**Гил Равив. Power Query в Excel и Power BI: сбор, объединение и преобразование данных**

Ранее я перевел книгу [Кен Пульс и Мигель Эскобар. Язык М для Power Query](http://baguzin.ru/wp/?p=20852). А спустя некоторое время с удивлением обнаружил, что страничка книги самая посещаемая среди опубликованных за последние 5 лет. Механизм Power Query для Excel относительно новый, но весьма необычный. Это не чистая работа с данными в Excel, а инструмент импорта внешних данных и их предварительной обработки. Я постоянно извлекаю данные из Интернета, поэтому использую Power Query довольно часто. Гил Равив описывает многое из того, что есть у Кена Пульса, поэтому здесь я не повторяюсь. Больше внимания уделяю новым аспектам: языку М, анализу текстов и извлечению знания из текста.

Гил Равив. Power Query в Excel и Power BI: сбор, объединение и преобразование данных. – СПб.: БХВ-Петербург, 2021. – 480 с.



Купить бумажную или электронную книгу в [издательстве](https://bhv.ru/product/power-query-v-excel-i-power-bi-sbor-obedinenie-i-preobrazovanie-dannyh/), бумажную книгу в [Ozon](https://www.ozon.ru/product/power-query-v-excel-i-power-bi-sbor-obedinenie-i-preobrazovanie-dannyh-231611763/?partner=baguzin) или [Лабиринте](https://www.labirint.ru/books/791818/?p=13320)

### Содержание

[Глава 2. Основные проблемы, возникающие при подготовке данных](#_Глава_2._Основные)

[Столбец из примеров](#_Столбец_из_примеров)

[Объединение чисел в группы](#_Объединение_чисел_в)

[Извлечение гиперссылок из текста](#_Извлечение_гиперссылок_из)

Работа с датами

Глава 3. Объединение данных, полученных из нескольких источников

Глава 4. Объединение несовпадающих таблиц

Глава 5. Сохранение контекста

Глава 6. Отмена свертывания столбцов таблицы

Глава 7. Дополнительные вопросы, связанные с применением преобразования сведения столбцов и отмены свертывания столбцов

Глава 8. Совместная работа

Глава 9. Введение в язык формул M Power Query

Глава 10. От ловушек к робастным запросам

Глава 11. Основы текстовой аналитики

Глава 12. Расширенная текстовая аналитика: извлечение смысла

### Глава 2. Основные проблемы, возникающие при подготовке данных

Примеры в книге часто используют базу AdventureWorks, придуманную Microsoft для демонстрации реализации приложений в бизнес-сценариях. Для учебных целей базу данных AdventureWorks можно развернуть на сервере SQL на локальном ПК (подробнее см. [Дик Куслейка. Визуализация данных при помощи дашбордов и отчетов в Excel](https://baguzin.ru/wp/?p=24370), раздел Базы данных SOL Server). Вы также можете развернуть базу данных в облаке Azure (см. [Quickstart: Create an Azure SQL Database single database](https://docs.microsoft.com/en-us/azure/azure-sql/database/single-database-create-quickstart?tabs=azure-portal)).

#### Столбец из примеров

Загрузите файл C02E02.xlsx с [сайта](https://www.microsoftpressstore.com/store/collect-combine-and-transform-data-using-power-query-9781509307951#downloads) или из приложенного архива. Создайте новую книгу Excel. Пройдите по меню *Данные* –> *Получить данные* –> *Из файла* –> *Из книги*. Выделите файл C02E02.xlsx и нажмите *Импорт*. В открывшемся окне *Навигатор* выберите таблицу *Products* и нажмите кнопку *Преобразовать данные*. Откроется окно редактора Power Query. Столбец *Product Number* содержит четыре кода, разделенных дефисом (например, в первой строке VE-C304-S-BE). Нужно извлечь третье значение, которое соответствует размеру товара.

Выделите столбец *Product Number,* пройдите по меню *Добавление столбца* –> *Столбец из примеров* –> *Из выделения*. Редактор Power Query переходит в новое состояние. В верхней части панели просмотра появляется сообщение: *Введите примеры значений, чтобы создать столбец* (рис. 1). В правой части панели в первую ячейку столбца *Текст между разделителями* введите *S*. Переименуйте столбец в *Size*. Нажмите *Enter*, Power Query заполнит столбец *Size* предложенными значениями, как показано на рис. 1. Нажмите комбинацию клавиш Ctrl+Enter или щелкните мышью на кнопке Ok для создания столбца. Загрузите данные в книгу Excel.

Изображение выглядит как стол

Автоматически созданное описание

Рис. 1. Редактор Power Query в режиме *Добавить столбец из примеров*

Если в окне *Параметры запроса* на строке *Вставленный текст между…* кликнуть на шестеренке, откроется окно *Текст между разделителями*, а верхней части можно увидеть код на языке М, соответствующий настройкам на рис. 2:

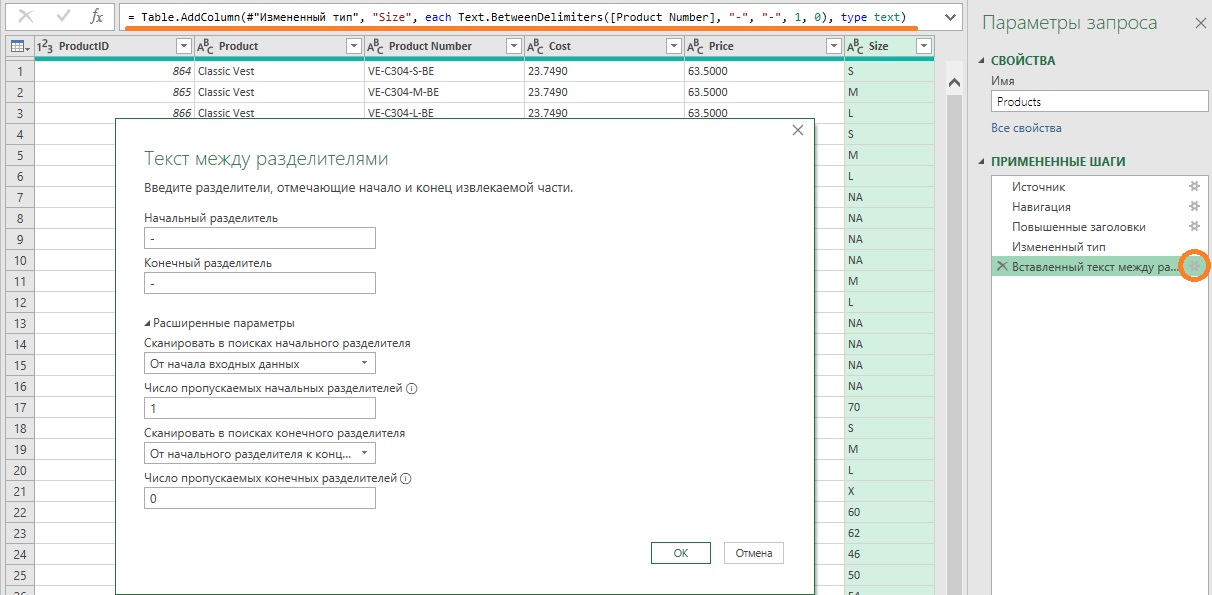


Рис. 2. Интерфейс и код на языке М для извлечения символа между разделителями

С полным перечнем функций, доступных в раскрывающемся меню *Столбец из примеров*, можно ознакомиться на [сайте MS](https://docs.microsoft.com/en-us/power-bi/create-reports/desktop-add-column-from-example).

#### Объединение чисел в группы

Столбец *Size* содержит как буквенные, так и числовые значения. Игнорируем текстовые значения, а числовые значения разобьем на группы:

* ≥ 70 – X
* ≥ 60 – L
* ≥ 50 – M
* < 50 – S

Для извлечения числовых размеров используем обработку ошибок. Для этого кликните правой кнопкой мыши на столбце *Size* и выберите команду *Создать дубликат столбца*. Переименуйте новый столбец в *Size Numbers*. Кликните на нем правой кнопкой мыши и выберите *Изменить тип* –> *Целое число*. Вместо текстовых значения отразятся ошибки – Error. Кликните правой кнопкой мыши на столбце *Size Numbers* и выберите *Удалить ошибки*. Выделите столбец *Size Numbers* и пройдите по меню *Добавление столбца* –> *Столбец из примеров* –> *Из выделенного*. Проставьте для нескольких первых строк буквенные значения группы размеров:

Изображение выглядит как стол

Автоматически созданное описание

Рис. 3. Столбец из примеров для групп размеров

Нажмите Ok или Ctrl+Enter. Отредактируйте код в соответствии с нашими диапазонами:

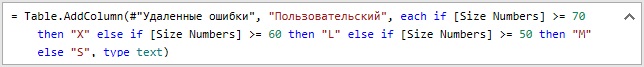


Рис. 4. Отредактированный код

#### Извлечение гиперссылок из текста

Загрузите файл C02E04.xlsx. Создайте новую книгу Excel (нажмите Ctrl+N). Импортируйте файл C02E04.xlsx в редактор Power Query. Для этого пройдите по меню *Данные* –> *Получить данные* –> *Из файла* –> *Из книги*. В окне навигатора выберите лист *Sheet1*, нажмите *Преобразовать данные*. Откроется окно редактора Power Query. Видно, что столбец *Message* содержит текст произвольного формата. Создайте копию столбца *Message*, чтобы сохранить исходное сообщение без изменений. Выделите столбец и пройдите по мнею *Добавление столбца* –> *Создать дубликат столбца*. Переименуйте новый столбец – *Hyperlink*.

Выделите столбец *Hyperlink* и пройдите по мнею *Преобразование* –> *Разделить столбец* –> *По разделителю*. Настройте параметры разделения:

Изображение выглядит как текст

Автоматически созданное описание

Рис. 5. Параметры разделения

Отлично! Но что если некоторые гиперссылки начинаются с "https://" или просто с "www."? На панели *Примененные шаги* выделите шаг *Разделить столбец по разделителю*. В строке формул отобразится:

= Table.SplitColumn(#"Переименованные столбцы", "Hyperlink", Splitter.SplitTextByEachDelimiter({"http://"}, QuoteStyle.Csv, false), {"Hyperlink.1", "Hyperlink.2"})

Для разделения столбца *Hyperlink* с помощью дополнительных разделителей измените имя функции с на Splitter.SplitTextByAnyDelimiter. Разделитель "https://" заключен в двойные кавычки в фигурных скобках. Фигурные скобки представляют собой списки на языке запросов M. Дополните список новыми разделителями. Формула примет вид:

= Table.SplitColumn(#"Переименованные столбцы", "Hyperlink", Splitter.SplitTextByAnyDelimiter ({"http://", "https://", "www."}, QuoteStyle.Csv, false), {"Hyperlink.1", "Hyperlink.2"})

Обратите внимание, что в столбце *Hyperlink.2*, в строке 15 после гиперссылки есть дополнительный текст. Разделите столбец *Hyperlink.2* по разделителю пробел, самый левый разделитель. Удалите столбец *Hyperlink.2.2*.

Проверьте результаты в столбце *Hyperlink.2.* В некоторых строках со значением null не были найдены наши разделители: "http://", "https://", "www." Однако текст в столбце Hyperlink.1 содержит гиперссылку на бесплатную электронную книгу с сайта: aka.ms/containersebook.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 6. Необработанные исключения

Если добавить в список разделителей еще и "aka.ms", этот фрагмент адреса исчезнет из поля после разделения (для предыдущих разделителей это было не страшно, так как www.microsoft.com и microsoft.com ведут на один и тот же сайт). Поэтому сначала мы добавим "aka.ms" в список разделителей, а затем для записей, начинающихся с символа / добавим в начало доменное имя aka.ms. Для этого перейдите к последнему шагу в панели *Примененный шаги*, и пройдите по меню *Добавление столбца* –> *Условный столбец*. Откроется диалог *Добавление условного столбца*.

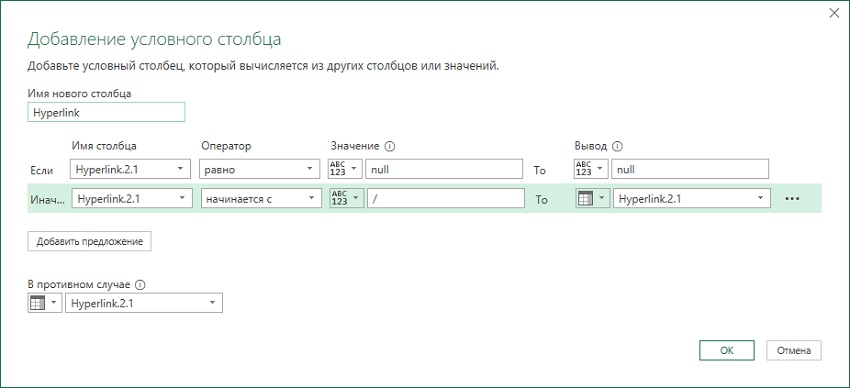


Рис. 7. Настройки условного столбца

Поскольку пользовательский интерфейс ограничен, его иногда применяют для формирования кода, который затем будет отредактирован. Исходный код добавления условного столбца…

= Table.AddColumn(#"Удаленные столбцы", "Hyperlink", each if [Hyperlink.2.1] = null then null else if Text.StartsWith([Hyperlink.2.1], "/") then [Hyperlink.2.1] else [Hyperlink.2.1])

… изменим на

= Table.AddColumn(#"Удаленные столбцы", "Hyperlink", each if [Hyperlink.2.1] = null then null else if Text.StartsWith([Hyperlink.2.1], "/") then "aka.ms" & [Hyperlink.2.1] else [Hyperlink.2.1])

Нулевое условие здесь было добавлено, чтобы не допустить применения функции Text.StartsWith к значению null в поле [Hyperlink.2.1], что приведет к ошибке. Теперь столбец Hyperlink.2.1 можно удалить.

Продолжая проверять результаты в столбце *Hyperlink*, вы увидите, что строка 149 пуста. Но в столбце *Message* есть гиперссылка https://www.microsoftpressstore.com/Ignite. Проблема связана с тем, что на шаге *Разделить столбец по разделителю* в качестве разделителей использовались "https: //" и "www.". Поэтому результат разделения содержал три отдельных значения, но загружались только первые два. Чтобы устранить проблему, отредактируем код. Было:

= Table.SplitColumn(#"Переименованные столбцы", "Hyperlink", Splitter.SplitTextByAnyDelimiter({"http://", "https://", "www.", "aka.ms"}, QuoteStyle.Csv, false), {"Hyperlink.1", "Hyperlink.2"})

Стало:

= Table.SplitColumn(#"Переименованные столбцы", "Hyperlink", Splitter.SplitTextByAnyDelimiter({"http://", "https://", "www.", "aka.ms"}, QuoteStyle.Csv, false), 3)

Функция Table.SplitColumn принимает либо имена новых столбцов при разбиении, либо их число. Теперь, гиперссылки расположены в двух столбцах: *Hyperlink.2* и *Hyperlink.3*.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 8. Разделение столбца *Hyperlink* на три

Объединим эти два столбца. Для этого выделите столбцы *Hyperlink.2,* *Hyperlink.3* и пройдите по меню *Преобразование* –> *Объединить столбцы*. Появится предупреждение о том, что этот шаг будет вставлен между существующими шагами и может нарушить выполнение последующих шагов. Подтвердите действие. Сохраните заданные по умолчанию настройки:

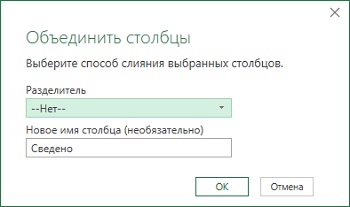


Рис. 9. Параметры объединения столбцов

Удалите столбцы *Hyperlink.2* и *Hyperlink.3.* Переименуйте столбец *Сведено* в *Hyperlink.2*. Каждый раз будет появляться предупреждение о вставке нового шага. Подтверждайте действия. Теперь вы можете продолжить двигаться по шагам, и увидите, что вставка нескольких шагов не нарушила сценарий.

Проблемы остались. В строке 149 гиперссылка закончилась не пробелом, а точкой. В результате она может не открыться. Или если за точкой вплотную (без пробела) следует лишний текст, разделение на следующем шаге с разделителем пробел не сработает. Следует убрать конечные знаки пунктуации во всех строках столбца *Hyperlink.2*. Выделите столбец и пройдите по меню *Преобразование* –> *Формат* –> *Усечь*. Эту команду можно также запустить, кликнув правой кнопкой мыши на столбце. По умолчанию команда *Усечь* удаляет пробелы в начале и конце текстовых значений. Можно ее использовать для очистки знаков пунктуации. Для этого отредактируйте код:

= Table.TransformColumns(#"Переименованные столбцы1",{{"Hyperlink.2", Text.Trim, type text}})

Функция языка M *Text.Trim* имеет два необязательных аргумента. В коде выше она просто удаляет пробелы в начале и в конце текста в каждой строке столбца *Hyperlink.2*. Но мы можем передать функции два аргумента: обрабатываемый текст, и список элементов, которые будут усекаться (помимо пробелов). Обрабатываемый текст передается символом подчеркивания. А символы усечения в виде списка. Новый код:

= Table.TransformColumns(#"Переименованные столбцы1",{{"Hyperlink.2", each Text.Trim(\_, {".",",",")"}), type text}})

Следующая проблема в строке 174, где гиперссылка заканчивается новой строкой, за которой следует лишний текст. Для устранения этой проблемы в сценарии перейдите к шагу Разделить текст по разделителю1. В строке формул вы увидите код:

= Table.SplitColumn(#"Измененный тип1", "Hyperlink.2", Splitter.SplitTextByEachDelimiter({" "}, QuoteStyle.Csv, false), {"Hyperlink.2.1", "Hyperlink.2.2"})

Измените его на:

= Table.SplitColumn(#"Измененный тип1", "Hyperlink.2",Splitter.SplitTextByAnyDelimiter({" ", "#(lf)"}, *QuoteStyle.Csv, false), {"Hyperlink.2.1","Hyperlink.2.2"})*

Значение "#(lf)" описывает специальный символ перевода строки. Для использования нескольких разделителей вы заменили функцию Splitter.SplitTextByEachDelimiter на Splitter.SplitTextByAnyDelimiter.

Наконец, в части строк используются сокращенные ссылок на основе сервиса Bitly, что-то типа: bit.ly/28Cj0Yx. Добавьте в *Разделить столбец по разделителю* в список разделителей еще один:

= Table.SplitColumn(#"Переименованные столбцы", "Hyperlink", Splitter.SplitTextByAnyDelimiter ({"http://", "https://", "www.", "aka.ms", "bit."}, QuoteStyle.Csv, false), 3)

Добавьте в строку Условный столбец доавлен еще одно условие замены:

= if [Hyperlink.2.1] = null then null else if Text.StartsWith([Hyperlink.2.1], "/") then "aka.ms" & [Hyperlink.2.1] else if Text.StartsWith([Hyperlink.2.1], "ly/") then "bit." & [Hyperlink.2.1] else [Hyperlink.2.1]

Удалите столбец *Hyperlink.1*. Переименуйте запрос в *Internet*. Загрузите его в таблицу на лист Excel. Изучите, какие еще проблемы не были решены. Попробуйте обновить код самостоятельно, чтобы сократить число проблем.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 10. Таблица в файле Internet.xlsx (в приложенном архиве)

**Стр. 84**

Работа с датами

Одна из наиболее распространенных проблем подготовки данных — работа с типами данных. Несмотря на то что работать с текстовыми столбцами несложно, числа и даты могут значительно затруднить анализ даже самых простых наборов данных. В настоящем разделе показано, как выполнять обработку дат и времени. Начнем с общей задачи — преобразования текста в даты, а затем перейдем к более сложным случаям, связанным с недействительными датами. В конце этого раздела показано, каким образом можно извлекать определенные элементы даты или времени из значений даты/времени.

При загрузке таблицы со значениями даты или даты/времени Power Query выполняет преобразование соответствующих столбцов с учетом правильного формата даты/времени. В упражнении 2.4 был импортирован набор данных со столбцом Дата/Время (Date/Time). Power Query автоматически преобразует столбец Дата в Дата/Время. В упражнении 2.5 будет показано, каким образом Power Query обрабатывает столбец со смешанными форматами для дат.

УПРАЖНЕНИЕ 2.5.

Обработка нескольких форматов для дат

Загрузите файл C02E05.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C02. Эта рабочая книга содержит каталог товаров AdventureWorks с датой выпуска каждого товара в последнем столбце. Представьте, что разные команды ввода данных указали даты выпуска товаров в пяти различных форматах:

■ 1/7/2018

■ 2018-01-07

■ 1.7.2018

■ Июл 1, 2018

■ 1 июля, 2018

Для простоты предположим, что все команды используют региональную настройку Русский (Россия), поэтому первой датой будет 1/7/2018, т. е. 1 июля (а не 7 января).

При попытках работать с подобными форматами и применять вычисления даты обратите внимание, что Excel не распознает даты во всех ячейках, а для некоторых ячеек возвращает ошибку #ЗНАЧ! (#VALUE!).

1. Откройте копию рабочей книги C02E05.xlsx. (Удостоверьтесь, что применяется не исходная книга, а ее копия.)

2. В ячейку H2 введите следующую формулу:

= G2 + 1

Эта формула должна увеличивать дату выпуска на один день, в результате получим 43283. Дело в том, что Excel сохраняет даты в виде последовательных номеров, чтобы их можно было использовать при вычислениях. По умолчанию 1 января 1900 года — последовательный номер 1, а 1 июля 2018 года — последовательный номер 43282 (число дней начиная с 01.01.1900).

3. Для корректного просмотра дат измените формат столбца H на вкладке Дата (Date). После этого скопируйте и вставьте формулу из шага 2 в следующие четыре ячейки: H3, H4, H5 и H6.

В ячейках H4 и H6 возвращается ошибочное значение #ЗНАЧ!, поскольку Excel не может найти способ преобразования в даты или числа для увеличения их на единицу.

4. Теперь пришло время узнать, как Power Query может помочь нам в этом сценарии, поэтому откройте новую книгу или запустите Power BI Desktop. Поскольку входные данные основаны на региональных настройках русского языка (Россия), необходимо убедиться, что в рассматриваемом упражнении установлен этот языковой стандарт.

В Excel: поскольку мы продолжаем находиться в редакторе Power Query, выберем команду Файл (File). Затем выберите команду Параметры и настройки (Options and Settings) и далее — команду Параметры запроса (Query Options). В разделе Текущая книга (Current Workbook) выберите команду Региональные настройки (Regional Settings) и затем из раскрывающегося меню Локаль (Locale) выберите опцию Русский (Россия).

В Power BI Desktop: выполните последовательность команд Файл (File) Па-раметры и настройки (Options and Settings) Параметры (Options). В разделе Текущий файл (Current File) выберите команду Региональные настройки (Regional Settings), а затем в раскрывающемся меню Локаль (Locale) выберите опцию Русский (Россия).

5. Щелкните на кнопке OK для закрытия окна диалога Параметры (Options).

6. Импортируйте книгу C02E05.xlsx в редактор Power Query:

а. В Excel: на вкладке Данные (Data) выполните последовательность команд Получить данные (Get Data) Из файла (From File) Из книги (From Workbook). В Power BI Desktop: в раскрывающемся меню Получить данные (Get Data) выберите Excel.

б. Выберите файл C02E05.xlsx и нажмите кнопку Импорт (Import).

в. В окне диалога Навигатор (Navigator) выберите лист Sheet1 и затем щелкните мышью на кнопке Изменить (Edit).

7. После открытия окна редактора Power Query заметим, что все значения дат в столбце Release Date (Дата выпуска) отображаются корректно, как показано на рис. 2.11.

8. Закройте редактор Power Query и загрузите данные в отчет.

Рис. 2.11. Power Query автоматически изменяет тип столбца из Текст на Дата и корректно конвертирует различные форматы дат

СОВЕТ

Если вы импортируете даты, локальные настройки которых отличаются от настроек, определенных для вашей операционной системы Microsoft Windows, определите кон-кретную настройку в окне диалога Параметры запроса, как показано на шаге 4. После обновления предварительного просмотра в редакторе Power Query можно убе-

диться, что даты распознаны правильно.

УПРАЖНЕНИЕ 2.6. Работа с датами

при использовании двух локальных настроек

При выполнении упражнения 2.5 использовалось несколько форматов дат, но предполагалось, что для всех дат выбрана одна локальная настройка. В данном упражнении показано, каким образом обрабатываются случаи, когда даты относятся к нескольким локальным настройкам, причем значения месяцев и дней следует для некоторых строк поменять местами.

Загрузите файл C02E06.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C02. Для простоты книга содержит только две строки из каталога товаров AdventureWorks — каждая строка для товара с указанием страны и датой выпуска, которая отформатирована с учетом локальной настройки конкретной страны:

■ Country (Страна): USA (США), Release Date (Дата выпуска): 7/23/2018.

■ Country (Страна): United Kingdom (Великобритания), Release Date (Дата выпуска): 23/7/2018.

Как можно видеть на рис. 2.12, столбец Data Release при изменении типа Дата возвращает сообщение об ошибке для строки UK, если в качестве локального языка в окне диалога Параметры запроса выбран Английский (США); возвращается значение ошибки для строки USA, если в этом окне диалога для локального языка выбран Английский (Великобритания).

Для решения этой проблемы не следует полагаться на локальный язык в окне диалога Параметры (Settings). Вместо этого можно применить команду Разделить столбец по разделителю (Split Column by Delimiter) и затем создать условный столбец, как описано в этом упражнении.

1. Создайте новую пустую книгу Excel или новый отчет Power BI Desktop.

2. Импортируйте файл C02E06.xlsx в редактор Power Query:

а. В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook). В Power BI Desktop: в раскрывающемся меню Получить данные (Get Data) выберите команду Excel.

б. Выберите файл C02E06.xlsx и нажмите кнопку Импорт (Import).

в. В окне диалога Навигатор (Navigator) выберите вкладку Products (Товары), затем щелкните мышью на кнопке Изменить (Edit).

3. Выберите столбец Data Release (Дата выпуска). На вкладке Преобразование выберите команды Разделить столбец (Split Column) По разделителю (By Delimiter). В появившемся окне диалога Разделить столбец по разделителю (Split Column bu Delimiter) в качестве разделителя уже выбрана косая черта (/). Щелкните мышью на кнопке ОК для закрытия окна диалога.

Как видите, теперь столбец Data Release разделен на три столбца: Data Release.1, Data Release.2 и Data Release.3. Если выбранная страна — United Kingsdom (Великобритания), то Data Release.1 — это день, а Data Release.2 — месяц. Если выбранная страна — USA (США), то Data Release.1 — это месяц, а Data Release.2 — день. На следующем шаге показано, как создается новый столбец с корректной датой, при помощи формулы M и пользовательского столбца. Затем, на шагах 6-10, продемонстрирован альтернативный способ, когда не требуется знание языка M, но предполагается наличие взаимодействия с пользовательским интерфейсом редактора Power Query.

4. На вкладке Добавление столбца (Add Column) выберите команду Настраивае-мый столбец (Custom Column). Откроется окно диалога Настраиваемый столбец.

а. Введите название Date в поле Имя нового столбца (New Column Name).

б. В поле Пользовательская формула столбца (Custom Column Formula) введите следующую формулу:

if [Country] = "US" then #date([Date Release.3],

[Date Release.1], [Dare Release.2]) else #date([Date Release.3],

[Date Release.2], [Date Release.1])

в. Затем щелкните мышью на кнопке OK для закрытия окна диалога.

г. Измените тип столбца Date на Дата.

д. Удалите столбцы Date Release.1, Data Release.2 и Date Release.3.

ПРИМЕЧАНИЕ

На шагах 6-10 продемонстрирован альтернативный способ создания даты. Прежде чем продолжить, необходимо продублировать текущий запрос для сохранения обеих версий.

5. Для дублирования запроса Products (Товары) правой кнопкой мыши щелкните на названии запроса Products на панели Запросы (Queries) и в контекстном ме-ню выберите команду Дублировать (Duplicate). Это приведет к созданию ново-го запроса, Products (2). Выберите его на панели Примененные шаги и удалите последние три шага (включая шаг Добавлен пользовательский).

6. С помощью клавиши <Ctrl> выделите три разделенных столбца дат в следую-щем порядке: Date Release.1, Date Release.2 и Date Release.3. Теперь, на вкладке Добавление столбца выберите команду Объединить столбцы (Merge Columns). Откроется окно диалога Объединить столбцы (Merge Columns).

а. В качестве разделителя задайте Пользовательский (Custom) и в находящееся ниже поле введите косую черту (/).

б. В поле Имя нового столбца (New Column Name) введите US Date и щелкните мышью на кнопке OK для закрытия окна диалога.

7. Теперь создайте столбец дат для локальной настройки UK. С помощью клавиши <Ctrl> выберите три разделенных столбца дат в следующем порядке: Date Release.1, Date Release.2 и Date Release.3. Теперь на вкладке Добавление столбца (Add Column) выберите команду Объединить столбцы (Merge Columns). Откроется окно диалога Объединить столбцы (Merge Columns).

а. В качестве разделителя задайте Пользовательский (Custom) и в находящееся ниже поле введите косую черту (/).

б. В поле New Column Name введите UK Date и щелкните мышью на кнопке OK для закрытия окна диалога.

Теперь имеются столбцы US Date и UK Date с разными форматами дат. Пришло время выбрать определенный формат даты для каждой строки.

8. На вкладке Добавление столбца (Add Column) выберите команду Условный столбец (Conditional Column). Откроется окно диалога Добавление условного столбца (Add Conditional Column). Чтобы изменить настройки в этом окне, выполните следующие действия (рис. 2.13):

а. В поле New Column Name (Имя нового столбца) введите Date.

б. В раскрывающемся меню Имя столбца (Column Name) выберите пункт Country (Страна).

в. Выберите опцию Равно (Equals) в качестве оператора и введите US в поле Значение (Value).

г. В раскрывающемся меню Вывод (Output) выберите команду Выбрать стол-бец (Select a Column), затем выберите опцию US Date.

д. В раскрывающемся меню, которое находится ниже метки В противном слу-чае (Otherwise), выберите команду Выбрать столбец (Select a Column) и за-тем выберите опцию UK Date.

е. Удостоверьтесь, что введенные настройки соответствуют рис. 2.13, и щелкните мышью на кнопке OK для закрытия окна диалога Добавление условного столбца.

X

Добавление условного столбца

Добавьте условный столбец, который вычисляется из других столбцов или значений.

Имя нового столбца

| Date

Имя столбца Оператор Значение ф Вывод ф

Если | Country ’ | | равно “ЖЗП us 1 То |из ’ I US Date | —

Добавить правило

В противном случае ф

ИЗ ’ UK Date 1

Отмена

Рис. 2.13. Для выбора корректного формата даты можно добавлять условный столбец

9. Измените тип столбца Date на Дата.

10. Удалите столбцы Date Release.1, Date Release.2, Date Release.3, US Date и UK Date.

11. Загрузите запрос в рабочую книгу Excel или отчет Power BI. Даты отображаются в соответствии с локальными настройками Windows, и ошибки отсутствуют.

Можно загрузить файлы решений C02E06 - Solution.xlsx и C02E06 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

Выборка элементов даты и времени

Итак, в запросах появились столбцы типа Дата и Дата/Время, их формат не вызывает нареканий. Каким образом можно извлекать из них значимые элементы? Редактор Power Query предлагает широкий спектр функций, а также преобразований, находящихся на вкладках Преобразование (Transform)u Добавление столбца (Add Column), которые позволяют извлекать годы, кварталы, месяцы, дни и многие другие вычисляемые элементы, обогащающие набор данных при анализе временных промежутков.

Далее приводятся некоторые полезные преобразования, доступные на вкладках Преобразование и Добавление столбца в раскрывающихся меню Дата (Date) при выборе типа столбца Дата (Date) или Дата/Время (Date/Time):

■ Возраст (Age) — данное преобразование присваивается выделенному столбцу типа Дата или Дата/Время для вычисления времени, прошедшего после опре-деленной даты. Реализуется вычитанием текущей даты/времени из определенного значения.

■ Только дата (Date Only) — извлекает сведения о дате из выбранного столбца типа Дата/Время. Вместе с преобразованием Только время (Time Only) можно разделить столбец Дата/Время на два отдельных столбца, типов Дата и Время, что упрощает модель, уменьшает размер отчета и сокращает объем требуемой для него памяти.

■ Год (Year) — извлекает из столбца типа Дата или Дата/Время часть, относя-щуюся к году.

■ Начало года (Start of Yea^/Конец года (End of Year) — извлекает первый или последний день года как значение типа Дата.

■ Месяц (Month) — извлекает часть, относящуюся к месяцу из столбца типа Дата или Дата/Время.

■ Начало месяца (Start of Month)/Конец месяца (End of Month) — извлекает первый или последний день месяца как значение типа Дата.

■ Дней в месяце (Days in Month) — вычисляет число дней в текущем месяце вы-бранного столбца типа Дата.

■ Название месяца (Name of Month) — возвращает название месяца.

■ Квартал года (Quarter of Year) — возвращает квартал текущего года.

■ Начало квартала (Start of Quarter)/Конец квартала (End of Quarter) — извлекает первый или последний день квартала как значение типа Дата.

■ Неделя года (Week of Yea^/Неделя месяца (Week of Month) — вычисляет номер недели в году или месяце.

■ Начало недели (Start of Week)/Конец недели (End of Week) — извлекает первый или последний день недели как значение типа Дата.

■ День (Day) — извлекает часть дня из столбца типа Дата или Дата/Время.

■ Начало дня (Start of Day)/Конец дня (End of Day) — извлекает первую или последнюю дату и время дня из столбца типа Дата/Время.

■ День недели (Day of Week)/День года (Day of Year) — вычисляет номер текущего дня недели или года.

■ Название дня (Name of Day) — возвращает название дня (например, Воскресе-нье, Понедельник).

При выборе нескольких столбцов типа Дата или Дата/Время также можно при-менять команду Вычесть дни (Subtract Days) для расчета времени, прошедшего между двумя датами, или вычислять самые ранние или самые поздние даты.

Подготовка модели данных

Подготовка данных служит ключом к успеху их анализа. Для подготовки данных, предназначенных для эффективного анализа, чаще всего необходимо делить таблицы на несколько таблиц, тогда формируется одна таблица для данных или для транзакций (например, Sales Order) и дополнительные таблицы для поддержки (например, Products, Clients).

В упражнении 2.2, часть 3, было показано, каким образом следует создавать связи между таблицами данных и справочными таблицами. И хотя элементы моделирования в Power Pivot и Power BI не рассматриваются в данной книге, остановимся на вопросе формирования таблиц в соответствии с конкретными потребностями. В следующих трех главах показано, как можно объединять несколько таблиц, что позволит упростить модель. В главе 6 будет показано, каким образом можно разбить сложные структуры таблиц для более успешного выполнения анализа.

В данном разделе пойдет речь об обязательных подготовительных шагах, которые позволят разбивать агрегированную таблицу на несколько таблиц, строить звездообразные схемы и управлять гранулярностью объектов.

УПРАЖНЕНИЕ 2.7. Разделение данных на таблицы поиска и таблицы фактов

В этом упражнении выполним импорт данных из другой фиктивной компании: Wide World Importers. Этот набор данных расширен и обновлен по сравнению с набором данных AdventureWorks, набор также предоставлен корпорацией Майкрософт для учебных целей. Чтобы узнать больше об этом наборе данных, перейдите на сайт https://blogs.technet.microsoft.com/dataplatforminsider/2016/06/09/ wideworldimporters-the-new-sql-server-sample-database/.

Файл рабочей книги C02E07.xlsx можно загрузить с сайта https://aka.ms/ DataPwrBIPivot/downloads. Эта книга суммирует заказы Wide World Importer. Цель упражнения — показать, как применять Power Query для разделения таблицы на таблицу фактов и таблицу поиска с помощью команд Ссылка (Reference) и Удалить дубликаты (Remove Duplicate).

Загрузите файл C02E07.xslx и сохраните его в папке C:\Data\C02\.

1. Откройте новую пустую рабочую книгу Excel или новый отчет Power BI Desktop и импортируйте рабочую книгу C02E07.xslx в редактор Power Query.

2. В окне диалога Навигатор (Navigator) выберите таблицу Sales\_Order, затем щелкните мышью на кнопке Изменить (Edit).

3. Переименуйте запрос Sales\_Order в Sales Order - Base.

Ваша цель — разделить исходную таблицу на две таблицы с корректными уровнями детализации, как показано на рис. 2.14. Одна из таблиц (нижняя, слева, на рис. 2.14) служит для заказов и идентификаторов товарных позиций (таблица фактов), а вторая (нижняя, справа, на рис. 2.14) — для товарных позиций (таблица поиска).

4. Правой кнопкой мыши щелкните на запросе Sales Order - Base и в контекстном меню выберите команду Ссылка. Переименуйте новый запрос на Stock Items. (Для переименования запроса правой кнопки мыши щелкните на запросе в панели Запросы (Queries) и в контекстном меню выберите команду Переименовать

(Rename), или переименуйте запрос на панели Параметры запроса (Query Settings), выбрав раздел Свойства и поле Имя.)

5. Выделите запрос Stock Items, и на вкладке Главная выберите команду Выбор столбцов (Choose Columns).

6. В открывшемся окне диалога Выбор столбцов (Choose Columns) отмените вы-деление всех столбцов и затем выберите столбцы Stock ID (Код позиции на складе), Stock Item (Позиция на складе) и Stock Lead Time (Время нахождения позиции на складе). Щелкните мышью на кнопке OK для закрытия этого окна диалога.

7. Выберите столбец Stock ID, и на вкладке Главная выполните последователь-ность команд Удалить строки (Remove Rows) Удалить дубликаты (Remove Duplicates).

Теперь имеется таблица поиска для товаров со склада (позиций на складе), где в каждой строке приведены уникальные позиции на складе (позиции для товаров со склада).

ПРИМЕЧАНИЕ

Если имеется несколько значений Stock ID в таблице позиций на складе, то создать связь на шаге 13 не удастся. В упражнении сознательно опущены два важных шага, объяснение которых приведено в конце раздела. Без этих шагов команды Удалить дубликаты (Remove Duplicates) недостаточно для гарантии того, что таблица поиска будет включать только уникальные значения Stock ID, и в результате обновление может не реализоваться.

8. Для создания новой таблицы фактов для заказов на панели Запросы щелкните правой кнопкой мыши на запросе Sales Order - Base и в контекстном меню вы-берите команду Ссылка (Reference).

9. Переименуйте новый запрос в Sales Orders. (Для переименования запроса пра-вой кнопки мыши щелкните на запросе в панели Запросы и в контекстном меню выберите команду Переименовать (Rename) или переименуйте запрос на панели Параметры запроса, перейдя в раздел Свойства (Properties) и выбрав поле Имя (Name).)

10. Выберите запрос Sales Orders и на вкладке Главная выберите команду Выбор столбцов (Choose Columns). После этого появится окно диалога Выбор столб-цов (Choose Columns). Отмените выделение столбцов Stock Item (Позиция на складе) и Stock Lead Time (Время нахождения позиции на складе). Щелкните мышью на кнопке OK, чтобы закрыть это окно диалога.

11. В Excel: на вкладке Главная выберите команду Закрыть и загрузить в (Close & Load To (Закрыть и загрузить в), что позволит избежать загрузки базового запроса на рабочий лист или в модель данных. Затем на панели Запросы и подключения (Queries and Connections) выберите запросы Sales Orders и Stock Items и загрузите их в модель данных с помощью команды Загрузить в (Load To), как это было выполнено в упражнении 2.2, часть 2 на шаге 11.

В Power BI Desktop: отключите загрузку отчета Sales Order - Base, как это было сделано в упражнении 2.2, часть 2 на шаге 11.

12. Создайте связь между двумя таблицами, используя столбцы Stock ID.

Когда связи некорректны

При формировании таблицы поиска в Power Query, как это сделано в упражне-нии 2.7, и применении текстовых столбцов в качестве ключей для связей необходимо выполнить два дополнительных шага, прежде чем удалять продублированные ключи на шаге 8, что позволит убедиться, что отчет будет удачно обновлен.

В упражнении 2.7, на шаге 8, из таблицы поиска удалены повторяющиеся значения столбца Stock ID. К сожалению, Power Query базируется на принципах, отличающихся от принципов, на которых основана модель данных в Excel и Power BI. Power Query всегда чувствителен к регистру символов, а модель данных — нет. В результате, если значение Stock ID, "Stock\_1", включает вариант, записанный с использованием нижнего или верхнего регистров (т. е. "STOCK\_1") или включает завершающий пробел (например, "Stock\_1 "), то Power Query сохранит оба экземпляра как уникальные значения, а модель данных рассмотрит их как дубликаты и выдаст такое сообщение об ошибке:

Column 'Stock ID' in Table 'Stock Items' contains a duplicate value 'Stock\_1' and this is not allowed for columns on the one side of a many-to-one relationship or for columns that are used as the primary key of a table.

(Столбец 'Stock ID' в таблице 'Stock Items' содержит повторяющееся значение 'Stock\_1', и это не допускается для столбцов на одной стороне отношения "многие к одному" или для столбцов, которые используются в качестве первичного ключа таблицы.)

Чтобы увидеть эту ошибку, откройте книгу C02E07.xslx и измените два значения в столбце Stock ID.

Измените одно из значений на верхний регистр и добавьте символ пробела в конце другого значения. Затем сохраните книгу и попробуйте обновить книгу решения для упражнения 2.7 или отчета Power BI.

Чтобы устранить эту проблему, выполните следующие действия для запроса Sales Orders - Base:

1. Откройте редактор Power Query и выберите запрос Sales Orders - Base.

2. Выделите столбец Stock ID, на вкладке Преобразование (Transform) задайте команду Формат (Format), затем выберите команду нижний регистр (Lowercase), ВЕРХНИЙ РЕГИСТР (Uppercase) или Каждое Слово С Прописной (Capitalize Each Word). Любая из этих трех команд гарантирует, что ключи будут нормализованы, поэтому с помощью Power Query варианты значений в нижнем и верхнем регистре удаляются как повторяющиеся значения.

3. Выделите столбец Stock ID, на вкладке Преобразование (Transform) выберите сначала команду Формат (Format), а затем — команду Усечь (Trim).

4. Обновите данную рабочую книгу или отчет, ошибки больше не будет.

Формирование ссылок и удаление дубликатов для создания таблиц поиска обычно являются часто встречающимися операциями. Многие пользователи, применяющие шаг Удалить дубликаты (Remove Duplicates), не знают о возможных сбоях при обновлении, которые происходят в наборах данных, содержащих строчные и прописные варианты написания или конечные пробелы. Часто ошибки проявляются и раньше, поскольку пользователи пытаются формировать отношения. Обработка чувствительных к регистру значений и устранение пробелов перед удалением дубликатов помогают разрешить эти проблемы.

Загрузите файлы решений C02E07 - Solution.xlsx и C02E07 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

См. также

Описание ошибки, связанной с применением этих операций для предотвращения сбоев обновления отношений, приведено в моем блоге https://datachant.com/2017/02/14/ power-bi-pitfall-10/. В главе 10 описаны наиболее распространенные ошибки, которые приводят к серьезным сбоям в отчетах.

УПРАЖНЕНИЕ 2.8. Разбиение на строки значений с разделителями

В предыдущем разделе показано, как можно разбивать таблицы на таблицы фактов и таблицы поиска. В данном упражнении описан особый случай таблиц фактов и поиска. Представьте себе, что таблица фактов действует как таблица сопоставления, которая связывает сущности участников и уникальные идентификаторы их групп. В этом сценарии справочная таблица описывает группы объектов.

В данном упражнении снова применяется таблица Products (Товары) из базы данных AdventureWorks. Каждый товар может иметь один или несколько цветов. Соответствующие цвета представлены в виде значений, разделенных запятыми, как видно из таблицы слева на рис. 2.15. В упражнении показано, как можно разбить таблицу Products на две таблицы: одну — для сущностей (в данном случае для сущностей товаров без цветов), а другую — для связей между группами и их участниками (в данном случае кодов товаров и цветов).

Для данного упражнения применяется пример книги C02E08.xlsx, которую можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads. В рабочей книге приведены коды товаров AdventureWorks по средней стоимости, средней цене и цветам, разделенными запятыми. Для формирования отчета, показывающего количество продуктов каждого цвета, как показано на рис. 2.15, выполните следующие действия.

1. Загрузите файл C02E08.xslx и сохраните его в папке C:\Data\C02\.

2. Откройте новую пустую книгу Excel или новый отчет Power BI Desktop и им-портируйте книгу C02E08.xslx из C:\Data\C02\ в редактор Power Query.

3. В окне диалога Навигатор (Navigator) выберите вкладку Products (Товары), за-тем нажмите кнопку Изменить (Edit).

4. На панели Запросы (Queries) щелкните правой кнопкой мыши на вкладке Products и выберите команду Ссылка (Reference). Ваша цель — создать новую таблицу с отображением между кодами товаров и цветами.

5. Переименуйте новый запрос в Products and Colors. (Для переименования запроса правой кнопкой мыши щелкните на запросе в панели Запросы (Queries) и в контекстном меню выберите команду Переименовать (Rename) или переименуйте запрос на панели Настройки запроса (Query Settings), перейдя в раздел Свойства (Properties) и выбрав поле Имя (Name).)

6. При выделенном новом запросе, на вкладке Главная выполните команду Вы-бор столбцов (Choose Columns).

7. В открывшемся окне диалога Выбор столбцов (Choose Columns) выберите столбцы ProductCodeNew и Colors и щелкните мышью на кнопке OK для закрытия окна диалога.

8. Выделите столбец Colors (Цвета), на вкладке Преобразование (Transform) щелкните мышью на кнопке Разделить столбец (Split Column) и затем выпол-ните команду По разделителю (By Delimiter).

9. В появившемся окне диалога Разделить столбец по разделителю (Split Column by Delimiter) обратите внимание, что по умолчанию выбран разделитель в виде запятой и установлен переключатель По каждому вхождению разделителя (Each Occurence of the Delimiter), как показано на рис. 2.16. Разверните раздел Расширенные параметры (Advanced Options) и установите переключатель Строки (Rows) в разделе Разделить на (Split into). Щелкните мышью на кнопке OK для закрытия этого окна диалога.

Рис. 2.16. В окне диалога Разделить столбец по разделителю в разделе Расширенные параметры не забудьте установить переключатель Строки в разделе Разделение на

10. В запросе Colors and Products (Цвета и товары) выберите столбец Colors (Цве-та). На вкладке Преобразование (Transform) выберите команду Формат (Format), затем выберите команду Усечь (Trim). Этот шаг приводит к удалению ведущих пробелов из значений для цвета.

Теперь можно загрузить две таблицы в модель данных в Excel или Power BI Desktop и создать связь между полем ProductCodeNew из таблицы Products и ProductCodeNew в таблице Products and Colors. Загрузите и просмотрите файлы решения C02E08 - Solution.xlsx и C02E08 - Solution.pbix с сайта https://aka.ms/ DataPwrBIPivot/downloads.

Г") СОВЕТ

Для уменьшения размеров отчета удалите столбец Colors из заключительной таблицы Products. Однако, если попытаться реализовать это сейчас, возможны ошибки, поскольку на запрос Products ссылается запрос Products and Colors. Для устранения этой проблемы и получения возможности удалить столбец Colors из таблицы Products необходимо создать новую ссылку из таблицы Products и убрать столбец Colors из ссылочного запроса. Затем убедитесь в том, что исходный запрос Products применяется только как базовый запрос и не будет загружаться в модель данных или отчет. (В упражнении 2.2, часть 2, показано, каким образом можно отключить загрузку отчета.)

Резюме

Подготовка данных может представлять собой довольно трудоемкий процесс. Благодаря обширным возможностям по выполнению преобразований, которые доступны в редакторе Power Query, можно очистить и подготовить любой набор данных для анализа. В этой главе описаны общие проблемы с данными, а также методики преодоления этих проблем, предоставляемые Power Query.

В упражнении 2.2 было показано, каким образом Power Query извлекает коды и выполняет связывание значений с кодами, причем без применения довольно сложных формул, приведенных в упражнении 2.1.

В упражнении 2.3 проиллюстрирована весьма полезная возможность по выборке значений из столбцов и приведены примеры. Показано, что столбец из примеров позволяет извлечь текст, дату, время и числа, и предложены различные преобразования на основе предоставленных исходных примеров. Показано, каким образом данную функцию можно применить для извлечения текста между разделителями, продемонстрировано применение условного столбца и формирование сегментов из диапазонов чисел.

В упражнении 2.4 показано, каким образом можно проанализировать гиперссылки в текстовых данных и применять различные методы преобразования для достижения поставленных целей. Одним из наиболее важных уроков этого упражнения служит тот факт, что необходимо почаще проверять запросы и исследовать крайние их значения.

Распространенная и довольно трудоемкая задача — обработка несовместимых дат в наборах данных.

В упражнениях 2.5 и 2.6 показано, насколько просто можно работать в Power Query с несовместимыми датами. Наконец, в упражнениях 2.7 и 2.8 продемонстрировано, каким образом выполняется разбиение таблицы на таблицу фактов и таблицу поиска с помощью команд Ссылка и Удалить дубликаты. Также показано, каким образом можно избежать ошибок обновления, применяя преобразования нижнего регистра и усечения, а также путем разбиения заданных через запятую значений на строки для формирования таблиц групп/участников.

Даже если вы только начали вникать в содержание этой книги, на основе этих упражнений, пользуясь ими, уже сможете упорядочить и уменьшить свои усилия по очистке данных, добиваясь существенной экономии рабочего времени. В следующей главе показано, каким образом можно объединить несколько таблиц, что позволит сэкономить время на их исследование.

Глава 3

Объединение данных, полученных из нескольких источников

Попытка объединить мудрость и силу редко бывает успешной, да и то ненадолго.

- Альберт Эйнштейн

В ЭТОЙ ГЛАВЕ ВЫ...

■ Освоите способы добавления таблиц, полученных из нескольких выбранных источников данных.

■ Узнаете о том, как добавлять две и более книги Excel из одной папки.

■ И научитесь добавлять несколько листов из книги Excel.

И хотя с цитатой Альберта Эйнштейна трудно не согласиться, в Power Query все немного не так. Благодаря объединению нескольких источников данных можно получить ценную информацию, требуемую для принятия важных решений. В этой главе будет показано, каким образом можно объединять несколько таблиц, имеющих различные уровни сложности, тем самым получая представление о комбинированных наборах данных.

В главе 2 были рассмотрены различные варианты обработки неудовлетворительно отформатированных и плохо подготовленных данных. Нередко обрабатываемые данные хорошо отформатированы, но при этом поступают из нескольких источников. Поэтому эти данные приходится объединять. Подобный процесс, реализуемый путем добавления нескольких таблиц, рассматривается в этой главе.

Глава начинается с рассмотрения базовых сценариев объединения небольшого числа таблиц, имеющих общий формат. Затем мы перейдем к решению реальных задач, которые характеризуются постоянно изменяющимися источниками данных.

Объединение нескольких таблиц

Одна из наиболее популярных команд Power Query, Объединить запросы (Append Queries), доступна на вкладке Главная (Home) редактора Power Query. Эта команда позволяет добавлять предварительно определенный набор таблиц. Сначала рассмотрим объединение двух таблиц, имеющих одинаковый формат (т. е. таблиц с одинаковыми названиями столбцов).

Объединение двух таблиц

Прежде чем перейти к объединению двух таблиц, подумаем над следующим вопросом: почему при выполнении анализа нужно объединять таблицы, а не разделять их? И еще вопрос: почему бы не скопировать и вставить две таблицы в новую книгу Excel, каждая из которых займет там отдельный лист, а третий лист использовать для выполнения необходимых вычислений? И хотя такой подход оптимален для простых таблиц с несколькими строками и столбцами, выполнение подобных вычислений для больших и постоянно изменяющихся таблиц существенно усложняется.

Указанную проблему можно решить в Excel без применения Power Query. Для этого нужно скопировать и вставить таблицы, а затем объединить их вручную. При каждом изменении данных придется снова копировать и вставлять таблицу. В качестве альтернативы, можно выполнять сложные вычисления, суммирующие диапазоны из различных листов. Всякий раз при изменении размеров наборов данных будут обновляться диапазоны в ваших формулах. К счастью, для автоматизации этого процесса можно обратиться к Power Query. В этом случае при появлении в таблицах новых данных Power Query сделает за вас всю работу, автоматически обновив всю информацию.

УПРАЖНЕНИЕ 3.1.

Пример c таблицами Bikes и Accessories

В этом упражнении мы будем работать с двумя таблицами, относящимися к вы-мышленной компании по производству велосипедов AdventureWorks, представленной в главе 2. Представьте себе, что компания AdventureWorks, которая производит велосипеды, приобретает новую компанию, выпускающую аксессуары для велосипедов. И вам, как главному аналитику компании AdventureWorks, нужно создать новый отчет, выводящий сведения о велосипедах и аксессуарах к ним (рис. 3.1).

Чтобы объединить таблицы Bikes и Accessories, начните с создания новой книги Excel или отчета Power BI Desktop, куда будут импортированы упомянутые две таблицы. Процесс импорта реализуется путем выполнения двух отдельных запросов. Для выполнения этого упражнения загрузите файлы C03E01 - Bikes.xlsx и C03E01 - Accessories.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads.

1. Создайте новую пустую книгу Excel или Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните последовательность команд По-лучить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: выполните команды Получить данные (Get Data) Excel Подключиться (Connect).

2. Выделите файл C03E01 - Bikes.xlsx и нажмите кнопку Импорт (Import).

3. В окне диалога Навигатор (Navigator) выберите лист Bikes, затем нажмите кнопку Изменить (Edit).

4. В окне редактора Power Query выполните команду Закрыть и загрузить в (Close & Load To). В Power BI Desktop выполните команду Закрыть и приме-нить (Close & Apply) и пропустите шаг 5.

5. В окне диалога Импорт данных (Import Data) установите переключатель Таб-лица (Table) и щелкните мышью на кнопке OK. Теперь все данные из таблицы Bikes будут импортированы в новую книгу или в Power BI Desktop. Если же данные в таблице C03E01 - Bikes.xlsx изменятся и в список товаров будут добавлены новые велосипеды, то для получения самых свежих данных нужно щелкнуть мышью на кнопке Обновить все (Refresh All). Эта кнопка находится на вкладке Excel Данные (Data). Аналогичный результат даст комбинация клавиш <Ctrl>+<Alt>+<F5>. В Power BI Desktop можно просто нажать кнопку Обновить (Refresh), находящуюся на вкладке Главная (Home).

Теперь можно импортировать таблицу Accessories в книгу или в отчет Power BI.

6. В Excel: на вкладке Данные (Data) выполните последовательность команд По-лучить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: на вкладке Главная выберите команды Получить данные (Get Data) Excel Подключиться (Connect).

7. Выберите файл C03E01 - Accessories.xlsx и нажмите кнопку Импорт (Import).

8. В окне диалога Навигатор выберите таблицу Accessories и щелкните мышью на кнопке Изменить.

9. Чтобы объединить запросы Bikes и Accessories в окне редактора Power Query, убедитесь, что выбран запрос Accessories, и на вкладке Главная щелкните на кнопке Объединить запросы (Append Queries). Обратите внимание, что раскрывающееся меню Объединить запросы также содержит опцию Объединить запросы в новый (Append Queries as New), которая будет описана далее в упражнении 3.2.

10. После открытия окна диалога Добавление (Append) выберите добавляемую таблицу Bikes и щелкните мышью на кнопке OK.

11. В Excel: откройте панель Запросы (Queries) (эта панель находится в левой части окна Power Query).

В Power BI Desktop: удостоверьтесь, что панель Запросы открыта (эта панель находится в левой части окна редактора Power Query).

Обратите внимание, что на панели Запросы находятся два запроса: Bikes и Accessories. И поскольку запрос Bikes добавлен к запросу Accessories, его имя не соответствует сути добавленных данных. Поэтому следует переименовать запрос Accessories и убедиться, что именно этот запрос, а не запрос Bikes будет загружен в книгу Excel или отчет Power BI. Чтобы переименовать запрос, щелкните правой кнопкой мыши на запросе Accessories, выберите параметр Переименовать (Rename) и введите имя Products. Также можно переименовать запрос в разделе Свойства (Properties) на панели Параметры запроса (Query Settings).

12. Чтобы выгрузить запрос Bikes, выполните следующие действия:

В Excel:

а. На вкладке Главная (Home) выберите команду Закрыть и загрузить в (Close & Load to).

б. В окне диалога Импорт данных (Import Data) установите переключатель Таблица (Table) и щелкните мышью на кнопке OK.

в. Для выгрузки таблицы Bikes из рабочей книги выберите панель Запросы и подключения (Queries & Connections), щелкните правой кнопкой мыши на запросе Bikes и в контекстном меню выберите команду Загрузить в (Load To).

г. В окне диалога Импорт данных установите переключатель Только создать подключение (Only Create Connection) и щелкните мышью на кнопке OK.

д. После появления окна диалога Возможна потеря данных (Possible Data Loss) щелкните мышью на кнопке OK для выгрузки данного запроса и удаления таблицы.

е. И наконец, удалите лист Sheet2, на который предварительно загружалась таблица Bikes, и переименуйте последний лист на Products.

В Power BI Desktop: выберите панель Запросы, правой кнопки мыши щелкните на запросе Bikes, отмените установку флажка Включить загрузку (Enable Load) и нажмите кнопку Закрыть и применить (Close & Apply).

Теперь в вашем распоряжении имеется общая таблица Products, которая включает объединенные данные из таблиц Bikes и Accessories. Но если вы присмотритесь к панели Запросы, то заметите асимметричный характер, присущий шагу добавления. На панели находятся запрос Bikes и комбинированный запрос Products. Но куда же делся запрос Accessories? Напомним, что сразу после завершения импорта запроса Accessories вы добавили к нему запрос Bikes и переименовали его в Products. В будущем, при необходимости внести в таблицу Accessories некоторые коррективы, вы столкнетесь с трудностями при выборе корректного шага для изменения, поскольку имеется запрос с комбинированными шагами преобразования Accessories и добавленной таблицей Products. В такой ситуации лучше использовать Accessories в качестве отдельного запроса, и поэтому преобразование Объединить запросы включает похожее преобразование под названием Объединить запросы в новый.

УПРАЖНЕНИЕ 3.2. Часть 1. Применение команды Объединить запросы в новый

Благодаря вкладке Главная (Home) редактора Power Query можно добавлять не-сколько запросов в раздел Комбинировать (Combine). При этом используются два способа: команда Объединить запросы (Append Queries) и Объединить запросы в новый (Append Queries as New), рис. 3.2.

Рис. 3.2. Преобразование Объединить запросы в новый позволяет формировать отдельные запросы для добавляемой таблицы и ее частей

При выполнении предыдущего упражнения были объединены две таблицы. При этом с помощью преобразования Объединить запросы (Append Queries) второй запрос добавляется к текущему запросу. После выполнения команды Объединить запросы в новый (Append Queries as New) формируется третий запрос, объединяющий два предыдущих запроса, и, таким образом, исходные два запроса остаются неизменными.

Вместо повторного применения запроса Accessories для включения всех продуктов сохраняется запрос Accessories в виде отдельного запроса и создается новый запрос Products.

1. Выполните действия 1-8 из упражнения 3.1 для загрузки запросов Bikes и Accessories в окно редактора Power Query.

2. Удостоверьтесь, что выбран запрос Accessories, затем на вкладке Главная (Home) в раскрывающемся меню Объединить запросы (Append Queries) выбе-рите команду Объединить запросы в новый (Append Queries as New).

3. После открытия окна диалога Добавление (Append) выберите таблицу Bikes, которая будет добавлена к основной таблице, и щелкните мышью на кнопке OK. В результате будет получен новый запрос, Append1, комбинирующий данные из таблиц Bikes и Accessories.

4. Переименуйте новый запрос на Products и проделайте шаг 12 из упражнения 3.1 для выгрузки таблицы Bikes. Чтобы выгрузить таблицу Accessories, можно также следовать шагу 12, но в данном случае примените инструкции для таблицы Accessories вместо аналогичных инструкций для таблицы Bikes.

5. Сохраните рабочую книгу или отчет Power BI. В следующем разделе рассмотрим иной метод добавления таблиц Bikes и Accessories в новый запрос — с помощью команды Ссылка (Reference). Также мы сравним базовое M-выражение, используемое двумя этими методами.

Г") СОВЕТ

При добавлении небольшого набора таблиц рекомендуется применять команду Объ-единить запросы в новый вместо команды Объединить запросы, что позволит в дальнейшем облегчить изменение отдельных блоков добавленной таблицы.

УПРАЖНЕНИЕ 3.2. Часть 2.

Окна диалога Зависимости запроса и Ссылки

В упражнении 3.2, часть 1, применялось преобразование Объединить запросы в новый, в результате чего был создан новый запрос, объединяющий таблицы Bikes и Accessories. Этот запрос ссылался на два других запроса и использовал их вывод в качестве входных данных. Если запрос ссылается на другой запрос, то между запросами устанавливается зависимость. Если запрос включает зависимости от других запросов, его невозможно удалить. (Например, если попытаться удалить запрос Bikes, то отображается сообщение, в котором говорится о невозможности удаления запроса, поскольку таблица Products зависит от него.)

Для просмотра зависимости между запросами откройте рабочую книгу Excel или отчет Power BI, созданный в упражнении 3.2, часть 1, и запустите редактор Power Query. На вкладке Просмотр (View) выберите команду Зависимости запроса (Query Dependencies).

Откроется окно диалога Зависимости запроса, показанное на рис. 3.3. В этом окне отображается блок-схема, которая начинается с файлов Bikes и Accessories в ка-честве источников данных, проходит через промежуточные запросы Bikes и Accessories и, наконец, заканчивается запросом Products, в который добавлены две таблицы. Такое представление пригодится при работе со сложными сценариями подготовки данных.

Теперь, когда известно, каким образом можно просматривать зависимости между запросами, познакомимся со ссылками, позволяющими формировать обширные и многократно применяемые потоки преобразования данных и решать сложные задачи. Ссылку можно применять для формирования нового запроса, источник или входные данные которого являются выходными данными для другого запроса.

Ранее, при выполнении упражнении 3.2, часть 1, таблицы Bikes и Accessories за-гружались в качестве запросов, а также применялась команда Объединить запро-сы в новый (Append Queries as New), что позволяло сформировать новый запрос, включающий добавленные результаты. Альтернативный метод, ориентированный

на достижение аналогичного результата, может быть реализован с помощью комбинации команд Объединить запросы (Append Queries) и Ссылка (Reference).

1. Выполните шаги 1-8 из упражнения 3.1 для загрузки запросов Bikes и Acces-sories в окно редактора Power Query.

2. Откройте панель Запросы (Queries) и правой кнопкой мыши щелкните на запросе Accessories. Выберите команду Ссылка (Reference), после чего появится новый запрос, Accessories (2), который начинается с вывода, созданного запросом Accessories.

3. Переименуйте запрос Accessories (2) в Products.

4. Выделив запрос Products, выберите команду Объединить запросы (Append Queries) на вкладке Главная (Home), чтобы добавить запрос Bikes в Products: После открытия окна диалога Добавление (Append) выберите таблицу Bikes, чтобы добавить ее, и щелкните мышью на кнопке OK.

Теперь рассмотрим, каким образом Power Query обрабатывает шаги Ссылка (Reference) и Объединить запросы (Append Queries), используя формулы языка M, и сравните вывод комбинации шагов Ссылка и Объединить запросы с выводом, полученным в упражнении 3.2, часть 1. Обратите внимание на код языка M.

ПРИМЕЧАНИЕ

Напомним, что в главе 1 был представлен код языка M, который генерируется Power Query для выполнения шагов преобразования. Каждый шаг преобразования, создаваемый в редакторе Power Query, преобразуется в строку кода, который можно просмотреть в строке формул.

На вкладке Просмотр (View) установите флажок Строка формул (Formula Bar), затем на панели Примененные шаги (Applied Steps) области Параметры запроса (Query Settings) выберите шаг Источник (Source). Здесь вы увидите следующую формулу M:

= Accessories

Данный код представляет вывод запроса Accessories, который применяется в качестве входных данных для нового запроса. Как можно догадаться, этот код сгенерирован путем применения команды Ссылка (Reference) по отношению к таблице Accessories.

Щелкните мышью на кнопке Расширенный редактор (Advanced Editor), находя-щейся на вкладке Главная (Home), чтобы просмотреть выражение языка M:

let

Источник = Accessories,

#"Appended Query" = Table.Combine({Source, Bikes})

in

#"Appended Query"

В главе 9 данный синтаксис рассматривается более подробно, также объясняется значение ключевых слов let и in, но сейчас можно просто сфокусировать внимание на двух строках, находящихся между let и in.

В первой строке после let размещается конструкция вывода запроса Accessories in Источник. (Заметим, что Источник также является первым шагом на панели Примененные шаги (Applied Steps).) Вторая строка применяет конструкцию Table.Combine, которая вносится между Source и Bikes, — это операция добавления, реализующая вставку объекта, между Accessories и Bikes. Возможно, уже заметили, что запросы заключаются в фигурные скобки, объяснение этому действию содержится в следующем разделе. Подсказка: фигурные скобки служат для создания списка.

Если теперь просмотреть код языка M, сгенерированный при выполнении упражнения 3.2, часть 1, путем выполнения команды Объединить запросы в новый (Append Queries as New), в окне расширенного редактора вы увидите следующий код:

let

Источник = Table.Combine({Accessories, Bikes})

in

Источник

При внимательном просмотре этих двух блоков кода можно заметить, что два выражения возвращают один и тот же результат: объединенные таблицы Accessories и Bikes.

Для завершения упражнения применим комбинацию команд Ссылка (Reference) и Объединить запросы (Append Queries) для получения тех же результатов, что и при использовании команды Объединить запросы в новый (Append Queries As New). И хотя последний метод приводит к уменьшению количества строк кода и реализует только один шаг пользовательского интерфейса вместо двух шагов, между этими методами нет особой разницы. Зачем же применялась команда Ссылка (Reference)? Как будет показано в следующих главах, команда Ссылка (Reference) может быть полезна при формировании конвейера запросов преобразования или нескольких ветвей преобразований для одного базового запроса.

Объединение трех и большего числа таблиц

До сих пор шла речь о добавлении двух таблиц. Рассмотрим более общую задачу: как добавить три таблицы или более. Один из подходов к решению этой задачи заключается в том, чтобы начать с добавления двух таблиц, а затем добавить выходные данные двух таблиц к третьей таблице. Затем можно добавить вывод этих шагов в четвертую таблицу и т. д. И хотя данный подход вполне удовлетворительный, ему трудно следовать постоянно, поскольку требуется формировать промежуточные запросы. Если нужно добавлять три или четыре таблицы, это приемлемо, но при большем числе запросов требуется уже иной подход.

УПРАЖНЕНИЕ 3.2. Часть 3.

Bikes + Accessories + Components

Снова обратимся к нашей вымышленной истории. После успешной интеграции линейки аксессуаров с велосипедами компания AdventureWorks решает приобрести новую компанию и включить дополнительные компоненты. Ваша цель как главного аналитика — свести все линейки товаров в единую таблицу.

Для выполнения этого упражнения можно загрузить следующие три файла с сайта https://aka.ms/DataPwrBIPivot/downloads:

■ C03E01 - Bikes.xlsx;

■ C03E01 - Accessories.xlsx;

■ C03E01 - Components.xlsx.

Создайте новую пустую книгу Excel или новый отчет Power BI Desktop и импортируйте три книги Excel в редактор Power Query. Каждая книга содержит один рабочий лист с соответствующим именем (т. е. рабочий лист Bike в C03E01 - Bikes.xlsx, рабочий лист Accessories в C03E01 - Accessories.xlsx, и рабочий лист Components в C03E01 - Components.xlsx). Если не удается импортировать данные рабочие книги, обратитесь к упражнению 3.1, где приведены шаги, применяемые для загрузки Bikes и Accessories. Затем выполните аналогичные шаги, чтобы загрузить Components.

1. На панели Запросы (Queries) редактора Power Query выберите запрос Com-ponents.

2. На вкладке Главная (Home) выберите команду Объединить запросы в новый (Append Queries as New).

3. После открытия окна диалога Добавление (Append) установите переключатель Три таблицы или больше (Three or More Tables). На экране появятся списки

Доступные таблицы (Available Table(s)) и Таблицы для добавления (Tables to Append), как показано на рис. 3.4. В списке Доступные таблицы находятся таб-лицы Bikes и Accessories.

4. Добавьте таблицы Bikes и Accessories в список Таблицы для добавления и щелкните мышью на кнопке OK.

В результате выполнения этих действий получим новый запрос, Append1, с объе-диненной линейкой товаров. Теперь можно переименовать его на Products и выгрузить промежуточные запросы, как показано в упражнении 3.1.

Г") СОВЕТ

В окне диалога Добавление (Append), показанном на рис. 3.4, можно контролировать порядок следования таблиц в добавленной таблице. Есть возможность выбрать таблицу, изменить ее порядок в списке, щелкая на стрелки "вверх и вниз" в правой части списка Таблицы для добавления (Tables to Append). Можно неоднократно добавлять таблицу в список Таблицы (Tables) для добавления для дублирования данных в добавленной таблице.

Теперь, после добавления таблиц Bikes, Accessories и Components, можно заметить, что применять одно и то же преобразование в строке формул редактора Power Query очень удобно. После выбора запроса Products в строке формул появится следующая формула:

= Table.Combine({Components, Bikes, Accessories})

Не кажется ли она вам знакомой? Ранее, при добавлении двух таблиц, строка формул содержала следующий код:

= Table.Combine({Accessories, Bikes})

Из синтаксиса этой формулы можно видеть, что Table.Combine — это функция, которая предназначена для выполнения преобразования по добавлению нескольких таблиц. Функция может включать названия нескольких таблиц, заключенные в фигурные скобки.

ПРИМЕЧАНИЕ

В языке M фигурные скобки служат для определения списка. Элементы списка разделяются запятыми. В следующих главах приведено больше примеров списков. С помощью списков можно формировать универсальные преобразования и даже управлять своим запросом в процессе его выполнения.

Теперь, когда мы выяснили, каким образом добавляются три таблицы, можно добавить четвертую таблицу, но на этот раз выполните это, используя только код формул. Сохраните свою книгу Excel или отчет Power BI для следующего упражнения.

УПРАЖНЕНИЕ 3.2. Часть 4.

Bikes + Accessories + Components + Clothing

Вернемся к нашей вымышленной компании. После расширения линейки товаров главный операционный директор компании AdventureWorks решает объединить подразделение Clothing (Одежда) с основными бизнес-единицами. В результате будет получена четвертая таблица линейки товаров, Clothing. В части 4 этого упражнения будет показано, каким образом можно свести все четыре таблицы вместе, — на этот раз с использованием языка M и без применения диалогового окна Добавление (Append).

Для выполнения этого упражнения можно загрузить файл C03E01 - Clothing.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads . Откройте рабочую книгу Excel или отчет Power BI Desktop из упражнения 3.2, часть 3, и импортируйте таблицу Clothing в качестве нового запроса. (Обратитесь к упражнению 3.1, где описаны шаги, применяемые для загрузки таблиц Bikes и Accessories. Затем примените аналогичные шаги для таблицы Clothing.) Переименуйте новый запрос в Clothing.

В окне редактора Power Query выберите запрос Products. Обратитесь к строке формул, в которой отобразится следующая формула:

= Table.Combine({Components, Bikes, Accessories})

Измените эту формулу, добавив Clothing в любом месте внутри фигурных скобок. Убедитесь, что все четыре таблицы разделены внутри фигурных скобок запятыми. По завершении нажмите клавишу <Enter>. Ваша формула может выглядеть следующим образом:

= Table.Combine({Clothing, Components, Bikes, Accessories})

или

= Table.Combine({Components, Clothing, Bikes, Accessories})

или

= Table.Combine({Components, Bikes, Accessories, Clothing}).

Для проверки корректности формулы Table.Combine щелкните мышью на элементе управления фильтром в заголовке столбца ParentProductCategoryName. После открытия панели Фильтр (Filter) отобразятся четыре значения: Accessories, Bikes, Components и Clothing. Проверка значений на панели Фильтр служит удобным и практичным методом тестирования правильности шагов преобразований.

Теперь, когда таблица Clothing добавлена к остальным таблицам, описывающим товар, можно перейти к новым сценариям и узнать, каким образом добавляются несколько таблиц большего масштаба.

Объединение большого количества таблиц

В предыдущем разделе показано, как можно добавлять несколько таблиц с помощью команд Объединить запросы (Append Queries) и Объединить запросы в новый (Append Queries As New) в редакторе Power Query. И хотя эти преобразования полезны, если имеется изначально определенный набор имен таблиц, однако для некоторых распространенных сценариев заранее неизвестно, какие имена таблиц доступны, а когда нужно регулярно создавать сводные данные. Поэтому ручное манипулирование запросами для включения имен текущих таблиц всякий раз, когда они изменяются, может быть довольно трудоемким процессом, и лучше его избегать.

Добавление таблиц из папки

При выполнении примера можно продемонстрировать наиболее распространенный сценарий и основное затруднение, с ним связанное: представьте, что имеется несколько таблиц, которые необходимо свести в одну. Каждая таблица находится в отдельной книге Excel, имя которой включает различные значения для года и месяца:

■ Jan2018.xlsx

■ Feb2018.xlsx

■ Mar2018.xslx

■ Apr2018.xslx

Каждый месяц новая книга Excel добавляется в общую папку, и предлагается подготовить обновленный отчет, который сводит воедино все таблицы из папки. Предположим, что отчет создан, при этом отдельно импортируется каждая из таблиц, а затем добавляется с помощью предыдущего метода. При добавлении нового файла (например, May2018.xlsx) необходимо выполнять последовательность действий для изменения запросов. Причем эта операция обычно выполняется вручную. Сначала импортируется файл Excel в виде нового запроса с именем May2018. Затем нужно изменить шаг добавления в разделе Примененные шаги (Applied Steps) в окне редактора Power Query, чтобы включить новый запрос. Можно добавить запрос в следующую формулу (новый запрос выделен полужирным шрифтом):

= Table.Combine({Jan2018, Feb2018, Mar2018, Apr2018, May2018})

Описанный метод в конечном итоге окажется довольно утомительным, если его нужно повторять каждый месяц. Разве не лучше создать усовершенствованный запрос, который автоматически добавляет любую книгу из папки — с одним обновлением книги и без ручного редактирования запроса? Это возможно благодаря импорту таблицы из папки.

УПРАЖНЕНИЕ 3.3.

Добавление таблицы AdventureWorks Products из папки Вернемся к AdventureWorks для иллюстрации данной задачи. Скажем, разработчик создал VBA-сценарий, который обновляет файл Excel при каждом выпуске нового продукта. Ежегодно создается новый файл Excel, и он обновляется с учетом новых товаров "на лету". Файлы товаров объединяются в общую папку, как показано на рис. 3.5. Как главному аналитику, вам необходимо сформировать отчет, объединяющий все товары.

В 1 0 В ’ 1 СОЗЕОЗ - Products - □ X

1 Главная Поделиться Вид “ ©

<- - ф в « 9781509307951\_files2 > СОЗ > СОЗЕОЗ - Products v О Поиск СОЗЕОЗ - Products p

\_ А

Загрузки У А Имя Дата изменения Тип Разм ер

U Документы \* щ СОЗЕОЗ . 2015 26.10.201723:41 Лист Microsoft Ex... 18 КБ

И Изображена \* Щ СОЗЕОЗ • 2016 26.10.201723:42 Лист Microsoft Ex... 19 КБ

СОЗЕОЗ-2017 26.10.201723:41 Лист Microsoft Ex... 22 КБ

Этот компьютер

Локальный дис

c i Локальный дис v

Элементов: 3 [|Н|Е

Рис. 3.5. В данном примере компания AdventureWorks собирает в общую папку книги Excel с товарами, отсортированные по годам выпуска

Для выполнения этого упражнения загрузите файл C03E03 - Products.zip с сайта https://aka.ms/DataPwrBIPivot/downloads и извлеките файлы Excel в папку C:\Data\C03\C03E03 - Products\).

1. Откройте новую книгу в Excel или новый отчет Power BI Desktop.

В Excel: на вкладке Данные (Data) или на вкладке Главная (Home) выполните команды Получить данные (Get Data) Из файла (From File) Из папки (From Folder).

В Power BI Desktop: на вкладке Главная (Home) выполните последовательность команд Получить данные (Get Data) Файл (File) Папка (Folder) Подключиться (Connect).

2. В открывшемся окне диалога Папка (Folder) просмотрите созданную ранее папку (приведена на рис. 3.5) или скопируйте и вставьте соответствующий путь в путь для папки и щелкните мышью на кнопке OK.

3. В новом открывшемся окне диалога (озаглавленном по наименованию папки, выбранной на шаге 2) задайте команду Объединить (Combine), затем выберите команду Объединить и изменить (Combine and Edit).

Г") СОВЕТ

Не рекомендуется использовать команду Объединить и загрузить (Combine and Load), которая приведет к объединению всех файлов и загрузке добавленного результата в отчет без возможности его изменения. В случае, когда вы не уверены, какие файлы следует объединить, нужно выбирать команду Изменить (Edit) в окне диалога, описанном на шаге 3. После этого откроется редактор Power Query, в котором доступен предварительный просмотр папки и где можно применять фильтры к файлам, которые нужно объединить. Например, если в папке имеются другие типы документов, можно отфильтровать файлы по расширению xlsx, что позволит объединять только файлы Excel и игнорировать другие типы. Затем можно объединить файлы в окне редактора Power Query, выбирая команду Объединить файлы (Combine Files) в заголовке столбца Content.

4. В открывшемся окне диалога Объединить файлы (Combine Files) выберите лист Sheet1, как показано на рис. 3.6, и щелкните мышью на кнопке OK.

Объединить файлы

Выберите объект, который следует извлечь из каждого файла. Дополнительные сведения □ X

Образец файла: | Первый файл р| Sheetl D

Отобразить параметры т Name ProductN Limber Color StandardCost ListF

J Параметр файла при мера"! [1] HL Mountain Frame - Black, 46 FR-M94B-46 Black 739,041

□ Sheet! HL Mountain Frame - Black, 38 FR-M94B-38 Black 739,041 A

HL Mountain Frame - Silver, 38 FR-M94S-38 Silver 747,2002

Sport-100 Helmet, Blue HL-U509-B Blue 13,0863

AWC Logo Cap CA-1098 Multi 6,9223

Long-Sleeve Logo Jersey, S U-0192-S Multi 38,4923

Long-Sleeve Logo Jersey, M U-0192-M Multi 38,4923

Long-Sleeve Logo Jersey, L U-0192-L Multi 38,4923

Long-Sleeve Logo Jersey, XL U-0192-X Multi 38,4923

HL Road Frame - Red, 62 FR-R92R-62 Red 868,6342

HL Road Frame - Red, 44 FR-R92R-44 Red 868,6342

HL Road Frame - Red, 48 FR-R92R-48 Red 868,6342

HL Road Frame - Red, 52 FR-R92R-52 Red 868,6342

HL Road Frame - Red, 56 FR-R92R-56 Red 868,6342

LL Road Frame - Black, 58 FR-R38B-58 Black 204,6251

LL Road Frame - Black, 60 FR-R38B-60 Black 204,6251

LL Road Frame - Black, 62 FR-R38B-62 Black 204,6251

Sport-100 Helmet, Red HL-U509-R Red 13,0863

Sport-100 Helmet, Black HL-U5O9 Black 13,0863

LL Road Frame - Black, 44 FR-R38B-44 Black 204,6251 V

< >

□ Пропустить файлы с ошибками Г Отмена

Рис. 3.6. В окне Объединить файлы можно выбрать рабочий лист или таблицу для объединения.

После выбора все таблицы или таблицы с одинаковыми именами в каждой книге Excel,

находящейся в папке, будут объединены.

5. В открывшемся окне редактора Power Query обратитесь к панели предварительного просмотра и рассмотрите первый столбец, Source.Name (Источник.Имя), куда входят имена файлов. Щелкните мышью на элементе управления для фильтра с заголовком Source.Name. На панели Фильтр (Filter) обратите внимание, что значения в столбце те же самые, что и реальные имена файлов в папке.

ПРИМЕЧАНИЕ

Окно диалога Объединить файлы (Combine Files) является эквивалентом окна диалога Навигатор (Navigator), которое открывается при загрузке одной книги Excel. В окне Навигатор можно выбрать рабочий лист или таблицу для изменения или для загрузки в отчет, в окне диалога Объединить файлы можно выбрать рабочий лист или таблицу для использования всех файлов из папки. Необходимо применить выбор к выбранной рабочей книге и рабочему листу, тогда этот выбор применяется ко всем другим рабочим книгам в папке.

В столбце Source.Name может содержаться некое важное значение. Так, в предыдущем примере столбец SellStartDate отображает дату выпуска каждого продукта AdventureWorks. Из этого столбца можно извлечь год выпуска, если в отчете нужно указывать годы. Однако во многих случаях подобный столбец отсутствует, и контекстная информация будет доступна только в имени файла. К счастью, эту контекстную информацию можно также извлечь из столбца Source.Name.

Чтобы извлечь сведения за 2017 год из файла C03E03 - 2017.xlsx в столбце Source.Name, можно применять различные методы преобразований. В главе 2 подробно описываются схожие сценарии, поэтому здесь выделим из них лишь некоторые. Имена файлов в этом случае содержат префикс C03E03, суффикс xlsx, и год между ними. Значение года можно извлечь из столбца Source.Name двумя путями:

1. Замените C03E03 (пустой строкой), затем замените xlsx пустой строкой. Порядок применения команды Заменить значения (Replace Values) был рассмотрен в главе 2.

2. В качестве альтернативы разделите столбец Source.Name на 9 символов, отсчитывая слева (используется разделение по количеству символов), а затем разделите второй столбец точкой или, как альтернатива, разделите второй столбец по позиции на 4 символа, находящиеся справа. Удалите первый и третий столбцы и оставьте второй столбец, который и включает значение года.

Необходимо загрузить уже добавленную таблицу Products в вашу книгу Excel или отчет Power BI. Файлы решений, C03E03 - Solution.xlsx и C03E03 - Solution.pbix, которые доступны на сайте https://aka.ms/DataPwrBIPivot/downloads.

Команда Импорт из папки

Имеются три важных момента, связанных с объединением файлов с помощью команды Импорт из папки (Import from Folder).

■ Команда импорта из папки нескольких таблиц обладает серьезными возможностями. При добавлении нового файла созданный ранее запрос не потребуется обновлять, чтобы включить новый файл в состав преобразованного. Для того чтобы удостовериться в этом, можно выполнить небольшой тест. Скопируйте и вставьте одну из существующих исходных книг в ту же папку и переименуйте продублированный файл C03E03 - 2018.xlsx. Теперь обновите книгу или отчет Power BI, созданный при выполнении упражнения 3.3, и убедитесь, что имеются новые товары 2018 года. Как видите, эта функция действительно универсальна.

■ После выполнения шага 5 в упражнении 3.3 можно заметить, что на панели Запросы появились новые элементы, такие как группы и различные типы запросов. На данный момент можете игнорировать их. Вернемся к ним позже, когда речь пойдет о функциях запросов и других важных темах.

■ Можно объединить несколько файлов из общих папок, расположенных в сети или находящихся на сайтах SharePoint. Использование общих папок позволяет поделиться книгой Excel или отчетом Power BI с другими авторами отчетов и обновить их. Если применяется Power BI, то можно опубликовать отчет в службе Power BI и наслаждаться запланированным обновлением. Если папка находится в SharePoint Online или OneDrive для бизнеса, то служба Power BI имеет прямой доступ к данным и может быть настроена на периодическое обновление. Если общая папка является локальной, можно применять Power BI On-Premises Gateway для поучения доступа к этой папке для периодического обновления со стороны службы Power BI.

Добавление листов из книги

Напоследок рассмотрим еще один важный сценарий этой главы по добавлению таблиц: объединение таблиц из нескольких рабочих листов. В предыдущем примере объединялись товары AdventureWorks, находящиеся в различных книгах. Каждая книга включала собственные товары в зависимости от года выпуска. Однако иногда случается так, что одни и те же таблицы сохраняются на разных листах в одной книге Excel. Можно ли сводить все рабочие листы в одну таблицу, сохраняя при этом контекст года выпуска для каждого продукта, как показано на рис. 3.7? Есть ли гарантия того, что каждый год не нужно обновлять запросы при добавлении нового листа? Каким образом можно обеспечить автоматическое объединение товаров следующего года в отчете при щелчке на кнопке Обновить (Refresh)?

УПРАЖНЕНИЕ 3.4.

Добавление рабочих листов — решение

Для решения описанной проблемы загрузите файл C03E04 - Year per Worksheet.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads.

1. Откройте новую книгу в Excel или новый отчет Power BI Desktop.

В Excel: перейдите на вкладку Данные (Data) и выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

Рис. 3.7. Цель обработки товаров AdventureWorks, разделенных на отдельные таблицы

по годам выпуска, состоит в том, чтобы добавить их, сохранив контекст года выпуска,

и избежать дальнейшей модификации запроса в последующие годы

В Power BI Desktop: на вкладке Главная (Home) сначала выполните команду Получить данные (Get Data), а затем — последовательность команд Файл (File) Excel Подключиться (Connect).

2. Выберите файл C03E04 - Year per Worksheet.xlsx и щелкните мышью на кнопке Импорт (Navigator).

3. В окне диалога Навигатор (Navigator) выберите строку, где находится значок папки (в данном примере, C03E04 - Year per Worksheet.xlsx).

