",
             end_color="B9E8A2", fill_type="solid")
         yellow_fill = PatternFill(start_color="FFF9D4",
             end_color="FFF9D4", fill_type="solid")

         # Проход по таблице и применение условного форматирования
         for row in ws2.iter_rows(min_row=2, min_col=1,
                max_col=len(complete_cases.columns)):
             # Преобразование индекса столбца к индексации с 0
             post_pct = row[post_pct_loc - 1].value
             if post_pct > .9:
                 for cell in row:
                     cell.fill = green_fill
             elif post_pct > .7:
                 for cell in row:
                     cell.fill = yellow_fill
```

Автоподбор ширины столбцов

Несмотря на то что в `openpyxl` нет функциональности для автоматического изменения ширины столбцов рабочего листа, мы можем сделать это с помощью следующего кода. В нем для каждого столбца ищется самый длинный текст, и в зависимости от длины текста соответствующим образом корректируется ширина этого столбца:

```
In [31]: for column in ws2.columns:
         max_length = 0
         column_letter = column[0].column_letter
         for cell in column:
             try:
                 if len(str(cell.value)) > max_length:
                 max_length = len(cell.value)
             except:
                 pass
```

```
        adjusted_width = (max_length + 2) * 1.2
        ws2.column_dimensions[column_letter].width = adjusted_width
```

Заполнив рабочую книгу данными, мы можем сохранить ее файл ch12-output.xlsx в папку output:

```
In [32]: wb.save('output/ch12-output.xlsx')
```

Заключение

Эта глава была посвящена исключительной роли языка Python в развитии современного Excel, его универсальности как «склеивающего» инструмента при разработке сложных систем и его способности расширять функциональные возможности Excel. На практических примерах мы рассмотрели, как с помощью Python можно автоматизировать задачи Excel, выполняя действия, которые сложно или даже невозможно реализовать средствами только программы Excel.

Поскольку Microsoft продолжает интегрировать Python в свои приложения для анализа данных, взаимодействие Python и Excel будет только усиливаться. И в этой главе дана надежная основа для совместного использования Python и Excel, раскрывающая их широчайший потенциал.

Упражнения

Чтобы создать краткий сводный отчет по файлу websites.xlsx из папки exercises\ch_12_exercises сопроводительного репозитория к этой книге[10], откройте в Jupyter Notebook файл ch_12_exercises.ipynb, который я подготовил для вас в качестве основы для этого упражнения. Заполните недостающие пробелы в коде Python, чтобы собрать сводный отчет. Готовое решение можно посмотреть в файле ch_12_exercise_solution.ipynb, расположенном в той же папке репозитория.

Чтобы проверить правильное написание кода, вы всегда можете вернуться к примерам этой главы. Рекомендую вам усложнить себе задачу, расширив отчет несколькими дополнительными визуализациями.

[10] См. **https://clck.ru/3KLhZT**.

Заключение и дальнейшие шаги

В предисловии я сформулировал следующую цель нашего обучения:

> По завершении чтения книги вы научитесь использовать инструменты современного Excel для очистки данных, их анализа, создания отчетов и расширенной аналитики.

Я искренне надеюсь, что эта цель достигнута, и теперь вы уверены в том, что сможете продвинуться в других областях аналитики. Поскольку мы подошли к завершению начального этапа вашего знакомства с современной аналитикой, я хотел бы упомянуть несколько перспективных тем, которые еще сильнее расширят ваш кругозор.

Другие функциональности Excel

В предисловии я уже упоминал, что эта книга не может охватить все интересные возможности современной аналитики в Excel. Тем не менее я хочу привести список достойных внимания ресурсов и функциональностей для самостоятельного изучения. Знакомство с ними еще больше улучшит ваше знание предмета.

Безусловно, существует множество инструментов, которые заслуживают нашего внимания, и постоянно появляются новые. Если вы найдете еще какие-либо полезные инструменты, не упомянутые здесь, пожалуйста, потратьте время, чтобы разобраться, как они работают, и поделитесь своими выводами со мной и сообществом. В конце концов, чтобы охватить все возможности Excel, нужны коллективные усилия и не одна книга.

Функции *LET()* и *LAMBDA()*

Функции `LET()` и `LAMBDA()` значительно повышают эффективность, читаемость и гибкость работы с формулами Excel. Приведу краткое описание обеих функций:

- `LET()`

  Функция `LET()` позволяет присваивать значения переменным, улучшая читаемость формул и облегчая сложные вычисления. Определив переменные в самом начале, можно просто ссылаться на них в формулах, упрощая вид сложных формул.

- `LAMBDA()`

  Функция `LAMBDA()` позволяет создавать в Excel пользовательские функции. Это поддерживает модульность кода и минимизирует избыточность формул, разре-

шая использовать сложные операции повторно в разных формулах. Пользовательские функции могут быть написаны специально для каких-либо конкретных аналитических задач, что повысит производительность и позволит разрабатывать сложные специализированные модели и отчеты.

По сути, функции `LET()` и `LAMBDA()` предоставляют пользователям Excel расширенные инструменты для усовершенствования аналитических процессов, улучшения читаемости формул и усиления гибкости и адаптивности электронных таблиц. Для подробного изучения этих функций обратитесь к *главе 15* книги Alan Murray. «Advanced Excel Formulas: Unleashing Brilliance with Excel Formulas» (Apress, 2022)[1].

Power Automate, сценарии Office и Excel Online

Развитие аналитики и автоматизации в Excel значительно ускорилось благодаря интеграции Power Automate[2], сценариев Office[3] и Excel Online[4] — каждый из этих инструментов может внести свой уникальный вклад в оптимизацию рабочих процессов и повышение производительности.

Power Automate является ключевым инструментом для автоматизации широкого спектра задач в Excel и других приложениях. Его возможности охватывают как простой ввод данных, так и сложные бизнес-процессы. Отдельно нужно отметить его умение работать с Power Query для автоматизации преобразования данных.

Кроме преобразования данных, Power Automate позволяет автоматизировать создание отчетов и внедрить систему уведомлений. Всё это позволяет автоматизировать рутинные задачи, разрешая пользователям сосредоточиться на более важных стратегических вопросах.

Интеграция сценариев Office и Excel Online еще больше расширила возможности Power Automate. Сценарии Office, доступные в Excel Online, позволяют с помощью скриптового языка записать и автоматизировать повторяющиеся в Excel задачи. В сочетании с Power Automate это дает возможность автоматически запускать такие сценарии в ответ на определенные триггеры или события. Например, можно настроить автоматический запуск сценария, предназначенного для корректировки данных в рабочей книге, при каждой загрузке нового файла в указанную папку SharePoint.

Кроме того, взаимодействие Power Automate с Excel Online открывает возможности для совместной работы и обработки данных в режиме реального времени. Это упрощает такие операции, как создание и обновление рабочих книг, а также извлечение данных из файлов Excel, хранящихся в облаке, и обеспечивает бесперебойную совместную работу нескольких пользователей в Excel Online. Такая интегра-

[1] См. **https://clck.ru/3KLkzr**.

[2] См. **https://clck.ru/3KLmSt**.

[3] См. **https://clck.ru/3KLnHd**.

[4] См. **https://clck.ru/3KLoBt**.

ция позволяет не только автоматизировать и масштабировать рабочие процессы, но и выполнять совместную работу нескольким пользователям.

Фактически совместное использование Power Automate, сценариев Office и Excel Online предоставляет пользователям Excel расширенные возможности для автоматизации, совместной работы и повышения производительности. Это открывает новые возможности в управлении данными и их анализе, и, как следствие, эти инструменты стали незаменимыми для пользователей Excel, стремящихся оптимизировать свои рабочие процессы. Несмотря на то что подробные описания или книги, посвященные конкретно этому трио, не всегда можно найти, полезная информация и понятные примеры есть в документации Microsoft[5].

Дальнейшее изучение Power Query и Power Pivot

Значительная часть этой книги посвящена изучению Power Query и Power Pivot в Excel — инструментов, необходимых для очистки и анализа данных. Несмотря на то что с их базовыми функциями можно работать, владея минимальными техническими навыками, более глубокое знание основных концепций и этих инструментов значительно повысит эффективность их использования.

Power Query и M

Дальнейшее изучение таких понятий Power Query и M, как параметры, пользовательские функции, оптимизация запросов, автоматизация обновлений и управление запросами, даст вам множество преимуществ.