ПРИМЕЧАНИЕ

Не выбирайте определенные листы в окне диалога Навигатор. И хотя это не запрещено, и такой подход реализован в упражнении 3.1, подобное решение в долгосрочной перспективе может привести к нежелательным результатам. В этом случае новые рабочие листы добавляются в рабочую книгу, причем придется изменять запросы и добавлять новые рабочие листы. Вместо этого неудобного подхода на шаге 3 была выбрана вся книга, что позволит реализовать надежное решение.

В Excel: щелкните мышью на кнопке Изменить (Edit).

В Power BI Desktop: если параметр Изменить (Edit) недоступен, щелкните пра-вой кнопкой мыши на файле C03E04 - Year per Worksheet.xlsx и в контекстном меню выберите параметр Изменить (Edit).

4. После открытия редактора Power Query переименуйте запрос на Products. На главной панели просмотра можно увидеть таблицу с рабочими листами в от-дельных строках. Фактическое содержание каждого листа инкапсулировано в столбце Data (Данные). Перед тем как комбинировать рабочие листы, можно удалить три столбца: Item (Элемент), Kind (Вид) и Hidden (Скрыто).

СОВЕТ

Если в исходной книге имеются скрытые рабочие листы или некоторые рабочие листы с несвязанными данными, то можно отфильтровать их до момента добавления всех рабочих листов на шаге 5; удобно применить необходимые фильтры. Например, можно применить фильтр к столбцу Hidden для того, чтобы оставить только строки, содержащие значение FALSE, что позволит исключить скрытые рабочие листы.

5. В заголовке столбца Data щелкните мышью на кнопке Развернуть (Expand).

6. После открытия панели Развернуть появятся столбцы для развертывания. Щелкните на кнопке OK для развертывания всех столбцов. После развертыва-ния столбца Data в объединенной таблице появляются лишенные смысла заголовки (Column1, Column2 и т.д.), а также заголовки и строки из каждого листа, как показано в первой таблице на рис. 3.8.

Рис. 3.8. На этой схеме преобразования показано, как очистить комбинированные листы

ПРИМЕЧАНИЕ

Если таблицы товаров на каждом рабочем листе состоят из объектов Table (Таблицы), то на шаге 6 можно развернуть фактические имена столбцов продуктов, вместо того чтобы применять названия Column1, Column2 и т. д. Для преобразования диапазона в Excel в таблицы выберите диапазон или одну ячейку в диапазоне и щелкните на кнопке Таблица (Table) на вкладке Вставка (Insert) или нажмите комбинацию клавиш <Ctrl>+<T>.

Блок-схема, изображенная на рис. 3.8, описывает три шага преобразования, необходимые для очистки объединенной таблицы. На первом шаге перемещаемся вверх по первой строке, которая была строкой заголовков таблицы 2015, а теперь служит заголовком объединенной таблицы. Затем, на следующем шаге, удаляются заголовки рабочих таблиц 2016 и 2017 годов. Наконец, на третьем шаге переименовывается столбец 2015 года, что позволит представить год выпуска каждого продукта. В дальнейшем можно применять этот тип диаграммы, что позволит представить последовательность основных шагов преобразования и зафиксировать обоснование для разрешения сложных задач подготовки данных. Далее, подробно рассмотрим эти шаги.

7. На вкладке Преобразование (Transform) выберите параметр Использовать первую строку в качестве заголовков (Use First Row as Headers). Тогда для заголовков столбцов для добавленной таблицы сохранятся заголовки столбцов первого листа.

8. На панели просмотра обратите внимание, что заголовки столбцов на рабочих листах 2016 и 2017 годов пересекаются как строки добавленной таблицы. Эти строки включают значение Name (Название) из столбца Name и должны быть удалены из таблицы. Чтобы отфильтровать эти строки, щелкните мышью на элементе управления фильтра в левой части заголовка Название. После открытия панели Фильтр (Filter) введите в поле поиска Название и отмените выделение значения Название. Затем щелкните на кнопке OK.

9. Дважды щелкните мышью на заголовке первого столбца под названием 2015 и переименуйте его на Год выпуска (Release Year).

ПРИМЕЧАНИЕ

Шаги 7-9 могут привести к осложнениям в долгосрочной перспективе. Вскоре покажем, как можно устранить эту проблему.

Теперь можно загрузить комбинированную таблицу в книгу Excel или отчет Power BI. Успешно скомбинированы по году выпуска три рабочих листа, и из названий листа извлечен год выпуска каждого товара. Пришло время протестировать масштабируемость решения и просмотреть, корректно ли оно обновляется после добавления четвертого рабочего листа с товарами, выпущенными в 2018 году.

10. Если книга Excel или отчет Power BI с комбинированными товарами по- прежнему открыты, загрузите исходный файл Excel C03E04 - Year per Worksheet.xlsx, который применялся в этом упражнении. Продублируйте рабочий лист 2017 года в качестве четвертого рабочего листа и переименуйте его в 2018.

Г") СОВЕТ

Если в Excel открыт редактор Power Query (но не Power BI Desktop), возможно, вы уже заметили, что, пока активирован редактор Power Query, нельзя получить доступ к другим открытым рабочим книгам или открыть еще больше рабочих книг Excel на вкладке Файл (File) в Excel. Следовательно, на шаге 10 может потребоваться закрыть редактор Power Query до открытия исходной книги. Чтобы обойти это ограничение, можно запустить новый экземпляр Excel с помощью панели задач или меню Пуск (Start) и открыть файл C03E04 - Year per Worksheet.xlsx. Новый экземпляр Excel не будет блокироваться текущим окном редактора Power Query.

11. Обновите запрос Products в вашей рабочей книге Excel или отчете Power BI и удостоверьтесь, что товары, продублированные из рабочего листа 2018, теперь добавляются со значением 2018 в качестве даты выпуска. Прекрасно, что это работает!

Добавьте в качестве первого лист 2014 года с историческими данными для товаров, выпущенных в 2014 году. Для этого необходимо продублировать одну из имеющихся таблиц и добавить ее как первую таблицу в документ C03E04 - Year per Worksheet.xlsx. Теперь, когда крайний левый лист — 2014, а не 2015, обновление завершается неудачно, со следующей ошибкой: "Expression.Error: The column ‘2015' of the table wasn't found" (Выражение. Ошибка: столбец "2015" таблицы не найден).

12. Чтобы быстро устранить эту ошибку, откройте редактор Power Query Editor, выберите запрос Products и выберите в панели Примененные столбцы (Applied Steps) шаг Изменение типа (Changed Types). В строке формул отобразится следующая формула:

= Table.TransforniColunmTypes(#"noBbiiiieHHbre3ai,o.noBKM",{{"2015", Int64.Type}, {"Name", type text}, {"ProductNumber", type text}, {"Color", type text}, {"StandartCost", type any}, {"ListPrice", type any}, {"Size", type text}, {"Weight", type any}, {"ParentProductCategoryName", type text}, {"ProducrCategoryName", type text}})

В предыдущей формуле найдите текст "2015" и замените его текстом "2014".

13. На панели Примененные шаги выберите шаг Переименованные столбцы (Renamed Columns). В строке формул отобразится следующая формула:

= Table.RenameColumns(#"Стрoки с примененным фильтром ",{{"2015", "Release Year"}})

Найдите текст "2015" в предыдущей формуле и замените его на "2014".

14. Закройте редактор Power Query и обновите данный отчет. Товары, выпущенные в 2014 году, теперь будут правильно комбинироваться в данном отчете.

Эти шаги можно просмотреть в файлах решений C03E04 - Solution 01.xlsx или C03E04 - Solution 01.pbix, которые доступны на сайте https://aka.ms/ DataPwrBIPivot/downloads.

Если предположить, что первым рабочим листом всегда будет 2015 или теперь, после исправления, 2014, и если вы сможете гарантировать, что новые товары будут всегда добавляться в виде новых рабочих листов справа, то ваша миссия выполнена. Как быть, если это предположение неверно? Как усовершенствовать запрос, чтобы при добавлении нового листа в качестве первого листа не нужно было повторять шаги 12-14?

В следующем разделе показано, каким образом можно улучшить запрос для поддержки сценария, когда нельзя узнать, какой лист будет первым.

Надежный подход к объединению нескольких листов

В данном разделе кратко представлены новые концепции, позволяющие повысить надежность запросов и избежать ошибок обновления. В упражнении 3.4 три рабочих листа товаров компании AdventureWorks объединены по году их выпуска. Несмотря на то что можно добавить три рабочих листа в масштабируемом виде и подключать объединенную таблицу из новых рабочих листов по мере их добавления в источник данных, можно обнаружить, что запрос не будет выполняться, если в качестве первого рабочего листа добавляется неизвестный рабочий лист.

В упражнении 3.4 выполнено три шага (7-9), что позволит объединять рабочие листы в одну таблицу. В главе 10 будет более подробно показано, как иначе выполнять каждый из этих шагов, чтобы избежать потенциальных ошибок. А в этом разделе просто кратко представлены ошибки и способы их устранения для данного конкретного сценария.

■ На шаге 7 применялась команда Использовать первую строку в качестве за-головков (First Row as Headers) и дозволенный Power Query для автоматического изменения типов столбцов. В результате 2015 год явно указан в формуле. Формула сохранена без изменений, поэтому запрос может в будущем иметь ошибки при обновлениях. В главе 10 эта ошибка называется ловушкой измененных типов.

■ На шаге 9 заголовок 2015 был переименован на Release Year. Это действие привело к созданию явной ссылки на столбец 2015 в строке формул и открыло возможность для появления сбоев, что наблюдалось при добавлении 2014 в качестве первой рабочей таблицы. В главе 10 эта ошибка упоминается как ловушка переименования столбцов.

См. также

Следует заметить, что на шаге 8 вы чуть не попали в другую ловушку — поиска в фильтрах с помощью панели Поиск вместо выбора текстовых фильтров, а затем явного задания условия фильтра. Эти и подобные им ловушки также будут рассмотрены в главе 10.

Чтобы исправить эти две ошибки, выполните следующие действия:

1. Откройте запрос Products в редакторе Power Query и удалите из панели Примененные шаги шаг Измененный тип (Changed Type). Позднее вы будете явно применять уже правильные типы.

2. Выберите шаг Переименованные столбцы (Renamed Columns) на панели При-мененные шаги. В строке формул отобразится следующая формула:

= Table.RenameColumns(#"Cmpoku с примененным фильтром",{{"2015", "Release Year"}})

3. Замените "2015" в предыдущей формуле на код Table.ColumnNames(#"CTpoku с примененным фильтром"){0}. Заключительная строка формул с выделенным полужирным шрифтом модифицированным кодом имеет следующий вид:

= Table.RenameColumns(#"Cmpoku с примененным фильтром",

{{Table.ColumnNames(#"CTpoku с примененным фильтром"){0}, "Release Year"}})

В главе 9 подробно описывается язык М. На данный момент неважно, что вам недостаточно знаний. Не нужно досконально знать этот язык для разрешения 99% проблем с данными, которые обычно встречаются в работе. Однако, поскольку формула M была здесь изменена, кратко рассмотрим, что же произошло.

В данном случае вместо того, чтобы сказать Power Query "Пожалуйста, пере-именуйте столбец 2015 в Release Year", вы изменили инструкции на "Пожалуйста, переименуйте первый столбец в Release Year".

Каким образом выполнена ссылка на первый столбец? С помощью функции Table.ColumnNames. Эта функция возвращает список имен столбцов таблицы, которую получает в качестве аргумента. В данном случае таблица называется "Строки с примененным фильтром", она была названа так на предыдущем шаге, когда выполнялось фильтрование таблицы.

Наконец, для направления Power Query к первому элементу в списке имен столбцов можно задать в фигурных скобках нулевой индекс. Этот метод в языке M позволяет получить доступ к определенным элементам в списке или в определенной строке в таблице. Индекс в языке М начинается с нуля вместо единицы. Таким образом, индекс первого элемента равен нулю, а для n-го элемента будет равен n - 1. В главе 10 применяется функция Table.ColumnNames для ссылки на столбцы с помощью жестко заданных имен, и, следовательно, это позволит избежать ошибок обновления при изменении имен столбцов.

4. Для завершения формирования надежного запроса можно явно изменить типы столбцов следующим образом:

а. Для столбца Release Year выберите тип Целое число.

б. Для столбца StandardCost выберите тип Десятичное число.

в. Для столбца ListPrice выберите тип Десятичное число.

Теперь можно закрыть редактор Power Query и обновить отчет. После этого товары будут корректно комбинироваться независимо от того, какой год указан в качестве первого листа.

Можно просмотреть предыдущие шаги в файлах решения C03E04 - Solution 02.xlsx или C03E04 - Solution 02.pbix, которые доступны на сайте https://aka.ms/ DataPwrBIPivot/downloads.

Резюме

В этой главе мы познакомились с одним из самых распространенных преобразований в Power Query. Объединение таблиц, имеющих один и тот же формат, упростит их анализ и позволит избежать сложностей с отчетностью, которые могли бы возникнуть при работе с изолированными наборами данных.

Глава начинается с рассмотрения основных преобразований Объединить запросы (Append Queries) и Объединить запросы в новый (Append Queries as New), которые позволяют скомбинировать две или более таблиц. Также описаны различия между этими двумя методами и подчеркнута важность команд Ссылка (Reference) и Зависимости запроса (Query Dependencies) для анализа потока преобразований.

Следующие разделы главы посвящены реальным сценариям, когда число таблиц и их имена неизвестны. Описан импорт из папки, которая объединяет таблицы из папки и позволяет добавлять таблицы, сохраняя контекст, составленный из имен их файлов (доступно в столбце Source.Name).

Также показано, как выполнять добавление таблицы, состоящей из нескольких рабочих листов, сохраняя исходные имена рабочих листов в качестве дополнительного контекста и гарантируя, что новые рабочие листы добавляются к запросу без дальнейшей модификации.

Наконец, в главе рассказано о некоторых распространенных "подводных камнях", но они более подробно обсуждаются в главе 10; показано, как можно улучшить запрос с помощью функции языка M Table.ColumnNames, помогающей избегать сбоев обновления при изменении имени первого рабочего листа.

При изложении материала предполагается, что добавленные таблицы имеют единый формат. Это предположение упрощает работу, но в главе 4 будут рассмотрены общие сценарии, когда даже небольшие различия в форматах таблиц могут приводить к тому, что таблицы становится невозможно комбинировать без правильной подготовки данных. Я полагаю, что именно глава 4 станет краеугольным камнем ваших знаний. При обращении к Power Query вы сможете объединить силу и мудрость. Это получилось бы у Эйнштейна, если у него в свое время был бы доступ к Power Query!

Глава 4

Объединение несовпадающих таблиц

Объедините крайности, и вы получите истинный центр.

- ФриДрих Шлегель

В ЭТОЙ ГЛАВЕ ВЫ...

■ Рассмотрите результаты некорректного сочетания таблиц.

■ Узнаете, как правильно объединять таблицы с несовпадающими именами столбцов.

■ Научитесь правильно комбинировать несовпадающие таблицы, находящиеся в папке.

■ Уточните, каким образом нормализовать имена столбцов с помощью таблицы преобразования.

■ Изучите различные возможности для нормализации таблиц с различными уровнями сложности и ограничениями по производительности.

В главе 3 было показано, как объединять несколько таблиц в Power Query, причем таким образом, чтобы тем самым упростить возможности анализа и составления отчетов. Часто таблицы имеют различный формат, они постепенно изменяются, поскольку разные владельцы манипулируют ими, не подозревая о потенциально пагубном влиянии изменений на заключительный отчет.

Эта глава посвящена решению важной задачи, а именно каким образом можно объединять несоответствующие таблицы. Для того чтобы осознать эту проблему, давайте обратимся к аналогии, связанной с ношением носков. Работа, которая проделана в главе 3, похожа на рассмотрение вопроса, как носить одну пару носков. В данной главе речь пойдет о том, как пользоваться разными носками. Как легко догадаться, если носки совпадают, надеть их проще простого. Но найти пару одинаковых носков в большой куче постиранной одежды порой довольно затруднительно.

Можно ли сочетать крайности и носить разные носки? В принципе можно, да и разными таблицами также можно пользоваться. В данной главе речь идет о работе с несовпадающими таблицами в Power Query, причем их форматы могут различаться довольно существенно.

Трудности,

вызванные несоответствием таблиц

В главе 3 при добавлении нескольких таблиц использовались различные методы: вручную загружались несколько таблиц, на вкладке Главная (Home) редактора Power Query выбирались команды Объединить запросы (Append Queries) или Объединить запросы в новый (Append Queries), что приводило к объединению отдельных таблиц в одну общую таблицу. Также таблицы импортировались из нескольких файлов, находящихся в папке, и добавлялись таблицы, взятые из нескольких листов одной рабочей книги. При выполнении различных упражнений в той главе было единственное условие — неявное предположение, не сформулированное подробно: все таблицы имели одинаковый формат.

Разумно ожидать, что для объединения таблиц нужно выполнить преобразование в единый согласованный формат. Однако, хотя это требование вполне подходит для идеальных сценариев, оно редко исполнимо в реальной жизни. Возможно, вам придется работать с таблицами, которые имеют разные форматы или, что еще хуже, могут время от времени изменяться. В этом разделе определяются основные приемы работы с несовпадающими таблицами, а также способы определения влияния расхождений на результат.

Что такое несоответствующие таблицы?

Несоответствующие таблицы в контексте этой главы — это таблицы, которые описывают одни и те же семантические сущности и факты, но имеют различающиеся названия столбцов. Например, пусть Таблица1 содержит столбцы employee (сотрудник), age (возраст) и gender (пол). Таблица2 содержит столбцы Name (Имя), Age (Возраст) и F/M (Ж/М). В таблице 4.1 проиллюстрированы различия между именами столбцов в этих двух таблицах.

Таблица 4.1. Пример несоответствия имен столбцов

Столбцы Таблица1 Таблица2 Причина несоответствия

Столбец1 Employee Name Различные имена столбцов

Столбец2 Age Age Различные имена столбцов из-за чувствитель-ности к регистру

СтолбецЗ Gender F/M Различные имена столбцов

Особенности и риски, связанные с несоответствующими таблицами

Ясно, что именно несоответствующие таблицы находятся в центре внимания в этой главе. Рассмотрим основные симптомы, с которыми можно столкнуться при попытке объединить несовпадающие таблицы или же обновить отчет, который был изначально создан из таблиц, имеющих одинаковый формат, но потом они изменились и к моменту формирования итогового отчета стали несоответствующими.

При объединении двух таблиц в редакторе Power Query каждый соответствующий столбец корректно добавляется в объединенную таблицу. Для каждой строки из первой таблицы значения в соответствующих столбцах копируются в объединенную таблицу. Та же логика применяется и для второй таблицы: для каждой строки во второй таблице значения в соответствующих столбцах копируются в объединенную таблицу.

Каким же образом Power Query обрабатывает несовпадающие столбцы? В боль-шинстве случаев Power Query объединяет таблицы в новую таблицу, которая включает расширенный набор (или объединение) всех столбцов из исходных таблиц. Каждая строка из первой таблицы копируется в объединенную таблицу, причем пустые значения отображаются в столбцах, находящихся только во второй таблице. Каждая строка из второй таблицы будет скопирована таким же образом с пустыми значениями в столбцах, которые являются исключительными для первой таблицы. Эта особенность называется разделением данных (проиллюстрирована на рис. 4.1). Вместо объединения значений из двух таблиц в один столбец Power Query сохраняет исходные столбцы с унаследованными несвязанными данными.

Рис. 4.1. Несоответствующие таблицы могут иметь два типа особенностей, которые приводят к неверным результатам: разделение данных и пропуск значений

Вторая особенность, которая может возникнуть при объединении несовпадающих таблиц, — пропущенные значения (рис. 4.1). Так происходит, если несоответствующие столбцы из первой таблицы выбираются для объединенной таблицы, а все несоответствующие столбцы из второй таблицы игнорируются. В результате для каждой строки из второй таблицы в объединенной таблице оказываются пустые значения именно в несоответствующем столбце. Далее в этой главе более подробно показано, как устранить эту проблему.

Если перечисленные особенности игнорируются, то риск получения неверного итога очевиден. Разделение данных и пропущенные значения могут приводить к неточным результатам. К счастью, теперь понятно, как выявлять особенности несовпадающих таблиц, а также насколько важно обращать на них внимание. Пришло время узнать, как решить эти проблемы, что позволит успешно комбинировать несовпадающие таблицы.

УПРАЖНЕНИЕ 4.1. Устранение несоответствующих названий столбцов: простейший подход

В этом упражнении будет показано, каким образом можно устранить несоответствующие имена столбцов и избежать разделения данных и пропущенных значений, описанных в предыдущем разделе (см. рис. 4.1). Для демонстрации проблемы и ее решения применяются книги Excel с данными компании AdventureWorks.

Рассмотрим особенность, связанную с разделением данных в простом сценарии, а также покажем, как внести исправления путем сопоставления имен столбцов. Для данного упражнения можно загрузить следующие рабочие книги с сайта https://aka.ms/DataPwrBIPivot/downloads:

■ C04E01 - Accessories.xlsx;

■ C04E01 - Bikes.xlsx.

Каждая из этих двух книг содержит перечень товаров соответствующей родительской категории: Accessories или Bikes. Стоимость товара указана в столбце Cost для Bikes (Велосипедов) и в столбце StandardCost в таблице Accessories. Проделайте это упражнение и попытайтесь выполнить объединение таблиц.

1. Откройте новую рабочую книгу в Excel или отчет в Power BI Desktop и импортируйте две рабочие книги в редактор Power Query.

2. Выберите запрос Accessories и выполните команду Объединить запросы в но-вый (Append Queries as New).

3. Обратитесь на панели просмотра (Preview) к новому запросу Append1, где показан предварительный просмотр комбинированных таблиц. Обратите внимание, что оба столбца (Cost и StandardCost) включены в объединенные результаты. Столбец Cost содержит фактические затраты из таблицы Bikes и нулевые значения из таблицы Accessories, а столбец StandardCost содержит фактические затраты из таблицы Accessories и нулевые значения из таблицы Bikes.

ПРИМЕЧАНИЕ

Возможно, вы заметили, что при слиянии добавленных таблиц Accessories и Bikes утрачивается информация о родительской категории, которая была доступна в имени файла и в таблице. В главе 5 будет показано, как можно сохранить контекст таблицы.

4. Выберите запрос Accessories и переименуйте столбец StandardCost на Cost. Те-перь, при повторном выборе запроса Append1, заметим, что особенность разделения данных устранена, несовпадающие столбцы объединены.

Описанное здесь решение несложно, его часто можно применять для устранения несоответствующих имен столбцов. Но иногда использовать ручное переименование для устранения несовпадающих имен столбцов нельзя. Например, пусть имеются две таблицы со 100 столбцами, которые нужно переименовать для достижения соответствия. Переименование вручную, о котором здесь идет речь, превратится в трудоемкую задачу. Более того, в большинстве случаев при объединении таблиц из папки ручное переименование столбцов не работает, что будет продемонстрировано в следующем разделе.

Объединение несоответствующих таблиц

из одной папки

В предыдущем разделе было показано, каким образом можно вручную изменять несоответствующие имена столбцов перед сведением таблиц. Если в папке не-сколько таблиц с несовпадающими именами столбцов, то ситуация усложняется. С одной стороны, их ручное сведение не всегда удовлетворит вас, особенно если имеется большое число файлов. С другой стороны, объединение файлов из папки не позволит применить различную логику для каждого файла.

В этом случае может возникнуть желание перейти к исходному коду и переименовать столбцы там. Допустим, в таблице имеется 20 столбцов в каждой из 60 книг в папке. Придется обработать более 1200 столбцов, чтобы убедиться, что их имена совпадают, и переименовать их вручную в исходном коде. Такой подход непрактичен, особенно если вы не располагаете исходными файлами, а будете получать их периодически, часто с несоответствующими именами столбцов.

Напомним, что в главе 3 было показано, как объединить несколько таблиц из папки. Следуя шагам, описанным в этой главе, Power Query использовал заголовки первой таблицы в качестве заголовков объединенной таблицы. При объединении таблиц с несовпадающими именами столбцов из папки вскоре возникают пропущенные значения. В следующем разделе демонстрируется эта особенность, там показано, как можно успешно комбинировать несовпадающие таблицы из папки.

УПРАЖНЕНИЕ 4.2. Часть 1. Иллюстрация особенности, связанной с пропущенными значениями

Обратимся к вымышленной компании AdventureWorks. Каждый из четырех менеджеров по линейке товаров компании располагает книгой Excel, описывающей запасы товаров. Главному аналитику необходимо объединить четыре таблицы и поделиться результатами анализа, но при получении доступа к различным рабочим книгам обнаруживается, что менеджеры линейки товаров творчески подошли к работе, изменив многие имена столбцов в таблицах товаров (рис. 4.2). Если просто объединять таблицы из папки, обнаружится, что это работает неудовлетворительно. При выполнении упражнения рассмотрим поэтапное решение этой проблемы.

О С04Е02 - Accessories.xlsx \*■ Product D Color Standardcost ListPrice

® С04Е02 - Bikes.xlsx \* me Product Number Color Standard Cost List Price

® С04Е02 - Clothing.xlsx \* Product name Prod num Pro color Prod cost Prod price

ОД С04Е02 - Components.xlsx - \*■ Name Product Number Color Standard Cost List Price

Рис. 4.2. В папке AdventureWorks Products рабочие книги содержат несоответствующие имена столбцов

1. Загрузите файл C04E02 - Products.zip с сайта https://aka.ms/DataPwrBIPivot/downloads.

2. Извлеките содержимое данного ZIP-файла в новую папку: C:\Data\C04\C04E02 - Products\. Теперь эта папка содержит следующие файлы:

а. C04E02 - Accessories.xlsx

б. C04E02 - Bikes.xlsx

в. C04E02 - Components.xlsx4

г. C04E02 - Clothing.xlsx

3. Изучите содержимое этих рабочих книг и обратите внимание, что имена столбцов слишком сильно отличаются, что затрудняет их ручное переименование.

4. Импортируем четыре несоответствующие таблицы из папки, при этом мы можем столкнуться с особенностью пропущенных значений:

В Excel: откройте новую рабочую книгу, перейдите на вкладку Данные (Data) и выполните команды Получить данные (Get Data) Из файла (From File) Из папки (From Folder).

В Power BI Desktop: запустите новый отчет, выполните команду Получить данные (Get Data), в окне диалога Получение данных (Get Data) выполните команды Файл (File) Папка (Folder).

5. В открывшемся окне диалога Папка (Folder) выберите папку C:\Data\C04\ C04E02 - Products и щелкните мышью на кнопке OK.

6. В следующем окне диалога, которое называется так же, как выбранная на шаге 5 папка, в раскрывающемся меню Объединить (Combine) выберите команду Объединить и изменить (Combine and Edit).

7. После открытия окна диалога Объединить файлы (Combine Files) в раскры-вающемся меню Образец файла (Example File) выберите любую команду, затем выделите лист Sheet1 и щелкните на кнопке OK.

ПРИМЕЧАНИЕ

Раскрывающееся меню Образец файла (Example File) позволяет выбрать рабочую книгу и применить ее формат ко всем файлам в папке. В рамках этого выбора необходимо решить, какие имена столбцов будут унаследованы объединенной таблицей. Каждый файл содержит разные имена столбцов, и скоро проявится особенность пропущенных значений.

8. После открытия окна редактора Power Query прокрутите вниз основную панель просмотра до тех пор, пока строки не изменятся от Accessories до Bikes. Столбцы Product, StandardCost и ListPrice содержат нулевые значения (которые преобразуются в пустые значения в отчете), как показано на рис. 4.3.

Прежде чем устранить синдром пропущенных значений, нужно открыть панель Запросы (Queries) и выбрать запрос Преобразовать пример файла из файла C04E02 - Products (Transform Sample File from C04E02 - Products), который в данной книге играет роль примера запроса. Этот запрос сгенерирован в результате объединения файлов из папки. После выбора примера запроса можно увидеть на главной панели просмотра редактора Power Query, что данные из таблицы Accessories отображаются правильно. Устранение синдрома пропущенных значений начинается с примера запроса.

Г") СОВЕТ

При объединении файлов из папки Power Query создает функцию запроса, пример запроса, запрос файла и параметр. Эти параметры помогают объединять несколько файлов из папки на основе выбранных данных. Основное преобразование для каждо-го файла реализовано в функции. Для того чтобы настроить преобразование для каж-дого файла, можно вносить изменения в пример запроса. Эти изменения распростра-няются на функцию и применяются ко всем файлам. Всякий раз при объединении файлов из папки найдите примерный запрос, который начинается с текста Преобразование примера файла из (Transform Sample File from) и рассматривает как изменение каждого файла, так и его добавление.

УПРАЖНЕНИЕ 4.2. Часть 2. Предположение

о том же порядке столбцов и обобщение заголовка

Чтобы устранить проблему пропущенных значений, предположим, что все таблицы товаров имеют одинаковый порядок столбцов, даже если имена их столбцов различны. При таком допущении можно задать произвольные имена столбцов, такие как стандартные имена столбцов Column1, Column2 и т. д., поскольку для корректного добавления таблиц можно учесть порядок следования столбцов. Сначала выполните действия 1-8 из упражнения 4.2, часть 1, а потом — следующие шаги:

9. При открытом редакторе Power Query из шага 8 выберите запрос Преобразо-вать пример файла из C04E02 - Products (Transform Sample File from C04E02 - Products) на панели Запросы (Queries) и переименуйте его на Products Sample.

10. На панели Примененные шаги (Applied Steps) удалите последний шаг, По-вышенные заголовки (Promote Headers). После удаления этого шага объеди-ненная таблица будет иметь общие столбцы с именами Column1 (Столбец1), Column2 (Столбец2), Column3 (Столбец3) и т. д. Если все таблицы имеют одинаковые общие имена столбцов, то данные в добавленной таблице будут корректными.

11. Выберите запрос C04E02 - Products и просмотрите объединенную таблицу. Обратите внимание на следующую ошибку:

Expression.Error: Столбец "Product" таблицы не найден

(The column 'Product' of the table wasn't found).

12. На панели Примененные шаги (Applied Steps) удалите последний шаг, Изме-ненный тип (Changed Type). Теперь в объединенной таблице отсутствуют пропущенные значения. Можно убедиться в этом, выполнив прокрутку снова и удостоверившись, что включены все значения в таблице Accessories.

13. Теперь, если в запросе C04E02 - Products все данные объединены корректно, можно повысить первую строку до уровня заголовка, выбрав команду Использовать первую строку в качестве заголовков (Use First Row as Headers) на вкладке Преобразование (Transform).

К сожалению, заголовки из каждой исходной таблицы добавляются как строки в итоговой таблице. В главе 3 аналогичная проблема была устранена путем фильтрации заголовков. В данном случае имеются различные имена столбцов в виде строк, и трудно определить, какие значения следует явно исключить из фильтра. Для решения этой проблемы можно пометить заголовки, добавляя столбец индекса перед шагом объединения. Индекс служит подсказкой при идентификации встроенных заголовков.

14. На панели Запросы (Queries) выберите запрос Products Sample (Примеры то-варов). На вкладке Добавление столбца (Add Column) выберите команду Столбец индекса (Index Column). Можно видеть, что данные примера содер-жат увеличивающийся индекс для каждой строки, и первая строка, которая включает исходные имена столбцов таблицы, теперь имеет индекс, равный нулю.

15. Вернитесь к запросу C04E02 - Products и удалите все заголовки, фильтруя строки с нулевым индексом. Для выполнения этого действия щелкните мышью на элементе управления фильтром в последнем столбце, который помечен символом 0. Отмените выбор значения 0 на панели фильтра и щелкните мышью на кнопке OK.

16. Удалите первый столбец (столбец C04E02 - Accessories.xlsx) и последний стол-бец (столбец 0). После удаления этих столбцов можно настроить формулу, обеспечивающую повышение надежности запросов, как показано в последнем разделе главы 3. В этом случае поменяйте статическую ссылку на имя столбца C04E02 - Accessories.xlsx динамической ссылкой с помощью функции Table.ColumnName(...){0}.

Измените код в строке формул, заменив формулу:

= Table.RemoveColumns(#"CTpoku с примененным фильтром",

{"C04E02 - Accessories.xlsx", "0"})

на формулу:

= Table.RemoveColuшns(#"Стpоки с примененным фильтром ", {Table.ColuшnNames(#"Отфильтpованные строки"){0}, "0"})

Теперь можно загрузить комбинированную таблицу товаров и приступать к анализу.

Решение этого упражнения доступно в книге Excel C04E02 - Solution.xlsx и в отчете Power BI C04E02 - Solution.pbix, которые можно загрузить с сайта https:// aka.ms/DataPwrBIPivot/downloads.

Вспомните, что на шаге 10 были понижены заголовки первой таблицы, чтобы применить обобщенные столбцы Column1, Column2 и т. д. Этот шаг стал возможен, поскольку мы предположили, что порядок столбцов во всех таблицах одинаков. Представьте, что случится, если одна из этих таблиц будет иметь иной порядок. Например, что произойдет, если менеджер по линейке товаров подразделения Bikes компании AdventureWorks поменяет положение столбцов Name (Название) и Product\_Number (Номер\_Товара). Применение методики, описанной в этом разделе, может привести в негодность объединенную таблицу. Все строки из таблицы Bikes получат значения кода товара в качестве названий продукта и значения названия товара в качестве кода товара. Чтобы избежать этого, необходимо заранее договориться с вашими менеджерами линейки товаров, что позволит гарантировать сохранность порядка столбцов в исходных файлах. Другой подход заключается в объединении таблиц с использованием иного метода и нормализации всех заголовков до объединения таблиц. В следующем разделе рассматривается именно этот способ.

УПРАЖНЕНИЕ 4.3. Простая нормализация, выполняемая с помощью функции Table.TransformColumnNames

Автоматическое сопоставление и переименование имен столбцов в нескольких таблицах является важной частью объединения несовпадающих таблиц. Если имеется ряд несовпадающих столбцов, ручное переименование определенных столбцов, как показано в упражнении 4.1, нецелесообразно и не масштабируется при частых обновлениях или большом количестве объединяемых файлов. Если предположить, что имена столбцов имеют согласованный порядок в таблицах, то подход по обобщению заголовков, приведенный в упражнении 4.2, вполне целесообразен. Но если нельзя полагаться на согласованный порядок, то именно нормализация имен столбцов может стать находкой.

По сути, нормализация столбцов — это простое изменение текста. Во многих случаях необходимо ко всем несоответствующим именам столбцов применить незначительные манипуляции с текстом, что позволит обеспечить их согласованный формат. Например, можно заменить все символы подчеркивания пробелами или использовать только заглавные буквы в именах столбцов.

Простые корректировки текста, такие как выбор строчных или прописных букв для каждого слова, могут быть эффективными шагами при сопоставлении имен столбцов. Power Query по своей природе чувствителен к регистру символов, поэтому, например, столбцы column1 и Column1 будут обрабатываться как два разных столбца. Если предполагается изменение таблиц, а в именах столбцов может свободно меняться регистр, то преобразование с учетом одного этого подхода устранит несоответствия.

Для преобразования имен столбцов с помощью простой текстовой манипуляции можно применить для таблицы формулу M Table.TransformColumnNames. Например, если последний шаг на панели Примененные шаги (Applied Steps) называется Предыдущий шаг (Previous Step), то подойдет следующая формула для преобразования в заглавные букв каждого слова в названиях столбцов:

= Table.TransformColuninNames(#"ripegMgyinuft шаг ", Text.Proper)

Заменить подчеркивание пробелами можно с помощью следующего выражения языка M:

= Table.TransformColumnNames(#"ripegMgynuft шаг", each Replacer.ReplaceText( , " ", " "))

Эти две формулы были использованы в данном упражнении для объединения таблиц товаров AdventureWorks.

ПРИМЕЧАНИЕ

Следует отметить, что на этом этапе совершенно не обязательно понимать синтаксис этих формул, описанный в главе 9. На данный момент следует знать, что и Text.Proper, И Replacer.ReplaceText ЯВЛЯЮТСЯ M-фуНКЦИЯМИ, которые помогают преобразовывать текст. Можно использовать любую функцию, включенную в Table. TransformColumnNames. В предыдущих примерах функциЯ Text.Proper задает заглавные буквы в каждом тексте, а Replacer.ReplaceText заменЯет в тексте все экземплЯры второго аргумента третьим аргументом.

1. Загрузите файл C04E03 - Products.zip с сайта https://aka.ms/DataPwrBIPivot/downloads.

2. Извлеките содержимое ZIP-файла в новую папку: C:\Data\C04\C04E03 - Products\. Эта папка сейчас включает следующие файлы:

а. C04E03 - Accessories.xlsx;

б. C04E03 - Bikes.xlsx;

в. C04E03 - Components.xlsx;

г. C04E03 - Clothing.xlsx.

Здесь используются те же файлы, как и при выполнении упражнения 4.2, но с двумя преднамеренными изменениями. Во-первых, все несоответствующие имена столбцов могут быть согласованы, если вводить каждое слово с заглавной буквы и заменять подчеркивания пробелами. Во-вторых, в таблице Bikes первые два столбца поменялись местами, поэтому для объединения таблиц невозможно воспользоваться способом обобщения заголовка из упражнения 4.2.

3. Откройте новую пустую рабочую книгу в Excel или новый отчет в Power BI и выполните следующие действия.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из папки (From Folder).

В Power BI Desktop: на вкладке Главная (Home) выберите команду Получить данные (Get Data), в окне диалога Получить данные (Get Data) выполните команды Файл (File) Папка (Folder).

4. В открывшемся окне диалога Папка (Folder) выберите папку C:\Data\C04\ C04E03 - Products и щелкните мышью на кнопке OK.

5. В следующем окне диалога, которое называется так же, как папка, выбранная на шаге 4, в раскрывающемся меню Объединить (Combine) щелкните на кнопке Объединить и изменить (Combine и Edit).

6. После открытия окна диалога Объединить файлы (Combine Files) в раскры-вающемся меню Пример файла (Example File) выберите любой параметр, затем выберите вкладку Sheet1 и щелкните мышью на кнопке OK.

Как и ожидалось, при открытии редактора Power Query вы убедитесь в наличии проблемы с пропущенными значениями.

7. Выберите пример запроса Преобразовать пример файла из C04E03 - Products (Transform Sample File from C04E03 - Products) и переименуйте его в Products Sample.

8. При выбранном запросе Products Sample (Пример товаров) щелкните на значке fx в строке формул (рис. 4.4).

9. Обратите внимание, что на панели Примененные шаги (Applied Steps) создан новый шаг, Пользовательская1, который имеет следующую формулу:

= #"Повышенные заголовки

Шаг Повышенные заголовки (Promoted Headers) был последним шагом на панели Примененные шаги перед тем, как выполнен щелчок на кнопке fx.

#”Повышенные заголовки” — переменная, используемая для вывода этого шага. Поскольку данная переменная возвращает таблицу с несовпадающими именами столбцов, к ней можно применить функцию Table.TransformColumnNames с помощью метода Text.Lower, что позволит задать строчные буквы в именах столбцов. Для этого измените формулу следующим образом:

= Table.TransformColumnNames(#"noBbmeHHHe заголовки", Text.Lower)

10. Нажмите клавишу <Enter> и обратите внимание, что все имена столбцов находятся в нижнем регистре. Если предпочтительнее заглавные буквы в именах столбцов, то можно в предыдущей формуле заменить функцию Text.Lower на Text.Proper.

ПРИМЕЧАНИЕ

Даже если в наборах данных заголовки соответствуют нужным и совпадающим именам столбцов, эту формулу можно применить для предотвращения возможных ситуаций, когда владелец исходной таблицы может принять решение о переходе с заглавных букв в верхний или нижний регистр или комбинировать разные регистры.

11. Для замены символа подчеркивания в именах столбцов пробелами снова щелкните на значке fx в строке формул.

12. Обратите внимание на то, что в панели Примененные шаги (Applied Steps) появился новый шаг, Пользовательская2 (Custom2), а в строке формул имеет-ся следующая формула:

= Custom1

Примените функцию Table.TransformColumnNames, включающую аргументы Пользова- тельская1 и Replacer.ReplaceText, для замены подчеркивания пробелами в именах столбцов, изменив формулу следующим образом:

= Table.TransforniColuninNames(no.nb3OBaTe.nbc:Ka41, each Replacer.ReplaceText( ," ", " "))

13. Нажмите клавишу <Enter> и обратите внимание на то, что все имена столбцов теперь содержат пробелы вместо подчеркивания. Как видите, имена столбцов стали удобными для пользователя.

14. Выберите запрос C04E03 - Products и удалите последний шаг, Измененный тип (Changed Types), на панели Примененные шаги (Applied Steps). Теперь заметьте, что все имена столбцов содержат пробелы вместо подчеркивания, файлы корректно сведены вместе, без каких-либо дополнительных признаков пропущенных значений.

Можно загрузить и просмотреть файлы решения C04E03 - Solution.xlsx и C04E03 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

Таблица преобразования

При выполнении упражнения 4.3 применялись простые текстовые манипуляции к несоответствующим именам столбцов, также была выполнена замена подчеркиваний пробелами. В реальных сценариях простых операций над текстом, к сожалению, обычно недостаточно. Для того чтобы это продемонстрировать, вернемся к вымышленной компании AdventureWorks и к набору данных, применяемому в упражнении 4.2. На этот раз, как и в упражнении 4.3, первые два столбца поменялись местами, поэтому метод обобщения заголовка, показанный в упражнении 4.2, в данном случае не работает.

Для следующей серии упражнений понадобится набор данных C04E04 - Products.zip, который можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

В папке Products находятся различные таблицы со столбцами ID, Product\_Number, Product\_num и Product Number, которые используются разными менеджерами продуктовой линейки для представления одного и того же столбца. Нормализовать имена этих столбцов с помощью простых операций над текстом нельзя. Вместо этого можно сформировать таблицу преобразования с парами несоответствующих имен столбцов и их требуемых нормализованных имен. На рис. 4.5 показаны несоответствующие названия столбцов, а также каким образом они могут быть представлены в таблице преобразования. В столбце Source (Источник) приведены исходные несоответствующие имена, которые необходимо заменить; в столбце Target (Цель) — требуемые нормализованные имена.

Создание таблицы преобразования может занять много времени. Однако создавать строки для каждого имени столбца из исходных файлов не нужно. В примере, показанном на рис. 4.5, таблица преобразования не содержит пару Name Name, в этом нет необходимости. В эту таблицу должны включаться только пары несоответствующих столбцов и их желаемые замены. Если имеются сотни столбцов, то

Рис. 4.5. В таблице преобразования каждое несоответствующее имя столбца

связано с целевым именем столбца

в таблицу преобразования следует включить только несколько несовпадающих столбцов.

Для просмотра фактической таблицы переходов, которая применяется в упражнении 4.4, можно загрузить файл рабочей книги C04E04 - Conversion Table.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads.

Г") СОВЕТ

При использовании внешней книги Excel в качестве таблицы преобразования можно начать с небольшой таблицы с несколькими несоответствующими именами столбцов и постепенно добавлять новые несоответствующие столбцы и необходимые замены. При подключении таблицы преобразования к запросам не нужно редактировать за-прос для уточнения нормализации новых несоответствующих имен столбцов. Вместо этого просто измените таблицу внешних преобразований и обновите отчет.

Теперь, когда стало понятно, каким образом вручную создается таблица преобразований, можно узнать, как применить ее для замены несоответствующих столбцов. Для этого можно воспользоваться преобразованием Объединение запросов (Merge Queries), которое доступно на вкладке Главная (Home) редактора Power Query. Как было показано в главе 2, запросы на объединение позволяют объединять несколько таблиц на основе общих значений указанных столбцов.

Прежде чем познакомиться с запросами на объединение для применения таблицы преобразования, рассмотрим одну из основных проблем нормализации имен столбцов. В последующих разделах этой главы показано, как применять функцию Table.TransformColumnNames, что и продемонстрировано в упражнении 4.3, и как использовать таблицу преобразования внутри пользовательской функции, служащей для замены имен столбцов. Поскольку описанная методика требует более глубокого знания языка M, в данной главе сначала демонстрируются три других подхода к применению таблицы преобразования на основе пользовательского интерфейса, в которых не требуется расширенных M-формул.

Пользовательский интерфейс редактора Power Query включает много вариантов преобразований текста в столбцах, но не в заголовках. Упражнения 4.4, 4.5 и 4.6 фокусируются на нормализации имен столбцов при помощи трех методик, которые задействуют пользовательский интерфейс и сводят к минимуму необходимость

знания языка M. В этих упражнениях отсутствует функция Table.TransformColumnNames, поскольку требуется создание пользовательской функции. Вместо этого в упражнениях продемонстрировано, как работать с элементами управления пользовательского интерфейса редактора Power Query.

СОВЕТ

Если вы знакомы с языком M, то вам пригодится четвертый подход, который будет описан в последнем разделе этой главы. Решение, рассматриваемое в упражнении 4.7, окажется лучшим способом объединения несовпадающих таблиц и нормали-зации имен их столбцов

Желательно проделать и другие упражнения, что позволит оценить их, узнать плюсы и минусы подобных подходов. Кроме того, они могут быть полезны при рассмотрении других задач с данными.

УПРАЖНЕНИЕ 4.4. Методика транспонирования, реализуемого с помощью таблицы преобразования

Для нормализации несовпадающих столбцов с помощью таблицы преобразования необходимо временно преобразовать имена столбцов в таблицу с одним столбцом, объединить таблицу преобразования, заменить несоответствующие имена столбцов требуемыми целевыми и преобразовать имена столбцов обратно в заголовки. В этом упражнении показан первый из трех способов, применяемых для выполнения этой задачи. Начнем с той же последовательности операций, которая использовалась в упражнениях 4.2 и 4.3 для загрузки таблиц из папки.

1. Загрузите архив C04E04 - Products.zip и извлеките его содержимое в папку C:\Data\C04\C04E04 - Products.

2. Загрузите файл рабочей книги C04E04 - Conversion Table.xlsx с сайта https:// aka.ms/DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C04\. Теперь папка C:\Data\C04\C04E04 - Products содержит следующие файлы:

а. C04E04 - Accessories.xlsx;

б. C04E04 - Bikes.xlsx;

в. C04E04 - Components.xlsx;

г. C04E04 - Clothing.xlsx.

3. Импортируйте таблицу преобразования в новую рабочую книгу или отчет Power BI:

В Excel: откройте новую рабочую книгу и на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook). Выберите файл C04E04 - Conversion Table.xlsx. Затем, в окне диалога Навигатор (Navigator), выберите строку Header\_Conversion и щелкните мышью на кнопке Изменить (Edit). Наконец, выберите команды Закрыть и загрузить в (Close & Load To) и установите переключатель Только создать подключение (Connection Only).

В Power BI Desktop: запустите новый отчет, выберите команду Получить данные (Get Data), в окне диалога Получить данные (Get Data) выполните команды Файл (File) Excel. Выберите файл C04E04 - Conversion Table.xlsx. Затем в окне Навигатор (Navigator) выберите строку Header\_Conversion и щелкните на кнопке Изменить (Edit). Наконец, выберите запрос в панели Запросы и отмени-те выбор опции Включить загрузку (Enable Load). Затем щелкните мышью на кнопке Закрыть и применить (Close & Apply).

Теперь, при наличии таблицы преобразований в качестве нового запроса, можно загрузить содержимое папки Products.

4. Импортируйте таблицы товаров из папки:

В Excel: на вкладке Данные (Data) выполните последовательность команд По-лучить данные (Get Data) Из файла (From File) Из папки (From Folder).

В Power BI Desktop: на вкладке Главная (Home) выберите команду Получить данные (Get Data), в окне диалога Получить данные (Get Data) выполните команды Файл (File) Папка (Folder).

5. В окне диалога Папка (Folder) выберите папку C:\Data\C04\C04E04 - Products и щелкните мышью на кнопке OK.

6. В следующем окне диалога, которое называется так же, как папка, выбранная на шаге 5, в раскрывающемся меню Объединить (Combine) щелкните мышью на кнопке Объединить и изменить (Combine and Edit).

7. После открытия окна диалога Объединить файлы (Combine Files) в раскры-вающемся меню Пример файла (Example File) выделите любую опцию и затем выберите лист Sheet1 и щелкните на кнопке OK.

8. На панели Запросы (Queries) выберите запрос Преобразовать пример файла из C04E04 - Products (Transform Sample File from C04E04 - Products) и пере-именуйте его в Products Sample. Напомним, что этот запрос является примером запроса. Введенные здесь изменения повлияют на таблицу, добавленную в запросе C04E04 - Products.

9. Переименуйте запрос C04E04 - Products в Appended Products и сохраните рабо-чую книгу или отчет Power BI. Затем создайте три ее копии; они пригодятся при выполнении упражнений 4.5, 4.6 и 4.7.

Преобразование транспонирования

Первый способ нормализации имен столбцов основан на преобразовании Транспонировать (Transpose), которое доступно на вкладке Преобразование (Transform) в редакторе Power Query. С помощью транспонирования каждая ячейка в столбце X и строке Y будет помещена в столбец Y и строку X.

При манипулировании значениями имен столбцов выполнить это в редакторе Power Query гораздо проще, если имена столбцов представлены вертикально в столбце. Именно здесь удобно применить преобразование транспонирования.

Однако при попытке транспонирования таблицы обнаруживается, что исходные имена столбцов не сохраняются после преобразования. Для сохранения имен столбцов необходимо понизить имена столбцов до первой строки, а затем транспонировать таблицу. Для этого откройте сохраненную книгу или отчет Power BI из предыдущих шагов (1-9) и запустите редактор Power Query. На панели Запросы (Queries) выберите запрос Products Sample и выполните следующие дополнительные действия:

10. В разделе Примененные шаги (Applied Steps), при выбранном запросе Products Sample, удалите шаг Повышенные заголовки (Promoted Headers). Теперь заголовки примера таблицы находятся в первой строке.

11. На вкладке Преобразование (Transform) выберите команду Транспонировать (Transpose). Теперь столбец Column1 включает имена столбцов, которые необходимо нормализовать.

Последовательность объединения

Теперь вы достигли этапа нормализации при объедении таблицы преобразования в транспонированную таблицу для выполнения массовой замены несоответствующих имен столбцов. Выполните следующие дополнительные действия:

12. При выбранном запросе Products Sample выберите команду Объединить за-просы (Merge Queries), находящуюся на вкладке Главная (Home).

13. В открытом окне диалога Слияние (Merge) выберите столбец Column1 в Products Sample и в раскрывающемся меню Products Sample выберите строку Header\_Conversion (рис. 4.6). Затем выберите столбец Source, находящийся под строкой Header\_Conversion.

14. Удостоверьтесь, что в раскрывающемся меню Тип соединения (Join Kind) вы-бран параметр Внешнее соединение слева (все из первой таблицы, совпадающие из второй) (Left Outer (All from First, Matching from Second)), и щелкните мышью на кнопке OK.

15. В окне редактора Power Query, при добавлении нового столбца Header\_ Conversion как последнего столбца, с объектами таблицы в качестве значений разверните столбец Header\_Conversion (щелкнув мышью на элементе управ-ления справа от его заголовка) или выберите столбец Header\_Conversion, а за-тем щелкните мышью на кнопке Развернуть (Expand), находящейся на вкладке Преобразование (Transform).

16. Для извлечения значения Target (Цель) из таблицы Header\_Conversion на па-нели Развернуть (Expand) отмените выбор переключателя Источник и щелк-ните на кнопке ОК. Новый столбец преобразуется в столбец Header\_ Conversion.Target со значениями Target в строках, где Column1 и Source имели одинаковое имя столбца, или с нулевым значением в строках, где отсутст-вовали значения Source, которые бы соответствовали значениям в Column1. Теперь, когда имеются нулевые значения в Header\_Conversion.Target, будем применять исходные имена столбцов в Column1.

На следующем шаге добавим условный столбец для копирования целевых значений из строк, которые должны быть нормализованы, и исходных значений из

Слияние

Выберите таблицу и совпадающие столбцы для создания объединенной таблицы.

Products Sample

Columnl Column2 Column3 Column4 Column5 Column6 Column

Product Water Bottle - 30 oz. Mountain Bottle Cage Road Bottle Cage Patch Kit/8 Patches Hitch Rack - 4-Bike Bike We

ID WB-H098 BC-M005 BC-R2O5 PK-7098 RA-H123 CL-900S

Color null null null null null

StandardCost 1,8663 3,7363 3,3623 0,8565 44,88

ListPrice 4,99 9,99 8,99 2,29 120

< >

Header Conversion

Source

Product Target

Name

Product\_name Name

ID Product Number

Product\_Number Product Number

Prod\_num Product Number

Тип соединения

Внешнее соединение слева (все из первой таблиц.

О Выбор согласовал 5 из первых строк (9).

Рис. 4.6. Для нормализации несоответствующих имен столбцов можно применить

таблицу преобразования в окне диалога Слияние

строк, где в столбце Header\_Conversion.Target имеются нулевые значения. Новый столбец будет включать нормализованные имена столбцов или исходные имена столбцов, если исходные имена столбцов не были найдены в таблице преобразования.

17. На вкладке Добавление столбца (Add Column) выберите команду Условный столбец (Conditional Column). Откроется окно диалога Добавление условного столбца (Add Conditional Column). Чтобы настроить параметры в этом окне, выполните следующие действия (рис. 4.7):

а. В поле Имя нового столбца (New Column Name) введите New Column Names.

б. В раскрывающемся меню Имя столбца (Column Name) выберите пункт Header\_Conversion.Target.

в. В поле Значение (Value) введите null.

г. В левом раскрывающемся меню Вывод (Output) выделите опцию Выберите столбец (Select a Column).

д. В правом раскрывающемся меню Вывод выберите пункт Column1.

е. В левом раскрывающемся меню В противном случае выберите опцию Вы-берите столбец (Select a Column).

ж. В правом раскрывающемся меню В противном случае (Otherwise) выберите пункт Header\_Conversion.Target и щелкните на кнопке OK.

18. Удалите столбцы Column1 и Header\_Conversion.Target.

19. Переместите столбец New Column Names на место первого столбца таблицы (просто перетащите и опустите его с правой на левую сторону или воспользуйтесь приведенным далее советом).

СОВЕТ

Если в таблице слишком много столбцов, а нужно переместить столбец в начало, вместо перетаскивания столбца можно выделить столбец и на вкладке Преобразова-ние (Transform) выбрать команду Переместить (Move), а затем — В начало (To Beginning). Также можно щелкнуть правой кнопкой мыши на заголовке столбца и вы-брать в контекстном меню команду Переместить, а затем выбрать опцию В начало.

ПРИМЕЧАНИЕ

Переупорядочение столбцов в таблицах может быть довольно сложным, если использовать средства Power Query. В главе 10 будет показано, как можно избежать рисков, связанных с этим процессом. При перемещении столбца базовая формула M, которая ссылается на все столбцы таблицы, создает ненадежный запрос, который не будет обновляться, если какой-либо из столбцов отсутствует или переименован.

Обратное транспонирование

На шаге 11 транспонировалась таблица и перемещались имена пониженных заголовков в столбец Column1. Команда Транспонировать (Transpose) позволила объединить исходные заголовки таблиц в столбце 1 и заголовки исходных таблиц в таблице преобразования. Затем извлекались заголовки Target из объединенной таблицы преобразования. Путем добавления условного столбца исходные несоответствующие заголовки были заменены целевыми. Теперь, рассматривая значения в столбце Column1, можно заметить, что уже имеются нормализованные значения, которые пригодятся в качестве корректных названий имен столбцов. Пришло время транспонировать таблицу обратно, чтобы восстановить имена столбцов из Column1. Выполните следующие дополнительные шаги.

20. На вкладке Преобразование (Transform) выберите команду Транспонировать (Transpose) В результате выполнения транспонирования восстанавливаются заголовки из Column1, но теперь они сохранятся в первой строке таблицы, пришло время транспонировать их обратно, в качестве заголовков.

21. На вкладке Преобразование (Transform) выберите команду Использовать первую строку в качестве заголовков (Use First Row as Headers).

22. На панели Запросы (Queries) выберите запрос Appended Products и рассмотрите скомбинированную таблицу. Если заметите ошибку, удалите последний шаг Измененный тип (Changed Types) на панели Примененные шаги (Applied Steps), и все будет в порядке.

Файлы решений C04E04 - Solution - Transpose.xlsx и C04E04 - Solution - Transpose.pbix доступны на сайте https://aka.ms/DataPwrBIPivot/downloads.

К сожалению, метод транспонирования работает только при небольших наборах данных. Большие наборы данных с огромным количеством строк занимают много памяти. Power Query поддерживает таблицы с 16 384 столбцами. Если транспонировать таблицу с более чем 16 384 строками, то вы столкнетесь с ошибками, связанными с медленным импортом и высоким потреблением памяти на этапе преобразования этих строк в столбцы на шаге транспонирования. Даже если испробовать этот метод на 15 000 строках, потребление памяти и низкая частота обновления наводят на мысль о необходимости альтернативного метода. В таком случае потребуется более удобное решение для нормализации имен столбцов, которое окажется подходящим при обработке больших таблиц.

УПРАЖНЕНИЕ 4.5. Отмена свертывания, слияние и повторное сведение

Более удобный способ преобразования имен столбцов в столбец и настройки этапа объединения таблицы преобразования — операция отмены свертывания. В этом упражнении будет показано, как применить это преобразование и внести в него изменения с помощью нормализованных имен столбцов, созданных посредством повторного сведения.

См. также

В главе 6 рассматривается множество различных применений преобразования отме-ны свертывания столбца. В главе 7 описываются удобные сценарии использования столбца сведения.

Это упражнение выполняется сразу же после импорта файлов из папки, что и было выполнено в шагах 1-9 упражнения 4.4.

1. Начните с сохраненной рабочей книги или отчета Power BI из упражнения 4.4, шаг 9. Откройте файл в Excel или Power BI Desktop и запустите редактор Power Query.

2. На вкладке Добавление столбца (Add Column) выберите команду Столбец ин-декса (Index Column). На следующем шаге вы поймете, зачем необходим ин-декс.

3. Выделите столбец индекса, правой кнопки мыши щелкните на нем и в контекстном меню выберите команду Отменить свертывание других столбцов (Unpivot Other Columns). Power Query выравнивает таблицу, создав новую таблицу с тремя столбцами: Index (Индекс), Attribute (Атрибут) и Value (Значе-ние). В столбце Index сохраняется исходный идентификатор строки, что поможет на шаге 6 вернуть таблицу в исходное состояние. Столбец Attribute включает имя столбца, а столбец Value содержит исходные значения, которые имелись в таблице. Теперь можно выполнить обработку текста в столбце Attribute.

4. Переименуйте столбец Attribute в Column1, чтобы следующие шаги были по-следовательными.

5. Примените последовательность слияния, описанную в упражнении 4.4, шаги 1219 для замены несоответствующих имен столбцов.

6. Для отмены шага развертывания выберите команду New Column Names (Назва-ния нового столбца) и на вкладке Преобразование (Transform) щелкните мы-шью на кнопке Столбец сведения (Pivot Column).

7. В открытом окне диалога Столбец сведения (Pivot Column) выберите команду Значение (Value) в меню Столбец значений (Values Column) и откройте раздел Расширенные параметры (Advanced Options). Затем установите флажок Не аг-регировать (Don't Aggregate) и щелкните мышью на кнопке OK.

8. Удалите столбец Index (Индекс).

9. На панели Запросы (Queries) выберите запрос Appended Products и просмотрите скомбинированную таблицу. Если заметите ошибку, удалите последние шаги Измененный тип (Changed Types), и все будет сделано.

См. также

Преобразование "Столбец сведения" весьма удобно. В главе 7 будет показано, как можно улучшить неудовлетворительно структурированные наборы данных.

Можно просмотреть файлы решений C04E05 - Solution - Unpivot.xlsx и C04E05 - Solution - Unpivot.pbix, которые доступны на сайте https://aka.ms/DataPwrBIPivot/ downloads.

Команда Отменить свертывание столбцов (Unpivot) может применяться к большим наборам данных, а команда Столбец сведения (Pivot), предназначенная для возврата таблицы в исходное состояние, может потребовать большого объема памяти и значительно замедлить время обновления, поскольку Power Query должен полностью рассмотреть набор данных для идентификации уникальных значений

в столбце Column1 и затем сформировать новые столбцы из этих уникальных значений. В результате этот метод не идеален при наличии больших наборов данных, а также при необходимости объединять таблицы оптимизированным способом, с сокращенным временем их обновления.

УПРАЖНЕНИЕ 4.6.

Транспонирование лишь имен столбцов

Оптимизированная по производительности методика преобразования имен столбцов в столбец при установке этапа объединения таблицы преобразования состоит из транспонирования лишь имен столбцов (в качестве промежуточного этапа преобразования) и процедуры объединения. На рис. 4.8 показано, как можно выполнить транспонирование имен столбцов, их нормализацию, а также обратное преобразование имен столбцов в исходную таблицу.

Теоретически метод, показанный на рис. 4.8, реализовать несложно. Однако при объединении нескольких конкретных файлов, которое реализуется с помощью команды Объединить запросы (Append Queries), применить этот метод к папке довольно проблематично, поскольку потребуются пользовательские функции. В связи с этим применять такой подход рекомендуется только в учебных целях.

Данная методика целесообразна, если методы, описанные в упражнениях 4.4 и 4.5, не оправдывают себя из-за ограничений производительности, а методика, основанная на применении языка М, описанная далее в упражнении 4.7, тоже не подходит.

1. Начните с открытия сохраненных запросов, созданных на шагах 1-9 упражне-ния 4.4.

2. На панели Примененные шаги (Applied Steps) при выбранном запросе Products Sample удалите шаг Повышенные заголовки (Promoted Headers). Теперь заго-ловки таблицы примеров находятся в первой строке.

3. На панели Запросы (Queries) щелкните правой кнопкой мыши на запросе Products Sample и в контекстном меню выберите команду Дублировать (Duplicate). Теперь, при выбранном новом запросе, Products Sample (2), выпол-ните команду Сохранить строки (Keep Rows) на вкладке Главная (Home) и за-тем выберите команду Сохранить верхние строки (Keep Top Rows). В окне диалога Сохранить верхние строки (Keep Top Rows), в поле Количество строк (Number of Rows) введите 1 и щелкните на кнопке OK.

ПРИМЕЧАНИЕ

При объединении файлов из папки Power Query создает функцию запроса, пример запроса, запрос файла и параметр. Эти объекты помогают объединить несколько файлов из папки на основе одного выбранного файла. Основное преобразование для каждого файла реализовано в виде функции. Для настройки преобразования каждого файла можно внести изменения в пример запроса. В этом упражнении пример запроса — это Transform Sample File from Appended Products (Преобразование примера файла из объединенных товаров), и он переименован в Products Sample (Пример товаров). Изменения, произведенные в примере запроса, распространятся на функцию и, как результат, применяются ко всем файлам в папке. Имеется одна важная оговорка: если в примерах запросов применяется команда Дублировать (Dublicate) или Ссылка (Reference), то преобразования в новом запросе не будут распространяться с помощью Power Query на объединение файлов из папки (если только не задействована усовершенствованная методика "обратной привязки" нового запроса к последовательности преобразований из примера запроса). Данный подход применяется на шагах 21 и 22.

4. На вкладке Преобразование (Transform) выберите команду Транспонировать (Transpose) для подготовки почвы для нормализации. Именно с этого момента будет применяться последовательность слияния, аналогичная описанной в шагах 12-19 упражнения 4.4. Однако теперь эти шаги выполнять не следует. Есть несколько вариантов, которые имеют решающее значение.

5. При выбранном запросе Products Sample (2) выберите команду Объединить запросы (Merge Queries), находящуюся на вкладке Главная (Home).

6. В открытом окне диалога Слияние (Merge) выберите столбец Column1 в Products Sample (2), в раскрывающемся меню Products Sample (2) выберите пункт Header\_Conversion. Затем выберите столбец Source, как это выполнено в упражнении 4.4 (рис. 4.6).

7. Удостоверьтесь, что в раскрывающемся меню Тип соединения (Join Kind) вы-брана команда Внешнее соединение слева (все из первой таблицы, совпадающие из второй) Left Outer (All from First, Matching from Second)), и щелкните мышью на кнопке OK. В окне редактора Power Query появится новый столбец Header\_Conversion с объектами Table (Таблица) как значениями.

8. На этом этапе нужно добавить столбец индекса, чтобы на следующем шаге не изменялся порядок имен столбцов в Column1, поэтому на вкладке Добавление столбца (Add Column) выберите команду Столбец индекса (Index Column). На шаге 13 будет пояснена важность этого шага.

9. Выберите столбец Header\_Conversion и выполните команду Развернуть (Expand), находящуюся на вкладке Преобразование (Transform).

10. В панели Развернуть (Expand) отмените установку флажка Источник (Source) и щелкните на кнопке OK. Столбец преобразуется в новый столбец Header\_Conversion.Target, где будут значения Target в строках, для которых имена столбцов товара не совпадают, и нулевые значения в строках, где следует сохранять исходные имена столбцов для товаров.

11. На вкладке Добавление столбца (Add Column) щелкните мышью на кнопке Условный столбец (Conditional Column).

12. В открытом окне диалога Добавление условного столбца (Add Conditional Column) выполните следующие настройки:

а. В поле Имя нового столбца (New Column Name) введите заголовок New Column Names.

б. В поле Имя столбца (Column Name) выберите пункт Header\_Conversion. Tarhet.

в. В поле Значение (Value) введите null.

г. В левом раскрывающемся меню Вывод (Output) выберите значение Выбе-рите столбец (Select a Column).

д. В правом раскрывающемся меню Вывод (Output) выберите параметр Column1.

е. В левом раскрывающемся меню В противном случае (Otherwise) выберите опцию Выберите столбец.

ж. В правом раскрывающемся меню В противном случае выберите пункт Header\_Conversion.Target и щелкните мышью на кнопке OK.

13. Убедитесь, что значения в столбце индекса находится все еще в порядке возрастания. Если это не так, можно отсортировать столбец Index (Индекс) в порядке возрастания. Порядок имен столбцов имеет в данном случае решающее значение. На шаге 9 была развернута таблица, этот шаг приводит к изменению порядка строк. Наличие столбца Index, добавленного на шаге 8, поможет сохранить порядок.

14. Удалите столбцы Column1, Header\_Conversion.Target и Index (Индекс), со-храните лишь последний столбец, New Column Names.

15. На вкладке Преобразование (Transform) выберите команду Transpose (Транс-понировать). В результате получим нормализованные имена столбцов.

На этом завершена часть нормализации запроса Product Sample (2), который показан на рис. 4.4 (шаг 5 на рис. 4.8). Начиная с этого места, можно завершить оставшуюся последовательность шагов преобразования.

16. Чтобы удалить прежние заголовки из таблицы, вернитесь к исходному запросу Products Sample. При выделенном запросе Products Sample на панели Запросы (Queries) выберите команду Удаление строк (Remove Rows), находящуюся на вкладке Главная (Home), затем выполните команду Удаление верхних строк (Remove Top Rows). В окне диалога Удаление верхних строк, в поле Количество строк введите 1 и щелкните на кнопке ОК.

17. При выбранном запросе Products Sample на панели Запросы (Queries) выбери-те команду Объединить запросы (Append Queries), находящуюся на вкладке Главная (Home). Откроется диалоговое окно Слияние (Append). Выберите за-прос Products Sample (2) в качестве добавляемой таблицы и щелкните на кноп-ке OK.

18. Теперь на панели просмотра видно, что нормализованные имена столбцов из запроса Products Sample (2) теперь размещаются в нижней строке, поскольку запрос Products Sample (2) добавлен в виде второй таблицы. Можно подключить эти таблицы. После этого в строке формул появится следующая формула:

= ТаЫе.СотЬ1пе({#"Уда.пенные верхние строки ", #"Products Sample (2)"})

Измените эту формулу, переставив местами #"Удаленные верхние строки" и #"Products

Sample (2) ":

= Table.Combine({#"Products Sample (2)", #"Удаленные верхние строки "})

19. Теперь, при наличии нормализованных имен столбцов в первой строке, повышайте их как заголовки. Для этого на вкладке Преобразование (Transform) щелкните на кнопке Использовать первую строку в качестве заголовков (Use First Row As Headers) и на панели Примененные шаги (Applied Steps) удалите шаг Измененный тип (Changed Type).

ПРИМЕЧАНИЕ

В данном случае имеет значение удаление шага Измененный тип. Столбец Size в таблице Accessories содержит только цифры. Следовательно, Power Query преобразовал тип столбца Size в Целое число (Whole Number). Но для других товаров, таких как велосипеды, допустимы текстовые значения. Если не удалить преобразование типов на этом этапе, то в ячейках столбца Size, которые должны включать текстовые значения, появятся ошибки.

20. На панели Запросы (Queries) выберите запрос Appended Products, чтобы про-смотреть объединенную таблицу. Если вы заметили ошибку, устраните ее удалением последнего шага, Измененный тип, на панели Примененные шаги (Applied Steps).

Теперь, при прокрутке панели просмотра и просмотре строк из C04E04 - Bikes.xlsx, можно обнаружить, как показано на рис. 4.9, что значения в столбце Name поменялись местами со значениями в столбце Product Number. Нормализация не сработала.

Рис. 4.9. В запросе Appended Products коды товаров и названия товаров поменялись местами в строках, относящихся к велосипедам

Г") СОВЕТ

Проверка конечных результатов — важный шаг в использовании Power Query. Панель просмотра нередко может вводить в заблуждение, поскольку показывает частичные данные. В этом случае все отлично работало с данными, относящимися к таблице Accessories, но если внимательно посмотреть на добавленные результаты, то можно обнаружить, что значения идентификатора товара и названия товара в таблице Bikes поменялись местами. Всегда внимательно изучайте результаты.

Рассмотрим, что же произошло и почему нормализация не сработала. Хорошая новость заключается в том, что запрос Products Sample (2) теперь работает с одним файлом. Выбирая запрос Products Sample (2), можно заметить, что на последнем шаге в панели Примененные шаги (Applied Steps) заголовки нормализуются, как и ожидалось. Но плохо то, что нормализация заголовка выполнена только для одного файла, и теперь, когда этот пример запроса повторно используется через пользовательскую функцию для всех файлов в папке, все без исключения имена столбцов с данными преобразуются в один файл.

Почему используется один файл? Посмотрите на запрос Product Sample (2). Как только шаг Source (Источник) будет выбран на панели Примененные шаги (Applied Steps), в строке формул показано, что запрос загружает книгу Excel с помощью параметра # "Параметр файла примера1". Этот параметр был создан Power Query при объединении файлов из папки. Этот параметр можно выбрать на панели Запросы (Queries) для определения того, что текущим значением параметра является Пример файла (Sample File). Можно найти запрос Пример файла под названием #"Параметр файла примера1" на панели Запросы (Queries). Этот запрос возвращает содержимое рабочей книги C04E04 - Accessories.xlsx.

Почему значения кодов товара и названия товаров поменялись местами для Bikes в запросе Appended Products? Так произошло потому, что таблица Accessories, файл C04E04 - Accessories.xlsx, содержит два столбца, которые следуют в обратном порядке по сравнению с файлом C04E04 - Bikes.xlsx. Если порядок столбцов отличается, нельзя выполнить нормализацию для таблицы Accessories и применить ее к другим файлам.

Для решения этой проблемы необходимо научиться повторно использовать логику запроса Products Sample (2) для каждого файла в папке. Вспомните из шага 3 этого упражнения, что запрос Products Sample (2) является копией запроса Products Sample, который служит примером запроса. В качестве примера запрос Products Sample запускается для каждого файла в папке и запускает Products Sample (2). Но поскольку Products Sample (2) в настоящее время не является примером запроса, файлы из папки не передаются в Products Sample (2). Этот запрос всегда выполняет последовательность преобразования в одном и том же файле: C04E04 - Accessories.xlsx.

Необходимо найти способ преобразования Products Sample (2) в повторно ис-пользуемый запрос, который воспримет файл в качестве параметра из запроса Products Sample и нормализует заголовки для каждого файла. Для решения этой проблемы можно превратить запрос Product Sample (2) в функцию (также называемую пользовательской функцией), которую можно вызывать повторно для каждого файла в папке. Выполните следующие дополнительные шаги для создания функции из запроса Products Sample (2):

21. На панели Запросы (Queries) щелкните правой кнопкой мыши на запросе Products Sample (2) и выберите команду Создать функцию (Create Function). Откроется окно диалога Создать функцию (Create Function). В поле Имя функции (Function Name) введите FnNormalizeColumnNames и щелкните мы-шью на кнопке OK. Как показано на рис. 4.10, теперь мы располагаем новой функцией, которую можно применять ко всем файлам.

Теперь имеем функцию, FnNormalizeColumnNames, которую можно вызывать из запроса Products Sample вместо вызова запроса Products Sample (2).

22. На панели Запросы (Queries) выберите запрос Products Sample. На панели Примененные шаги (Applies Steps) выберите шаг Объединенный запрос (Appended Query). В строке формул появится формула:

= Table.Combine({#"Products Sample (2)", #"Удаленные верхние строки "})

замените ее на следующую формулу:

= Table.Combine({FnNormalizeColumnNames(#"napaMeTp файла примера1"), #"Удаленные верхние строки "})

Рис. 4.10. Запрос Products Sample (2) преобразован в функцию FnNormalizeColumnNames с Products Sample (2) в качестве примера запроса

Функция FnNormalizeColumnNames получает содержимое рабочей книги Excel из папки в качестве аргумента и возвращает нормализованные столбцы. Шаг #"Удаленные верхние строки" — последний шаг запроса Products Samples перед применением добавления. Теперь таблица включает оригинальные строки без заголовка.

Почему здесь применялся параметр #"Параметр файла примера1"? Напомним, что этот параметр всегда указывал на файл Accessories, когда вызывался в запросе Sample Products (2). Теперь, в контексте функции, этот параметр переназначается при использовании корректного файла, поскольку Power Query выполняет итерации по файлам в папке.

23. Вернитесь к добавленным результатам в запросе Appended Products и прокру-тите вниз панель предварительного просмотра, чтобы убедиться в том, что все столбцы таблицы C04E04 - Bikes.xlsx выровнены правильно. Закройте редактор Power Query и загрузите данные в отчет.

Можно загрузить файлы решения C04E06 - Solution - Transpose Headers.xlsx и C04E06 - Solution - Transpose Headers.pbix с сайта https://aka.ms/DataPwrBIPivot/ downloads.

УПРАЖНЕНИЕ 4.7. Применение языка M для нормализации имен столбцов

В данном упражнении показан наиболее эффективный метод применения таблицы преобразования и нормализации имен столбцов. Здесь используется язык M и применяется преобразование непосредственно к именам столбцов.

Напомним, что в упражнении 4.3 осуществлялась простая манипуляция с текстом, относящаяся к именам столбцов, т. е. применялась функция языка M Table. TransformColumnNames с подходящей текстовой операцией в качестве второго аргумента. Например, для написания заглавными буквами каждого слова был задан аргумент Text.Proper.

В этом упражнении по нормализации имен столбцов формируется пользовательская функция, которая нормализует имена столбцов в соответствии с правилами из таблицы преобразования.

Прежде чем создавать функцию, необходимо осознать одно изменение преобразования, которое упрощает код и ускоряет поиск в таблице преобразования. Напомним, ранее таблица преобразования была сформирована для пар значений Source (Источник) и Target (Цель). Транспонируйте эту таблицу и используйте первую строку в качестве заголовков. Таким образом, новая таблица преобразования включит значения Source в качестве заголовков и значения Target в качестве первой строки, как показано на рис. 4.11.

Рис. 4.11. Транспонирование исходной таблицы преобразования пар Source/Target в таблицу из одной строки может служить эффективным методом нормализации имен столбцов

Это преобразование позволяет находить исходные значения как заголовки в таблице преобразования и получать их целевые значения в строке с помощью простой пользовательской функции. Для создания транспонированной таблицы преобразования начните с сохраненных запросов из шагов 1-9 упражнения 4.4.

1. Откройте рабочую книгу или отчет Power BI и запустите редактор Power Query.

2. На панели Запросы (Querues) выберите запрос Header\_Conversion.

3. На вкладке Преобразование (Transform) выберите команду Транспонировать (Transpose), затем выполните команду Использовать первую строку в качестве заголовков (Use First Row As Headers).

4. На панели Примененные шаги (Applied Sreps) удалите шаг Измененный тип, созданный автоматически. В нем нет необходимости, и если продолжать выполнять этот шаг, любые изменения, внесенные в значения Source в таблице преобразования, могут привести к ошибкам обновления в будущем.

5. На панели Запросы (Queries) выберите запрос Products Sample и щелкните мышью на значке fx, как проиллюстрировано в шаге 8 из упражнения 4.3.

6. Измените формулу в строке формул следующим образом:

= #"Повышенные заголовки "

на следующую:

= Table.TransformColumnNames(#"noBbmeHHHe заголовки",

each try Table.Column(Header\_Conversion, \_){0} otherwise \_)

7. Нажмите клавишу <Enter> и просмотрите заголовки в панели просмотра. Они нормализованы. Теперь можно выбрать запрос Appended Products и увидеть, что все несоответствующие таблицы объединены правильно.

Кратко рассмотрите предыдущее выражение. В упражнении 4.3 уже использова-лась формула Table.TransformColumnNames. В данном случае обратим внимание на второй аргумент:

each try Table.Column(Header\_Conversion, \_){0} otherwise \_

В главе 9 подобные выражения будут описаны подробнее. Комбинация составляющих each и \_ используется в качестве ярлыка для определения функции. В этом случае Table.TransformColumnNames требуется функция в качестве второго аргумента, и с помощью each и символа подчеркивания выполняется вызов этой функции для каждого имени столбца.

Рассмотрим базовый синтаксис и исследуем вышеприведенное выражение шаг за шагом. Для получения доступа к ячейке первой строки в одном из столбцов таблицы преобразования указан параметр {0}. Помните, что в языке M все основано на индексах, начинающихся с нуля, поэтому для доступа к первой ячейке задан нуль вместо единицы. Фигурные скобки служат для доступа к элементу в списке по его нулевому индексу.

Допустим, столбец Column1 находится в таблице Header\_Conversion. Для получения доступа к списку значений в Column1 применяется следующая формула:

Header\_Conversion[Column1]

В этом случае имя столбца становится входным параметром для определенной функции с помощью символа подчеркивания. Для получения доступа к каждому из столбцов (в виде списков) по их именам в таблице Header\_Conversion можно написать следующую функцию:

each Table.Column(Header\_Conversion, \_)

Заметим, что здесь применяется функция Table.Column. Следующая формула не будет работать:

each Header\_Conversion[\_]

С помощью этой формулы механизм языка M выполнит поиск столбца, действи-тельное название которого представляет собой символ подчеркивания \_. В результате получим следующую ошибку:

Expression.Error: The column '\_' of the table wasn't found (Столбец ‘\_' таблицы не найден).

Чтобы избежать появления подобной ошибки, применим функцию Table.Column, которая получает таблицу и имя столбца и возвращает нужный столбец в виде списка. Комбинируя функции, можно получить формулу для доступа к значению "Цель" из столбца "Источник" в таблице транспонированных преобразований, созданной на шаге 3:

each Table.Column(Header\_Conversion, \_){0}

Итак, почему же появились элементы try/otherwise на шаге 6? Напомним, что таблица преобразования ранее была предназначена для хранения не всех возможных имен столбцов, а только несоответствующих имен столбцов.

Поэтому выражение на шаге 6 включает элементы try/otherwise, которые выделены полужирным шрифтом:

each try Table.Column(Header\_Conversion, \_){0} otherwise \_

Если значение Source не обнаружено в таблице Header\_Conversion, часть try потерпит неудачу, запуская другую часть, которая вернет значение Source. Здесь снова приведена функция полностью:

= Table.TransformColumnNames(#"noBbmeHHHe заголовки",

each try Table.Column(Header\_Conversion, \_){0} otherwise \_)

Можно прочитать ее как следующую инструкцию на естественном языке: "Возьмите таблицу, приведенную на шаге Повышенные заголовки, и преобразуйте имена столбцов по следующему правилу: для каждого столбца найдите его в качестве имени столбца в таблице преобразования и верните соответствующее значение в ячейку таблицы преобразования. Если этого сделать невозможно (значит, такого имени столбца в таблице преобразования нет), то верните имя столбца Повышенные заголовки в качестве выходных данных".

Если не все детали этого раздела вам понятны, не беспокойтесь. По-прежнему для объединения несовпадающих таблиц с таблицей преобразования можно применять три других метода, описанных в упражнениях 4.4, 4.5 и 4.6.

Цель упражнения 4.7 — продемонстрировать, как можно быстро комбинировать несовпадающие таблицы, используя язык M. Желательно вернуться к этому упражнению после того, как вы познакомитесь с главой 9 и больше узнаете о языке M.

Файлы решений C04E07 - Solution - M.xlsx и C04E07 - Solution - M.pbix доступны на сайте https://aka.ms/DataPwrBIPivot/downloads.

Резюме

В этой главе рассказано о последствиях объединения несовпадающих таблиц и представлены различные методики, которые применяются для сопоставления имен столбцов и успешного объединения нескольких таблиц.

Было продемонстрировано, как выявлять пропущенные значения и разбивать данные в несовпадающих таблицах, которые объединены неудовлетворительно. Также рассмотрено, каким образом до добавления таблицы можно вручную переименовывать столбцы в соответствии с заданными требованиями.

Если несколько таблиц в папке имеют общий порядок столбцов, показано, как применять метод обобщения заголовка для объединения таблиц. Также рассмотрены свойства функции Table.TransformColumnNames, эта функция применялась для простой нормализации имени столбца, такой как использование заглавных букв в каждом слове или замена подчеркивания пробелами.

Таблица преобразования помогает нормализовать имена столбцов в тех случаях, когда манипулирование текстом с помощью одного правила не приводит к нужному результату. Показано, каким образом объединить таблицу преобразования для замены несоответствующих имен столбцов, рассмотрены три метода: транспонирование, раскрытие и транспонирование только имен столбцов. Наконец, приведен наиболее эффективный способ нормализации имен столбцов: с помощью расширенного M-кода.

В следующей таблице рассмотрены различные методики, их плюсы и минусы:

Методика Плюсы Минусы

1. Обобщение заголовка Простая реализация. Не снижается производительность При изменении порядка между столбцами этот метод приведет к обмену значениями

2. Транспонирование, объединение и обратное транспонирование Простая реализация. Может обрабатывать противоречивый порядок следования столбцов Чрезмерное потребление памяти. Сбой, если одна из комбинированных таблиц достигнет 16 384 строк

3. Отмена развертывания, объединение и сведение столбца Простая реализация. Может работать с большими наборами данных. Может обрабатывать противоречивый порядок следования столбцов Значительно медленнее, чем методы 1, 4 и 5

4. Транспонирование только имен столбцов Может обрабатывать несоответст-вующий порядок следования столбцов. Обновление выполняется значительно быстрее, чем при использовании методов 2 и 3 Наибольшее время реализации

5. Использование языка M для нормализации имен столбцов Наиболее эффективный метод с точки зрения производительности. Быстрое время внедрения Необходимо знакомство с языком М

Возможность объединять несовпадающие таблицы очень полезна на практике. Представьте, что вы работаете в корпорации, которая постоянно приобретает новые компании. Методы, описанные в этой главе, могут применяться для быстрого объединения различных наборов данных каждой из приобретенных компаний и получения информации для консолидации IT-данных. Сценарии самообслуживания BI могут исключительно хорошо проявить себя, когда дело доходит до объединения подобных изолированных наборов данных.

Глава 5

Сохранение контекста

Для меня контекст — это ключ, отсюда и понимание всего.

- Кеннет НоланД

В ЭТОЙ ГЛАВЕ ВЫ...

■ Узнаете, каким образом при добавлении запросов сохранять контекст из имен файлов и листов.

■ Познакомитесь, как применить детализацию для преобразования заголовков и динамического сохранения их содержимого в таблицах.

■ Научитесь сохранять заголовки на этапе предварительного добавления, используя детализацию и пользовательский столбец для объединения нескольких таблиц из папки.

■ Узнаете, как сохранить заголовки после добавления, задействуя детализацию, условные столбцы и преобразование мгновенного заполнения для объединения нескольких листов из рабочей книги.

■ Изучите "продвинутые" методики, применяемые для сохранения определенных ячеек из сетки с помощью индексов и меток близости.

В главах 3 и 4 была устранена проблема, возникающая при объединении таблиц, мы также близко подобрались к некоторым реальным сценариям по обработке несоответствующих блоков данных, которые перед объединением требовали нормализации. Зачастую решающее значение для анализа имеет ключевая контекстная информация, которая сохраняется вне таблиц. Объединение таблиц без контекста часто приводит к лишенным смысла результатам. В данной главе будет показано, каким образом можно сохранять контекст таблицы и как объединять отдельные таблицы с их контекстной информацией. В главе рассматривается контекст таблицы для различных сценариев и показано, каким образом можно динамически обнаруживать и извлекать ключевые элементы контекста из данных, игнорируя при этом ненужные элементы. В конце концов, как отметил Кеннет Ноланд (хотя, вероятно, в совершенно ином случае), именно с контекстом приходит понимание всех процессов. Без этого можно упустить важные идеи.

Сохранение контекста

в именах файлов и листах

В главах 3 и 4 были изучены различные методики, применяемые для объединения нескольких таблиц. Продемонстрированы методики обработки данных о товарах, применяемые фиктивной компанией AdventureWorks. В упражнениях в главах 3 и 4 имелись отдельные таблицы для родительских категорий товаров, таких как Bikes (Велосипеды) и Accessories (Аксессуары). К счастью (отчасти намеренно), родительская категория представлялась в виде столбца данных. Поэтому при объединении таблицы родительская категория оставалась без изменений. В данном разделе рассматривается общий сценарий, когда важная контекстная информация доступна в виде наименования листа или имени файла, но не в виде столбца в таблице. Цель рассмотрения — найти способы включения этих данных в виде столбца для сохранения контекста. Начнем с простого примера.