Параметры и пользовательские функции обеспечивают гибкость и кастомизацию рабочих процессов по очистке и анализу данных. *Параметры* позволяют определять значения, которые затем можно по-разному использовать в запросах, — например, в условиях фильтрации или в настройке соединения, что делает запросы более динамичными и гибкими. Так, вы можете создать параметр для диапазона дат, который будет применяться при получении данных, что дает возможность легко обновлять данные без изменения логики запроса.

Пользовательские функции нужны для определения повторно используемых фрагментов кода для выполнения действий, напрямую не доступных в стандартной функциональности Power Query, или для сложных преобразований, которые нужно применить к нескольким наборам данных. Определив пользовательскую функцию, вы можете объединить ряд действий в единую вызываемую функцию, упрощая таким образом выполнение задач по обработке данных и обеспечивая согласованность ваших запросов.

Оптимизация запросов имеет огромное значение по мере роста объема и сложности наборов данных. Оптимизация запросов обеспечивает эффективную обработку

[5] См. **https://clck.ru/3KLp5b**.

данных и позволяет сократить время выполнения. Знание способов оптимизации запросов помогает ускорить время выполнения запросов, устранить ненужные преобразования и повысить общую производительность. Эти навыки пригодятся вам при работе с большими наборами данных и сложными преобразованиями для повышения производительности и скорости анализа.

Автоматизация обновлений и *управление запросами* тоже важны для обеспечения актуальности и надежности анализа данных. Автоматизация процесса обновления гарантирует периодическое обновление данных без ручного вмешательства, что экономит время и силы. Эффективное управление запросами позволяет организовывать, отслеживать и устранять ошибки при преобразовании данных и подключении к их источникам. Этот навык поможет сохранять целостность данных, их надежность и точность для принятия обоснованных решений.

Кроме всех этих понятий, обеспечивающих гибкость, эффективность и повышенную надежность данных, при работе с Power Query, как и при работе с любым приложением, важно иметь четкую цель. Для большинства проектов целью будет создание удобных источников данных, которые легко интегрируются с моделью данных в Power Pivot. Знание принципов моделирования данных, таких как нормализация и проектирование схем, станет решающим фактором для полноценного использования Power Query в качестве инструмента преобразования данных. А эффективное моделирование данных, в свою очередь, пригодится при переходе от Power Query к Power Pivot.

Power Pivot и DAX

Чтобы в полной мере использовать возможности Power Pivot в Excel, важно уметь работать с DAX и моделированием данных. Овладев этими понятиями, вы сможете по максимуму использовать Power Pivot и эффективно решать сложные бизнес-вопросы.

Прежде чем добавлять меры в свою модель данных, очень важно обеспечить правильное проектирование самой модели данных. Для этого необходимо знать такие понятия, как схема «звезда» (кратко представлена в *главе 7*), схема «снежинка» и третья нормальная форма. Эти понятия играют ключевую роль в моделировании и в эффективном хранении данных.

По мере того как вы будете создавать все более сложные меры на языке DAX, возникнет необходимость в оптимизации кода и улучшении его читаемости. Одним из эффективных способов достижения этого является использование переменных DAX. Сохраняя промежуточные результаты внутри меры, с помощью переменных DAX можно повысить прозрачность и производительность кода.

Тем, кто хочет углубиться в изучение продвинутых техник Power Pivot и DAX, я рекомендую прочитать книгу Matt Allington «Supercharge Excel: When You Learn to Write DAX for Power Pivot» (Holy Macro! Books, 2018)[6]. Она содержит ценные

[6] См. **https://clck.ru/3KLrKf**.

идеи и практические рекомендации для расширения ваших навыков и знакомства с полным потенциалом Power Pivot в Excel.

Power BI для информационных панелей и отчетов

В *главе 7* вы уже познакомились с загрузкой вашей модели из Power Pivot в Power BI и получили краткое представление о том, как построена совместная работа Excel и Power BI. Дальнейшее развитие своих навыков в использования Power BI даст вам множество дополнительных преимуществ в области анализа данных, визуализации и создания отчетов. Учитывая то, что при чтении этой книги вы уже познакомились с Power Query и Power Pivot/DAX, переход к Power BI может стать для вас очень легким и естественным шагом вперед.