УПРАЖНЕНИЕ 5.1. Часть 1. Методика, предусматривающая применение настраиваемого столбца

В данном упражнении импортируются записи о товарах компании из двух роди-тельских категорий Bikes и Accessories, расположенных в двух отдельных книгах, которые сводятся вместе. Добавленный результат не включает названий категорий товаров, и поэтому сравнивать характеристики двух линий товаров невозможно. В упражнении будет показано, каким образом можно в добавленных результатах легко сохранять значения родительских категорий.

Для начала загрузите файлы C05E01 - Bikes.xlsx и C05E01 - Accessories.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads.

1. Загрузите каждую рабочую книгу в новую пустую книгу или отчет Power BI. Для каждого файла выберите вкладку Sheet1 в окне диалога Навигатор (Navigator) перед его загрузкой. Затем, если открыт редактор Power Query, можно заметить, что запросы называются Sheet1 и Sheet1 (2), как показано на рис. 5.1. Информация, относящаяся к родительской категории, недоступна в виде столбца.

2. Переименуйте лист Sheet1 в книге Bikes и Sheet1 (2) в книге Accessories. Если вы не помните, какой из запросов должен быть Bikes, а какой — Accessories, можно выбрать первый шаг на панели Примененные шаги (Applied Steps) и найти соответствующую родительскую категорию, указанную в строке формул как имя файла. Например, вот формула для таблицы Bikes:

= Excel.Workbook(File.Contents("C:\Data\C05\C05E01 - Bikes.xlsx"), null, true)

Кроме того, можно выбрать значок настроек в правой части шага Источник (Source) панели Примененные шаги (Applied Steps) для каждого запроса. В открывшемся окне диалога Excel можно посмотреть настройку Путь к файлу (File Path) и определить, является ли выбранный запрос Bikes или Accessories.

v I Sheet! - Редактор Power Query

3. Выберите запрос Bikes на панели Запросы. На вкладке Главная (Home) в рас-крывающемся меню Объединить запросы (Append Queries) выберите команду Объединить запросы в новый (Append Queries As New).

4. В окне диалога Добавление (Append) выберите таблицу Bikes в качестве пер-вичной таблицы и Accessories — в качестве таблицы, которую нужно добавить к первичной таблице, и щелкните мышью на кнопке OK.

Новый запрос Append1 содержит данные, относящиеся к товарам, которые приведены в таблицах Bikes и Accessories, но родительская категория утеряна. Переименование исходных запросов Sheet1 и Sheet1 (2) в Bikes и Accessories, соответственно, на шаге 2 не является достаточным для сохранения контекста, но эту ситуацию можно исправить.

5. Выберите запрос Bikes на панели Запросы.

6. На вкладке Добавление столбца (Add Column) выберите команду Настраиваемый столбец (Custom Column).

7. После открытия окна диалога Настраиваемый столбец (Custom Column) в поле Имя нового столбца (New Column Name) введите Parent Category (Родительская категория), а в окне Пользовательская формула столбца (Custom Column Formula) введите формулу:

Bikes

8. Щелкните мышью на кнопке OK для закрытия окна диалога Настраиваемый столбец (Custom Column) и выберите запрос Accessories на панели Запросы (Queries).

9. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column).

10. После открытия окна диалога Настраиваемый столбец (Custom Column) в поле Имя нового столбца (New Column Name) введите Parent Category (Родительская категория), а в окне Пользовательская формула столбца (Custom Column Formula) введите формулу:

= "Accessories"

11. Щелкните мышью на кнопке OK для закрытия окна диалога Настраиваемый столбец (Custom Column) и выберите запрос Append1 на панели Запросы (Queries). При прокрутке вниз панели предварительного просмотра можно увидеть новый столбец, Parent Category, включающий правильный контекст для строк из Bikes и Accessories.

Книгу с готовым решением C05E01 - Solution.xlsx и отчет Power BI C05E01 - Solution.pbix вы можете загрузить с сайта https://aka.ms/DataPwrBIPivot/ downloads.

УПРАЖНЕНИЕ 5.1. Часть 2. Обработка контекста, состоящего из имен файлов и листов

Как правило, при использовании команд Объединить запросы (Append Queries) и Объединить запросы в новый (Append Queries As New) можно утратить кон-текст, доступный в именах запросов, и если вы поступите таким образом, следует добавить предварительный настраиваемый столбец для каждой таблицы. При объединении нескольких файлов из папки, если имена файлов содержат контекст, можно найти необходимый контекст в столбце Source.Name и пользовательский столбец добавлять не нужно. Этот подход продемонстрирован в главе 3: год выпуска товаров AdventureWorks был сохранен (см. раздел "Добавление таблиц из папки" главы 3).

При работе с контекстом, находящимся в именах листов, имеется два способа для сохранения этой информации. Если число листов невелико и их имена не должны изменяться, то можно импортировать каждый из отдельных рабочих листов и для сохранения контекста воспользоваться методикой, основанной на применении настраиваемых столбцов, описанной в упражнении 5.1, часть 1.

Теперь можно загрузить книгу C05E01 - Bikes & Accessories.xlsx и начать с ней работу. Импортируйте каждый лист и добавляйте запросы Bikes и Accessories с помощью команды Объединить запросы в новый (Append Queries As New). Без добавления настраиваемого столбца для каждого запроса сохранить родительскую категорию нельзя.

Если нужно добавить много листов из книги при сохранении имен рабочих листов, то можно применить метод, описанный в упражнениях 3 и 4 из главы 3. Далее будет показано, каким образом можно применить эту методику для файла C05E01 - Bikes & Accessories.xlsx.

1. Откройте пустую новую рабочую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data) и в окне диалога Получить данные (Get Data) вы-полните команды Файл (File) Excel.

2. Выберите файл C05E01 - Bikes & Accessories.xlsx и щелкните мышью на кнопке Импорт (Import).

3. В окне диалога Навигатор (Navigator) щелкните правой кнопки мыши на эле-менте книги и в контекстном меню выберите команду Изменить (Edit).

4. В редакторе Power Query сохраните столбцы Name (Название) и Data (Данные), а также извлеките информацию из столбца Data. Благодаря столбцу Name можно сохранить контекст родительской категории.

5. Используйте методику, описанную в упражнении 3.4, для обработки имен столбцов, встроенных в строки.

Книгу решения C05E01 - Solution 2.xlsx или отчет Power BI C05E01 - Solution 2.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Предварительное добавление

для сохранения заголовков

Во второй части упражнения 5.1 объединялись таблицы Bikes и Accessories при сохранении атрибутов родительской категории из имен файлов или листов. К сожалению, в реальной жизни приходится сталкиваться со многими ситуациями, в которых ключевая контекстная информация сохраняется вне таблиц, а не в виде имен файлов или листов.

Обратите внимание, что в упражнении 5.1, часть 1, перед добавлением таблиц Bikes и Accessories в каждую таблицу добавлялся настраиваемый столбец, включающий значения родительской категории. Теперь рассмотрим расширенный вариант этой методики по сохранению заголовков, которые доступны в источнике данных перед строками таблицы.

Этот метод называется методом предварительного добавления, поскольку применяется перед этапом добавления. Техника предварительного добавления очень полезна при добавлении нескольких таблиц с помощью команд Объединить запросы (Append Queries) или Объединить запросы в новый (Append Queries As New). Также эта методика очень полезна при объединении нескольких файлов из папки, как будет показано позже. На рис. 5.2 проиллюстрированы шаги преобразования, необходимые для извлечения информации из заголовка в сетке и объединения ее в качестве нового столбца.

ПРИМЕЧАНИЕ

В отличие от метода, описанного в упражнении 5.1, часть 1, где в качестве нового столбца для каждого запроса жестко закодированы ключевые слова Bikes и Accessories, здесь необходим универсальный способ извлечения значения из заголовка, поскольку обычно значение в заголовке неизвестно, причем нужен динамический метод для работы со многими таблицами с разными заголовками.

УПРАЖНЕНИЕ 5.2. Сохранение заголовков

с помощью детализации

При выполнении упражнения будем следовать шагам, показанным на рис. 5.2, что позволит извлечь заголовок Bikes из ячейки Excel и включить его как новый столбец в таблицу Bikes. Затем покажем, как можно обобщить эту методику и применить ее к нескольким файлам, прежде чем добавлять все таблицы из папки.

ПРИМЕЧАНИЕ

Данный метод также может применяться с текстовыми и CSV-файлами в папке, если контекст сохраняется в виде строки перед строками с данными.

Для выполнения данного упражнения загрузите файл рабочей книги C05E02 - Bikes.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads.

1. Откройте новую пустую книгу или новый отчет Power BI.

В Excel: выберите вкладку Данные (Data) и выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data), в окне диалога Получить данные (Get Data) вы-полните команды Файл (File) Excel.

2. Выберите файл C05E02 - Bikes.xlsx и щелкните на кнопке Импорт (Import).

3. Выберите лист Sheet1 в окне диалога Навигатор (Navigator). Щелкните на кнопке Изменить (Edit), чтобы открыть окно редактора Power Query, и пере-именуйте запрос в Bikes.

4. На панели Примененные шаги (Applied Steps) удалите шаг Измененный тип (Changed Type).

ПРИМЕЧАНИЕ

Это действие не является обязательным, но на данном этапе применять автоматическую команду Измененный тип бессмысленно, поскольку Power Query не может правильно распознать типы столбцов таблицы товаров, первые строки которых содержат заголовок и пустые строки.

5. На вкладке Главная (Home) выберите команду Удалить строки (Remove Rows), затем выберите команду Удалить пустые строки (Remove Blank Rows) для удаления пустых строк из импортируемых данных. Почему этот шаг столь важен? Поскольку обычно имеются пустые строки, отделяющие заголовок от реальных таблиц. Удалив эти строки, значительно легче позднее импортировать строки таблицы товаров, после их извлечения и сохранения родительской категории в заголовке при выполнении следующих шагов.

6. Правой кнопкой мыши щелкните на запросе Bikes, находящемся на панели Запросы (Queries), и в контекстном меню выберите команду Ссылка (Reference). Это приведет к созданию запроса Bikes (2). Выберите этот запрос и обратите внимание, что первая ячейка в первой строке содержит значение Bikes. Щелкните правой кнопкой мыши на этой ячейке и выберите опцию Детализация (Drill Down). Это действие приведет к отображению слова Bikes на панели предварительного просмотра.

Оставьте отчет открытым для перехода к шагу 7 после детального объяснения в следующем разделе.

Применение детализации к ячейке таблицы

(объяснение с помощью M-формул)

После выбора на последнем шаге параметра Детализация (см. последний раздел) в строке формул появится следующая строка:

= Источник{0}[Со1итп1]

Данная формула языка M извлекает значение Bikes из таблицы. Здесь Источник — имя таблицы, предоставленной на предыдущем шаге. Также можете заметить, что Источник используется в качестве первого шага на панели Примененные шаги (Applied Steps).

Формула Источник {0} возвращает первую строку в таблице Источник (а Источник {1} возвращает вторую строку и т. д.). Индекс изменяется от 0 до n - 1 для доступа к первой или к n-й строкам. Вначале это может показаться сложным, но в этом легко разобраться. В большинстве сценариев можно просто применять пользовательский интерфейс и детализацию для доступа к отдельным ячейкам.

В Power Query строки представлены в виде записей. Запись является списком пар ключ/значение, где ключ — имя столбца, а значение — соответствующее значение в строке. Получить доступ к значению по заданному ключу в записи можно с по-мощью формулы запись [ключ], где запись — имя записи, а ключ — имя ключа. Следо-вательно, если формулой является Источник {0} [Column1], вы получаете доступ к первой строке в таблице Источник, а затем получаете доступ к значению в столбце Column1. На рис. 5.3 показано, как формула детализации Источник {0} [Column1] обращается к значению Bikes в два этапа: из таблицы в запись и затем из записи в ячейку.

СОВЕТ

Вовсе не обязательно знать синтаксис языка M для доступа к строкам, столбцам или ячейкам. Просто необходимо применять преобразование детализации и просматривать строку формул для получения напоминания о синтаксисе. Также следует учесть, что вы можете получить сначала доступ к столбцу, а затем — к строке. Например, формула Источник {0} [Column1] вернет то же значение, что и Источник [Column1] {0}. Последний вариант синтаксиса работает немного иначе: сначала возвращается список всех элементов в столбце Column1, а затем возвращается первый элемент в спи-

ске.

Объединение в запросе результатов детализации

Для продолжения упражнения 5.2 выполните следующие действия.

7. Теперь, когда известно, каким образом применяется детализация для извлечения ячейки, удалите запрос Bikes (2) и выберите запрос Bikes.

Рассмотрим немного иное применение детализации, теперь уже в применении к запросу Bikes.

8. Правой кнопкой мыши щелкните на ячейке, которая содержит значение Bikes, и в контекстном меню выберите команду Детализация (Drill Down).

9. В строке формул вы увидите следующую формулу:

= #"Удалены пустые строки"{0}[Со1итп1]

Теперь внимательнее рассмотрим синтаксис формулы. На этот раз формула применяется на шаге Удалены пустые строки, отображенном на панели При- меленные шаги (Applied Steps). Формат # "...", который в языке M называется "кавычки-идентификаторы", применяется для того, чтобы любая последовательность, состоящая из символов Юникода, включая пробелы и другие специальные символы, использовалась в формуле в качестве идентификатора.

Теперь на панели предварительного просмотра появился вывод "Bikes".

10. Заметим, что на панели Примененные шаги (Applied Steps) этот шаг отмечен как Column1. Для придания значения этому имени переименуйте его в Title (Заголовок). Для этого правой кнопкой мыши щелкните на шаге Column1 в панели Примененные шаги (Applied Steps) и в контекстном меню выберите команду Свойства (Properties). Затем в окне диалога Свойства (Properties) введите Title в поле Имя (Name) и закройте это окно диалога. (Окно диалога Свойства позволяет переименовывать шаги, добавлять описание и облегчать понимание логики преобразования ваших запросов.)

Теперь, при наличии заголовка, вернемся к таблице и очистим строку заголовка, как показано на рис. 5.2, шаг 3. Поскольку имеется заголовок на панели предварительного просмотра, нужно вернуться к предыдущему шагу еще до применения команды Детализация (Drill Down), сохраняя при этом шаг Title без изменений.

11. Щелкните мышью на значке fx в левой части строки формул и замените фор-мулу:

= Title

на следующую формулу:

= #"Удаленные пустые строки"

Этот подход позволяет пропустить шаг Title и вернуться к предыдущему шагу. Теперь на панели предварительного просмотра вместо слова Bikes отображается таблица. Но, как видите, шаг Title остался в разделе Примененные шаги (Applied Steps). Вы будете пользоваться этим шагом в ближайшее время.

12. Для удаления первой строки, как показано на шаге 3 на рис. 5.2, на вкладке Главная (Home) выберите команду Удалить строки (Delete Rows). Затем вы-берите команду Удаление верхних строк (Remove Top Rows). После открытия окна диалога Удаление верхних строк введите единицу в поле Количество строк (Number of Rows) и щелкните на кнопке OK для закрытия окна диалога.

ПРИМЕЧАНИЕ

Удаление заданного количества верхних строк (например, одной строки в текущем упражнении) не приветствуется. Далее в этой главе, в разделе "Использование подсказок контекста", будет показано, каким образом можно игнорировать контекст и удалять верхние строки, используя расположение таблицы в сетке, но не удалять жестко заданное число строк.

13. Следуя шагу 4 на рис. 5.2, повышайте имена столбцов товара так, чтобы они стали заголовками.

На вкладке Преобразование (Transform) выберите команду Использовать первые строки в качестве заголовков (Use First Rows As Headers).

14. И наконец, для реализации последнего шага, показанного на рис. 5.2, создайте настраиваемый столбец с Bikes как родительской категорией, обратившись ко вкладке Добавление столбца (Add Column) и выбирая команду Настраиваемый столбец (Custom Column). После открытия окна диалога Настраиваемый столбец (Custom Column) введите Parent Category в поле Имя нового столбца (New Column Name) и введите следующую формулу в окне Формула настраиваемого столбца (Custom Column Formula):

= Title

Затем щелкните на кнопке OK для закрытия окна диалога Настраиваемый столбец (Custom Column).

Теперь можно видеть, что был добавлен новый столбец Bikes (Велосипеды). Удалось динамически извлечь слово Bikes из ячейки, которая использовалась как заголовок рабочего листа, и выполнить шаги, показанные на рис. 5.2, чтобы сохранить его в таблице. Эта методика чрезвычайно полезна при решении реальных задач с данными, ее можно применять во многих сценариях, источниках данных и форматах данных, а не только к заголовкам в ячейках Excel.

ПРИМЕЧАНИЕ

Возможность разделить преобразования на два потока, а затем объединить в один поток является универсальной методикой, применяемой при работе в Power Query. По мере знакомства с языком M подобные манипуляции можно применять непосредственно в расширенном редакторе вместо формирования настраиваемых шагов, как это

выполнено в данном упражнении.

Чтобы просмотреть результаты выполнения упражнения, загрузите книгу решения C05E02 - Solution.xlsx и отчет Power BI C05E02 - Solution.pbix с сайта https:// aka.ms/DataPwrBIPivot/downloads.

УПРАЖНЕНИЕ 5.3. Сохранение заголовков из папки

При выполнении упражнения 5.2 был сохранен заголовок из одного листа. Можно обобщить эту методику и применить ее к нескольким рабочим книгам в папке.

1. Загрузите файл C05E03 - Products.zip с сайта https://aka.ms/DataPwrBIPivot/downloads.

2. Извлеките содержимое ZIP файла в новую папку: C:\Data\C05\C05E03 - Products\. Эта папка будет включать следующие файлы:

а. C05E03-01.xlsx;

б. C05E03-02.xlsx;

в. C05E03-03.xlsx;

г. C05E03-04.xlsx.

3. Просмотрите файлы и обратите внимание, что каждый из них содержит таблицу товаров с отличающимся названием, как показано на рис. 5.4. Файлам преднамеренно даны имена, лишенные смысла, поэтому единственный способ сохранить родительские категории продуктов в каждой книге — это извлечь заголовки.

4. Откройте пустую новую рабочую книгу или создайте новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните последовательность команд Получить данные (Get Data) Из файла (From File) Из папки (From Folder). В Power BI Desktop: на вкладке Главная страница (Home) выберите вкладку Получить данные (Get Data), а в окне диалога Получить данные (Get Data) выполните команды Файл (File) Папка (Folder).

5. В окне диалога Папка (Folder) выберите папку C:\Data\C05\C05E03 - Products\ и закройте это окно диалога.

6. В следующем окне диалога, которое называется так же, как папка, выбранная на шаге 5, в раскрывающемся меню Объединить (Combine) выберите параметр Объединить и изменить (Combine and Edit).

7. После открытия окна диалога Объединить файлы (Combine Files) в раскры-вающемся меню Пример файла (Example File) выберите любую команду, за-тем выделите лист Sheet1 и щелкните мышью на кнопке OK.

8. После открытия окна диалога редактора Power Query заметьте, что на панели просмотра в объединенной таблице товаров отсутствует контекст родительской категории. Опять же, это результат наличия бессмысленных имен файлов.

9. Выберите запрос Transform Sample File from C05E03 - Products и переиме-нуйте его на Products Sample.

10. На вкладке Главная (Home), при выбранном запросе Products Sample, выпол-ните команду Удалить строки (Remove Rows), затем задайте команду Уда-лить пустые строки (Remove Blank Rows) для применения первого шага, показанного на рис. 5.2.

11. Для извлечения заголовка из таблицы Bikes, как это выполнено в упражне-нии 5.2 (шаги 8-14), и сохранения всех заголовков в новом столбце Parent Category (Родительская категория) выполните следующие шаги для запроса Products Sample:

Правой кнопкой мыши щелкните на ячейке, которая включает значение Bikes, и в контекстном меню выберите команду Детализация (Drill Down).

В строке формул отобразится следующая формула:

= #"Удалены пустые строки"{0}[Со1итп1]

Обратите внимание, что на панели Примененные шаги (Applied Steps) этот шаг отмечен как Column1. Правой кнопкой мыши щелкните на шаге Column1, находящемся на панели Примененные шаги, и выберите пункт Свойства (Properties). Затем в окне диалога Свойства введите Title в поле Имя (Name) и щелкните на кнопке OK для закрытия этого окна диалога.

Щелкните на кнопке fx в левой части строки формул и замените формулу:

= Title

на формулу:

= #"Удалены пустые строки"

На вкладке Главная (Home) выберите команду Удалить строки (Remove Rows). Затем выберите команду Удаление верхних строк (Remove Top Rows). После открытия окна диалога Удаление верхних строк (Remove Top Rows) введите 1 в поле Количество строк (Number of Rows) и щелкните на кнопке OK для закрытия окна диалога.

На вкладке Преобразование (Transform) выберите команду Использовать первую строку в качестве заголовков (Use First Rows As Headers).

На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). После открытия окна диалога Настраиваемый столбец в поле Имя нового столбца (New Column Name) введите Parent Category, а в окно диалога Пользовательская формула столбца (Custom Column Formula) введите следующую формулу:

= Title

Затем щелкните мышью на кнопке OK для закрытия окна диалога Настраи-ваемый столбец.

12. На панели Примененные шаги (Applied Steps) выделите запрос Products Sample (Пример товаров) и удалите шаг Измененный тип (Changed Type).

(Шаг "Измененный тип" может в дальнейшем привести к ошибкам в столбце Size (Размер), типы столбцов лучше явно изменять попозже.)

13. На панели Запросы (Queries) выберите запрос C05E03 - Products.

14. На панели Примененные шаги (Applied Steps) удалите последний шаг, Изме-ненный тип.

15. Выберите элемент управления фильтра для заголовка столбца Parent Category (Родительская категория), что позволит гарантировать, что заголовки из четырех рабочих книг правильно извлечены на этапе предварительного добавления для запроса Products Sample. Добавленные результаты должны включать все четыре родительские категории.

Можно загрузить рабочую книгу решения C05E03 - Solution.xlsx и отчет Power BI C05E03 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

В следующем разделе показано, как сохранять контекст заголовков после добавления. Это позволит применить преобразования к приложенным результатам в запросе C05E03 - Products вместо применения изменений в запросе Products Sample. Такая методика особенно полезна, если объединить несколько листов, как показано в упражнении 5.4.

Сохранение заголовков

при постдобавлении контекста

В предыдущем разделе мы воспользовались методом предварительного добавления для динамического извлечения значения заголовка и присоединения его в качестве настраиваемого столбца перед этапом добавления. Этот подход эффективен, если имеется ограниченное число таблиц для добавления или если он применяется для объединения нескольких таблиц из папки.

Однако в некоторых случаях значительно легче извлекать заголовки из добавленной таблицы. Этот метод называют постдобавлением, поскольку он применяется после шага добавления. Методика постдобавления особенно полезна, если добавляются несколько листов из книги или если источник данных включает несколько таблиц в одном текстовом файле, причем каждая таблица отделена от предыдущей таблицы строкой заголовка.

УПРАЖНЕНИЕ 5.4. Сохранение заголовков из рабочих листов в одной книге

В упражнении демонстрируется использование метода постдобавления для извлечения родительских категорий AdventureWorks из заголовка на каждом листе одной книги. В реальных сценариях можно обрабатывать десятки рабочих листов в одной рабочей книге, причем все они имеют одинаковые контекстные данные, которые сохраняются в виде заголовков в верхней части каждой таблицы, как показано на рис. 5.5. Обратите внимание, что в этом сценарии, подобно именам файлов в упражнении 5.3, имена рабочих листов не содержат какого-либо значимого контекста из таблиц.

Рис. 5.5. Товары компании AdventureWorks сохраняются на отдельных листах одной книги

с родительскими категориями в качестве заголовков таблиц (но не имен листов)

В этом упражнении будет показано, как извлечь заголовки после сведения вместе всех листов.

1. Загрузите файл C05E04 - Products.xlsx с сайта https://aka.ms/DataPwrBIPivot/ downloads и сохраните его в папке C:\Data\C05\.

2. Откройте рабочую книгу и просмотрите ее содержимое. Обратите внимание, что товары компании AdventureWorks разделены с помощью родительской категории на четыре рабочих листа: Sheet1, Sheet2, Sheet3 и Sheet4, на которых применяются заголовки таблиц Bikes, Accessories, Components и Clothing, находящиеся в ячейках над этими таблицами (см. рис. 5.5).

3. Откройте новую пустую рабочую книгу в Excel или новый отчет в Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните последовательность команд По-лучить данные (Get Data) Из книги (From File) Из книги (From Workbook).

В Power BI Desktop: выполните команду Получить данные (Get Data), а затем выберите команды Файл (File) Excel Подключиться (Connect).

4. Выберите файл C05E04 - Products.xlsx и щелкните на кнопке Импорт (Import).

5. В окне диалога Навигатор (Navigator) щелкните правой кнопкой мыши на вкладке C05E04 - Products.xlsx и в контекстном меню выберите команду Изменить (Edit).

6. На панели предварительного просмотра обратите внимание на то, что невоз-можно определить родительскую категорию товара. После объединения не-скольких таблиц из нескольких рабочих листов (см. главу 3) для имен каждого рабочего листа большое значение приобретает контекст. Удобно сохранить этот контекст при сохранении столбца Name. К сожалению, в данном упражнении имена рабочих листов не имеют контекста.

Можно игнорировать все столбцы, кроме содержащих данные. Щелкните правой кнопкой мыши на заголовке столбца Data (Данные) и в контекстом меню выберите команду Удалить другие столбцы (Remove Other Columns).

7. Разверните столбец Data, щелкнув на значке развертывания в правой части заголовка или выбирая столбец Data (Данные), а затем выберите команду Развернуть (Expand) на вкладке Преобразование (Transform). На панели Развернуть щелкните на кнопке OK.

В первой таблице на рис. 5.6 показаны добавленные результаты. В ней появи-лись названия Bikes, Accessories, Components и Clothing. Необходимо выполнить последовательность преобразований, которая представлена на рис. 5.6, что позволит извлечь заголовки и передать их в столбец Parent Category.

8. На вкладке Главная (Home) выберите команду Удалить строки (Remove Rows), затем задайте команду Удалить пустые строки (Remove Blank Rows). Смысл этого шага ясен; такой подход применен в упражнении 5.3. Поскольку за заголовками обычно следуют пустые строки, на этом этапе они удаляются.

9. На вкладке Добавление столбца (Add Column) щелкните мышью на кнопке Условный столбец (Add Conditional Column).

Ваша задача — создать условный столбец, который поможет идентифицировать строки с заголовками.

Существуют следующие два подхода к определению названий:

а. При интуитивном подходе вы можете определить, соответствует ли первый столбец названию Bikes, Accessories, Components или Clothing, и установить

новый столбец как 1, если это так, или как 0 — в противном случае. Подобный подход не рекомендован, поскольку для определения названий применяет жестко закодированные значения. В будущем могут появиться новые родительские категории в качестве заголовков, но на этом этапе нельзя предсказать их значения. Таким образом, каждый раз при получении нового листа с другой родительской категорией нужно будет изменять запрос, чтобы затем его подключить.

б. При надежном подходе заголовки идентифицируются в данных примера как строки со значениями null во всех столбцах, кроме Column1. В результате можно сформировать запрос однократно, и нет необходимости изменять его всякий раз, когда к рабочим листам добавляются новые родительские категории.

10. Используя надежный подход, выполните следующие действия в окне диалога Добавление условного столбца (Add Conditional Column) для создания условного столбца, в котором и будут заголовки (рис. 5.7 подтверждает правильность выполнения этих шагов):

а. Введите IsTitle в поле Имя нового столбца (New Column Name).

б. В качестве первого имени столбца выберите Data.Column2.

Рис. 5.7. Выполните настройки в окне диалога Добавление условного столбца для динамической идентификации заголовков

в. Выберите оператор "не равно" (Does Not Equal).

г. Введите null в текстовое поле Значение (Value).

д. Введите 0 в поле Вывод (Output).

е. Щелкните на кнопке Добавить правило (Add Rule). Повторите шаги а-е, но с именем Data.Column3, выбранным вместо Data.Column2 на шаге б.

ж. Щелкните на кнопке Добавить правило. Повторите шаги a-e, но с именем Data.Column4, выбранным вместо Data.Column2 на шаге б.

з. Щелкните на кнопке Добавить правило. Повторите шаги а-е, но с именем Data.Column5, выбранным вместо имени Data.Column2 на шаге б.

и. Щелкните на кнопке Добавить правило. Повторите шаги а-е, но с именем Data.Column6, выбранном вместо Data.Column2 на шаге б.

к. Введите единицу в поле В противном случае (Otherwise) и щелкните на кнопке OK для закрытия окна диалога Добавление условного столбца.

Теперь появился новый условный столбец с фактическим заголовком, если значение IsTitle равно единице, в противном случае — null. Это создает основу для шага заполнения, как показано на рис. 5.6, шаг 3.

11. На вкладке Добавление столбца (Add Column) выберите параметр Условный столбец (Conditional Column). В открывшемся окне диалога Добавление ус-ловного столбца (Add Conditional Column) выполните следующие действия:

а. Введите Title в поле Имя нового столбца.

б. Введите IsTitle в первое поле Имя столбца.

в. В качестве оператора выберите "равно" (Equal).

г. Введите 1 в поле Значение.

д. Выберите раскрывающееся меню ниже поля Вывод и остановитесь на опции Выберите столбец (Select a Column).

е. Выберите столбец Data.Column1 из раскрывающегося меню Вывод.

ж. Введите null в поле В противном случае и щелкните на кнопке OK для за-крытия диалогового окна Добавление условного столбца.

Теперь имеются заголовки в столбце Title (Заголовок) для строк с заголовками и пустыми значениями в строках продукта. Здесь пригодится трансформация заполнения.

12. Правой кнопкой мыши щелкните на заголовке столбца Title, выберите пункт Заполнить (Fill), а затем выберите команду Вниз (Down). Этот шаг позволяет сохранять родительскую категорию в правильном контексте и в каждой строке.

Теперь можно отфильтровать строки заголовков, и столбец IsTitle помогает это сделать.

13. Щелкните на элементе управления фильтра в заголовке столбца IsTitle. После открытия панели фильтра отмените выбор значения 1 и затем щелкните на кнопке OK.

14. Удалите столбец IsTitle. (После того, как его использовали для удаления строк заголовка, он больше не нужен.)

15. Выполните следующие шаги для обработки встроенных заголовков в строках таблицы, которые вам уже знакомы по их применению при объединении нескольких листов в главе 3:

а. На вкладке Преобразование (Transform) выберите команду Использовать первую строку в качестве заголовков (Use First Row As Headers).

б. Переименуйте столбец Bikes в Parent Category.

в. Отфильтруйте имена столбцов встроенного товара, отфильтровав строки, содержащие значение Name в столбце Name.

Г") СОВЕТ

Можно повысить надежность переименования на шаге 15б, заменив код "Bikes" в строке формул на List.Last(Table.ColumnNames(#''K3MeHeHHbiM тип")). Вот обновленная формула:

= ТаЬ1е.КепатеСо1ишп5(#"Измененный тип ", {{List.Last(Table.ColumnNames(#"teMeHeHHbM тип ")), "Parent Category "}})

В результате такого усовершенствования Power Query вместо явного переименования столбца Bikes переименует последний столбец. Это важно, потому что в реальной жизни данные не остаются неизменными. Исходный запрос не будет обновлен, если таблица Велосипеды больше не является первой книгой в исходной папке. Указание Power Query по переименованию последнего столбца гарантирует, что запрос не будет обновлен. В главе 9 содержится больше информации о применяемом в данном случае синтаксисе, а в главе 10 показано, как можно аналогичным образом оптимизировать запросы, в том числе и для переименования столбцов.

16. Закройте окно редактора Power Query и загрузите объединенную таблицу.

Можно загрузить рабочую книгу решения C05E04 - Solution.xlsx и отчет Power BI C05E04 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

Использование подсказок контекста

При выполнении упражнения 5.4 можно извлекать и сохранять заголовки из рабочих таблиц AdventureWorks, а после этапа добавления — использовать их в качестве родительской категории. Поскольку добавленные таблицы включали заголовки в виде встроенных строк, для определения заголовков создавался условный столбец. Для динамической идентификации строк с заголовками предположим, что заголовки существуют в строках, где первый столбец не пустой, а все остальные столбцы — пустые. Но как быть, если такое предположение сделать нельзя? Имеется ли способ идентификации заголовков по их близости к конкретным ячейкам?

В данном разделе показано, как определить контекст по его расположению и близости к конкретным ячейкам привязки. Эти ячейки привязки служат подсказками, помогающими находить нужный контекст. Вы также узнаете, как игнорировать неподходящий контекст.

Рассмотрим новый вариант рабочей книги товаров AdventureWorks. В отличие от примера рабочей книги из упражнения 5.4, на этот раз родительская категория находится внутри небольшой таблицы, в верхней части таблицы Products, как показано на рис. 5.8.

Рис. 5.8. Ваша задача — извлечь родительскую категорию из верхней таблицы и сохранить ее в качестве нового столбца в добавленных снизу таблицах

Зададим значение родительской категории, используя расположение текста Parent Category.

УПРАЖНЕНИЕ 5.5. Применение столбца индекса

в качестве подсказки

В этом упражнении применим различные приемы для сохранения родительской категории. Импортируем каждый рабочий лист с подсказкой, которая помогает объединять таблицы и сохранять родительскую категорию.

1. Загрузите файл рабочей книги C05E05 - Products.xlsx с сайта https://aka.ms/ DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C05.

Откройте книгу и просмотрите ее содержание. Вы увидите, что товары ком-пании AdventureWorks разделены на четыре рабочих листа. Как показано на рис. 5.8, каждая родительская категория — Bikes, Accessories, Components и Clothing — хранится в отдельной небольшой таблице над таблицей товаров каждого рабочего листа. Рабочие листы называются Sheet1, Sheet2, Sheet3 и Sheet4 и не могут помочь в различении родительских категорий.

2. Откройте пустую новую рабочую книгу в Excel или новый отчет в Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: выполните команду Получить данные (Get Data), затем команды Файл (File) Excel Подключиться (Connect).

3. Выберите файл C05E05 - Products.xlsx и щелкните на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) правой кнопки мыши щелкните на вкладке C05E05 - Products.xlsx и в контекстном меню выберите команду Изменить (Edit).

5. Правой кнопкой мыши щелкните на заголовке в столбце Data и в контекстном меню выберите команду Удалить другие столбцы (Remove Other Columns). Как и в упражнении 5.4, теперь следует найти способ для поиска строки со значениями родительской категории. Развертывание этой таблицы в столбце Data (Данные) на данном этапе будет преждевременным. Сначала необходимо извлечь из каждой таблицы некоторые подсказки: номера их строк.

В упражнении 5.4 предполагалось, что родительская категория находится в виде заголовка в строках, не имеющих других значений. Следуя этому предположению, был добавлен условный столбец для обнаружения строк с непустыми значениями в первом столбце и пустыми — в остальных. В этом упражнении нет необходимости делать такое предположение. Как показано на рис. 5.8, значение родительской категории (ячейка A3) расположено рядом со значением Report Date (Дата отчета), которое находится в ячейке B3.

Как же определить родительскую категорию? Можно предположить, что строки со значениями родительской категории всегда имеют непустые значения в первых двух столбцах и пусты во всех остальных столбцах, но это допущение слишком рискованно. Что произойдет, если одна из таблиц будет содержать новую информацию именно в ячейках C3 и C4?

Для решения описанной проблемы можно сделать новое предположение: после удаления пустых строк значение родительской категории в каждом рабочем листе всегда находится в ячейке на пересечении второй строки и первого столбца. (В упражнении 5.6 будет продемонстрирован иной метод извлечения родительской категории.)

Для определения значения родительской категории сначала необходимо отметить вторую строку каждого листа (после удаления пустых строк). Для этого воспользуемся столбцом индекса и применим его к каждой таблице в столбце Data.

6. На вкладке Добавление столбца (Add Column) выберите команду Столбец индекса (Index Column) и удостоверьтесь, что оставлена настройка, заданная по умолчанию, и создается нулевой индекс. Затем скопируйте следующую формулу из строки формул:

= ТаЬ1е.Ас1с11пс1ехСо1ишп(#"Дру1ие удаленные столбцы", "Индекс", 0, 1)

Теперь, когда формула скопирована, можно удалить этот шаг из панели Примененные шаги (Applied Steps). Вам просто нужна формула, чтобы подготовить почву для следующих шагов.

7. На вкладке Добавление столбца (Add Column) выберите команду Настраиваемый столбец (Custom Column). Откроется окно диалога Настраиваемый столбец.

8. Как показано на рис. 5.9, вставьте формулу из шага 6 в окно Пользовательская формула столбца (Custom Column Formula) и модифицируйте ее следующим образом:

а. Удалите знак равенства. По умолчанию знак равенства добавляется в окно Пользовательская формула столбца.

б. Замените строку # "Другие удаленные столбцы" на [Data]. Это можно выполнить, выбирая текст # "Другие удаленные столбцы" и затем выбирая столбец Data в окне Доступные столбцы (Available Columns).

Получим результирующую формулу:

= Table.AddIndexColumn([Data], "Индекс", 0, 1)

Г") СОВЕТ

После щелчка на одном из столбцов в окне Доступные столбцы (Available Columns) окна диалога Настраиваемый столбец (Custom Column) генерируется код на языке M, что позволит получить доступ к значению этого столбца в каждой строке и применить его в результатах нового пользовательского столбца. В этом случае после щелчке на поле Data в окне формулы генерируется код [Data].

9. Оставаясь в окне диалога Настраиваемый столбец (Custom Column), в поле Имя нового столбца (New Column Name) введите название Таблица с индексом

и щелкните на кнопке OK. Теперь есть два столбца в панели предварительного просмотра: столбец Data с исходными объектами таблицы для каждого импортированного рабочего листа и столбец Таблица с индексом с теми же объектами таблицы и недавно добавленным столбцом индекса.

Имя нового столбца

Таблица с индексом

10. Удалите столбец Data. Теперь можно работать со столбцом Таблица с индексом.

11. Разверните столбец Таблица с индексом, щелкая мышью на значке развертывания в правой части заголовка. Отмените установку флажка Использовать исходное имя столбца как префикс (Use Original Column Name As Prefix) и щелкните на кнопке OK.

Теперь можно увидеть, что родительская категория всегда доступна в строках, где столбец индекса имеет значение, равное единице. Контекстные заголовки Parent Category (Родительская категория) и Report Date (Дата отчета) всегда находятся в строках, где столбец индекса имеет нулевое значение. Теперь шаги подготовки данных выполнять легче.

12. Для того чтобы отметить строки со значениями Parent Category, на вкладке Добавление столбца выберите параметр Условный столбец (Add Conditional Column). В открывшемся окне диалога Добавление условного столбца сделайте следующие изменения:

а. В поле Имя нового столбца введите Parent Category.

б. В качестве имени первого столбца укажите Индекс.

в. В качестве оператора выберите "равно" (Equal).

г. Введите 1 в поле Значение (Value).

д. Выберите раскрывающееся меню, находящееся ниже меню Вывод (Output), и выберите команду Выберите столбец (Select a Column).

е. Выберите Column1 из раскрывающегося меню Вывод (Output).

ж. Введите null в поле В противном случае (Otherwise) и щелкните мышью на кнопке OK для закрытия окна диалога Добавление условного столбца (Add Conditional Column).

13. Выполните команды Заполнить (Fill) и Вниз (Down) для столбца Родитель-ская категория (Parent Category).

14. Отфильтруйте строки со значениями 0, 1 или 2 в столбце Индекс.

Теперь, после сохранения контекста родительской категории и игнорирования неподходящего контекста в строках с индексом 1, столбец Индекс можно удалить.

15. Выполните следующие три уже знакомых шага:

а. На вкладке Преобразование (Transform) выберите команду Использовать первую строку в качестве заголовков (Use First Row As Headers).

б. Переименуйте столбец Bikes в Parent Category.

в. Отфильтруйте имена столбцов встроенного товара, отфильтровав строки, содержащие значение Name в столбце Name.

Г") СОВЕТ

Как показано в упражнении 5.4 (шаг 15б), можно повысить надежность переименования на шаге 15б этого упражнения, заменив код "Bikes" в строке формул на List.Last(Table.ColumnNames(# "Измененный тип")).

Теперь можно загрузить сформированный запрос в отчет. Контекст родительской категории сохранен в добавленных результатах благодаря подсказке, которая применялась на шаге перед добавлением, — в этом случае подсказкой служит столбец Индекс для каждой таблицы.

Можно загрузить книгу решения C05E05 - Solution.xlsx и отчет Power BI C05E05 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

УПРАЖНЕНИЕ 5.6. Идентификация контекста

с учетом близости ячейки

В упражнении 5.6 предполагалось, что значение родительской категории всегда находится во второй строке и первом столбце каждого листа (после удаления пустых строк). Теперь изменим предположение и допустим, что расположение строки значения родительской категории может измениться и больше не использоваться для определения контекста, который необходимо извлечь.

Для данного упражнения предположим, что значение родительской категории (как показано на рис. 5.10 в ячейке A6) всегда можно найти ниже ячейки со значением Parent Category (Родительская категория) в первом столбце (как показано на рис. 5.10 в ячейке A5). Можно использовать местоположение ячейки с жестко закодированным текстом Parent Category в качестве "якоря" или метки для расположения значения родительской категории. Привязка контекста к фиксированным элементам помогает сделать надежные предположения и позволяет формировать надежные запросы, которые лучше функционируют с течением времени при изменениях данных.

Чтобы задача стала более интересной, в данном упражнении произвольные значения расположены выше родительской категории. Поэтому нельзя предполагать, что родительская категория находится в определенной строке. Более того, не следует предполагать, что таблица товаров всегда занимает определенные строки. На рис. 5.10 иллюстрируется эта сложная задача.

Рис. 5.10. Ваша задача состоит в извлечении родительской категории из верхней таблицы и сохранении ее в виде нового столбца в добавленных нижних таблицах, игнорируя при этом все ненужные данные

В данном упражнении рассматривается новая методика динамического поиска индексов строк контекста и заголовков таблиц товаров, что позволяет игнорировать нерелевантный контекст в каждой рабочей таблице, извлекать именно родительскую категорию и сводить все таблицы товаров вместе. Методика основана на применении функции M List.PositionOf. Эта функция получает список в качестве входных аргументов и значение для поиска своей позиции, а возвращает индекс (начиная с нуля) позиции значения в списке.

1. Загрузите файл рабочей книги C05E06 - Products.xlsx с сайта https://aka.ms/ DataPwrBIPivot/downloads. Сохраните эту книгу в папке C:\Data\C05.

2. Либо повторите шаги 2-9 из упражнения 5.5, либо создайте книгу C05E06 - Jump Start.xlsx или отчет C05E06 - Jump Start.pbix (оба файла находятся на сайте https://aka.ms/DataPwrBIPivot/downloads).

По завершении на панели предварительного просмотра должно появиться два столбца: Data (Данные) с исходными объектами таблицы для каждого импортированного листа и Таблица с индексом с теми же объектами таблицы и столбцом индексов.

3. В редакторе Power Query, на вкладке Добавление столбца (Add Column), щелкните мышью на кнопке Настраиваемый столбец (Custom Column). После открытия окна диалога Добавление настраиваемого столбца (Add Column) выполните следующие действия:

а. В поле Имя нового столбца (New Column Name) введите Индекс строки родительской категории (Row Index of Name).

б. В окно Пользовательская формула столбца (Custom Column Formula) ско-пируйте следующую формулу и щелкните на кнопке OK для закрытия окна диалога Настраиваемый столбец (Custom Column):

List.PositionOf([Data][Column1], "Parent Category ")

Эта формула применит функцию List.PositionOf для столбца Column1, который имеет вид списка и предоставляется в качестве первого аргумента [Data] [Column1]. Второй аргумент, "Parent Category", — это значение, которое желательно найти в списке для возврата его позиции в списке, представляющей индекс строки, где значение "Parent Category" жестко задано.

Вы обнаружите, что возвращаемый столбец Индекс строки родительской ка-тегории (Row Index of Parent Category) содержит различные числа.

Каждое число представляет местоположение ячейки Parent Category (Родительская категория) в индексе, начинающемся с нуля.

Теперь создадим новый настраиваемый столбец. На этот раз детализируем каждый объект таблицы в столбце Data, применяя индекс, найденный на предыдущем шаге, для получения фактических значений родительской категории. Напомним, что значение родительской категории находится ниже названия столбца Parent Category. Можно использовать индекс в столбце Индекс строки родительской категории и увеличивать его на единицу для получения значения родительской категории.

4. На вкладке Добавление столбца (Add Column) выберите команду Настраивае-мый столбец (Custom Column). После открытия окна диалога Настраиваемый столбец (Custom Column) внесите следующие изменения:

а. В поле Имя нового столбца (New Column Name) введите Parent Category.

б. В окно Пользовательская формула столбца (Custom Column Formula) ско-пируйте следующую формулу и щелкните на кнопке OK для закрытия окна диалога Настраиваемый столбец (Custom Column):

[Data]{[HngeKc строки родительской категории] + 1}[Column1]

Ранее в этой главе, в разделе "Применение детализации к ячейке таблицы (объяснение с помощью M-формул) ", мы имели дело с более простой версией этой формулы:

Source{0}[Column1]

Здесь синтаксис аналогичный. Ссылаться на одну ячейку в таблице можно, написав следующее выражение:

ТаЬ1е{индекс строки}[имя столбца]

Напомним, что фигурные скобки служат для доступа к строке в таблице с ис-пользованием нулевого индекса. Квадратные скобки обеспечивают доступ к имени столбца. В формуле на шаге 4б сначала получают доступ к объекту таблицы в каждой строке столбца Data, вызывая [Data]. Затем получают доступ к индексу строки по значению, которое доступно в столбце Индекс строки родительской категории, после увеличения индекса на единицу. Наконец, в указанной строке получаем доступ к ячейке в первом столбце.

Родительская категория сохранена в столбце Parent Category. Определим ин-декс строки заголовков главной таблицы — товаров AdventureWorks. Следующий шаг аналогичен шагу 3, но на этот раз выполняется поиск столбца Name вместо Parent Category.

5. На вкладке Добавление столбца (Add Column) выберите команду Настраивае-мый столбец (Custom Column). После открытия окна диалога Настраиваемый столбец (Custom Column) выполните следующие действия:

а. В поле Имя нового столбца (New Column Name) введите Индекс строки названия (Row Index of Name).

б. В окно Пользовательская формула столбца (Custom Column Formula) ско-пируйте следующую формулу и щелкните на кнопке OK для закрытия окна диалога Настраиваемый столбец (Custom Column):

List.PositionOf([Data][Column1], "Name")

6. Теперь можно удалить столбцы Индекс строки родительской категории и Data. На следующем шаге выполняется обработки столбца Таблица с индек-сом, созданном при выполнении упражнения 5.5.

7. Разверните столбец Таблица с индексом с помощью щелчка мышью на значке развертывания, который находится справа от заголовка. На панели развертывания отмените установку флажка Использовать исходное имя столбца как префикс (Use Original Column Name As Prefix) и щелкните на кнопке OK.

На этом этапе в добавленных результатах сохранена родительская категория. Следующая задача — использовать две подсказки, которые имеются в столбцах Индекс и Строка индекса названия. Все строки, индекс которых находится ниже строки индекса названия, должны быть отфильтрованы, поскольку они не относятся к таблице товаров. Это можно сделать в два этапа: сначала создать условный столбец, а затем отфильтровать таблицу по результатам нового столбца.

8. На вкладке Добавление столбца (Add Column) выберите команду Условный столбец (Conditional Column). После открытия окна диалога Добавление условного столбца (Add Conditional Column) установите настройки, как показано на рис. 5.11, выполнив следующие действия:

а. В поле Имя столбца (Column Name) введите Индекс.

б. Выберите оператор "больше или равно" (Greater Than or Equal To).

в. В раскрывающемся меню под полем Значение (Value) выберите параметр Выберите столбец (Select a Column).

г. В раскрывающемся списке Значение (Value) выберите значение Индекс строки имени (Row Index of Name).

д. Введите 1 в поле Вывод (Output).

е. Введите 0 в поле В противном случае (Otherwise) и щелкните на кнопке OK для закрытия окна диалога Добавление условного столбца (Add Conditional Column).

Добавление условного столбца

Добавьте условный столбец, который вычисляется из других столбцов или значений.

Имя нового столбца

Отмена

Рис. 5.11. Чтобы отфильтровать строки перед таблицей товаров, выполните шаг 8 и убедитесь, что в диалоговом окне Добавление условного столбца выбраны настройки, показанные на этом рисунке

9. Отфильтруйте данные в столбце Пользовательская с помощью значения 1 и удалите столбцы Индекс, Индекс строки названия и Пользовательская.

10. Выполните следующие три известных шага для удаления заголовков добавленных товаров:

а. На вкладке Преобразование (Transform) выберите команду Использовать первую строку в качестве заголовков (Use First Row As Headers).

б. Переименуйте столбец Bikes на Parent Category (Родительская категория).

в. Отфильтруйте имена столбцов встроенного товара, отфильтровав строки, содержащие значение Name в столбце Name.

СОВЕТ

Как и в упражнениях 5.4 и 5.5, можно улучшить надежность переименования на шаге 10б, заменив код "Bikes" в строке формул с помощью функции List.Last(Table.

ColumnNames(# "Измененный тип ")).

Книгу решения C05E06 - Solution.xlsx и отчет Power BI C05E06 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Резюме

В главе было показано, каким образом можно сохранить контекст, который обычно находится вне таблиц. Рассказано о различных сценариях сохранения контекста, причем особое внимание уделено возможности комбинирования контекстов и таблиц из нескольких источников.

Рассмотрение было начато с использования базовой техники добавления запросов с именами файлов или таблиц, которые содержат ключевую контекстную информацию о данных. Затем к каждому запросу применялся настраиваемый столбец для сохранения контекста.

Показано, каким образом можно применить метод предварительного добавления для динамического обнаружения и извлечения заголовков на листах и объединения их в дополненные результаты при объединении файлов из папки. С помощью шага детализации вы сумели сохранить значения родительской категории из таблиц продуктов AdventureWorks, а познакомившись с синтаксисом M, — получить доступ к указанным ячейкам таблицы.

Для того чтобы справиться с контекстом в нескольких рабочих листах одной и той же папки, предложена методика постдобавления для определения контекста и его сохранения после объединения нескольких таблиц. Для сохранения категории родительских продуктов AdventureWorks из заголовков нескольких рабочих листов применялись условные столбцы и преобразование заполнения вниз. (Этот подход особенно полезен, если предварительное добавление невозможно.)

Продемонстрировано, как можно использовать контекстные подсказки в источниках данных для обнаружения как существенного, так и необязательного контекста и добавления результатов при сохранении контекста. В одном из упражнений применялся язык M для формирования столбца индекса на этапе предварительного добавления, а также для определения значений родительской категории AdventureWorks и сохранения данных. Также вы узнали, как можно применять условные столбцы к столбцу индекса для копирования контекста в новый столбец и как сохранять контекст с помощью заполнения вниз.

В конце главы описаны предварительно добавленные подсказки для идентификации контекста по его близости к ячейкам привязки. Привязка ценных данных к фиксированным элементам в ваших данных повышает надежность запросов.

Рассматривалось, как посредством функции языка M List.PositionsOf выполнить по-иск подходящих якорей и осуществить детализацию на основе найденного индекса позиции.

В главе 6 будут рассмотрены несколько распространенных сценариев, когда контекст присутствует в таблицах, но искажен и плохо структурирован, что замедляет работу и не позволяет раскрыть реальный бизнес-потенциал данных.

Контекст — ключ к анализу данных, но если вы плохо его понимаете, то многие идеи останутся нереализованными.

Глава 6

Отмена свертывания столбцов таблицы

Дайте мне достаточно длинный рычаг и точку опоры, чтобы закрепить его, и я переверну Землю.

- Архимед

В ЭТОЙ ГЛАВЕ ВЫ...

■ Узнаете, как идентифицировать плохо структурированный контекст в таблицах.

■ Изучите различные виды преобразований отмены свертывания столбцов, включая обычную отмену свертывания столбцов, отмену свертывания других (кроме текущего) столбцов и отмену свертывания выделенных столбцов.

■ Научитесь применять базовые преобразования отмены свертывания столбцов для улучшения контекста таблицы.

■ Узнаете, почему важно удалять итоги при импорте исходной таблицы и как это реализовать эффективно.

■ Выясните, как отменить отмену свертывания столбцов таблицы с уровнями иерархии 2x2 в строках и столбцах.

■ Узнаете, как отфильтровать промежуточные итоги из сведенных таблиц.

Ранее в книге было показано, как решать самые разнообразные задачи по анализу плохо подготовленных данных. Были изучены основы очистки данных для отдельных столбцов и профессиональные приемы по объединению нескольких таблиц. В главе 5 шла речь о важности контекста и о том, как его сохранить. Теперь, когда вы уже знаете, как правильно форматировать данные, собирать их из нескольких источников и полностью сохранять их контекст, пришло время узнать, как решается более сложная задача.

Таблицы отображают контекст, который они представляют. Сохранение контек-ста — вполне понятная задача, но решить, как лучше всего представить контекст, — это совсем другая проблема. В данной главе показано, как улучшить структуру таблиц в отчетах, чтобы лучше представлять предмет данных и усовершенствовать способность анализировать эти данные.

Преобразование отмены свертывания столбцов — основной метод при работе с неудовлетворительно спроектированными таблицами, оно имеет ключевое значение для достижения успеха при обработке данных. С помощью преобразования отмены свертывания столбцов Power Query будет служить "архимедовым рычагом" и опорой для перемещения данных во всех видах и формах.

Идентификация

плохо спроектированных таблиц

Специалисты в области лингвистики различают синтаксис, семантику и прагматику. При этом синтаксис связан с отношениями между словами, семантика имеет дело со значением предложений, а прагматика сосредотачивается на том, как различный контекст может влиять на интерпретацию сообщения. Синтаксис, семантику и прагматику можно использовать в качестве метафор для подготовки данных и в процессе их преобразования.

В предыдущих было главах показано, как улучшить "синтаксис" данных путем очистки неважно отформатированных данных. Вы обогатили "семантику" своих данных, комбинируя несвязанные наборы данных и сохраняя их контекст. В данной главе сосредоточимся на более тонкой задаче и изучим "прагматику" подготовки данных. Принципы разработки таблиц существенно влияют на интерпретацию данных. Как и в прагматике, где различный социальный контекст или тон голоса могут оказать влияние на интерпретацию передаваемого сообщения, в анализе данных способ формирования таблиц может существенно повлиять на результат и интерпретацию данных.

Если исходные таблицы хорошо составлены, то их можно эффективно использовать в сводных таблицах и сводных диаграммах в Excel или в визуальных представлениях Power BI. Неудовлетворительное оформление таблицы усложняет реализацию необходимых аналитических расчетов.

В этой главе будет показано, каким образом можно реструктурировать плохо спроектированные таблицы, что улучшит способ представления данных и позволит получить больше информации с помощью сводных таблиц в Excel и визуальных элементов в Power BI. В этом контексте плохо спроектированная таблица — это таблица, неудовлетворительно отображающая реальные объекты, а именно данные, которые отслеживаются и представляются в отчетах. Не всегда легко определить, правильно ли спроектированы таблицы. Один из способов получить первоначальное представление о том, хорошо ли структурирована таблица, — посмотреть, достаточно ли верно каждая строка таблицы представляет реальный предмет.

Данные, структурированные в таблице или в наборе взаимосвязанных таблиц, всегда соответствуют реальным объектам. Это могут быть транзакции продаж компании, измерения в научных экспериментах, результаты тестов учащихся или статистика игр любимой баскетбольной группы. Все эти объекты и факты, которые собраны, представляются в виде набора текстовых и числовых значений в таблицах. Вы знаете, что таблица хорошо спроектирована, если каждая ее строка представляет уникальный объект.

Чтобы понять трудности, связанные с идентификацией плохо спроектированных таблиц, давайте рассмотрим четыре таблицы, которые были по-разному сформированы для описания одной и той же тематики: суммарных доходов вымышленной компании AdventureWorks. Как показано на рис. 6.1, каждая из четырех таблиц имеет различную схему для отображения общего дохода по счету клиента и родительской категории товара.

Таблица 1, показанная на рис. 6.1, структурирована как типичная сводная таблица Excel. Учетная запись задается как поле строки, родительская категория — это поле столбца, а поле Revenues (Доходы) находится в области значений сводной таблицы. И хотя таблица 1 может неплохо выглядеть в окончательном отчете, вы признаете такую структуру неэффективной, если используете ее в качестве источника данных для сводных таблиц в Excel или визуальных элементов в Power BI.

Таблица 2, показанная на рис. 6.1, также структурирована в виде сводной таблицы. Поля Account (Позиция) и Parent Category задаются как поля столбцов, а "Доходы" — как значения сводной таблицы. Если вы знакомы со сводными таблицами, то наверняка знаете, что и таблица 1, и таблица 2 могут быть представлением одной и той же исходной таблицы.

Таблица 3, показанная на рис. 6.1, иллюстрирует крайний случай, когда структура таблицы превращается в общий контейнер пар "атрибут/значение". Несмотря на то что эта таблица достаточно понятна для просмотра и определения контекста отдельных значений, она не может восприниматься сводными таблицами в Excel или визуальными элементами в Power BI.

Таблица 1. Сводная таблица с полями Account в области строк и Parent category в области столбцов

Account Accessories Bikes Clothing Components

Action Bicycle Specialists 1,299.89 76,613.65 2,461.66 9,494.08

Aerobic Exercise Company 1,732.89

Bulk Discount Store 730.46 70,597.28 851.56 1,980.92

Таблица 2. Сводная таблица с полями Account и Parent Category в области столбцов

Action Bicycle Specialists Aerobic Exercise Company Bulk Discount Store

Accessories Bikes Clothing Components Components Accessories Bikes Clothing Components

1,299.89 76,613.65 2,461.66 9,494.08 1,732.89 730.46 70,597.28 851.56 1,980.92

Рис. 6.1. Глядя на эти четыре варианта представления доходов AdventureWorks в зависимости

от позиции и родительской категории, можно ли догадаться, какие таблицы плохо спроектированы?

Таблица 4 обладает наилучшим дизайном среди таблиц, представленных на рис. 6.1. Эта таблица представляет доходы компании AdventureWorks по учетным записям и родительской категории в удобном виде, который хорошо подходит для формирования сводных таблиц в Excel и визуальных элементов в Power BI. Фактически при выборе таблицы 4 в качестве источника данных для сводной таблицы в Excel можно легко создать и таблицу 1, и таблицу 2, перетаскивая поля Account (Позиция), Category (Категория) и Revenues (Доходы) в разделы полей сводной таблицы Столбцы (Columns), Строки (Rows) и Значения (Values). Таблицы, структурированные как таблица 4, иногда называют таблицами фактов. Они представляют факты вовлеченных сущностей. В рассматриваемом случае сущностями служат общие доходы (revenues), суммированные по уровням позиций (accounts) и родительских категорий (parent category).

См. также

Таблицы фактов часто рассматриваются вместе с таблицами поиска/измерения и с при-менением звездообразной схемы. В главе 2 описывается методика формирования таблицы фактов и таблицы поиска путем применения ссылок и удаления дубликатов. Чтобы узнать больше о таблицах фактов в контексте моделирования в Power BI и Power Pivot, прочитайте статью "Анализ данных с помощью Power BI и Power Pivot для Excel", написанную Альберто Феррари и Марко Руссо, Microsoft Press (Analyzing Data with Power BI and Power Pivot for Excel by Alberto Ferrari and Marco Russo).

Таблицы 2 и 3 можно легко идентифицировать как неудачные представления доходов компании AdventureWorks, и, следовательно, они спроектированы неправильно. Сложнее понять, что таблицы, подобные таблице 1, тоже плохо спроектированы. В отличие от таблиц 2 и 3, которые совершенно не вписываются в сводную таблицу Excel или визуальное представление Power BI, таблицы, подобные таблице 1, могут соответствовать требованиям. В чем же недостаток дизайна таблицы 1? Эта таблица содержит четыре столбца одного типа — все они отображают доходы по различным родительским категориям товаров.

Если обнаружится, что исходные таблицы содержат несколько числовых столбцов с разными контекстами для одного и того же атрибута, можно подозревать, что имеется лучший способ представления данных. Например, если в отдельных столбцах таблицы указаны годы, причем каждый столбец имеет одинаковый тип числового значения, то такая структура явно плохая. В таблице 1 на рис. 6.1 доходы четырех родительских категорий товаров разнесены по отдельным столбцам. Для определения общего дохода по счету необходимо сложить четыре столбца в каждой строке, но эту задачу на практике не решить в сводных таблицах Excel или визуальных представлениях Power BI.

Таким образом, из всех вариантов, приведенных на рис. 6.1, таблица 4 является наилучшим примером представления доходов компании AdventureWorks. Но что делать, если исходные данные уже отформатированы, как показано в таблице 1, таблице 2 или таблице 3?

Можно ли преобразовать любую из этих таблиц в таблицу, подобную таблице 4, для улучшения возможностей анализа? В обыденной жизни вы найдете много примеров плохо спроектированных таблиц, которые напоминают первые три таблицы на рис. 6.1, и вполне возможно, что сможете их преобразовать. Далее показано, как преобразовать их в таблицы фактов, которые отлично подходят для составления сводных таблиц в Excel и визуальных представлений в Power BI.

В следующем разделе описывается преобразование отмены свертывания столбцов и объясняется, как его применять для сводных таблиц, таких как таблица 1 на рис. 6.1.

Знакомство с преобразованием

отмены свертывания столбцов

Преобразование отмены свертывания столбцов доступно на вкладке Преобразование (Transform) редактора Power Query или в контекстном меню, которое появится после щелчка правой кнопкой мыши на заголовке одного или нескольких выбранных столбцов. В контекстном меню и на вкладке Преобразование (Transform) имеются три команды Отметить свертывание столбцов (Unpivot):

■ Отменить свертывание столбцов (Unpivot Columns).

■ Отменить свертывание других столбцов (Unpivot Other Columns).

■ Отменить свертывание только выделенных столбцов (Unpivot Only Selected Columns).

Выполнение этих трех команд приводит к одинаковому результату для таблицы, но они могут действовать и по-разному, поскольку в будущем в вашу таблицу могут быть добавлены новые столбцы. О различиях между этими тремя командами пойдет речь в следующем разделе.

В случае применения команды отмены свертывания к выбранным столбцам Power Query реструктурирует вашу таблицу, создавая отдельную строку для каждой ячейки в столбцах, для которых отменено свертывание. Строка каждой такой ячейки включает столбцы, для которых не было отменено свертывание, а также два дополнительных столбца: Attribute (Атрибут) и Value (Значение). Столбец Attribute содержит исходное имя столбца ячейки, а столбец Value содержит в этой ячейке исходное значение. На рис. 6.2 показано, каким образом каждая ячейка в столбцах, где отменено свертывание, размещается после выполнения этого преобразования.

Преобразование отмены свертывания столбцов может показаться несколько сложным, особенно при первом рассмотрении. В основном при использовании такого преобразования в таблице рассматривается два типа столбцов:

■ Столбцы привязки — это столбцы, к которым не применяется преобразование отмены свертывания (эти столбцы не свертываются). Они представляют атрибуты, которые желательно сохранить в сущностях. На рис. 6.2 единственным столбцом привязки является столбец Account. Далее в этой главе приведены примеры с несколькими столбцами привязки.

■ Столбцы, для которых отменено свертывание, — это столбцы, для которых имеются разные проявления одного и того же атрибута. Например, столбцы

Accessories, Bikes, Components и Clothing относятся к категории доходов. И хотя для каждого из этих столбцов родительская категория товара компании AdventureWorks различна, все значения имеют один атрибут: Revenues (До-ходы).

Account

Action Bicycle Specialists Accessories

[ 1,293.83 Bikes

76,613.55 Clothing 2,461.66 Components

9,494.08

Aerobic Exercise Company 1,732.89

Bulk Discount Store 730.46 70,597.28 851.56 1,980.92

Преобразование отмены свертывания столбца обращается к любой ячейке в столбцах, для которых отменено свертывание, и представляет ее в строке, содержащей столбцы привязки и два новых столбца: Attribute и Value для соответствующего исходного имени и значения столбца.

УПРАЖНЕНИЕ 6.1. Применение преобразований отмены свертывания столбцов и отмены свертывания других столбцов

В данном упражнении будем практиковаться в использовании двух основных преобразований отмены свертывания столбцов: Отмена свертывания столбцов (Unpivot Columns) и Отмена свертывания других столбцов (Unpivot Other Columns). Для выполнения этого упражнения обратимся к примеру рабочей книги AdventureWorks в том же формате, что и таблица 1 на рис. 6.1.

1. Загрузите файл C06E01.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C06\.

2. Откройте пустую новую книгу в Excel или новый отчет в Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: в раскрывающемся меню Получить данные (Get Data) выберите команду Excel.

3. Выберите файл C06E01.xlsx и щелкните на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите вкладку Revenues (Доходы) и щелкните на кнопке Изменить (Edit).

5. В открытом окне редактора Power Query обратите внимание, что панель про-смотра включает доходы компании AdventureWorks по столбцам Account, Accessories, Bikes, Clothing и Components. Выберите последние четыре столбца, либо нажимая на клавишу <Ctrl> и щелкая на заголовках, либо применяя клавиши <Shift>+<npaeafl стрелка>, либо выбирая столбец Accessories (Аксес-суары) и затем нажимая комбинацию клавиш <Shift>+<End>.

6. На вкладке Преобразование (Transform) выберите команду Отменить сверты-вание столбцов (Unpivot Columns). В качестве альтернативы щелкните правой кнопкой мыши на одном из выбранных столбцов и в контекстном меню выберите параметр Отменить свертывание столбцов (Unpivot Columns). Теперь для таблицы отменено свертывание столбцов (см. таблицу 4 на рис. 6.2).

Заметим, что в строке формул появилась следующая функция M Table.

UnpivotOtherColumns:

= Table.UnpivotOtherColumns(#"teMeHeHHbM тип ", {"Account"}, "Атрибут", "Значение")

Теперь можно удалить шаг Несвернутые столбцы (Unpivoted Columns) на панели Примененные шаги (Applied Steps) и выбрать первый столбец, Account. После этого на вкладке Преобразование (Transform) выберите раскрывающийся список Отменить свертывание столбцов (Unpivot Columns), а затем выберите команду Отменить свертывание других столбцов (Unpivot Other Columns). Кроме того, можно щелкнуть правой кнопкой мыши на заголовке столбца Account и в контек-стном меню выбрать команду Отменить свертывание других столбцов. Как видите, результаты выполнения этой команды (на панели просмотра) будут такими же, что и на шаге 6. Просмотрите строку формул и убедитесь, что она включает ту же функцию Table.UnpivotOtherColumns. Единственное существенное отличие в том, что шаг преобразования на панели Примененные шаги (Applied Steps) теперь называется Другие столбцы с отмененным свертыванием (вместо шага Несвернутые столбцы).

Итак, какова реальная разница между преобразованиями Отменить свертывание столбцов и Отменить свертывание других столбцов, если базовая функция M одинакова? Нет никакой разницы. Все дело в простоте использования. Если число столбцов привязок больше, чем столбцов, для которых отменено свертывание, то можно выбрать столбцы, для которых отменено свертывание, и применить преобразование отмены свертывания столбцов. Если больше столбцов, для которых отменено свертывание, чем столбцов привязок, то можно выбрать столбцы привязки, а потом применить команду Отменить свертывание других столбцов.

На этом этапе можно задаться вопросом, почему формула M одна и та же для команд Отменить свертывание столбцов (Unpivot Columns) и Отменить свертывание других столбцов (Unpivot Other Columns). Два пользовательских интерфейса запускают одно и то же преобразование вследствие принятого предположения: чаще всего столбцы привязки остаются неизменными, а столбцы, для которых отменено свертывание, изменяются со временем. Например, исходная таблица в упражнении 6.1 включает четыре родительских категории товара. В будущем, если компания AdventureWorks выйдет на новые рынки или приобретет новую компанию, которая занята поставками новых товаров, исходная таблица будет включать новые столбцы доходов с другими родительскими категориями. Функция Table.UnpivotOtherColumns обеспечивает масштабирование запроса с течением времени для отмены свертывания новых столбцов без необходимости изменять запрос.

Распространенный сценарий, при котором функция Table.UnpivotOtherColumns приоб-ретает большое значение, — если исходная таблица содержит несколько столбцов с одним и тем же атрибутом, относящимся к разным периодам времени, например к годам, кварталам или месяцам. В подобных случаях преобразование отмены свертывания столбцов делает запрос достаточно устойчивым и отменяет свертывание столбцов для новых периодов, что и требуется, а в последующие периоды времени можно добавлять новые столбцы без дальнейших изменений.

СОВЕТ

Поскольку преобразование отмены свертывания столбцов и отмены свертывания других столбцов исполняются на уровне формул как команда Отменить свертывание других столбцов (Unpivot Other Columns), рекомендуется удостовериться, что столбцы привязки не меняются при применении этих преобразований.

Если применять команды Отменить свертывание столбцов (Unpivot Columns) и Отменить свертывание других столбцов (Unpivot Other Columns), то формула M явно ссылается на столбцы привязки по имени. Например, в предыдущей формуле столбец Account являлся столбцом привязки. Если столбец привязки переименован в источнике данных, то при следующем обновлении он будет по-врежден. И что еще хуже, если в источник данных добавляется новый столбец привязки, то для нового столбца будет отменено свертывание вместе с другими столбцами, что нарушит структуру вашей таблицы.

На рис. 6.3 показано, что преобразование отмены свертывания столбцов завершится с ошибкой при выполнении упражнения 6.1, если в исходную таблицу добавить новый столбец для адресов позиции. При использовании преобразования отмены свертывания столбцов Power Query полагает, что для столбца Address (Адрес) должно быть отменено свертывание вместе со столбцами Accessories, Bikes, Clothing и Components. Поскольку столбец Address не является родительской категорией товара, а значения адреса не отражают доходы, вы получите ошибки, если

А В C D E F G

1

2 Доходы по позиции и родительской категории

=5

4 Account ■ Address Accessories Bikes Clothing Components

5 Action Bicycle Specialists ■ Address 1 1,299.89 76,613.55 2,461.55 9,494.08

5 Aerobic Exercise Company ■ Address 2 1,732.89

7 Bulk Discount Store ■ Address 5 750.46 70,597.28 851.55 1,980.92

A В c

1 Account Q Attribute □ Value Г8^

2 Action Bicycle Specialists .Address Address1 I

3 Action Bicycle Specialists Accessories 1299.885'268

4 Action Bicycle Specialists Bikes 76613.6518

5 Action Bicycle Specialists Clothing 2461.65725

6 Action Bicycle Specialists Components 9494.082

7 Aerobic Exercise Company ^Address Address 2 ;

8 Aerobic Exercise Company Components 1732.89

9 Bulk Discount Store fMdress"" ■&ИгёЫ5":

10 Bulk DISCOLnt Store Accessories 730.464

11 Bu Ik Discount Store Bikes 70597.284

12 BuIkDIscoLnt Store clothing 851.562

13 Bulk Discount Store Components 1980.918

1 - Отменить свертывание других столбцов

2 - Добавлен столбец Address (Адрес)

3 - Результаты после обновления

Рис. 6.3. Добавление столбца привязки нарушает структуру таблицы, для которой отменено

свертывание столбцов при использовании опции Отменить свертывание других столбцов

или Отменить свертывание столбцов

попытаетесь изменить тип столбца Value (Значение), выбрав опцию Десятичное число (Decimal Number).

Еще более серьезная проблема возникает, если недавно добавленный столбец привязки включает числа. Представьте, что в исходной таблице доходов AdventureWorks в упражнении 6.1 добавлен новый столбец с указанием общего количества продаж для позиции. Теперь, во время следующего обновления, для нового столбца будет отменено свертывание независимо от родительских категорий товара, и все продажи будут учитываться как доходы. Последствия могут ввести в заблуждение и даже полностью исказить окончательный отчет.

Итак, что же делать, если ожидается, что исходная таблица будет иметь новые столбцы привязки? Ответ на этот вопрос можно найти в следующем разделе.

УПРАЖНЕНИЕ 6.2. Использование команды Отменить свертывание только выделенных столбцов

Как было показано в предыдущем разделе, при изменении столбцов привязки происходит сбой при выполнении команды Отменить свертывание других столбцов (Unpivot Other Columns). В таких случаях недостаточно применить команду Отменить свертывание столбцов, выбрав ее в пользовательском интерфейсе, поскольку этот шаг в строке формул преобразуется в формулу Отменить свертывание других столбцов (при этом используется функция Table.UnpivotOtherColumns). Вместо этого следует применять третью команду преобразования отмены свертывания столбцов в редакторе Power Query: Отменить свертывание только выделенных столбцов (Unpivot Only Selected Columns). В данном упражнении опробуем эту команду при работе с новой исходной таблицей. На этот раз воспользуемся примером данных вымышленной компании Wide World Importers и откроем общий доход за 2015, 2016, 2017 и 2018 годы.

1. Загрузите файл рабочей книги C06E02.xlsx с сайта https://aka.ms/DataPwrBIPivot/ downloads и сохраните его в папке C:\Data\C06\.

2. Откройте пустую новую рабочую книгу в Excel или новый отчет в Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: в раскрывающемся меню Получить данные (Get Data) выберите команду Excel.

3. Выберите файл C06E02.xlsx и щелкните на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите вкладку Revenues (Доходы) и затем щелкните на кнопке Изменить (Edit).

5. В открывшемся окне редактора Power Query заметим, что панель просмотра включает доходы поставщика компании Wide World Importer за годы 2015, 2016, 2017 и 2018. Выберем последние четыре столбца.

6. На вкладке Преобразование (Transform) щелкните мышью на раскрывающемся меню Отменить свертывание столбцов (Unpivot Columns) и затем выберите команду Отменить свертывание только выделенных столбцов (Unpivot Only Selected Columns). В качестве альтернативы правой кнопкой мыши щелкните на одном из выбранных столбцов и выберите в контекстном меню команду Отменить свертывание только выделенных столбцов. Теперь можно заметить, что в таблице, как и ожидалось, отменено свертывание столбцов.

В строке формул обратим внимание, что в формулу M входит функция Table.Unpivot:

= Table.Unpivot(#"teMeHeHHbM тип", {"2015", "2016", "2017", "2018"}, "Атрибут", "Значение")

Обратите внимание, что годы теперь закодированы в формуле М. В следующем году, когда при обновлении отчета в исходной таблице, содержащей новый столбец для доходов 2019 года, ваш отчет будет обрабатывать столбец 2019 как столбец привязки, что и показано на рис. 6.4.

Для того чтобы столбец, относящийся к 2019 году, не обрабатывался как столбец привязки, в этом сценарии лучше выбрать команду Отменить свертывание других столбцов (Unpivot Other Columns). Учитывая ограничения, имеющие место для команды Отменить свертывание только для выбранных столбцов, рекомендуется использовать ее только в том случае, если столбцы, для которых отменено

Рис. 6.4. Столбец, относящийся к новому году разбивает таблицу, для которой отменено свертывание столбцов, если команда Отменить свертывание только выделенных столбцов применялась для предыдущих годов

свертывание, не будут изменены в дальнейшем, а ожидаемые изменения могут произойти только в столбцах привязки.

Обработка итогов

При отмене свертывания столбцов для сводных таблиц следует обращать внимание на итоговые столбцы и строки. И хотя может возникнуть желание импортировать эти элементы в запрос, рекомендуется так не делать, а удалять такие элементы на этапе импорта. Наличие итогов данных повлияет на возможность создания простых и повторно используемых отчетов. Этот сценарий демонстрируется в упражнении 6.3.

См. также

Как и в случае с итоговыми значениями, не рекомендуется использование промежуточных итогов в данных, где отменено свертывание столбцов. Обработка промежуточных итогов рассматривается далее в этой главе.

УПРАЖНЕНИЕ 6.3. Применение преобразования отмены свертывания столбцов к общим итогам

В этом упражнении показано, какое влияние оказывают общие итоги при отмене свертывания исходной таблицы. Также описан наилучший способ удаления столбца и строки Общие итоги (Grand Totals) с помощью Power Query.

1. Загрузите файл рабочей книги C06E03.xlsx с сайта https://aka.ms/DataPwrBIPivot/ downloads и сохраните его в папке C:\Data\C06\.

2. Откройте пустую новую рабочую книгу в Excel или новый отчет в Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните последовательность команд Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: в раскрывающемся меню Получить данные (Get Data) выберите команду Excel.

3. Выберите файл C06E03.xlsx и щелкните на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите вкладку Revenues (Доходы), затем щелкните на кнопке Изменить (Edit).

В открывшемся окне редактора Power Query обратите внимание, что на панели предварительного просмотра отображаются доходы компании AdventureWorks по позициям и родительские категории: Accessories, Bikes, Clothing и Components. Последняя строка и последний столбец содержат итоговые значения Grand Total (Общий итог), как показано на рис. 6.5.

5. Выберите столбец Account (Позиция) и на вкладке Преобразование (Transform) щелкните на раскрывающемся списке Отменить свертывание столбцов (Unpivot Columns) и выберите пункт Отменить свертывание других столбцов (Unpivot Other Columns). В качестве альтернативы можно щелкнуть правой кнопкой мыши на заголовке столбца Account и выбрать в контекстном меню команду Отменить свертывание других столбцов.

После этого в таблице будет отменено свертывание столбцов. Обратите внимание, что для каждой сущности из столбца Account в столбце Атрибут имеется строка Grand Total (Общий итог), а при прокрутке таблицы последние пять строк включают Grand Total (Общий итог) в столбце Account.

Если импортировать в рабочую книгу или отчет Power BI строку и столбец Grand Total и суммировать весь столбец Value после отмены свертывания столбца, в итоговом столбце и строке происходит удваивание фактических доходов, поскольку каждый доход по конкретной позиции и родительской категории суммируется четырежды. Для того чтобы понять, почему это происходит, можно подсчитать, сколько раз один и тот же доход учитывается в оригинальной исходной таблице, до применения преобразования отмены свертывания столбцов. Доход от конкретной позиции (родительской категории) суммируется в четырех местах таблицы: один раз в конкретной внутренней ячейке, второй раз — в строке конкретной позиции в столбце Grand Total (Общий итог), третий раз — в строке Grand Total (Общий итог) столбца определенной родительской категории и, наконец, в правом нижнем углу ячейки Grand Total (Общий итог).

Включение итогового столбца или строки в результаты, для которых отменено свертывание столбцов, приведет к усложнению вычислений и визуальных эффектов. Отчет окажется уязвим для ошибок. Необходимо будет отфильтровать по значениям общего итога по столбцам Account и Parent Category в визуальных фильтрах или создать сложную формулу DAX, которая будет выполнять фильтрацию.

Рассмотрим подробнее формулу DAX, которую можно использовать в этом случае. (Можно также загрузить пример отчета Power BI C06E03 - Wrong Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.)

Для суммирования всех доходов можно применить следующую формулу DAX (при условии, что в таблице имеются столбцы Revenues, а сумма выручки находится в столбце Revenue):

Total Revenue = SUM ( Revenues [Revenue] )

К сожалению, этот результат увеличен в четыре раза, поэтому необходимо применить следующую формулу DAX:

Total Revenue = CALCULATE (

SUM (Revenue[Revenue] ),

FILTER (

Revenues,

Revenues [Account] <> "Grand Total" && Revenues [Parent Category] <>

"Grand Total"

)

)

И хотя эта мера поможет, если вы вынуждены работать с данными, содержащими общие доходы, лучше всего убедиться, что общие суммы отфильтрованы на уровне запроса.

Удаление общих итогов

И хотя можно продолжить предыдущее упражнение и отфильтровать значения Grand Total (Общий итог) сразу после шага отмены свертывания столбцов, этот подход неэффективен, поскольку снижает производительность и увеличивает время загрузки. Можно сократить время, необходимое для выполнения операций отмены свертывания столбцов, удаляя ненужные столбец и строку Grand Total (Общий итог) перед отменой свертывания столбцов. Для правильного удаления общего итога продолжите упражнение, выполняя следующие шаги.

6. Отмените шаг Отменить свертывание других столбцов на панели Примененные шаги (Applied Steps).

7. Выберите столбец Grand Total (Общий итог) и нажмите клавишу <Delete>.

В данном упражнении Grand Total (Общий итог) служит именем столбца, его можно безопасно удалить с помощью пользовательского интерфейса. В сле-дующем упражнении показан случай, когда столбец Grand Total (Общий итог) называется Column10. Удаление подобного столбца с обобщенным именем может привести к неожиданным результатам при дальнейшем расширении таблицы (и смещении вправо столбца Grand Total (Общий итог)). В упражнении 6.4 показано, как с помощью формулы M удалить столбец Grand Total, если он является последним столбцом.

8. На вкладке Главная (Home) выполните команду Удалить строки (Remove Rows) и затем выберите команду Удалить нижние строки (Remove Bottom Rows).

9. В окне диалога Удалить нижние строки (Remove Bottom Rows) введите едини-цу в поле Количество строк (Number of Rows) и щелкните мышью на кнопке OK для закрытия этого окна диалога.

Использование окна диалога Удалить нижние строки (Remove Bottom Rows) действительно является самым простым способом удаления строки общего итога. Однако при работе с необработанными данными, особенно если в качестве исходных данных фигурируют электронные таблицы, необходимо удалять дополнительные строки, находящиеся в нижней части таблицы. Для увеличения масштаба запроса, для удаления общих итогов вместе с дополнительными строками внизу таблицы, можно применить формулу M, динамически определяющую местоположение строки общего итога и удаляющую ее вместе с другими несоответствующими строками.

В описанных примерах для удаления нижней строки можно использовать следующие формулы:

= Table.RemoveLastN(#"Уgаленные столбцы",1)

= Table.FirstN(#"Удаленные столбцы ", List.PositionOf(#"Удаленные столбцы "[Account],

"Общий итог"))

Нажмите клавишу <Enter>, чтобы получить то же преобразование: удаление строки общего итога. Однако теперь запрос игнорирует любые конечные строки в данных.

Рассмотрим формулу с функцией Table.FirstN. Эта формула обрабатывает пер-вые N строк из таблицы. Она получает два аргумента: таблицу и количество сохраняемых строк. Параметр #"Удаленные столбцы" — это идентификатор таблицы. Второй аргумент включает число строк, которые необходимо сохранить. Ранее упоминалась функция List.PositionOf (глава 5). В данном случае доступ к столбцу Account осуществляется в формате списка и разыскивается ключевое слово "Grand Total". Результатом является нулевой индекс строки Grand Total, и это именно то число строк, которое желательно сохранить.

Теперь, когда строка Grand Total (Общий итог) удалена, можно перейти к шагу отмены свертывания столбцов.

10. Выберите столбец Account и на вкладке Преобразование (Transform) щелкните на раскрывающемся списке Отменить свертывание столбцов (Unpivot Columns) и выберите команду Отменить свертывание других столбцов (Unpivot Other Columns). Также можно щелкнуть правой кнопкой мыши на заголовке столбца Account и выбрать в контекстном меню команду Отменить свертывание других столбцов. Теперь можно переименовать столбец Атрибут в Parent Category (Родительская категория), а столбец Значение — в Revenue.

Как показано в этом упражнении, при использовании преобразования отмены свертывания столбцов важно удалять общие итоги. Удалять столбцы общих итогов несложно, для удаления общих итогов из нижней строки вместе с любыми несоответствующими конечными данными, которые имеются в рабочей таблице, можно применить интеллектуальную логику языка M.

Можно загрузить файлы решения C06E03 - Solution.xlsx и C06E03 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

Применение преобразования отмены свертывания столбцов на уровнях иерархии 2x2

В предыдущем разделе показано, как использовать преобразование отмены свертывания столбцов для простых таблиц. Исходные таблицы часто отформатированы как простые сводные таблицы, и преобразование отмены свертывания столбцов в Power Query станет одним из самых полезных инструментов. Это верно и для источников электронных таблиц, которые не столь хорошо подготовлены, как данные в хранилище данных вашей компании.

Поскольку большинство пользователей Excel незнакомы со сводными таблицами, они, как правило, вместо таблиц фактов работают с общими статическими отчетами. Если в ваших собственных отчетах необходимы уже обобщенные данные, то можно применить преобразование отмены свертывания столбцов для разбиения этих обобщенных таблиц и преобразовать данные в формат, пригодный для использования в сводных таблицах Excel и визуальных представлениях Power BI. Однако в реальной жизни приходится сталкиваться с более сложными таблицами, которые включают несколько уровней агрегирования в столбцах и строках.

Следующие два упражнения позволят сделать еще один шаг вперед и показать, как можно отменить свертывание столбцов в наиболее распространенных вложенных таблицах. Материал этого раздела позволит отменять свертывание столбцов для большого числа структурированных таблиц и преобразовывать их в таблицы фактов, которые можно вписывать в сводные таблицы Excel, а также в визуальные элементы сводных диаграмм и визуальных представлений Power BI.

УПРАЖНЕНИЕ 6.4. Применение преобразования отмены свертывания таблиц для уровней иерархии 2x2 с датами

Для того чтобы приступить к применению преобразования отмены свертывания столбцов для сложных таблиц, рассмотрим пример: техническая компания Wide World Importers планирует расширяться и разыскивает источники финансирования. Как главному аналитику инвестиционной компании, вам необходимо оценить эффективность работы компании Wide World Importers. Имеем отчет с обобщенными сведениями о доходах по годам, месяцам, категориям поставщиков и поставщикам, как показано в верхней части рис. 6.6.

Определим возможность по отмене свертывания столбцов дат и преобразования в таблицу фактов обобщенной сводной таблицы, как показано в нижней части рис. 6.6.