Одним из ключевых преимуществ Power BI, привлекающих пользователей Excel, является превосходная визуализация данных и функциональность для создания информационных панелей. В отличие от универсального Excel, Power BI специализируется на создании интерактивных отчетов и информационных панелей, которые доступны для совместного использования с различных устройств, что упрощает одновременный доступ к аналитической информации. Кроме того, аналитика в режиме реального времени в Power BI позволяет мгновенно принимать решения на основе актуальных данных, что очень важно для тех, кто работает с динамичными данными или нуждается в постоянном мониторинге показателей.

Для тех, кто только начинает работать с Power BI, очень важно понимать экосистему сервисов Power BI, которая включает в себя Power BI Desktop для создания отчетов и Power BI Services для их распространения. При этом ключевыми навыками считаются умение создавать интерактивные визуализации, расширенное моделирование данных, выходящее за рамки возможностей Excel, и знание DAX для выполнения сложных анализов и вычислений. Кроме того, понимание, как Excel интегрируется с Power BI, позволит вам легко работать на обеих платформах, повышая эффективность анализа данных и отчетности.

Azure и облачные вычисления

Azure предоставляет огромные преимущества для выполнения современной аналитики в Excel. С помощью облачной инфраструктуры и сервисов Azure пользователи Excel могут усовершенствовать свои рабочие процессы по аналитике данных. Azure предоставляет интеграцию с такими инструментами, как Power BI — для создания интерактивных отчетов и визуализации, Azure Machine Learning — для построения предсказательных моделей и Azure Cognitive Services — для анализа неструктурированных данных. Вы уже немного познакомились с возможностями Azure в *главе 11* при выполнении анализа настроений.

Объединение мощных функциональных возможностей Excel с расширенными способностями Azure откроет вам новые горизонты для анализа данных, машинного обучения и принятия решений на основе данных. Для более подробного знакомства с Azure я могу порекомендовать книгу Jonah Andersson «Learning Microsoft Azure:

Cloud Computing and Development Fundamentals» (O'Reilly, 2023)[7]. За ценными сведениями и практическими советами по внедрению машинного обучения и ИИ в Power BI с использованием Azure можно обратиться к книге Tobias Zwingmann «AI-Powered Business Intelligence: Improving Forecasts and Decision Making with Machine Learning» (O'Reilly, 2022)[8].

Программирование на Python

Знание Python очень полезно для современных аналитиков в Excel, независимо от того, хотите ли вы автоматизировать создание электронных таблиц (как было показано в *главе 12*) или оптимизировать модели машинного обучения в Azure. Python признан лидирующим языком программирования для разработки ИИ и широко используется в таких фреймворках, как TensorFlow, PyTorch и Keras.

С постоянным развитием ИИ появляются все новые инструменты и фреймворки, что еще раз подчеркивает важность уверенного владения Python для разработчиков. Овладев навыками работы с Python, вы обеспечите себе уверенную позицию в быстро развивающейся области ИИ и в смежных областях. Более того, универсальность Python выходит за рамки ИИ и охватывает различные сферы применения, такие как веб-разработка, анализ данных и автоматизация. Чтобы начать профессионально изучать Python, я рекомендую книгу Эла Свейгарта «Автоматизация рутинных задач с помощью Python», 2-е издание (Вильямс, 2021)[9]. Она послужит прекрасным пособием для начинающих, поскольку отличается практическим подходом к изучению Python и его применению в реальных задачах.

Большие языковые модели и инженерия запросов

В *главе 11* мы рассматривали концепцию запросов на естественном языке в Excel с помощью надстройки Analyze Data, которая представляет собой начало интеграции инструментов ИИ, таких как Copilot, для расширенного анализа данных. Поскольку Copilot постоянно развивается, освоение инструментов на базе ИИ становится все более критичным.

Тем, кто занимается современной аналитикой в Excel, крайне важно знать большие языковые модели (Large Language Model, LLM) и инженерию запросов (Prompt Engineering). Большие языковые модели, такие как GPT (Generative Pre-trained Transformer) от компании OpenAI, умеют понимать и генерировать текст на естественном человеческом языке, что делает их незаменимыми для анализа неструктурированных данных, получения аналитических выводов и составления детальных отчетов.

---

[7] См. **https://clck.ru/3KLsQ6**.

[8] См. **https://clck.ru/3KLsku**.