Прежде чем перейти к применению преобразования отмены свертывания столбцов к этой таблице, сосредоточимся на основных проблемах. Определить столбцы при-

Рис. 6.6. Power Query можно применять для отмены свертывания столбцов сложных таблиц, таких как сводная таблица доходов компании Wide World Importers. Нижняя таблица, к которой не применялось преобразование отмены свертывания столбцов, лучше подойдет для анализа

вязки можно как Year (Год) и Month (Месяц), но столбец Year содержит нулевые значения в окне редактора Power Query. Кроме того, в строке Supplier Category (Категория поставщика) имеются нулевые значения. Попытка по отмене свертывания столбцов с доходами потерпит неудачу, поскольку в качестве атрибутов необходимо указать пару Supplier Category (Категория поставщика) и Supplier (Поставщик). Если применить преобразование отмены свертывания столбцов ко всем столбцам по годам и месяцам, то поставщики окажутся в столбце Value (Value) вместо столбца Attribute (Атрибут).

Чтобы подготовить почву для применения преобразования отмены свертывания столбцов, следует выполнить последовательность преобразований, которые не кажутся тривиальными: Заполнить вниз (Fill Down) для столбца Year (Год), Объе-динить (Merge) для столбцов Year (Год) и Month (Месяц), Транспонировать (Transpose) для таблицы, Заполнить вниз (Fill Down) для столбца Supplier Category (Категория поставщика), Применить первую строку в качестве заголовков (Apply First Row as Headers). После этого можно выполнить преобразование отмены свертывания столбцов. Эти действия подробно описаны в следующем упражнении:

1. Загрузите файл рабочей книги C06E04.xlsx с сайта https://aka.ms/DataPwrBIPivot/ downloads и сохраните его в папке C:\Data\C06\.

2. Откройте пустую новую рабочую книгу в Excel или новый отчет в Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: в раскрывающемся меню Получить данные (Get Data) выберите команду Excel.

3. Выберите файл C06E04.xlsx, а затем щелкните на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите вкладку Revenues (Доходы) и затем щелкните на кнопке Изменить (Edit).

5. В открывшемся окне редактора Power Query обратите внимание на то, что панель предварительного просмотра включает доходы компании Wide World Importers за год, месяц, категории поставщиков и самих поставщиков, как показано в верхней таблице на рис. 6.6.

6. Пришло время удалить общие итоги, как показано в упражнении 6.3. Можно просто удалить столбец Grand Total (Общий итог), затем удалить нижнюю строку Grand Total (Общий итог).

Для удаления нижней строки на вкладке Главная (Home) выберите команду Удалить строки (Remove Rows), затем выберите команду Удалить нижние строки (Remove Bottom Rows). В открывшемся окне диалога Удалить нижние строки введите единицу в поле Количество строк (Number of Rows) и щелкните на кнопке OK для закрытия этого окна диалога.

СОВЕТ

Обратите внимание, что импортированная таблица имеет заданные по умолчанию имена столбцов Column1, Column2 и т. д. Удаление столбца Grand Total (Column10) с помощью пользовательского интерфейса приводит к следующей формуле M:

= Table.RemoveColumns(#"ra3MeHeHH™ тип ",{"Column10"})

Дополнительную информацию о рисках при удалении столбцов по именам можно найти в главе 10, но стоит упомянуть об этом и в текущем контексте. Если в данном случае удалить столбец Grand Total, при добавлении новых поставщиков в исходную таблицу, Column10 больше не будет представлять общий итог. Вместо удаления столбца Grand Total лучше убрать шаг Измененный тип на панели Примененные шаги (Applied Steps), а затем удалить последний столбец с помощью формулы M. Вот соответствующий код, выполняющий эту задачу:

= Table.RemoveColumns(Revenues\_DefinedName, List.Last(Table.ColumnNames(Revenues\_DefinedName)))

В данной формуле применим функцию Table.ColumnNames для получения списка имен столбцов из таблицы, идентификатором которой является Revenues\_DefinedName. Затем получим имя последнего столбца, применяя формулу List.Last для получения списка имен столбцов.

7. Выберите столбец Column1, на вкладке Преобразование (Transform) выполните команды Заполнить (Fill) Вниз (Down). Этот шаг приводит к заполнению столбца Год (Year) и замене в каждой строке нулевых значений корректными значениями лет. В главе 5 было рассказано о важности команды Заполнить вниз (Fill Down) при сохранении контекста. В данном случае примените ее для сохранения в каждой строке правильного значения года.

Затем нужно объединить столбцы Year (Год) и Month (Месяц) в столбец Date (Дата) для свертывания иерархии год/месяц в одну ячейку. (Вы скоро оцените эффективность этого шага.)

8. Выделите столбцы Column1 и Column2, на вкладке Преобразование (Transform) выберите команду Объединить столбцы (Merge Columns). В открывшемся окне диалога Объединить столбцы выберите параметр Разделитель, пробел (Space as Separator), затем щелкните на кнопке OK для закрытия этого окна диалога.

Рассматривая значения столбца Supplier Category (Категория поставщика) в первой строке, на этом этапе можно воспользоваться преобразованием Запол-нить вправо (Fill Right) для заполнения нулевых значений строки Supplier Category. Подобная функциональность в данном случае отсутствует, но можно транспонировать таблицу, а затем применить преобразование Заполнить вниз (Fill Down) к столбцу Supplier Category (Категория поставщика). Как вы помните, в главе 4 была описана опция Транспонировать (Transpose), которая позволяет реконструировать таблицу так, чтобы, например, каждая ячейка в i-й строке и j-м столбце теперь располагалась в j-й строке и i-м столбце.

9. На вкладке Преобразование (Transform) выберите команду Транспонировать (Transpose).

10. Теперь можно указать категории поставщиков и удалить пустые значения. Выберите первый столбец и на вкладке Преобразование выполните команды Заполнить (Fill), Вниз (Down).

11. На вкладке Преобразование (Transform) выберите команду Использовать первую строку в качестве заголовка (Use First Rows as Headers).

12. Выберите первые два столбца (в данном случае это столбцы привязки, пред-ставляющие категорию поставщика и название поставщика). На вкладке Пре-образование (Transform) щелкните на раскрывающемся меню Отменить свер-тывание столбцов (Unpivot Columns) и выберите команду Отменить сверты-вание других столбцов (Unpivot Other Columns). В качестве альтернативы правой кнопкой мыши щелкните на выбранных столбцах и в контекстном меню выберите команду Отменить свертывание других столбцов (Unpivot Other Columns).

13. Переименуйте столбцы следующим образом:

а. Первый столбец: Supplier Category (Категория поставщика).

б. Второй столбец: Supplier (Поставщик).

в. Третий столбец: Date (Дата).

г. Четвертый столбец: Revenue (Доход).

14. Измените тип столбца Date на Дата (Date). Удостоверьтесь, что для столбца Revenue (Доход) выбран тип Десятичное число (Decimal Number) или Валюта (Currency).

Теперь можно загрузить данные, к которым применено преобразование отмены свертывания столбцов, в рабочую книгу или отчет Power BI.

Для просмотра решения загрузите файлы C06E04 - Solution.xlsx и C06E04 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

УПРАЖНЕНИЕ 6.5. Применение преобразования отмены свертывания столбцов к таблицам с уровнями иерархии 2x2

Обобщим шаги преобразования отмены свертывания столбцов, которые необходимы для преобразования в таблицу фактов любых обобщенных таблиц с уровнями иерархии 2\*2. В упражнении 6.4 столбцы Year (Год) и Month (Месяц) представлены в виде двухуровневой иерархии в каждой строке. Несложно объединить столбцы Year и Month в один столбец даты при сохранении соответствующего контекста. Рассмотрим, каким образом можно обрабатывать случаи, если нельзя объединить двухуровневую иерархию строк.

На рис. 6.7 показана сводная таблица доходов AdventureWorks со столбцами Sales Person (Продавец), Client Company (Клиент компании), Product Parent Category (Родительская категория товара) и Category (Категория). Необходимо реализовать шаги, предшествующие применению отмены свертывания столбцов. Эти операции вам уже знакомы, поскольку они являются обобщением упражнения 6.4.

Рис. 6.7. Power Query можно применять для отмены свертывания столбцов сложных таблиц, таких как эта сводная таблица доходов AdventureWorks. Нижняя таблица, в которой отменено свертывание столбцов, безусловно, лучше подходит для анализа

1. Загрузите файл рабочей книги C06E05.xlsx с сайта https://aka.ms/DataPwrBIPivot/ downloads и сохраните его в папке C:\Data\C06\.

2. Откройте пустую новую рабочую книгу в Excel или новый отчет в Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: в раскрывающемся меню Получить данные (Get Data) выберите команду Excel.

3. Выберите файл C06E05.xlsx и щелкните на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите команду Revenues (Доходы), затем щелкните на кнопке Изменить (Edit). В открывшемся окне редактора Power Query обратите внимание, что панель предварительного просмотра включает доходы компании AdventureWorks, как показано в верхней таблице на рис. 6.7.

5. Удалите столбец и строку Grand Total (Общий итог), как предложено в упраж-нении 6.4.

6. Выделите столбец Column1 и на вкладке Преобразование (Transform) выберите команды Заполнить (Fill), Вниз (Down). Этот шаг завершает формирование

столбца Sales Person (Продавец) и заменяет нулевые значения корректными значениями из каждой строки.

7. Выделите столбцы Column1 и Column2, на вкладке Преобразование (Transform) выберите команду Объединить столбцы (Merge Columns). В открывшемся окне диалога Объединить столбцы (Merge Columns) выберите опцию Разделитель Двоеточие (Colon as Separator), а затем щелкните на кнопке OK для закрытия этого окна диалога.

ПРИМЕЧАНИЕ

При выполнении упражнения 6.4 объединение столбцов было интуитивно понятным шагом, поскольку выполнено преобразование в даты значений в столбцах Year (Год) и Month (Месяц). В данном упражнении объединение столбцов Sales Person (Продавец) и Company (Компания) может сначала показаться деструктивным, поскольку выполняется обработка первых двух столбцов. Однако этот шаг имеет решающее значение для преобразования отмены свертывания столбцов. На шаге 13 будет исправлена временная путаница и разделены столбцы.

Почему при объединении столбцов было использовано двоеточие? Предполагается, что значения в столбцах Sales Person (Продавец) и Company (Компания) вообще не будут содержать двоеточия. Таким образом, можно будет правильно разделить их позже. Если этот подход применяется для ваших собственных таблиц, убедитесь, что задан разделитель, который отсутствует в данных. В качестве разделителя на шаге 7 можно выбирать редкую комбинацию символов.

8. На вкладке Преобразование (Transform) выберите команду Транспонировать (Transpose).

9. Выберите столбец Column1 и на вкладке Преобразование (Transform) выберите команды Заполнить (Fill), Вниз (Down).

10. На вкладке Преобразование выберите опцию Использовать первую строку в качестве заголовка.

11. Выберите первые два столбца (которые являются столбцами привязки, представляющие категорию поставщика и имя поставщика).

12. На вкладке Преобразование (Transform) щелкните на раскрывающемся меню Отменить свертывание столбцов (Unpivot Columns), а затем выберите команду Отменить свертывание других столбцов (UnpivotOther Columns). В качестве альтернативы правой кнопкой мыши щелкните на одном из выбранных столбцов и выберите из контекстного меню пункт Отменить свертывание других столбцов (UnpivotOther Columns).

Затем необходимо внести изменения в столбец Attribute (Атрибут), который состоит из объединенных столбцов Sales Person (Продавец) и Company (Ком-пания).

13. Выберите столбец Attribute (Атрибут) и на вкладке Преобразование (Trans-form) выберите команды Разделить столбец (Split Column) По разделителю (By Delimiter). В открывшемся окне диалога Разделить столбец по разделителю (Split Column By Delimiter) установите переключатель Выберите или введите разделитель (Enter Delimiter to Colon), затем щелкните на кнопке OK для закрытия этого окна диалога.

14. Переименуйте столбцы следующим образом:

а. Первый столбец: Parent Category (Родительская категория).

б. Второй столбец: Category (Категория).

в. Третий столбец: Sales Person (Продавец).

г. Четвертый столбец: Company (Компания).

д. Пятый столбец: Revenue (Доход).

15. Измените тип последнего столбца на Десятичное число (Decimal Number) или Валюта (Currency).

Теперь можно загружать в отчет данные с отменой свертывания столбцов.

Для просмотра решения загрузите файлы C06E05 - Solution.xlsx и C06E05 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

Обработка промежуточных итогов при работе с данными, для которых отменено свертывание столбцов

Ранее в этой главе было показано, почему наличие в таблице итоговых данных может привести к чрезмерному усложнению. Аналогичная ситуация складывается и для промежуточных итогов. Для базовых операций суммирования в сводных таблицах или визуальных представлениях Power BI необходимо применять явную логику фильтрации, что позволит игнорировать промежуточные итоги и приведет к тому, что каждая ячейка в суммированных данных будет добавляться в отчет только один раз. Поэтому важно фильтровать промежуточные итоги на уровне запросов.

В упражнении 6.3 было показано, что удаление итогов — это относительно простой шаг, выполняемый перед применением преобразования отмены свертывания столбцов. Однако удалить столбцы и строки промежуточных итогов из местоположения в исходной таблице довольно сложно. Чтобы проиллюстрировать эту проблему, поработаем с исходной таблицей из упражнения 6.5, которая включает промежуточные итоги в строках и столбцах, как показано на рис. 6.8.

УПРАЖНЕНИЕ 6.6. Обработка промежуточных итогов

В данном упражнении будет показано, каким образом можно отфильтровывать элементы промежуточных итогов после выполнения шага отмены свертывания столбцов. Удалять промежуточные итоги полезно именно после отмены свертывания столбцов, поскольку позиции столбцов для промежуточных итогов в исходной таблице могут измениться. При удалении промежуточных итогов до этого шага возникнут проблемы, если в будущем позиции промежуточных итогов изменятся.

А В c D E F G H 1 К M N О p Q R S T U V W X Y Z AA AB AC

1 Продажи по продавцам клиентам компании, родительской категории и категории

2

3

4 Salesperson CompanyName Accessories

Bike Racks Bcttlesanc Cleaners Helmets Hydration Tires a nd Tubes Bikes Mounta'n Road Bikes Touring 3' Clothing

Caps Gloves kerseys Shorts Socks Vests Components

Bottom Bra Bra kes Chains Cranksets Derailleurs Handlebar Mountain Pedals | Grand Tot

Road FramSaddles Touring Frames

5 jaeO Action Bicycle Specialist 432 31,19943 33,16 490,5378 296,946 10,992 76613,65 53,94 161,634 1173,172 1067,911 63,168 9430,914 89359.23

6 Bulk Discount Store 216 23,952 23,62 236,934 230,958 70597,28 5,394 29,388 441,516 32,364 342,9 72,162 587,954 1320,792 74152,23

7 Central B'cyc e Spedalist 31,584 31,534

3 Cha nnel Outlet 216 194,364 114,3 524,664

9 Closest B'cyc e Store 20339,67 559,1641 136,038 7659,777 158,73 47,856 28952,58

10 Essential Bike Works 37,758 37,758

11 l nstruments a nd Pa rts ton- pa ny 50771,95 388,758 63,9 12,144 1150,158 164,682 94,752 602,346 53248,59

12 MetropoTta n Bicycle supply 20,994 50339,74 38,164 1060.38 388,758 255,6 36,432 1773,846 638.85 292,656 24018,86 390,36 284,472 7=589,52

13 №a rby cycle shop 32,994 25125,71 545,922 183,618 3937,584 97,182 29=23,01

14 Ridi ng Cycles 720 31,19948 70,61588 448,5998 329,94 3.244 51328,24 32,364 249,798 1433,595 16,182 801,2113 144,324 416,274 9029,706 63,168 55123,46

15 sports store 991,1868 818,7 74,508 1384,3=5

16 1376.994 335,952 2712,945

17 Vigorous Sports Store 353,9 858,9

13 west side Mart 63,9 63,9

19 Iinda3 Aerobe Exercise compa ny 257,532 1475,358 1732,89

20 Eastside Department st 144 29,94 38,16 702.9456 296,946 8,244 51096,05 56,20908 264,492 1567,453 5.394 878,6813 72,162 490,782 10031,9 6=683,37

21 Extreme Riding Suppries 41337,94 54,942 31.554 6423,564 47343,'23

22 Good Toys 713,795 713,795

23 Professional Sales and! 216 17,964 47,7 335,904 296,946 27329,09 53,94 14,694 733,45 331 323,97 63,9 435,983 323,416 31,584 2609,436 33319,99

24 Remarkable Bike Store 5426,954 32,61091 24,294 ==33,559

25 Sports Products Store 20,994 2540,564 55,99708 48,538 Z““, 143

26 Thrifty Parts a nd Sales 10,788 916,128 926,915

27 Thrilling Bike Tours 4,77 11138,28 16,182 124,776 127,3 12,144 54,894 11523,34

23 shuO Coalition Bike Company 529,4928 124,776 26,724 1126,62 48,594 1356,207

29 Discount Tours 72 2041,138 226,753 114,3 72,382 2=2",123

30 Engineered Bike System 2504,762 21,576 157,17 63,9 2847,428

31 Futuristic Bikes 59,988 114,3 46,968 221,256

32 Many Bikes Store 288 20,958 23,85 272,922 164,97 49800,6 16,182 29,388 568,692 587,916 457,2 146,466 7221,324 231,618 64,128 5=894,21

33 pa ints and solvents company 10364,29 21,576 97,182 38,1 63,9 10585,05

34 The Bicycle Accessories company 2,994 93,11482 =6.10582

35 Trailblazing sports 22272,5 59.988 209,97 218,682 127,8 36.432 558,876 54,894 63,876 10167,29 97,188 251,04 34113.54

36 Transport Bikes 445,41 54,894 560,304

37 Grand Total 2304 153,207 251,8759 2523,331 1649,7 27,43 170325,9 183130,3 220655,4 277,3631 837,553 7017.BE 3299,805 345,2088 4309,904 1320,168 330,7 97,152 3963,868 1296,63 1192,968 54949,6 2996,496 24346,58 1010,304 19056,26 703590,2

Рис. 6.8. Эта исходная таблица AdventureWorks суммирует доходы и содержит промежуточные итоги

Отмена свертывания столбцов таблицы 207

Например, можно удалить столбец товара вместо столбца промежуточных итогов. Чтобы корректно игнорировать промежуточные итоги, рекомендуется применить преобразование отмены свертывания столбцов, а затем исключить промежуточные итоги с помощью фильтра. Это упражнение иллюстрирует описанный процесс:

1. Загрузите файл рабочей книги C06E06.xlsx с сайта https://aka.ms/DataPwrBIPivot/ downloads и сохраните его в папке C:\Data\C06\.

2. Выберите файл C06E06.xlsx и щелкните мышью на кнопке Импорт (Import).

3. В диалоговом окне Навигатор (Navigator) выберите вкладку Revenues, затем щелкните на кнопке Изменить (Edit).

4. Следуйте шагам 6-15 из упражнения 6.5 для применения к таблице преобразования отмены свертывания столбцов. В качестве альтернативы можно открыть рабочую книгу решения или отчет Power BI из упражнения 6.5 и на панели Примененные шаги (Applied Steps) изменить шаг Source (Источник) и заменить исходный файл C06E05.xlsx на C06E06.xlsx. Кроме того, удалите шаги, которые применялись для удаления столбца или строки Grand Total (Общий итог), — следует удалить их после применения преобразования отмены свертывания столбцов в этом упражнении.

5. В окне редактора Power Query обратите внимание на то, что теперь столбец Parent Category (Родительская категория) включает промежуточные итоги под заголовками Accessories Total, Bikes Total, Clothing Total и Component Total (рис. 6.9). Следует отфильтровать их. Выберите элемент управления фильтром в заголовке столбца Parent Category.

На этом этапе может возникнуть соблазн применить панель Фильтр (Filter) и отменить выбор отметок промежуточных итогов, выделенных на рис. 6.9. Но если так поступить, то запрос отфильтрует только текущие надписи и пропустит в будущем новые надписи при добавлении в исходную таблицу новых родительских категорий.

6. Для корректного выполнения фильтрации выберите команду Текстовые фильтры (Text Filters), затем задайте команду Не заканчивается на (Does Not End With).

7. В открывшемся окне диалога Фильтрация строк (Filter Rows) введите в поле значения Total, а затем щелкните на кнопке OK для закрытия этого окна диалога. Можно проверить шаг фильтрации, щелкая на элементе управления фильтром в заголовке столбца Parent Category и убеждаясь, что все значения Total, выделенные на рис. 6.9, включая Общий итог (Grand total), теперь отсутствуют.

Применение этого фильтра позволяет удалить из столбцов исходной таблицы промежуточные итоги родительской категории. Но еще имеются промежуточные итоги продавца из строк исходной таблицы (обратитесь к рис. 6.8 для просмотра промежуточных итогов строки и столбца).

8. Выберите элемент управления фильтром в заголовке столбца Sales Person. Ви-дите ли вы общие значения? Для их удаления выберите опцию Текстовые фильтры (Text Filters), а затем задайте опцию Не заканчивается на (Does Not End With).

Рис. 6.9. Надписи столбца Parent Category следует фильтровать после применения преобразования отмены сведения столбцов. Не поддавайтесь искушению отфильтровать их, отменив выбор значений.

Вместо этого воспользуйтесь текстовыми фильтрами

9. В открывшемся окне диалога Фильтры строк (Filter Rows) введите в поле зна-чения Total, затем щелкните на кнопке OK для закрытия этого окна диалога. Можно проверить шаг фильтрации, снова щелкая на элемент управления фильтром в заголовке столбца Sales Person и убедившись, что все значения итогов теперь отсутствуют.

10. Загрузите в отчет данные, для которых отменено свертывание столбцов. Теперь будут отфильтрованы промежуточные и общие итоги.

Для просмотра решения загрузите файлы C06E06 - Solution.xlsx и C06E06 - Solution.pbix с сайта aka.ms/DataPwrBIPivot/downloads.

Резюме

Вновь повторяя приведенную ранее лингвистическую аналогию, в главе переходим из области синтаксиса (или очистки данных) и семантики (или сохранения контекста) в область прагматики, где различные контексты могут формировать воспринимаемое значение. Показано, что даже если данные очищены и не искажены, они могут быть неудовлетворительно подготовлены для использования в сводных таблицах или визуальных представлениях Power BI.

В этой главе дано определение неудовлетворительно подготовленным (плохо спроектированным) таблицам и рассказано о преобразовании отмены свертывания столбцов и трех различных его вариантах: Отменить свертывание столбцов (Unpivot Columns), Отменить свертывание других столбцов (Unpivot Other Columns) и Отменить свертывание только выделенных столбцов (Unpivot Only Selected Columns).

В главе было показано, как отменять свертывание столбцов для простых таблиц и как обращаться к итогам по строкам и столбцам в исходной таблице. Преобразование отмены свертывания столбцов также применялось к сводным таблицам с уровнями иерархии 2\*2. В простом сценарии имелись столбцы Year (Год) и Month (Месяц), которые можно легко объединить в один столбец, чтобы облегчить подготовку для применения преобразования отмены свертывания столбцов. Затем мы перешли к обобщенному решению, где преобразование объединения необходимо в качестве промежуточного шага перед выполнением преобразования отмены свертывания столбцов.

В данных упражнениях, связанных с применением отмены свертывания столбцов, уже встречались аналогичные шаги: заполняют столбцы привязки, объединяют их в один столбец, транспонируют таблицу, заполняют новые столбцы привязки, выбирают опцию Использовать первую строку в качестве заголовков (Use First Row as Headers), применяют преобразование отмены свертывания столбцов и разделяют столбец Атрибут (Attribute).

Наконец, в данной главе продемонстрирована простая методика фильтрации промежуточных итогов после шага отмены свертывания столбцов, что позволит избежать явного и рискованного удаления столбцов и строк промежуточных итогов перед применением отмены свертывания столбцов, поскольку их положение может измениться.

В главе 7 будет продолжено изучение преобразования отмены свертывания столбцов и обобщено решение, которое поможет выполнить это преобразование для любой сводной таблицы, независимо от числа уровней иерархии в строках и столбцах. Также будет рассказано о случаях чрезмерного увлечения применением преобразования отмены свертывания столбцов и о том, как скорректировать это с помощью преобразования сведения в столбец, обратного к преобразованию отмены свертывания столбцов.

Глава 7

Дополнительные вопросы, связанные с применением преобразования сведения столбцов и отмены свертывания столбцов

Некоторые художники превращают солнце в желтое пятно, другие же превращают желтое пятно в солнце.

- Пабло Пикассо

В ЭТОЙ ГЛАВЕ ВЫ...

■ Познакомитесь с применением преобразования отмены свертывания столбцов к сводным таблицам с уровнями иерархии 3\*3 и научитесь обобщать шаги данного преобразования для любой сводной таблицы.

■ Получите полное представление о том, как преобразовать в функцию последовательность отмены свертывания столбцов и применять ее к нескольким файлам из папки и нескольким рабочим листам, где это необходимо.

■ Научитесь с помощью сведения столбцов отменять преобразование отмены свертывания столбцов, а также использовать сведение столбцов при трансформации многострочных записей в таблицы фактов.

В этой главе будет продолжено изучение преобразования отмены свертывания столбцов, причем, обобщая решение, вы научитесь отключать любую сводную таблицу, независимо от числа уровней иерархии в виде строк и столбцов. Также узнаете, как обрабатывать сценарии, в которых преобразование отмены свертывания столбцов встречается слишком часто, и каким образом можно внести коррективы, задействуя преобразование сведения столбцов, которое является обратным по отношению к преобразованию отмены свертывания столбцов. Показано, как с помощью преобразования сведения столбцов можно трансформировать в таблицы многострочные записи.

На некоторых этапах работы потребуются возможности M, языка формул Power Query. Поскольку до сих пор язык М был лишь кратко представлен в главе 1, а также были приведены небольшие ссылки на него в других главах, предлагается вернуться к этой главе снова после изучения подробного обзора возможностей M в главе 9.

Рассматриваемые в этой главе методы позволят выполнить широкий спектр преобразований и научиться преобразовывать чрезмерно структурированные наборы данных в универсальную и гибкую аналитическую платформу. Многие аналитики осведомлены, как оформить собственные представления в виде статических сводных таблиц, а вы дополнительно научитесь преобразовывать в представления любые статические таблицы.

Применение преобразования отмены свертывания столбцов к таблицам

с несколькими уровнями иерархии

В главе 6 рассматривалась важность преобразования отмены свертывания столбцов для обработки сводных таблиц и изменения их структуры, что позволит применять данные таблицы для таблиц в Excel и визуальных представлений в Power BI. В главе 6 сначала рассматривалась обработка простых таблиц, затем мы перешли к более сложным таблицам с двумя уровнями иерархии.

Теперь продолжим рассмотрение этой темы и познакомимся с более сложной процедурой отмены свертывания столбцов, которая включает три уровня иерархии как в строках, так и в столбцах. В данной главе вы научитесь выстраивать шаги таким образом, чтобы в конечном итоге можно было применить обобщенную технику к любой обобщенной таблице с несколькими уровнями иерархии.

Виртуальная сводная таблица, определение полей строк и столбцов

Для обобщения шагов преобразования, необходимых для применения преобразования отмены свертывания столбцов к сводной таблице любого типа, следует четко понимать термины поля строк и поля столбцов и научиться представлять сводную таблицу как виртуальную сводную таблицу.

Если источником данных служит сводная таблица, можно рассматривать ее таким образом, как будто она является сводной таблицей в Excel. И хотя это не настоящая сводная таблица, а ее поля "жестко закодированы", визуализация исходной таблицы в виде сводной таблицы Excel и определение полей, которые войдут в таблицу Excel, помогут применить преобразование отмены свертывания столбцов. На рис. 7.1 показано, какой вид получит исходная таблица в форме сводной таблицы, какие поля будут перетаскиваться в область строк, столбцов и значений.

Исходная таблица, показанная на рис. 7.1, обобщает суммы доходов по товарам компании AdventureWorks, причем в качестве критерия выбирается страна, штат/регион, город, цвет, родительская категория и обычная категория товара. При визуализации исходной таблицы в качестве сводной таблицы можно определить, какие именно поля заданы в виде строк и столбцов в области полей сводной таблицы.

Загрузите файл C07E01.xlsx с исходной таблицей с сайта https://aka.ms/ DataPwrBIPivot/downloads. Как показано на рис. 7.1, исходная таблица напоминает сводную таблицу с полями Country (Страна), State/Region (Штат/Регион) и City (Город), которые находятся в области строк, а поля Color (Цвет), Parent Category (Родительская категория) и Category (Категория) расположены в области столбцов. Поле Revenue (Доход) находится в области значений.

На данный момент, возможно, еще непонятно, почему на этом этапе следует обращаться к представлению PivotTable (сводной таблицы). Теперь же нужно формализовать термины, которые понадобятся в следующем разделе. Если же требуется применить преобразование отмены свертывания столбцов к обобщенной сводной таблице, то можно представить, какой вид она будет иметь при использовании классической сводной таблицы, и выполнить перетаскивание столбцов из таблицы, в которой отменено свертывание столбцов, в области полей сводной таблицы. В следующем разделе термин "поле области столбцов" служит для указания столбцов в таблице, где отменено свертывание столбцов, которые затем перетаскиваются в раздел строк виртуальной области полей сводной таблицы. Следуя той же логике, термин "поле столбцов" вводится для указания столбцов в таблице, в которой отменено свертывание столбцов, которые затем перетаскиваются в раздел столбцов области полей сводной таблицы.

Теперь можно перейти к первому упражнению, в котором преобразование отмены свертывания столбцов применяется к сводной таблице AdventureWorks с уровнями иерархии 3\*3. Ради обобщения показанных здесь шагов будем ссылаться на эту таблицу как N\*M вместо 3\*3. Подобный подход поможет лучше понять материал, излагаемый далее в этой главе.

УПРАЖНЕНИЕ 7.1. Применение преобразования отмены свертывания столбцов к таблице AdventureWorks с уровнями иерархии N\*M

Приступим к выполнению шагов преобразования отмены свертывания столбцов по отношению к исходной таблице, показанной на рис. 7.1. Прежде чем продолжить, обратим внимание на два важных момента. Если вы выполняли упражнения в главе 6, то, наверное, заметили, что приведенная методика схожа с той, что применялась для таблиц размером 2\*2. На этот раз мы будем рассматривать более общий случай, с применением подобных инструкций при обработке нескольких уровней иерархии, при наличии N строк и M столбцов.

Основные шаги, необходимые для выполнения преобразования отмены свертывания столбцов в любой обобщенной сводной таблице, показаны на рис. 7.2. На этом рисунке наглядно видна вся последовательность операций. Если вы выполняли упражнение 6.5 (см. главу 6), то эта последовательность покажется вам довольно знакомой.

1. Загрузите файл C07E01.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C07\.

2. Откройте новую пустую книгу в Excel или новый отчет в Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: в раскрывающемся меню Получить данные (Get Data) выберите команду Excel.

3. Выберите файл C07E01.xlsx и щелкните мышью на кнопке Импорт (Import).

4. В диалоговом окне Навигатор (Navigator) выберите вкладку Revenues (Доходы), затем щелкните на кнопке Изменить (Edit).

После открытия окна редактора Power Query можно визуализировать в своем представлении виртуальные поля сводной таблицы — поля строк и поля столбцов: в разделе полей строк находятся поля Country (Страна), State/Region (Штат/Регион) и City (Город). В области полей столбцов находятся поля Color (Цвет), Parent Category (Родительская категория) и Category (Категория). Буква N обозначает количество полей строк; в данном случае N = 3. Буква M обо-значает количество полей столбцов; в данном случае M = 3.

5. Обратите внимание, что первые два столбца (N-1) в панели предварительного просмотра включают нулевые значения (вследствие иерархической структуры полей строк). Выберите первые N-1 столбцов и на вкладке Преобразование (Transform) выполните команды Заполнить (Fill) Вниз (Down).

1 Заполнить вниз N-1 столбцов

Columnl Column N-1 Column N

Black

Accessories Bikes

Helmets Mountain Bikes Road Bikes

United Kingdom

1 England

1 Liverpool 238.659792 24565.028

London 104.97 33290.832 33038.094

Milton Keynes 14012.916

2. Объединить N столбцов с помощью разделителя

5. Использовать первую строку в качестве заголовков

Columnl Column M-l Column M A

United Kingdom:England;Liverpool United Kingdom:England:London United Kingdom;...

Black Accessories Helmets 238.659792 104.97

Bikes Mountain Bikes 33290.832 14012.916

Road Bikes 24566.028 33038.094

6. Выбрать М столбцов и отменить свертывание других столбцов

United Kingdom:England;Liverpool United K1ngdom;England;London United Kingdom:...

Black Accessories Helmets 238.659792 104.97

Black Bikes Mountain Bikes 33290.832 14012.916

Black Bikes Road Bikes 24566.028 33038.094

7. Разделить атрибут по разделителю

Attribute Value

Black Accessories Helmets United KingdomlEnglandliverpool 238.659792

Black Bikes Road Bikes United KingdomlEnglandiiverpool 24566.028

Black Accessories Helmets United Kingdom;England j.ondon 104.97

Black Bikes Mountain Bikes United Kingdcm;Englandiondon 33290.832

Black Bikes Road Bikes United Kingdom'EnglandJ-ondon 33038.094

Black Bikes Mountain Bikes United Kingdom\*England\*Klilton Keynes 14012.916

8. Переименовать столбцы

—> Column Field 1 Column Field 2 —> Column Field M Attribute.l —> Row Field 1 Attribute.2 Row Field 2 Attribute.N —> Row Field N Value

Black Accessories Helmets United Kingdom England Liverpool 238.66

Black Bikes Road Bikes United Kingdom England Liverpool 24566

Black Accessories Helmets United Kingdom England Liverpool 104.97

Black Bikes Mountain Bikes United Kingdom England Liverpool 33290.8

Black Bikes Road Bikes United Kingdom England Liverpool 33038.1

Black Bikes Mountain Bikes United Kingdom England Liverpool 14012.9

Рис. 7.2. Используя данную последовательность операций, можно применить преобразование отмены свертывания столбцов к любой обобщенной сводной таблице с уровнями иерархии NxM

Затем нужно консолидировать все строки и объединить их. При выполнении упражнения 6.5 преобразование Объединить (Merge) применялось по отношению к двум столбцам. В данном случае оно будет применено к трем столбцам, или, в более общем случае, к N столбцам.

6. Выберите первые N столбцов и на вкладке Преобразование щелкните на кноп-ке Объединить столбцы (Merge Columns). Откроется окно диалога Объединить столбцы (Merge Column). В поле Разделитель (Separator) выберите параметр Двоеточие (Colon) и щелкните на кнопке OK, чтобы закрыть это окно диалога.

ПРИМЕЧАНИЕ

Убедитесь, что выбран разделитель, который отсутствует в наборе исходных данных. В качестве разделителя вместо двоеточия можно выбрать редкую комбинацию символов. Запомните выбранный разделитель, он понадобится на шаге 12.

7. Перейдите на вкладку Преобразование (Transform) и щелкните на кнопке Транспонировать (Transpose).

8. Для заполнения полей столбцов в направлении вниз и удаления нулей из более высоких уровней иерархии выберите первые M - 1 столбцов и на вкладке Преобразование (Transform) выполните параметры Заполнить (Fill) Вниз (Down).

9. На вкладке Преобразование (Transform) щелкните мышью на кнопке Исполь-зовать первую строку в качестве заголовков (Use First Row as Headers).

10. Выберите первые M столбцов.

11. На вкладке Преобразование выберите раскрывающееся меню Отменить свер-тывание столбцов (Unpivot Columns), а затем выберите параметр Отменить свертывание других столбцов (Unpivot Other Columns). В качестве альтерна-тивы правой кнопкой мыши щелкните на одном из выбранных заголовков столбцов и выберите в контекстном меню параметр Отменить свертывание других столбцов.

Теперь столбец Attribute (Атрибут) содержит объединенные комбинации зна-чений полей строк (например, United Kingdom:England:Liverpool), которые были объединены на шаге 6.

12. Выберите столбец Атрибут (Attribute) и на вкладке Преобразование (Transform) выполните команды Разделить столбец (Split Column) По разделителю (By Delimiter). В открывшемся окне диалога Разделить столбец по разделителю (Split Column By Delimiter) выберите поле Выберите или введите разделитель (Select или Enter Delimiter) задайте разделитель Двоеточие (Colon) и закройте это окно диалога.

ПРИМЕЧАНИЕ

Как отмечено на шаге 6, если данные уже содержат символ двоеточия, используйте комбинацию символов, которую применяли в качестве разделителя на шаге 6, и введите их вместо двоеточия на шаге 12.

13. Переименуйте первые M столбцов, являющиеся полями столбцов, следующим образом: Color (Цвет), Parent Category (Родительская категория) и Category (Категория). Переименуйте следующие N столбцов, которые являются полями строк, выбрав следующие названия: Country (Страна), State/Region (Штат/ Регион) и City (Город).

14. Переименуйте столбец Value (Значение) с учетом семантики данных (в на-стоящем случае выберите название Revenue (Доход)) и явно измените тип столбца на Десятичное число (Decimal Number) или Валюта (Currency).

15. Загрузите в рабочую книгу или отчет Power BI данные после выполнения преобразования отмены свертывания столбцов.

Для просмотра решения загрузите файлы C07E01 - Solution.xlsx и C07E01 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

Обобщение последовательности преобразований отмены свертывания столбцов

В предыдущем разделе выполнялась отмена свертывания столбцов обобщенной таблицы с уровнями иерархии 3\*3. В данном разделе будет показано, каким образом можно изменить формулы и преобразовать запрос в обобщенную функцию, которую можно затем использовать для различных таблиц. Польза подобной функции очевидна: при наличии десятков различных отчетов, в каждом из которых в качестве исходных данных применяются различные обобщенные таблицы, именно функция позволит быстро их обработать.

С помощью такой функции вместо ручного выполнения предыдущей последова-тельности для каждого типа обобщенной таблицы можно объединить все шаги в один шаг преобразования, который будет пригоден для любой обобщенной таб-лицы. В данном разделе усовершенствуем запрос, созданный в упражнении 7.1, и преобразуем последовательность отмены свертывания столбцов для работы с любым типом обобщенной таблицы при произвольном уровне иерархии в строках и столбцах.

УПРАЖНЕНИЕ 7.2. Начнем с конца

Перед рассмотрением обобщения последовательности преобразования отмены свертывания столбцов обратимся к небольшому упражнению, в котором показано, как использовать обобщенную функцию и насколько просто с ее помощью отменить свертывание столбцов различных таблиц. Импортируем обобщенную таблицу AdventureWorks с уровнями иерархии 2x2 (показана в верхней части панели предварительного просмотра на рис. 7.3) и обратимся к функции, применяющей последовательность преобразований отмены свертывания столбцов в упражнении 7.1 (показана в нижней части панели предварительного просмотра на рис. 7.3).

1. Загрузите файл рабочей книги C07E02.xlsx или отчет Power BI C07E02.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads и сохраните в папке C:\Data\ C07\.

Рис. 7.3. При выполнении данного упражнения обратимся к функции, которая выполняет всю последовательность отмены свертывания столбцов для новой обобщенной таблицы

2. Откройте файл в Excel или Power BI Desktop и запустите редактор Power Query:

В Excel: откройте книгу C07E02.xslx и на вкладке Данные (Data) выберите команды Получить данные (Get Data) Запустить редактор запросов (Launch Power Query Editor).

В Power BI Desktop: откройте отчет C07E02.pbix и на вкладке Главная страница (Home) выберите команду Изменить запросы (Edit Queries).

3. В открывшемся окне редактора Power Query на панели Запросы (Queries) выберите запрос Revenues (Доходы). Как показано на панели предварительного просмотра, импортирована другая обобщенная таблица AdventureWorks, с уровнями иерархии 2\*2 (см. верхнюю таблицу на рис. 7.3). В данном случае полями строк в виртуальной сводной таблице служат поля Parent Category (Родительская категория) и Category (Категория), а полями столбцов являются Color (Цвет) и Country (Страна).

Можно заметить, что на панели Запросы (Queries) появился еще один элемент: FnUnpivotSummarizedTable, помеченный значком fx. Эта функция сформирована на основе запроса, созданного при выполнении упражнения 7.1. В следующем разделе будет показано, как создать эту функцию. На данном этапе вы просто узнаете, как использовать эту функцию для отмены свертывания столбцов при выполнении запроса Revenues. Вместо того чтобы полностью соблюдать последовательность действий для отмены свертывания столбцов из упражнения 7.1, можно отменить свертывание столбцов в таблице Revenues всего лишь за два простых шага.

4. При выбранном запросе Revenues щелкните мышью на кнопке fx в строке фор-мул. Новый шаг состоит в создании следующей формулы:

= Источник

5. Измените эту формулу следующим образом, а затем нажмите клавишу <Enter>:

= FnUnpivotSummarizedTable(Source, {"Parent Category ", "Category"}, {"Color", "Country"}, "Revenue")

Эта формула преобразует обобщенную таблицу в таблицу фактов (см. нижнюю таблицу на рис. 7.3).

Как видите, эта функция делает просто чудеса. Представьте, что имеется 50 разных отчетов, созданных на основе различных типов сводных таблиц. В таком случае данная функция будет просто незаменима.

Рассмотрим аргументы функции FnUnpivotSummarizedTable. Первым аргументом служит исходная таблица — в данном случае Source. Второй аргумент — список полей строк: Parent Category (Родительская категория) и Category (Категория). Фигурные скобки определяют список и заключают в него разделенные запятыми элементы. Третий аргумент — это список полей столбцов: Color (Цвет) и Country (Страна). И наконец, четвертым аргументом служит название, которое необходимо использовать для столбца значений, — Revenue (Доход).

СОВЕТ

Функцию FnUnpivotSummarizedTable можно применять при обработке любой обобщенной таблицы. Для этого в книгах Excel или отчетах Power BI необходимо копировать данную функцию. Простая ссылка на функцию FnUnpivotSummarizedTable в формулах в рабочей книге или отчете, без копирования реальной функции, не будет работать. Для копирования функции нужно щелкнуть на ней правой кнопкой мыши в панели Запросы (Queries) и в контекстном выбрать команду Копировать (Copy). Затем в назначенной рабочей книге или отчете Power BI правой кнопкой мыши щелкните на пустой области в панели Запросы (Queries) и выберите команду Вставить (Paste). Также в окне редактора Power Query можно открыть расширенный редактор для файла C07E02.xslx или C07E02.pbix и скопировать исходный код. Затем создайте пустой запрос в назначенном отчете, перейдите в расширенный редактор и вставьте запрос.

В упражнении 7.3 будет показано, как можно пошагово создавать функцию FnUnpivotSummarizedTable.

УПРАЖНЕНИЕ 7.3.

Создание функции FnUnpivotSummarizedTable

При выполнении этого упражнения вы узнаете о том, как можно обобщить последовательность операций для отмены свертывания столбцов, созданную при выполнении упражнения 7.1, и превратить ее в функцию. Примененные здесь методы полезны не только для обобщения последовательности операций для отмены свертывания столбцов при работе с широким спектром обобщенных таблиц, но и помо

гут оценивать возможности уточнения формул языка M для обобщения и повышения надежности в процессе автоматизации подготовки данных.

Часть 1. Создание отчетов RowFields, ColumnFields и ValueField

Начните с обращения к результатам упражнения 7.1 или загрузите книгу C07E01 - Solution.xlsx или отчет Power BI C07E01 - Solution.pbix с сайта https://aka.ms/ DataPwrBIPivot/downloads.

1. Откройте файл C07E01 - Solution.xlsx либо C07E01 - Solution.pbix и запустите редактор Power Query.

2. Откройте пустой запрос следующим образом:

В Excel: правой кнопкой мыши щелкните на панели Запросы (Queries), выпол-ните команды Новый запрос (New Query) Другие источники (Other Sources), а затем выберите команду Пустой запрос (Blank Query).

В Power BI Desktop: правой кнопкой мыши щелкните на панели Запросы (Queries) и выполните команды Новый запрос (New Query) Пустой запрос (Blank Query).

3. Переименуйте новый запрос в RowFields и скопируйте следующий код в строку формул для создания списка полей строк:

= {"Country", "State/Region", "City"}

4. Создайте новый пустой запрос. Переименуйте его в ColumnFields и скопируйте в строку формул следующий код для создания списка полей столбцов:

= {"Color", "Parent Category", "Category"}

ПРИМЕЧАНИЕ

Важно назвать новые запросы как RowFields и ColumnFields. Вскоре вы будете использовать их в коде для обобщения шагов из упражнения 7.1. Обратите внимание, что в упражнении 7.1 буква N представляет длину запроса RowFields, а буква M — длину запроса ColumnFields.

5. Создайте новый пустой запрос. Переименуйте его в ValueField, и скопируйте следующий код в строку формул:

= "Revenue"

Часть 2. Удаление шагов Измененный тип

При выполнении этой части упражнения удаляются шаги Измененный тип, кото-рые слишком специфичны для текущего источника данных и не помогают обоб-щать запрос для обработки произвольной таблицы.

6. В панели Запросы (Queries) выберите запрос Revenues (Доход). На панели Примененные шаги (Applied Steps) выберите третий шаг, Измененный тип. В строке формул отобразится следующий код:

= Table.TransformColumnTypes(Revenues\_DefinedName,{{"Column1", type text}, {"Column2", type text}, {"Column3", type text}, {"Column4", type any}, {"Column5", type any}, {"Column6", type any}, {"Column7", type any}, {"Column8", type any}, {"Column9", type any}, {"Column10", type any}, {"Column11", type any}, {"Column12", type any}, {"Column13", type any}, {"Column14", type any}, {"Column15", type any}, {"Column16", type any}, {"Column17", type any}, {"Column18", type any}, {"Column19", type any}, {"Column20", type any}, {"Column21", type any}, {"Column22", type any}, {"Column23", type any}, {"Column24", type any}, {"Column25", type any}, {"Column26", type any}, {"Column27", type any}, {"Column28", type any}, {"Column29", type any}, {"Column30", type any}, {"Column31", type any}, {"Column32", type any}, {"Column33", type any}})

Данный шаг генерируется автоматически при импорте исходной таблицы в редактор Power Query. Если в дальнейшем будете работать с таблицами меньшего размера, в которых нет одного из перечисленных здесь столбцов, например Column33, запрос не будет выполняться. Сейчас предпочтительно удалить этот шаг, поэтому щелкните на значке Удалить (Delete) для шага Измененный тип на панели Примененные шаги (Applied Steps), а затем в окне диалога Удалить шаг (Delete Step) выберите команду Удалить (Delete).

См. также

В главе 10 более подробно рассказано о шаге Измененный тип и о том, почему предпочтительнее его удалить и при необходимости явно изменять типы.

7. На панели Примененные шаги (Applied Steps) выберите шаг Измененный шаг1. Далее приводится фрагмент формулы, которая появится в строке формул:

= Table.TransformColumnTypes(#"noebweHHHe заголовки ",{{"::Color", type text}, {"::Parent Category ", type text}, {"Country:State/Region:City\Category", type text}, {"United Kingdom:England:Abingdon", type number}, {"United Kingdom:England:Cambridge", type number}, {"United Kingdom:England: Gloucestershire", type number}, {"United Kingdom:England:High Wycombe", type number}, {"United Kingdom:England:Liverpool", type number}, {"United Kingdom:England:London", type number}, ...

Подобно шагу Измененный тип, который упоминался на шаге 6, данный шаг был сформирован при выборе в упражнении 7.1 параметра Использовать пер-вую строку в качестве заголовка (Use First Row as Headers). Данная формула бесполезна и не будет обновляться, если применять ее к другим таблицам. Таким образом, следует удалить этот шаг, поэтому щелкните на значке Удалить (Delete) для шага Измененный тип1, а затем в окне диалога Удалить шаг (Delete Step) выберите команду Удалить (Delete).

8. На панели Примененные шаги (Applied Steps) выберите шаг Измененный тип2. В строке формул отобразится следующий код:

= Table.TransformColunmTypes(#"Pa3ge.nMTb столбец по разделителю", {{"Атрибут.1", type text}, {"Атрибут.2", type text}, {"Атрибут.3", type text}})

Этот шаг был сгенерирован при разделении столбца Атрибут (Атрибут) на три столбца. Этот столбец также можно удалить, поскольку он не будет применяться в некоторых ситуациях. Например, для преобразования отмены свертывания столбцов при двух уровнях иерархии нет смысла применять код, изменяющий типы трех столбцов. Поэтому щелкните на значке Удалить (Delete) для шага Измененный тип2, а затем в окне диалога Удалить шаг (Delete Step) выберите команду Удалить (Delete).

Часть 3. Функции Table.ColumnNames, List.FirstN и List.Count

В этой части упражнения изменим формулы, которые явно ссылаются на имена столбцов, и обобщим их для формирования динамических ссылок на столбцы, исходя из значений длин полей RowFields и ColumnFields, которые были созданы в части 1 упражнения. При этом будут применяться следующие три полезные функции: Table.ColumnNames, List.FirstN и List.Count.

9. На панели Примененные шаги (Applied Steps) выберите шаг Заполнить вниз.

В строке формул появится следующий код:

= Table.FillDown(Revenues\_DefinedName, {"Column1", "Column2"})

Этот код был сгенерирован на шаге 5 из упражнения 7.1 и заполнен первыми столбцами N - 1 (представляют собой все поля строк, кроме последней). Следует модифицировать код и изменить жестко закодированный список {"Column1", "Column2"} на динамический список имен первых N - 1 столбцов в текущей таблице.

Прежде всего кратко рассмотрим, почему в этой формуле применяется шаг Revenues\_DefinedName вместо шага Навигация. Посмотрев на панель Приме-ненные шаги (Applied Steps), заметим, что шаг Заполнение вниз находится ниже шага Навигация. Обычно в строке формул отображается ссылка на имя таблицы, которое совпадает с именем предыдущего шага на панели Примененные шаги (Applied Steps). Но в данном случае видно, что в формуле шаг Revenues\_DefinedName применяется вместо шага Навигация.

Чтобы понять, в чем дело, выберите команду Расширенный редактор (Advanced Editor) на вкладке Главная (Home) и обратите внимание на следую-щую строку:

Revenues\_DefinedName = Source {[Item="Revenues", Kind="DefinedName"]}[Data],

Как видите, корректным идентификатором для применения преобразования Заполнить вниз является Revenues\_DefinedName (данный идентификатор применяется перед первым знаком равенства). Теперь, когда известно, как обратиться к предыдущему шагу, изменим шаг Заполнить вниз, чтобы в общем случае заполнить N - 1 столбцов.

Вот исходная формула Заполнить вниз, показанная в строке формул:

= Table.FillDown(Revenues\_DefinedName, {"Column1", "Column2"})

Теперь изменим ее на следующую формулу:

= Table.FillDown(Revenues\_DefinedName,

List.FirstN(

Table.ColumnNames(Revenues\_DefinedName),

List.Count(RowFields) - 1))

Рассмотрим изменения, имеющие место в данном случае. Функция Table.FillDown получает статический список в качестве второго аргумента. Для превращения его в динамический список сначала необходимо получить список всех имен

столбцов таблицы. Это можно сделать с помощью функции Table.ColumnNames на предыдущем шаге Revenues\_DefinedName:

Table.ColumnNames(Revenues\_DefinedName)

В предыдущей формуле применяется шаг Revenues\_DefinedName для ссылки на таблицу, которая поддерживается на шаге Навигация.

Функция Table.ColumnNames уже применялась ранее в примерах из других глав. В данном случае она возвращает все имена столбцов для текущей таблицы:

Columnl, Column2, Column3 и т. д. Для возвращения только первых N - 1 столбцов необходимо вычислить значение N - 1. В данном случае N представляет счетчик списка RowFields:

List.Count(RowFields)

Поэтому N - 1 будет иметь следующий вид:

List.Count(RowFields) - 1

Теперь, когда известно, как вычислить N - 1, можно получить имена первых N - 1 столбцов, применяя функцию List.FirstN, которая возвращает список с первыми элементами исходного списка. Функция List.FirstN имеет два аргумента: список ввода и количество первых элементов для выборки. Можно сформировать список {"Column1", "Column2"} с помощью следующего кода:

List.FirstN(Table.ColumnNames(Revenues\_DefinedName), List.Count(RowFields) - 1)

Теперь эта формула применяется в качестве второго аргумента в обновленной функции Table.FillDown:

= Table.FillDown(Revenues\_DefinedName,

List.FirstN(

Table.ColumnNames(Revenues\_DefinedName), List.Count(RowFields) - 1))

10. На панели Примененные шаги (Applied Steps) выберите шаг Объединенные столбцы. Этот шаг сгенерирован при объединении первого из N столбцов. В строке формул появится следующая жестко закодированная ссылка на список {"Column1", "Column2", "Column3"}:

= Table.CombineColumns(

#"Заполнено вниз ",

{"Column1", "Column2", "Column3"},

Combiner.CombineTextByDelimiter(":", QuoteStyle.None),

"Объединено")

ПРИМЕЧАНИЕ

В строке формул отображается однострочная версия этой формулы. Для удобства чтения формула приведена здесь в многострочном формате.

Для получения динамической ссылки на N имен столбцов исходной таблицы можно применить тот же код, как и на шаге 8, но на этот раз с N в качестве второго аргумента функции List.FirstN вместо N - 1.

Далее приводится модифицированная формула для шага Объединенные столбцы (представлена в многострочном формате):

= Table.CombineColumns( #"Заполнено вниз", List.FirstN(

Table.ColumnNames(#"3anojiHeHO вниз"), List.Count(RowFields)

),

Combiner.CombineTextByDelimiter(":", QuoteStyle.None), "Объединено")

11. Перейдите к следующему шагу, который требует изменения на панели Примененные шаги: Заполнено вниз1. Этот шаг сгенерирован ранее для второй операции заполнения (см. упражнение 7.1, шаг 8). Вот код, который появляется в строке формул:

= Table.FillDown(#"TpaHcnoHupoBaHHa4 таблица", {"Columnl", "Column2"})

Теперь следует изменить жестко закодированные имена столбцов с учетом динамической версии. На этот раз, поскольку таблица транспонирована и обработаны поля столбцов, необходимо заполнить первые M - 1 столбца ColumnFields. Вот модифицированный код:

= Table.FillDown(

#"Транспонированная таблица",

List.FirstN(

ТаЬ1е.Со1итпЫал^(#"Транспонированная таблица") , List.Count(ColumnFields) - 1

)

)

В отличие от модифицированной функции Table.FillDown (см. шаг 9), где применялась функция List.Count для области RowFields, на этом шаге используется функция List.Count для области ColumnFields для получения числа M - 1.

12. Для улучшения шага отмены свертывания столбцов на панели Примененные шаги (Applied Steps) выберите шаг Отменить свертывание других столбцов. В строке формул будут отображены следующие жестко закодированные столбцы:

= Table.UnpivotOtherColumns(#"noBbmieHHbie заголовки ", {"::Color", "::Parent Category ","Country:State/Region:City\Category"}, "Атрибут", "Значение")

При выполнении упражнения 7.1 команда Отменить свертывание других столбцов (Unpivot Other Columns) применялась путем выделения первых M столбцов. Для динамического получения первых M столбцов применим сле-дующий обновленный код:

= Table.UnpivotOtherColumns( #"Повышенные заголовки ",

List.FirstN(

Table.ColulIlnNames(#"Повышенные заголовки "),

List.Count(ColumnFields)

),

"Атрибут", ValueField

)

Здесь применяется эта же методика, за исключением того, что M - 1 заменено на M, и функция Table.ColumnNames применяется к таблице #"Повышенные заголовки, что представляет последний шаг преобразования.

Также включается запрос ValueField вместо "Значение" в качестве нового имени столбца Значение (Value). Эта модификация позволит использовать название столбца Revenue (Доход). Можно сделать это позже, на этапе переименования.

13. На панели Примененные шаги (Applied Steps) выберите шаг Разделить столбец по разделителю (Split Column By Delimiter). В результате появится сле-дующий код, который разбивает столбец Attribute (Атрибут) на N столбцов Attribute.1, Attribute.2 и т. д. Далее приведена многострочная запись этого кода:

= Table.SplitColumn(

#"Отменить свертывание других столбцов",

"Attribute", Splitter.SplitTextByDelimiter(":", QuoteStyle.Csv), {"Attribute.1", "Attribute.2", "Attribute.3"}

)

Третий аргумент функции Table.SplitColumn определяет два важных элемента: число столбцов для разделения и названия разделенных столбцов. Вместо названий Attribute.X, заданных по умолчанию, можно использовать поля строк Country, State/Region, City, которые определены в запросе RowFields.

Вот модифицированный код с отступами:

= Table.SplitColumn(

#"Отменить свертывание других столбцов ", "Attribute",

Splitter.SplitTextByDelimiter(":", QuoteStyle.Csv), RowFields

)

Часть 4. Переименование столбцов и функция ListZip

Теперь можно обновить наиболее сложный шаг панели Примененные шаги: Переименованные столбцы.

14. На панели Примененные шаги (Applied Steps) выберите шаг Переименованный столбец (Renamed Column). При просмотре строки формул можно видеть следующий код:

= ТаЬ1е.КепатеСо1ишп5(#"Разделить столбец по разделителю ", {{"::Color", "Color"}, {"::Parent Category", " Parent Category"}, "Country:State/Region:City \ Category",

"Category"}, {"Атрибут.1", "Country"}, {"Атрибут.2", "State/Region"},

{"Атрибут.3", "City"}, {"Value", "Revenue"}})

Этот код переименовывает столбцы, передавая список списков в качестве второго аргумента функции Table.RenameColumns. Каждый внутренний список представляет пару текстовых значений: первый текст — это имя существующего столбца, а второй — имя нового столбца.

15. Посмотрите на панель предварительного просмотра, обратите внимание на то, что на шаге 13 уже правильно переименованы поля строк и столбец Значение. Теперь нужно переименовать первые M столбцов. Для получения списка названий первых M столбцов выполните следующий код:

List.FirstN(

Table.ColuшnNames(#"Разgелить столбец по разделителю"),

List.Count(ColumnFields)

)

Для получения новых названий можно просто выполнить запрос ColumnFields.

Каким образом создать список списков, в котором каждый элемент из первого имени столбца M связан с полями столбцов? Для этого можно применить функцию языка M List.Zip, которая получает два списка в качестве входных данных и возвращает один список списков с парами из обоих списков. Например, следующие входные данные:

List.Zip({"A","B","C"}, {"1", "2", "3"})

возвращают такой список:

{{"A","1"}, {"B", "2"}, {"C", "3"}}

Таким образом, применяя функцию List.Zip для получения названий первых

M столбцов таблицы для полей столбцов, имеем следующий код:

List.Zip(

{

List.FirstN(

Table.ColumnNames(#"Разделить столбец по разделителю"),

List.Count(ColumnFields)

),

ColumnFields

}

)

и получим желаемый список списков, который используется в оригинальной формуле:

{{"::Color", "Color"}, {"::Parent Category ", "Parent Category "}, {"Country:State/Region:Country \ Category", "Category"}

И наконец, далее приводится обновленный код для шага Переименованные столбцы:

= Table.RenameColumns(

#"Разделить столбец по разделителю",

List.Zip(

{

List.FirstN(

ТаЫе.Со1ишпЫап1е5(#"Разделить столбец по разделителю"),

List.Count(ColumnFields)

),

ColumnFields

}

)

)

16. Ранее в этом упражнении были удалены шаги Измененный тип. Теперь при-шло время явно изменить тип столбца Revenue (Доход) на Десятичное число (Decimal Number). Если выполнить эту операцию с помощью пользовательского интерфейса, то базовый код в строке формул получит следующий вид:

= Table.TransformColurnnTypes(#"nepeuMeHoeaHHbie столбцы", {{"Revenue", type number}})

Измените жестко закодированное значение "Revenue" на ValueField:

= Tab1e.TransforшCo1uшnTypes(#"Переименованные столбцы ", {{ValueField, type number}})

Пришло время отсоединить обобщенную последовательность отмены свертывания столбцов от ее источника. Можно заметить, что первые два шага зависят от конкретной рабочей книги и рабочего листа. Также можно обобщить эти шаги и запустить последовательность отмены свертывания столбцов из любого источника данных.

17. На панели Примененные шаги (Applied Steps) щелкните правой кнопкой мы-ши на шаге Заполнено вниз и выберите команду Извлечь предыдущий (Extract Previous). В окне диалога Извлечь шаги (Extract Steps) выберите команду Извлечение шагов до выбранных шагов в новый запрос - Имя нового запроса (Source as Extract Query Name), а затем закройте окно диалога. Создается новый запрос, который загрузит исходную таблицу из файла Excel. Обобщенный запрос запустится как ссылка на новый запрос.

Часть 5. Преобразование запроса в функцию

Итак, мы подобрались к финальной стадии — преобразованию запроса в функцию. В главе 4 содержится краткое введение в вопросы, связанные с применением функций. В Power Query можно создавать параметры, ссылаться на них с помощью строки формул, а затем преобразовывать запрос в функцию с помощью команды Создать функцию (Create Function) на панели Запросы (Queries).

Позже в этой книге вы больше узнаете о функциях и попрактикуетесь в новых примерах по созданию функций из запросов с использованием команды пользовательского интерфейса Создать функцию (Create Function). В этом же примере будет показано, как создать функцию с помощью расширенного редактора, поскольку окно диалога Создать функцию (Create Function) не слишком успешно может применяться для работы со списками.

18. Щелкните правой кнопкой мыши на запросе Revenues, отображенном на панели Запросы (Queries), и выберите команду Расширенный редактор (Advanced Editor). Отобразившийся на экране код уже включает в себя уточненную и обобщенную версию последовательности для отмены свертывания столбов, которая уже реализована.

19. Для преобразования последовательности, применяемой для отмены свертывания столбов в функцию, дополните следующую строку как первую строку в окне расширенного редактора (над строкой let):

(Source, RowFields, ColumnFields, ValueField)=>

20. Удалите следующую строку:

Source = Source,

21. Щелкните на кнопке Готово (Done) для закрытия окна диалога расширенного редактора и переименуйте запрос Revenues на FnUnpivotSummarizedTable.

Обратите внимание, что пункт FnUnpivotSummarizedTable в панели Запросы (рис. 7.4) отмечен значком fx. После выбора этой функции предварительный просмотр данных больше не отображается. Вместо этого на панели предварительного просмотра отображается панель Ввод параметров (Enter Parameters), которую можно использовать для вызова функции. Однако в данном случае это не будет нас устраивать, поскольку нельзя заполнить функцию списками. Обратите внимание, что в разделе Примененные шаги (Applied Steps) весь набор шагов сведен к одному шагу и нельзя изменять функцию средствами пользовательского интерфейса.

Рис. 7.4. Если функция создается с помощью расширенного редактора, а не с помощью команды пользовательского интерфейса Создать функцию, то вне расширенного редактора изменить функцию невозможно. Как только запрос преобразован в функцию, на панели Примененные шаги появится лишь один шаг и его нельзя изменить

СОВЕТ

В отличие от функций, которые создаются с помощью средства Создать функцию (Create Function) в пользовательском интерфейсе, который включает редактируемый пример запроса, функцию, созданную непосредственно в расширенном редакторе (как показано в шагах 18-21), в дальнейшем нельзя редактировать вне среды расширенного редактора, по крайней мере непосредственно. Однако можно продублировать функцию, а в дублированном запросе можно удалить первую строку или закомментировать ее, добавив символы // в начале строки. Таким образом, удается преобразовать дублированную функцию в редактируемый запрос. После этого можно просмотреть шаги на панели Примененные шаги (Applied Steps) и внести изменения средствами пользовательского интерфейса. Завершив изменение нового запроса, можно скопировать код из расширенного редактора в исходную функцию и модифицировать функцию в соответствии с изменениями пользовательского интерфейса, внесенными в новый запрос.

Рассмотрим, чего мы достигли. На шаге 19 добавлена строка, преобразующая запрос в функцию.

Функция может быть написана как отдельный запрос, как это выполнено в этом случае, или как внутренний элемент внутри запроса. При вызове с аргументами создается новое значение. Можно вызвать функцию, перечислив параметры функции в формате "через запятую" в круглых скобках, за именем функции.

Вот аргументы, которые объявлены в функции:

■ Source — применяется таблица исходных данных;

■ RowFields СПИСОК ПОЛеЙ строк, В данном случае {"City", "State/Region", "City"};

■ ColumnFields список Полей столбцов, В данном случае {"Color", "Parent Category ",

"Category"};

■ ValueField — имя столбца для применения к столбцу Value, где выполнена отме-на свертывания столбцов, в данном случае "Revenue".

Символ "следования" => служит разделом между объявлением интерфейса функции и реализацией функции. Уже завершена часть реализации функции, когда обобщен каждый из соответствующих шагов с помощью параметров Source, RowFields, ColumnFields и ValueField.

Нужно отметить, что заголовок функции также может быть задан с явными типами, которые требуются в качестве входных аргументов, и ожидаемым типом вывода, используя элемент as, за которым следует конкретный тип. Вот пример строго типизированного заголовка функции:

(Source as table, RowFields as list, ColumnFields as list, ValueField as text) as table =>

Этот заголовок можно использовать вместо строки на шаге 19. Наличие объяв-ленных типов может гарантировать, что при вызове функции с неправильными типами возвращается ошибка. Например, неправильная передача текста "Country, State/Region, City" вместо списка в аргументе RowFields приведет к следующей ошибке:

Expression.Error: We cannot convert the value "Country, State/Region, City" to type List. Details: Value=Country, State/Region, City Type=Type

Часть 6. Тестирование функции

Теперь можно использовать функцию для отмены свертывания столбцов в любой обобщенной таблице, как это выполнено в упражнении 7.2. Для проверки подобного подхода при обработке исходной таблицы из упражнения 7.1 выполните следующие действия:

22. Щелкните правой кнопкой мыши на запросе Source (который сгенерирован на шаге 17) и выберите команду Ссылка (Reference). Переименуйте новый запрос на Results.

23. В строке формул измените следующий код:

= Source

на

= FnUnpivotSummarizedTable(Source, RowFields, ColumnFields, ValueField)

На этом этапе, если необходимо, также можно удалить запросы RowFields, ColumnFields и ValueField и выполнить отмену свертывания столбцов сле-дующим образом:

= FnUnpivotSummarizedTable(Source, {"Country", "State/Region", "City"},

{"Color", "Parent Category ", "Category"}, "Revenue")

Интересно, что эта функция может работать даже со сводной таблицей 1x1, но применять ее в этом аспекте не очень практично.

Всегда полезно тестировать функции и запросы с использованием экстремальных случаев. Например, можно загрузить исходную таблицу доходов AdventureWorks, которая использовалась в упражнении 6.1 (файл C06E01.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads), и применить следующий код на последнем шаге (т. е. шаге #"Измененный тип"):

= FnUnpivotSumniarizeclTable(#"I43MeHeHHbiM тип", {"Account"}, {"Parent Category"}, "Revenue")

Как это работает? Если поля строк и столбцов содержат единственный элемент, то операция заполнения вниз не приводит к ошибке для пустого списка столбцов, а просто возвращает входную таблицу. Кроме того, объединенные столбцы (Merge Columns) также устойчивы к граничным случаям и могут применяться к одному столбцу, возвращая в результате входную таблицу. Может быть, и не столь важно применять функцию FnUnpivotSummarizecTable к обобщенным таблицам 1x1, поскольку целесообразнее просто выполнить преобразование отмены свертывания столбцов, неплохо иметь уверенность в том, что эта функция надежна для обработки подобных крайних случаев.

Чтобы познакомиться с решением данного упражнения, загрузите файлы C07E03 - Solution.xlsx и C07E03 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/ downloads.

Выводы

В данном разделе показано, как обобщать последовательность преобразований отмены свертывания столбцов для обработки сложных обобщенных таблиц с любым количеством уровней иерархии в строках и столбцах. Также показано, как можно формировать функции с помощью расширенного редактора. При уточнении в запросе каждого шага применялись функции Table.ColumnNames, List.FirstN и List.Zip, тем самым мы узнали о потенциальных возможностях по совершенствованию запросов в будущем, в других сценариях, научились масштабировать отчет для поддержки автоматического импорта и поддержки преобразований различных исходных таблиц. В главе 9 приводится дополнительная информация о пользовательских функциях, глава 10 посвящена изучению сценариев, когда облегченное редактирование M-формул помогает противостоять экстремальным изменениям в формате данных ваших запросов.

Преобразование сведения в столбец

В этой и предыдущей главах вы познакомились с различными сценариями по применению преобразования отмены свертывания столбцов, узнали, каким образом данное преобразование помогает трансформировать обобщенные таблицы в таблицы фактов. Но иногда исходная таблица структурирована в слишком "плоском" формате, как будто кто-то злоупотребляет преобразованием отмены свертывания столбцов и формирует исходную таблицу с несколькими строками, по идее принадлежащими одному и тому же объекту. В данном разделе рассматриваются подобные сценарии, показано, как применяется преобразование, обратное к отмене свертывания столбцов: сведение в столбец.

УПРАЖНЕНИЕ 7.4. Обращение таблицы

с некорректной отменой свертывания столбцов

В данном упражнении будет показано, как можно с помощью единственного преобразования отмены свертывания столбцов исказить данные, а также продемонстрирован обратный путь для редактора Power Query, приводящий к устранению подобных изменений (учитывается, что данные на исходном уровне остаются без изменений).

1. Загрузите файл C07E04.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C07\.

2. Откройте новую пустую книгу в Excel или новый отчет в Power BI Desktop.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: в раскрывающемся меню Получить данные (Get Data) выберите команду Excel.

3. Выберите файл C07E04.xlsx и щелкните на кнопке Импорт (Import).

4. В диалоговом окне Навигатор (Navigator) выберите вкладку Revenues (Доходы), а затем щелкните на кнопке Изменить (Edit).

5. В открывшемся окне редактора Power Query, в панели предварительного про-смотра Preview, посмотрите на таблицу (изображенную вверху на рис. 7.5), которая содержит транзакции продаж вымышленной компании AdventureWorks.

Последние четыре столбца представляют четыре числовых атрибута для каждой транзакции продаж: столбец SubTotal представляет сумму продаж до вычета налогов и расходов на фрахт, TaxAmt — сумму налога с продаж, Freight — стоимость доставки, а TotalDue — общую сумму (TotalDue = SubTotal + TaxAmt + Freight).

Таблица продаж, проиллюстрированная в верхней части рис. 7.5, сформирована хорошо. Каждая транзакция продаж указана в отдельной строке, что позволит выполнять простые вычисления на уровне строк, например рассчитывать процент налога.

В процессе выполнения следующих шагов исказим структуру этой таблицы, чтобы продемонстрировать влияние некорректного применения преобразования отмены свертывания столбцов.

И Country Q SalesOrderlD Q OrderDate Q SubTotal П TaxAmt П Freight Q TotalDue pl|

| United States 71899 6/1/2008 2415.6727 193.2538 60.3918 2669.3183

United States 71915 6/1/2008 2137.231 170.9785 53.4308 2361.6403

United States 71895 6/1/2008 246.7392 19.7391 6.1685 272.6468

United Kingdom 71831 6/1/2008 2016.3408 161.3073 50.4085 2228.0566

Отмена свертывания столбцов (некорректная)

| Country Q SalesOrderlD H OrderDate Q Attribute Q Value ™

United States 71899 6/1/2008 SubTotal 2415.6727

United States 71899 6/1/2008 TaxAmt 193.2538

United States 71899 6/1/2008 Freight 60.3918

United States 71899 6/1/2008 TotalDue 2669.3183

United States 71915 6/1/2008 SubTotal 2137.231

United States 71915 6/1/2008 TaxAmt 170.9785

United States 71915 6/1/2008 Freight 53.4308

United States 71915 6/1/2008 TotalDue 2361.6403

United States 71895 6/1/2008 SubTotal 246.7392

United States 71895 6/1/2008 TaxAmt 19.7391

United States 71895 6/1/2008 Freight 6.1685

United States 71895 6/1/2008 TotalDue 272.6468

United Kingdom 71831 6/1/2008 SubTotal 2016.3408

United Kingdom 71831 6/1/2008 TaxAmt 161.3073

United Kingdom 71831 6/1/2008 Freight 50.4085

United Kingdom 71831 6/1/2008 TotalDue 2228.0566

Рис. 7.5. Здесь показано, как исправить данные, искаженные в процессе некорректного применения

преобразования отмены свертывания столбцов. В этом примере отмена свертывания

четырех числовых столбцов приводит к новой таблице с четырьмя строками для каждого заказа клиента

6. Выберите четыре последних столбца, а затем на вкладке Преобразование (Transform) выберите команду Отменить свертывание столбцов (Unpivot Columns). Как показано в нижней части рис. 7.5, таблица с отменой сворачивания столбцов сложнее для анализа, чем исходная таблица, поскольку контекст строки теперь содержит только один из четырех числовых атрибутов.

Представьте, что эта таблица и служит исходной, и нужно реверсировать отмену свертывания столбцов. Далее показано, как выполняется эта операция.

7. Выберите столбец Атрибут (Attribute) и на вкладке Преобразование (Transform) щелкните на кнопке Столбец сведения (Pivot Column).

8. После открытия окна диалога Столбец сведения (Pivot Column) внесите сле-дующие изменения:

а. Убедитесь, что в окне диалога Столбец сведения (Pivot Column) указано: "Использовать имена в столбце "Attribute", чтобы создать новые столбцы". На этом этапе нельзя изменить выбор столбца сведения. Таким образом, если в тексте упоминается другой столбец, это означает, что на шаге 7 был выбран некорректный столбец: щелкните на кнопке Отмена (Cancel) в окне диалога Столбец сведения и выберите требуемый столбец.

б. В списке Столбец значений (Values Column) выберите Значение (Value).

в. Раскройте раздел Расширенные параметры (Advanced) и для параметра Функция агрегирования (Aggregate Value Function) выберите значение Не агрегировать (Don't Aggregate), как показано на рис. 7.6.

г. Щелкните мышью на кнопке OK для закрытия этого окна диалога.

Как видите, прежняя структура таблицы была восстановлена.

Для просмотра готового решения загрузите файлы C07E04 - Solution.xlsx и C07E04 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

Рис. 7.6. Диалоговое окно Столбец сведения с выбранным параметром Не агрегировать позволяет преобразовывать таблицы, для которых выбрана отмена свертывания столбцов

УПРАЖНЕНИЕ 7.5. Применение преобразования сведения в столбец к таблицам с многострочными записями

В упражнении 7.5 преобразование отмены свертывания столбцов применялось к числовым столбцам. Столбцы привязки (см. главу 6) помогают поддерживать контекст каждой транзакции в процессе операции сведения в столбец. В этом упражнении показано, что произойдет, если преобразование отмены свертывания столбцов было применено ко всей таблице, и как разрешить это затруднение.

1. Повторите шаги 1-4 из упражнения 7.4 для пустой книги или нового отчета Power BI.

2. Выберите первый столбец и затем нажмите комбинацию клавиш <Ctrl>+<A> для выбора всех столбцов.

3. На вкладке Преобразование (Transform) выберите команду Отменить сверты-вание столбцов (Unpivot Columns).

На панели предварительного просмотра можно просмотреть таблицу, для которой было отменено свертывание столбцов и которая включает только два столбца — Атрибут (Attribute) и Значение (Value). Применение преобразования сведения в столбец на этом этапе не приведет к цели, поскольку утеряны столбцы привязки.

Сохраните запрос на этом этапе. В ближайшее время вы к нему вернетесь.

Прежде чем научиться работать с парами "атрибут/значение", обговорим важную роль формата "атрибут/значение". Этот формат часто встречается, он особенно характерен для файлов журналов и неструктурированных данных, которые не отформатированы как JSON или XML.

Рассмотрим, почему этот формат столь распространен. Представьте, что нужно импортировать данные из системы, где большой набор данных сущностей может включать сотни различных атрибутов, настраиваемых пользователями системы. Немногие из этих атрибутов являются обязательными, все остальные — необязательны.

Экспорт подобного набора данных в текстовые файлы в обычном табличном формате неэффективен. В результате формируется разреженная таблица с сотнями столбцов и пустыми ячейками, поскольку большинство атрибутов необязательные или отсутствуют. Для того чтобы не экспортировать разреженные таблицы, которые просто "раздувают" файл десятками и даже сотнями символов-разделителей для пустых значений, разработчики соответствующей системы могут реализовать экспорт в виде последовательности многострочных записей пар "атрибут/зна- чение". Таким образом, можно найти этот формат довольно распространенным, особенно если нужно импортировать журналы или экспортированные текстовые файлы.

В следующей части упражнения будет показано, как обрабатывать многострочные записи пар "атрибут/значение". Начнем с простого решения, когда можно предположить, что каждая многострочная запись занимает фиксированное число строк.

Затем применим общий подход, когда каждая запись всегда начинается с одного и того же атрибута.

Часть 1. Работа с фиксированным числом атрибутов

и применение целочисленного деления

На шаге 3 данного упражнения выполнялась отмена свертывания всех столбцов. Таблица, в которой было отменено свертывание столбцов, содержит последова-тельность групп, причем каждая группа состоит из семи строк пар "атрибут/зна- чение". Для реверсирования свертывания столбцов таблицы нужно восстановить контекст каждой группы и найти способ применить для каждой группы уникальные номера перед выполнением операции сведения в столбец.

4. На панели Запросы (Queries) правой кнопки мыши щелкните на запросе Revenues и выберите команду Ссылка (Reference). Переименуйте новый запрос в Revenues - Fixed Number of Attributes.

5. На вкладке Добавить столбец (Add Column) щелкните на кнопке Столбец ин-декса (Index Column). Новый столбец содержит порядковый номер, начиная с нуля для первой строки.

На следующем шаге применяется целочисленное деление на 7 по порядковому номеру. Целочисленное деление — это деление, при котором отбрасывается остаток. Этот шаг позволит преобразовать текущий индекс из отдельного индекса строки в групповой индекс, поскольку каждая группа состоит из семи строк.

6. Выберите столбец индекса. На вкладке Преобразование (Transform), в разделе Столбец "Количество" (Number Column) выберите команды Стандартный (Standard) Целочисленное деление (Integer-Divide). В открывшемся окне диалога Целочисленное деление введите 7 в поле Значение (Value) и щелкните на кнопке OK для закрытия этого окна диалога. Можно заметить, что столбец индекса имеет нулевое значение для первой группы из семи строк, 1 — для следующей группы из семи строк, 2 — для третьей группы строк и т. д. Теперь можно переходить к шагу сведения в столбец.

ПРИМЕЧАНИЕ

На шаге 6 не добавляйте новый столбец с помощью целочисленного деления. Убедитесь, что целочисленное деление применяется к имеющемуся столбцу индекса.

7. Выберите столбец Attribute (Атрибут) и на вкладке Преобразование (Transform) выберите команду Столбец сведения (Pivot Column).

8. В открывшемся окне диалога Столбец сведения внесите следующие изменения:

а. В списке Столбец значения (Values Column) выберите пункт Значение (Value).

б. Раскройте раздел Расширенные параметры (Advanced).

в. Для параметра Функция агрегирования (Aggregate Value Function) выберите значение Не агрегировать (Don't Aggregate).

г. Щелкните мышью на кнопке OK для закрытия этого окна диалога.

9. Удалите столбец индекса. Теперь в таблице успешно сведены столбцы. Сохраните отчет для последующего перехода к более общей методике свертывания в столбец многострочных записей.

Можно просмотреть готовое решение в файлах C07E05 - Solution.xlsx и C07E05 - Solution.pbix, которые доступны для загрузки с сайта https://aka.ms/DataPwrBIPivot/ downloads. (Данное решение можно найти в окне Power Query в виде запроса, Revenues - Fixed Number of Attributes.)

Часть 2. Обработка нефиксированного числа атрибутов с помощью команд Условный столбец и Заполнение вниз

Иногда приходится работать с многострочными записями, которые включают необязательные атрибуты. В подобных ситуациях нельзя предположить, что имеется фиксированное число строк для каждой записи, как это сделано в части 1 упражнения.

Здесь описывается более общая методика, которая позволяет выполнять сведение столбцов таблицу. В данном случае необходимы следующие предположения: каждая запись будет всегда начинаться с одного и того же имени атрибута, и, в то время как остальные атрибуты могут быть необязательными, первый атрибут является обязательным. На рис. 7.7 показаны шаги, необходимые для преобразования многострочных записей в таблицу фактов.

Откройте редактор Power Query и выполняйте эту часть упражнения с сохраненным файлом из части 1.

10. На панели Запросы (Queries) правой кнопки мыши щелкните на запросе Revenues и в контекстном меню выберите параметр Ссылка. Переименуйте новый запрос в Revenues - Fixed First Attribute. Этот новый запрос будет ис-пользоваться для сведения в столбцы данных с учетом требования о том, что каждая новая многострочная запись будет начинаться с атрибута Country (Страна).

11. На вкладке Добавить столбец (Add Column) выберите команду Столбец индекса (Index Column).

Текущий индекс будет использоваться для представления каждой строки. Теперь же предстоит решить более сложную задачу: необходимо идентифицировать индексы строк с помощью атрибута Country в столбце Attribute (Атрибут). Для этой цели можно применить преобразования Условный столбец (Conditional Column) и Заполнение вниз (Fill Down), которые удобны для сохранения контекста. Они широко встречались в главе 5, когда нужно было сохранить контекст из заголовков таблиц.

12. На вкладке Добавить столбец (Add Column) щелкните мышью на кнопке Условный столбец (Conditional Column). В открывшемся окне диалога Доба-вить условный столбец (Add Conditional Column) внесите следующие изме-нения:

1. Добавить столбец индекса

2. Добавить условный столбец

I Attribute H Value

Country United States

SalesOrderiD 71899

OrderDate 6/1/2008

SubTotal 2415.6727

TaxAmt 193.2538

Freight 60.3918

TotalDue 2669.3183

Country United States

SalesOrderiD 71915

OrderDate 6/1/2003

SubTotal 2137.231

TaxAmt 170.9785

Freight 53.4308

TotalDue 2361.6403

Country United States

SalesOrderiD 71895

OrderDate 6/1/2008

SubTotal 246.7392

TaxAmt 19.7391

Freight 6.1685

TotalDue 272.6468,

Country United States 0

SalesOrderiD 71899 1

OrderDate 6/1/2003 2

SubTotal 2415.6727 3

TaxAmt 193.2538 4

Freight 60.3918 5

TotalDue 2669.3183 6

Country United States 7

SalesOrderiD 71915 8

OrderDate 6/1/2003 9

SubTotal 2137.231 10

TaxAmt 170.9785 11

Freight 53.4308 12

TotalDue 2361.6403 13

Country United States 14

SalesOrderiD 71895 15

OrderDate 6/1/2008 16

SubTotal 246.7392 17

TaxAmt 19.7391 18

Freight 6.1685 19

TotalDue 272.6468

| Attribute 3 Value Q Index

1 AttributeQ Value Д Index О Row ID В

Country | United Sta 0 0

SalesOrderlf 71899 1 null

OrderDate 6/1/2008 2 null

SubTotal 2415.673 3

TaxAmt 193.2538 4

Freight 60.3918 5

TotalDue 2569.318 6

1 Country 1 United Sta 7 — 7

SalesOrderIL 71915 8 null

OrderDate 6/1/2008 9 null

SubTotal 2137.231 10 •

TaxAmt 170.9785 11 ■

Freight 53.4308 12 •

TotalDue 2361.64 13

ГЖП United Sta 14 ^>14

SalesOrderlt 71895 15 null

OrderDate 6/1/2008 16 null

SubTotal 246.7392 17 •

TaxAmt 19.7391 18 •

Freight 6.1685 19 ■

TotalDue 272.6468 20 J

[

4. Сведение по столбцу Attribute

1 Attribute Q Value A ndex El 4ow ID В

Country United States 0

SalesOrderiD 71899 null

OrderDate 6/1/2008 I 2 null

SubTotal 2415.6727 I I3

TaxAmt 193.2538 1 / 4

Freight 60.3918 1 1 5

TotalDue 2669.3183 1 / 6

Country United States 1 / 7 7

SalesOrderiD 71915 1/ 8 null

OrderDate 6/1/2008 V 9 null

SubTotal 2137.231 A 10

TaxAmt 170.9785 l\ 11

Freight 53.4308 / I 12 \

TotalDue 2361.6403 / I 13 1

Country United States / I14 14

SalesOrderiD 71895 / I15 null

OrderDate 6/1/2008 / Г6 null

SubTotal 246.7392 / r

TaxAmt 19.7391 1 V

Freight 6.1685 \

TotalDue 272.6468 / 2P

1 Attribute P Value H Row ID E”!

Country United Sta 0

SalesOrderiD 71899 0

OrderDate 5/1/2008 0

SubTotal 2415.673 0

TaxAmt 193.2538 0

Freight 60.3918 0

TotalDue 2669.318 0

Country United Sta 7

SalesOrderiD 71915 7

OrderDate 5/1/2008 7

SubTotal 2137.231 7

TaxAmt 170.9785 7

Freight 53.4308 7

TotalDue 2361.64 7

Country United Sta 14

SalesOrderiD 71895 14

OrderDate 5/1/2008 14

SubTotal 246.7392 14

TaxAmt 19.7391 14

Freight 6.1685 14

TotalDue 272.6468 И

5. Удалить столбец Row ID

Rew ID Д CountryП SalesOrderiD^ OrderDateQ SubTotalpl TaxAmt Я Freight Д TotalDueB

у / 0 United Stat /у 7 United Stat / \14 United Stat

71899 6/1/2008 2415.6727 193.2538 60.3918 2669.3183

71915 6/1/2008 2137.231 170.9785 53.4308 2361.6403

71895 6/1/2008 246.7392 19.7391 6.1685 272.6468л

Рис. 7.7. Когда каждая многострочная запись начинается с одного и того же имени атрибута

(в данном случае "Country"), можно применить эту последовательность сведения в столбцы

для превращения исходной таблицы в хорошо спроектированную таблицу фактов

а. В поле Имя нового столбца (New Column Name) введите Row ID.

б. В списке Имя столбца (Column Name) выберите значение Атрибут (Attri-bute).

в. В качестве оператора выберите "равно" (Equals).

г. Введите Country в текстовое поле Значение (Value).

д. В списке Вывод (Output) выберите параметр Выберите столбец (Select a Column) и выберите столбец Индекс.

е. Введите null в поле В противном случае (Otherwise). Нужно именно значе-ние null для строк со значениями, не равными Country (Страна), что позво-лит подготовиться к следующему шагу: Заполнение вниз (Fill Down).

ж. Щелкните на кнопке OK для закрытия этого окна диалога.

13. Выберите столбец Row ID и на вкладке Преобразование (Transform) выберите команды Заполнить (Fill) Вниз (Down). Теперь можно удалить столбец ин-декса.

Идентификатор строки (Row ID) теперь содержит число 0 для первой группы, число 7 для второй группы, число 14 для третьей группы и т. д. Если применять эту методику для многострочных записей с необязательными атрибутами, числа в идентификаторе строки (Row ID) не будут кратны 7 или любому другому фиксированному числу. Неважно, какие значения представлены в идентификаторе строки. Важно лишь то, что для каждой записи или группы строк он состоит из уникальных значений (благодаря использованию команды Заполнить (Fill) Вниз (Down).

Теперь, когда восстановлен контекст каждой записи, можно перейти к завер-шению этапа сведения в столбец.

14. Выберите столбец Attribute (Атрибут) и на вкладке Преобразование (Transform) выберите команду Столбец сведения (Pivot Column).

15. После открытия окна диалога Столбец сведения (Pivot Column) внесите сле-дующие изменения:

а. В списке Столбец значения (Values Column) выберите Значение (Value).

б. Раскройте раздел Расширенные параметры (Advanced).

в. Для параметра Функция агрегирования (Aggregate Value Function) выбери-те значение Не агрегировать (Don't Aggregate).

г. Щелкните на кнопке OK для закрытия этого окна диалога.

16. Удалите столбец Row ID.

Можно просмотреть файлы решения C07E05 - Solution.xlsx и C07E05 - Solution.pbix, доступные для загрузки на сайте https://aka.ms/DataPwrBIPivot/downloads. Это решение можно найти в окне редактора Power Query, запрос Revenues - Fixed First Attribute.

Резюме

В данной главе завершено исследование обобщенных таблиц, по отношению к которым применена отмена свертывания столбцов. Показано, как можно выполнять отмену свертывания столбцов для сложных таблиц при наличии уровней иерархии 3^3, а также как усложнять шаги, заменяя жестко закодированные имена столбцов в формулах M-функциями Table.ColumnNames и List. Также стало известно, каким способом преобразовать обобщенный запрос в функцию, создана функция FnUnpivotSummarizedTable, которая успешно протестирована на нескольких обобщенных таблицах.

Во второй части главы речь шла об отмене шага отмены свертывания столбцов и применении сведения в столбец для улучшения структуры таблицы. Также показаны последствия отмены свертывания всех столбцов, рассмотрена причина, по которой многострочные записи пар "атрибут/значение" представляют общий формат данных, для которых следует выполнять сведение в столбец. Применяя команды Столбец индекса (Index Column), Условный столбец (Conditional Column) и Заполнить вниз (Fill Down), можно ознакомиться с необходимой последовательностью этапов выполнения преобразования сведения в столбец в подобных случаях. С помощью этой новой методики можно выполнять многие сценарии, обрабатывающие многострочные записи.

Глава 8

Совместная работа

Великие дела в бизнесе никогда не совершаются одним человеком. Они делаются командой.

- Стив Джобс

В ЭТОЙ ГЛАВЕ ВЫ...

■ Узнаете, каким образом можно применять параметры в Power Query для формирования путей к локальным источникам данных.

■ Научитесь применять шаблоны Power BI в сочетании с параметрами, что позволит нескольким соавторам импортировать данные из локальных файлов.

■ Познакомитесь с ограничениями параметров в Excel, а также с тем, как можно в качестве параметров применять таблицы или именованные диапазоны.

■ Ознакомитесь с тем, как устраняются ошибки брандмауэра и выполняется объединение запросов.

■ Научитесь подключаться к общедоступным файлам в OneDrive for Business и SharePoint.

■ Узнаете, как можно подключаться к определенным подпапкам в SharePoint и как изменять запросы, которые подключаются к локальным папкам для того, чтобы они импортировали данные из SharePoint.

■ Рассмотрите вопросы безопасности, связанные с совместным использованием рабочих книг и файлов отчетов Power BI.

Приобретая больше опыта в обработке отчетов, вы сталкиваетесь с новыми проблемами, когда отчеты создаются совместно с коллегами. И хотя одновременное редактирование запросов не поддерживается и невозможно сравнивать различные редакции отчетов, имеется большое число проблем, возникающих при предоставлении общего доступа к "исходному коду" отчета.

Как автор отчета, вы работаете в группе специалистов, имеющих общий доступ для записи к определенным рабочим книгам Excel или файлам отчетов Power BI. Можно часто вносить изменения в отчеты, исправляя ошибки или совершенствуя их. Случается, что обмен отчетами выполняется традиционным и простым способом — в виде файла. Цель этой главы — познакомить с некоторыми основными проблемами совместной работы и путями их решения. Данная глава поможет вам нала

дить сотрудничество с другими соавторами и избежать ряда ошибок. Также вы сможете улучшить написание повторно используемых запросов и узнаете, как воспринимать отчеты в новом контексте.

ПРИМЕЧАНИЕ

Эта глава не предназначена для обмена опытом между конечными пользователями отчетов. Подобная тема быстро развивается, и она сильно различается в Excel и Power BI. Excel более ограничен с точки зрения обмена в Интернете и запланирован-ных обновлений с помощью Power Query, а Power BI позволяет публиковать файлы \*.pbix в службе PowerBI.com, запускать запланированное обновление и даже исполь-зовать локальные шлюзы данных, обеспечивающие безопасное подключение к услуге локальных источников данных, что дает возможность потребителям получать актуальные отчеты. Служба Power BI предполагает различные возможности обмена данными (см. https://docs.microsoft.com/en-us/power-bi/service-share-dashboards), а также см. по вопросам сотрудничества и совместного использования сред (https:// docs.microsoft.com/en-us/power-bi/service-collaborate-power-bi-workspace).

Локальные файлы, параметры и шаблоны

Одна из наиболее распространенных проблем, с которыми сталкиваются при совместном написании отчета в Excel или Power BI, связана с ситуацией, когда источники данных являются локальными файлами Excel или текстовыми файлами. Можно проиллюстрировать эту проблему очень распространенной историей двух авторов отчета. Познакомьтесь с Алисой и Бобом.

Алиса и Боб, два вымышленных героя, рассматриваемые в этой главе, являются опытными пользователями Power Query. До недавнего времени они работали в разных группах и были там единственными экспертами Power Query, поэтому им не приходилось делиться своими запросами с другими. На прошлой неделе они объединили усилия в создании довольно сложного отчета, причем работу над ним начала Алиса.

Ошибка при доступе к локальным файлам

Можно загрузить рабочую книгу Алисы, C08E01 - Alice.xlsx, или отчет Power BI, C08E01 - Alice.pbix, с сайта https://aka.ms/DataPwrBIPivot/downloads. Источник данных C08E01.xlsx также доступен для загрузки на этом сайте.

Когда Боб получает отчет Алисы впервые, при попытке обновления он получает следующую ошибку:

DataSource.Error: Could not find a part of the path 'C:\Users\Alice\Documents\C08\C08E01.xlsx' (Невозможно найти часть пути.)

Для устранения этой проблемы Боб выполняет несколько операций, а потом сохраняет данные Алисы, C08E01.xlsx, в своей локальной папке, C:\Users\Bob\Documents \C08\C08E01.xlsx:

1. Он открывает рабочую книгу или отчет Power BI и запускает редактор Power Query.

2. Для каждого из трех запросов, Revenues, Categories и Colors, Боб выполняет следующие действия:

а. Выбирает шаг Source (Источник) на панели Примененные шаги (Applied Steps).

б. В строке формул заменяет путь в строке:

= Excel.Workbook(File.Contents("C:\Users\Alice\Documents\C08\C08E01.xlsx"), null, true)

на следующий:

= Excel.Workbook(File.Contents("C:\Users\Bob\Documents\C08\C08E01.xlsx"), null, true)

Обновленная строка содержит путь, который Боб применял для сохранения локального файла C08E01.xlsx.

Применение шага 2 к трем запросам приводит к исправлению ошибки обновления. После этого Боб может приступить к работе над отчетом. Несколько часов спустя Боб посылает отчет назад Алисе, которая обнаруживает, что отчет не обновляется. Следуя инструкциям Боба, направленным по электронной почте, она изменяет путь в трех запросах.

Поскольку Алиса и Боб продолжают обмениваться отчетами, со временем они постепенно увеличивают число запросов, которые импортируют данные из локальных файлов. В конце концов всякий раз, когда им нужно работать над запросами и обмениваться рабочей книгой или отчетом, они вынуждены менять наименование пути в десятках запросов при переключении с одного на другой. Все становится более трудоемким по мере "разбухания" команды и присоединения к ней еще двух разработчиков. И вот однажды они узнают, что имеется удобный способ для разрешения проблемы: применение параметра.

УПРАЖНЕНИЕ 8.1.

Применение параметра для имени пути

Power Query позволяет создавать параметры и включать их в имеющиеся М-фор- мулы. Используя данный подход, можно располагать несколькими запросами, которые ссылаются на один параметр. Также можно проконтролировать значение одного параметра и тем самым повлиять на все запросы. В нашей истории Боб и Алиса могут применить параметр для наименования пути, это избавит их от всех проблем.

Продемонстрируем теперь, как использовать параметр для пути в Excel и Power BI при решении проблемы, возникшей у Алисы и Боба.

1. Загрузите исходные данные C08E01.xlsx и рабочую книгу Алисы, C08E01 - Alice.xlsx, или отчет Power BI, C08E01 - Alice.pbix, с сайта https://aka.ms/ DataPwrBIPivot/downloads и сохраните их в папке C:\Data\C08.

2. Запустите редактор Power Query Editor следующим образом:

В Excel: откройте рабочую книгу Алисы, C08E01 - Alice.xlsx. На вкладке Данные (Data) выберите команды Получить данные (Get Data) Запустить редактор запросов (Launch Power Query Editor).

В Power BI Desktop: откройте файл C08E01 - Alice.pbix. На вкладке Главная страница (Home) выберите команду Изменить запросы (Edit Queries).

3. На вкладке Главная (Home) редактора Power Query выберите раскрывающееся меню Управление параметрами (Manage Parameters) и выберите опцию Создать параметр (New Parameter). В окне диалога Параметры (Parameters) выполните следующие действия (показаны на рис. 8.1):

а. Введите Path в поле Имя (Name).

б. Введите C:\Data\C08 в поле Текущее значение (Current Value).

в. Щелкните мышью на кнопке OK для закрытия окна диалога.

ПРИМЕЧАНИЕ

При выполнении упражнения 8.2 уточните этот параметр, что позволит улучшить взаимодействие между Алисой и Бобом и даст им возможность переключаться между различными предопределенными значениями пути. На данный момент это упражнение упрощает задачу, здесь задан простой параметр, который может принимать любое значение.

4. На панели Запросы (Queries) выбирайте по очереди запросы Revenues, Colors и Categories и для каждого запроса выполняйте следующие шаги:

а. На панели Примененные шаги (Applied Steps) выберите шаг Source (первый шаг).

б. В строке формул найдите следующую строку:

= Excel.Workbook(File.Contents("C:\Users\Alice\Documents\C08\C08E01.xlsx"), null, true)

С помощью параметра Path и знака амперсанда (&) обновите строку для включения Path & "\08E01.xlsx" вместо "C:\Users\Alice\Documents\C08\C08E01.xlsx".

Вот так выглядит обновленная строка:

= Excel.Workbook(File.Contents(Path & "\C08E01.xlsx"), null, true)

5. Чтобы удостовериться в том, что три обновленных запроса работают должным образом, выберите каждый запрос еще раз и убедитесь, что на панели предварительного просмотра нет ошибок.

Отныне Алиса, Боб и их команда могут внести одно изменение, что позволит поменять значение параметра Path, и они смогут легко обновлять свои отчеты.

СОВЕТ

Параметры полезны для многих сценариев. Использовать параметры можно в любом контексте, который требует разных входных данных от автора отчета. Одна из наиболее распространенных ситуаций заключается в необходимости переключения между источниками данных разработки, тестирования и производства. В упражнении 8.1 один и тот же пользователь может изменить значение пути, чтобы оно соответствовало либо папке Development, либо папке Production. Таким образом, преимущества применения параметров не ограничиваются совместным редактированием запросов,

они могут помочь в контроле и управлении жизненным циклом отчетов.

Можно загрузить файлы решения C08E01 - Solution.xlsx и C08E01 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

УПРАЖНЕНИЕ 8.2. Создание шаблона Power BI

Параметры Power Query имеют определенные преимущества при их использовании в Power BI Desktop. В данном упражнении рассказано о шаблоне Power BI и о том, каким образом можно его применять в сочетании с параметрами для более успешного совместного создания и повторного использования отчетов.

Шаблон Power BI — это специальный формат файла для отчета Power BI. Шаблон Power BI можно экспортировать из любого имеющегося отчета. Экспортированный файл сохраняется как файл с расширением pbit, в него войдут данные из исходного отчета. Файлом шаблона можно делиться с другими соавторами или клиентами. При открытии шаблона, если в исходный отчет включены параметры, будет предложено указать их значения. После установки параметров отчет загружается на основе заданных во время загрузки значений.

В данном упражнении формируется шаблон отчета Power BI, который был создан в Power Query в упражнении 8.1. Можно загрузить файл C08E01 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads, если упражнение 8.1 не выполнялось с помощью Power BI Desktop.

1. Откройте файл отчета Power BI C08E01 - Solution.pbix.

2. На вкладке Главная (Home) выполните последовательность команд Файл (File) Экспорт (Export) Шаблон Power BI (Power BI template).

3. В открывшемся окне диалога Экспорт шаблона (Export a Template) введите описание шаблона. Этот текст отображается для любого пользователя, открывающего шаблон. Вот возможное описание:

Please provide the path of C08E01.xlsx file. For example: C:\Data\C08 (Пожалуйста, укажите путь к файлу C08E01.xlsx file. Например: C:\Data\C08)

4. Щелкните мышью на кнопке OK для закрытия данного окна диалога.

5. В открывшемся окне диалога Сохранить как (Save As) установите местополо-жение для сохранения файла нового шаблона как C08E02 - Solution.pbit и выберите команду Сохранить (Save).

6. Поскольку Power BI Desktop еще открыт для отчета \*.pbix, а не для файла шаблона, откройте окно Проводника и найдите новый файл C08E02 - Solution.pbit в месте, указанном вами на шаге 5. Можно поделиться этим шаблоном с другими соавторами, тогда им не нужно будет изменять путь для редактора Power Query, как это происходило с исходным решением из упражнения 8.1.

Как показано на рис. 8.2, если соавтор открывает шаблон, появляется окно диалога C08E02 - Solution. После ввода пользователем пути и выбора команды Загрузить (Load) отчет загружается, а запросы импортируют данные из локальной папки.

Рис. 8.2. При открытии файла шаблона Power BI (\*.pbit) пользователь может предоставить данные для всех использованных параметров

Для улучшения взаимодействия между Алисой и Бобом, поскольку уже известно, какие пути заданы у каждого из них, можно продолжить это упражнение, выполнив следующие шаги.

7. При открытом в окне файле C08E01 - Solution.pbix, на вкладке Главная (Home), выберите команду Изменить запросы (Edit Queries) для возвращения к редактору Power Query.

8. На вкладке Главная (Home) выберите команду Управление параметрами (Manage Parameters). В открывшемся окне диалога Параметры (Parameters) (рис. 8.3), в раскрывающемся меню Предлагаемые значения (Suggested Values), выберите команду Список значений (List of Values).

9. В таблицу, находящуюся под заголовком Список значений (List of Values), введите следующие значения (значения пути):

• C:\Data\C08

• C:\Users\Alice\Documents\C08

• C:\Users\Bob\Documents\C08

10. В раскрывающемся меню Значение по умолчанию (Default Value) выберите опцию C:\Users\Alice\Documents\C08, которая будет служить заданным по умолчанию именем пути (поскольку Алиса была первой, кто работал над этим отчетом, в ее честь будет назван путь, заданный по умолчанию). Однако выбрать любое из трех значений пути, введенных на шаге 9, нельзя.

11. В раскрывающемся списке Текущее значение (Current Value) выберите пункт C:\Data\C08, поскольку именно там хранится локальный источник данных.

х

Параметры

Создать Имя

Авс Path X Path

Описание

Путь к файлу C08E02.xlsx

яЛ Требуется

Тип

Любой

Предлагаемые значения Список значений

1 C:\Data\C08

2 C:\ll sers\AI ice\Docu ments\C08

3 C:\Users\Bob\Documents\C08

\*

Значение по умолчанию

C:\ll sers\AI ice\Docu ments\C08

Текущее значение rC:\Data\C08

ОК Отмена

Рис. 8.3. Можно настроить значение параметра Path для включения списка предложенных значений

12. Для добавления к параметру описания введите следующий текст в поле Описание (Description): "Путь к файлу C08E02.xlsx". Если описание сформировано, то справа от метки Path (Путь) появится информационная иконка, если шаблон открыт, и когда пользователи наводят на него курсор, во всплывающей подсказке отображается предоставленное здесь описание.

13. Прежде чем закрыть окно диалога Параметры (Parameters), убедитесь, что оно сконфигурировано, как показано на рис. 8.3.

14. Повторите шаги 2-5 для формирования нового шаблона Power BI и назовите его C08E02 - Solution 2.pbit.

Теперь шаблон подготовлен для совместной работы Алисы и Боба. Когда один из них откроет шаблон, можно выбрать подходящий путь из раскрывающегося списка.

Готовые шаблоны решений C08E02 - Solution.pbit и C08E02 - Solution 2.pbit можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Создание шаблона Power BI — это эффективный способ повторного использования отчета в разных наборах данных с помощью одинаковой схемы. Для получения дополнительной информации о работе с шаблонами посетите сайт https:// datachant.com/2016/04/29/power-bi-templates/.

УПРАЖНЕНИЕ 8.3. Часть 1.

Применение параметров в Excel

В отличие от Power BI Desktop, в Excel нельзя создать окно диалога, куда соавторам предлагалось бы вводить значения параметров. Шаблоны Excel (\*.xltx) функционируют иным образом и не предполагают пользовательского ввода при загрузке. В результате, если необходимо поделиться книгой Excel, для редактирования которой нужен соавтор, пользователю потребуется запустить редактор Power Query и изменить параметр.

Для улучшения взаимодействия между пользователями ознакомьтесь с данным упражнением, где показано, каким образом задействовать ячейки в электронной таблице вместе с именованными диапазонами или таблицами, чтобы пользователи могли вводить значения параметров без запуска редактора Power Query. Однако, как далее будет показано, подобная методика имеет и серьезные ограничения. Тем не менее этот метод позволяет загружать таблицу параметров из внешнего источника, такого как электронная таблица или файл конфигурации, и располагать внешним централизованным источником для параметров запроса. Подобные параметры могут определять некие пороговые значения на этапах преобразования данных, переключаться между несколькими предварительно определенными источниками данных (такими, как тестовая и производственная базы данных), а также ограничивать число строк или диапазон дат, которые необходимо извлекать из базы данных.

В данном упражнении начнем с файла решения Excel, созданного в упражнении 8.1. Этот файл C08E01 - Solution.xlsx можно загрузить с сайта https://aka.ms/ DataPwrBIPivot/downloads.

1. Откройте рабочую книгу C08E01 - Solution.xlsx.

2. Щелкните на значке Новый лист (New Sheet) и переименуйте его в Start Here (Пуск).

3. В ячейку A1 введите Path (Путь).

4. В ячейку A2 введите C:\Data\C08.

5. Выберите ячейки A1 и A2 и на вкладке Вставка (Insert) выберите команду Таблица (Table). В открытом окне диалога Создание таблицы (Create Table) выберите переключатель Таблица с заголовками (My Table Has Headers) и щелкните на кнопке OK для закрытия этого окна диалога.

6. Выберите ячейку A1 или ячейку A2 и затем на вкладке Конструктор (Design) набора вкладок Работа с таблицами (Table Tools) введите Параметры в поле Имя таблицы (Table Name).

ПРИМЕЧАНИЕ

В качестве альтернативы шагам 5 и 6 можно выбрать ячейки A1 и A2 и создать именованный диапазон, введя Параметры в поле Имя (Name), которое находится слева от строки формул.

Как показано на рис. 8.4, на этом этапе путь располагается на месте первой ячейки новой таблицы и может легко изменяться соавторами книги без необходимости запуска Power Query и поиска соответствующего параметра на панели Запросы (Queries). Если поделиться такой рабочей книгой с соавторами или коллегами, которым необходимо обновить рабочую книгу, можно упростить их работу; кроме того, инструкции, которые, возможно, понадобятся им (для изменения пути), будут намного проще.

В Ъ ’ С\* ’ ’ С08Е03 - Soluti... ИтаЯ Е - □ X

Файл Вета Раз\*/ Фор| Дань Реце Вид Разр Спр< Коне Q Помощь]

Буфер обмена ” Шрифт S

Выравнивание н

Число Условное форматирование Форматировать как таблиц Стили ячеек ’

Стили

Parameters ’ : I X ✓ А C:\Data\C08

СОВЕТ

Если необходимо передать несколько параметров, можно добавить столбцы, каждый с соответствующим именем параметра в строке 1 и его значениями в строке 2. Затем, используя преобразование детализации, которое описано в шаге 10, можно получить доступ к значениям параметров в редакторе Power Query.

7. После подготовки таблицы Parameters для направления в запросы значения параметра Path (Путь) выберите какую-либо ячейку (A1 или A2 в данном случае) в таблице Parameters (Параметры) и на вкладке Данные (Data) выберите команду Из таблицы/диапазона (From Table/Range).

8. После открытия редактора Power Query расширьте панель Запросы (Queries), где можно заметить новый запрос Параметры (Parameters). На панели про-смотра теперь есть таблица со столбцом Path (Путь). В будущем, если понадобятся дополнительные параметры, можно добавить их в качестве новых столбцов таблицы Parameters (Параметры) в электронной таблице. По этой причине можно сохранить параметры в виде отдельного запроса и создать новую ссылку для этого пути.

9. На панели Запросы (Запросы) щелкните правой кнопкой мыши на опции Parameters (Параметры) и выберите команду Ссылка (Reference). Переиме-нуйте новый запрос в Path2. (Пока еще присутствует старый параметр Path из упражнения 8.1, но скоро он будет удален.)

10. Выбрав запрос Path2 на панели Запросы (Queries), щелкните правой кнопкой мыши на его ячейке C:\Path\C08 и в контекстном меню выберите команду Детализация (Drill Down). Таблица преобразуется в текст со значением пути.

См. также

В главе 5 уже использовалось преобразование детализации для извлечения контекста из определенных ячеек и сохранения его в комбинированных таблицах. В данном упражнении применяется аналогичная методика для просмотра параметров.

На запрос Path2 можно ссылаться для получения значения пути из любого места в ваших запросах. Генерируемая функция языка M проста и удобна:

= Source{0}[Path]

Поскольку идентификатор Source (Источник) идентичен Parameters, можно избе-жать шагов 9-10, ссылаясь на следующий код всякий раз, когда необходимо получать доступ к значению Path:

= Parameters{0}[Path]

Если в будущем появятся новые параметры, можете просмотреть значение нового параметра, сославшись на имя его столбца вместо Path. Например, если имеется новый столбец для имени файла, в следующей строке можно получить доступ к новому параметру Filename:

= Parameters{0}[Filename]

В предыдущей формуле сначала выбрана первая строка, затем — столбец по имени. Для получения такого же результата можно использовать следующую

формулу, выбирая сначала столбец по имени и затем — первую ячейку (с помощью ссылки на нулевой индекс):

= Parameters[Filename]{0}

Пришло время обновить все запросы, что позволит загрузить путь из запроса Path2 вместо считывания его из собственного параметра запроса Path, который создан в упражнении 8.1. Параметр Path хорошо подходит для работы с Power BI Desktop при экспорте отчета в шаблон Power BI, а запрос Path2, который загружает данные из электронной таблицы, более удобен в Excel.

11. Для каждого из запросов Revenues, Colors и Categories измените в формуле на шаге Source (Источник) Path на Path2. Вот модифицированная формула:

= Excel.Workbook(File.Contents(Path2 & "\C08E01.xlsx"), null, true)

К сожалению, после этого шага можно получить следующую ошибку в панели предварительного просмотра для запросов Revenues, Colors и Categories:

Formula.Firewall: Query 'Colors' (step 'Source') references other queries or steps, so it may not directly access a data source. Please rebuild this data combination (Formula.Firewall: Query 'Colors' (step 'Source') ссылается на другие запросы или шаги, поэтому не может напрямую обращаться к источнику данных. Пожалуйста, восстановите эту комбинацию данных.)

12. Для устранения этой ошибки измените параметры конфиденциальности этой книги, выполнив следующие действия:

а. В редакторе Power Query выполните последовательность команд Файл Параметры и настройки Параметры запроса (File Query and Options Query Options).

б. В открывшемся окне диалога Параметры запроса (Query Options), в разделе Текущая книга (Current Workbook), выберите на левой панели раздел Конфиденциальность (Privacy).

в. Выберите поле Игнорировать уровни конфиденциальности для возмож-ного улучшения производительности (Ignore the Privacy Levels and Potentially Improve Performance) и щелкните на кнопке OK для закрытия этого окна диалога.

ПРИМЕЧАНИЕ

Игнорирование уровней конфиденциальности должно выполняться с некоторой осторожностью. Если вы не доверяете владельцу книги или видите, что запросы связаны с неизвестными внешними источниками, не рекомендуется применять параметры, выбранные на шаге 12. Однако в некоторых случаях Power Query не удается поддерживать изоляцию между несколькими источниками данных, тогда отображается ошибка Formula.Firewall. Во второй части данного упражнения показана усовершенствованная методика, которая позволит устранять ошибки без необходимости игнорировать уровни конфиденциальности.

13. Удалите параметр Path и переименуйте Path2 в Path. Переименование Path2 авто-матически изменяет все ссылки на новое имя запроса в запросах Revenues, Colours и Categories, поэтому не нужно повторять шаг 11 и изменять Path2 на Path.

14. На вкладке Главная (Home) выберите команду Закрыть и применить к (Close & Apply To).

15. При открытом в Excel окне диалога Импорт данных (Import Data) установите переключатель Создать только подключение (Create Connection Only), что гарантирует, что запрос Path не будет загружен в новую таблицу.

На этом этапе, пока используется отчет, параметр Path можно изменить в рабочей книге, затем следует обновить рабочую книгу для загрузки данных, и тогда не нужно запускать редактор Power Query для изменения параметра. Но если вы поделитесь рабочей книгой с коллегами, то при попытках обновления рабочей книги они столкнутся с той же ошибкой Formula.Firewall, которая имелась на шаге 11. В результате теперь нужно будет объяснять коллегам, как они могут отключить уровни конфиденциальности в книге (как это сделано на шаге 12). И хотя игнорирование уровней конфиденциальности также можно реализовать в Excel, не открывая редактор Power Query, выбирая команду Получить данные (Get Data) на вкладке Данные (Data), затем выбирая Параметры запроса (Query Options), все равно можно поделиться книгой решения из упражнения 8.1 и рассказать коллегам, каким образом изменяется параметр в окне редактора Power Query, не вдаваясь в деликатную тему конфиденциальности. Чтобы лучше использовать параметр Path в электронной таблице и избежать обсуждения вопросов конфиденциальности с коллегами, во второй части этого упражнения будет показано, как предотвратить появление ошибки Formula.Firewall.

Можно загрузить книгу с решением C08E03 - Solution.xlsx с сайта https:// aka.ms/DataPwrBIPivot/downloads. При ее открытии вы можете столкнуться с ошибкой Formula.Firewall (если только не установлено, что уровни конфиденциаль-ности игнорированы для всех рабочих книг). Для устранения этой ошибки выберите команду Получить данные (Get Data) на вкладке Данные (Data), затем выберите опцию Параметры запроса (Query Options) для выполнения шагов 12б и 12в.

УПРАЖНЕНИЕ 8.3. Часть 2.

Изменения в комбинациях данных

Ошибки Formula.Firewall можно избежать в целом, перестроив запросы таким обра-зом, чтобы все ссылки на внешние источники выполнялись в одном запросе. Всякий раз, когда вы сталкиваетесь с ошибкой, которая начинается Formula.Firewall, а завершается сообщением Please rebuild this data combination (Измените комбинацию данных), можно либо осторожно отключить уровни конфиденциальности, как описано на шаге 12 части 1 этого упражнения, либо объединить в один запрос все запросы, ссылающиеся на запросы с внешними источниками. Представление Зависимости запросов (Query Dependencies) — удобный момент для понимания подобной задачи.

Можно открыть файл рабочей книги C08E03 - Solution.xlsx, запустить редактор Power Query и на вкладке Просмотр (View) выбрать команду Зависимости запроса (Query Dependencies). Рассматривая зависимости (рис. 8.5), можно увидеть, что каждый из запросов в нижней части зависит как от текущей книги, так и от книги c:\data\c08\c08e01.xslx. Но если последняя рабочая книга ссылается непосредственно на запросы Revenues, Colors и Categories, то текущая книга не ссылается непосредственно на эти запросы, а реализует это посредством параметров Path (Путь) и Parameters (Параметры).

Объединяя в одном запросе ссылки на зависимые внешние источники, можно избежать ошибки. В этом примере консолидация произойдет, если каждый из за-просов Revenues, Colors и Categories непосредственно ссылается как на текущую рабочую книгу, так и на источник данных c:\data\c08\c08e01.xslx.

Чтобы реализовать описанную методику, выберите каждый из запросов, Revenues, Colors, и Categories и в расширенном редакторе добавьте следующую строку кода после слова let:

Path = Excel.CurrentWorkbook(){[Name="Parameters"]}[Content]{0}[Path],

Предыдущая строка объединит в одну строку все шаги преобразования запроса Parameters (Параметры). Теперь можно удалить запрос Parameters, поскольку на него не ссылаются ни в одном из запросов.

□ х

Зависимости запроса

Структура 14 + ►

Закрыть

Рис. 8.5. В окне диалога Зависимости запроса можно увидеть, что несколько запросов обращаются к внешним источникам данных Еще более понятный подход заключается в создании нового запроса, который возвращает двоичный файл рабочей книги, и тогда запросы Revenues, Colors и Categories ссылаются на запрос рабочей книги. Затем в запросе книги можно получить ссылки на внешний файл Excel и текущую книгу.

Далее приведены шаги, применяемые для реализации описанного подхода:

1. Откройте файл C08E03 - Solution.xlsx и на вкладке Данные (Data) выберите команду Получить данные (Get Data), затем выберите команду Запуск редактора Power Query (Launch Power Query Editor).

2. Удостоверьтесь, что для уровней конфиденциальности в окне Параметры за-проса (Query Options), в разделе Текущая книга (Current Workbook), выбран переключатель Совмещать данные в соответствии с настройками уровней производительности для каждого источника (Combine Privacy According to Your Privacy Level Settings for Each Source).

3. На панели Запросы (Queries) щелкните правой кнопкой мыши на запросе Colors и в контекстном меню выберите команду Дублировать (Duplicate).

4. Переименуйте новый запрос в Workbook и удалите все шаги на панели При-мененные шаги (Applied Steps), кроме первого шага Source.

5. На вкладке Главная (Home) выберите опцию Расширенный редактор (Advanced Editor) и добавьте следующий выделенный полужирным шрифтом код в выражение M:

let

Path =

Excel.CurrentWorkbook(){[Name="Parameters"]}[Content]{0}[Path],

Source = Excel.Workbook(File.Contents(Path & "\C08E01.xlsx"), null, true)

in

Source

6. Щелкните мышью на кнопке Готово (Done) для закрытия окна расширенного редактора. На этом этапе запрос Workbook возвращает аналогичное содержимое таблицы рабочей книги C08E01.xlsx, которое имелось перед шагом 4.

7. На панели Запросы (Queries) выберите каждый из запросов Revenues, Colors и Categories и для каждого из них выберите шаг Source на панели Примененные шаги (Applied Steps) и измените код в строке формул следующим образом:

= Workbook

8. На панели Запросы (Queries) удалите запрос Path и затем удалите запрос Parameters.

9. На вкладке Главная (Home) выберите команду Закрыть и применить к (Close & Apply To).

10. В открывшемся в Excel окне диалога Импорт данных (Import Data) установите переключатель Только создать подключение (Create Connection Only), что гарантирует, что новый запрос Workbook не будет загружаться как новая таблица.

Выполнив эти шаги, вы объединили все ссылки на внешние источники в запросе Workbook и избежали ошибки Formula.Firewall. Теперь можно просмотреть окно диалога Зависимости запроса (Query Dependencies), как показано на рис. 8.6, и увидеть, что запрос Workbook — единственный, имеющий доступ к двум внеш-ним источникам данных (в отличие от исходных, более сложных зависимостей, показанных на рис. 8.5).

Рис. 8.6. Окно диалога Зависимости запроса после выполнения консолидации, которая устраняет появление ошибки Formula.Firewall

Можно загрузить рабочую книгу решения C08E03 - Solution 2.xlsx с сайта https:// aka.ms/DataPwrBIPivot/downloads. Благодаря предпринятым мерам вашим коллегам не потребуется изменять или отключать собственные уровни конфиденциальности.

СОВЕТ

Изменение ваших запросов для предотвращения ошибок брандмауэра — не всегда простая задача. При объединении нескольких запросов в один теряется некоторая гибкость в плане поддержки и повторного использования запросов. Следует оценить все "за" и "против", прежде чем приступать к консолидации. Как показано в следующем разделе, иногда имеются лучшие решения для полного исключения необходимости обращения к двум источникам данных.

См. также

И хотя имеются предостережения, связанные с использованием в качестве внешних источников данных для Power Query локальных электронных таблиц Excel, можно публиковать отчеты Power BI в службе Power BI и наслаждаться запланированными обновлениями отчетов, даже при их подключении к локальным источникам данных. Служба Power BI может подключаться к локальным файлам (или к любому иному локальному источнику данных) с помощью локальных шлюзов данных. Дополнительные сведения по этому вопросу содержатся на сайте https://docs.microsoft.com/en-us/ power-bi/service-gateway-onprem.

Работа с общедоступными файлами и папками

В упражнениях 8.1, 8.2 и 8.3 применялись различные методы, упрощающие совместное редактирование при работе с локальными файлами. Алиса и Боб, наши вымышленные персонажи, теперь могут настроить путь и загрузить в общий отчет данные из локальной книги Excel.

Возможно, вы полагаете, что имеются и лучшие способы сотрудничества между авторами отчетов. Разве Алиса и Боб не могут воспользоваться общим файлом, размещенным на OneDrive for Business или на сайте SharePoint Online? Вы правы. Упражнения 8.1, 8.2 и 8.3 знакомят с важными приемами использования параметров, шаблонов Power BI, параметров рабочих листов и консолидации запросов, что позволит избежать ошибок обновления в распространенных сценариях, когда локальные файлы служат источниками данных, но применяется общий доступ, подобные файлы могут преодолеть многие препятствия при совместной работе команды.

В данном разделе приводятся советы и рекомендации по работе с общими файлами в OneDrive for Business (OneDrive для бизнеса) и SharePoint Online. Для выполнения упражнений, описанных далее, понадобится учетная запись OneDrive for Business и сайт SharePoint Office 365 (SharePoint for office 365), где имеется доступ к сохраненным файлам.

Импортирование данных из файлов

в OneDrive for Business или SharePoint

Импортирование данных из файлов источника, находящегося в OneDrive for Business или SharePoint, с помощью Power Query может вызвать затруднения у начинающих. В Power Query невозможен переход к OneDrive for Business или сайту SharePoint с помощью Проводника, как это выполняется для локальных файлов. Если отобразить OneDrive for Business или сайт SharePoint в виде локальной папки, то можно импортировать данные, как и для любого другого локального файла, но исходный файл не будет доступен всем остальным, поскольку путь окажется ло-

кальным и определенным для вашего компьютера. Чтобы обойти это ограничение, можно найти правильный URL-адрес в OneDrive for Business или SharePoint и воспользоваться веб-коннектором для импорта рабочей книги.

Несмотря на наличие нескольких различных способов получения URL-ссылки, которые позволят получать доступ к файлу, большинство из них приводит к URL- ссылке, которая не пригодна для Power Query. Для получения корректного URL- адреса и импортирования книги Excel в OneDrive выполните следующие действия.

1. Зарегистрируйтесь на OneDrive for Business или SharePoint и перейдите к папке, где находится рабочая книга.

2. Щелкните на рабочей книге, чтобы открыть ее в Excel Online.

3. Выберите команду Edit Workbook (Редактировать книгу), затем выберите команду Edit in Excel (Редактировать в Excel).

4. После открытия рабочей книги в Excel, на вкладке File (Файл), выберите команду Info (Сведения) и укажите путь.

Затем выберите команду Copy Path to Clipboard (Копировать путь в буфер обмена), как показано на рис. 8.7.

C08E01.xkx - Last Saved

Info

Рис. 8.7. С помощью команды Copy Path to Clipboard можно отыскать URL рабочей книги

для ее загрузки в Power Query

ПРИМЕЧАНИЕ

Существует множество способов получения URL-адресов для открытия книги в браузере, но большинство из них не будут работать, если попытаться применить их в Power Query для импорта книги. Благодаря методике Copy Path to Clipboard (Копи-ровать путь в буфер обмена) вы гарантированно получите корректную URL-ссылку, позволяющую Power Query импортировать рабочую книгу.

Вот пример URL-ссылки для OneDrive for Business:

https://contoso-my.sharepoint.com/personal/alice\_contoso\_com/Documents/

Book/Resources/Exercises/08/C08E01.xlsx

Этот вымышленный пример создан Алисой, работающей со своим OneDrive for Business. Элементы, выделенные в URL-ссылке полужирным шрифтом, появились в силу следующих причин: компания Алисы — contoso.com, а адрес электронной почты — alice@contoso.com (поэтому в данном пути присутствует элемент alice\_contoso\_com). Источник данных C08E01.xlsx хранился в папке Book, в подпапках Resources/Exercises/08.

А это пример URL-ссылки из SharePoint Online: https://contoso.sharepoint.com/sites/Book/Shared%20Documents/Resources/Exercises/ 08/C08E01.xlsx

В данном примере выделенные элементы в URL сгенерированы на основе следующих фактов: компания Алисы — contoso.com (следовательно, субдоменом в URL- ссылке является contoso). Источник данных C08E01.xlsx хранился на сайте SharePoint под названием Book, в папке Documents, в подпапках Resources/ Exercises/08.

5. Для импорта этой рабочей книги в Power Query выполните следующие действия:

В Excel: откройте пустую рабочую книгу. Затем, на вкладке Данные (Data) выберите команду Из Интернета (From Web). Заметим, что это не команда Вебзапрос (Web Query), которую можно выбрать в более старых версиях Excel, в разделе Получить и преобразовать данные (Get & Transform Data).

В Power BI Desktop: откройте пустой отчет. На вкладке Главная страница (Home) выберите раскрывающееся меню Получить данные (Get Data) и затем выберите команду Интернет (Web).

6. После открытия окна диалога Из Интернета (From Web) вставьте URL-ссылку из буфера обмена (скопирована на шаге 4) в поле URL-ссылки. Затем удалите суффикс "?web=1" и щелкните мышью на кнопке OK для закрытия этого окна диалога.

(Суффикс следует удалить, поскольку вы не заинтересованы в импорте книги в Excel Online.)

7. Если впервые подключаетесь к OneDrive for Business или SharePoint с помощью Power Query, открывается окно диалога Доступ к веб-содержимому (Access Web Content). На левой панели выберите раздел Учетная запись в организации (Organizational Account), затем щелкните на кнопке Вход (Sign In). После завершения фазы входа можно щелкнуть на кнопке Подключиться (Connect).

8. После открытия окна диалога Навигатор (Navigator) воспользуйтесь знакомым пользовательским интерфейсом для выбора соответствующих рабочих листов для импорта, затем щелкните на кнопке Изменить (Edit) для выполнения подготовки данных.

Г") СОВЕТ

Для применения Power Query в целях импорта текстового файла из OneDrive for Business или с сайта SharePoint можно сформировать временную рабочую книгу Excel в той же папке и следовать шагам 1-4 для получения пути. Затем продолжить и выполнить шаги 5-7 для импорта текстового файла.

УПРАЖНЕНИЕ 8.4. Миграция запросов для подключения

к OneDrive for Business или SharePoint

В данном упражнении вы измените запросы Алисы, подключившись не к ее ло-кальному компьютеру, а к папке OneDrive for Business или сайту SharePoint в Интернете. (Понадобится учетная запись One-Drive for Business или доступ к сайту SharePoint для вашей организации.)

1. Загрузите файл C08E01.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads и выгрузите его в OneDrive for Business или SharePoint.

2. Загрузите рабочую книгу Алисы, C08E01 - Alice.xlsx, или отчет Power BI, C08E01 - Alice.pbix, с сайта https://aka.ms/DataPwrBIPivot/downloads.

3. Откройте рабочую книгу C08E01 - Alice.xlsx или отчет Power BI C08E01 - Alice.pbix.

4. Проделайте шаги из предыдущего раздела для отслеживания URL-ссылки рабочей книги Excel C08E01.xlsx, которая была сохранена в OneDrive for Business или на сайте SharePoint на шаге 1.

5. Выполните шаги, описанные в предыдущем разделе, для импорта рабочей книги Excel C08E01.xlsx с помощью окна диалога Из Интернета (From Web). После открытия окна диалога Навигатор (Navigator) выберите таблицу Colors и затем щелкните на кнопке Изменить (Edit).

6. Переименуйте новый запрос на Workbook и, выделив его на панели Запросы (Queries), удалите последние два шага на панели Примененные шаги (Applied Steps), сохраняя только шаг Source (Источник).

7. Как и в упражнении 8.3, необходимо изменить запросы таким образом, чтобы они ссылались на запрос рабочей книги, а не на локальный файл в каждом запросе. Для этого выполните следующие действия для каждого из запросов Revenues, Colors и Categories:

а. На панели Примененные шаги (Applied Steps) выберите шаг Source.

б. В строке формул измените строку

= Excel.Workbook(File.Contents("C:\Users\Alice\Documents\C08\C08E01.xlsx"), null, true)

на следующую:

= Workbook

8. Для того чтобы гарантировать, что новый запрос Workbook не будет загружен, выполните такие шаги:

В Excel: на вкладке Главная (Home) редактора Power Query выберите команду Закрыть и применить к (Close & Apply To). После открытия окна диалога Импорт данных (Import Data) установите переключатель Создать только подключение (Create Connection Only), что гарантирует, что новый запрос Workbook не будет загружен как новая таблица.

В Power BI Desktop: правой кнопкой мыши щелкните на запросе Workbook на панели Запросы (Queries) в редакторе Power Query и отмените выбор переключателя Включить загрузку (Enable Load) в раскрывающемся меню.

9. Чтобы гарантировать корректность выполнения шагов, запустите редактор Power Query и на вкладке Просмотр (View) выберите команду Зависимости запроса (Query Dependencies). Убедитесь, что макет ваших запросов совпадает с макетом, показанным на рис. 8.8.

Рис. 8.8. Окно диалога Зависимости запроса показывает, каким образом все запросы импортируют рабочую книгу OneDrive for Business

Можно загрузить файлы решения C08E04 - Solution.xlsx и C08E04 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

Различия между OneDrive for Business

и SharePoint Online

Сервис OneDrive for Business позволяет быстро перейти от локального файла к общему файлу, если вы сотрудничаете с другими членами группы, но для сохранения общего источника данных все же рекомендуется сайт SharePoint. Сервис OneDrive for Business обеспечивает возможность создания личных папок, и если владелец папки покидает компанию, вы не сможете подключиться к этому источнику.

Если нужно подключиться к нескольким книгам Excel или текстовым файлам в папке, можно сохранять файлы на сайте SharePoint и обратиться к коннектору папок SharePoint. В упражнении 8.5 будет показано, как подключиться к папке SharePoint и перенести запросы из локальной папки в папку SharePoint.

УПРАЖНЕНИЕ 8.5. От локальной папки к общедоступной

Выполнение упражнения начнем с имеющейся таблицы Excel или отчета Power BI, который подключается к локальным источникам данных в папке. Затем выгрузим источники данных на сайт SharePoint и изменим запросы для импорта источников данных из SharePoint. Перемещение папки в SharePoint, как видно из рис. 8.9, не очень сложно. Но, как будет показано далее, модификация запросов не является тривиальной задачей.

Рис. 8.9. Перемещение локальной папки или источников данных в SharePoint может улучшить сотрудничество между авторами отчета

Перед началом выполнения упражнения загрузите следующие файлы с сайта https://aka.ms/DataPwrBIPivot/downloads:

■ C08E05.xlsx;

■ C08E05.pbix;

■ C08E05 - Folder.zip.

1. Разархивируйте файл C08E05 - Folder.zip в вашей локальной папке C:\Data\C08. После завершения этого процесса должна появиться подпапка с именем

C08E05 - Folder, которая включает три рабочие книги Excel, представляющие продукты компании AdventureWorks по годам выпуска. Можно просмотреть содержимое книг Excel в папке. Каждая книга имеет свой год выпуска продуктов AdventureWorks.

2. Скопируйте папку C08E05 - Folder во вложенную папку, находящуюся под папкой Documents на сайте SharePoint.

3. Откройте файл C08E05.xlsx в Excel или C08E05.pbix в Power BI Desktop и запустите редактор Power Query.

В Excel: на вкладке Данные (Data) выберите команду Получить данные (Get Data) и затем выберите команду Запустить редактор Power Query (Launch Power Query Editor).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Изменить запросы (Edit Queries).

4. На панели Запросы (Queries) выберите запрос C08E05 - Folder. На панели просмотра можно видеть продукты AdventureWorks. Эти запросы покажутся вам знакомыми. И это не случайно, поскольку в упражнении 3.3 и главе 3 использовался коннектор папки для добавления этих книг из папки.

5. Подключитесь к папке SharePoint и узнайте, в чем разница между локальной папкой и папкой SharePoint. Для этого в редакторе Power Query на вкладке Главная (Home) выберите команду Создать источник (New Source) и выпол-ните одно из следующих действий:

В Excel: выполните команды Файл (File) Папка SharePoint (SharePoint Folder).

В Power BI Desktop: в открывшемся окне диалога Получить данные (Get Data) выберите команду Папка SharePoint (SharePoint) и затем щелкните на кнопке Подключиться (Connect).

6. После открытия окна диалога Папка SharePoint (SharePoint Folder) в поле URL- адрес сайта (Site URL) введите корневую URL-ссылку на ваш сайт SharePoint, без всяких подпапок. Вот пример подобной URL-ссылки, если имя поддомена компании — contoso, а сайт SharePoint — Book:

https://contoso.sharepoint.com/sites/Book

ПРИМЕЧАНИЕ

Ключевым фактором при работе с папками SharePoint является тот факт, что вы не можете подключиться напрямую к определенной подпапке, но только к основному корню сайта. Для перехода к определенной папке необходимы фильтры по относи-тельному пути и именам файлов, что позволит ограничить число файлов до тех, которые необходимо добавить.

7. Если вы впервые получаете доступ к SharePoint с помощью Power Query, откроется окно диалога Доступ к веб-содержимому (Access Web Content). На левой панели выберите раздел Учетная запись организации (Organizational Account) и затем щелкните на кнопке Вход (Sign In). После выполнения фазы регистрации можно щелкнуть на кнопке Подключиться (Connect).

8. После открытия предварительного просмотра содержимого папки выберите команду Изменить (Edit).

9. Поскольку вы связались с корневой папкой сайта SharePoint, вероятно, в корневой папке на панели предварительного просмотра вы найдете много не относящихся к делу записей для всех файлов. Чтобы отфильтровать строки и сохранить только файлы C08E05 - Folder, выберите фильтр в заголовке столбца Folder Path (Путь к папке).

10. После открытия панели Фильтр (Filter) выполните команду Текстовые фильтры (Text Filters) и затем выберите опцию Содержит (Contains).

11. После открытия окна диалога Фильтры по строкам (Filter Rows) в текстовое поле, находящееся справа от раскрывающегося меню Содержит (Contains), введите текст /C08E05 - Folder/ и затем щелкните на кнопке OK для закрытия окна диалога.

Теперь в панели просмотра отображаются только три файла Excel.

12. Переименуйте запрос на SharePoint Folder. Теперь можно применить его в имеющихся запросах, поэтому на панели Запросы (Queries) выберите запрос Sample File.

13. На панели Примененные шаги (Applied Steps) выберите шаг Source (Источ-ник). В строке формул отобразится формула, возвращающая локальную папку:

= Folder.Files("C:\Data\C08\C08E05 - Folder")

Замените эту формулу следующей:

= #"SharePoint Folder"

14. Повторите шаг 13 после выбора запроса C08E05 - Folder на панели Запросы (Queries). Данный запрос также следует обновить для связи с папкой SharePoint вместо локальной папки.

15. Для гарантии того, что новый запрос SharePoint Folder не загрузится в отчет, выполните следующие действия:

В Excel: на вкладке Главная (Home) выберите команду Закрыть и применить к (Close & Apply To). В открывшемся окне Импорт данных (Import Data) установите переключатель Создать только подключение (Create Connection Only) для гарантии того, что новый запрос SharePoint Folder не будет загружаться как новая таблица.

В Power BI Desktop: Правой кнопкой мыши щелкните на запросе SharePoint Folder в панели Запросы (Queries) и отмените установку переключателя Включить загрузку (Enable Load) из раскрывающегося меню.

Теперь ваши запросы загружают книги из SharePoint, а не из локальных файлов. Алиса и Боб будут счастливы.

Можно загрузить файлы решения C08E05 - Solution.xlsx и C08E05 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

Вопросы безопасности

Power Query разработан с учетом мер безопасности. Если создается отчет в Power Query, который подключается к базе данных организации или любому другому набору данных, требующему аутентификации, ваши учетные данные хранятся на вашем компьютере, тогда можно обновлять отчет по требованию и не вводить учетные данные для каждого обновления.

Если файл отчета передается другим пользователям, учетные данные остаются на вашем компьютере. Ваши соавторы должны предоставить свои собственные учетные данные для подключения к соответствующему источнику данных. Этот подход защищает ваши учетные данные и гарантирует, что посторонний автор отчета не сможет получить доступ к вашим учетным данным.

В некоторых случаях можно поделиться запросами, моделью и визуализациями, но убедитесь, что данные в отчете очищены перед тем, как предоставить отчет другим пользователям. Это можно сделать с помощью шаблонов Excel или Power BI. В упражнении 8.2 создан шаблон Power BI, который позволял пользователю вводить значение параметра до загрузки данных. Шаблоны Power BI также можно создавать без параметров. Данные по-прежнему не будут загружаться в шаблон, можно поделиться ими с другими соавторами, не раскрывая ваших данных. Обычно это удобно, если каждый соавтор отчета может получить доступ только к подмножеству строк из тех же таблиц в источниках данных.

Excel работает аналогично. Для сохранения книги Excel в качестве шаблона и удаления данных из модели или таблиц выполните команды Файл Сохранить как (File Save As). Перед сохранением файла убедитесь, что тип файла изменен на Шаблон Excel (\*.xltx) (Excel Template (\*.xltx)). Перед сохранением файла отобразится следующее приглашение:

The Workbook contains external data. Do you want Microsoft Excel to clear the data before saving the template, and then automatically refresh the data whenever the template is opened? (Рабочая книга содержит внешние данные. Желаете ли вы, чтобы Microsoft Excel очистил данные перед сохранением шаблона, а затем автоматически обновлял данные при каждом открытии шаблона?)

Щелкните на кнопке Да (Yes) для удаления данных. Если ваш коллега откроет отчет, то Power Query запустит обновление и предложит пользователю предоставить свои учетные данные, если это необходимо. Ваши данные не будут сохраняться в отчете.

Удаление всех запросов в Excel

с помощью инспектора документов

В некоторых случаях может возникнуть необходимость поделиться преобразованными данными или статическими диаграммами в рабочих листах, но удалить запросы и модель из рабочей книги. И хотя можно вручную удалить каждый запрос и сохранить без изменений отдельные таблицы, имеется более быстрый способ. Для мгновенного удаления всех запросов в вашей книге можно воспользоваться Инспектором документов (Document Inspector). Для того чтобы очистить все запросы с помощью Инспектора документов, выполните следующие шаги:

1. Перейдите на вкладку Файл (File) в Excel и в разделе Сведения (Info) щелкните на кнопке Поиск проблем (Check for Issues). Затем выберите команду Инспектор документов (Inspect Document).

2. В открытом окне диалога Инспектор документов (Document Inspector) выберите опцию Inspect.

3. Выполните прокрутку вниз до тех пор, пока не перейдете к разделу Настраи-ваемые XML-данные (Custom XML Data), как показано на рис. 8.10.

ПРИМЕЧАНИЕ

Power Query сохраняет запросы в специальном месте в книге, известном как На-страиваемые XML-данные (Custom XML Data). Эта область в файле также использу-ется сторонними надстройками. Поэтому в качестве меры предосторожности всегда следует запускать Инспектор документов вместе с копией рабочей книги на тот слу-чай, если в файле хранятся дополнительные данные для надстроек, которые будут удалены Инспектором вместе с запросами.

4. Щелкните мышью на кнопке Удалить все (Remove All), затем щелкните на кнопке Закрыть (Close).

5. На вкладке Данные (Data) выберите команду Запросы и подключения (Queries and Connections). Теперь можно видеть, что все запросы удалены. Если вы загрузили запросы в модель данных, они также будут удалены.

Резюме

В этой главе вы столкнулись с новыми проблемами, которые часто встречаются, когда вы начинаете совместно создавать отчеты, делиться файлами с коллегами и работать в составе команды.

В упражнениях 8.1 и 8.2 показано, как использовать параметры и шаблоны для улучшения запросов при совместной работе с другими авторами отчетов и при импорте данных из локальных файлов. В упражнении 8.3 описано, как создать таблицу параметров на рабочем листе и загрузить ее в виде запроса. Затем применялась детализация для загрузки значения отдельного параметра и в процессе объединения конкретных запросов устранялись ошибки Formula.Firewall.

В упражнениях 8.4 и 8.5 было показано, как подключиться к OneDrive для бизнеса и SharePoint для импорта данных с помощью Power Query. Также рассмотрена реорганизация имеющихся запросов для загрузки данных из общих рабочих книг или папок вместо локального источника в вашей файловой системе.

Наконец, в этой главе раскрыты некоторые вопросы безопасности, которые позволяющие понять, как можно обмениваться рабочими книгами и настольными Power BI и как удалять отчеты из данных или удалять все следы Power Query в рабочей книге с помощью Инспектора документов. По мере приобретения опыта работы с Power Query настало время расширить свои познания в области языка M и улучшить собственное мастерство по обработке данных.

Глава 9

Введение в язык формул

M Power Query

Вы делаете их, а я забавляю их.

- Доктор Сьюз

В ЭТОЙ ГЛАВЕ ВЫ...

■ Ознакомитесь с этапами глубокого понимания возможностей языка M.

■ Познакомитесь с локальными и глобальными ресурсами, применяемыми для изучения языка M.

■ Изучите основные строительные блоки языка M, включая выражение let.

■ Узнаете об основных операторах и встроенных функциях для работы с типами данных языка M, включая число, время, дату, продолжительность, текст, нуль и логические выражения.

■ Разберетесь в методике формирования условий и выражений.

■ Узнаете, как определять и вызывать пользовательские функции.

■ Познакомитесь с методикой программной обработки ошибок.

■ Изучите методику выполнения итерации по циклу для реализации сложных преобразований, применения рекурсии и расширенных функций списка.

Язык формул M Power Query генерируется Power Query при внесении шагов преобразования в пользовательском интерфейсе Power Query Editor. Каждый шаг преобразования превращается в одну строку кода М (иногда даже в несколько строк кода), которую можно просмотреть и отредактировать в строке формул редактора Power Query Editor. Совместно эти шаги формируют выражение M, блок кода, который определяет шаги преобразования в запросе и может редактироваться в расширенном редакторе.

Несмотря на то что предусмотрена возможность создавать полные выражения M для определения запросов с нуля в расширенном редакторе, большинство проблем с данными можно решить с помощью пользовательского интерфейса, без необходимости создания выражения M или даже запуска редактора Power Query Editor. Тем не менее более глубокое понимание языка M позволяет решать сложные проблемы с данными и расширять сферу использования запросов для самых сложных наборов данных.

Если вы не желаете изучать новый язык программирования, все равно эта глава будет вам полезной. В отличие от типичных языков программирования, при обращении к возможностям M вы будете в состоянии достичь многого, особо не погружаясь в детали.

Изучаем возможности языка M

Теперь, после выполнения упражнений в предыдущих главах, вам будет легко при рассмотрении материала этой главы, вы узнаете больше о языке запросов M, используя непосредственный подход. В следующих разделах подробно описан синтаксис M, а в этом разделе основное внимание уделено изучению языка M в целом.

До сих пор возможности языка M были реализованы с помощью следующих трех пользовательских интерфейсов:

■ Строка формул — дает возможность изменить функцию M в строке формул для настройки шагов преобразований в соответствии с вашими потребностями. Обычно мы так поступали, если в пользовательском интерфейсе отсутствовала конкретная возможность (например, по применению расширенного логического условия при фильтрации столбца).

■ Окно диалога Настраиваемый столбец (Custom Column) — обеспечивает выполнение некоторых вычислений на уровне строк и сохранение результатов в новом столбце.

■ Расширенный редактор — позволяет изменить несколько шагов в запросе, включить пример кода из другого запроса или создать пользовательскую функцию для решения конкретной задачи.

Даже если вы не до конца разобрались с синтаксисом или с кодом, тот факт, что код генерируется с помощью пользовательского интерфейса, помогает в некоторой степени понять его, — и это важный элемент, который нужно помнить. Лучший способ изучить возможности языка M — применить шаги преобразования в пользовательском интерфейсе, а затем постепенно пытаться освоить базовый код.

Изучая "продвинутые" возможности языка М

Отправляясь в путешествие для изучения языка M, вы, вероятно, пройдете шесть этапов, которые проиллюстрированы на рис. 9.1. Каждый этап позволяет решать новые типы проблем с данными, но при этом он будет удерживать вас в определенной "зоне комфорта". Задайте себе вопрос: "Будете ли вы готовы приложить значительные усилия, чтобы перейти к следующему этапу?"

Этап 1. Только интерфейс пользователя

Если вы впервые познакомились с языком М именно в этой книге и до сих пор прилежно выполняли упражнения, значит, вы уже овладели начальным уровнем подготовки в изучении М и можете считаться опытным пользователем Power

Время

Рис. 9.1. Проходя через этапы зрелости в обучении языку М, вы сможете решать большее число проблем, причем большинство из них могут быть решены уже при достижении следующего этапа

Query. Как известно, язык M используется в качестве основного языка программирования, но у вас могут быть сомнения в необходимости изучать его, поскольку непонятно, стоит ли игра свеч. Возможно, вы также опасаетесь приступать к изучению языка программирования или же отдаете предпочтение другим языкам, которые отличаются от языка M. Имеется большое число языков, включая формулы Excel, VBA, DAX, Python и R, и вы можете освоить многие из них. Почему же следует изучать М, который значительно отличается от всего, с чем мы привыкли работать?

Катализатор для перехода к следующему этапу изучения этого языка заключается в понимании, что пользовательский интерфейс поможет устранить только около 40% ваших проблем. Если же задаться целью разрешить большее число проблем с данными с помощью Power Query, вдруг обнаруживается, что многие проблемы могут быть преодолены с помощью небольших изменений кода. Поскольку данная настройка не требует глубокого понимания синтаксиса, это естественный этап на пути изучения языка, даже если вы не очень глубоко разбираетесь в формулах Excel или мало осведомлены в вопросах программирования.

Этап 2. Простое редактирование в строке формул

На втором этапе зрелости в строке формул применяются небольшие модификации к М-коду. Можно не понимать формулы полностью, но уже связывать некоторые элементы в формуле с действиями, которые вы предприняли в пользовательском интерфейсе, и расширить его для удовлетворения своих потребностей. На этом эта

пе можно устранить около 60% проблем, с которыми вы сталкиваетесь, и существенно повлиять на возможности по обработке данных.

Желание перейти к следующему этапу приходит, когда появляется растущая по-требность в более сложных вычислениях в отчетах и когда вы становитесь достаточно опытны в других областях, таких как DAX в Power BI или Power Pivot в Excel. Когда вы узнаете о возможностях вычисляемых столбцов в DAX, то можете подумать, что проще реализовать решение с помощью настраиваемого столбца в языке M.

Этап 3. Формулы M в настраиваемых столбцах

На третьем этапе рассматривается методика применения настраиваемых столбцов, что позволит вам создавать простые, но достаточно эффективные M-формулы в поле Пользовательская формула столбца (Custom Column Formula) окна диалога Настраиваемый столбец (Custom Column). Вы узнаете об использовании логических условий в формулах, успешно примените в работе операторы if/then/else в сочетании с логическими операторами and, or и not. Окно диалога Настраиваемый столбец (Custom Column) удобнее, чем окно Условный столбец (Conditional Column), и позволит разрешать до 80% проблем, с которыми обычно сталкиваются пользователи.

И хотя некоторым пользователям вполне достаточно знаний, полученных на этом этапе, многие уже представляют спектр широких возможностей, предоставляемых М, приобретают уверенность в работе с языком M для решения новых задач. В этом случае появляется желание покинуть границы строки формул и приступить к изучению выражений M в расширенном редакторе, чтобы применять свои знания при создании формул M и расширять запросы для решения дополнительных задач.

Этап 4. Настраиваемые функции в окне расширенного редактора

На четвертом этапе осваивают формирование настраиваемых функций для повторного использования шагов преобразования. И хотя средство Создать функцию (Create Function) редактора Power Query трудно переоценить, можно без затруднений преобразовывать запросы в пользовательские функции с помощью расширенного редактора. Также можно применять расширенный редактор для масштабного изменения кода.

На этом этапе уже можно успешно разрешать до 95% проблем, с которыми обычно сталкивается пользователь. Ваша репутация агента по сбору данных уже выйдет за пределы команды, многие коллеги смогут обращаться к вам за помощью в решении их проблем с данными.

На этом этапе вы уже можете успешно решить практически любую проблему с данными. Основным катализатором для перехода на следующий уровень является ваша личная любознательность, а также желание досконально овладеть возможностями языка М. Находясь на этом этапе, можно приступить к формированию библиотеки полезных пользовательских функций и искать новые способы эффективного решения проблем и повторного использования запросов в сложных ситуациях.

Этап 5. Расширенные вычисления

(List.Accumulate, List Generate)

На пятом этапе можно переходить к сложным сценариям и создавать сложный М-код для их воплощения. Продолжая использовать Power Query Editor для создания кода, вы сами формируете существенные и сложные выражения кода в расширенном редакторе. Также вы сможете легко создавать вложенные выражения для блоков let/in и обобщать автоматически сгенерированный код для расширения его возможностей.

На этом этапе обнаруживается, что функции List.Accumulate и List.Generate чрезвы-чайно полезны для создания итераций преобразований. Эти функции пригодны для создания небольших фрагментов кода М, которые применяются как аргументы в других функциях М. Даже если число случаев, где это действительно нужно, невелико, тот факт, что вы создаете расширенные формулы M с помощью функций List.Accumulate и List.Generate, является свидетельством высокого уровня вашего мастерства при работе в языком M. Если вам известно, как задействовать эти функции, то можно полагать, что вы находитесь на этапе 5. В некоторых организациях вы уже заслуженно получите репутацию специалиста по сбору данных.

На этом этапе можно устранить 99% возникающих проблем. Оставшиеся 1% нерешенных проблем могут быть решены с помощью других языков программирования, исключая М. Поэтому ваш переход к следующему этапу определяется исключительно интеллектом или опытом разработки программного обеспечения. Если вы не являетесь суперпрофессионалом в области разработки программного обеспечения, существенной причины для перехода к последнему этапу нет.

Этап 6. Кодирование в произвольной форме

в окне расширенного редактора

На шестом и последнем этапе освоения языка M вы сможете создавать сложнейшие M-выражения "с нуля", не используя пользовательский интерфейс редактора Power Query. В этом случае кодирование в произвольной форме является естественным для вас, у вас должна быть исключительно хорошая память, поскольку расширенный редактор не предоставляет возможности автозаполнения или IntelliSense (по крайней мере, на момент написания этой книги). На этом этапе вам необходимо запоминать большое число имен функций и их аргументов.

Обширный опыт работы с языками программирования поможет вам достичь этого этапа, и если вы находитесь на этом этапе, то можно искать новые способы совершенствования. Можно написать код M для более широкой аудитории, поделиться функциями M с коллегами или со всем сообществом или, в качестве следующего этапа, приступить к разработке пользовательских коннекторов данных в Visual Studio.

См. также

Если вы являетесь разработчиком на языке M и находитесь на этапе 6, то можете создавать настраиваемые коннекторы данных для Power BI и позволять бизнес- пользователям подключаться к службам и приложениям, получать доступ к данным, которые не предоставлены Power BI как готовые коннекторы. Описание настраиваемых коннекторов данных выходит за рамки этой книги. Чтобы узнать, как их разработать, обратитесь на сайт https://github.com/Microsoft/DataConnectors.

Ресурсы в Интернете

Для перехода к следующему этапу обучения или ознакомлению с новой функцией M обратитесь к официальной ссылке на Power Query M, которая является изначальным ресурсом. Этот ресурс доступен в библиотеке Microsoft MSDN по адресу: https://msdn.microsoft.com/query-bi/m/power-query-m-reference

Довольно часто на официальном справочном сайте отсутствуют конкретные примеры, которые помогают лучше понять ту или иную функцию. Вам предлагается поискать в Интернете соответствующие статьи в блогах на эту тему. Вот некоторые из основных блогов, посвященные Power Query:

■ Блог Криса Уэбба BI (Chris Webb): https://blog.crossjoin.co.uk/category/power-query/

■ Блог Кена Палса (Ken Puls) (гуру Excel) (Excelguru): https://www.excelguru.ca/blog/

■ Блог Мэтта Аллингтона (Matt Allington) (Excelerator BI): https://exceleratorbi.com.au/category/power-query-2/

■ Блог Реза Рэда (Reza Rad) (RADACAD): http://radacad.com/category/power-query

■ Блог Имке Филдмана (Imke Feldmann) The BICCOUNTANT: http://www.thebiccountant.com/

■ Блог DataChant: https://datachant.com

Локальные ресурсы

Представьте, что вы путешествуете в самолете. Не имея подключения к Интернету, но при наличии острой необходимости понять, как работает определенная функция M, обратитесь к редактору Power Query, который включает автономные справочные страницы для всех своих функций. Иногда такие автономные страницы содержат больше информации и примеров, чем онлайн-версия MSDN, и вам предлагается это проверить.

Для просмотра автономной документации по той или иной функции просто введите знак равенства и имя функции в строке формул редактора Power Query. Не включайте в запись скобки или аргументы. После нажатия клавиши <Enter> вы увидите автономную документацию на панели предварительного просмотра. Например, при использовании функции Table.ColumnNames (Источник) для возвращения

Файл

Главная

Преобразование Добавление столбца

Просмотр

Закрыть и

загрузить’

Do

Обновить предварительный

просмотр ’

|\_д Свойства

Д Расширенный редактор й Управление’

Закрыть

Запрос

Выбор Удалить Сократить

столбцов\* столбцы строки’

Управление столбцами Сортировать

Разделить Группировать столбец ’ по

Тип данных: Любой \*

Н Использовать первую строку в качестве заголовков \*

Замена значений

Преобразование

Объединить

| □!

Управление

параметрами ’

Параметры

ИС1

Ист

У yf fx \ = Table.СоlumnNames V

g\_ Table.ColumnNames

Возвращает имена столбцов в таблице table в виде текстового списка.

Введите параметр

table

Вызвать Очистить

function (table as table) as list

Пример: Нахождение имен столбцов таблицы.

Использование:

Table.ColumnNames(Table.FromRecords({ [CustomerlD = lr Name = "Bob", Phone = "123—4567"], [CustomerlD = 2, Name = "Jim", Phone = "987-6553"] , [CustomerlD = 3, Name = "Paul", Phone = "543-7890"] , [CustomerlD = 4, Name = "Ringo", Phone = "232-1550"]}))

Выходные данные:

{"CustomerlD", "Name", "Phone"}

ГОТОВО

Рис. 9.2. Введите имя функции в строку формул для просмотра автономной документации

по функциям языка М в окне редактора Power Query

Введение в язык формул М Power Query 273

списка имен столбцов таблицы "Источник" можно удалить скобки и Источник, как показано здесь, и увидеть автономную документацию по функции Table.ColumnNames (рис. 9.2):

= Table.ColumnNames

УПРАЖНЕНИЕ 9.1. Применение переменной #shared для изучения встроенных функций

Для нахождения новых функций и ознакомления с автономной документацией можно запустить редактор Power Query и в строке формул ввести строку:

= #shared

Эта переменная возвращает запись всех поддерживаемых функций в Power Query, включая ваши запросы.

Рассмотрим, как использовать переменную #shared для получения доступа ко всем функциям, а затем перейдем к конкретной функции.

1. Создайте пустой запрос, как показано ниже.

В Excel: на вкладке Данные (Data) выберите команду Получить данные (Get Data) и затем выполните команды Из других источников (From Other Источник) Пустой запрос (Blank Query).

В Power BI Desktop: на вкладке Главная страница (Home) выберите раскры-вающееся меню Получить данные (Get Data) и затем выберите команду Пустой запрос (Blank Query).

2. После открытия окна Power Query, если не отображается строка формул, активизируйте ее на вкладке Вид (View).

3. Введите в строку формул следующую строку и нажмите клавишу <Enter>:

= #shared

На панели предварительного просмотра отобразится таблица. Это не классическая таблица, а запись. В первом столбце приведен список ключей записи (также называемых именами полей). Во втором столбце приведены все значения. За исключением первой строки, в которой показан ключ Query1 (текущий запрос, над которым вы работаете), все остальные значения имеют формат функций.

4. Выберите пустое пространство в любой из ячеек функции, чтобы просмотреть в нижней части ячейки документацию по этой функции. Щелкните на ссылке Функция (Function) для перехода к фактической функции.

5. Для нахождения всех функций списка с помощью пользовательского интерфейса Power Query преобразуйте выходную запись #shared в таблицу, выбирая команду В таблицу (Into Table) на вкладке Инструменты преобразования записи (Record Tools Convert).

6. Выберите элемент управления фильтра в столбце Имя (Name) и задайте команду Текстовые фильтры (Text Filters). Затем установите переключатель Начинается с (Begins With).

7. После открытия окна диалога Фильтры по строкам (Filter Rows) введите List (Список) в текстовое поле справа от раскрывающегося списка Начинается с и щелкните мышью на кнопке OK в окне диалога.

Теперь можно просмотреть все функции из списка и изучить их автономную документацию, как описано на шаге 4.

Файлы решения C09E01 - Solution.xlsx и C09E01 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Компоненты языка M

Центральный компонент языка М — выражение — блок кода, в результате вычисления которого получаем одно значение. Каждый шаг преобразования, созданный в редакторе Power Query, генерирует M-выражение. Панель предварительного просмотра всегда отображает значение, полученное в результате вычисления выражения.

В языке M каждый шаг преобразования представляет собой пару, состоящую из идентификатора и его выражения. Строка формул отображает элемент выражения на выбранном шаге, а на панели Примененные шаги (Applied Steps) можно просмотреть идентификатор. В окне диалога Расширенный редактор (Advanced Editor) можно увидеть все выражение M-запроса. И хотя код в окне расширенного редактора немного сложнее, чем одношаговое M-выражение в строке формул, в результате выполнения этого кода возвращается одно значение. Следующее упражнение демонстрирует это на простом выражении запроса, возвращающем текст Hello World.

УПРАЖНЕНИЕ 9.2. Hello World

В упражнении показано, как сопоставить имеющиеся пользовательские интерфейсы в Power Query с выражениями и значениями языка M.

1. Запустите редактор Power Query в режиме пустого запроса. (Не знаете, как это сделать? Выполните упражнение 9.1, шаг 1.)

2. На вкладке Главная (Home) выберите команду Расширенный редактор (Advanced Editor) и вставьте следующий код:

let

Источник = "Hello",

Step2 = Источник & " ",

Step3 = Step2 & "World"

in

Step3

3. Щелкните мышью на кнопке Готово (Done) для закрытия окна расширенного редактора. На панели предварительного просмотра появится сообщение Hello World.

Поздравляю. Вы написали первую программу "Hello World" на языке M. Рассмотрим код в терминологии M и выясним, как он связан с элементами пользовательского интерфейса. На рис. 9.3 приведены основные связи между элементами пользовательского интерфейса расширенного редактора, строкою формул и панелью Примененные шаги и соответствующими строительными блоками языка М.

На панели предварительного просмотра можно видеть значение Hello World. Обычно на панели просмотра отображаются таблицы, но в нашем случае показано текстовое значение. Выражение, включенное в окно расширенного редактора, возвращает единственное значение Hello World. В строке формул выводится такой код:

= Step2 & "World"

Этот код является выражением, которое присваивается идентификатору Step3 в окне расширенного редактора. Он объединяет значение Step2 с текстом World. И хотя выражение находится в строке формул, идентификатор Step3 представлен на панели Примененные шаги (Applied Steps).

Если выбрать шаг Step2 на панели Примененные шаги (Applied Steps), то в строке формул отобразится следующий код:

= Источник & " "

В окне расширенного редактора это выражение представлено с сопутствующим идентификатором Step2 следующим образом:

Step2 = Источник & " ",

Завершающая запятая в этом выражении не отображается в строке формул, но служит для разделения между парами идентификатор/выражение, которые являются частью выражения let.

И хотя большинство запросов транслируется в "линейную" последовательность из M-шагов, таких как Источник, Step2 и Step3 (как показано в упражнении 9.2), язык M позволяет описывать сложные потоки и применять условия, которые приведут к различным путям для исполнения. Далее это будет показано в разделе "Выражение if внутри выражения let". Но прежде чем рассмотреть более сложные сценарии, внимательно изучим возможности применения выражения let. В конце концов, let — это первый элемент языка M, с которым мы сталкиваемся при открытии окна редактора Power Query.

Выражение let

Всякий раз при создании новых запросов с помощью редактора Power Query и импорте наборов данных в отчет после некоторой подготовки данных генерируется базовое выражение let. После открытия окна расширенного редактора отображается выражение M, которое начинается с ключевого слова let и обычно заканчивается in, за которым следует идентификатор последнего шага.

Выражения let позволяют создавать подмножества меньших выражений, которые объединяются в одно значение. Рассмотрим простой пример из упражнения 9.2. В этом случае выражение let состоит из трех шагов. Step1 возвращает значение Hello. Step2 возвращает значение в Step1 и завершающий пробел, а Step3 возвращает значение в Step2 и завершающее значение World. Значение in определяет вторую фазу вычислений, которую возвращает выражение let. В этом базовом примере после in определяется только Step3. В результате возвращаемое значение является тем же значением в Step3: Hello World.

Ключевое слово let позволяет записывать выражения, которые выполняются после ключевого слова in. Вот пример кода, возвращающий текст Hello World!!!:

let

Источник = "Hello",

Step2 = Источник & " ",

Step3 = Step2 & "World"

in

Step3 & "!!!"

В этом выражении вместо возвращения значения в Step3 применяется новое выражение после ключевого слова in, которое объединяет Step3 с !!!.

Далее приводится более простой пример, где определяется идентификатор ExclamationMarks внутри раздела let, затем он вычисляется в выражении in.

let

Источник = "Hello",

Step2 = Источник & " ",

Step3 = Step2 & "World",

ExclamationMarks = "!!!"

in

Step3 & ExclamationMarks

Выражение let может создаваться как сложное выражение, составленное из вло-женных выражений let. Далее приводится пример, возвращающий текст Hello World!!!:

let

FirstWord =

let

Источник = "Hello"

in

Источник,

SecondWord =

let

Источник = "World"

in

Источник

in

FirstWord & " " & SecondWord & "!!!"

И хотя это выражение полностью допустимо, его применение усложняет работу с редактором Power Query. Вследствие вычислений, которые необходимы в конечном выражении, выражение M приводится в строке формул, а отдельные шаги не отображаются на панели Примененные шаги (Applied Steps). Для того чтобы решить эту проблему и получить отдельные шаги на панели Примененные шаги (Applied Steps), как показано на рис. 9.4, можно переписать код М, добавляя идентификатор Result (Результат), и переместить выражение, располагая его перед заключительным ключевым словом in:

let

FirstWord =

let

Источник = "Hello"

in

Источник,

SecondWord =

let

Источник = "World"

in

Источник,

Result = FirstWord & " " & SecondWord & "!!!"

in

Result

Можно заметить, что идентификатор Источник дважды применяется в двух предыдущих выражениях. Обычно, если один и тот же идентификатор дважды встречается внутри выражения let, получаем следующую ошибку:

The variable named 'Источник' is already defined in this scope

(Переменная 'Источник' уже определена в этом диапазоне).

Рис. 9.4. Наличие выражения вместо идентификатора после раздела in в окне расширенного редактора приводит к созданию отдельного шага на панели Примененные шаги.

Приведенное ранее выражение M демонстрирует это решение

Тем не менее в двух предыдущих выражениях содержатся два различных выражения let, вложенных во внешнее выражение let. Все подобные идентификаторы, определенные внутри выражения let, являются локальными, и к ним невозможен доступ из внешнего выражения let. Если Источник оценивается как значение SecondWord, то Источник в FirstWord не является доступным, поскольку два экземпляра Источник не могут совместно использовать один и тот же диапазон.

Объединение выражений

из нескольких запросов и областей доступа

Представление об области доступа в языке M и умение писать выражения let может быть весьма полезным. Часто возникает необходимость повторно использовать некоторые этапы преобразования из запроса и применить их в другом запросе. Чтобы правильно скопировать и вставить строки выражений из одного запроса в другой, обычно следует выполнить три корректирующих шага в объединенном выражении. Рассмотрим этот вопрос подробнее.

Представим, что у вас есть запрос с тремя шагами преобразования, как показано в этом псевдокоде:

let

Stepl =

Step2 = ..,

Step3 = .

in

Step3

Во втором запросе, находящемся в разделе Step2, применялся полезный фильтр, который желательно использовать в первом запросе:

let

Stepl = .,

Step2 = Table.SelectRows(Stepl, .)

in

Step2

Таким образом Step2 копируется из второго запроса в первый запрос, как показано в следующем примере:

let

Stepl = .,

Step2 = .,

Step3 = .

Step2 = Table.SelectRows(Stepl, .)

in

Step3

Конечно, этот код неверен. Простого копирования и вставки строки будет недостаточно. Обратите внимание на три ошибки, которые следует устранить:

1. Step2 уже определен в исходном выражении, и получаем следующую ошибку:

The variable named 'Step2' is already defined in this scope

(Переменная 'Step2' уже определена в этой области доступа)

2. Первым аргументом функции Table.SelectRows в исходном варианте был Stepl, но теперь необходимо применить данное преобразование к Step3.

3. После ключевого слова in сначала возвращается Step3 как значение этого выражения, теперь следует вернуть значение в последнем идентификаторе.

Вот измененное выражение:

let

Stepl =

Step2 = ..,

Step3 = .

Step4 = Table.SelectRows(Step3, ...)

in

Step4

Полужирным шрифтом выделены изменения, которые необходимы для коррекции выражения. Сначала необходимо переименовать второй идентификатор Step2 как Step4. Затем следует модифицировать аргумент в Table.SelectRows для передачи последнего шага перед Table.SelectRows, которым является Step3. Наконец, следует обновить идентификатор, находящийся после ключевого слова in для правильной передачи последнего шага, которым является Step4.

Всякий раз при копировании строк кода из одного запроса в другой необходимо настраивать код и "видоизменять" новые шаги преобразования.

В некоторых случаях можно скопировать множество строк кода из одного запроса в другой. В подобных ситуациях иногда довольно сложно изменять сразу все строки, особенно если в объединенном выражении много дублирующих идентификаторов. Для разрешения этой проблемы можно применить вложенное выражение let, как показано в этом примере, которое возвратит то же значение, что и предыдущее выражение:

let

Stepl = .,

Step2 = .,

Step3 = .

Step4 =

let

Step2 = Table.SelectRows(Step3, .)

in

Step2

in

Step4

Обратите внимание, что область во вложенном выражении let содержит все идентификаторы in внешнего диапазона, исключая Step2, который локально переопределяется в области вложенного выражения let и не воспринимается в дальнейшем как дублирующий идентификатор. Когда вычисляется Step4, внешнее значение Step2 не переписывается с помощью значения, которое возвращается внутренним Step2. Как объяснялось ранее, два идентификатора Step2 имеют различные диапазоны.

Наконец, вот пример, где внешний идентификатор используется во вложенном выражении let и затем переопределяется во внутреннем диапазоне вложенного let:

let

A = l,

B =

let

A = A + 1

in

A

in

B

Это выражение возвращает число 2. Идентификатор A определяется дважды. Сначала A определяется в контексте верхнего выражения let и получает значение 1. Затем внутри вложенного let определяется новый идентификатор A с выражением, которое использует внешний идентификатор A и увеличивает его на единицу.

Однако внутренний идентификатор A отличается от внешнего идентификатора A и недоступен во внешнем выражении let. Рассмотрите, что происходит, когда используется то же выражение, но на сей раз идентификатор A возвращается последним выражением in:

let

A = 1,

B =

let

A = A + 1

in

A

in

A

Этот код возвращает число 1 вместо 2 даже в том случае, если A увеличен во внутреннем выражении let. Во внешней области A по-прежнему будет равен 1. Можно снова переписать выражение, на этот раз с учетом правильной области, в которой механизм языка M оценивает выражения:

let

A\_external\_scope = 1,

B =

let

A\_internal\_scope = A\_external\_Scope + 1

in

A\_internal\_scope

in

A\_external\_scope

СОВЕТ

Определение диапазона может быть довольно сложным, но суть в том, что можно избегать неожиданных результатов и гарантировать, что идентификаторы всегда будут уникальными. Теперь, после изучения выражения let, можно рассмотреть применение различных типов в языке M.

Типы, операторы

и встроенные функции в языке M

Значение в языке M — это данные, полученные путем вычисления выражения. В этом разделе рассказано об основных видах значений в языке М. Каждый тип значения связан с определенным синтаксисом, набором значений подобного типа, набором операторов, функциями, которые применяются со значениями подобных типов, и определением типа, который наследуется.

Рассмотрим различные языковые элементы, зависящие от типа, обращая внимание на числовые значения в языке M. Числа могут быть представлены в виде литералов, таких как 0, 100, -2.5 или 2.5e-7. Этими значениями можно манипулировать с помощью семейства встроенных M-функций, начинающихся с префикса Number, за которым следуют точка и имя функции. Например, Number.Abs — встроенная функция, которая принимает число в качестве аргумента и возвращает его абсолютное значение. Числа в языке M имеют встроенные операторы, такие как =, <, >, <=, >=, + и -. Более подробно операторы рассмотрены далее в этом разделе.

Язык M позволяет объявлять типы в аргументах или возвращаемых значениях пользовательских функций и проверять, относится ли конкретное значение к определенному типу. Например, следующее выражение, введенное в строке формул, вернет TRUE, как показано на рис. 9.5.

= 1000 is number

ПРИМЕЧАНИЕ

Язык M чувствителен к регистру символов. Логические значения, отображаемые в верхнем регистре, TRUE и FALSE, как показано на рис. 9.5, используются только для вывода информации. В выражениях M следует применять версию в нижнем регистре (true или false); в противном случае вы получите ошибку. Если, например, записать TRUE в формуле M, то возникнет ошибка:

Expression.Error: The name 'TRUE' wasn't recognized.

Make sure it's spelled correctly

(Имя 'TRUE' не распознано. Убедитесь в том, что оно написано корректно). Типы также определяются во встроенных функциях. Например, функция Number.Abs объявляется следующим образом:

Number.Abs(number as nullable number) as nullable number

Данная строка, называемая объявлением функции, объясняет, что Number.Abs принимает один аргумент, который может быть числом или нулем. Возвращаемое значение этой функции — число или нуль. Знание этого синтаксиса позволяет понять, что вы можете выполнить выражение Number.Abs, обращаясь к числу, следующим образом:

= Number.Abs(-1000)

В результате возвращается число:

1000

Вы также можете выполнить эту функцию, используя аргумент null следующим образом:

= Number.Abs(null)

В результате отображается значение null:

null

Если попытаться выполнить функцию для других типов данных, например текстового значения, то вы получите сообщение об ошибке. Применив функцию к текстовому значению "hello":

= Number.Abs("hello")

в результате получим такую ошибку:

Expression.Error: We cannot convert the value "hello" to type Number

(Невозможно преобразовать значение "hello" в числовой тип).

Для простоты изложения в этой главе будем ссылаться на встроенные функции определенного типа, такого как Type.X. Например, уже в следующем разделе чаще всего встретитесь с функциями Number.X, такими как Number.Abs. После ознакомления с применением типов в языке M в следующих разделах подробнее рассмотрим основные типы, их операторы и встроенные функции.

Основные типы данных языка M

Понимание типов в языке M, а также способов работы со значениями полезно при преобразовании столбцов, добавлении настраиваемых столбцов или применении фильтров со сложными логическими выражениями. И хотя пользовательский интерфейс редактора Power Query позволяет реализовать большое число преобразований, знание типов M, их операторов и встроенных функций помогает расширять возможности по передаче данных в столбцах.