[9] См. **https://clck.ru/3KLtWv**.

Модель GPT, одна из ключевых LLM, имеет различные реализации — например, чат-бот ChatGPT[10], предназначенный для общения с пользователем. Чтобы по максимуму использовать возможности разговорного ИИ и ChatGPT, необходимо знать основы инженерии запросов. Это включает в себя создание эффективных подсказок (prompts), которые четко формулируют аналитические цели модели, что помогает в получении точных и ценных ответов от ИИ. Работа с Copilot, построенным на основе модели GPT, значительно облегчится, если вы будете владеть этими техниками.

Пользователи Excel могут обнаружить, что умение составлять вопросы, структурировать подсказки и включать в них необходимый контекст способно значительно улучшить процесс анализа данных. Все эти навыки откроют новые возможности для поиска аналитической информации и принятия обоснованных решений.

Напутствие

Писатель-фантаст Уильям Гибсон как-то сказал: «Будущее уже здесь, оно просто неравномерно распределено». Эта мысль сильно резонирует с использованием современного Excel. Если учесть весь его обширный набор возможностей и постоянное появление новых функциональностей, вполне понятно, что это может испугать. Более того, в условиях стремительного развития технологий часто возникает страх остаться позади. Тем не менее шаг за шагом, осознавая, что никто не требует от вас владения Excel в совершенстве, вы можете раскрыть для себя больший потенциал Excel, чем вам казалось вначале, и сохранить конкурентоспособность в условиях современного бизнеса.

Подумайте о том, чего вы уже достигли, прочитав эту книгу, — у вас есть все основания для гордости! Но не останавливайтесь слишком долго на этом. Вам предстоит открыть для себя еще много нового, и совсем скоро вы обнаружите, что эта книга — всего лишь поверхностный обзор. Заканчивая чтение этой главы и этой книги, примите вызов: двигайтесь только вперед, продолжайте учиться и расти как специалист по современной аналитике в Excel.

[10] См. https://clck.ru/3KLuEB.

Предметный указатель

E

ETL (Extract, Transform, Load (извлечение, преобразование, загрузка)) 31

K

KPI (Key Performance Indicators, ключевые показатели эффективности) 89, 124

S

Self-service BI-системы (Business Intelligence) 163

А

Автоматизация обновлений 200
Анализ настроений (Sentiment Analysis) 173
Аргументы в Excel 150

Б

Большие языковые модели (Large Language Model, LLM) 202

В

Визуализация данных 190
Внешнее соединение 79
Внутреннее соединение 75, 79
Вычисляемые столбцы 61, 63, 106

Г

Графики Python 193
Группировка запросов 76

Д

Действия со столбцами 57
Диаграммы Excel 190
Диапазон в Excel 146
Динамические массивы 147
Дополненная аналитика (Augmented Analytics) 164

З

Зависимости запросов 77
Заголовки столбцов 22
Значение null 30, 48

И

Иерархия 110
Инструмент
◊ Azure Cognitive Services 201
◊ Azure Machine Learning 201
◊ Power Automate 198
◊ Power BI 112, 201
◊ Power Pivot 83, 89, 200
◊ Power Query 29, 67, 167, 186
Инструменты ИИ 202
Искусственный интеллект (ИИ) 162

К

Кардинальность 96, 100
Качественные переменные 46
Ключевые показатели эффективности (KPI) 88
Количественные переменные 46
Контекст фильтра (Filter Context) 130, 132

Л

Левое внешнее соединение (left outer join) 72, 79, 84
Лента Power Query 34
Ложноотрицательный результат 172
Ложноположительный результат 172

М

Массив в Excel 146
Меры DAX 118
Метод
◊ get_loc() 194
◊ head() 185
◊ info() 187