В данном разделе приведены примеры простых формул, которые используются вне контекста подготовки данных. Если вы поймете, как управлять этими типами, можно легко применять операторы или встроенные функции в строке формул для уточнения преобразований на основе столбцов.

Например, представьте, что имеется функция Number.X, которую нельзя разыскать в пользовательском интерфейсе, допустим Number.Sqrt, которая помогает преобразовать числа в первом столбце в значения квадратного корня. Можно выбрать числовой столбец в окне редактора Power Query и применить любое из числовых преобразований. Далее приводится формула, сгенерированная после выбора типа данных Научный Абсолютное значение (Scientific Absolute Value):

= Table.TransformColumns(fcTO4HUK,{{"Column1", Number.Abs, type number}})

Теперь можно изменить Number.Abs на функцию Number.X (в данном случае Number.Sqrt) для преобразования числовых значений в каждой ячейке Column1:

= Table.TransformColumns(fcTO4HUk,{{"Column1", Number.Sqrt, type number}})

Если функция Number.X требует больше чем один аргумент, можно указать ключевое слово each, за которым следует имя функции и символ подчеркивания \_, что позволит представить числовое значение, затем можно добавить другие аргументы.

Вот, например, формула для возведения в степень чисел, находящихся в столбце Column1:

= Table.TransformColumns(fcTO4HUk,{{"Column1", each Number.Power( ,3), type number}})

В следующих разделах рассматриваются основные типы данных языка M: число, время, дата, длительность, текст, ноль и логический.

Числовой тип данных

В языке M числовое значение используется для числовых и арифметических операций. Вот несколько примеров чисел: 1, -3.14 и 2.0e5. Допускается шестнадцатеричный формат. Например, значение 0xff возвращает число 255.

Число в языке М представлено с точностью по меньшей мере 64 разряда, и числа могут находиться в диапазоне, приблизительно от 5,0\*10 3 4 до 1,7\*1030 , с точностью 15 или 16 цифр.

Следующие специальные значения также считаются числовыми значениями:

■ Положительная бесконечность: +#infinity

■ Отрицательная бесконечность: -#infinity

Бесконечные значения формируются в результате выполнения таких операций, как деление ненулевого числа на нуль. Например, 1.0/0.0 дает положительную бесконечность, а -1.0/0.0 — отрицательную бесконечность.

■ Не-число, часто употребляется сокращение NaN: #nan

Значение NaN может быть получено с помощью недопустимых операций с плавающей точкой, таких как деление нуля на нуль.

Вы можете сравнивать два числа, применяя логические операторы >, > =, <, <=, = и <>. Основные арифметические операции могут быть выполнены с использованием +, -, \* и /.

Язык M предоставляет богатый набор встроенных функций Number.X. Постоянные функции, такие как Number.Pi, которая возвращает значение числа п, позволяют использовать специальные числа в преобразованиях. Функции преобразования, такие как Number.From, могут преобразовывать значения в числа. Например, следующая функция преобразует текст "6" в число 6:

= Number.From("6")

Многие математические операции и тригонометрические функции Number.X также доступны в редакторе Power Query, например Number.Power и Number.Sin, на вкладках Преобразование (Transform) и Добавить столбец (Add Column). Для просмотра перечня основных функций Number.X обратитесь на сайт https://msdn.microsoft.com/ query-bi/m/number-functions.

Функции времени

В языке М значение времени представлено как число 100-наносекундных тактов, прошедших с полуночи. Максимальное число тактов с полуночи соответствует 23:59:59.9999999 часам. Значения времени могут быть созданы с использованием выражения #time, синтаксис которого следующий: #time(4ac, минута, секунда). Например, 2:53 после полудня можно записать в строке формул следующим образом:

= #time(14, 53, 0)

что приведет к такому результату:

2:53:00 PM

Значения времени можно обрабатывать с помощью логических операторов =, <>, >=, >, < и <=, причем эти значения могут быть левым или правым операндом в выражениях с операторами +, - и & со значениями времени и продолжительности. Например, добавить 7 минут ко времени 2:53 вечера можно следующим образом:

= #time(14,53,00) + #duration(0,0,7,0)

Что приведет к такому результату:

3:00:00 PM

Функции Time.X позволяют извлекать элементы времени из значения времени или вычислять начало или конец часа. Например, применение функции Time.Hour к временному значению

= Time.Hour(#time(14, 53, 0))

вернет значение часа в виде числа:

14

Для того чтобы узнать больше о функциях времени, обратитесь к соответствующим преобразованиям редактора Power Query на вкладках Преобразование (Transform) и Добавить столбец (Add Column). Дополнительная информация о функциях времени приведена на сайте https://msdn.microsoft.com/query- bi/m/time-functions.

Тип даты

Значение даты представляется в виде числа дней, прошедших с начала эпохи (1 января 0001 г., общее время, прошедшее с начала летоисчисления по григорианскому календарю). Максимальное число дней с начала эпохи составляет 3 652 058, что соответствует 31 декабря 9999 года. Значения даты можно получить с помощью выражения #date, используя следующий синтаксис: #date(year, month, day). Например, дату 2 апреля 2018 года можно записать в строке формул следующим образом:

= #date(2018, 4, 2)

что приведет к следующему результату (в формате США):

4/2/2018

Тип даты поддерживает логические операторы =, <>, >=, >, < и <= и может быть левым или правым операндом в выражениях с операторами +, - и & со значениями длительности.

В главе 2 было показано, как извлекать значения даты и применять вычисления, связанные с датой, в редакторе Power Query. Для каждого шага преобразования можно найти эквивалентную функцию Date.X. В языке М имеется большое число функций Date.X. Например, функция Date.AddDays получает дату в качестве первого аргумента и число дней для добавления в качестве второго аргумента.

В следующем выражении добавляются 5 дней к дате 2 апреля 2018 г.:

= Date.AddDays(#date(2018, 4, 2), 5)

что приводит к выводу соответствующей даты, 7 апреля 2018 (в формате США): 4/7/2018

Полный список встроенных функций Date.X можно просмотреть на сайте https://msdn.microsoft.com/query-bi/m/date-functions.

Подобно значениям даты и времени, язык M поддерживает также значения DateTime и DateTimeZone. Типами для них являются datetime и datetimezone, их можно автоматически генерировать в редакторе Power Query в случае применения соответствующих преобразований, доступных на вкладке Преобразование (Transform) или на вкладке Добавить столбец (Add Column). Имеется большое число функций DataTime.X и DateTimeZone.X, включая DateTime.LocalNow, которые позволяют вычислить текущее время и дату, рассчитать прошедшее время транзакций или применить фильтры относительного времени.

Тип длительности

Значение длительности хранит числовое представление интервала между двумя точками на временной шкале, измеренное в 100-наносекундных тактах. Величина длительности может быть как положительной, так и отрицательной, с положительными значениями, обозначающими увеличение времени, и отрицательными значениями, обозначающими уменьшение времени. Минимальные и максимальные значения, которые можно сохранить в продолжительности, составляют 10 675 199 дней 2 часа 48 минут и 5,4775808 секунд назад или вперед во времени.

Значения длительности строятся на основе выражения #duration. Например, продолжительность 5 дней и 10 минут может быть представлена следующим образом:

#duration(5, 0, 10, 0)

и дает такой результат:

5.00:10:00

Чаще всего значения длительности применяют к двоичным операторам, таким как + и - в сочетании со значениями даты, времени, datetime и datetimezone. Например, можно добавить 5 дней к дате 2 апреля 2018:

= #date(2018, 4, 2) + #duration(5, 0, 0, 0)

что приводит к результату, соответствующему 7 апреля 2018:

4/7/2018

Вычитание двух значений даты или двух значений времени приводит к такому значению длительности:

= #date(2018, 08, 28) - #date(2018, 01, 01)

что дает результат:

239.00:00:00

С помощью встроенных функций Duration.X можно вычислить значения длительности, что позволит извлечь дни, часы, минуты или секунды для компонентов длительности. Функции Duration.TotalX, такие как Duration.TotalSeconds, позволяют организовать подсчет количества секунд, которые составляют определенную продолжительность. Например, следующее выражение вычисляет общее количество секунд в 5 днях:

= Duration.TotalSeconds(#duration(5, 0, 0, 0))

и выводит такой результат:

432000

Полный перечень встроенных функций Duration.X можно просмотреть на сайте https://msdn.microsoft.com/query-bi/m/duration-functions.

Текстовый тип данных

Текстовые типы данных могут содержать произвольные символы Юникода. Можно объединять два текстовых значения с помощью оператора &, как было показано в упражнении 9.1.

Язык M предоставляет широкий спектр функций Text.X, которые можно применять к текстовым значениям. Например, с помощью функции Text.From можно преобразовать число в текст. В следующем примере выполняется объединение текста и числа, но сначала число преобразуется в текст:

= "Testing, Testing " & Text.From(123)

и выводится следующее значение:

Testing, Testing 123

В отличие от сложных типов данных, в языке M для доступа к определенному символу в тексте нельзя применять операторы, такие как [] или {}. Для получения доступа к первому символу можно использовать функцию Text.At с текстом в качестве первого аргумента и нулевым базовым индексом в качестве второго аргумента:

= Text.At("ABCD", 0)

Это выражение дает следующий результат:

A

А функция:

= Text.At("ABCD", 1)

дает следующий результат:

B

Полный перечень текстовых функций можно просмотреть на сайте https://msdn.microsoft.com/query-bi/m/text-functions.

Тип данных Null

Тип данных null представляет отсутствие значения. Можно применять логические или арифметические операторы как нулевые, и возвращаемый результат также будет нулевым. Например, в следующих выражениях, если A равно нулю, то результаты будут равны нулю:

= A < 5 = A >= 5 = A + 1 = A \* 5

Особый случай — применение операторов равенства (=) и неравенства (<>) к нулевому значению.

В следующих выражениях результат равен true или false, если A равно нулю:

= A = null

на панели предварительного просмотра будет выведено:

TRUE

Выражение

= A <> 5

приводит к следующему результату, отображаемому на панели предварительного просмотра:

FALSE

Выражение

= A <> null

приводит к следующему результату, отображаемому на панели предварительного просмотра:

FALSE

= A is null

приводит к следующему результату на панели предварительного просмотра:

TRUE

Обратите внимание, что null является как типом, так и значением. Таким образом, можно применять либо соответствующий оператор, либо знак равенства, чтобы проверить, является ли определенное значение нулевым.

Логический тип данных

Логическое значение используется для логических операций и может быть true или false. Логические значения могут применяться вместе со следующими операторами:

■ A > B (A больше чем B)

■ A >= в (A больше чем или равно B)

■ A < в (A меньше чем B)

■ A <= в (A меньше чем или равно B)

■ A = в (A равно B)

■ A <> в (A не равно B)

■ A or B (A или B)

■ A and B (A и B)

■ not A (не A)

Можно проверить, является ли определенное значение логическим, используя выражение is logical. Например, если A содержит значение false, то следующее выражение вернет значение true:

= A is logical

На панели предварительного просмотра отображается результат TRUE.

Логические операторы обычно присутствуют в выражениях if, которые описаны далее в этой главе.

Сложные типы данных

Пришло время познакомить вас со сложными типами языка M — списками, записями и таблицами — и связями между ними. Базовые типы определяют одно или атомарное значение, сложные типы имеют структурированный формат, который позволяет включить несколько базовых типов или другие сложные типы.

При импорте данных в Power Query обычно работают с таблицами. Но таблица может содержать вложенные записи, списки и таблицы (при работе с неструктурированными данными, такими как файлы JSON или XML). Поэтому можно преобразовать ее в список (при детализации до столбца) и даже — в список записей. Рассмотрим самый простой из трех типов, тип списка.

Тип списка

Список включает последовательность значений, которые могут быть перечислены. Список может содержать любые значения, в том числе списки, записи, таблицы или любые комбинации различных типов. Списки в выражении M формируются с использованием синтаксиса инициализации, в котором фигурные скобки инкапсулируют значения, разделенные запятыми. Например, данный список включает числа, логические элементы и текст:

= {1, 3, true, false, "Hello World"}

Следующее выражение, как показано на рис. 9.6, является примером инициализации выражения списка чисел 1, 2 и 3:

= {1, 2, 3}

Рис. 9.6. Можно инициализировать список в строке формул, используя фигурные скобки

Предыдущий список можно инициализировать, записав следующее выражение:

= {1:3}

Если необходимо создать список увеличивающихся чисел от 1 до 100, можно применить следующее выражение инициализации:

= {1:100}

Использование подобного выражения — полезный метод для создания потоков итераций в Power Query, которые рассматриваются далее в этой главе.

Для создания списка инкрементных чисел от 1 до N, где N предоставляется другим запросом (например, на основе другого источника данных), можно применить следующее выражение:

= {1:N}

Достичь того же результата позволяет функция List.Numbers:

= List.Numbers(1, N)

Оператор двух точек (..) дает возможность определить числовой диапазон в списке. Например, следующее выражение возвращает список чисел 1, 4, 5, 6, 7, 8, 9, 10, 15:

= { 1, 4..10, 15 }

Количество элементов в списке можно определить с помощью функции List.Count:

= List.Count( { 1..10} )

Эта функция возвращает следующий результат:

10

Также можно определить пустой список в M, используя фигурные скобки, без каких-либо элементов между ними.

Применение функции List.Count для пустого списка

= List.Count( {} )

приведет к выводу нуля.

Операторы списка

Для объединения двух списков предусмотрен оператор объединения (&). Вот пример, возвращающий список {1, 2, 3, 4, 5}:

= {1, 2} & {3, 4, 5}

Чтобы определить, равны ли два списка, можно использовать операторы равенства (=) и неравенства (< >).

Следующее выражение, введенное в строке формул, возвращает значение TRUE, которое отображается на панели предварительного просмотра:

= {1, 2} = {1, 2}

Следующее выражение, введенное в строке формул, тоже возвращает TRUE на панели предварительного просмотра:

= {2, 1} <> {1, 2}

Два списка с одинаковыми элементами и разным порядком их следования не равны. Для получения доступа к элементу в списке по его местоположению можно задать нулевой индекс в фигурных скобках. Например, выражение

= {"A", "B", "C"}{0}

возвращает значение:

A

Выражение

= {"A", "B", "C"}{2}

возвращает значение:

C

При попытке доступа к элементу, расположенному за пределами списка, получаем ошибку выражения. Например, попытка доступа к четвертому элементу (индекс 3) в списке из трех элементов

= {"A", "B", "C"}{3}

приводит к следующей ошибке:

Expression.Error: There weren't enough elements in the enumeration to complete the operation

(В перечислении не хватило элементов для завершения операции).

Получать доступ к первым или последним элементам в списке позволяют встроенные функции List.First и List.Last:

= List.First({"A", "B", "C"})

получим следующий результат:

A

Выражение

= List.Last({"A", "B", "C"})

возвращает следующий результат:

C

Функции List.X

Знание общих встроенных функций M для списков удобно, если нужно выполнить определенные операции с содержимым столбца. Например, можно определить среднее значение в числовом столбце, а затем применить к нему фильтр. Для того чтобы начать работу со значениями в столбце в M, можно щелкнуть правой кнопкой мыши на заголовок столбца и выбрать детализацию; результатом будет список. Тогда можно применить одну из встроенных функций. На рис. 9.6 можно увидеть шаги преобразования, которые доступны на вкладке Преобразование (Transform) вкладки Средства для списков (List Tools). На самом деле язык M предоставляет гораздо больше функций, чем показано на вкладке.

Язык M включает богатую библиотеку встроенных функций для списков. Вот некоторые из наиболее распространенных функций и их категории:

■ Информационные функции — в эту категорию попадает уже описанная функ-ция List.Count, а также функция List.IsEmpty, которая возвращает true, если список пуст, и false — в противном случае.

■ Функции выбора — возвращают выбранный элемент из списка или подсписка. Например, функция List.Select может применяться для фильтрации элементов в списке с соблюдением определенных критериев. Функции List.FirstN и List.LastN возвращают подсписок первых или последних N элементов исходного списка.

■ Функции преобразования — управляют списками. Одними из наиболее упот-ребляемых функций являются List.Transform, которые управляют элементами с помощью данной функции, и List.Combine, действующая как оператор конкате-нации (&). Функция List.Accumulate, которая упоминалась на пятой стадии зрело-сти в обучении языку M, также входит в эту категорию и рассматривается далее в этой главе.

■ Функции принадлежности — применяют логические операции к элементам в списке и возвращают true или false, если выполняются определенные условия. Например, функция List.Contains возвращает значение true, если в списке найдено определенное значение.

■ Операции над множествами — применяют операции над множествами для двух или более списков. Функция List.Intersect позволяет найти общие элементы в двух списках, а функция List.Union дает возможность добавлять списки и удалять дубликаты. Полезная функция List.Difference, которая понадобится в главе 10, возвращает все элементы в списке, недоступные во втором списке. Она применяется в именах столбцов, что позволит ссылаться на другие имена столбцов, помимо упомянутых.

■ Функции упорядочения — в дополнение к функции List.Sort, которая пригодна для чисел или аргументов любого типа, где можно определить функцию упорядочения, также позволяют извлекать элементы по внутреннему порядку значений, а не только по их порядку в списке. Например, функции List.Min и List.Max извлекают самые малые или самые большие значения, а функции List.MinN и List.MaxN извлекают n-е самые малые или самые большие значения.

■ Арифметические функции — в эту категорию входят функции, выполняющие усреднение, сложение и другие числовые операции. Например, функции List.Average и List.StandardDeviation могут работать с элементами чисел, датами и временем в списках одного типа. Функция List.Sum оперирует с числами и значениями длительности, возвращая общую сумму значений в списке.

■ Функции генератора — в этой категории имеются функции, которые генери-руют списки. Существует несколько основных функций, таких как List.Numbers, описанных ранее в этом разделе, и List.Dates, которая возвращает последова-тельный список дат от даты начала, количество дат и длительность между датами. В этой категории также присутствует расширенная функция List.Generate, которая может формировать список из исходного состояния, она описана в последнем разделе этой главы.

Дополнительные сведения о функциях для операций со списками можно найти на сайте https://msdn.microsoft.com/query-bi/m/list-functions.

Записи

В Power Query обработка значений записей выполняется в двух распространенных сценариях: при загрузке неструктурированных данных, таких как файлы JSON, и при манипулировании данными в строках таблицы. Запись — это набор пар "ключ/значение". Ключи (их также называют именами полей) являются уникальными текстовыми значениями, а значения (их также называют значениями полей) могут принимать значения любого типа, базовые или сложные.

Инициировать новую запись можно, поместив разделенные запятыми пары между операторами открывающей и закрывающей скобок. При этом употребляется синтаксис ключ=значение. Например, следующая формула представляет запись с парами имен полей и значений: ID=123, Name="Alice" и City="Wonderland":

= [ID=123, Имя="Алиса", Город="Страна чудес"]

Пример записи в окне редактора Power Query показан на рис. 9.7.

Рис. 9.7. Запись в строке формул Power Query Editor можно инициализировать с помощью квадратных скобок []

При работе с таблицами можно перейти к просмотру записи, выбирая или указывая определенную строку с помощью фигурных скобок. Представьте, что имеется таблица сотрудников, определяемая идентификатором Employees (Сотрудники). Чет

вертая строка в таблице Employees содержит информацию об Алисе, чей идентификатор — 123, а город — Страна чудес. Получить доступ к четвертой строке можно, указав в следующем выражении значение 3 в качестве индекса отсчета "с нуля":

= Employees{3}

Если вы не уверены, в какой строке находится Алиса, можно выполнить поиск строки, имя сотрудника которой Алиса, или ID, равный 123. Далее приведены соответствующие выражения, оба возвращают одну и ту же запись (как показано на панели предварительного просмотра, рис. 9.7):

= Етр1оуее5{[ИМя="Алис:а"]}

= Employees{[ID=123]}

ПРИМЕЧАНИЕ

Извлечение определенной строки в виде записи из таблицы, как показано в предыдущих выражениях, завершится неудачей, если связанный столбец содержит несколько значений. Например, если имеются два сотрудника с именем Алиса или две строки с идентификатором 123, вы получите следующую ошибку:

Expression.Error: The key matched more than one row in the table (Ключ соответствует более чем одной строке в таблице).

Если известно, как создать конкретную запись, получить к ней доступ или найти ее, то вы пожелаете получить доступ к определенному полю в этой записи. Для этого можно заключить имя поля в скобки. Например, в предыдущей записи можно получить доступ к записи Алисы, применяя следующую формулу:

= [ID=123, Имя="Алиса", Город="Страна чудес"][Имя]

Далее приводится пример типичного выражения M, которое автоматически генерируется Power Query при применении фильтра к определенному столбцу. Точнее, для повествования о Стране чудес строки в таблице Employees (Сотрудники) фильтруются так, чтобы обнаружить Алису. Обратите внимание, что выражение ссылается на имя столбца в скобках:

= Tab1e.Se1ectRows(Emp1oyees, each ([Имя] = "Алиса"))

Данный код ссылается не на столбец в таблице, а на поле в записи. Каждая строка в таблице Employees (Сотрудники) представлена как запись при вызове функции Table.SelectRows. Ключевое слово each позволяет сократить выражение функции и сделать его понятнее. В данном случае функция получает запись в качестве аргумента и возвращает true, если имя сотрудника — Алиса. Далее в этой главе будет приведено больше информации о выражении each.

Операторы записи

Для объединения двух записей или возвращения записи с обновленным значением в поле записи можно использовать оператор &. Вот пример реализации слияния, когда возвращается запись [ID=123, Имя="АИсе", City="Wonder1and"]:

= [ID=123, Имя="Алиса"] & [Город="Страна чудес"]

Возвращение новой записи с измененным значением в имеющееся поле данной записи является одной из наиболее распространенных операций. Для реализации этого можно использовать оператор &. Следующее выражение изменяет ID Алисы с 123 на 456:

= [ID=123, Имя="Алиса"] & [ID=456]

Изменение записи, полученной в результате детализации из строки таблицы, может повлиять не на фактическую строку, а лишь на ее новую копию. Для изменения значения в таблице необходимо применить одну из функций Table.X, которые описаны в следующем разделе. Обратите внимание, что язык M не позволяет изменять имеющееся значение, независимо от его типа; как только значение оценено, можно изменять его копию только в новом выражении. Например, на первый взгляд кажется, что в следующем выражении можно изменить идентификатор Алисы от 123 до 456, но в результате возникнет ошибка, как показано на рис. 9.8:

let

Источник = [ID=123, Имя="Алиса", Город="Страна чудес"],

Источник[1С>]=456

у I Expression.SyntaxErron Ожидался токен Equal.

CL '

С

pool: let

0002: Source = [ID=123f Name=”Alice", City="Wonderland"]r

0003: Source[ID]=456

> A

0004: in

0005: Customl

Рис. 9.8. Вы не можете изменить запись, получив доступ к ее полю

Для смены ID Алисы на 456 можно применить следующее выражение:

let

Источник = [ID=123, Имя="Алиса", Город="Страна чудес"],

Result = Источник & [ID=456]

in

Result Для проверки равенства двух записей можно использовать операторы равенства (=) и неравенства (< >).

Следующее выражение, введенное в строке формул, возвращает TRUE на панели предварительного просмотра:

= [ID=123, Имя="Алиса"] = [Имя="Алиса", ID=123]

Порядок полей в записи не важен.

Функции Record.X

Возможно, вы заметили на рис. 9.7, что редактор Power Query предлагает только одну команду для преобразования записи на ленте — команду В таблицу (Into Table), которая эквивалентна функции Record.ToTable. Хотя лента не предлагает иных команд, в языке М можно найти несколько полезных встроенных функций Record.X.

Функция Record.FieldCount возвращает число полей в записи. Функция Record.HasFields возвращает true, если поля имеются, и false, если запись пуста. Обратите внимание, что пустая запись может представляться открывающей скобкой, за которой следует закрывающая скобка:

[]

Можно добавлять пары полей "ключ/значение" с помощью функции Record.AddField и обновлять значение поля в записи, возвращая новую запись, которая объединяет две записи с помощью функции Record.Combine:

= Record.Combine( { [ID=123, Имя="Алиса"], [ID=456] } )

Дополнительные сведения о функциях Record.X можно получить на сайте https:// msdn.microsoft.com/en-us/query-bi/m/ record-functions.

Табличный тип

Табличное значение — наиболее распространенный тип, с которым вы, вероятно, и работаете в Power Query. При подключении к источнику данных с помощью Power Query обычно вы попадаете в окно редактора Power Query с помощью таблицы.

Большинство преобразований, которые формируются с помощью команд ленты, транслируются во встроенные функции Table.X. Таблицы очень распространены в Power Query, и понимание их встроенных функций позволяет решать различные задачи по подготовке данных.

Как было описано в предыдущих разделах, можно получать доступ к столбцу в таблице, используя имя столбца, заключенное в квадратные скобки. Например, следующее выражение возвращает Column1 (как список) в таблице MyTable:

= MyTable[Column1]

Получить доступ к первой строке таблицы MyTable можно с помощью следующего выражения:

= MyTable{0}

Получить доступ к значению в строке X и в столбце Y из таблицы MyTable можно с помощью одного из следующих выражений:

= MyTable{X-1}[ColumnY]

= MyTable[ColumnY]{X-1}

Первое выражение обращается к X-й записи (например, 4-й записи, которая находится в строке номер 3, поскольку индекс начинается с нуля), а затем извлекает значение поля ColumnY. Второе выражение обращается к списку всех значений в столбце ColumnY, а затем извлекает X-й элемент этого списка.

Чтобы инициализировать таблицу, можно применить выражение #table, как показано на рис. 9.9, и в этом примере:

= #table({"ID", "Имя", "Город"}, {{123, "Алиса",

"Страна чудес"}, {234, "Боб", "Страна чудес"}})

В языке M имеется большое число встроенных табличных функций. Наиболее распространенные из них представлены на лентах Power Query. Познакомиться с ними можно на сайте https://msdn.microsoft.com/en-us/query-bi/m/table-functions.

Давайте рассмотрим краткий перечень табличных функций, а также сценарии, в которых они обычно встречаются:

■ Функции Table.RowCount, Table.ColumnCount и Table.IsEmpty полезны для проверки

размера и ширины таблицы, а также для обработки краевых случаев пустых таблиц. Эти функции удобно применять в сочетании с выражениями if, которые описаны в следующем разделе.

■ Table.Firstvaiue — возвращает значение в ячейке первого столбца и первой строки.

■ Table.Profile — это очень полезная функция для исследования данных, которая возвращает новую таблицу со статистическими атрибутами числовых столбцов (Min, Max, Average, Standard Deviation, Count, Null Count и Distinct Count) (минимальное, максимальное, среднее, стандартное отклонение, количество, нулевое число и различное количество).

■ Table.ColumnNames — возвращает имена столбцов таблицы в виде списка. Это чрезвычайно полезная функция, она понадобится в главе 10 для создания динамической ссылки на столбцы и предотвращения сбоев обновления при изменении столбцов.

■ Table.FromColumns, Table.FromList, Table.FromRecords и Table.FromRows — это расширенные функции, которые служат для создания новых таблиц из сложных структур данных, таких как список вложенных списков или список вложенных записей.

■ Функции Table.ToColumns, Table.ToList, Table.ToRecords и Table.ToRows преобразуют

таблицы в сложные структуры данных.

Условия и выражения if

Необходимо знать, как применять условия и выражения if в языке M для решения сложных задач, возникающих при подготовке данных. Впервые условия встретятся вам при использовании фильтров к определенным столбцам в редакторе Power Query. Если вы полагаете, что пользовательский интерфейс недостаточно гибок и плохо соответствует логике фильтрации, то можно попытать счастья, изменяя условия в строке формул.

В предыдущем разделе этой главы речь шла о логических операторах: =, <>, <, <=, >=, >, and, or и not (и, или, или нет). Логические выражения можно создавать путем применения логических операторов к поддерживаемым типам значений, а также заключая их в скобки для формирования более сложных выражений.

Обратите внимание на базовое выражение, которое было создано в редакторе Power Query для фильтрации строк, если имя Алиса содержится в столбце Имя:

= Table.SelectRows(Employees, each ([Имя] = "Алиса"))

А вот базовое выражение, применяемое для фильтрации строк, если имя Боб не находится в столбце Имя:

= Table.SelectRows(Employees, each ([Имя] <> "Боб"))

= Table.SelectRows(Employees, each not ([Имя] = "Боб"))

Второе выражение демонстрирует использование оператора not вместо знака неравенства (<>) для получения тех же результатов.

В следующей формуле логическое выражение, которое возвращает true или false, выделено полужирным шрифтом. Им довольно легко манипулировать для создания сложных условий. Например, просмотрите, каким образом можно фильтровать строки, если имя Алиса найдено в столбце Имя, а ее ID (идентификатор) равен 123 или 456:

= Table.SelectRows(Employees, each ([Имя] = "Алиса" and ([ID]=123 or [ID]=456))) Обратите внимание, что в выражении присутствует новая пара скобок для вкладывания условия [ID] = 123 and [ID] = 456. Если не использовать внутренние скобки, как показано в следующем выражении, то в конечных результатах вы сохраните других сотрудников с идентификатором 456:

= Table.SelectRows(Employees, each ([Имя] = "Алиса" and [ID]=123 or [ID]=456))

В этом смысле написание логических условий на языке М не отличается от создания подобных условий в других языках программирования.

Другой сценарий, который может побудить вас изменить M-выражения и усовершенствовать логические выражения, — это использование условных столбцов. Например, при выполнении упражнения 5.4 в главе 5 условные столбцы применялись для обеспечения дополнительного контекста, если в других столбцах имелись определенные значения. Если обнаружится, что окно диалога Условный столбец (Conditional Column) недостаточно гибко реагирует при обращении к продвинутой логике, можно управлять логикой в строке формул.

Вскоре после начала работы с условными столбцами обнаружатся выражения if, которые Power Query создает в процессе применения условных столбцов. Как только вы познакомитесь с их синтаксисом, более сложную логику будет удобнее писать прямо в окне формул окна диалога Настраиваемый столбец (Custom Column).

Выражение if-then-else

Вот синтаксис выражения if:

= if (условие) then A else B

Для облегчения просмотра кода рекомендуется разбить выражение на отдельные строки и записать его следующим образом:

= if (condition) then

A

else

B

Можно вкладывать дополнительные операторы if, как показано в следующем выражении:

= if (условие\_1) then

A

else if (условие\_2) then

B

else

C

Если вы являетесь опытным пользователем Excel, можете сделать вывод, что синтаксис языка M отличается от синтаксиса Excel. Далее приведены предыдущие выражения, написанные в виде формул Excel:

IF(ycnoeue, A, B)

IF(ycnoBue 1, A, IF(ycnoeue 2, B, C))

Выражение if внутри выражения let

Распространенная ошибка в использовании выражений if иногда возникает, если попытаться записать выражения if тем же способом, который применяется при написании условий на других языках программирования. Например, в следующем сценарии по обработке крайних случаев имеется таблица в запросе Query1, и желательно вернуть null, если запрос пуст; в противном случае необходимо вернуть первую строку Query1 (как запись). В этом коде имеется типичная ошибка, которая часто встречается на практике:

let

if (Table.IsEmpty(Query1) then

Result = null

else

Result = Query1{0}

in

Result

Это выражение не будет работать. Внутри выражения let всегда нужно указывать идентификатор, за которым следует знак равенства, и только затем записывать выражение. Правильный код должен быть таким:

let

Result =

if (Table.IsEmpty(Query1) then

null

else

Query1{0}

in

Result

Г") СОВЕТ

В окне расширенного редактора внутри выражений let нельзя начинать строку с вы-ражения if. В подобных случаях выражения if должны иметь идентификаторы, записанные после знака равенства.

Выражения if можно использовать внутри функций (описаны в следующем разделе):

(employeeName) =>

if employeeName = "Алиса" then

"Страна чудес"

else

"Реальный мир "

Если преобразования применяются к столбцам, то можно включать выражения if в выражения M, изменяя код в строке формул. Рассмотрим этот прием в общем случае. Представьте, что вы импортируете источник данных, который содержит столбец Страна, причем имеет место несоответствие регистра символов (для некоторых стран применяются строчные буквы, для других — прописные). Вы решаете

применить шаг преобразования Capitalize Each Word (Капитализировать каждое слово) в редакторе Power Query. Далее приведено выражение М для этого случая:

= Table.TransformColumns(fcTO4HUk,{{"CipaHa", Text.Proper, type text}})

Теперь обнаружится, что для некоторых стран изменения неправильны (например, сокращения US и UK были заменены на Us и Uk). Для решения этой проблемы можно изменить предыдущее выражение с помощью оператора if:

= Table.TransformColumns(fcTO4HUk,{{"Country", each if = "US" or = "UK" then else Text.Proper(\_), type text}})

ПРИМЕЧАНИЕ

Пользователи Basic Power Query, вероятно, будут предпринимать ненужные шаги для решения проблемы с написанием названий стран. Например, можно продублировать запрос. После этого можно сохранить строки US и UK в первом запросе и удалить их из второго запроса. Затем можно применить заглавные буквы во втором запросе и окончательно объединить два запроса, как это изложено в главе 3.

Другой вариант нерационального решения описанной проблемы — дублирование столбца Страна. Затем можно перейти на заглавные буквы. Далее создать условный столбец со значениями из исходного столбца Страна, если только дублированный столбец не содержит значения US или UK. В этом случае вы копируете значения из дублированного столбца для сохранения US и UK в новом столбце. Наконец, необхо-димо удалить исходный и дублированный столбец и переименовать условный столбец Страна. Несложные манипуляции в выражении M, как было показано ранее, могут сэкономить много времени и лишних шагов и в результате повысить производительность ваших запросов.

В предыдущем примере была создана пользовательская функция и с помощью выражения each уточнялся шаг преобразования, что позволило избежать использования заглавных букв в названиях определенных стран. Далее более подробно рассматривается этот синтаксис, а также содержится дополнительная информация о пользовательских функциях.

Пользовательские функции

По мере того как вы продвигаетесь этапами в рассмотрении языка M, важной вехой в обучении методам решения сложных задач с данными станет овладение пользовательскими функциями. И хотя вы уже умеете создавать сложные функции в пользовательском интерфейсе Power Query путем преобразования существующего запроса в функцию с помощью команды контекстного меню Создать функцию (Create Function) на панели Запросы (Queries), возможность создавать произвольные пользовательские функции в языке M с помощью расширенного редактора довольно полезна. Например, в главе 7 была создана пользовательская функция, выполняющая последовательность преобразований для отмены сворачивания столбцов в обобщенных таблицах. Эту функцию невозможно создать, если бы не поддерживались свободные модификации в окне расширенного редактора.

При работе с языком M помимо применения пользовательского интерфейса Соз-дать функцию (Create Function) можно создавать пользовательские функции,

используя пустой запрос или методом встраивания пользовательской функции в существующее выражение M в расширенном редакторе. Вот пример простой функции, которая увеличивает число на единицу:

(X) => X + 1

ПРИМЕЧАНИЕ

Результат X+1 является копией числа, поскольку в языке M все значения неизменны.

Внутри функции можно выполнить несколько шагов, используя выражение let. В этом искусственном примере, при вызове функции с помощью ввода X, возвращается X+2, после выполнения двух последовательных шагов по увеличению значения на единицу:

(X) => let

Step1 = X + 1,

Step2 = Step1 + 1

in

Step2

Часто, при написании подобных выражений в расширенном редакторе, обнаруживается, что Power Query изменяет выражение и обертывает код в выражение let или in. Следующее выражение, измененное Power Query, эквивалентно предыдущему: let

Источник =

(X) => let

Step1 = X + 1,

Step2 = Step1 + 1

in

Step2

in

Источник

Можно создать функцию, которая получает несколько аргументов. Вот пример функции, которая складывает два числа:

(X, Y) => X + Y

При создании пользовательской функции можно явно определять типы аргументов и тип возвращаемого значения. В следующем примере as number объявляет, что типом возвращаемого значения функции является число:

(X as number, Y as number) as number => X + Y

Конечно, пользовательские функции могут принимать различные типы аргументов. В следующем примере текст и число принимаются в качестве аргументов, а возвращаемый результат представляет объединенный текст:

(X as text, Y as number) as text => X & Text.From(Y)

Функции могут определяться с помощью необязательных аргументов. В следующем выражении Y объявляется как необязательный аргумент. Когда функция будет вызвана без необязательного аргумента, Y получит нулевое значение:

(X, optional Y) => if (Y = null) then X else X + Y

Обратите внимание, что следующая реализация возвращает null вместо X, если Y не вызывается в качестве аргумента, потому что значение выражения X+null будет null: (X, optional Y) => X + Y

Наконец, можно объявить функцию, которая не получает аргументов. Такие функции, например List.Generate, встречаются в расширенных сценариях, один из которых представлен в последнем разделе этой главы. Например, рассмотрим функцию, которая не принимает аргументов и возвращает нуль:

() => 0

Вызов функции

Функции могут определяться внутри области выражения let или в виде отдельного запроса. Можно вызывать их в виде встроенных функций, задавая имя функции, за которым следует открывающая скобка, значения аргумента и закрывающая скобка. В данном примере пользовательская функция func, определенная в первой строке внутри выражения let, вызывается в третьей строке:

let

func = (X) => X+1,

Number1 = 1000,

Result = func(Number1)

in

Result

Функции могут быть переданы в качестве аргументов других функций. Когда пользовательская функция с одним аргументом передается в качестве аргумента другой функции, можно вызывать ее в качестве аргумента, указывая имя функции без скобок и аргумента. Подобный подход показан в следующем примере.

В приведенном далее выражении показан запрос, который загружает таблицу цен из запроса PricesTable и умножает столбец значение в столбец Цена на 0,98 (скидка 2%), если цена больше 10:

let

FnGetPriceAfterDiscount =

(price) =>

if price > 10 then

price \* 0.98

else

price,

Источник = PricesTable,

DiscountedPrices = Table.TransformColumns(

Источник,

{{"Цена", FnGetPriceAfterDiscount, type number}}

)

in

DiscountedPrices

В этом выражении пользовательская функция FnGetPriceAfterDiscount вызывается внутри Table.TransformColumns. Поскольку пользовательской функции необходим один числовой аргумент, можно вызвать ее внутри Table.TransformColumns без указания скобок или точной ссылки на число в качестве аргумента. Например, функция:

= Table.TransformColumns(fcTO4HUK,{{'^eHa", FnGetPriceAfterDiscount, type number}})

эквивалентна такому синтаксису:

= ТаЬ1е.ТгапБ:Еогп1Со1ип1П5(Ис:точник,{{"Цена", each FnGetPriceAfterDiscount( ), type number}})

Выражение each

Выражение each в Power Query разработано для того, чтобы позволить вам определять простые функции в удобной для пользователя форме.

Например, следующее выражение получает число и увеличивает его на единицу:

(X) => X + 1

Аналогичная функция имеет такой вид:

(\_) => \_ + 1

Это выражение использует подчеркивание в качестве имени аргумента. Для упрощения кода приведем сокращенную версию:

each \_ + 1

В некоторых случаях можно даже не указывать подчеркивание в каждом выражении, если в аргументе имеются поля (например, таблица или запись). Рассмотрите общее выражение, которое фильтрует значения в столбце:

Tab1e.Se1ectRows(Источник, each [Price] > 5 )

Вот более длинный синтаксис функции:

Table.SelectRows( Источник, ( )=> [Price] > 5 )

Первое упрощение предыдущего выражения может быть записано с использованием each:

Tab1e.Se1ectRows(Источник, each [Price] > 5 )

Но поскольку вы ссылаетесь на поле в квадратных скобках, можно избежать подчеркивания. Таким образом, в строке формул отображается следующее выражение:

Tab1e.Se1ectRows(Источник, each [Price] > 5 )

Это выражение более удобно для восприятия, чем остальные. Полезно помнить, что это просто ярлык для обычной функции. Таким образом, в будущем можно переписать его и расширить для решения конкретных проблем с данными.

Дополнительные темы

Для полного представления об M как языке программирования недостаточно одной главы или даже целой книги. В этом языке имеется большое число "продвинутых" элементов, которые здесь не описаны. Тем не менее в последнем разделе этой главы мы затронем часть сложных тем, которые могут быть полезны, особенно если в дальнейшем вы решите освоить язык M как программист.

Обработка ошибок

Power Query располагает встроенными функциями, которые позволяют просматривать и устранять ошибки с помощью пользовательского интерфейса. Также можно заменить значения ошибок или удалить их. В дополнение к существующим пользовательским интерфейсам язык M позволяет программно обрабатывать случаи ошибок с помощью логики преобразования.

В языке M ошибка является признаком того, что вычисление выражения не может привести к значению. Если выражение приводит к ошибке, то запрос возвращает запись Expression.Error. Обычно, если не применяются явные обработчики ошибок, ваши запросы отменяются, выдавая сообщения об ошибках. Чтобы перехватить ошибки, не вызывая сбоя запроса, можно применить выражение try/else. Рассмотрим пример:

let

Result = try Источник[Со1итп1] otherwise null

in

Result

В этом выражении try/otherwise используется для того, чтобы попытаться перейти к столбцу Column1 в таблице Источник. Но если столбец Column1 недоступен в исходной таблице, можно вернуть значение null вместо того, чтобы вызывать прерывание запроса с ошибкой. Это же выражение можно переписать без части otherwise:

let

Result = try Источник[Со1итп1]

in

Result

Это выражение возвращает запись об ошибке, когда отсутствует столбец Column1. Возвращенная запись об ошибке, показанная на рис. 9.10, может обрабатываться в редакторе Power Query, как и любой шаг преобразования.

При успешной реализации выражения try в результат входит запись, в которой HasError имеет значение false, а поле Value содержит значение успешного выражения. Следующая формула демонстрирует методику программной обработки ошибки без применения ключевого слова otherwise:

let

Attempt = try McTO4HUK[Column1],

Result =

if Attempt[HasError] then

null

else

Attempt[Value]

in

Result

Данное выражение возвращает null, если Column1 отсутствует в таблице Источник, и эквивалентно выражению try/otherwise:

let

Result = try Источник[Со1итп1] otherwise null

in

Result

Рис. 9.10. Конструкцию try можно использовать для возврата записи об ошибке, когда выражение некорректно

Язык M позволяет изменить встроенное сообщение об ошибке. В следующем примере показано, как можно перезаписать значение Message записи об ошибке:

= try Источник[Со1итп1] otherwise error "Columnl is missing from AdventureWorks database"

Дополнительную информацию об обработке ошибок можно найти на сайте https://msdn.microsoft.com/en-us/query-bi/m/errors.

Отложенные и немедленные вычисления

Внутри let язык M вычисляет выражения не в последовательном порядке, а использует "отложенный" подход. Каждый идентификатор вычисляется только по необходимости. Например, в следующем выражении Step2 пропускается:

let

Step1 = 1,

Step2 = 10000000,

Step3 = Step1 + 1

in

Step3

В дополнение к выражениям let, выражения списка и записи также обрабатываются с помощью отложенного вычисления. Остальные выражения вычисляются немедленно, и результат появляется, как только выражение встречается в процессе вычисления.

Циклы

При наличии опыта программирования иногда задаются вопросом, почему в языке M отсутствуют встроенные циклы. Ответ не прост. Язык M разработан для выполнения последовательностей преобразований, и итерация уже встроена в структуру данных. Например, для замены значения в столбце не нужно писать код, который будет пересматривать каждую строку в таблице. Язык M выполняет итерацию за вас, когда применяются какие-либо функции из встроенных в таблицу.

Тем не менее иногда вам могут понадобиться явная логика и выполнение итерации последовательности для агрегирования данных. Один из наиболее распространенных сценариев — импорт данных с веб-сайтов, причем требуется номер страницы или смещение для импорта небольшой части данных. Чтобы разбивать на страницы по всему набору данных, нужно перебирать диапазон чисел.

Для выполнения нумерации страниц можно начать с пользовательской функции, которая импортирует определенную страницу по ее номеру. Например, созданная вами пользовательская функция FnLoadPageByNumber может получать постраничные результаты поиска с веб-сайта. Затем можно сформировать список чисел, как показано в этом примере:

= {1..10}

Затем вы конвертируете этот список в таблицу, как показано на рис. 9.6, выбирая функцию В таблицу (To Table) на вкладке Преобразовать (Transform) в группе вкладок Средства для записей (List Tools). Теперь, после переименования Column1 в PageNumber, преобразуем тип столбца PageNumber в Текст (Text), а затем во вкладке Добавить столбец (Add Column) можно выбрать команду Вызвать на-страиваемую функцию (Invoke Custom Function) и вызвать функцию FnLoadPageByNumber для столбца PageNumber. Наконец, можно расширить новый столбец, и все объекты таблицы страниц будут добавлены вместе.

Рекурсия

Другой способ повторить последовательность шагов преобразования — задать рекурсию. Рекурсия в языках программирования позволяет решать сложные вычислительные задачи с использованием подхода "разделяй и властвуй". Классический пример — числа Фибоначчи.

Последовательность Фибоначчи характеризуется тем, что каждое число, кроме 0 и 1, является суммой двух предыдущих чисел в последовательности. Итак, поскольку 0 + 1 = 1, следующим числом будет единица, а поскольку 1 + 1 = 2, значит, следующим числом будет двойка, затем 3 (1 + 2 = 3), затем 5 (2 + 3 = 5) затем 8 (3 + 5 = 8), затем 13 (5 + 8 = 13) и т. д.

В языке M можно реализовать рекурсию, используя знак @ для повторного вызова основной функции внутри функции, избегая ограничений внутренней области видимости. Вот пример реализации рекурсии Фибоначчи в языке M:

let

Fibonacci =

(n)=>

if (n = 0) then

0

else if (n = 1) then

1

else

@Fibonacci(n-1) + @Fibonacci(n-2)

in

Fibonacci

Однако применение рекурсии в языке M не рекомендуется. Выполнение рекурсии обременительно и потребует много памяти на любом языке программирования, поскольку требуется, чтобы стек памяти операционной системы содержал все внутренние состояния рекурсии. В результате при запуске рекурсии для больших наборов данных или при наличии большого значения для вычисления числа Фибоначчи вы быстро получите переполнение стека памяти.

Функция List.Generate

Функция List.Generate позволяет реализовывать бесконечные типы итераций, включая вложенные циклы. Эта функция генерирует список значений из начального состояния и набора функций. Вот синтаксис объявления функции List.Generate:

List.Generate(начальное значение функции, условие функции,

следующее выражение функции, необязательный селектор n, обнуляемая функция) список

Давайте рассмотрим действие этой функции на простом примере, который возвращает последовательность Фибоначчи, не превышающую 100 (т. е. {1, 2, 3, 5, 8, 13, 21, 34, 55, 89}) при введении в строку формул следующего кода:

= List.Generate(

()=>[Previous = 0, Current = 1],

each [Previous] + [Current] < 100,

each [

Current = [Previous] + [Current],

Previous = [Current]

],

each [Previous] + [Current]

)

В данном случае List.Generate начинается с начальной функции, которая не получает входных данных и возвращает числовую запись Previous (Предыдущий) и Current (Текущий). Previous присвоено значение 0, а Current — значение 1. Второй аргумент функции List.Generate, условная функция each [Previous] + [Current] < 100, гарантирует, что список генерируется, пока условие выполняется. Когда Previous и Current достигают числа, превышающего или равного 100, генерация списка завершается.

Третьим аргументом служит следующая функция, которая определяет следующее состояние и перемещает значения из полей Current и Previous в запись. Итак, в начальном состоянии значение Previous было 0, а Current — 1. На следующем этапе значение Current будет [Previous] + [Current], что составляет 1 (0+1=1), и Previous присваивается значение Current, которое было равно 1. На следующем шаге Current присваивается 2 (1+1=2), а Previous присваивается 1. Затем Current присваивается 3 (2+1=3), а Previous — 2, и т. д.

Четвертый аргумент назначает элемент для списка вывода во время каждой итерации, что приводит к последовательности Фибоначчи по завершении итерации.

Функция List.Generate позволяет решать сложные задачи и обходиться без циклов. Тем не менее для реализации логики итерации следует сначала попытаться обойтись встроенными табличными функциями, поскольку List.Generate при выполнении "отложенных" вычислений может выполняться очень медленно и потреблять больше памяти. И все же для небольших наборов данных эта функция удобна. Для просмотра других примеров использования функции List.Generate, таких как вложенные циклы, разбиения на страницы курсора и умножение матриц перейдите на сайт https://datachant.com/tag/list-generate/.

Функция List.Accumulate

Функция List.Accumulate накапливает результат из списка ввода. Начиная с начального значения, данная функция применяет функцию аккумулятор и возвращает конечный агрегированный результат. Эта функция подойдет для перебора элементов в списке и постепенного объединения результатов для получения желаемого результата.

Вот синтаксис объявления этой функции:

List.Accumulate (список как список, начальное значение любое, аккумулятор функция) любой тип

Рассмотрим различные аргументы функции в простом выражении, которое про-сматривает список инкрементных чисел и суммирует их, используя функцию List.Accumulate:

= List.Accumulate({1, 2, 3, 4}, 0, (state, current) => state + current)

Результатом выполнения этого выражения является 10. Давайте рассмотрим процесс формирования шаг за шагом. Функция начинается со списка ввода {1, 2, 3, 4} и 0 в качестве исходного состояния (аргумент seed). На первом этапе функция аккумулятора получает нулевое значение в качестве начального состояния (которое определено как второй аргумент функции List.Accumulate) и значение 1 как текущее (так как первый элемент в списке равен единице) и суммирует их. Результат, значение 1, присваивается новому состоянию.

На втором этапе функция аккумулятора получает значение 1 в качестве состояния и значение 2 в качестве текущего (поскольку второй элемент в списке равен 2) и суммирует их. Значение 3 (1 + 2 = 3) присваивается новому состоянию.

На третьем этапе функция аккумулятора получает значение 3 в качестве состояния и значение 3 в качестве текущего (поскольку третий элемент в списке равен 3) и суммирует их. Значение 6 (3 + 3 = 6) присваивается новому состоянию.

На четвертом и последнем этапе функция аккумулятора получает значение 6 в качестве состояния и значение 4 в качестве текущего (поскольку последний элемент в списке равен 4) и суммирует их. Значение 10 (4 + 6 = 10) присваивается в качестве конечного выхода. Итерации завершаются после того, как функция аккумулятора запускается на последнем элементе в списке. Последнее значение функции аккумулятора всегда выводится как результат выполнения функции List.Accumulate.

Далее приводится реализация функции List.Accumulate на примере последовательности Фибоначчи, возвращающей число 89. В отличие от List.Generate, при обращении к функции List.Accumulate нельзя определить условие динамического разрыва. Функция перебирает все элементы в списке. Вы инициализируете список числами от 0 до 9, но действительные значения списка не участвуют в подсчете:

let

Fibonacci =

List.Accumulate(

{0..9},

[PreviousNum = 0, CurrentNum = 1],

(state, current) =>

[

PreviousNum = state[CurrentNum],

CurrentNum = state[CurrentNum] + state[PreviousNum]

]

),

Result = Fibonacci[CurrentNum]

in

Result

В данном примере аргумент seed определяется как запись из двух числовых полей: PreviousNum и CurrentNum. Аккумулятор возвращает обновленную версию этой записи, передавая значение CurrentNum в PreviousNum и суммируя CurrentNum и PreviousNum в новом значении CurrentNum.

В главе 11 функция List.Accumulate применяется для выполнения итеративной заме-ны текста в столбце. Информация о дополнительных возможностях функции List.Accumulate приведена на сайте https://datachant.com/2016/06/02/power-query- list-accumulate-unleashed/.

Резюме

Изучение возможностей языка M — это путь к высокому мастерству при обработке данных. При работе с редактором Power Query, который поддерживает изменяющийся пользовательский интерфейс, вам не обязательно достигать мастерства в языке M для решения большого числа проблем с данными, с которыми вы обычно сталкиваетесь. В этой главе рассмотрены основные принципы языка М. Теперь вам будет значительно удобнее манипулировать выражениями в строке формул и в расширенном редакторе.

В главе вы узнали о шести этапах эволюции обучения языку М. Определив этап, на котором вы находитесь в данный момент, можно уточнить ваши следующие цели, а также очертить области, сосредоточившись на которых перейти к следующему этапу. Рассмотрены разные компоненты языка M, включая выражение let, а также различные типы, операторы и встроенные функции, включая числа, время, дату, продолжительность, текст, логические элементы, список, запись и таблицу.

Применение условий и запись выражений if в строке формул, в окне диалога На-страиваемый столбец (Custom Column) и в расширенном редакторе помогает рас-ширить логику ваших запросов за пределы возможностей, предоставляемых встроенным пользовательским интерфейсом. Показано, как записать последовательность логических выражений, используя операторы and, or и not и круглые скобки. Также рассмотрено, каким образом и где именно можно создавать выражения if, then и else в вашем коде.

Показано, как вызывать встроенные функции и как расширять их возможности с помощью пользовательских функций. Наконец, в данной главе вы познакомились с обработкой ошибок, рекурсией и расширенными функциями списков, которые помогут вам реализовывать циклы и выполнять перебор данных для создания сложных сценариев.

Вы будете периодически возвращаться к этой главе, поскольку с увеличением объема ваших знаний в области языка M вы сталкиваетесь и с более сложными проблемами. В главе 10 показано, каким образом облегченные модификации выражений M помогают формировать робастные и масштабируемые запросы.

Глава 10

От ловушек

к робастным запросам

Данные драгоценны, они будут работать дольше, чем сами системы.

- Тим Бернерс-Ли

В ЭТОЙ ГЛАВЕ ВЫ...

■ Рассмотрите основные факторы, способствующие формированию неудачных запросов и появлению распространенных ошибок, приводящих к сбоям обновления и появлению неправильных данных.

■ Увидите, каким образом предусмотрительность, применение лучших практик и модификаций формул M помогают избегать ловушек.

■ Изучите методики по недопущению ошибок обновления вследствие автоматического обнаружения и изменения типов столбцов.

■ Выясните, как избежать небезопасной фильтрации, которая может приводить к потере части данных в отчетах.

■ Поймете, как изменить порядок расположения и подмножеств столбцов.

■ Рассмотрите удаление и выбор столбцов, увидите, как избежать ошибок обновления в будущем.

■ Узнаете о методиках переименования столбцов по их расположению в таблице или по указанным значениям.

■ Изучите "подводные камни" процесса разделения столбца на столбцы вместо строк.

■ Узнаете, как можно улучшить формулу M при объединении нескольких столбцов.

Опыт работы с редактором Power Query Editor в Excel и Power BI чрезвычайно полезен, поскольку помогает превратить повседневные, но важные задачи по подготовке данных в процессы автоматического обновления. К сожалению, по мере продвижения в деле обработки данных можно столкнуться с некоторыми распространенными ошибками, характерными для многих пользователей при работе с редактором Power Query. Эти ошибки могут повлечь за собой формирование неудачных запросов, которые не будут обновляться при изменении данных. И еще хуже, ошибки могут привести к появлению неверных данных в отчетах. В этой главе рассказано о самых распространенных ошибках, продемонстрировано, как создавать надежные запросы, чтобы избегать подобных ошибок.

Методы, описанные в этой главе, помогут вам создавать робастные запросы и сделать большой шаг вперед в бесконечной борьбе за поддержку системы отчетности, которая будет востребована значительно дольше, чем данные.

См. также

10 ловушек, рассмотренные в этой главе, я подробно обсуждаю в своем блоге (https:// datachant.com/tag/pitfalls/). И хотя можно подробно изучить все "подводные камни", обратившись к указанному блогу, в данной главе изложены основные моменты, пред-ставлены новые примеры, упражнения и примерные файлы решений. Десятая ловушка, которая относится к удалению дубликатов, уже рассматривалась в главе 2.

Причины и следствия ловушек

В основе редактора Power Query лежат два основополагающих принципа:

■ Предоставляется простой в применении пользовательский интерфейс, который переводит шаги в инструкции программных преобразований.

■ Загружается снимок состояния данных или предварительный просмотр данных, что позволяет формировать логику преобразования.

И хотя эти принципы в основном и определяют успех при решении многих задач, связанных с подготовкой данных, они также являются виновниками ключевых ошибок, которые можно совершить непреднамеренно, что может привести к сбоям при обновлении или неожиданным ситуациям, связанным с отсутствующими или неверными данными, как показано на рис. 10.1.

Каждый шаг, исполняемый с помощью пользовательского интерфейса редактора Power Query Editor, добавляется на панель Примененные шаги (Applied Steps), выражаясь в виде автоматически сгенерированной строки формулы, которая соответствует большей последовательности шагов преобразования. При формировании шагов в редакторе Power Query ему приходится постоянно принимать "решения" о преобразовании в код выбранных элементов пользовательского интерфейса. Например, шаг Отменить сведение столбцов (Unpivot Columns), описанный в главе 6, генерируется как выражение языка M, которое служит для ссылки на невы- бранные столбцы в таблице вместо выбранных столбцов. Другой пример — импорт листа в файл Excel, тогда Power Query "разрешает" в выражении M ссылаться на лист по имени, а не по расположению листа в книге.

Решения по преобразованию шагов в код, которые принимает редактор Power Query Editor, не случайны. Они разработаны командой Power Query и помогают создавать запросы, необходимые для достижения ваших целей. И хотя подобные переводы очень удобны при создании запроса, в будущем они могут вас подвести. При создании запроса некие решения, которые редактор Power Query использует для генерации программного кода, зависят от предварительного просмотра ваших

данных. Если данные изменяются при будущем обновлении, то некоторые запросы могут быть недостаточно устойчивыми и не выдержать изменений.

В типичном случае, при изменении данных, появляется ошибка обновления, и можно без труда скорректировать запрос. Часто ошибки обновления продолжают появляться вновь и вновь, поскольку данные могут зигзагообразно перемещаться между несколькими форматами. Исправление запроса позволяет учесть определенные изменения в данных на текущий момент, но может и не устранить другие изменения в дальнейшем.

Наиболее распространенный сценарий, приводящий к сбоям обновления, возникает при изменении имени столбца в исходных таблицах. Большинство ошибок, приведенных в этой главе, связано с неправильной обработкой имен столбцов, что и служит причиной появления сбоев при обновлениях. На первый взгляд может показаться, что это не имеет большого значения: изменение запросов для ссылки на новые имена столбцов является простой задачей, которая занимает всего несколько минут вашего времени. И это действительно так. Несложно отредактировать запрос и адаптироваться к изменениям столбцов.

Если же серьезно подходить к работе с редактором Power Query, то ваши запросы могут оказаться довольно сложными. И если ваши запросы не выполняются из-за отсутствующих имен столбцов, можно потратить драгоценное время на исправление нескольких шагов, где содержатся ссылки на имена столбцов. Как вы полагаете, удобно ли быть привязанным к бесконечной задаче фиксации запросов всякий раз, когда в вашем источнике данных отсутствует имя столбца? Теперь ваши отчеты уже настолько важны, что сотни людей в вашей организации могут он них зависеть. Невозможность обновить отчет может застать вас врасплох.

Наконец, некоторые ловушки могут не приводить к ошибкам обновления при изменении данных. Вместо этого запросы могут проигнорировать часть данных, что приведет к неправильным вычислениям, искаженным диаграммам и неверным KPI в вашем отчете. Обнаружить подобные сбои не так просто, их желательно предотвращать, что и служит предметом рассмотрения в данной главе.

Прежде чем подробно описать некоторые "подводные камни" и методики их преодоления, рассмотрим несколько первоочередных принципов. Эти принципы помогают минимизировать влияние ловушек, они подразделяются на три тематики: максимальная осведомленность, лучшие практические методики и модификации кода M.

Осведомленность

Работайте над повышением осведомленности, что уменьшит вероятность сбоев обновления или появления неверных результатов в отчетах. Обязательно следуйте этим рекомендациям после каждого шага преобразования, который выполняется в Power Query Editor:

■ Исследуйте преобразованные результаты на панели предварительного просмотра. Убедитесь, что данные выглядят правильно. Прокрутите вниз для просмотра большего числа строк. Прокрутите вправо, чтобы увидеть все столбцы. Если обнаружатся странности, определите, создан ли ваш шаг правильно, или же рассмотрите альтернативный шаг преобразования.

■ Всегда отображайте строку формул и попросите коллег делать то же самое. (По умолчанию строка формул скрыта.) Для отображения строки формул перейдите на вкладку Просмотр (View) и установите флажок Строка формул (Formula bar). Отображение строки формул имеет решающее значение для предотвращения попадания в ловушки.

■ Просмотрите код в строке формул. Можно и не понимать язык M в деталях, чтобы эффективно просматривать код. Обратите особое внимание на жестко закодированные элементы в формуле, такие как имена и значения столбцов, особенно если эти элементы предоставлены в окне диалога или путем выбора элемента управления пользовательского интерфейса на соответствующем шаге. Например, если удалить столбец Column1 в запросе, то можно заметить в строке формул, что столбец на самом деле назван Column1 (с завершающим пробелом в названии). Существует большая вероятность того, что имя этого столбца будет изменено владельцем источника данных и ваш шаг завершится неудачей. Частый просмотр строки формул повысит вашу осведомленность о таких ситуациях и позволит предотвратить сбои в будущем.

Лучшие практические методики

Описанных в этой главе ловушек довольно часто можно избежать, если придерживаться нескольких хорошо зарекомендовавших себя методик:

■ Поддерживать договоренность по обмену данными с коллегами, в чьем ведении находятся внешние источники данных. Общайтесь с ними регулярно и делитесь предположениями о ваших данных. Планируйте совместную работу по созданию элементов управления, которые помогают подготовиться к ожидаемым изменениям. Убедитесь, что коллеги осведомлены о ваших отчетах, взаимодействуйте с ними, чтобы они знали, что их работа является частью вашей отчетности и влияет на успешность проекта.

■ Сохраняйте версии вашего отчета на этапе до обновления, что позволит отслеживать критические изменения. Обновленный отчет внезапно может привести к сбою или отразить неожиданные результаты. Поддерживая старые версии отчета, можно сравнить отчеты и уточнить основные причины сбоев.

■ Как показано в этой главе, некоторых ловушек можно избежать, сделав пра-вильный выбор операции (например, удаление столбца) или, наоборот, избегая выполнения некоторых ненужных преобразований (например, изменение типа или порядка столбцов).

Модификации кода M

Имеется ряд модификаций, которые можно выполнить на уровне формул для создания надежных запросов. И хотя для того, чтобы избегать ловушек, применяют уникальные модификации кода M, наиболее распространенный способ основан на функции Table.ColumnNames. Эта функция позволяет преобразовать ссылку на жестко закодированное имя столбца в динамическую ссылку, которая не дает сбоев. С подобным сценарием вы уже встречались при выполнении упражнения 3.4 в главе 3. В данном примере несколько листов добавлялись в одну таблицу и в качестве имени первого столбца выделялись контекстные данные за 2015 год. Затем столбец данных за 2015 год переименовывался на Release Year (Год выпуска). Функция Table.ColumnNames позволила переименовать первый столбец вместо переименования столбца 2015, что повысило надежность запроса и помогло избежать потенциальной ошибки, связанной с обновлением в будущем.

В данной главе показаны различные модификации M-формул, которые включают функцию Table.ColumnNames, что позволит избегать ловушек. Рассмотрим и большое число других модификаций M, которые помогут создавать робастные запросы и минимизировать сбои при изменении данных.

Ловушка 1. Игнорирование строки формул

В начале этой книги, при рассмотрении упражнений, мы много раз обращались к информации в строке формул для лучшего понимания выполняющегося преобразования или небольших изменений формул для достижения ваших целей. В главе 9 были рассмотрены основные принципы языка M, и к настоящему моменту у вас уже сформировались четкие предпочтения относительно целей применения языка M. В то время как многие опытные пользователи стремятся использовать язык M для достижения своих целей, другие рассматривают M как довольно странный язык программирования, который нелегок для понимания.

Если вы входите в первую группу пользователей, то сможете довольно легко избежать первой ловушки. Но если вы принадлежите ко второй группе и вам кажется, что язык M невозможно изучить в принципе, то для вас имеется утешительное сообщение: совершенно не обязательно досконально знать язык M, чтобы устранить до 90% проблем с данными, с которыми вы столкнетесь. Легких манипуляций с формулой М будет достаточно для достижения большинства, если не всех, ваших целей. Но если вы используете редактор Power Query для создания отчетов, которые имеют большое значение для вашей организации, тогда строка формул послужит стратегическим союзником вашего успеха.

Активизируйте строку формул. Просматривайте ее регулярно. Узнайте, как идентифицировать статические элементы, такие как имена столбцов и значения. Эти статические значения заключаются в квадратные или двойные кавычки. Посмотрите, имеют ли эти значения смысл, и придерживайтесь лучших практик, перечисленных ранее в этой главе, что позволит предотвратить большинство сбоев.

Если же вы сразу перешли к этой главе и пропустили все предыдущие главы, то строка формул активизируется следующим образом: в редакторе Power Query на вкладке Просмотр (View) установите флажок Строка формул (Formula Bar), как показано на рис. 10.2.

УПРАЖНЕНИЕ 10.1. Применение строки формул для обнаружения статических ссылок на имена столбцов

Это небольшое упражнение демонстрирует, как обнаружить статические имена столбцов в строке формул, и создает основу для устранения второй ловушки. Применяется рабочая книга C10E01.xlsx, которую можно загрузить с сайта https:/ /aka.ms/DataPwrBIPivot/downloads. В этом примере рабочей книги обобщаются доходы от заказов вымышленной компании Wide World Importers.

1. Загрузите файл рабочей книги C10E01.xslx и сохраните его в папке C:\Data\C10\.

2. Откройте пустую рабочую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data) и в окне диалога Получить данные (Get Data) вы-полните команды Файл (File) Excel.

3. Выберите файл C10E01.xlsx и щелкните мышью на кнопке Импорт (Import).

7 I Sheetl - Редактор Power Query

а

Параметры запроса | | 0 Моноширинный

|vj Показать пр обелы

Структура

Главная Преобразование Добавление столбца

Всегда разрешать

столбцу.

Предварительным про смотр дан ным Столбцы

Параметры

Расширенный

редактор

Подробнее

Зависимости

запроса

Зависимости

= Tab’le.TransformColuniinType5(#''noBb«ietitiue заголовки",{-["Pate", type date}, {"Color", type text}, {''Supplier Category", type text}, A

fSupplier Maine"j type text}, {"Customer Category", type text}, {"City", type text}, {“Random Column!", type number}, {"Random Coluiwi2", type number}, {"Revenue", type number}}-}

п. 3 Date 3 АВС Color H ABC Supplier Category □ ABC Suppl er Name t] ^C Customer Category - ASC Chy | | 1,2 Random Columnl | \* | 1,2 Random Colur

1 01.012015 Black Clothing Supplier Fabrikam, nc. Computer Store Beaver Bay 0,669669664 I

2 01.01.2015 Black Clothing Supplier Fabrikam, nc. Computer Store Caruthers 0,791225426 I

3 01.01.2015 Black Clothing Supplier Fabrikam, nc. Computer Store Con-esus Lake 0,027167715 Л

4 0101.2015 Black Clothing Supplier Fabrikam, nc. Computer Store Darling 0,905738004

ПС 01.012015 Black Clothing Supplier Fabrikam, Inc Computer Store Flowella 0,308817947 I

6 01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Kopperl 0,814972919 л

7 01.01.2015 Black Clothing Supplier Fabrikam, nc. Computer Store La Bolt 0,404748404

8 0101.2015 Black Oothing Supplier Fabrikam, nc. Computer Store Lompoc 0,534107271

9 0101.2015 Black Clothing Supplier Fabrikam, nc. Computer Store Nitro 0,945517618

10 01.01 2015 Black Clothing Supplier Fabrikam, nc. Computer Store Panton 0,252428749 -

11 0101.2015 Black Qothing Supplier Fabrikam, nc. Computer Store Pastura 0,304248267 i

12 0101.2015 Black Clothing Supplier Fabrikam, nc. Computer Store Phelps City 0,825332819 <

13 0101.2015 Black Clothing Supplier Fabrikam, nc. Computer Store Switz City 0,039420669 -

и4 01.012015 Black Clothing Supplier Fabrikam, nc. Computer Store Tolchester Beach 0^301936785

15 01.012015 Black Clothing Supplier Fabrikam, nc. Computer Store Wounded Knee 01029286036 i

16 01.01.2015 Black Clothing Supplier Fabrikam, nc. Computer Store Wyndham 0,344525327

17 01.012015 Black Clothing Supplier Fabrikam, nc. Corporate Arrowbear Lake 0Д65583033 i

И8 01.012015 Black Clothing Supplier Fabrikam, nc. Corporate Buchanan Lake Village 0,221596492 i

И9 01.012015 Black Clothing Supplier Fabrikam, nc. Corporate Capa 01574033168 IV

м т ъл-tc Hl^-t Hlnthwww Camnliar PshriUam П

СТОЛБЦОВ: 9, СТРОК: 999 +

Параметры запроса x

J СВОЙСТВА

Имя

I Sheetl

B-ce свойства

J ПРИМЕНЕННЫЕ ШАГИ

Источник

Навигация

Повышенные заголовки #

X Измененным тип I

ПРЕДВАРИТЕЛЬНЫЙ ПРОСМОТР ЗАГРУЖЕН В 21:32

Рис. 10.2. После активизации строки формул можно видеть, как шаги, подобные шагу Измененный тип, ссылаются на имена столбцов

4. В окне диалога Навигатор (Navigator) выберите лист Sheet1 и щелкните на кнопке Изменить (Edit).

5. Если строка формул не отображается, перейдите на вкладку Просмотр (View) и установите флажок Строка формул (Formula Bar).

6. Обратите внимание, что последним шагом на панели Примененные шаги (Applied Steps) является Измененный тип (Changed Type), и в строке формул появится следующая формула (рис. 10.2):

= Table.TransformColumnTypes(#"Promoted Headers",{{"Date", type date}, {"Color", type text}, {"Supplier Category", type text}, {"Supplier Name", type text}, {"Customer Category", type text}, {"City", type text}, {"Random Column1", type number}, {"Random Column2", type number}, {"Revenue", type number}})

Не вникая в смысл этого шага, можно заметить, что шаг содержит много значений. Эти значения — именно то, что следует искать в строке формул при создании собственных запросов, если желаете избежать большого числа ловушек. Во многих случаях двойные кавычки представляют собой имена столбцов или значения, которые введены в диалоговых окнах.

В этом случае можно видеть, что имеются два столбца с именами Random Column1 (Случайный столбец1) и Random Column2 (Случайный столбец2). Предположим, что в этом упражнении данные столбцы содержат бессмысленные случайные числа, и они не нужны в вашем отчете. В реальных сценариях вы встретите имена столбцов, которые могут иметь произвольный или временный контекст. Можно сохранить их в отчете или удалить, но на шаге Измененный тип становится ясно, что код уже жестко запрограммирован со ссылкой на эти столбцы. Если эти столбцы не важны, то велика вероятность, что в будущем вы не найдете их в исходной таблице, поскольку владелец внешнего источника данных может удалить их.

Следующие шаги демонстрируют ошибку обновления на этом этапе.

7. Закройте редактор Power Query и загрузите запрос на свой лист Excel или отчет Power BI.

8. На этом этапе представим, что вы — пользователь и являетесь владельцем источника данных. Откройте книгу C10E01.xlsx и удалите столбцы Random Column1 и Random Column2 из рабочего листа.

Сохраните рабочую книгу. Теперь можно вернуться к предыдущей роли и уточнить влияние этого шага.

9. Вернитесь к своему отчету и обновите его. Вы заметите очевидную ошибку обновления:

Expression.Error: The column 'Random Column1' of the table wasn't found

(Столбец 'Random Column1' таблицы не найден).

Важно отметить, что в отчете явно не указаны случайные столбцы.

Вместо этого редактор Power Query ссылается на эти столбцы. Больше сведений вы узнаете при описании ловушки 2, но сейчас рассмотрим, как можно справиться с описанным сценарием с помощью трех основных принципов, упомянутых ранее: осведомленность, лучшие практики и модификации кода на языке M.

Открыв строку формул и просмотрев содержимое в двойных кавычках на шаге 6, вы увидите, что имеет место вероятность жестко закодировать формулу с использованием второстепенных названий столбцов. Теперь можно обратиться к владельцам данных и сообщить им о существовании таких столбцов.

Затем можно заключить контракт с владельцами данных, который позволит вам получать уведомления в случае, когда они намерены в будущем удалять или переименовывать эти столбцы, причем, если эти столбцы имеют какую-либо ценность для бизнеса, можно включать их в отчеты.

10. Наконец, модифицируя формулу M, можно удалить ссылки на случайные столбцы путем удаления части формулы:

{"Random Column1", type number}, {"Random Column2", type number},

Вот надежная версия формулы, которая игнорирует случайные столбцы:

= Table.TransformColumnTypes(#"Promoted Headers",{{"Date", type date}, {"Color", type text}, {"Supplier Category", type text}, {"Supplier Name", type text}, {"Customer Category", type text}, {"City", type text}, {"Revenue", type number}})

В этой формуле имеется еще большое число столбцов, которые могут быть удалены или переименованы в будущем.

Действительно ли нужно ссылаться на все имена столбцов в этой строке? Это — суть второй ловушки.

Ловушка 2. Измененные типы

В упражнении 10.1 вы столкнулись с наиболее распространенной ошибкой в редакторе Power Query и фактором "номер один" в перечне ошибок обновления вследствие изменений имени столбца на шаге Измененный тип. Как показано на панели Примененные шаги (Applied Steps), последний шаг, Измененный тип, автоматически добавляется при загрузке листа в редактор Power Query (рис. 10.2).

Чтобы избежать второй ловушки, проверьте, отображаются ли на панели Приме-ненные шаги (Applied Steps) шаги Измененный тип, которые вы явно не создавали. Основной сценарий, в котором шаг Измененный тип подвержен ошибкам об-новления, — это загрузка текстовых файлов и таблиц в редактор Power Query Editor или же выдвижение первой строки в качестве заголовков. Любые незначительные изменения в именах столбцов или удаление столбцов в источнике данных приведут к ошибкам обновления.

Самое простое и наиболее распространенное решение, позволяющее избежать этой ловушки, состоит в удалении шага Измененный тип на панели Примененные шаги (Applied Steps), как показано на рис. 10.3.

Рис. 10.3. Удалите шаг Измененный тип, чтобы избежать второй ловушки

Уточним, в чем состоит суть проблемы. Вам необходимы правильные типы столбцов для некоторых столбцов в ваших таблицах. Без определения типов в некоторых столбцах нельзя выполнять арифметические операции, такие как Сумма (Sum) и Среднее (Average), или применять анализ времени к этим столбцам в сводных таблицах Excel или визуализации Power BI. Поэтому редактор Power Query учитывает это обстоятельство и неявно определяет типы всех столбцов в таблице, а также может их изменять. В случае отсутствия этого автоматического шага, если вы забудете явно изменить тип, то не сможете рассчитать сумму по числовому столбцу при обращении к панели Значения (Values) для сводных таблиц или к визуальным представлениям Power BI.

Вместо того чтобы разрешить автоматический шаг Измененный тип, удалите его и задайте типы вручную.

Г") СОВЕТ

"Чем позже, тем лучше" — полезный девиз для изменения типов в Power Query. Реко-мендуется изменять типы столбцов в редакторе Power Query, а не полагаться на ав-томатический шаг Измененный тип, и чем позже явно измените типы, тем лучше.

При изменении типов в качестве последнего необходимого шага вы получите следующие преимущества:

■ Производительность/время обновления — изменение типа (например, преоб-разование текста в значения даты и времени) требуют определенных вычислительных усилий механизма языка М. Если нужно уменьшить число строк, применяя фильтры по всей последовательности преобразования, можно сократить время обновления запроса, вручную применяя шаг Измененный тип после выполнения последнего шага фильтра.

■ Обработка ошибок — изменение типов столбцов может привести к ошибкам вследствие того, что в ячейках содержатся значения, которые невозможно преобразовать в новый тип. Если подобные ошибки появляются раньше в цепочке этапов преобразования, тогда проблему устранить гораздо сложнее.

См. также

Пример ошибки, которую трудно обнаружить вследствие применения шага Изменен-ный тип на раннем этапе, описывается в работе, с которой можно ознакомиться на сайте https://datachant.com/2017/01/11/10-common-mistakes-powerbipowerquery-pitfall-2/.

■ Постоянство при сохранении — изменение типа на ранних стадиях могжет не сохраняться в некоторых сценариях. Например, если добавить несколько книг из папки, то изменение типа, выполненное на уровне образца запроса, не распространяется на добавленные результаты.

Можно задаться вопросом, нельзя ли настроить редактор Power Query так, чтобы он приостанавливал автоопределение и изменение типов, что позволило бы избежать удаления шагов вручную, как показано на рис. 10.3.

Ответ на этот вопрос положителен. Но при этом необходимо настраивать программу для каждой рабочей книги или файла Power BI, но нельзя сохранить настройки для всех будущих отчетов.

Для настройки редактора Power Query Editor с целью прекращения автопроверки и изменения типов откройте окно диалога Параметры запроса (Query Options) в Excel. Для открытия этого окна выполните команды Данные (Data) Получить данные (Get Data) Параметры запроса (Query Options). В Power BI Desktop нужно открыть окно диалога Параметры (Options). Для открытия этого окна выполните команды Файл (File) Option and Settings (Параметр и настройки) Options (Параметры). Выберите команду Текущая книга (Current Workbook) в окне диалога Параметры запроса (Query Options) в Excel или команду Текущий файл (Current File) в окне диалога Параметры (Options) в Power BI Desktop и отмените выбор поля Автоматически определять типы и заголовки столбцов для неструктурированных источников (Automatically Detect Column Types and Headers for Unstructured Sources), как показано на рис. 10.4.

К сожалению, подобный флажок отсутствует в разделе Глобальные Загрузка данных (Global Data Load), поэтому для каждого нового отчета необходимо по-вторять процесс отмены выбора данного флажка.

И наконец, после удаления или предотвращения создания заданного по умолчанию шага Измененный тип (Changed Type) необходимо явно изменять типы для числовых столбцов и столбцов даты/времени, которые необходимы для анализа, или обновлять исходный шаг Измененный тип (Changed Type), как описано на шаге 10, в упражнении 10.1. Обратите внимание, что в формуле, которая присутствует в шаге 10 упражнения 10.1, имеется несколько текстовых столбцов, которые желательно удалить из шага Измененный тип, что позволит предотвратить будущие сбои при обновлениях, если данные столбцы будут переименованы. Эти столбцы выделены полужирным шрифтом в следующей формуле:

= Table.TransformColumnTypes(#"Promoted Headers",{{"Date", type date}, {"Color", type text}, {"Supplier Category", type text}, {"Supplier Name", type text}, {"Customer Category", type text}, {"City", type text}, {"Revenue", type number}})

Рис. 10.4. Отключите автоматическую проверку и изменение типов

Для усовершенствования запроса можно либо удалить шаг Измененный тип (Changed Type) на панели Примененные шаги (Applied Steps) и вручную изменять типы столбцов Date (Дата) и Revenue (Доход), либо обновить формулу M следую-щим образом:

= Table.TransformColumnTypes(#"noebweHHbie заголовки",{{"Date", type date}, {"Revenue", type number}})

СОВЕТ

Если выполняется обработка таблиц, которые включают слишком большое число столбцов для выполнения явного преобразования типов и возникает необходимость в автоматическом способе определения типов столбцов без ссылки на все имена столбцов, удобное выражение M можно найти в следующей статье: http://datachant.com/ 2018/05/14/automatic-detection-of-column-types-in-powerquery.

Ловушка 3. Небезопасная фильтрация

Третья ловушка — из разряда наиболее опасных. Она почти незаметна, можно легко пропустить ее при создании запроса и игнорировать ее влияние при обновлении. Третий недостаток может привести к смещениям в ваших отчетах, а когда вы их обнаружите, будет уже слишком поздно: неверно составленный отчет приведет в итоге к неудовлетворительным деловым решениям. Все начинается с элемента управления фильтром в редакторе Power Query и производного шага — фильтрации. Это настолько тривиальный и распространенный шаг, что его включают большинство запросов.

УПРАЖНЕНИЕ 10.2. Часть 1. Фильтрация товаров, окрашенных в черный цвет

Прежде чем действительно изучить риски, связанные с третьей ловушкой, рассмотрим базовый сценарий, который демонстрирует ошибку фильтрации, а также способ, как ее избежать. Возьмем те же примеры данных, что и в упражнении 10.1. Как главного аналитика компании Wide World Importers, вас попросили проанализировать влияние на бизнес прекращения импорта товаров, окрашенных в черный цвет. Поэтому необходимо отфильтровать все запросы и выбрать товары, окрашенные в черный цвет.

1. Откройте новую рабочую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните последовательность команд По-лучить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data), затем в окне диалога Получить данные (Get Data) выполните команды Файл (File) Excel.

2. Выберите файл C10E01.xlsx и щелкните на кнопке Импорт (Import).

3. В окне диалога Навигатор (Navigator) выберите вкладку Sheet1 и щелкните на кнопке Изменить (Edit).

4. На панели Примененные шаги (Applied Steps) удалите шаг Измененный тип (Changed Type).

5. Удостоверьтесь, что строка формул видима.

6. Измените тип столбца Date на Дата (Date), а тип столбца Revenue — на Деся-тичное число (Decimal Number).

(С этого момента, следуя шагам 4-6, можно попасть в ловушки 1 и 2.)

7. Поскольку данные содержат намного больше цветов, чем Black (Черный) и Blue (Синий), отфильтруйте все товары, окрашенные в черный цвет, щелкнув на элементе управления фильтра в столбце Color (Цвет) и отменив выбор флажка Black (Черный), как показано на рис. 10.5.

После выполнения этого шага фильтрации можно ожидать, что в качестве выходных данных будут применяться все цвета, кроме черного (Black).

8. Проверьте полученный код в строке формул и в особенности обратите внимание на следующий фрагмент:

= Table.SelectRows(#"Changed Type", each ([Color] = "Blue"))

К счастью, вы знаете о первой ловушке и теперь чаще проверяете строку формул и привыкли обращать пристальное внимание на значения в двойных кавычках. Итак, проблема очевидна, и вы легко заметите ее на данном этапе. Отмена выбора флажка Black приводит к тому, что редактор Power Query неправильно запрашивает об установке флажка Blue. В результате отфильтрованы только товары синего цвета. Если применяются результаты этого запроса без проверки строки формул, может случиться, что в анализе будут присутствовать только товары синего цвета, а не все товары, цвет которых не является черным.

Исправить формулу на этом этапе легко. Необходимо изменить знак равенства в выражении Blue (= "Blue") на знак неравенства перед Black (<> "Black"):

= Table.SelectRows(#"Changed Type", each ([Color] <> "Black"))

Также можно изменить неправильное условие, не добавляя эту модификацию формулы M, а выполняя шаги 9 и 10.

9. На панели Примененные шаги (Applied Steps) удалите шаг Filtered Rows (Фильтрованные строки) и выберите элемент управления фильтром для столбца Color (Цвет). Теперь применим фильтр снова, но уже с умом.

На панели Фильтр (Filter) выберите команду Текстовые фильтры (Text Filters), а затем задайте команду Не равно (Does Not Equal). После открытия окна диалога Фильтрация строк (Filter Rows) выберите опцию Не содержит (Does Not Equal), затем укажите опцию Black (Черный) и щелкните мышью на кнопке OK для закрытия этого окна диалога.

СОВЕТ

Если вас не устраивают условия в формуле M, как описано на шаге 8, можно исполь-зовать окно диалога Текстовые фильтры (Text Filters) и задать условие фильтрации, как описано в шагах 9 и 10, что позволит вообще избежать третьей ловушки. Всегда лучше использовать текстовые фильтры, чем выбирать значения в панели ввода тек-

ста.

Файлы решения C10E02 - Solution.xlsx и C10E02 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Логика, определяющая условия фильтрации

На этом этапе следует перестать доверять обоснованию фильтрации редактора Power Query и беспокоиться о том, что третья ловушка уже "сработала" во многих существующих запросах. Часто ли вы применяли элемент управления фильтром так, как показано в предыдущем примере? К счастью, в большинстве случаев при отмене выбора определенных значений на панели Фильтр (Filter) условие формируется ожидаемым образом, и ловушка, продемонстрированная в упражнении 10.2, предотвращается.

Вот два правила, которым следует Power Query при автоматической генерации условия фильтрации при выборе значений на панели Фильтр (Filter):

■ При попытке отфильтровать значения на панели Фильтр (Filter), если число невыбранных значений на панели Фильтр равно числу выбранных значений, редактор Power Query создает "позитивное" условие, применяя знак равенства для выбранных значений.

■ Если число значений, выбранных на панели Фильтр, превышает число невы- бранных значений, то редактор Power Query всегда поддерживает "отрицательное" условие, используя знак неравенства для невыбранных элементов.

Рассмотрим эти правила на простом примере. Пусть имеется столбец, содержащий значения от 1 до 6:

1. Создайте новый файл Excel или отчет Power BI и откройте редактор Power Query с пустым запросом.

2. В строке формул введите следующее выражение:

= {1..6}

3. В группе вкладок Средства для списков (List Tools) выберите вкладку Преоб-разование (Transform), щелкните мышью на кнопке В таблицу (To Table), затем щелкните на кнопке OK для закрытия окна диалога В таблицу (To Table).

4. Выберите элемент управления фильтром для столбца Column1. Откроется па-нель Фильтр (Filter).