◊ str.contains() 186
Модель
◊ GPT (Generative Pre-trained Transformer) 203
◊ данных (Data Model) 86, 118
Модульное тестирование (Unit Testing) 180

Н

Надстройка
◊ Analyze Data (Анализ данных) 164, 165
◊ Azure Machine Learning 164, 173
◊ XLMiner 164, 168, 170
Направление фильтрации 96, 103
Неструктурированные данные 162
Неупорядоченные данные 63
Неявные меры DAX 119, 123

О

Оператор динамического диапазона 151
Оптимизация запросов 199
Оптическое распознавание символов (Optical Character Recognition, OCR) 171
Опция OCR в Excel 171
Отличающиеся (distinct) значения 150
Очистка данных 48

П

Пакет
◊ openpyxl 182–185, 189, 190, 195
◊ pandas 178, 182–189
◊ PyXLL 184
◊ XlsxWriter 183
◊ xlwings 184
Пакеты Python 181, 182
Параметры 199
◊ в Excel 150
Показатели KPI 128
Пользовательские функции 199
Правила упорядоченных данных (tidy data) 27
Предсказательная аналитика 162, 177
Приложение Power BI Desktop 113
Профилирование данных 42, 46, 47
Процентильное ранжирование 188

Р

Разделитель (delimiter) 52
Регулярные выражения (Regular Expression) 186
Редактор
◊ Power Pivot 97
◊ Power Query 34, 47
Реляционная модель данных 84, 91
Реляционное соединение 71, 72

С

Сводная диаграмма 97
Сводная таблица 97
Связи между таблицами 92
Сетевой эффект 180
Система контроля версий 181
Системы
◊ генеративной обработки естественного языка (Natural Language Processing, NLP) 162
◊ управления предприятием (ERP, Enterprise Resource Planning) 55
Список Applied Steps (Примененные шаги) 37, 58
Статические массивы 147
Структурированные ссылки 27
Схема «звезды» 96
Сценарии Office 198

Т

Таблицы
◊ измерений (dimension tables) 95
◊ фактов (fact tables) 95
Типы данных 62

У

Уникальные (unique) значения 150
Упорядоченные данные 63
Управление запросами 200
Условная логика 132
Условное форматирование 195

Ф

Формула
◊ Excel 145
◊ TOTALYTD() 137
Формульный язык DAX 86
Функции
◊ аналитики времени 135
◊ динамического массива 145, 160
Функция
◊ ABS() 150
◊ CALCULATE() 130, 132, 135, 138
◊ COUNTA() (СЧЁТЗ()) 152
◊ CROSSFILTER() 106
◊ DATEADD() 138
◊ DATESYTD() 138
◊ EXACT() (СОВПАД()) 152
◊ Excel 145
◊ FILTER() (ФИЛЬТР()) 152, 154
◊ LAMBDA() 197
◊ LET() 197
◊ pairplot() 191
◊ RANDARRAY() (СЛУЧМАССИВ()) 159
◊ read_excel() 185
◊ SAMEPERIODLASTYEAR() 138
◊ SEQUENCE() (ПОСЛЕДОВ()) 159
◊ SORTBY() (СОРТПО()) 154
◊ SUMIFS() (СУММЕСЛИМН()) 154
◊ SWITCH() 108, 109, 117
◊ TEXTSPLIT() (ТЕКСТРАЗД()) 160
◊ UNIQUE() (УНИК()) 149, 150
◊ UPPER() (ПРОПИСН()) 22
◊ USERELATIONSHIP() 137
◊ VLOOKUP() (ВПР()) 71, 72, 83, 90, 94, 156
◊ VSTACK() (ВСТОЛБИК()) 160
◊ XLOOKUP() (ПРОСМОТРХ()) 72, 90, 156, 158

Ч

Чат-бот ChatGPT 203

Я

Явные меры DAX 121, 123
Язык
◊ Visual Basic for Applications (VBA) 30
◊ программирования DAX 108
◊ программирования M 36, 61
◊ программирования Python 178, 179, 196

Об авторе

Джордж Маунт (George Mount) — основатель и CEO консалтинговой компании Stringfest Analytics, специализирующейся на обучении аналитиков. Джордж регулярно выступает с докладами на эту тему и ведет блог **stringfestanalytics.com**.

Помимо того, что он является автором книги «Advancing into Analytics: From Excel to Python and R» (O'Reilly, 2021)[1] (перевод вышел в 2023 г. в издательстве БХВ-Петербург: «Погружение в аналитику данных: от Excel к Python и R»[2]), Джордж был признан самым ценным профессионалом Microsoft (Most Valuable Professional, MVP) за свой вклад в развитие сообщества и технические знания в области Excel.

Джордж Маунт получил степень бакалавра по экономике в колледже Хиллсдейл (Hillsdale College), а также степень магистра в сфере финансов и информационных систем в Университете Кейс Вестерн Резерв (Case Western Reserve University). В настоящее время он проживает в Кливленде, штат Огайо.

[1] См. **https://clck.ru/3KLveW**.

[2] См. **https://clck.ru/3KLw53**.

Об изображении на обложке

Животное на обложке книги — это жук-геркулес (*Dynastes hercules*), разновидность жука-носорога, обитающий в тропических лесах Центральной и Южной Америки, а также на некоторых островах Карибского моря.

Они титаны в мире насекомых — самые длинные жуки на Земле: самцы достигают 7 дюймов в длину (17 см), включая рога, которые они используют в схватках с другими самцами за доминирование (у самок рога полностью отсутствуют). Жуки-геркулесы имеют оливково-зеленую или коричневато-желтую окраску, иногда с радужным отливом, и небольшие черные пятна, разбросанные по всему туловищу. Окраска может меняться в зависимости от влажности окружающей среды. Несмотря на свои внушительные размеры, жуки-геркулесы в целом безобидные насекомые. Они питаются гниющими фруктами и древесным соком, а мощные рога, которые позволяют им поднимать вес, в сотни раз превышающий их собственный, помогают им искать пищу и зарываться в лесную подстилку.

В настоящее время жуки-геркулесы не считаются вымирающим видом. Однако потеря среды обитания из-за вырубки лесов представляет угрозу для их популяций. Многие животные, изображенные на обложках книг издательства O'Reilly, находятся под угрозой исчезновения, и все они важны для нашего мира.

Иллюстрация на обложке выполнена Карен Монтгомери на основе линогравюры из старинной книги Голдсмита «Естественная история» (Goldsmith's Natural History). Дизайн серии разработан Эди Фридман, Элли Фолькхаузен и Карен Монтгомери.

Джордж Маунт

Современная аналитика данных в Excel

Современная аналитика данных в Excel

Если вы все еще не модернизировали процессы очистки данных и создания отчетов в Microsoft Excel, вы, возможно, упускаете шанс существенно повысить эффективность своей работы. А если ваша цель — глубокий и детальный анализ данных, стоит знать: возможности Excel гораздо шире, чем принято думать. Это практическое руководство открывает читателю доступ к современному арсеналу функций Excel и демонстрирует другие мощные инструменты аналитики.

Автор показывает бизнес-аналитикам, специалистам по данным и всем, кто работает с цифрами, как извлечь максимум пользы из привычных таблиц, используя новейшие возможности Excel. Вы научитесь создавать воспроизводимые сценарии очистки данных с помощью Power Query и строить реляционные модели прямо внутри рабочей книги, используя Power Pivot. Кроме того, вы познакомитесь с современными инструментами анализа — от динамических массивов и функций на базе искусственного интеллекта до интеграции с языком Python.

Откройте для себя способы создания отчетов и аналитики, которые раньше казались трудновыполнимыми, а порой и вовсе не возможными в Excel.

- **Создавайте воспроизводимые и надежные процессы очистки данных в Excel с помощью Power Query**
- **Проектируйте реляционные модели и настраивайте аналитические показатели, используя Power Pivot**
- **Быстро извлекайте и трансформируйте данные благодаря функциям динамических массивов**
- **Применяйте возможности искусственного интеллекта для выявления скрытых закономерностей и трендов**
- **Интегрируйте Python в работу с Excel для автоматизации анализа и отчетности**

«Как человек, время от времени использующий Excel в работе, я с легкостью разобралась в этой книге и нашла в ней массу практических советов. Простой и ясный подход Джорджа Маунта будет полезен как опытным аналитикам, так и тем, кто работает с Excel от случая к случаю».

— Меган Финли,
технический писатель
и редактор

Джордж Маунт — основатель и генеральный директор Stringfest Analytics, консалтинговой фирмы, специализирующейся на обучении и повышении квалификации в области аналитики данных, ранее работал с ведущими учебными платформами и компаниями в этой сфере. Удостоен звания Microsoft Most Valuable Professional (MVP) — этой наградой компания отмечает выдающихся экспертов, обладающих глубокими техническими знаниями и активно делящихся ими с сообществом пользователей Excel. Джордж регулярно ведет блоги и выступает с докладами на темы обучения анализу данных.