5. Отмените установку флажка для значения 1 и закройте панель Фильтр (Filter). Обратите внимание, что значение 1 корректно отфильтровано из таблицы. Также заметьте, что строка формул содержит следующую формулу:

= Table.SelectRows(#"Converted to Table", each ([Column1] <> 1))

6. Щелкните снова мышью на панели Фильтр (Filter), на этот раз отмените выбор флажков для значений 1 и 2. Результаты будут ожидаемыми. Обратите внимание на следующую формулу в строке формул:

= Table.SelectRows(#"Converted to Table", each ([Column1] <> 1 and [Column1] <> 2))

7. Щелкните мышью на панели Фильтр (Filter) снова. На этот раз отмените выбор значений 1, 2 и 3. Результаты выглядят отлично. На панели предварительного просмотра отображаются только значения 4, 5 и 6. Но строка формул содержит "позитивное" условие (т. е. условие, в котором используется знак равенства для значений, выбор которых не отменен на панели Фильтр (Filter)):

= Table.SelectRows(#"Converted to Table", each ([Column1] = 4 or [Column1] =

5 or [Column1] = 6))

Тем не менее корректное условие в данном случае — это "отрицательное" условие (т. е. условие, в котором используется знак неравенства для значений, выбор которых отменен на панели Фильтр (Filter)), и оно должно выглядеть следующим образом:

= Table.SelectRows(#"Converted to Table", each ([Column1] <> 1 and [Column1] <> 2 and [Column1] <> 3))

Почему выбор позитивного условия небезопасен, если отменен выбор флажков, соответствующих значениям 1, 2 и 3? Представьте, что в ваших данных в будущем также будут значения 7, 8, 9 и 10, а не только 1...6. На шаге 7 необходимо отфильтровать значения 1, 2 и 3 и оставить все остальные. При обновлении отчета в будущем значения 7, 8, 9 и 10 будут изъяты из него в случае применения позитивного условия.

Этот пример прост, но он дает вам представление о том, что может произойти, если вы работаете с большими наборами данных и сталкиваетесь с третьей ловушкой: тогда только часть значений загружается на панель. Например, в упражнении 10.2 в наборе данных имелось более двух цветов, но на панели Фильтр (Filter) загружались только значения черного (Black) и синего (Blue) цветов. Отмена выбора значения цвета Black ("Черный") обусловила равное количество выбранных и невыбран- ных значений, что затем привело к положительному условию фильтрации.

Один из наиболее опасных и распространенных факторов, которые приводят к третьей ловушке, — использование поля поиска на панели Filter (Фильтр). Обычно поле поиска применяется при наличии длинного списка значений на панели Фильтр, причем желательно увеличить количество нужных значений вместо прокрутки вниз к результатам. Отмена выбора из найденных значений всегда приводит к положительному условию, что может привести к неприятностям.

УПРАЖНЕНИЕ 10.2. Часть 2.

Поиск значений на панели Фильтр

Найдите пример рабочей книги C10E01.xlsx, используемой при выполнении упражнений 10.1 и 10.2, часть 1. Эту рабочую книгу можно загрузить на сайте https://aka.ms/DataPwrBIPivot/downloads.

1. Если вы пропустили упражнения этой главы, выполненные ранее, загрузите файл рабочей книги C10E01.xslx и сохраните его в папке C:\Data\C10\.

2. Откройте новую пустую рабочую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data) и в появившемся окне диалога Получить данные (Get Data) выполните команды Файл (File) Excel.

3. Выберите файл C10E01.xlsx и щелкните мышью на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите вкладку Sheet1 и щелкните на кнопке Изменить (Edit).

5. Чтобы избежать второй ловушки, удалите шаг Changed Type (Измененный тип) из панели Примененные шаги (Applied Steps) и измените тип столбца Date (Дата) на Дата (Date), а тип столбца Revenue — на Десятичное число (Decimal Number).

Представьте, что вы хотите отфильтровать все строки, относящиеся к городу Baldwin City.

6. Выберите элемент управления фильтром для столбца City (Город). После этого откроется панель Фильтр (Filter). Поскольку список городов, начинающийся с буквы "А", довольно длинный и необходимо прокрутить его вниз, чтобы найти города, начинающиеся с буквы "В", можно воспользоваться полем поиска на панели Фильтр (Filter). Рассмотрим все названия городов, начинающихся с Ba, чтобы найти значение Baldwin City.

Введите префикс Ba в поле поиска. В результате получим несколько городов на панели, и, к счастью, Baldwin City оказывается первым результатом, поэтому можно легко отменить выбор. К сожалению, при просмотре строки формул оказывается, что полученное выражение содержит неверное положительное условие:

= Table.SelectRows(#"Changed Type", each ([City] = "Baraboo" or [City] = "Bayou Cane" or [City] = "Bazemore" or [City] = "Beaver Bay" or [City] = "Bombay Beach" or [City] = "Greenback" or [City] = "Wilkes-Barre"))

В результате вы попадете в третью ловушку, поскольку в ваш отчет будут включены все города, содержащие подстроку Ba, за исключением Baldwin City, а все другие города (и многие из них не начинаются с Ba) будут отфильтрованы.

7. Для устранения этой ошибки нужно изменить формулу:

= Table.SelectRows(#"Changed Type", each ([City] <> "Baldwin City"))

Как можно заметить, шаги, выполненные в этом упражнении, являются распро-страненными, поэтому попасть в третью ловушку можно довольно часто. Лучшая защита — это применение окна диалога Текстовые фильтры (Text Filters) вместо поля поиска на панели Фильтр (Filter). Но если вы все-таки пользуетесь окном поиска, проверяйте содержимое строки формул и исправляйте формулы при необходимости.

Ловушка 4. Переупорядочение столбцов

В четвертую ловушку можно попасть при изменении порядка расположения столбцов в окне редактора Power Query. Во многих случаях речь идет не о точном порядке ваших столбцов, а о желании применить определенный порядок к небольшому подмножеству столбцов. Например, можно переключиться между двумя столбцами или переместить новый добавленный настраиваемый столбец из правого края таблицы в определенную позицию.

При выполнении шага переупорядочения генерируется функция Table.ReorderColumns со ссылкой на все имена столбцов в вашей таблице. Ссылаясь на все столбцы, вы тем самым ослабляете запрос и увеличиваете вероятность ошибок обновления в будущем, если столбцы будут переименовываться или удаляться из исходных данных.

Во многих случаях этап переупорядочения может быть вообще удален из панели Примененные шаги (Applied Steps). Например, при создании настраиваемого столбца, который применяет некоторые вычисления к другому столбцу в таблице, можно размещать новый столбец рядом с исходным столбцом, чтобы проверить правильность кода. В таких случаях после подтверждения правильности кода рекомендуется удалить шаг переупорядочения, что позволит уменьшить вероятность сбоев при обновлении.

В некоторых случаях изменение порядка столбцов таблицы довольно важно. Если необходимо контролировать порядок полей, которые будут отображаться в сводной таблице в Excel, или если необходимо загрузить запрос на лист, можно проконтролировать порядок столбцов. Имеется также несколько сложных сценариев, которые требуют определенного порядка столбцов в запросе. Например, в упражнении 4.4 из главы 4 применялся шаг переупорядочения для перемещения вычисляемого столбца в начало таблицы, что позволило транспонировать его, а затем использовать в качестве заголовков.

Если шаг переупорядочения имеет решающее значение, попробуйте применить его после того, как в отчете останутся только те столбцы, которые действительно необходимы. (Дополнительные сведения о выборе столбцов будут рассмотрены при исследовании пятой ловушки.) Не имеет смысла загружать таблицу с сотнями столбцов, изменять их порядок, а затем сохранять нужные столбцы. При сохранении только дюжины (или около того) необходимых столбцов изменение их порядка гарантирует, что сформируется более робастный запрос. В упражнении 10.3 будет показано, как изменить порядок подмножеств столбцов без ссылки на полные имена столбцов, что, несомненно, повысит надежность вашего запроса.

УПРАЖНЕНИЕ 10.3. Часть 1.

Переупорядочение подмножества столбцов

Начнем это упражнение с набора данных о доходах компании Wide World из упражнения 10.1 и выполним базовое переупорядочение столбцов City (Город) и Revenue (Доход).

1. Откройте новую рабочую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data) и в окне диалога Получить данные (Get Data) вы-полните команды Файл (File) Excel.

2. Выберите файл C10E01.xlsx и щелкните мышью на кнопке Импорт (Import).

3. В окне диалога Навигатор (Navigator) выберите вкладку Sheet1 и щелкните мышью на кнопке Изменить (Edit).

4. Удалите шаг Changed Type (Измененный тип) с панели Примененные шаги (Applied Steps).

5. Переместите столбец City (Город) на место второго столбца, Revenue (Доход) — на место третьего столбца, как показано на рис. 10.6.

Рис. 10.6. Переупорядочение столбцов City и Revenue в окне Power Query

6. Просмотрите строку формул, в которой будет отображен следующий код:

= Table.ReorderColumns(#"Promoted Headers",{"Date", "City", "Revenue", "Color", "Supplier Category", "Supplier Name", "Customer Category", "Random Column1", "Random Column2"})

Конечно, данная формула не будет обновляться, если имена столбцов, такие как "Random Column1" (Случайный столбец1) и "Random Column2" (Случайный столбец2), изменятся в исходной таблице. Можно ли переписать эту формулу, чтобы ссылки выполнялись на приведенный выше список имен столбцов для шага переупорядочения, причем следует упомянуть в коде только "City" (Город) и "Revenue" (Доход)? Да, конечно.

7. Удалите из предыдущей формулы фрагмент:

{"Date", "City", "Revenue", "Color", "Supplier Category", "Supplier Name", "Customer Category", "Random Column1", "Random Column2"}

и замените его следующим кодом, который возвращает тот же список имен столбцов:

List.InsertRange(List.Difference(Table.ColumnNames(#"Promoted Headers"), {"City", "Revenue"}), 1, {"City", "Revenue"})

В итоге получим окончательную формулу:

= Table.ReorderColumns (#"Promoted Headers", List.InsertRange (List.Difference( Table.ColumnNames(#"Promoted Headers"), {"City", "Revenue"}), 1, {"City", "Revenue"}))

Рассмотрим, каким образом с помощью комбинации функций List.InsertRange, List.Difference и Table.ColumnNames можно достичь цели и сгенерировать аналогичный список имен столбцов, ссылаясь только на столбцы City и Revenue.

Функция List.InsertRange получает список в качестве входных данных и вставляет другой список с определенным смещением от нуля. Итак, если имеется список А и нужно добавить поля City и Revenue в качестве второго и третьего пунктов списка A, можно применить следующую формулу:

List.InsertRange(A, 1, {"City", "Revenue"}

Как можно видеть в предыдущей формуле для функции Table.ReorderColumns, функция List.InsertRange применяется так, как описано ранее. Теперь, если список А включит все имена столбцов, кроме City и Revenue, можно применить функцию List.InsertRange для списка A и получить желаемый порядок. В этом и заключается назначение функции List.Difference. Эта функция воспринимает один список в качестве первого аргумента, а другой список — в качестве второго аргумента и возвращает новый список со всеми элементами в первом списке, которых нет во втором списке (левостороннее объединение двух списков). Итак, если функция List.Difference применяется для списка и при этом подмножество списка служит вторым аргументом, то вы получите в результате новый список, который содержит все элементы из исходного списка, за исключением всех элементов в подмножестве.

Следовательно, если B — список имен столбцов, то следующая функция вернет все названия столбцов, кроме City и Revenue:

A = List.Difference(B, {"City", "Revenue"})

Теперь при указании функции Table.ColumnNames в следующей формуле вместо B можно собрать все части для формирования полного выражения:

B = Table.ColumnNames(#"Promoted Headers")

И наконец, при объединении всех элементов вместе получим окончательную формулу (представлена в данном случае в многострочном формате для удобства просмотра):

= Table.ReorderColumns(

#"Promoted Headers",

List.InsertRange(

List.Difference(

Table.ColumnNames(#"Promoted Headers"),

{"City", "Revenue"}

),

1,

{"City", "Revenue"}

)

)

УПРАЖНЕНИЕ 10.3. Часть 2.

Пользовательская функция FnReorderSubsetOfColumns Методика, рассмотренная в первой части упражнения, полезна во многих отчетах, ее можно использовать неоднократно. Можно упростить работу с помощью пользовательской функции, которая реализует логику, примененную на шаге 7. Для иллюстрации этого продолжите упражнение, выполняя следующие шаги:

8. Создайте пустой запрос и вставьте следующий код в окне расширенного редактора:

(tbl as table, reorderedColumns as list, offset as number)

as table =>

Table.ReorderColumns(

tbl,

List.InsertRange(

List.Difference(

Table.ColumnNames(tbl), reorderedColumns

),

offset,

reorderedColumns

)

)

9. Переименуйте пользовательскую функцию FnReorderSubsetOfColumns.

10. Удалите шаг Reordered Columns (Переупорядоченные столбцы) на панели Примененные столбцы (Applied Steps) и щелкните на значке fx в строке фор-мул для создания нового шага. Затем введите следующую формулу в строке формул:

= FnReorderSubsetOfColumns(#"Promoted Headers", {"City", "Revenue"}, 1)

Результаты будут аналогичны тем, что получены после выполнения шага 5. Можно заметить, что вызов функции FnReorderSubsetOfColumns выполняется очень просто. Функция получает таблицу, подмножество переупорядоченных имен столбцов в виде списка и индекс, начинающийся с нуля, затем выполняется переупорядочение очень надежным способом, без ссылок на имена других столбцов.

См. также

Имеются и другие способы реализации функции Table.ReorderColumns, которые позво-ляют избежать ошибок обновления. Можно задать третий аргумент, который позволит игнорировать пропущенные поля или добавлять нулевые значения для пропущенных полей. И хотя эти методы предотвращают сбои обновления, они могут приводить к неожиданным результатам. Подробнее с этими вопросами можно ознакомиться на сайте https://datachant.com/2017/01/18/power-bi-pitfall-4/.

Файлы решения C10E03 - Solution.xlsx и C10E03 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads .

Ловушка 5. Удаление и выделение столбцов

Пятая ловушка связана с довольно распространенной операцией в редакторе Power Query Editor: удалением столбцов.

И хотя удаление ненужных столбцов является важным моментом при формировании эффективных отчетов (меньшее число столбцов приводит к сокращению объема памяти и размера файла), при удалении столбца вы ухудшаете запрос, подвергая его риску будущих сбоев при обновлениях.

Всякий раз, удаляя столбец в окне редактора Power Query, вы рискуете, что в будущем обновление может завершиться ошибкой, если удаленный столбец будет отсутствовать во внешнем источнике данных. Имеется ли в вашем распоряжении хороший контракт на обмен данными с владельцем исходной таблицы? Учтите, что по тем же причинам, по которым вы решили удалить определенные столбцы, владелец исходной таблицы может сделать то же самое в будущем и определить, что эти столбцы не важны.

Чтобы снизить риск сбоев при обновлениях, можно следовать простой рекомендации: сосредоточьтесь на столбцах, которые необходимо сохранить, а не на тех, которые следует удалить. Редактор Power Query позволяет удалять или сохранять столбцы. И хотя более естественно нажать на кнопку Удалить (Delete) для выбранных столбцов, которые желаете удалить, во многих случаях предпочтительнее выбирать столбцы, которые желательно сохранить. Упражнение 10.4 демонстрирует этот процесс для столбцов Random Column1 и Random Column2 в наборе данных Wide World Importers из упражнения 10.1.

УПРАЖНЕНИЕ 10.4. Обработка случайных столбцов

в таблице Wide World Importers

1. Откройте пустую рабочую книгу или новый отчет Power BI.

В Excel: На вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data), и в окне диалога Получить данные (Get Data) выполните команды Файл (File) Excel.

2. Выберите файл C10E01.xlsx и щелкните мышью на кнопке Импорт (Import).

3. В окне диалога Навигатор (Navigator) выберите вкладку Sheet1 и щелкните на кнопке Изменить (Edit).

4. На панели Примененные шаги (Applied Steps) удалите шаг Changed Type (Из-мененный тип).

5. Удалите два случайных столбца из упражнения 10.1, выбирая Random Column1 и Random Column2 и нажимая кнопку Удалить (Delete).

6. Обратите внимание, что строка формул включает следующий код:

= Table.RemoveColumns(#"Promoted Headers",{"Random Column1", "Random Column2"})

Теперь, если исходная таблица больше не содержит один из этих столбцов, то обновление не будет выполняться. На следующем шаге вы удалите случайные столбцы другим способом и повысите робастность запроса.

7. Удалите последний шаг на панели Примененные шаги (Applied Steps), затем выберите команду Выбрать столбцы (Choose Columns) на вкладке Главная (Home) и отмените выделение столбцов Random Column1 и Random Column2. Закройте окно диалога и обратите внимание на следующую формулу в строке формул:

= Table.SelectColumns(#"Promoted Headers",{"Date", "Color", "Supplier Category",

"Supplier Name", "Customer Category", "City", "Revenue"})

Игнорирование пропущенного столбца

В некоторых ситуациях необходимо сохранить очень большое число столбцов, а риск удаления нескольких столбцов может быть ниже, чем проблемы, связанные с указанием столь большого числа столбцов.

Довольно часто не имеет значения, удаляются или выбираются столбцы. В конечном итоге вы имеете дело с внешними источниками данных, которые могут измениться. Для предотвращения ошибок обновления существует необязательный аргумент, который можно задать в функциях Table.RemoveColumns и Table.SelectColumns, что позволит игнорировать ошибки вместо того, чтобы прекращать обновление.

Третьим аргументом служит MissingField.Ignore или MissingField.UseNull.

Аргумент MissingField.Ignore игнорирует пропущенный столбец, MissingField.UseNull сохраняет имя столбца при ошибках, но заполняет его нулями. Аргумент MissingField.UseNull более практичен, нежели его "побратим" MissingField.Ignore в со-единении с Table.SelectColumns, поскольку гарантирует, что выбранные имена столб-цов будут включены в конечные результаты. Однако оба варианта могут привести к ошибкам, которые трудно обнаружить. Таким образом, ошибка обновления может оказаться предпочтительнее неожиданных результатов, которые вы получите, задав эти аргументы.

Выделение или удаление столбцов

на основе их расположения

Во многих сценариях удаление или выбор столбцов на основе их расположения является более точным, чем обращение к ним по имени. Используя функцию Table.ColumnNames для получения списка всех имен столбцов и функцию List.Range для формирования подмножества столбцов, можно выбрать любое подмножество столбцов в зависимости от их расположения.

Каждая из следующих формул удаляет первый столбец в таблице:

= Table.RemoveColumns(Source, List.First(Table.ColumnNames(Source)))

= Table.RemoveColumns(Source, Table.ColumnNames(Source){0})

Указав функцию List.FirstN в данной формуле, получите удаление первых двух столбцов таблицы:

= Table.RemoveColumns(Source, List.FirstN(Table.ColumnNames(Source), 2))

Следующая формула удаляет последний столбец таблицы:

= Table.RemoveColumns(Source, List.Last(Table.ColumnNames(Source), 1))

А эта формула сохраняет имена второго и третьего столбцов в таблице:

= Table.SelectColumns(Source, List.Range(Table.ColumnNames(Source), 1, 2))

Функция List.Range получает список в качестве первого аргумента, смещение от нуля и количество возвращаемых элементов.

Можно применить код List.Range(Table.ColumnNames(Source), 1, 2) для возвращения имен двух столбцов в исходной таблице со смещением 1, которое соответствует второй позиции в списке.

Также можно выбрать отдельные столбцы. Следующая формула эквивалентна приведенной ранее:

= Table.SelectColumns(Source, {Table.ColumnNames(Source){0}, Table.ColumnNames(Source){1}})

Данный код полезен, если необходимо выбирать несмежные столбцы. В случае таблицы Wide World Importers эта формула приведет к удалению случайных столбцов, предполагая, что они всегда являются седьмым и восьмым столбцами (смещения равны 6 и 7):

= Table.RemoveColumns(#"Promoted Headers", Table.ColumnNames(#"Promoted Headers"){6}, Table.ColumnNames(#"Promoted Headers"){7}})

Выделение или удаление столбцов на основе их имен

Существует бесчисленное множество возможностей выбора или удаления столбцов в языке M. Приведем два примера, иллюстрирующих удаление случайных столбцов общим способом. Следующая формула применяет функцию List.Select к именам столбцов, что позволит удалить столбцы, содержащие подстроку Random:

= Table.RemoveColumns(#"Promoted Headers", List.Select(Table.ColumnNames(

#"Promoted Headers"), each Text.Contains(\_, "Random")))

Аналогичный результат можно получить с помощью функции Table.SelectColumns и негативной логики (т. е. можно выбрать все столбцы, которые не содержат "Random"):

= Table.SelectColumns(#"Promoted Headers", List.Select (Table.ColumnNames(#"Promoted Headers"), each not Text.Contains(\_, "Random")))

Файлы решения C10E04 - Solution.xlsx и C10E04 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads .

Ловушка 6. Переименование столбцов

Переименование столбцов — еще один довольно распространенный этап при подготовке данных. Столбцы принято переименовывать довольно часто, что позволяет улучшать взаимодействие с пользователем и предоставлять удобные имена в отчетах. Всякий раз при переименовании столбца в окне редактора Power Query запрос подвергается опасности — возрастает вероятность сбоев обновления в будущем. Следуя методам, описанным в предыдущем разделе в этой главе, можно повысить надежность запроса, изменяя формулу и избегая ссылок на имена текущих столбцов.

Рассмотрим этот вопрос на примере набора данных Wide World Importers и исследуем различные способы переименования столбцов. Предположим, что в будущем все имена столбцов, начинающиеся с префикса Random Column, будут переименованы в исходной таблице. Допустим, вас попросили переименовать эти столбцы в Factor 1, Factor 2 и т. д., как показано на рис. 10.7.

две

123 CrtV □ 1 ji Factor 1 ’ 123 Factor 2 ’ th Factor 3 - th Factor 4 V " th Factors 1 ► ’

1 Beaver Bay 0.752523351 0.494369688 0.699779851 0.S3755J933 0.912590286

2 Caiuth&rs 0.921114323 0.384524761 0,247184147 0.397545311 0585212042

3 Conesus lake 0.002781273 0308639648 0134198089 0.466973202 0652912272

4 Darling 0.655667915 0 077581958 0056702854 0.909697481 0806891503

5 Flowella 0163006662 0 761110523 0376981389 0.037078548 0667297386

Рис. 10.7. Проблема заключается в том, чтобы переименовать столбцы Random без ссылок на их имена

УПРАЖНЕНИЕ 10.5. Переименование столбцов Random в таблице Wide World Importers

Для этого упражнения необходимо взять файл рабочей книги C10E05.xlsx, который можно загрузить на сайте https://aka.ms/DataPwrBIPivot/downloads. В примере рабочей книги суммируются вымышленные доходы компании Wide World Importers.

1. Загрузите рабочую книгу C10E05.xslx и сохраните ее в папке C:\Data\C10\.

2. Откройте новую пустую рабочую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из файла (From Workbook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data) и в окне диалога Получить данные (Get Data) вы-полните команды Файл (File) Excel.

3. Выберите файл C10E05.xlsx и щелкните мышью на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите вкладку Sheet1 и щелкните на кнопке Изменить (Edit).

5. На панели Примененные шаги (Applied Steps) удалите шаг Changed Type (Из-мененный тип). Обратите внимание на панель предварительного просмотра в окне редактора Power Query, которая включает семь столбцов Random. В этом упражнении рассмотрены несколько способов переименования этих столбцов.

В ручном режиме переименуйте столбцы Random Column1 и Random Column2 на Factor 1 и Factor 2 соответственно. Строка формул теперь содержит сле-дующий код:

= Table.RenameColumns(#"Promoted Headers",{{"Random Column1", "Factor 1"}, {"Random Column2", "Factor 2"}})

Если исходная таблица больше не включает один из этих столбцов, то обновление завершится неудачей.

Представьте, что владелец данных исходной таблицы уведомит вас о том, что он планирует в будущем переименовать эти столбцы, но с гарантией, что порядок столбцов сохранится. Тогда можно будет ссылаться на эти столбцы по их расположению. Например, вот так можно переименовать первые два случайных столбца:

= Table.RenameColumns(#"Promoted Headers",{{Table.ColumnNames (#"Promoted Headers"){6}, "Factor 1"}, {Table.ColumnNames(#"Promoted Headers"){7}, "Factor 2"}})

Настраиваемая функция FnRenameColumnsByIndices

Для того чтобы переименовать столбцы на основе их расположения в таблице, можно написать настраиваемую функцию, позволяющую переименовывать большое подмножество имен столбцов без необходимости отдельно записывать пары для старых и новых имен столбцов. В этом разделе показано, как создать подобную функцию.

Например, пусть функция уже подготовлена и называется FnRenameColumnsByIndices. Эту функцию можно вызывать для переименования случайных столбцов на шаге 5 на Factor 1 и Factor 2:

= FnRenameColumnsByIndices(#"Promoted Headers", {"Factor 1", "Factor 2"}, {6, 7})

Подход с применением настраиваемых функций является масштабируемым, позволяя переименовывать большое количество столбцов одновременно и даже импортировать старые и новые имена столбцов из внешнего списка. Эта функция позволяет задать последовательно несколько индексов. Например, для переименования всех столбцов, начиная с четвертого и заканчивая десятым, на Factor 1, Factor 2,..., Factor 7 без настраиваемой функции, можно выполнить интеллектуальные преобразования.

Начнем с неэффективного метода. Можно написать большой объем кода, который определяет каждую пару старых и новых имен столбцов следующим образом:

= Table.RenameColumns(

#"Promoted Headers",

{Table.ColumnNames(#"Promoted Headers"){6}, "Factor 1"},

{Table.ColumnNames(#"Promoted Headers"){7}, "Factor 2"},

{Table.ColumnNames(#"Promoted Headers"){8}, "Factor 3"},

{Table.ColumnNames(#"Promoted Headers"){9}, "Factor 4"},

{Table.ColumnNames(#"Promoted Headers"){10}, "Factor 5"},

{Table.ColumnNames(#"Promoted Headers"){11}, "Factor 6"},

{Table.ColumnNames(#"Promoted Headers"){12}, "Factor 7"}

Чтобы избежать этих сложностей, можно сформировать интеллектуальное пере-именование следующим образом:

= FnRenameColumnsByIndices(

#"Promoted Headers", List.Transform(

{1..7},

each "Factor " & Text.From(\_)

),

{6..12}

)

В этой формуле второй аргумент представляет динамический список, который создается функцией List.Transform. Первый аргумент функции — список индексов от 1 до 7. Его вывод — преобразованный список текста, объединяющий префикс Factor и соответствующий индекс. Третьим аргументом функции FnRenameColumnsByIndices служит список от 6 до 12 для индексов столбцов для переименования.

Рассмотрим функцию FnRenameColumnsByIndices:

(Source as table, ColumnNamesNew as list, Indices as list) =>

let

ColumnNamesOld = List.Transform( Indices, each

Table.ColumnNames(Source){\_} ),

ZippedList = List.Zip( { ColumnNamesOld, ColumnNamesNew } ),

#"Renamed Columns" = Table.RenameColumns( Source, ZippedList )

in

#"Renamed Columns"

Просмотрим действия этой функции шаг за шагом. Аргументы Source соответствуют исходной переименовываемой таблице, ColumnNamesNew — списку новых имен столбцов и индексов для списка индексов в исходной таблице. Первая строка внутри выражения let получает индексы и возвращает имена соответствующих столбцов в таблице Source:

ColumnNamesOld = List.Transform(Indices, each Table.ColumnNames(Source){\_}),

В следующей строке применяется функция List.Zip для формирования вложенных списков. Каждый вложенный список содержит два элемента — старые и новые имена столбцов, относящиеся к одному и тому же индексу в разных списках:

ZippedList = List.Zip( { ColumnNamesOld, ColumnNamesNew } ),

Например, эта функция List.Zip:

List.Zip({"a","b","c"}, {"A", "B", "C"}}

Возвратит следующий список из вложенных списков:

{{"a", "A"}, {"b", "B"}, {"c", "C"}}

Последний формат является обязательным для второго аргумента функции Table.RenameColumns — список вложенных списков, куда входят пары старых и новых имен столбцов, он применяется в третьей строке внутри выражения let:

#"Renamed Columns" = Table.RenameColumns(Source, ZippedList)

Функция Table.TransformColumnNames

Существует и другая методика для переименования столбцов. Напомним, что в главе 4 применялась функция Table.TransformColumnNames для переименования всех столбцов. Она заменяла подчеркивания пробелами или выполняла переход к заглавным буквам. Эта функция подойдет в рассматриваемом здесь сценарии для замены имен столбцов с помощью общего правила:

= Table.TransformColumnNames(#"Promoted Headers", each Text.Replace(\_, "Random Column",

"Factor "))

Преимуществом этого метода служит тот факт, что он успешно переименовывает столбцы, даже если случайные столбцы будут переупорядочены в исходной таблице. Тем не менее следует быть осторожным с логикой переименования, чтобы избежать переименования столбцов там, где не нужно.

СОВЕТ

В большинстве случаев достаточно простого переименования, как это обычно делается. Не особенно углубляйтесь в этот вопрос.

Можно загрузить файлы решения C10E05 - Solution.xlsx и C10E05 - Solution.pbix на сайте https://aka.ms/DataPwrBIPivot/downloads.

Ловушка 7. Разбиение столбца

на другие столбцы

Седьмая ловушка может быть столь же опасной, как и ловушка фильтрации (ло-вушка 4), поскольку способна приводить к пропуску данных. Она поджидает вас при использовании команды Разделить столбец по разделителю (Split Column By Delimiter) к столбцам, где значения разделены с помощью разделителя, причем заданы различные диапазоны значений.

Команда Разделить столбец по разделителю (Split Column By Delimiter) обычно применяется при выполнении операций двух типов:

■ Базовая операция позволяет разбивать столбец на несколько столбцов. Часто применяется для разделения столбцов даты и времени на отдельные столбцы даты и времени или при разделении имен клиентов на имена и фамилии. Окно диалога Разделить столбец по разделителю (Split Column By Delimiter) позволяет по умолчанию разделять столбец на столбцы.

■ Расширенная операция Разделить столбец по разделителю (Split Column By Delimiter) позволяет разделять несколько строк, разделенных запятыми или другими значениями, которые записаны через разделитель. Эта операция чрезвычайно полезна. Она позволяет создавать новую таблицу, которая связывает каждое разделяемое значение с его сущностью. Как показано на рис. 10.8, можно создавать справочную таблицу для исходной таблицы, с помощью которой

Рис. 10.8. Разделите столбец AdventureWorks Color с данными, записанными через запятую,

чтобы узнать, сколько выпущено товаров с разными цветами

устанавливается соответствие между кодами продуктов и их цветами, тогда легко судить, сколько товаров имеют различные цвета.

В упражнении 2.8 из главы 2 выполнялась обработка таблицы товаров Adven-tureWorks, а столбец Colors разделялся для установления связи между товарами и цветами, что позволяло определить, сколько выпущено товаров с учетом цвета. Вспомните, что для решения этой проблемы столбец Colors (Цвета) разбивался на строки. Следующее упражнение показывает, что может произойти, если вы разбиваете столбец на столбцы, а не на строки.

Седьмая ловушка, как показано в упражнении 10.6, возникает при разделении значений, записанных через разделитель, и игнорируете расширенную (и относительно скрытую) возможность по разделению столбца на строки. Если эта команда не будет выбрана, вы получите заданную по умолчанию команду по разбиению столбца на столбцы, которая может привести в дальнейшем к потерям важной информации, при загрузке в ваш отчет новых данных.

УПРАЖНЕНИЕ 10.6.

Выполнение некорректного разбиения

В данном упражнении повторяется упражнение 2.8, но теперь можно увидеть, что произойдет, если применяется некорректная команда разбиения, Разделить на столбцы (Split into Columns). Это упражнение демонстрирует риски и показывает, как легко здесь попасть в ловушку.

Воспользуйтесь примером рабочей книги C10E06.xlsx, которую можно загрузить на сайте https://aka.ms/DataPwrBIPivot/downloads. В этой книге приведены коды товаров компании AdventureWorks (новый вариант кодов), причем учитывается средняя стоимость, средняя цена и цвет, разделенные запятыми. Предположим, что вы, как руководитель отдела исследований данных компании AdventureWorks, желаете создать отчет, который показывает, сколько имеется товаров разного цвета (рис. 10.8).

1. Загрузите файл рабочей книги C10E06.xslx и сохраните его в папке C:\Data\ C10\.

2. Откройте пустую новую рабочую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data), и в окне диалога Получить данные (Get Data) выполните команды Файл (File) Excel.

3. Выберите файл C10E06.xlsx и щелкните мышью на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите вкладку Products (Товары) и щелкните на кнопке Изменить (Edit).

5. На панели Примененные шаги (Applied Steps) удалите шаг Changed Type (Из-мененный тип).

6. На панели Запросы (Queries) правой кнопки мыши щелкните на вкладке Products (Товары) и в контекстном меню выберите команду Ссылка (Refe-rence). Ваша цель состоит в создании новой таблицы с отображениями между кодами продуктов и цветами.

7. Переименуйте новый запрос в Products and Colors и оставьте его выделенным.

8. На вкладке Главная (Home) выберите команду Выбрать столбцы (Choose Columns), в открытом диалоговом окне выберите столбцы ProductCodeNew и Colors. Щелкните мышью на OK для закрытия диалогового окна, а все невыде-ленные столбцы удаляются из вашего запроса Products and Colors.

9. Выделите столбец Colors (Цвета) и на вкладке Преобразование (Transform) вы-берите команду Разделить столбец (Split Column) и затем выполните команду По разделителю (By Delimiter). В открывшемся окне диалога Разделить столбец по разделителю (Split Column by Delimiter) обратите внимание на разделитель в виде запятой, который выбран по умолчанию, также установлен переклю-чатель По каждому вхождению разделителя (Each Occurrence of the Delimiter), как показано на рис. 10.9. К сожалению, по умолчанию при закрытии окна выполняется разделение столбцов, а не строк. Для разделения строк нужно расширить раздел Расширенные параметры (Advanced Options) и вместо заданного переключателя Разделить на (Split Into) Столбцы (Columns) установить переключатель Разделить на (Split Into) на Строки (Rows), как это было выполнено в упражнении 2.8.

Рис. 10.9. В окне диалога Разделить столбец по разделителю (Split Column by Delimiter) переключатель Разделить на (Split Into) Строки (Rows) отображается при переходе в раздел Расширенные параметры (Advanced Options)

На этом этапе оставьте заданный по умолчанию переключатель разделения по столбцам, что позволит узнать о последствиях применения заданного по умолчанию переключателя. Щелкните на кнопке ОК для закрытия окна диалога.

10. Обратите внимание в окне редактора Power Query, что столбец Colors разделен на Colors.1 и Colors.2.

На панели Примененные шаги (Applied Steps) выберите шаг Split Column By Delimiter (Разделить столбец по разделителю) и рассмотрите код в строке формул:

= Table.SplitColumn(Products\_Table, "Colors", Splitter.SplitTextByDelimiter(",", QuoteStyle.Csv), {"Colors.1", "Colors.2"})

Эта формула указывает механизму языка M на разбивку столбца Colors (Цвета) на два столбца — именно на два. Далее эта операция будет рассмотрена подробнее.

11. Поскольку ваша цель — сопоставить коды товаров и цвета, необходимо ис-пользовать преобразование отмены свертывания столбцов для получения таблицы, состоящей из пар "код/цвет", поэтому выберем столбец ProductCodeNew.

12. На вкладке Преобразование (Transform) откроем раскрывающееся меню От-менить свертывание столбцов (Unpivot Columns) и выберем команду Отменить свертывание других столбцов (Unpivot Other Columns). Столбцы Colors.1 и Colors.2 трансформируются в столбцы Атрибут (Attribute) и Значение (Value). Последний столбец включает цвета, которые теперь корректно со-поставлены соответствующим кодам товаров. На этом этапе обратите внимание, что названия некоторых цветов начинаются с пробела. Чтобы это исправить, можно применить шаг Trim (Усечь) к столбцу Значение (Value) или перейти к панели Примененные шаги (Applied Steps) и в окне диалога Разделить столбец по разделителю (Split Column By Delimiter) щелкнуть мышью на значке настроек и измените разделитель на Настраиваемый (Custom), а затем ввести запятую и пробел (,). Вот исправленная формула для шага Split Column By Delimiter:

= Table.SplitColumn(Products\_Table, "Colors",

Splitter.SplitTextByDelimiter(", ",QuoteStyle.Csv), {"Colors.1", "Colors.2"})

13. Удалите столбец Атрибут (Attribute) и загрузите запросы в отчет для того, чтобы начать анализ. Если необходимо, постройте гистограмму, используя сводную диаграмму так, как показано на рис. 10.8.

14. Для того чтобы проверить, что произойдет, если определенный код продукта относится к более чем двум цветам, сохраните отчет и откройте файл C10E06.xslx.

15. В ячейку D3 добавьте еще два цвета к имеющимся двум и измените значение столбца Colors (Цвета). В результате к желтому и синему цвету будут добавлены черный и красный цвета. Сохраните рабочую книгу C10E06.xslx и закройте ее. Вернитесь к отчету упражнения и обновите его. В результате новые цвета не будут включены в отчет.

Для решения этой проблемы необходимо выполнить упражнение 2.4 и разбить столбец Colors (Цвета) на строки. Но если по какой-то причине необходимо раз-бить столбец на столбцы, а не на строки, то можно отмасштабировать решение, увеличивая число столбцов, на которые нужно выполнить разбиение. Допустим, заранее известно, что количество цветов не может превышать 20. Можно задать эту формулу на шаге Split Column By Delimiter (Разделить столбец по разделителю):

= Table.SplitColumn(Products\_Table, "Colors",

Splitter.SplitTextByDelimiter(", ", QuoteStyle.Csv), 20)

Заменяя жестко закодированную часть, {"Colors.1", "Colors.2"}, на 20, т. е. на макси-мальное число столбцов, которое вы ожидаете получить, можно усилить запрос и приобрести уверенность, что данные не будут пропущены.

Файлы решения C10E06 - Solution.xlsx и C10E06 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Убедитесь, что также загружена рабочая книга C10E06 - v2.xlsx, где содержится вторая версия рабочей книги C10E06.xlsx, с обновленными значениями цветов. Запросы в файлах решения затребуют наличия рабочей книги C10E06 - v2.xlsx в папке C:\Data\C10\.

Ловушка 8. Слияние столбцов

В предыдущем разделе было продемонстрировано, как можно избежать седьмой ловушки, разбивая столбцы на строки, а не на столбцы. Также показано, что при необходимости разбить на столбцы столбец, разделенный запятыми, можно выбрать второй переключатель и запросить количество столбцов для генерации. Теперь рассмотрим, как можно улучшить запрос при выполнении обратного преобразования: объединения столбцов.

При объединении в один столбец нескольких столбцов сгенерированная формула сначала преобразует все числовые столбцы в текст, а затем объединяет вместе все столбцы. Следующий код генерируется автоматически (с отступом и форматируется для лучшей читабельности с несколькими строками), при объединении трех столбцов в исходной таблице, первые из которых являются числовыми (Numeric Column1 и Numeric Column 2), а третий — текстовым (Textual Column3):

#"Merged Columns" = Table.CombineColumns(

Table.TransformColumnTypes(

Source, {

{"Numeric Column1", type text},

{"Numeric Column2", type text}

},

"en-US"

),

{"Numeric Column1", "Numeric Column2", "Textual Column3"}, Combiner.CombineTextByDelimiter(":", QuoteStyle.None),

"Merged"

Функция Table.TransformColumnTypes здесь обеспечивает преобразование типов всех числовых столбцов, а затем объединяет соответствующие столбцы. Эта формула может привести к ошибкам обновления. Целесообразнее изменить код и масштабировать его для объединения заданного списка столбцов, не ссылаясь на какие-либо жестко закодированные имена столбцов.

Допустим, имеются имена столбцов для слияния в списке ColumnsToMerge. Далее приводится модифицированная формула, которую можно применять вместо предыдущей:

#"Merged Columns" = Table.CombineColumns(

Table.TransformColumnTypes(

Source,

List.Transform(

ColumnsToMerge,

each {\_, type text}

),

"en-US"

),

ColumnsToMerge,

Combiner.CombineTextByDelimiter(":", QuoteStyle.None),

"Merged")

Основное различие этих двух формул сконцентрировано в следующей части кода:

{

{"Numeric Column1", type text},

{"Numeric Column2", type text}

}

Замена выполняется с помощью функции List.Transform, функция генерирует тот же код, но без ссылки на имена столбцов:

List.Transform(

ColumnsToMerge,

each {\_, type text}

)

Этот код выполняет итерации по каждому имени столбца в ColumnsToMerge и преобразует его в список имен столбцов и текстового типа: {\_, type text}.

И хотя обычно подобная функция может и не понадобиться, рассмотренный пример служит важной демонстрацией применения функций списка при формировании масштабируемых и надежных версий для ваших автоматически сгенерированных формул.

Дополнительные ловушки и методы

для создания надежных запросов

Я уже упоминал, что в серии блогов на DataChant.com описываются 10 ловушек, а в данную главу девятая и десятая ловушки не вошли. В этом разделе они упо-минаются кратко, а для получения дополнительной информации посетите мой блог. Девятая ловушка связана с расширением столбцов таблицы. Развернуть столбцы таблицы — значит выполнить обычное преобразование, которое применяется при объединении нескольких файлов из папки, объединении двух таблиц или при работе с неструктурированными наборами данных, такими как JSON.

При развертывании столбцов таблицы необходимо на первом этапе выбрать столбцы для развертывания. В результате редактор Power Query автоматически генерирует формулу с жестко закодированными именами столбцов, а новые имена столбцов могут быть пропущены. Чтобы дополнительно узнать о преобразовании Развернуть столбцы таблицы (Expand Table Columns), а также о том, как избежать пропусков новых столбцов, перейдите по ссылке https://datachant.com/2017/02/07/ power-bi-pitfall-9/.

Десятая ловушка относится к удалению дубликатов, она рассматривалась ранее в этой книге, в упражнении 2.7 в главе 2.

Резюме

В этой главе рассматриваются профессиональные навыки обработки данных, причем уделяется внимание именно долгосрочному подходу. В рамках этого подхода показано, каким образом можно "продлить жизнь" запросам, не прибегая к постоянному редактированию по мере изменения данных. Рассмотрены наиболее распространенные "ловушки", которые приводят к сбоям при обновлении и появлению некорректных данных. Также описаны три важные темы, позволяющие избегать ловушек: осведомленность, лучшие практические методики и модификации кода языка М. Активная строка формул и частый ее просмотр поможет вам обнаружить жестко заданные двойные кавычки или неправильные условия.

В данной главе также рассказано о передовых практиках, которые помогают уменьшить число сбоев. Наличие официального или неофициального контракта с коллегами, которые владеют источниками данных, помогает делать правильные предположения относительно данных и принимать верные решения при устранении проблем по подготовке данных.

Также показано, каким образом можно применять небольшие модификации для М-формул, что способствует улучшению ваших запросов. В заключение главы в таблице 10.1 перечислены основные "подводные камни", причины их появления, а также способы их устранения.

Таблица 10.1. Основные ловушки, причины их появления и методы преодоления

Номер и наимено-вание ловушки Причины появления Как избежать ловушки

1. Игнорирование строки формул Недостаточная осведом-ленность о потенциальных сбоях автоматически сгенерированного М-кода Активируйте строку формул и продолжайте просматривать ее код. Разыскивайте значения в двойных кавычках, что позволит убедиться в том, что они соответствуют выбранным интерфейсам пользователя

2. Измененные типы Высокая вероятность ошибок обновления в будущем Удалите шаг Changed Type (Измененный тип). Отключите автоматическое определение типов в окне диалога Параметры запроса (Query Options) для каждой книги или отчета Power BI. Измените типы вручную в числовых столбцах/столбцах даты/времени. Чем позже вы поменяете типы, тем лучше

3. Небезопасная фильтрация Высокая вероятность появления неверных данных Избегайте задания значений на панели фильтров. Вместо этого используйте окно диалога Фильтрация строк (Filter Rows) или выполните аудит строки формул, чтобы убедиться, что логика фильтрации создана так, как и предполагалось

4. Переупорядочение столбцов Высокая вероятность будущих ошибок обновления Избегайте переупорядочения столбцов, если это возможно. Используйте функцию языка M для изменения порядка только для подмножества столбцов

5. Удаление и выбор столбцов Средняя вероятность ошибок обновления в будущем Если имеется небольшое число столбцов, которые необходимо сохранить, воспользуй-тесь командой Выбрать столбцы (Choose Columns) вместо команды Удалить столбцы (Remove Columns). Во многих случаях столбцы, которые выбираются для сохране-ния, с большей вероятностью останутся в ваших исходных данных. Применяйте язык M для удаления или выбора столбцов по их расположению в таблице

6. Переименование столбцов Средняя вероятность будущих ошибок обновления Используйте функцию Table.TransformColumnNames в M для пере-именования столбцов по их значениям или функцию Table.ColumnNames с индексами "от нуля" для переименования по позициям

7. Разделение столбца на столбцы Высокая вероятность пропуска данных Избегайте преобразования Разбить на столбцы (Split into Columns). Вместо этого задайте расширенный параметр и команду Разбить на строки (Split into Rows). Если необходимо разделить на столбцы, назначьте в качестве второго аргумента максимальное число для количества столбцов

8. Слияние столбцов Средняя вероятность ошибок обновления На все столбцы будут приведены ссылки в коде. Модифицируйте код M, если необхо-димо объединить большое число столбцов или список имен столбцов для слияния является динамическим

Таблица 10.1 (окончание)

Номер и наимено-вание ловушки Причины появления Как избежать ловушки

9. Расширение столбца таблицы Средняя вероятность пропусков данных Обратитесь на сайт https://datachant.com/

2017/02/07/power-bi-pitfall-9/

10. Слияние дубликатов Высокая вероятность ошибок обновления и неудачных попыток соз-дать взаимосвязь между таблицами поиска и фак-тов Обратитесь к главе 2, упражнение 2.7, и убедитесь, что применяются строч- ные/прописные буквы и выполняется обрезка значений, прежде чем удалять дубликаты в столбце, который соответствует таблицам поиска

Глава 11

Основы текстовой аналитики

Слова пустые, как ветер, лучше не произносить.

- Гомер

В ЭТОЙ ГЛАВЕ ВЫ...

■ Узнаете, как искать ключевые слова в текстовых столбцах и применять эту технику в качестве простого метода определения темы.

■ Познакомитесь с методикой загрузки внешнего списка ключевых слов и их динамическим использованием для поиска.

■ Примените декартово произведение или пользовательскую функцию для поиска по ключевым словам.

■ Воспользуетесь функциями Table.Buffer и List.Buffer для повышения производи-тельности обработки отчетов при нахождении ключевых слов.

■ Рассмотрите несколько методов разделения слов, игнорирования общих слов и подсчета вхождений слов при выполнении анализа.

■ Узнаете, как можно подготовить текстовые данные и загрузить их в пользова-тельский визуальный элемент.

В этой главе будет показано, как использовать Power Query для получения фундаментальных сведений о текстовых данных. Многие таблицы в ваших отчетах могут включать большое число текстовых столбцов, которые не обрабатываются и вообще лишены смысла. Перефразируя высказывание Гомера о том, что пустые слова лучше не произносить, можно в отчетах обнаружить большое число слов, которые "пусты, как ветер". Эти текстовые столбцы обычно игнорируются в отчете или служат только в качестве контекста при детализации определенных строк.

Однако из этих текстовых столбцов можно получить дополнительные сведения. В главах 12 и 13 будут описаны полезные методы работы с текстовыми столбцами, вы получите доступ к новым способам проникновения в их суть, что позволит лучше оценить их.

Методы, описанные в настоящей главе, чрезвычайно полезны при анализе таблиц с фрагментами текста, например отзывов клиентов в опросах, оперативных комментариев или каналов социальных сетей. Показано, как можно применять общие преобразования для извлечения значений из слов. Начнем рассмотрение с базовых методик обнаружения ключевых слов и постепенно расширим набор инструментов для выполнения общих текстовых аналитических задач, реализация которых в Power Query несколько сложнее. Путь к анализу текста будет продолжен в главе 12, где рассматриваются вопросы расширенной аналитики текста в Azure, включая перевод текста, анализ настроений и обнаружение ключевых фраз, — и все это можно выполнить без помощи специалистов по исследованию данных или разра-ботчиков программного обеспечения.

Поиск ключевых слов в текстовых столбцах

Первая задача по анализу текста часто встречается на практике: выполнение поиска конкретных ключевых слов в текстовых источниках. И хотя эта задача может показаться тривиальной, она значительно усложняется, если необходимо разыскать большое число ключевых слов или следует динамически загружать ключевые слова из внешнего источника без последующей модификации запросов. В упражнении 11.1 будет показано, как можно обнаружить небольшой набор предопределенных ключевых слов.

УПРАЖНЕНИЕ 11.1. Основная методика, применяемая для поиска ключевых слов

В данном упражнении ключевые слова разыскиваются в сообщениях, которые Microsoft Press использует в постах на своей официальной странице в Facebook (https://www.facebook.com/microsoftpress). Обнаруживая ключевые слова, можно узнать, какие темы наиболее активно продвигаются социальной группой Microsoft Press.

Для анализа сообщений Microsoft Press на Facebook необходимо загрузить исходную рабочую книгу C11E01.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads и сохранить ее в папке C:\Data\C11\.

Можно открыть файл и просмотреть его таблицу, куда включены следующие столбцы: сообщения Facebook (столбец Message), дата публикации поста (столбец Date), уникальный идентификатор поста (столбец Post ID) и URL-ссылка действительного поста на Facebook (столбец URL).

Для обновления рабочей книги с учетом последних постов Facebook можно вы-брать команду Обновить все (Refresh All) на вкладке Данные (Data), но убедитесь, что упражнения в этой главе выполняются с использованием оригинальной рабочей книги, без ее обновления. Также учтите, что в главе 13 будет показано, как импортировать фактические сообщения Facebook с помощью Power Query. Основное внимание в данной главе уделяется анализу текста. Поэтому начнем это упражнение с набора данных, сохраненного в файле Excel C11E01.xslx. Если у вас нет возможности выполнить все действия, можно просмотреть запрос в файле решения, который импортирует сообщения непосредственно из Facebook.

Представьте, что вы аналитик данных, работающий в команде социальных сетей Microsoft Press. Ваше первое задание — оценить, какие из приведенных далее тем книги наиболее популярны в обсуждениях на Facebook:

■ Microsoft Excel.

■ Visual Studio.

■ Azure.

■ Windows.

Как обнаружить эти четыре ключевых слова в столбце Message (Сообщение) из набора данных? Это упражнение разбито на шаги для удобства рассмотрения.

1. Откройте пустую новую рабочую книгу или новый отчет Power BI Desktop.

В Excel: откройте новую рабочую книгу Excel и на вкладке Данные (Data) вы-полните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: откройте новый отчет и выполните команды Получить данные (Get Data) Excel Подключиться (Connect).

2. Выберите файл C11E01.xslx в папке C:\Data\C11\ и щелкните мышью на кнопке Импорт (Import).

3. В окне диалога Навигатор (Navigator) выберите вкладку Sheet1 и щелкните на кнопке Изменить (Edit).

4. После открытия окна диалога Power Query обратите внимание, что посты Microsoft Press Facebook сохраняются в столбце Message (Сообщение), как показано на рис. 11.1.

Поскольку Power Query чувствителен к регистру при поиске, на следующих шагах продублируем столбец Message (Сообщение) и поработаем над дубликатом, где все символы текста переведены в нижний регистр. Таким образом, исходные сообщения сохранятся без изменений, что особенно важно, если следует включить исходный текст в отчет и сохранить определенные символы в сообщениях в верхнем регистре.

5. Выделите столбец Message (Сообщение). На вкладке Добавить столбец (Add Column) выберите команду Создать дубликат столбца (Duplicate Column).

6. Выберите новый столбец Message - Copy (Сообщение - Копия) и на вкладке Преобразование (Transform) выполните команду Формат (Format), затем вы-берите команду нижний регистр (Lowercase).

7. Переименуйте столбец Message - Copy в Lowercased (Копия в нижнем регистре).

Затем создадим столбец Topic (Тема) и заполним его одной из обнаруженных тем: Microsoft Excel, Visual Studio, Azure или Windows.

8. На вкладке Добавить столбец (Add Column) выберем команду Условный столбец (Conditional Column). Откроется окно диалога Добавить условный столбец (Add Conditional Column).

9. Как показано на рис. 11.2, создадим новый столбец, который возвращает одну из четырех тем, если они найдены в столбце Lowercased. Для реализации этой методики воспользуемся следующими шагами:

а. Введите Topic (Тема) в поле Имя нового столбца (New Column Name.)

б. В списке Имя столбца (Column Name) выберите значение Lowercased.

в. В списке Оператор (Operator) выберите значение Содержит (Contains).

г. Введите microsoft excel в поле Значение (Value).

д. Введите Microsoft Excel в поле Вывод (Output).

ПРИМЕЧАНИЕ

На шаге 9г используется строчная версия для темы (microsoft excel), поскольку поиск чувствителен к регистру символов. На шаге 9д применяется заглавная версия в целях создания отчетов.

10. Находясь в окне диалога Добавление условного столбца (Add Conditional Column), щелкните мышью на кнопке Добавить правило (Add Rule). В окне диалога появится новая строка условия. Повторите шаг 9 со следующими исключениями: на шаге 9г введите ключевое слово visual studio в поле Значение (Value), на шаге 9д введите Visual Studio в поле Вывод (Output).

11. Щелкните на кнопке Добавить правило (Add Rule). Повторите шаг 9 со сле-дующими исключениями: на шаге 9г введите ключевое слово azure в поле Значение (Value), на шаге 9д введите Azure в поле Вывод (Output).

12. Повторите шаг 9 со следующими исключениями: на шаге 9г введите ключевое слово windows в поле Значение (Value), на шаге 9д введите Windows в поле Вывод (Output).

13. Введите значение Other в поле В противном случае (Otherwise). Убедитесь, что окно диалога настроено так, как показано на рис. 11.2, и щелкните мышью на кнопке OK для закрытия окна диалога.

Рис. 11.2. Окно диалога Добавление условного столбца (Add Conditional Column) включает условия для проверки тем Microsoft Excel, Visual Studio, Azure и Windows

Теперь панель предварительного просмотра содержит новый столбец Topic (Тема). Очевидно, что этот метод недостаточно эффективен при обнаружении всех тем книги из постов Microsoft Press Facebook. Большое число строк классифицированы как Other (Другие). В некоторых постах рекламируются другие книги по таким темам, как Office 365 или SQL Server. Другие сообщения продвигают общие скидки на продажах. Подумайте о следующем сообщении: "visit www.microsoftpressstore.com today — and discover deals, new books, and special promotions!" (Посетите сайт www.microsoftpressstore.com сегодня и узнайте о предложениях, новых книгах и специальных акциях!). Для простоты можно предположить, что этот подход соответствует вашим целям, и теперь можно отфильтровать строки с помощью значения Other (Другое) в столбце Topic (Тема).

14. Щелкните мышью на элементе управления фильтром в заголовке столбца Topic, отмените выбор Other в панели Фильтр (Filter) и щелкните на кнопке OK.

15. Наконец, удалите столбец Lowercased, переименуйте запрос в Microsoft Press Posts и загрузите ваш отчет.

Можно просмотреть файлы решения C11E01 - Solution.xlsx или C11E11 - Solution.pbix, которые доступны на сайте https://aka.ms/DataPwrBIPivot/downloads.

Теперь, когда цель достигнута, рассмотрим особенности метода, примененного в этом упражнении:

■ Предупреждение 1. Определение темы ограничено кратким списком ключевых слов. Если имеется много ключевых слов, то нецелесообразно вручную добавлять большое число ключевых слов в окно диалога Добавление условного столбца (Add Conditional Column). Необходимо найти альтернативный способ решения этой проблемы.

■ Предупреждение 2. Определение темы ограничено статическим списком ключевых слов. При использовании этого метода необходимо вводить ключевые слова в окно диалога. Кроме того, можно изменить основную формулу M, но результат будет тем же. Ключевые слова жестко закодированы в формуле. Если необходимо загрузить список ключевых слов из внешнего источника, а заранее неизвестно, какие ключевые слова следует обнаружить, то данный метод будет неэффективным. Следует найти метод, который позволит динамически загружать список ключевых слов из внешнего источника (и гарантирует, что выполнение этой операции не замедляет вашей работы).

■ Предупреждение 3. Определение темы ограничено одним ключевым словом на одно сообщение. Некоторые посты Facebook могут содержать несколько ключевых слов. Например, рассмотрим следующий пост: "New book! Programming for the Internet of Things: Using Windows 10 IoT Core and Azure IoT Suite" (Новая книга! Программирование для Интернета вещей: использование Windows 10 IoT Core и Azure IoT Suite). В этой статье пропущено ключевое слово Windows, поскольку логические приоритеты выделили ключевое слово Azure в окне диалога Добавление условного столбца (Add Conditional Column) и производной для него формуле М. В некоторых случаях лучше распределить несколько ключевых слов для обработки одного сообщения, чтобы в сообщении учитывалось каждое ключевое слово.

■ Предупреждение 4. Слова, содержащиеся в качестве подстрок более длинных слов, могут быть указаны ошибочно. Поскольку вы применили оператор Со-держит (Contains), можно обнаружить ключевые слова, которые служат под-строками более длинных слов. Например, рассмотрим ключевое слово Excel. Подумайте: какое влияние на результат поиска окажут следующие два фейковых сообщения?

"This SQL Server book is excellent". ("Эта книга по SQL Server превосходна".) "Read this C# book to excel in software development". ("Прочитайте эту книгу по C#, чтобы преуспеть в разработке программного обеспечения".)

В обоих примерах не указана реальная тематика книги. Чтобы избежать ложного определения тем, необходимо разделить сообщения на отдельные слова и сопоставить их с ключевыми словами. Далее в этой главе более подробно рассматривается этот вопрос.

Теперь рассмотрим первые три предупреждения и представим отличный метод для сравнения нескольких наборов данных: применение декартового произведения.

Применение декартова произведения для определения ключевых слов

В этом упражнении рассмотрен эффективный метод обнаружения большого числа ключевых слов. В предыдущем упражнении для определения четырех ключевых слов применялся условный столбец (рис. 11.2). Целесообразен ли вариант такого же подхода для поиска большого набора ключевых слов? Добавление длинного списка ключевых слов вручную представляет собой трудоемкий и довольно сложный процесс, тем более что периодически необходимо обнаруживать различные ключевые слова.

Для расширения масштабов решения, примененного в упражнении 11.1, и обнаружения большого числа тем из динамического списка ключевых слов можно выполнять декартово произведение между постами Microsoft Press и списком ключевых слов. При этом формируется временная таблица большего размера, как показано на рис. 11.3. Таблица состоит из всех комбинаций между постами в Facebook и ключевыми словами.

ПРИМЕЧАНИЕ

В теории множеств декартово произведение может применяться к нескольким наборам, что позволит сформировать перечень всех комбинаций элементов из разных наборов — по одному из каждого набора. В Power Query можно применять декартово произведение к любым двум таблицам (и даже к одной таблице), добавляя в первую таблицу настраиваемый столбец со ссылкой на вторую таблицу. Расширяя в дальнейшем новый столбец, можно получать декартово произведение, как показано на рис. 11.3.

ГЯТГ ’ | Message Keyword Ы

1 New Book I Master Powerpoint excel

1 New Book! Master Powerpoint powerpoint

1 New Book! Master Powerpoint microsoft word

1 New Book! Master Powerpoint ...

2 Special offer: Microsoft Word excel

2 Special offer: Microsoft Word powerpoint

2 Special offer: Micr osoft Word microsoft word

2 Special offer: Microsoft Word

3 Excel the art of Microsoft Excel excel

3 Excel the ar t of Mier osoft Excel power point 3 Excel tiie art of Microsoft Excel microsoft word

3 Excel the ar t of Microsoft Excel ...

... excel

powerpoint microsoft word

Рис. 11.3. Применение декартова произведения между сообщениями и ключевыми словами

В верхней левой таблице на рис. 11.3 показаны три сообщения. Справа находится еще одна таблица с ключевыми словами. В результате выполнения декартова произведения создается новая таблица со всеми комбинациями пар "сообщение/клю- чевое слово", как показано в нижней части рис. 11.3. Данный метод может помочь при изучении всех комбинаций и применении дальнейших вычислений к каждой из пар. В данном случае вычисления предполагают поиск ключевого слова в сообщении и последующую фильтрацию соответствующих строк.

ПРИМЕЧАНИЕ

Имеется большое число применений декартова произведения, которые помогают по- новому взглянуть на ваши данные. Сопоставляя все комбинации между несколькими наборами данных или объектами, можно осуществлять вычисления "что если", выполнять баскет-анализ и реализовать вычисления с помощью методов из теории графов (например, поиск кратчайшего пути).

Если вы желаете применить декартово произведение или же узнать, в чем преимущества подобного подхода при широкомасштабном поиске по ключевым словам, выполните следующее упражнение.

УПРАЖНЕНИЕ 11.2.

Реализация декартова произведения

В качестве главного аналитика социальной группы Microsoft Press (вас повысили в должности после реализации методов из упражнения 11.1 — поздравляем!) перед вами стоит новая задача — анализировать посты Microsoft Press на Facebook по темам. Для определения темы можно применять декартово произведение между постами Facebook и большим набором ключевых слов, как показано на рис. 11.4.

Часть 1. Начальная подготовка

Возьмем тот же источник данных, что и в упражнении 11.1, которое включало посты Microsoft Press в Facebook, и выполним некоторую начальную подготовку.

1. Загрузите файл рабочей книги C11E01.xslx с сайта https://aka.ms/ DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C11\.

2. Откройте пустую новую рабочую книгу или новый отчет Power BI Desktop.

В Excel: откройте новую книгу Excel и на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: откройте новый отчет и выберите команды Получить данные (Get Data) Excel Подключиться (Connect).

3. Выберите файл C11E01.xslx в папке C:\Data\C11\ и щелкните мышью на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите вкладку Sheet1 и щелкните на кнопке Изменить (Edit).

5. Переименуйте запрос в Microsoft Press Posts.

Затем создается ссылка на запрос Microsoft Press Posts для формирования но-вой таблицы сообщений и темы, причем один и тот же идентификатор поста (Post ID) сообщения может включать несколько строк.

6. На панели Запросы (Queries) щелкните правой кнопкой мыши на запросе Microsoft Press Posts и в контекстном меню выберите команду Ссылка (Reference).

7. Переименуйте новый запрос в Post Topics.

8. Выделив запрос Post Topics на панели Запросы (Queries), выберите команду Выбор столбцов (Choose Columns) на вкладке Главная (Home).

9. В окне диалога Выбор столбцов (Choose Columns) выберите переключатели Post ID и Message и щелкните на кнопке OK.

10. Выделите столбец Message (Сообщение). На вкладке Добавление столбца (Add Column) выберите команду Создать дубликат столбца (Duplicate Column).

11. Выделите столбец Копия Message. На вкладке Преобразование (Transform) выберите команду Формат (Format) и затем выберите команду нижний ре-гистр (Lowercase).

12. Переименуйте столбец Копия Message в Lowercased.

13. Загрузите текстовый файл Keywords.txt с сайта https://aka.ms/DataPwrBIPivot/ downloads и сохраните его в локальной папке, C:\Data\C11\.

14. Для загрузки ключевых слов как нового запроса, при открытом редакторе Power Query щелкните на кнопке Новый источник (New Source) на вкладке Главная (Home) и импортируйте файл Keywords.txt путем выбора команд Файл Текст/CSV (File Text/CSV).

15. При открытом окне диалога Keywords.txt, при предварительном просмотре файла, щелкните на кнопке OK.

16. Поле загрузки запроса Keywords переименуйте соответствующий столбец в Keyword .

17. Для установки нижнего регистра во всех ключевых словах выберите столбец Keyword и затем на вкладке Преобразование (Transform) выполните команду Формат (Format), после чего выберите команду нижний регистр (Lowercase). Этот шаг гарантирует, что вы будете работать со строчной версией сообщений и не пропустите ничего из-за чувствительного к регистру символов процесса фильтрации, которая применяется в следующем разделе.

Часть 2. Декартово произведение

Теперь, после завершения начальной подготовки на шагах 1-17, все подготовлено к вычислению декартова произведения.

С помощью настраиваемого столбца можно формировать различные комбинации постов и тем Facebook, а затем применять фильтр для сопоставления постов и тем. Продолжите упражнение, выполняя следующие шаги:

18. На панели Запросы (Queries) выберите запрос Post Topics.

19. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). В окно диалога Настраиваемый столбец (Custom Column) внесите следующие изменения:

а. В поле Имя нового столбца (New Column Name) введите Cartesian.

б. В окне Пользовательская формула столбца (Custom Column Formula) введите следующую формулу:

= Keywords

в. Щелкните мышью на кнопке OK, чтобы закрыть окно диалога.

20. В результате выполнения описанных действий добавляется столбец Cartesian, с таблицей Keywords в каждой строке. На этом этапе каждая таблица Keywords представляется как объект таблицы редактора Power Query. Пришло время его расширить.

21. Щелкните на значке расширения в имени заголовка столбца Cartesian. (В качестве альтернативы выделите столбец Cartesian, затем выполните команду Расширить (Expand) на вкладке Преобразование (Transform).) На панели Расширить (Expand) сохраните столбец Keyword (Ключевое слово) отмеченным и отмените установку флажка Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix). Затем щелкните на кнопке OK.

Обратите внимание на панель предварительного просмотра — декартово произведение успешно реализовано. Пришло время узнать, почему декартово произведение столь полезно. Ко всем комбинациям сообщений и ключевых слов можно применить фильтр, что позволит сохранить только совпадающие строки. Можно использовать шаг преобразования Add Conditional Column (Добавить условный столбец), за которым следует фильтр для нового условного столбца.

22. На вкладке Добавление столбца (Add Column) выберите команду Условный столбец (Conditional Column). Откроется окно диалога Добавление условного столбца (Add Conditional Column). Внесите следующие изменения, как показано на рис. 11.5.

а. В поле Имя нового столбца (New Column Name) введите Topic.

б. В раскрывающемся списке Имя столбца (Column Name) выберите столбец Lowercased (В нижнем регистре).

в. В раскрывающемся списке Оператор (Operator) выберите опцию Содержит (Contains).

г. В раскрывающемся списке Значение (Value) выберите вариант Выбрать столбец (Select a Column) и затем выберите столбец Keyword.

д. В раскрывающемся списке Вывод (Output) выберите вариант Выбрать столбец (Select a Column) и затем выберите столбец Keyword.

е. Введите null в поле В противном случае (Otherwise).

ж. Щелкните мышью на OK для закрытия окна диалога.

Добавление условного столбца

Добавьте условный столбец, который вычисляется из других столбцов или значений.

Имя нового столбца

Отмена

Рис. 11.5. Окно диалога Добавление условного столбца (Add Conditional Column) может обрабатывать соответствия между каждым постом Microsoft Press Facebook и значением ключевого слова в декартовом произведении

После выполнения перечисленных действий добавляется столбец Topic (Тема). Он содержит большое число нулевых значений для всех комбинаций постов, которые не включают предложенные ключевые слова. Пришло время отфильтровать все строки с нулевыми значениями.

23. Щелкните на значке фильтра в столбце Topic и на панели Фильтр (Filter) вы-берите команду Удалить пустые (Remove Empty).

24. Удалите столбцы Message, Lowercased и Keyword.

Рассмотрим результат. Используя декартово произведение, вы смогли сопоставить сообщения с определенными темами из динамического списка. Для создания отчета понадобятся две таблицы. Первая таблица — Microsoft Press Posts, со многими столбцами, включая Post ID и Message. Второй таблицей служит Post Topics, со столбцами Post ID и Topic.

Часть 3. Создание соответствий

Теперь можно перейти от подготовки данных к моделированию, что позволит выполнять анализ сообщений по темам. И хотя моделирование данных не является предметом рассмотрения в этой книге, здесь приводятся шаги, которые помогут вам, если вы — новичок в работе с Power Pivot в Excel или в вопросах моделирования в Power BI Desktop. С помощью следующих шагов формируется связь между сообщениями и темами, которая позволит создавать различные визуальные элементы и сводные таблицы, составленные из двух таблиц.

25. При работе с Excel воспользуйтесь следующими шагами для загрузки запросов Microsoft Press Posts и Post Topics в модель данных:

а. На вкладке Главная (Home) редактора Power Query выберите команду Закрыть и загрузить (Close & Load) для загрузки запроса в отчет.

б. На вкладке Данные (Data) в Excel выберите команду Запросы и подключе-ния (Queries & Connections).

в. На панели Запросы и подключения (Queries & Connections) правой кнопкой мыши щелкните на запросе Microsoft Press Posts и в контекстном меню выберите команду Загрузить в (Load To).

г. После открытия окна диалога Импорт данных (Import Data) установите пе-реключатель Только создать подключение (Only Create Connection) и затем установите флажок Загрузить эти данные в модель данных (Load This Data to the Data Model).

д. На панели Запросы и подключения (Queries & Connections) правой кнопкой мыши щелкните на запросе Post Topics и в контекстном меню выберите команду Загрузить в (Load To).

е. После открытия окна диалога Импорт данных (Import Data) установите пе-реключатель Только создать подключение (Only Create Connection) и затем установите флажок Загрузить эти данные в модель данных (Load This Data to the Data Model).

ж. На панели Запросы и подключения (Queries & Connections) правой кнопкой мыши щелкните на запросе Keywords и в контекстном меню выберите команду Загрузить в (Load To).

з. После открытия окна диалога Импорт данных (Import Data) установите пе-реключатель Только создать подключение (Only Create Connection) и затем отмените установку флажка Загрузить эти данные в модель данных (Load This Data to the Data Model).

и. На вкладке Данные (Data) выберите команду Вернуться в окно Power Pivot (Manage Data Model). В открытом окне Power Pivot, на вкладке Главная (Home) выберите команду Представление диаграммы (Diagram View).

В Power BI Desktop выполните следующие шаги для выгрузки запроса Keywords из отчета:

а. На панели Запросы (Queries) редактора Power Query правой кнопкой мыши щелкните на запросе Keywords и отмените установку флажка Включить загрузку (Enable Load).

б. На вкладке Главная (Home) выберите команду Закрыть и применить (Close & Apply).

в. Выберите значок Связи (Relationships) в левой панели Power BI Desktop.

26. Для формирования взаимосвязи между двумя таблицами, как показано на рис. 11.6, перетащите поле Post ID из таблицы Microsoft Press Posts на поле Post ID в таблице Post Topics.

Теперь все подготовлено к выполнению анализа (рис. 11.4). Загрузите рабочую книгу решения C11E02 - Solution.xlsx или отчет Power BI C11E02 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads и рассмотрите, каким образом эти связанные таблицы в модели данных можно применять в сводной диаграмме и срезе для определения количества постов Facebook, которые опубликованы по этой теме за определенный промежуток времени.

Сохраните отчет. Важный шаг, связанный с повышением производительности, будет описан в ближайшее время, но прежде, чем приступить к этому вопросу, рассмотрим, что стало известно по этому поводу. Вернувшись к первым трем предостережениям, перечисленным в конце упражнения 11.1, мы можем констатировать, что метод декартовых произведений помогает преодолеть следующие затруднения:

■ Стало возможным выполнить загрузку большого списка ключевых слов, не вводя каждое вручную в окно диалога Добавление условного столбца (Add Conditional Column). Предостережение 1 учтено.

■ Теперь можно динамически загрузить список ключевых слов из внешнего файла, не зная заранее, какие именно ключевые слова будут использоваться. Предостережение 2 было учтено, но ваше решение можно улучшить с точки зрения производительности, поэтому оставьте редактор Power Query открытым, чтобы можно было в следующем разделе обсудить эти вопросы.

■ Представилась возможность связать несколько тем с отдельными сообщениями, когда одно сообщение содержит два или более ключевых слов. Предостережение 3 учтено.

Часть 4. Повышение производительности

Подумайте о производительности вашего решения. При обновлении отчета обратите внимание, что запрос загружается относительно быстро. Декартово произведение в сочетании с фильтрацией совпадающих строк реализовано эффективно и не требует много памяти. При создании декартова произведения и расширении всех комбинаций между постами в Facebook и ключевыми словами расширенная таблица не полностью загружалась в память. Вместо этого механизм языка M выполнял преобразования с небольшими фрагментами двух таблиц.

Это важная и универсальная способность Power Query: если возможно, для не-большой части данных применяется несколько шагов преобразования, и Power Query выполняет итерации по набору данных, порциями, пока не преобразуется весь массив данных. Такой подход гарантирует, что механизм языка M не будет загружать весь набор данных в память, если этого не требуется. В результате декартово произведение не требует значительного объема памяти.

Тем не менее вы можете существенно улучшить производительность декартова произведения, выполняя описанные далее шаги в своей книге или отчете Power BI из упражнения 11.2:

1. В окне редактора Power Query выберите запрос Post Topics и на панели Приме-ненные шаги (Applied Steps) выберите шаг Added Custom (Добавленный на-страиваемый). В строке формул получите следующую формулу:

= Table.AddColumn(#"Renamed Columns", "Cartesian", each Keywords),

На этом этапе механизм языка M обращается к ключевым словам для каждой строки. Удивительно, но способ, которым язык M реализует этот шаг вместе с последующим шагом раскрытия, приводит к избыточным загрузкам ключевых слов из внешнего текстового файла.

Для исправления этого шага и возможности принуждения механизма языка M к загрузке таблицы ключевых слов из внешнего текстового файла только один раз можно использовать для ключевых слов М-функцию Table.Buffer. Тем самым вы явно указываете на необходимость сохранять всю таблицу ключевых слов в памяти и гарантируете, что механизм М не будет обращаться к текстовому файлу несколько раз.

СОВЕТ

Несмотря на то что функция Table.Buffer удобна для сокращения времени загрузки в этом упражнении, не следует применять ее всякий раз, когда необходимо объединить две таблицы. Например, при объединении таблицы с помощью команды Объединить запросы (Merge Queries) или Добавить запросы (Append Queries) функция Table.Buffer не оказывает существенной помощи. Однако эта функция может помочь при использовании настраиваемых столбцов, для вычислений которых требуется дос-

туп ко второму источнику данных.

В чем польза функции Table.Buffer при загрузке внешних источников в формулы для пользовательских столбцов? Настраиваемые столбцы предназначены для выполнения вычислений на уровне строк без учета какой-либо эффективности между каждой вычисляемой ячейкой. Более того, по умолчанию механизм языка M предназначен для уменьшения объема требуемой памяти, что приводит к выгрузке таблицы ключевых слов во время вычисления пользовательского столбца. Функция Table.Buffer служит в качестве сигнала для механизма M, который свидетельствует о том, что следует прекратить использование подхода с низким потреблением памяти в этом особом случае, что привело бы к чрезмерному объему операций ввода-вывода (I/O) при загрузке файла Excel Keywords.

2. Для добавления функции Table.Buffer войдите в расширенный редактор и добавьте следующую строку после строки let:

KeywordsBuffer = Table.Buffer(Keywords),

3. Затем замените строку:

#"Added Custom" = Table.AddColumn(#"Renamed Columns", "Cartesian", each Keywords),

на

#"Added Custom" = Table.AddColumn(#"Renamed Columns", "Cartesian", each KeywordsBuffer),

Если загрузить запрос прямо сейчас, то вы можете не заметить уменьшение времени загрузки, поскольку набор данных относительно невелик. Для сравнения времени загрузки запроса Post Topics до и после использования функции Table.Buffer можно загрузить файл рабочей книги C11E02 - Refresh Comparison.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads.

Используя рабочую книгу C11E02 - Refresh Comparison.xlsx, можно получить представление о скорости загрузки запроса с помощью функции Table.Buffer. Как показано на рис. 11.7, можно сравнить время загрузки запроса Post Topics (без функции Table.Buffer) и запроса Post Topics - Faster (с функцией Table.Buffer) и засвидетельствовать сокращение времени загрузки для второго запроса.

4. Для сравнения откройте рабочую книгу, на вкладке Данные (Data) выполните команду Запросы и подключения (Queries & Connections), затем выберите команду Обновить все (Refresh All). Заметим, что запрос Post Topics - Faster загружается быстрее, чем Post Topics.

СОВЕТ

Для оценки уменьшения времени обновления рекомендуется протестировать запрос на достаточно больших наборах данных. Для того чтобы искусственно увеличить имеющийся набор данных, можно умножить элементы таблицы с помощью функции M Table.Repeat. В рабочей книге C11E02 - Refresh Comparison.xlsx можно умножить запрос Microsoft Press Posts 100 раз, применяя следующую функцию в качестве последнего шага запроса: = Table.Repeat(#"Changed Type", 100).

Рис. 11.7. Запрос Post Topics - Faster загружается быстрее, поскольку использует функцию Table.Buffer

В следующем упражнении рассматривается альтернативный метод определения ключевых слов: использование настраиваемой функции.

УПРАЖНЕНИЕ 11.3. Определение ключевых слов с помощью настраиваемой функции

Возможно, вы осознаете, что метод декартовых произведений не интуитивен и существует другой, более прямой подход для обнаружения списка ключевых слов. Это действительно так. В этом разделе показано, как можно вместо декартова произведения применять настраиваемую функцию. И хотя такая функция интуитивно понятна, загружается она намного медленнее, чем решение, связанное с применением декартова произведения. Впрочем, если применить функцию Table.Buffer, то быстродействие данного метода возрастет.

1. Откройте рабочую книгу C11E02 - Refresh Comparison.xlsx и запустите редактор Power Query.

2. Правой кнопкой мыши щелкните на отчете Post Topics на панели Запросы (Queries) и в контекстном меню выберите команду Дублировать (Duplicate).

3. Переименуйте новый запрос в Post Topics with Function и убедитесь, что этот запрос выбран на панели Запросы (Queries).

Затем удалите шаги декартова произведения и определите ключевые слова с помощью настраиваемой функции.

4. На панели Примененные шаги (Applied Steps) удалите последние пять шагов.

5. На вкладке Главная (Home) выполните последовательность команд Создать источник (New Source) Другие источники (Other Sources) Пустой запрос (Blank Query). Переименуйте новый запрос в FnDetectKeywords и выберите команду Расширенный редактор (Advanced Editor) на вкладке Главная (Home).

6. Удалите M-выражение в окне расширенного редактора, скопируйте и вставьте в окне расширенного редактора следующую функцию:

(Message, Keywords)=> Table.SelectRows(Keywords, each Text.Contains(Message, [Keyword]))

Эта функция получает сообщение и таблицу ключевых слов в качестве двух входных данных и возвращает подмножество таблицы Keywords с ключевыми словами, содержащимися в сообщении. Обратите внимание, предполагается, что таблица Keywords (Ключевые слова) содержит столбец Keyword. Щелкните мышью на кнопке Готово (Done) для закрытия окна расширенного редактора.

7. На панели Запросы (Queries) выберите запрос Post Topics with Functions.

8. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). Введите Keywords в поле Имя нового столбца (New Column Name) и введите следующую формулу в поле Пользовательская формула столбца (Custom Column Formula):

= FnDetectKeywords([Lowercased], Keywords)

Новый настраиваемый столбец определяет ключевые слова в каждой ячейке столбца Lowercased.

ПРИМЕЧАНИЕ

Как показано в упражнении 11.2, добавление настраиваемого столбца может занять много времени. В данном примере механизм языка M продолжил перезагрузку ключевых слов из источника данных. Чтобы преодолеть это, примените функцию Table.Buffer к ключевым словам в конце данного упражнения.

9. Щелкните мышью на кнопке OK для закрытия окна диалога Настраиваемый столбец (Custom Column), затем расширьте столбец Keywords. (Не используйте Keywords в качестве префикса для имени нового столбца.)

10. Вы увидите, что новый столбец Keyword (Ключевое слово) содержит обнару-женные темы. В строках без соответствующих ключевых слов отображаются значения null.

11. Щелкните на элементе управления фильтром в заголовке столбца Keyword и выберите команду Удалить пустые (Remove Empty).

12. Удалите столбцы Message и Lowercased и переименуйте столбец Keyword в Topic (Тема).

13. Для загрузки запроса в отчет и сравнения производительности трех запросов в книге выберите команду Обновить все (Refresh All) на вкладке Данные (Data) при открытой панели Книги и подключения (Workbooks & Connections).

Обратите внимание, что последний созданный запрос самый медленный. Он немного медленнее, чем запрос Post Topics, где декартово произведение при-менялось без функции Table.Buffer. Теперь можно исследовать, каким образом функция Table.Buffer улучшает производительность запроса.

14. Откройте окно редактора Power Query и правой кнопкой мыши щелкните на запросе Post Topics with Function на панели Запросы (Queries). В контекстном меню выберите команду Дублировать (Duplicate) и переименуйте новый запрос в Post Topics - Fastest.

15. При выбранном запросе Post Topics - Fastest выполните команду Расширенный редактор (Advanced Editor) на вкладке Главная (Home).

16. В окне расширенного редактора добавьте следующую строку после let:

KeywordsBuffer = Table.Buffer(Keywords),

17. Замените в этой строке Keywords на KeywordsBuffer:

#"Added Custom" = Table.AddColumn(#"Renamed Columns", "Keywords", each FnDetectKeywords([Lowercased], Keywords)),

Обновленная строка кода примет такой вид:

#"Added Custom" = Table.AddColumn(#"Renamed Columns", "Keywords",

each FnDetectKeywords([Lowercased], KeywordsBuffer)),

18. Загрузите запрос в отчет и сравните время, необходимое для загрузки четырех запросов в рабочей книге, выбирая команду Обновить все (Refresh All) на вкладке Данные (Data).

Как показано на рис. 11.8, запрос Book Reviews - Fastest быстрее, чем его связанный с декартовым произведением "побратим" Book Reviews - Faster, который был создан ранее. Однако если не применять в качестве улучшения функцию Table.Buffer, то метод с использованием настраиваемой функции окажется медленнее, как показано на примере запроса Post Topics with Function (рис. 11.8).

Почему запрос Book Reviews - Fastest работает быстрее, чем метод декартовых произведений, если применяется функция Table.Buffer, а без нее работает медлен-нее? Можно заметить, что ссылка на таблицу Keywords (Ключевые слова) реализована внутри функции на шаге 6. Механизм языка M, который разработан по умолчанию для минимизации потребления памяти, полагает, что данные таблицы Keywords могут быть выгружены после выполнения функции для высвобождения неиспользуемой памяти.

Таким образом, таблица Keywords может продолжать перезагрузку для каждого подсчета ячеек в новом столбце. При использовании метода декартовых произведений на таблицу Keywords поддерживается ссылка в объеме основного запроса. В результате механизм языка M загружает таблицу Keywords не так часто, как выполняется ее загрузка при использовании настраиваемой функции.

Если применяется функция Table.Buffer, то механизм языка M "осознает", что необ-ходимо сохранять в памяти таблицу Keywords. Поэтому метод декартова произведения замедляется (даже в том случае, если функция Table.Buffer применяется в методе декартова произведения). Метод декартовых произведений выполняется медленнее, потому что он дороже с точки зрения потребления памяти и шагов преобразования. Механизм дублирует каждую строку в таблице рецензий на рабочие книги по числу строк в таблице Keywords, а затем применяет соответствующий фильтр. Это дополнительные шаги, которые не выполняются при обращении к настраиваемой функции в запросе Book Reviews - Fastest, что влияет на общую производительность.

Файлы решения можно просмотреть в рабочей книге C11E03 - Solution.xlsx, которая доступна на сайте https://aka.ms/DataPwrBIPivot/downloads.

Выбор метода: статический поиск, декартово произведение или настраиваемая функция

При выполнении упражнений 11.1, 11.2 и 11.3 были продемонстрированы три метода определения ключевых слов. Первый предусматривает применение статического краткого списка ключевых слов в качестве условий в окне диалога Добавление условного столбца (Add Conditional Column). Второй метод заключается в применении декартова произведения для динамической загрузки длинного списка ключевых слов; показано, что функция Table.Buffer позволяет уменьшить время загрузки этого запроса. Наконец, третий метод предполагает применение настраиваемой функции; известно, что этот метод наиболее медленный, он может оказаться и самым быстрым, если применить функцию Table.Buffer.

В реальных задачах чаще всего при обработке коротких списков применяется первый метод. Для динамических или длинных списков, если написание настраиваемых М-функций не является вашим любимый занятием, можно использовать декартово произведение. Если вы достаточно полно владеете возможностями языка M, то можно применять метод, основанный на настраиваемых функциях, но не забудьте добавить функцию Table.Buffer.

Г") СОВЕТ

Методы обнаружения ключевых слов в текстовых столбцах также эффективны при фильтрации данного текста по ключевым словам. Как вы помните, столбец Topic создан в предыдущих упражнениях, затем выполнена фильтрация пустых значений. Такие методики могут быть полезны при выполнении фильтрации.

Разбиение слов

Вспомните четвертое предупреждение из упражнения 11.2: слова, содержащиеся в качестве подстрок более длинных слов, могут ошибочно попадать в результат поиска. В данном разделе показано, каким образом можно разбивать текстовые столбцы на слова, что позволит ускорить процесс обнаружения ключевых слов. Разделение слов также помогает в подсчете вхождений слов и визуализации данных в облаках слов Power BI.

Чтобы улучшить обнаружение ключевых слов и получить новое представление о текстовых столбцах, далее будет продемонстрировано, как разбить текстовые столбцы на слова и отобразить результаты в сводных таблицах и диаграммах Excel, а также в облаках слов Power BI.

УПРАЖНЕНИЕ 11.4. Простое разбиение слов

В этом упражнении загружаются посты Microsoft Press на Facebook и выполняется разбивка сообщений на слова. Затем вы узнаете, как можно игнорировать общие слова, такие как the, and, in и on.

Часть 1. Разбиение слов с помощью пробелов

В этом разделе вы начнете с интуитивно понятного подхода к нашей задаче и разделите слова пробелами, а затем перейдете к более эффективному решению в последующих разделах.

1. Загрузите файл рабочей книги C11E01.xslx с сайта https://aka.ms/DataPwrBIPivot/ downloads и сохраните его в папке C:\Data\C11\.

2. Откройте пустую новую рабочую книгу или новый отчет Power BI Desktop.

В Excel: откройте новую рабочую книгу Excel и на вкладке Данные (Data) вы-полните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: откройте новый отчет и выполните команды Получить данные (Get Data) Excel Подключиться (Connect).

3. Выберите файл C11E01.xslx, находящийся в папке C:\Data\C11\, и щелкните на кнопке Импорт (Import).

4. В окне диалога Навигатор (Navigator) выберите вкладку Sheet1 и щелкните на кнопке Изменить (Edit).

5. Переименуйте запрос на Microsoft Press Posts.

Для формирования ссылки на запрос Microsoft Press Posts для создания новой таблицы, составленной из ID постов и слов, на панели Запросы (Queries) пра-вой кнопкой мыши щелкните на запросе Microsoft Press Posts и в контекстном меню выберите команду Ссылка (Reference).

6. Переименуйте новый запрос в All Words.

7. При выделенном запросе All Words на панели Запросы (Queries) выберите команду Выбор столбцов (Select Columns) на вкладке Главная (Home). В окне диалога Выбор столбцов (Choose Columns) выделите столбцы Post ID и Message и затем щелкните мышью на кнопке OK.

8. Правой кнопкой мыши щелкните на столбце Message (Сообщения), в контекстном меню выберите команду Разделить столбец (Split Column), затем выполните команду По разделителю (By Delimiter). В качестве альтернативы на вкладке Главная (Home) выполните команду Разделить столбцы (Split Columns), а затем выберите команду По разделителю (By Delimiter).

Откроется окно диалога Разделить столбец по разделителю (Split Column By Delimiter). Как было показано в главе 10, разбиение текстового столбца на столбцы (что задано по умолчанию) плохо масштабируется, тем более что в столбце Message имеется большое число слов.

9. Выполните следующие шаги в окне диалога Разделить столбец по разделителю (Split Column By Delimiter), чтобы разделить на строки столбец Message:

а. В качестве разделителя выберите пробел.

б. Откройте раздел Расширенные параметры (Advanced Options).

в. Выберите переключатель Строки (Rows).

г. Щелкните мышью на кнопке OK для закрытия данного окна диалога.

10. Переименуйте новый столбец в Word.

Наивный пользователь может подумать, что описанное преобразование полностью решает проблему, но многие слова содержат конечные знаки препинания, что приведет к множественным вариациям одного и того же слова. Например, можно найти много вариантов слова "excel" с запятой, восклицательным знаком или двоеточием. Для эффективного решения этой проблемы необходимо отделить слова от знаков препинания.

Часть 2. Отделение слов от знаков препинания

В этом разделе применяется более разумный подход к разделению слов и учитывается любая пунктуация, которая может находиться в сообщениях. Продолжите упражнение 11.4, выполняя следующие шаги:

11. Создайте пустой запрос, назовите его Punctuations и в строку формул редактора Power Query введите следующий код:

= {" " "~" " " " " "?" "!" "(" ")" "[" "]" "{" "}" "@" "#" "$" "%" "А"

{ , , , , . , • , • , ( , ) , [ , ] , { , } ,@ , # , $ , 0 , ,

"&" "\*" "—" "+" "=" ":" ";" "|" "<" ">" "/" """ """ "\" """" "#(tab)"

"#(cr)", "#(lf)"}

Это выражение создает список знаков препинания, а также специальных символов для обнаружения вкладок, возвратов каретки и символов перевода строк.

12. На панели Запросы (Queries) выберите запрос All Words (Все слова).

13. На панели Примененные шаги (Applied Steps) выберите шаг Split Column By Delimiter (Разделить столбец по разделителю) и проверьте строку формул. Как показано на рис. 11.9, выражение M окажется довольно сложным.

Обратите внимание, что в строке формул имеется двухшаговая формула, которая разбивает столбец Message на объекты списка (List), затем расширяет его.

Рис. 11.9. Операция Split Column by Delimiter не обрабатывает знаки препинания. Чтобы решить эту проблему, необходимо изменить формулу

Вам необязательно вникать в детали, можно просто сосредоточиться на важной части, которая показана на рис. 11.9:

Splitter.SplitTextByDelimiter(" ", QuoteStyle.Csv)

Измените выделенный раздел следующим образом:

Splitter.SplitTextByAnyDelimiter(Punctuations, QuoteStyle.Csv)

После этой модификации можно выполнять разбиение по любому разделителю, определенному в списке Punctuations (Пунктуация). Правда, несложно?

Г") СОВЕТ

Если необходимо разбить столбец с учетом нескольких разделителей, можно исполь-зовать команду Разделить столбец по разделителю (Split Column By Delimiter) в ин-терфейсе пользователя для разделения с помощью одного разделителя, а затем модифицировать эту M-формулу. Замените функцию Splitter.SplitTextByDelimiter на Splitter.SplitTextByAnyDelimiter и задайте список текстовых значений вместо одного разделителя. Этот подход пригоден при разбиении столбца на столбцы или строки.

Благодаря разделению с помощью любого знака препинания теперь имеется большое число строк с пустыми значениями. Далее выполним их фильтрацию.

14. Выберите последний шаг на панели Примененные шаги (Applied Steps) и щелкните мышью на элементе управления фильтром в заголовке столбца Word. На панели Фильтр (Filter) выберите команду Удалить пустые (Remove Empty).

15. Закройте редактор Power Query и загрузите в модель данных запросы Microsoft Press Posts и All Words. Убедитесь, что запрос Punctuations не загружается, и создайте отношение между полями Post ID и двумя таблицами. (Аналогично тому, как это сделать в упражнении 11.2, когда устанавливалась связь Post ID с Microsoft Press Posts и Post Topics.)

16. Сохраните отчет. Он понадобится вам в следующем разделе.

Если применить столбец Word в All Words в качестве строк и значений в сводной таблице, как показано на рис. 11.10, или в гистограмме Power BI, то можно заметить, что список наиболее распространенных слов довольно прост. Наиболее часто встречающиеся слова — это http, за которыми следуют the и and.

Рис. 11.10. Наиболее часто встречающиеся слова в постах Microsoft Press — http, за которыми следуют the и and. Это и неудивительно

Почему именно http встречается наиболее часто? Поскольку сообщения в постах Microsoft Press включают много URL-ссылок, которые начинаются с префикса http://. Разве не следует избегать разбиения подобных фраз и не лучше ли рассматривать их как отдельные слова?

Часть 3. Разделение слов пробелами и удаление пунктуации

В этом разделе будем обрабатывать знаки препинания различным образом. Вместо разбивки сообщений с помощью любого из знаков препинания, как это было выполнено в предыдущем разделе, будем разбивать сообщения только при появлении пробела, знака табуляции, возврата каретки и перевода строки. Затем, если знаки препинания обнаружатся в начале или конце слов, вы их обрежете.

1. Если вы не сохранили рабочую книгу или отчет из упражнения 11.4, заново повторите шаги 1-16.

2. Запустите редактор Power Query. В панели Запросы (Queries) правой кнопкой мыши щелкните на запросе All Words и в контекстном меню выберите команду Дублировать (Duplicate).

3. Переименуйте в новый запрос All Words - Trim Punctuations.

4. При выбранном новом запросе на панели Примененные шаги (Applied Steps) выберите шаг Split Column by Delimiter (Разделить столбец по разделителю).

5. В строке формул удалите Punctuations в следующем разделе кода:

Splitter.SplitTextByAnyDelimiter(Punctuations, QuoteStyle.Csv)

Вместо Punctuations введите список {" ", "#(tab) ", "#(cr) образом:

Сообщения теперь разделены пробелами, знаком табуляции, символом возврата каретки и перевода строк, поэтому можно разыскивать начальную и конечную пунктуацию в значениях разделения и обрезать ее.

6. Если шаг Split Column by Delimiter остается выбранным на панели Примененные шаги (Applied Steps), выделите столбец Message на панели предварительного просмотра и на вкладке Преобразование (Transform) выберите команду Формат (Format), затем выполните команду Усечь (Trim).

7. В окне диалога Вставить шаг (Insert Step) выберите команду Вставить (Insert) для добавления нового шага. Шаг обрезки по умолчанию удаляет пробелы в начале и конце текста. Но можно изменить его, что позволит обрезать знаки препинания.

Обратите внимание на следующую формулу в строке формул:

= Table.TransformColumns(#"Split Column by Delimiter ",{{"Message", Text.Trim, type text}})

Измените Text.Trim на each Text.Trim(\_, Punctuations), как показано в этой формуле:

= Table.TransformColumns(#"Split Column by Delimiter ", {{"Message",each Text.Trim(\_, Punctuations), type text}})

Вместо заданной по умолчанию функции Text.Trim можно явно определить но-вую одноименную функцию Text.Trim с другими аргументами. Необходимую функцию формирует ключевое слово each. Напомним, что в главе 9 было показано, что с помощью each можно создавать простых функций. Вторым аргументом функции Text.Trim определяется список символов или строк для обрезки. В данном случае в качестве аргумента применяется список Punctuation, речь о котором шла в предыдущем упражнении.

ПРИМЕЧАНИЕ

Для повышения эффективности, поскольку на шаге разделения уже использованы пробел, символ табуляции, символ возврата каретки и символ новой строки, вы можете сформировать дублирующий список пунктуации и удалить эти символы из нового списка. Затем можно применить новый список в качестве второго аргумента функции Text.Trim.

Теперь при создании сводной таблицы на основе запроса All Words - Trim Punctuations заметьте, что как показано на рис. 11.11, http более не встречается.

Данный отчет желательно сохранить, так как он понадобится в следующем упражнении. Файлы решения C11E04 - Solution.xlsx и C11E04 - Solution.pbix доступны на сайте https://aka.ms/DataPwrBIPivot/Downloads.

УПРАЖНЕНИЕ 11.5. Фильтрация по стоп-словам

В упражнении 11.4 было показано, как разбивать сообщения на слова, но в итоге получен лишь тривиальный результат. Неудивительно, что наиболее часто встречающиеся слова — это союзы, предлоги и артикли, как показано на рис. 11.11. Разве не удобнее отфильтровать эти общие слова и сосредоточить внимание на более важных существительных, глаголах и прилагательных в сообщениях?

Чтобы отфильтровать общие слова или, как мы называем их в этом разделе, стоп- слова (термин, который обычно употребляют специалисты по информатике), можно сформировать список слов в отдельном текстовом файле, загрузить их в запрос и выполнить фильтрацию. В данном упражнении показано, как это выполнить.

1. Откройте решение из упражнения 11.4 либо откройте рабочую книгу C11E04 - Solution.xlsx или отчет Power BI C11E04 - Solution.pbix.

2. Запустите редактор Power Query, на панели Запросы (Queries) правой кнопкой мыши щелкните на запросе All Words - Trim Punctuations и в контекстном меню выберите команду Ссылка (Reference).

3. Если новый запрос создан, назовите его No Stop Words.

4. Загрузите файл Stop Words.txt с сайта https://aka.ms/DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C11\.

5. Импортируйте файл Stop Words.txt, выбирая команды Создать источник (New Source) Файл (File) и команду Текст/CSV (Text/CSV) на вкладке Главная (Home) редактора Power Query.

6. Убедитесь, что новый запрос называется Stop Words, и переименуйте соответ-ствующий столбец в Stop Word. На панели предварительного просмотра обратите внимание, что все стоп-слова в наборе данных записаны с применением нижнего регистра символов. В качестве хорошей практики следует явно указывать их строчными буквами и в будущем избегать ситуаций, когда при импорте заглавных символов стоп-слова будут игнорироваться из-за чувствительной к регистру логики, предполагающей, что все буквы строчные.

Выберите столбец Stop Word, на вкладке Преобразование (Transform) выберите команды Формат (Format) нижний регистр (Lowercase).

Затем отфильтруем текст по стоп-словам. В запросе No Stop Words столбец Word уже включает все слова в нижнем регистре из сообщений. Объединим слова и стоп-слова с помощью левого антислияния (left anti merge), которое отфильтровывает все слова в запросе No Stop Words, совпадающие со стоп- словами.

7. Выполните следующие шаги для фильтрации по стоп-словам:

а. На панели Запросы (Queries) выберите запрос No Stop Words.

б. На вкладке Главная (Home) выберите команду Объединить запросы (Merge Queries).

в. Если открыто окно диалога Слияние (Merge), в таблице No Stop Words щелкните на заголовке столбца Word и удостоверьтесь, что этот столбец выделен.

г. В раскрывающемся меню, ниже таблицы No Stop Words, выберите вариант Stop Words.

д. В таблице Stop Words щелкните мышью на заголовке столбца Stop Word и удостоверьтесь, что он выделен.

е. В раскрывающемся меню Тип соединения (Join Kind) выберите вариант Ан-тисоединение слева (только строки в первой таблице) (Left Anti (Rows Only In First)), как показано на рис. 11.12, и щелкните на кнопке OK для закрытия данного окна диалога.

При операции слияния столбец Stop Words объединяется вместе с объектами таблицы. Если щелкнуть на область любой ячейки из столбца Stop Words (но не на ссылку таблицы, которая сводится к одному шагу), то можно заметить, что все таблицы содержат нулевые значения. Нулевые значения отображаются вследствие выполнения операции левого антислияния, которая не возвращает строк из второй таблицы слияния. Из запроса Stop Words нечего извлекать.

8. Удалите столбец Stop Words и загрузите отчет. Убедитесь, что запрос Stop Words не загружен, а No Stop Words загружен в модель данных.

По завершении предыдущих шагов столбец Word из таблицы No Stop Words можно применить к сводной таблице и подсчитать слова вхождения (рис. 11.13). При фильтрации Stop Words можно заметить интересные особенности.

Вот два примера, которые следуют из просмотра списка 10 наиболее распространенных слов:

■ Одним из наиболее распространенных слов является "2016", которое отража-ет большое количество основных выпусков Microsoft в течение 2016 года (Office 2016, Excel 2016, Windows Server 2016 и т. д.).

■ Другое распространенное слово — "50", которое упоминается во многих сообщениях, рекламирующих книги со скидкой 50%. Знаете ли вы, что по субботам вы можете получить 50% скидки на книги Microsoft Press? Основываясь на втором наиболее часто встречающемся слове, можно перейти к данным и найти распространенное сообщение "Экономьте 50% в субботу!"

Рис. 11.13. Сводная таблица выводит наиболее часто встречающиеся слова в постах Microsoft Press без стоп-слов

Сохраните рабочую книгу или отчет Power BI для использования в следующем упражнении. Можно просмотреть решения этого упражнения, для чего загрузите рабочую книгу C11E05 - Solution.xlsx или C11E05 - Solution.pbix с сайта https:// aka.ms/DataPwrBIPivot/downloads.

УПРАЖНЕНИЕ 11.6. Поиск по ключевым словам с помощью разделенных слов

Напомним, что в упражнениях 11.2 и 11.3 ключевые слова разыскивались в постах Microsoft Press на Facebook с помощью оператора contains языка М. Один из недостатков этого подхода состоит в том, что можно прийти к ложному обнаружению слов, если ключевое слово является подстрокой другого слова. Например, ключевое слово excel может быть подстрокой слова excellent. В этом упражнении для повышения точности поиска по ключевым словам применяется методика разделения слов из упражнения 11.5.

Часть 1. Использование внутренних запросов на слияние для сопоставления столбцов

Выполнение этого упражнения начнем с рассмотрения полезной техники для фильтрации таблицы по нескольким значениям путем слияния запросов.

1. Откройте рабочую книгу или отчет Power BI из упражнения 11.5 или загрузите файл C11E05 - Solution.xlsx или C11E05 - Solution.pbix с сайта https://aka.ms/ DataPwrBIPivot/downloads и откройте его.

2. Запустите редактор Power Query. Выберите команду Создать источник (New Source) на вкладке Главная (Home) и импортируйте файл Keywords.txt, выби-рая команды Файл (File) Текст/CSV (Text/CSV). Если этот файл загружен в упражнении 11.2, он должен находиться в папке C:\Data\C11\Keywords.txt.

3. После открытия файла Keywords.txt в окне диалога для его просмотра щелкните мышью на кнопке OK, чтобы закрыть это окно диалога.

4. Если запрос Keywords загружен, переименуйте столбец этого запроса на Keyword.

5. Чтобы все ключевые слова находились в нижнем регистре, выделите столбец Keyword и затем на вкладке Преобразование (Transform) выберите команды Формат (Format) нижний регистр (Lowercase).

Затем создадим новый запрос, Post Topics, на основании запроса No Stop Words, который уже включает ID постов и ассоциируемые с ними слова.

6. На панели Запросы (Queries) правой кнопкой мыши щелкните на запросе No Stop Words и в контекстном меню выберите команду Ссылка (Reference). Пе-реименуйте новый запрос на Post Topics.

7. Для фильтрации всех слов, отображаемых в запросе Keywords, выполните сле-дующие шаги:

а. На панели Запросы (Queries) выберите запрос Post Topics.

б. На вкладке Главная (Home) выберите команду Объединить запросы (Merge Queries).

в. Если окно диалога Слияние (Merge) открыто, в таблице Post Topics щелкните на заголовке столбца Word и удостоверьтесь, что этот столбец теперь выделен.

г. В раскрывающемся меню под таблицей Post Topics выберите таблицу Keywords.

д. В таблице Keywords щелкните на заголовке столбца Keyword и удостоверь-тесь, что он выделен.

е. В раскрывающемся меню Тип соединения (Join Kind) выберите пункт Внут-реннее (только совпадающие строки) Inner (Only Matching Rows)) и щелкните на кнопке OK для закрытия окна диалога.

8. Переименуйте столбец Word в Topic.

9. Чтобы избежать ситуаций, когда одно ключевое слово несколько раз упоминается в одном сообщении, можно выделить столбцы Post ID и Topic, а затем на вкладке Главная (Home) выбрать команду Удалить строки (Remove Rows), после чего выполнить команду Удалить дубликаты (Remove Duplicates). Теперь каждая тема упоминается только один раз для каждого ID поста.

На этом этапе можно и завершить упражнение. Но появилась новая проблема: заметили ли вы, что некоторые ключевые слова в файле Keywords.txt состоят из двух слов, например power query, power bi и power pivot? Поскольку это упражнение начато с разделенных слов, вы не сможете обнаружить ключевые слова, состоящие из двух слов. Ваши сообщения уже разделены. В следующем разделе показано, как разрешить это затруднение.

Сохраните свою книгу или отчет Power BI. Вы будете использовать его для определения ключевых слов, состоящих из нескольких слов, в следующей части этого упражнения.

Часть 2. Обнаружение составных ключевых слов

Для определения таких ключевых слов, как power query и power bi в постах Microsoft Press в этом упражнении, нужно временно заменить разделительный пробел другим символом, например подчеркиванием, но только при обнаружении ключевых слов, состоящих из двух слов в столбце Message. Затем можно применить разделение на пробелы, которые больше не будут влиять на ключевые слова из двух слов. Как только сообщения разбиты на слова, можно преобразовать ключевые слова, состоящие из двух слов, обратно в форму, разделенную пробелами.

На рис. 11.14 представлено решение на более высоком уровне. Используется таблица преобразования со столбцами Source (Источник) и Destination (Цель) с ключевыми словами из двух слов с разделенными пробелами значениями в Source и значениями с подчеркиванием в Destination. После формирования таблицы преобразования необходимо заменить все ключевые слова из Source на ключевые слова в Destination (шаг 1 на рис. 11.14), применить разделение к сообщениям (шаг 2) и заменить каждое подчеркивание пробелом (шаг 3).

Message 4 Message 4 Word 4 Word

great book great book great great

power bi power bi book book

power pivot power pivot power bi power bi

power query power\_query power pivot power pivot

power\_query power query

1. Заменить источник на цель в столбце Message

3. Заменить подчеркивание пробелом

2. Разбить сообщение на слова в столбце Message

I Source Destination I

power bi power bi

power pivot power\_pivot

power query power\_query

Таблица преобразования Keywords, содержащая составные ключевые слова

Рис. 11.14. Можно обнаружить ключевые слова из двух слов в разделенных словах,

используя таблицу преобразования,

следующую за данной последовательностью преобразования

Как показано на рис. 11.14, вы запускаете это решение, создавая таблицу преобразования. Можно сформировать эту таблицу как ссылку из запроса Keywords (Ключевые слова). Продолжите это упражнение, выполняя следующие действия.

10. При открытом редакторе Power Query выберите запрос Keywords на панели Запросы (Queries) и затем в контекстном меню выберите команду Ссылка (Reference). Переименуйте новый запрос в Conversion Table.

11. Для сохранения ключевых слов, которые включают пробел в запросе Con-version Table, выполните такие шаги при выделенной таблице:

а. На вкладке Преобразование (Transform) выполните команды Формат (Format) Усечь (Trim). Даже если текстовый файл Keywords не содержит начальных или конечных пробелов, рекомендуется обрезать его перед выполнением следующего шага.

б. Щелкните мышью на элементе управления фильтром в заголовке столбца Keyword. На панели Фильтр (Filter) выберите команды Текстовые фильтры (Text Filters) Содержит (Contains).

в. В открытом окне диалога Фильтрация строк (Filter Rows) введите пробел в текстовое поле, которое размещается справа от раскрывающегося меню Содержит (Contains), и щелкните на кнопке OK для закрытия этого окна диалога. На панели предварительного просмотра отобразятся девять строк с ключевыми словами, состоящими из двух слов.

12. Переименуйте столбец Keyword на Source.

13. Для дублирования столбца Source для столбца Destination с одинаковыми зна-чениями, разделенными подчеркиванием вместо пробелов, выполните следующие действия:

а. Выделите столбец Source. На вкладке Добавление столбца (Add Column) выберите команду Создать дубликат столбца (Duplicate column).

б. Переименуйте столбец на Destination.

в. Выделите столбец Destination, затем выберите команду Замена значений (Replace Values) на вкладке Главная (Home).

г. После открытия окна диалога Замена значений (Replace Values) выберите символ пробела ( ) в качестве Значение для поиска (Value to Find) и выбе-рите символ подчеркивания (\_) в качестве Заменить на (Replace With). За-тем щелкните мышью на кнопке OK для закрытия этого окна диалога.

Запрос Conversion Table подготовлен (рис. 11.14), и теперь следует узнать, как им пользоваться. На следующих шагах необходимо заменить каждое значение Source на соответствующее значение Destination для каждого значения столбца Message перед выполнением шага Split. Замена отдельного значения легко выполняется в редакторе Power Query, но в данном случае имеется последовательность пар Source/Destination, которые необходимо заменять.

К сожалению, в редакторе Power Query отсутствует кнопка пользовательского интерфейса для применения нескольких замен строк. Кроме того, функция M

Table.ReplaceValue не имеет необходимого нам соответствующего свойства. Вместо этого можно воспользоваться усовершенствованной методикой, основанной на применении функции List.Accumulate для выполнения нескольких замен текста из Source на значения Destination.

Напомним, что функция List.Accumulate, которая рассмотрена в главе 9, получает список в качестве входных данных, начальный объект и функцию аккумулятора. Функция аккумулятора должна иметь объект состояния и текущий объект (который является членом списка ввода на текущей итерации). Эта функция возвращает объект состояния после его преобразования. В заключение функция List.Accumulate возвращает накопленный результат после того, как функция аккумулятора перебрала список. Прежде чем продолжить, можно вернуться к главе 9, просмотреть материал, связанный с функцией List.Accumulate, и убедиться в том, что вы понимаете выражения M, которые собираетесь использовать. Даже если вы не знакомы с языком M, все равно можно воспользоваться выражениями, поскольку совсем не обязательно полностью понимать нюансы выражений.

Прежде чем реализовать функцию List.Accumulate, рассмотрим, каким образом можно преобразовать запрос таблицы преобразования в список пар Source/ Destination (Источник/назначение), которые можно применять в качестве входных данных для функции List.Accumulate. На рис. 11.15 показано, как сформировать список с помощью функции Table.ToRecords, которая разработана для возврата списка записей (одна запись на одной строке исходной таблицы).

I Source Destination 1

power bi power bi

power pivot power pivot

power query power\_query

Table.ToRecords(#"Conversion Table"!

{

(Source = "power bi". Destination = "power\_bi"],

[Source = "power pivot", Destination = "power\_pivot"),

[Source = "power query", Destination = "power\_query"] )

Список записей

Рис. 11.15. Функция M Table.ToRecords может трансформировать таблицу преобразования

в список записей

Теперь, когда известно, как составить список записей, можно узнать, какая функция подойдет для реализации этапа аккумуляции (накопления) для функции List.Accumulate. В основном эта функция получает сообщение Facebook Post и заменяет текущий текст из столбца Source соответствующим текстом в столбце Destination . Например, power bi заменяется на power\_bi, что позволит избежать в дальнейшем разбиения ключевого слова power bi на два отдельных слова.

Вот как выглядит функция аккумулятора:

message, conversionRecord) =>

Text.Replace(

message,

conversionRecord[Source], conversionRecord[Destination]

)

Назначением этой функции является применение функции Text.Replace к сооб-щению поста Facebook (первый аргумент) и замена Source (второй аргумент) на Destination (третий аргумент).

14. Теперь у нас имеются все части, необходимые для создания полной формулы List.Accumulate:

List.Accumulate(

Table.ToRecords(#"Conversion Table"),

[Message],

(message, conversionRecord) =>

Text.Replace(

message,

conversionRecord[Source], conversionRecord[Destination]

)

)

Если формула применяется в пользовательском столбце, то она выполняет перебор каждой строки и замену значения в столбце Message, выполняя просмотр всех ключевых слов в столбце Source в сообщении и замену их одно за другим соответствующими ключевыми словами из столбца Destination.

Формула List.Accumulate на шаге 14 применяется внутри настраиваемой функции столбца. Она работает для каждой строки в постах Microsoft Press. Для каждой строки выполняется итерация по списку пар ключевых слов Source/Destination из таблицы преобразования. Начальным состоянием (второй аргумент функции List.Accumulate) является значение в столбце Message. Функция аккумулятора (третий аргумент функции List.Accumulate) содержит состояние и текущие аргументы (называются message и conversionRecord). Аргумент состояния содержит значение текущей ячейки в столбце Message. Текущим аргументом функции- аккумулятора является текущая запись из пары Source/Destination. Функция аккумулятора заменяет в сообщении текущее ключевое слово из Source с помощью его соответствия в Destination.

На очередной итерации для функции List.Accumulate оценивается следующая пара Source/Destination.

С помощью следующих шагов выражение List.Accumulate используется для замены пробелов подчеркиваниями в ключевых словах сообщений, это происходит перед выполнением шага Split.

15. При открытом окне редактора Power Query на панели Запросы (Queries) выберите запрос All Words - Trim Punctuations. Этот запрос выполнит разделение слова. На панели Примененные шаги (Applied Steps) выберите шаг Lowercased Text, который находится на один шаг раньше, чем шаг Split Column By Delimiter.

16. При выбранном шаге Lowercased Text на панели Примененные шаги (Applied Steps) выберите команду Настраиваемый столбец (Custom Column) на вкладке Добавление столбца (Add Column). При открытом окне диалога Вставить шаг (Insert Step) выполните команду Вставить (Insert).

17. Внесите следующие изменения в открытое окно диалога Настраиваемый столбец (Custom Column):

а. Введите Modified Message в поле Имя нового столбца (New Column Name).

б. Скопируйте и вставьте формулу List.Accumulate из шага 14 в окно Пользова-тельская формула столбца (Custom Column Formula) и щелкните мышью на кнопке OK для закрытия этого окна диалога.

Теперь новый столбец Modified Message (Измененное сообщение) включает посты Facebook с замененными значениями. Для тестирования рабочего со-стояния функции List.Accumulate можно добавить временный шаг, который вы-полняет фильтрацию столбца Message, если он включает слова power query и power pivot. При выполнении этих действий в панели предварительного про-смотра обнаружится следующий текст (в столбце Message):

"microsoft office 2016 step by step ... solve business intelligence problems with power pivot and power query ..."

Если просмотреть столбец Modified Message, то можно заметить, что сообще-ние заменяется на следующее:

"microsoft office 2016 step by step . solve business intelligence problems with power\_pivot and power\_query ."

18. Если добавлен временный фильтр, удалите его, отменяя шаг Filter Rows1 после шага Added Custom на панели Примененные шаги (Applied Steps).

19. Выберите столбец Added Custom на панели Примененные шаги (Applied Steps) и удалите столбец Message. Когда окно диалога Вставить шаг (Insert Step) откроется вновь, выберите команду Вставить (Insert).

20. Переименуйте столбец Modified Message в Message. После повторного откры-тия окна диалога Вставить шаг (Insert Step) выберите команду Вставить (Insert).

Теперь все ключевые слова, состоящие из двух слов, содержат подчеркивание в качестве разделителя. Они защищены от разделения на следующем шаге преобразования, который сформирован на панели Примененные шаги (Applied Steps).

21. Для продвижения вперед по этапам преобразования и восстановления символа пробела на панели Запросы (Queries) выберите запрос No Stop Words, а на панели Примененные шаги (Applied Steps) выберите последний шаг. Здесь необходимо исправить слова и заменить знак подчеркивания пробелом.

22. Для удаления символа подчеркивания имеются две возможности. Первый вариант относительно прост: если можно предположить, что отсутствуют важные слова с подчеркиванием, можно применить команду Замена значения (Replace Value) для столбца Word и заменить подчеркивания пробелами.

Второй вариант более эффективен, если ваши данные содержат слова с подчеркиванием (не ключевые слова). Для удаления подчеркивания только для ключевых слов с подчеркиванием выполните следующие действия:

а. При выбранном запросе No Stop Words выберите команду Настраиваемый столбец (Custom Column) на вкладке Добавление столбца (Add Column).

б. В окне диалога Настраиваемый столбец (Custom Column) введите название Modified Word в поле Имя нового столбца (New Column Name).

в. Скопируйте и вставьте следующий код в окно Пользовательская формула столбца (Custom Column Formula):

List.Accumulate(

Table.ToRecords(#"Conversion Table"),

[Word],

(word, conversionRecord) =>

if word = conversionRecord[Destination] then conversionRecord[Source]

else word

)

Щелкните мышью на кнопке OK для закрытия окна диалога Настраиваемый столбец (Custom Column).

Приведенная формула немного проще, чем функция List.Accumulate на шаге 14. Она будет применяться в составе пользовательской формулы столбца, поэтому будет выполняться для каждой строки в постах Microsoft Press. Как и в случае с функцией List.Accumulate на шаге 14, выполняется итерация по списку записей пар ключевых слов Source/Destination. Начальным состоянием этой функции (второй аргумент List.Accumulate) является значение в столбце Word. Функция аккумулятора (третий аргумент функции List.Accumulate) включает состояние и текущие аргументы (под названием word и conversionRecord). Аргумент состояния содержит значение в текущей ячейке столбца Word. Текущим аргументом функции аккумулятора является текущая запись пары Source/Destination. Функция аккумулятора сравнивает текущее слово и слово в столбце Destination (ключевые слова, разделенные подчеркиванием). Если соответствие установлено, значение возвращается столбцу Source, потому что необходимо вернуть ключевое слово обратно из Destination в Source. Если же ключевое слово не соответствует Destination, то сохраняется исходное слово. На очередной итерации для функции List.Accumulate оценивается следующая пара Source/Destination.

23. Для тестирования формулы отфильтруйте столбец Word с помощью ключевого слова power\_bi и заметьте, что новый столбец Modified Word содержит power bi. После подтверждения, что предыдущий шаг выполнен правильно, удалите шаг фильтра.

24. Удалите столбец Word и переименуйте столбец Modified Word в Word.

25. На панели Запросы (Queries) выберите запрос Post Topics и щелкните на эле-менте управления фильтром столбца Topic как на временном шаге. Вы увидите в панели Фильтр (Filter), что найдены все ключевые слова, составленные из двух слов.

И наконец, перед загрузкой в отчет запроса Post Topics можно уменьшить время загрузки, применяя функции Table.Buffer или List.Buffer. Напомним, что на шагах 14 и 22 добавлялся настраиваемый столбец, который обращался к таблице преобразования. В упражнениях 11.2 и 11.3 показано, что механизм языка M может обращаться к внешним источникам несколько раз, при ссылках на внешние запросы внутри настраиваемого столбца. На следующих шагах будем включать функцию List.Buffer для сокращения времени загрузки, а также для "исправления" данной проблемы.

26. При выбранном запросе All Words - Trim Punctuations выполните команду Расширенный редактор (Advanced Editor) на вкладке Главная (Home).

Вставьте следующую новую строку перед строкой, которая начинается с #”Added Custom”:

BufferedList = List.Buffer(Table.ToRecords(#"Conversion Table")),

На шаге #”Added Custom” замените Table.ToRecords(#”Conversion Table”), первый аргу-мент функции List.Accumulate, на BufferedList.

Вот как выглядит полный шаг #”Added Custom” после обновления:

#"Added Custom" = Table.AddColumn(#"Lowercased Text", "Modified Message",

each List.Accumulate(

BufferedList,

[Message],

(message, conversionRecord) =>

Text.Replace( message, conversionRecord[Source], conversionRecord[Destination]

)

)

),

27. Щелкните мышью на кнопке Готово (Done), чтобы закрыть окно расширенного редактора.

28. При выделенном запросе No Stop Words выберите команду Расширенный редактор (Advanced Editor) на вкладке Главная (Home).

Вставьте следующую новую строку перед строкой, которая начинается с #"Added

Custom":

BufferedList = List.Buffer(Table.ToRecords(#"Conversion Table")),

На шаге #”Added Custom” замените Table.ToRecords(#”Conversion Table”) в первом аргу-менте функции List.Accumulate на BufferedList.

Вот полное выражение для шага #”Added Custom” после обновления:

#"Added Custom" = Table.AddColumn(#"Removed Columns", "Modified Word",

each List.Accumulate(

BufferedList,

[Word],

(word, conversionRecord) =>

if word = conversionRecord[Destination] then

conversionRecord[Source]

else

word

)

),

Теперь можно загрузить запрос Post Topics и сформировать сводную таблицу и диаграмму, из которых видно, какие темы наиболее популярны на странице Microsoft Press Facebook (рис. 11.16).

Файлы решения C11E06 - Solution.xlsx или C11E06 - Solution.pbix доступны на сайте https://aka.ms/DataPwrBIPivot/downloads.

УПРАЖНЕНИЕ 11.7. Создание "облаков" слов в Power BI

В заключение этой главы можно рассмотреть, каким образом Power BI помогает визуализировать текстовые столбцы в "облаке" слов. В предыдущем разделе было показано, как можно разделять и считывать слова, а также каким образом игнорировать в отчете стоп-слова. Оказывается, для небольших наборов данных можно загружать все сообщения в "облако" слов в Power BI, не разбивая их на слова.

В данном упражнении рассматриваются пользовательские визуальные элементы для облаков слов и показано, как можно игнорировать стоп-слова без необходимости выполнять разбивку слова. Следуя приведенным далее инструкциям, можно создать "облако" слов, подобное показанному на рис. 11.17.

Откройте Power BI Desktop.

1. На вкладке Главная страница (Home), в разделе Настраиваемые визуальные элементы (Custom Visuals), выберите команду From Marketplace (Из рынка).

2. В открытом окне диалога Power BI Настраиваемые визуальные элементы (Custom Visuals) введите текст Word Cloud (Облако слов) в поле поиска и щелкните на иконку поиска.

Рис. 11.17. Для визуализации слов в сообщениях можно использовать "облако" слов в Power BI. Настройки стоп-слов разрывают сообщения на отдельные слова на уровне визуализации

3. Настраиваемая визуализация Word Cloud отображается в результатах поиска. Щелкните на кнопке Добавить (Add). Если визуализация загружается на панель Визуализации (Visualizations), щелкните на кнопке OK в окне диалога уведом-

ления.

4. Откройте пустую новую рабочую книгу или новый отчет Power BI Desktop.

В Excel: откройте новую рабочую книгу Excel и на вкладке Данные (Data) вы-полните команды Получить данные (Get Data) Из файла (From File) Из книги (From Workbook).

В Power BI Desktop: откройте новый отчет и выполните команды Получить данные (Get Data) Excel Подключиться (Connect).

5. Выберите файл C11E01.xslx в папке C:\Data\C11\ и щелкните мышью на кнопке Импорт (Import).

6. В окне диалога Навигатор (Navigator) выберите запрос Facebook Posts и щелк-ните на кнопке Загрузить (Load).

7. Выберите визуализацию Word Cloud на панели Визуальные элементы (Visualizations). Перетащите поле Message в раздел Категория (Category). Загрузится визуальный элемент Word Cloud.

8. Выберите визуальный элемент Word Cloud на холсте отчета. На панели Визу-альные элементы (Visualizations) выберите значок Формат (Format). Подклю-чите запрос Stop Words (Стоп-слова) и подключите запрос Default Stop Words (Стоп-слова, заданные по умолчанию). Теперь можно заметить, что Power BI эффективно удаляет стоп-слова запроса. Для добавления в список стоп-слов можно ввести соответствующие слова в текстовое поле Stop Words.

Если вы попытаетесь загрузить большое количество сообщений в визуальный элемент Word Cloud, то обработка вашего отчета может замедлиться. Для повышения производительности отчета можно применить редактор Power Query для разделения слов, как это выполнялось в упражнениях 11.4 и 11.5, а затем передать столбец разделенных слов в визуальный элемент Word Cloud. Еще одна важная причина для выполнения подобного разделения — показать наиболее распространенные слова в других визуальных элементах, таких как гистограмма. Эти визуальные элементы не включают встроенную реализацию стоп-слов и не будут выполнять автоматическую разбивку больших сообщений на отдельные слова, как это реализует Word Cloud.

Отчет Power BI C11E07 - Solutions.pbix, который сочетает несколько методов, рассмотренных в этой главе, и содержит анализ постов Microsoft Press Facebook по словам и темам, с течением времени можно просмотреть на сайте https://aka.ms/ DataPwrBIPivot/downloads,

Этот отчет объединяет основные решения, о которых шла речь в данной главе. Благодаря универсальному взаимодействию между визуальными элементами, доступными в Power BI, а также методами, описанными в этой главе, можно лучше понять истинную сущность текстовых информационных каналов.

Резюме

В этой главе рассмотрено несколько методов, помогающих составить представление о текстовых информационных каналах. Принимая во внимание, что можно применять прямое манипулирование текстом для очистки текстовых данных, таких как даты, коды или адреса, другие методы, описанные в этой главе, лучше подходят для анализа текста. В рассмотренных примерах применялся набор данных из постов Microsoft Press Facebook для поиска по ключевым словам и определения тем из текста.

С помощью условных столбцов в редакторе Power Query показано, как можно определить ограниченное число ключевых слов. Описано декартово произведение — эффективный метод динамического обнаружения большого числа ключевых слов из внешнего списка, также рассмотрено, как можно сформировать настраиваемую функцию для поиска ключевых слов. Для повышения производительности при обработке отчетов применяются функции Table.Buffer и List.Buffer, которые сокращают чрезмерные обращения к внешним источникам данных при наличии пользовательских столбцов.

В этой главе также показано, как можно разделять слова и игнорировать стоп- слова. Рассмотрены полезные методы для фильтрации таблицы по ключевым словам во внешнем источнике с помощью запросов слияния. Для того чтобы игнорировать стоп-слова, осуществлялось объединение запросов с помощью антисоединения слева. Этот метод удобен для фильтрации строк из внешнего списка. Чтобы найти соответствие между разделенными словами и списком ключевых слов, применялись запросы слияния с внутренним объединением, что полезно для фильтрации ваших данных из внешнего списка ключевых слов.

В этой главе также показано, как можно определять составные ключевые слова путем временных манипуляций с сообщениями с помощью функции List.Accumulate и таблицы преобразования. Наконец, было показано, каким образом можно загружать текстовые данные в визуальные элементы Word Cloud в Power BI.

Сейчас вы находитесь в начале своего путешествия по анализу текста. Далее в главе 12 будет показано, каким образом можно использовать возможности искусственного интеллекта, вы научитесь выполнять расширенные задачи анализа текста в Azure, такие как реализация перевода текста, анализ настроений и обнаружение фраз, причем без помощи учетных данных или разработчиков программного обеспечения.

Глава 12

Расширенная текстовая аналитика: извлечение смысла

Все в этом мире имеет скрытый смысл.

- Никос КазанДзакис

В ЭТОЙ ГЛАВЕ ВЫ УЗНАЕТЕ, КАК'...

■ Развертывать Microsoft Azure Cognitive Services для реализации возможностей искусственного интеллекта при анализе текстовых информационных каналов.

■ Транслировать сообщения в Excel и Power BI с помощью Microsoft Translator Text API.

■ Определять настроения в сообщениях с помощью API Cognitive Text Analytics.

■ Группировать сообщения в пакеты для сокращения вызовов API.

■ Извлекать фразы из сообщений с помощью Microsoft Text Analytics API.

■ Обрабатывать многоязычные информационные каналы и расширять запросы Text Analytics для обнаружения языков.

В главе 11 были рассмотрены основные шаги, позволяющие получить представление о работе с текстовыми информационными каналами. Было показано, как определяются ключевые слова в текстовых столбцах, выполняется разбивка и подсчет слов, что позволяет извлекать больше смысла из данных. В этой главе мы сделаем значительный шаг вперед, узнаем, как транслировать текст, применять анализ настроений, извлекать ключевые фразы и использовать возможности облачных служб Microsoft Cognitive Services, что позволит "творить чудеса". С помощью вебконнектора Power Query и нескольких функций языка M можно извлечь глубокий смысл из текста, не прибегая к услугам специалистов по обработке данных или разработчиков программного обеспечения.

Службы Microsoft Azure Cognitive Services

При переводе текстов на английский язык или анализе настроений с помощью компьютера совершенно не обязательно применять основы искусственного интеллекта или информатики. В недавнем прошлом, при внедрении подобных методов и применении их в отчетах, потребовались бы усилия больших групп разработчиков про

граммного обеспечения и специалистов по обработке данных. До недавнего времени из-за сложности вычислений и их внедрения очень немногие организации или предприятия располагали решениями для отчетности, которые для извлечения значений из текстовых источников использовали бы возможности искусственного интеллекта.

К счастью, в наше время искусственный интеллект стал товаром. Теперь можно применять облачные сервисы, что позволит понимать иностранный язык и воспринимать смысл. Когнитивные службы Microsoft на основе Azure позволяют интегрировать искусственный интеллект в бизнес-процессы. Предлагаются услуги в пяти направлениях познавательной деятельности человека: зрение, речь, язык, знания и поиск. С помощью несложных интерфейсов (веб-API) можно встроить некоторые из подобных сервисов в свои отчеты, что позволит выполнять анализ текстовых информационных каналов.

См. также

Службы Azure Cognitive Services доступны для загрузки на сайте https://azure.microsoft.com/en-us/services/cognitive-services/.

В данной главе показано, как формировать запросы Excel и Power BI, которые обрабатывают текстовые информационные каналы для языкового перевода, выполняют анализ настроений и обнаруживают ключевые фразы. Будут использоваться два веб-API, которые являются частью языковых служб в Azure Cognitive Services: API Text Analytics и Translator Text API, как показано на рис. 12.1.

Как правило, в расширенных реализациях анализа текста требуется разветвленная архитектура с множеством компонентов, которые в общем наборе задач специализируются на конкретных элементах.

К счастью, Microsoft реализовала все сложные компоненты в форме служб Azure Cognitive Services. Вы можете рассматривать эти службы как волшебный "черный ящик". После передачи текста в этот "черный ящик" возвращается текст, переведенный на желаемый язык, выполняется оценка настроения или формируется список извлеченных ключевых фраз. Таким образом, архитектура довольно проста: отдельная рабочая книга Excel или отчет Power BI, в котором вся подготовка и перемещение данных осуществляется с помощью механизма Power Query, а "черный ящик" выполняет всю работу, возвращая текст, обогащенный смыслом.

На рис. 12.2 проиллюстрированы три основных шага подобного решения: реализуется отчет, который загружает текстовый столбец из внешнего источника данных, затем передает текст в API переводчика текста Azure Cognitive Services или API текстовой аналитики, и, наконец, результат возвращается из Azure Cognitive Services в отчет. Важно отметить, что эти три шага выполняются в одной операции обновления.

При выполнении запроса каждый раз при обновлении отчета повторяются три шага, обеспечивая желаемые результаты.

Рис. 12.2. Одно обновление занимает три шага, от внешнего источника данных до результатов распознавания

Azure Cognitive Services служит примером программного обеспечения, выступающего в качестве службы (Software as a Service, SaaS). Это значит, что не нужно писать какой-либо серверный код или беспокоиться о доступности внешних серверов или компьютеров для выполнения данной работы. Все, что вам нужно, — это подключиться к набору данных, подготовить его, загрузить в Azure Cognitive Services, получить ответ и загрузить его в свой отчет. Все это выполняется в одном отчете, реализованном с помощью Power Query. Интерфейс между отчетом и службами

Cognitive Services использует HTTP-запросы POST для загрузки текста в службы. Все это делается с помощью простых функций языка M, которые описаны далее в этой главе.

Для того чтобы приступить к работе с API-интерфейсами Translator Text (Переводчик текста) и Text Analytics (Текстовая аналитика), следует получить два API- ключа, как описано в следующем разделе.

Развертывание ключей и ресурсов API в Azure

Службы Azure Cognitive Services созданы в виде облачного сервиса, который прост в использовании и реализован с помощью веб-API. Для использования Cognitive Services нужна учетная запись и подписка в Azure, поэтому необходимо развернуть соответствующий ресурс в Azure и получить ключи API, которые позволят обращаться к службе через веб-вызовы. В данном разделе поясняется, как развернуть службы на портале Azure и получить эти ключи API.

ПРИМЕЧАНИЕ

Службы Cognitive Services доступны на бесплатной основе без ограничения по времени, если необходимо формировать ежемесячный отчет с небольшими наборами данных. Если же службы Cognitive Services используется впервые, то доступен бесплатный пробный сеанс.

Развертывание API Translator Text

Выполните следующие шаги для развертывания Microsoft Translator Text API в Azure и начала использования его в книге Excel или отчете Power BI:

ПРИМЕЧАНИЕ

Поскольку Microsoft постоянно добавляет новые функции, следующие инструкции могут быть не совсем точны к тому времени, как вы знакомитесь с этой главой. Если это так, следуйте инструкциям, представленным на сайте www.microsoft.com/ru-ru/ translator/trial.aspx#get-started.

1. Подпишитесь на Azure:

а. При отсутствии учетной записи, подпишитесь на сайте https://azure.microsoft.com.

б. При наличии учетной записи подпишитесь на сайте https://portal.azure.com.

2. Для оформления подписки на Translator Text API, при наличии подписки на Azure, перейдите на сайт https://portal.azure.com/#create/Microsoft.Cognitive ServicesTextTranslation.

3. Выполните следующие шаги для создания ресурса на Azure для Translator Text API:

а. В поле Имя (Name) введите Translation-Resource.

б. Выберите вашу подписку Azure.

в. Выберите уровень цен. Можно использовать бесплатный уровень цен для текстовых каналов длиной менее 2 млн символов. Подробнее о ценах читайте на сайте https://azure.microsoft.com/ru-ru/pricing/details/cognitive-services/ translator-text-api/.

г. Выберите новую группу ресурсов или же имеющуюся группу ресурсов. Группы ресурсов помогают объединять и управлять несколькими ресурсами в Azure. Если вы обратились к Azure впервые, можно сформировать новую группу ресурсов и назвать ее Translation-Res-Group.

д. Прочитайте уведомление и подтвердите факт прочтения.

е. Выберите команду Закрепить на панели управления (Pin to Dashboard).

ж. Щелкните мышью на кнопке Создать (Create).

4. Щелкните на панели управления (Dashboard) в левой панели портала Azure и выберите плитку Translation-Resource. На открытой странице Translation-Resource щелкните на разделе Keys (Ключи) в левой панели, под разделом Resource Management (Управление ресурсами).

5. Как показано на рис. 12.3, теперь можно получить копию KEY 1 или KEY 2 (которая будет применяться далее в этой главе, в упражнении 12.1). Вам не нужно сохранять ключ как промежуточный шаг за пределами отчета. Вы можете в любое время войти на портал Azure и повторить шаг 4 для получения доступа к этой странице.

ПРИМЕЧАНИЕ

Убедитесь, что ключи не доступны никому вне вашей команды. Вы узнаете об аспектах безопасности ключей API в Power Query далее в этой главе.

Теперь вам известно, как развернуть ресурс Translator Text API в Azure и получить его ключ API, то же самое вы можете сделать для API Text Analytics, который необходим для анализа настроений и извлечения ключевых фраз.

Развертывание API Text Analytics

В данном разделе показано, как можно развернуть ресурс Text Analytics в Azure и получить ключ API. Выполнение этих шагов позволит проанализировать на-строения и извлечь ключевые фразы из вашей книги Excel или отчета Power BI:

1. Подпишитесь к Azure:

а. При отсутствии учетной записи создайте ее на сайте https://azure.microsoft.com.

б. При наличии учетной записи войдите в нее на сайте https://portal.azure.com.

2. Для оформления доступа к Text Analytics API, при наличии подписки на Azure, перейдите на сайт

https://portal.azure.com/#create/Microsoft.CognitiveServicesTextAnalytics.

3. Выполните следующие шаги для создания ресурса на Azure для Text Analytics API:

а. В поле Имя (Name) введите Text-Analytics-Resource.

б. Выберите вашу подписку на Azure.

в. Выберите расположение для ресурса.

г. Выберите уровень цен. Можно использовать бесплатный уровень для текстовых информационных каналов, длина которых не превышает 5000 сообщений и которые состоят не более чем из 5000 символов. Узнайте больше о ценах на Text Analytics на сайте https://azure.microsoft.com/ru-ru/pricing/details/ cognitive-services/text-analytics/.

д. Установите новую группу ресурсов или выберите существующую. Если Azure вы используете впервые, можно создать новую группу ресурсов и назвать ее Text-Analytics-Res-Group.

е. Прочитайте уведомление и подтвердите факт прочтения.

ж. Выберите опцию Закрепить панель управления (Pin to Dashboard).

з. Щелкните на кнопке Создать (Create).

4. Щелкните мышью на вкладке Dashboard на левой панели портала Azure и выберите плитку Text-Analytics-Resource. После открытия страницы Text-Analytics-Resource щелкните на левой панели, под вкладкой Resource Mana-gement (Управление ресурсами).

5. Как показано на рис. 12.3, теперь можно получить копию KEY 1 или KEY 2 (которая будет использоваться далее в этой главе в качестве API для анализа настроений в упражнении 12.2 и извлечения ключевых фраз в упражнении 12.4).

ПРИМЕЧАНИЕ

Как отмечалось ранее, убедитесь, что ключи не переданы никому, кроме членов вашей команды. Вы узнаете об аспектах безопасности ключей API в Power Query далее в этой главе.

Плюсы и минусы доступа к Cognitive Services, получаемого через Power Query

Методы, которые рассмотрены в данной главе, применяются для перевода текста, анализа настроений и выявления ключевых фраз, и они чрезвычайно полезны. Но, прежде чем рассматривать эти реализации, следует узнать о плюсах и минусах следования этому подходу.

Как показано далее, интеграция между Excel и службами Cognitive Services требует определенных знаний Power Query. Однако можно легко выполнить упражнения этой главы, а затем использовать изученный материал в качестве шаблона в Excel или Power BI для обработки любых текстовых источников, которые, возможно, вы захотите проанализировать.

Решение рассмотренных здесь задач не требует каких-либо современных данных, навыков машинного обучения или разработки программного обеспечения. Любой пользователь Excel, выполняющий действия, описанные в этой главе, сможет внедрить Cognitive Services и сразу же получить новые знания. Это означает, что потребители информационных услуг могут легко использовать текстовые информационные каналы в собственных инструментах анализа, тем самым повышая заинтересованность бизнес-структур в подобном анализе.

Прежде чем применять рассмотренные в этой главе методы, необходимо учесть предостережения, описанные в следующих разделах.

Чрезмерные вызовы API

Службы Cognitive Services доступны на платной основе. Важно, чтобы вы понимали ограничения Power Query при пользовании этой услугой, иначе вы в конечном итоге перейдете к созданию чрезмерных вызовов API и вас удивят незапланированные требования по обновлению уровня ценообразования. Чтобы избежать чрезмерных вызовов API, учтите следующее.

Небольшие наборы данных. Для создания запроса в редакторе Power Query рабо-тайте с небольшими наборами данных. Вы можете использовать шаг преобразования Keep Top Rows (Сохранить верхние строки) перед вызовом служб Cognitive Services для временного сокращения набора данных.

Предварительный просмотр данных в фоновом режиме: по умолчанию, возможно без вашего ведома, Power Query загружает данные в фоновом режиме для обеспечения плавного взаимодействия с пользователем, если вы выбираете какие- либо шаги преобразования на панели Примененные шаги (Applied Steps) редактора Power Query или при перемещении между запросами. При этом можно быстро увидеть преобразованные данные на панели предварительного просмотра. При работе с платными или ограниченными по скорости веб-API, такими как Cognitive Services, вы быстро достигнете ограничения вызовов API. Чтобы избежать запуска ненужных и незапланированных вызовов API, можно отключить предварительный просмотр фоновых данных для каждого отчета.

В окне редактора Power Query в меню Файл (File) выберите команду Параметры и настройки (Options and Settings) и затем выполните команду Параметры запроса (Query Options). Выберите команду Загрузка данных (Data Load) в разделе Теку-щая книга (Current Workbook в Excel) или Текущий файл (Current File) в Power BI Desktop и отмените установку флажка Разрешить скачивание в фоновом режиме для предварительного просмотра данных (Allow Data Preview to Download in the Background), см. рис. 12.4. Отмена установки этого флажка необходима для каждой книги или отчета Power BI, в который вы интегрируетесь с Cognitive Services. Если этот флажок установлен, то Power Query будет продолжать загружать предварительный просмотр данных в фоновом режиме и осуществлять ненужные вызовы API для Cognitive Services.

Рис. 12.4. В окне диалога Параметры запроса отмените предварительный просмотр данных для загрузки в фоновом режиме, чтобы предотвратить нежелательные вызовы API

Даже после того, как вы отмените установку этого флажка, имейте в виду, что каждый раз, когда вы открываете редактор Power Query и переходите к определенным запросам, будет запускаться серия вызовов API для Cognitive Services, чтобы предоставить вам последние результаты на панели предварительного просмотра.

В Power BI Premium можно реализовать инкрементное обновление, которое пре-дотвратит чрезмерные вызовы API. Однако Excel и Power BI (не Premium) не располагают подобным механизмом обновления. Каждый раз, когда обновляется отчет, запрос загружает весь набор данных и не сохраняет промежуточных результатов предыдущего обновления. В итоге если требуется часто обновлять отчет, который интегрируется с Cognitive Services, то вы будете отправлять большие порции одного и того же набора данных снова и снова, что довольно неэффективно с точки зрения поддержания низкой скорости вызовов API.

Для устранения этого ограничения можно реализовать описанные в этой главе запросы для нескольких файлов Excel. Каждый файл будет загружать только приращения данных в Cognitive Services на основе предварительно определенного периода, и результаты будут сохраняться в рабочем листе. Затем, с помощью коннектора папок, как описано в главе 3, можно внести все приращения в один отчет, используя при этом только необходимое количество вызовов API.

Конфиденциальность и безопасность

Проверьте, разрешает ли корпоративная политика вашей компании использовать Cognitive Services. Хотя Azure, вероятно, является наиболее безопасной облачной платформой, ваша компания может запретить ее применение при обработке конфиденциальных наборов данных. Если нет уверенности, что можно работать с Cognitive Services, обратитесь за консультацией в ИТ-отдел.

Если нужно поделиться отчетом с другими авторами отчетов, не входящими в команду, не забудьте удалить ключи API перед отправкой им отчета. Далее эта тема освещается более подробно.

Неподдерживаемое обновление по расписанию

Методы, описанные в этой главе, можно обновлять только с помощью Excel или Power BI Desktop. Если попытаться применить запланированное обновление к набору данных в сервисе Power BI, то вы получите ошибки обновления. Можно создать собственный коннектор, который будет поддерживать запланированное обновление через локальный шлюз данных, однако подобный подход потребует дополнительных навыков разработки, рассмотрение которых выходит за рамки этой книги.

СОВЕТ

Учитывая перечисленные ограничения, рекомендуется применять методы, описанные в этой главе, в сценариях специального анализа, в небольших наборах данных или в качестве подтверждения концепции для более крупных реализаций вне Excel и Power BI.

Перевод текста

Перевод текстов в Excel или Power BI с помощью Power Query может оказаться полезным способом по автоматизации сбора многоязычных данных. В этом разделе показано, как применять Power Query для автоматизации перевода текстовых столбцов путем взаимодействия с Cognitive Services Translator Text API. Возможно, вы уже привыкли к ручному копированию и вставке текста из ваших наборов данных в любимый веб-переводчик, что позволяло интерпретировать многоязычные данные. Рассмотренная в этом разделе методика позволит автоматизировать процесс, получая представление о неиспользованных текстовых информационных каналах, как показано на рис. 12.5.

Переведенный текст

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Рис. 12.5. Можно автоматизировать перевод массовых сообщений в Excel или Power BI

Справочник по Translator Text API

Несмотря на то что справочник по текстовому API переводчика довольно обширен и располагает широким спектром возможностей, в данной главе основное внимание уделяется основному методу перевода одного сообщения. Этим методом служит вызов API Translate версии 3.0 (в настоящее время это последняя версия). Полный список вызовов API можно просмотреть на сайте по адресу https://docs. microsoft.com/ru-ru/azure/cognitive-services/translator/reference/v3-0-reference.

Вызов запроса Translate переводит строку с одного языка на другой. Запрос передается как веб-запрос путем указания URI-ссылки запроса https://api.cognitive. microsofttranslator.com/translate, затем номера версии API в качестве значения параметра api-version и двузначного кода языка в качестве значения параметра to для целевого языка, на который выполняется перевод. Ниже приводится URI запроса, выполняющего перевод текста на английский при помощи текущей версии API 3.0:

https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to=en

Если нужно перевести текст на определенный язык, можно найти перечень языков на сайте https://docs.microsoft.com/en-us/azure/cognitive-services/translator/languages и применить двузначный код в качестве значения параметра to. Например, следующая URI-ссылка переводит приветствие Hola Mundo на голландский:

https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to=nl

Для получения результата перевода необходимо авторизоваться службой, и процесс авторизации реализуется путем передачи ключа API при вызове. В упражнении 12.1 будет показано, как передать ключ API в заголовке Ocp-Apim-Subscription-Key. Также передается соответствующий текст для перевода. В версии 3.0 API текст должен направляться в теле запроса в формате JSON:

[

{"Text":"Hola Mundo"}

]

Внутри этого массива JSON можно передавать до 25 текстовых сообщений по 1000 символов каждое (включая пробелы) и до 5000 символов (включая пробелы) для всего блока сообщений. Чтобы разбить длинные сообщения на небольшие части, которые может принять API, можно вызвать API BreakSentence, который объясняется на сайте https://docs.microsoft.com/en-us/azure/cognitive-services/translator/ reference/v3-0-break-sentence. В этой главе мы рассмотрим большое число методик, поэтому в упражнении 12.1 сосредоточимся на передаче одного текстового сообщения.

УПРАЖНЕНИЕ 12.1. Простой перевод

В упражнении реализуется простой перевод одного текстового сообщения, Hola Mundo (Здравствуй, мир), с помощью Translator Text API версии 3.0.

Часть 1. Формирование контента JSON для тела запроса

В первой части упражнения показано, как в теле запроса вызова Translate API формируется содержимое JSON с сообщением, которое необходимо перевести.

1. Откройте пустую рабочую книгу Excel или новый отчет Power BI Report:

В Excel: на вкладке Данные (Data), в разделе Получить и преобразовать данные (Get & Transform Data), выполните команды Из других источников (From Other Sources) Пустой запрос (Blank Query).

В Power BI Desktop: расширьте раскрывающееся меню Получить данные (Get Data) и выберите команду Пустой запрос (Blank Query).

2. В строке формул введите следующую формулу для формирования записи текста для перевода:

= [Text = "Hola Mundo"]

Данная M-формула, будучи транслирована в JSON, конвертируется в следующий формат:

{"Text" = "Hola Mundo"}

Конвертируйте эту запись в JSON, заключая содержимое строки формул в функцию Json.FromValue:

= Json.FromValue([Text = "Hola Mundo"])

3. На панели предварительного просмотра отобразится значок CSV. Щелкните правой кнопкой мыши на этом значке и в контекстном меню выберите команду Text (Текст). Теперь можно заметить, что первая строка новой таблицы включает необходимый формат:

{"Text" = "Hola Mundo"}

4. Пришло время сформировать полностью блок JSON, включая упаковочные скобки. На панели Примененные шаги (Applied Steps) выберите шаг Source (Источник). Теперь в строке формул измените формулу, заключая запись в фигурные скобки следующим образом:

= Json.FromValue( { [Text = "Hola Mundo"] } )

С помощью фигурных скобок M-запись добавляется в список. Теперь на панели Примененные шаги (Applied Steps) выберите последний шаг Imported Text (Импортированный текст) и заметьте, каким образом модификация влияет на формат JSON. Первая строка в таблице теперь имеет формат, необходимый для содержимого JSON:

[{"Text":"Hola Mundo"}]

5. Теперь, когда известно, как сформировать контент JSON с переведенным тек-стом, удалите запрос Query1. Вскоре используем последний код Json.FromValue в качестве тела запроса другого M-выражения.

СОВЕТ

Языки M и JSON имеют противоположные по смыслу определения для записей и списков. В языке M скобки служат для определения записи. В JSON скобки определяют список. В M фигурные скобки применяются для определения списка, а в JSON они используются для определения записи. Эти представления помогут сформировать соответствующие форматы JSON в языке M, поскольку вы работаете со многими веб- API.

Часть 2. Создание параметра API Key

Пришло время создать параметр для ключа API, который затем будет использо-ваться в вызове API. Продолжите упражнение 12.1, выполняя следующие шаги.

6. Войдите на портал Azure и повторите шаги 4-5 из предыдущего раздела и скопируйте один из двух ключей API-интерфейса текстового переводчика.

7. Для формирования параметра для ключа Translator Text API, скопированного на шаге 6, выполните следующие действия:

а. На вкладке Главная (Home) редактора Power Query расширьте раскрываю-щееся меню Manage Parameters (Менеджер параметров) и выберите опцию New Parameter (Новый параметр).

б. При открытом окне диалога Параметры (Parameters), в поле Имя (Name), введите APIKey, затем в поле Текущее значение (Current Value) вставьте API ключ из шага 6, как показано на рис. 12.6. Затем щелкните мышью на OK для закрытия этого окна диалога.

Часть 3. Создание веб-запроса

В данной части упражнения формируется основной веб-запрос и в качестве ответа получают переведенное сообщение. Выполните следующие шаги для продолжения упражнения 12.1:

8. Сформируйте новый пустой запрос и переименуйте запрос в Translated Message.

9. При выбранном запросе Translated Message на панели Запросы (Queries) выбе-рите команду Расширенный редактор (Advanced Editor) на вкладке Главная (Home) и введите следующее M-выражение:

let

Source = Web.Contents(

"https://api.cognitive.microsofttranslator.com/" & "translate?api-version=3.0&to=en",

[

Headers = [

#"Ocp-Apim-Subscription-Key" = APIKey, #"Content-Type" = "application/json"

],

Content = Json.FromValue({[Text = "Hola Mundo"]}) ]

)

in

Source

Рассмотрим это выражение. Основная функция здесь — Web.Contents, ее также можно создавать при выборе веб-коннектора посредством меню Получить данные (Get Data).

ПРИМЕЧАНИЕ

Как правило, при необходимости импортировать данные из Интернета не нужно записывать выражение M "с нуля", как это реализовано в данном упражнении. Можно выбрать команду Из Интернета (From Web) в раскрывающемся меню Получить данные (Get Data) в Excel или Интернет (Web) в раскрывающемся меню Получить данные (Get Data) в Power BI и воспользоваться окном диалога Из Интернета (From Web) для предоставления URL-адреса. Можно указать заголовки HTTP в разделе Дополнительно (Advanced) окна диалога Из Интернета (From Web). Однако, если необходимо передать тело запроса, как в случае Translator Text API, необходимо создать M-выра-жение Web.Contents.

Функция web.Contents включает два основных элемента: URL-адрес, передавае-мый в качестве первого аргумента, и аргумент Options, передаваемый в качестве второго аргумента и служащий контейнером для всех других элементов, которые можно использовать для создания веб-запроса, таких как Headers и Content. В M-выражении можно заметить, что Headers — это запись пар "ключ/значение". Каждая включает соответствующий заголовок HTTP, который требуется для вызова API.

Запись Headers включает заголовок Ocp-Apim-Subscription-Key с параметром APIKey, сформированный на шаге 7. Поскольку имя заголовка содержит тире, которое служит специальным символом в языке M, необходимо заключить его в двойные кавычки: Content-Type, являясь другим востребованным заголовком, не пред

ставляется вторым ключом в записи Headers.

После записи Headers следует элемент Content вместе с Json.Value, который создан на шаге 4.

Заметим, что также можно сформировать функцию Web.Contents с помощью до-полнительных настроек, включая RelativePath и Query, и можно переместить боль-шее число элементов из URL в аргумент Options. Далее приводится M-выражение с новыми элементами, выделенными полужирным шрифтом. В этом выражении URL-ссылка включает только доменное имя:

let

Source = Web.Contents(

"https://api.cognitive.microsofttranslator.com",

[

RelativePath = "translate",

Query = [

#"api-version" = "3.0",

to = "en"

],

Headers = [

#"Ocp-Apim-Subscription-Key" = APIKey, #"Content-Type" = "application/json"

],

Content = Json.FromValue({[Text = "Hola Mundo"]}) ]

)

in

Source

СОВЕТ

Имеется несколько преимуществ применения RelativePath и Query в Web.Contents. Поле RelativePath во многих случаях повышает вероятность, что ваши расширенные запросы поддержат обновление расписания в сервисе Power BI. Указание Query позволяет механизму языка M автоматически применять кодировку URI к значениям запроса и избегать того, что сервис неожиданно приступит к обработке текста.

10. Скопируйте и вставьте новое выражение, затем щелкните на кнопке Готово (Done) для закрытия окна расширенного редактора.

11. После появления на панели предварительного просмотра строки уведомления выберите команду Изменить полномочия (Edit Credentials). После этого откроется окно диалога Доступ в Интернете (Web Access). Удостоверьтесь, что выбран параметр Анонимно (Anonymous), и щелкните мышью на кнопке Подключиться (Connect).

12. На панели предварительного просмотра найдите значок JSON с текстом api.cognitive.microsofttranslator.com. Щелкните правой кнопкой мыши на этом значке и выберите пункт JSON. Отклик JSON будет включать список с единственной записью.

13. Поскольку передано только одно сообщение, выберите запись, щелкнув на ссылке.

На панели предварительного просмотра отображается запись detectedLanguages и список Translations. Если щелкнуть на пустом месте в ячейке detectedLanguages, то вы увидите результирующую запись, с language = es и score = 1. Это означает, что служба определила исходный язык как испанский со 100%-ной уверенностью. Поскольку цель состоит в получении переведенного текста на английском языке, можно игнорировать эту запись и перейти к переведенному тексту.

14. Щелкните на ссылке List в поле translations.

15. В панели предварительного просмотра, где теперь отображается список с одной записью, щелкните на ссылке Record (Запись). Теперь отобразится новая запись с полями text и to. Переведенный текст будет трактоваться как значение аргумента text.

16. Щелкните правой кнопкой мыши на ячейке со значением Hello World и в контекстном меню выберите команду Детализация (Drill Down).

17. Загрузите запрос Translated Message в рабочую книгу или отчет Power BI и сохраните его.

Файлы решений C12E01 - Solution.xlsx и C12E01 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Совместное использование отчета без ключа API

Если нужно поделиться с другими пользователями рабочей книгой или отчетом Power BI, созданным в упражнении 12.1, то следует учесть, что ключ API недостаточно защищен. Любой пользователь, который откроет этот отчет в Excel или Power BI Desktop, сможет получить и ключ API. В главе 8 рассматривались несколько методов обмена рабочими книгами и отчетами Power BI с другими пользователями. Удаление ключа API перед публикацией отчета — это минимальное изменение, которое следует внести, что позволит избежать предоставления ключа API другим лицам.

Выполните следующие действия, чтобы удалить ключ API перед публикацией книги или отчета.

1. Откройте рабочую книгу, созданную при выполнении упражнения 12.1, и запустите редактор Power Query. (Если книга не сохранена, то можно загрузить файл C12E01 - Solution.xlsx или C12E01 - Solution.pbix с сайта https://aka.ms/ DataPwrBIPivot/downloads.)

2. На вкладке Параметры (Parameters) выберите параметр APIKey.

3. Замените ключ API в поле Текущее значение (Current Value) с помощью сле-дующего текста:

"paste your key here"

4. При публикации отчета попросите коллег выполнить шаги 1 и 2 и замените текст "paste your key here” (вставьте свой ключ сюда) своим ключом API. После этого они могут обновлять отчет.

В Power BI Desktop теперь можно выполнить еще один шаг и улучшить взаимодействие с пользователем, создавая шаблон Power BI, как описано в главе 8. При создании шаблона пользователи могут вводить свои ключи перед загрузкой отчета. Как реализовать это, станет понятно после выполнения следующего упражнения. Прежде чем это сделать, покажем, каким образом выполняется обработка нескольких вызовов API.

УПРАЖНЕНИЕ 12.2. Перевод нескольких сообщений

В данном упражнении запрос Translate преобразуется в функцию, что позволит повторно использовать его для обработки нескольких сообщений. Напомним, что хотя Translator Text API и позволяет отправлять 25 сообщений за один вызов API, подобный подход недостаточен для отправки огромного числа сообщений. В данном упражнении показано, как отправлять большое количество сообщений, но для простоты реализации вы все равно будете отправлять только одно сообщение на один вызов API.

1. Откройте сохраненную книгу или отчет Power BI из упражнения 12.1. Если отчет не сохранен, загрузите один из файлов решений C12E01 - Solution.xlsx или C12E01 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads и откройте его.

2. Запустите редактор Power Query.

3. Выполните следующие шаги для формирования параметров запроса, которые применяются далее вместо жестко закодированных параметров text и to, которые применялись в упражнении 12.1:

Обработка параметра text в теле запроса:

а. На вкладке Главная (Home) откройте раскрывающееся меню Управление параметрами (Manage Parameters) и выберите команду Создать параметр (New Parameter).

б. После открытия окна диалога Параметры (Parameters) в поле Имя (Name) введите InputText; и затем в поле Текущее значение (Current Value) введите Hola Mundo. Потом щелкните мышью на кнопке OK.

Обработка параметра to:

в. Повторите шаг 3a, затем в поле Имя (Name) введите TranslateToLanguage.

г. В поле Текущее значение (Current Value) введите en. Затем щелкните на кнопке OK.

4. В панели Запросы (Queries) выберите запрос Translated Message и затем вы-полните команду Расширенный запрос (Advanced Editor).

Затем измените формулу M и замените жестко закодированные значения новыми параметрами из шага 3. Далее приведена часть выражения, которую можно найти в окне расширенного редактора. Полужирным шрифтом выделены жестко закодированные значения "en" и "Hola Mundo", которые необходимо заменить (обратите внимание, что выражение имеет отступы, чтобы формула стала читабельной):

= Json.Document(

Web.Contents(

"https://api.cognitive.microsofttranslator.com", [

RelativePath = "translate",

Query = [

#"api-version" = "3.0",

to = "en"

],

Headers = [

#"Ocp-Apim-Subscription-Key" = APIKey, #"Content-Type" = "application/json"

],

Content = Json.FromValue({[Text = "Hola Mundo"]})

]

)

)

5. Замените текст "Hola Mundo" на InputText и текст "en" на TranslateToLanguage.

Вот так выглядит обновленная формула:

= Json.Document(

Web.Contents(

"https://api.cognitive.microsofttranslator.com", [

RelativePath = "translate",

Query = [

#"api-version" = "3.0",

to = TranslateToLanguage

],

Headers = [

#"Ocp-Apim-Subscription-Key" = APIKey, #"Content-Type" = "application/json"

],

Content = Json.FromValue({[Text = InputText]})

]

)

)

6. Выберите последний шаг на панели Примененные шаги (Applied Steps) и удо-стоверьтесь в том, что на панели предварительного просмотра по-прежнему переводится текст Hello World. Это означает, что внесенные изменения верны.

7. Теперь можно конвертировать запрос Translated Message в настраиваемую функцию.

а. На панели Запросы (Queries) правой кнопки мыши щелкните на запросе Translated Message и в контекстном меню выберите команду Создать функцию (Create Function).

б. После открытия окна диалога Создать функцию (Create Function) в поле Имя функции (Function Name) введите FnTranslate, затем щелкните на кнопке OK для закрытия окна диалога.

8. Загрузите таблицу с несколькими сообщениями, как показано на рис. 12.5, и переведите их на английский, выполнив следующие шаги:

а. При открытом окне редактора Power Query загрузите файл рабочей книги C12E02.xlsx с сайта https://aka.ms/DataPwrBIPivot/downloads и сохраните его в папке C:\Data\C12\.

б. На вкладке Главная (Home) редактора Power Query выполните команды Создать источник (New Source) Файл (File) Excel.

в. После открытия окна диалога Импорт данных (Import Data) перейдите в папку C:\Data\C12\ и выберите файл C12E02.xlsx.

г. После открытия окна диалога Навигатор (Navigator) выберите таблицу TextForTranslation и щелкните мышью на кнопке OK.

9. В окне редактора Power Query переименуйте новый запрос на Translated Messages.

10. При выделенном запросе Translated Messages на панели Запросы (Queries) выберите команду Вызвать настраиваемую функцию (Invoke Custom Function) на вкладке Добавление столбца (Add Column).

11. В открытом окне диалога Вызвать настраиваемую функцию (Invoke Custom Function) в поле Запрос функции (Function Query) введите FnTranslate.

12. Обратите внимание, что столбец Text for Translation (Текст для перевода) не-правильно заполнен как APIKey.

К сожалению, нельзя назначить параметры в качестве аргументов функции с помощью окна диалога Вызвать настраиваемую функцию (Invoke Custom Function). Сохраните столбец Text for Translation (Текст для перевода) как APIKey, и его можно будет быстро исправить.

13. Выберите Имя столбца (Column Name) в раскрывающемся меню InputText. Заметьте, что имя Text for Translation (Текст для перевода) заполняет раскры-вающееся меню справа.

14. В нижнюю часть окна диалога Вызвать настраиваемую функцию (Invoke Custom Function) введите в текстовое поле en, в разделе TranslateToLanguage. Щелкните мышью на кнопке OK для закрытия этого окна диалога.

15. Переименуйте столбец FnTranslate как в Translated Text (Переведенный текст).

16. В строке формул отобразится формула:

= Table.AddColumn(#"Changed Type", "FnTranslate", each FnTranslate(

[Text for Translation], [Text for Translation], "en"))

Измените ее на следующую формулу, заменив первый экземпляр [Text for Translation] на параметр APIKey:

= Table.AddColumn(#"Changed Type", "FnTranslate", each FnTranslate(APIKey,

[Text for Translation], "en"))

В зависимости от настроек конфиденциальности в окне диалога Параметры запроса (Query Options), скорее всего, отобразится следующее предупреждение в панели предварительного просмотра редактора Power Query:

Information is required about data privacy

(Требуется информация о конфиденциальности данных)

17. Щелкните на кнопке Продолжить (Continue). Откроется окно диалога Уровни конфиденциальности (Privacy Levels). В этом окне можно задать уровни кон-фиденциальности для источников данных. Для проверки, что конфиденциальные данные не загружаются в ненадежные источники, можно установить уровни конфиденциальности для файлов C12E02.xlsx и Cognitive Services, как показано на рис. 12.7:

а. Выберите пункт c:\ в первом раскрывающемся меню и установите уровень конфиденциальности Organizational (Организационный).

б. Установите уровень конфиденциальности Organizational (Организацион-ный) для https://api.cognitive.microsofttranslator.com как.

в. Щелкните мышью на кнопке Save (Сохранить).

X

Privacy levels

The privacy level is used to ensure data is combined without undesirable data transfer. Incorrect privacy levels may lead to sensitive data being leaked outside of a trusted scope. More information on privacy levels can be found here.

О Ignore Privacy Levels checks for this file. Ignoring Privacy Levels could expose sensitive or confidential data to an unauthorized person.

D c;\ ’’ Organizational ”

@ https://api.cognitivemicrosofttransLator.com/

\* Organizational ’

Save Cancel

Рис. 12.7. Установите уровни конфиденциальности, применяемые

при передаче сообщений для перевода Power Query

СОВЕТ

Если случайно установить разные уровни конфиденциальности для источников данных, например Public (Общедоступный) и Organizational (Организационный), то запрос не будет выполняться. В подобных случаях, если необходимо очистить уровни конфи-денциальности, можно выбрать раздел Настройки источника данных (Data Source Settings) на вкладке Главная (Home) редактора Power Query и очистить разрешения

для двух источников данных.

Теперь переведенный текст можно увидеть на панели предварительного просмотра редактора Power Query. Все сообщения переведены на английский язык, Hello World! (Здравствуй, мир!). После этого можно закрыть и загрузить запрос. Если применяется Power BI Desktop, сохраните отчет. В следующем разделе показано, каким образом можно совместно использовать отчет, не раскрывая ключ API.

Совместное использование отчета без ключа API: удобный способ

Если упражнение 12.2 в Power BI Desktop уже проделано, то теперь покажем, как экспортировать отчет в шаблон Power BI, которым можно поделиться с другими авторами отчетов, не используя ключ API. Если же вы не выполнили упражнение 12.2, то можно начать с файла решения C12E02 - Solution.pbix, который доступен на сайте https://aka.ms/DataPwrBIPivot/downloads.

ПРИМЕЧАНИЕ

Если вас интересует только решение Excel, перейдите к концу этого раздела и загрузите рабочую книгу решения для Excel, которая включает методы, изученные в главе 8, и загружает ключ API из рабочего листа. Затем можно дать указание другим пользователям предоставить свой API-ключ в соответствующей ячейке. Кроме того, данное решение устраняет ошибки брандмауэра, описанные в главе 8.

Можно улучшить решение в отчете Power BI, создав шаблон, тогда ваш API-ключ не передается другим лицам, при этом поддерживается удобная форма работы для получателей отчета. Напомним, что в главе 8 было показано, как сформировать шаблон Power BI (файл \*.pbit).

Снова опишем этот процесс в данном разделе, поскольку текущий отчет является особым случаем: он имеет два параметра, InputText и TranslateToLangauge, которые желательно скрыть из окна диалога загрузки, оставив в качестве единственного параметра только APIKey.

1. Откройте отчет Power BI, который был сохранен в конце упражнения 12.2, или загрузите файл C12E02 - Solution.pbix и откройте его. Если применяется новый файл C12E02 - Solution.pbix, то придется обновлять APIKey. На вкладке Главная (Home) выберите команду Изменить запросы (Edit Queries). На панели Запросы (Queries) выберите раздел APIKey и введите ваш ключ API в поле Текущее значение (Current Value).

На вкладке Главная (Home) выберите команду Изменить запросы (Edit Queries). На панели Запросы (Queries) редактора Power Query отобразятся три параметра: APIKey, InputText и TranslateToLanguage. Последние два нужны для пользовательской функции в упражнении 12.2.

2. Для того чтобы избежать отображения параметров InputText и TranslateToLanguage в окне диалога загрузки при открытии шаблона, преобразуйте их в запросы:

а. На панели Запросы (Queries) правой кнопки мыши щелкните на поле InputText и в контекстном меню выберите команду Преобразовать в запрос (Convert to Query).

б. Правой кнопки мыши щелкните на поле TranslateToLanguage и в контекст-ном меню выберите команду Преобразовать в запрос (Convert to Query).

3. На вкладке Главная (Home) выберите команду Закрыть и применить (Close & Apply. Вернитесь в главное окно Power BI, на вкладку Файл (File), и выполните команды Экспорт (Export) Шаблон Power BI (Power BI Template).

4. В окне диалога Экспорт шаблона (Export a Template) запишите краткие инст-рукции для ваших коллег и щелкните на кнопке OK.

5. Выберите название для шаблона и сохраните его.

Можно загрузить файл решения C12E02 - Solution.pbit с сайта https://aka.ms/ DataPwrBIPivot/downloads. Альтернативное решение, применяющее Excel, под-держивается в файле C12E02 - Solution.xlsx.

Если вы поделитесь своим отчетом с другими авторами отчетов, то при открытии шаблона решения в Power BI или в рабочей книге Excel отобразится информация, приведенная на рис. 12.8.

Рис. 12.8. Можно поделиться переведенным текстом отчета Power BI (используя шаблон)

и рабочую книгу Excel (используя именованный диапазон), не раскрывая ключ API

Г") СОВЕТ

Чтобы отозвать ключ API, перейдите в представление Keys (Ключи) текстового ресурса Translator Text Resource на портале Azure и выберите команду Regenerate Key1 (Восстановить ключ 1) или Regenerate Key2 (Восстановить ключ 2).

Анализ настроений

Основополагающим элементом успеха любого бизнеса или организации является повторяющийся процесс сбора, анализа информации и улучшение обратной связи. Отзывы от своих клиентов можно получать во многих форматах, но текстовое представление является наиболее сложным. Как же можно количественно оценить качественные данные, такие как текстовые отзывы пользователей? Как определить общее настроение пользователя по поводу конкретного продукта или бренда? Какие изменения можно спрогнозировать с течением времени? Разве вам не нужен автоматический способ получения оценок настроения пользователей на базе информации из текстовых источников?

В этом разделе показано, как можно обрабатывать текстовые информационные каналы и получать оценку настроения для каждого сообщения. Вы узнаете, как использовать Power Query для автоматизации рабочей нагрузки по анализу настроений, будет рассмотрено взаимодействие с Cognitive Services Text Analytics API. Как показано на рис. 12.9, в конечном счете каждое сообщение получит оценку от 0 до 1, где 0 означает, что сообщения выражают отрицательные мнение, а 1 соответствует положительному мнению.

Вызов API Sentiment Analysis

В данном разделе кратко объясняется синтаксис вызова API, который помогает понять структуру пользовательской функции М, которая используется в следующем разделе. Если эта API вас не интересует, можно пропустить этот раздел и перейти к следующему.

Рис. 12.9. Вы можете использовать Sentiment Analysis API для загрузки текстовых сообщений и отображения оценок настроения от 0 (отрицательная оценка) до 1 (положительная оценка)

См. также

Для получения полной ссылки на вызов Sentiment API обратитесь на сайт https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/ text-analytics-how-tosentiment-analysis.

Данная служба может анализировать настроения с помощью нескольких языков, таких как английский, немецкий, испанский и французский. Полный список под-держиваемых языков приведен на сайте https://docs.microsoft.com/en-us/azure/ cognitive-services/text-analytics/text-analytics-supported-languages.

Запрос передается как веб-запрос POST со следующим URI запроса, который также называется Endpoint:

https://[location].api.cognitive.microsoft.com/text/analytics/v2.0/sentiment

Для нахождения определенного значения [location] (местоположения), которое требуется для Endpoint, можно получить доступ к вашему ресурсу Text Analytics на портале Azure и найти его на панели просмотра в разделе Endpoint. К примеру, если вашим местоположением служит West US, вот ваш Endpoint:

https://westus.api.cognitive.microsoft.com/text/analytics/v2.0/sentiment

Подготовка данных

Прежде чем направлять текстовый канал службе анализа настроений, необходимо познакомиться с некоторыми рекомендациями и ограничениями:

1. Вместе с каждым сообщением следует указывать уникальный идентификатор и язык сообщения.

2. Рекомендуется ограничивать сообщения одним или двумя предложениями на одно сообщение, а не направлять большой блок текста. Каждое сообщение должно содержать не более 5000 символов. Служба Sentiment Analysis более точна, если для работы предоставляются небольшие части текста.

3. Можно направлять до 1000 отдельных сообщений за один вызов API.

4. Сервис принимает до 100 запросов в минуту.

5. Общий размер сообщений в каждом запросе не должен превышать 1 Мбайт.

6. Содержание должно быть в формате JSON. (Не беспокойтесь о формате JSON. Power Query может легко преобразовать таблицу в формат JSON.)

Вот пример содержимого в формате JSON, которое можно направить в теле запроса API:

{

"documents": [

{

"language": "en",

"id": "1",

"text": "We love this trail and make the trip every year."

},

{

"language": "en",

"id": "2",

"text": "Poorly marked trails! I thought we were goners.

Worst hike ever."

},

{

"language": "en",

"id": "3",

"text": "This is my favorite trail. It has beautiful views."

}

]

}

Служба Sentiment Analysis API возвращает результаты в формате JSON следующим образом:

{

"documents": [

{

"score": 0.87631344795227051,

"id": "1"

},

{

"score": 0.05087512731552124,

"id": "2"

},

{

"score": 0.97651207447052,

"id": "3"

}

],

"errors": []

}

Не беспокойтесь, глядя на приведенные выражения: синтаксический разбор JSON в Power Query прост и интуитивно понятен.

УПРАЖНЕНИЕ 12.3. Реализация настраиваемой функции анализа настроений FnGetSentiment

В данном упражнении будет показано, как создать настраиваемую функцию в Power Query для подготовки сообщений к вызову запроса API и обработки оценки настроения в ответе.

Настраиваемая функция FnGetSentiment

Прежде чем начать упражнение, взгляните на приведенный в листинге 12.1 код настраиваемой функции, которую необходимо создать. Эта пользовательская функция позволит выполнить анализ настроений на основе данных во входной таблице или, что еще лучше, на основе выполнения подобного анализ для большого набора данных сообщений, повторно используя его на подмножествах, каждое из которых состоит из 1000 сообщений.

Листинг 12.1. Настраиваемая функция FnGetSentiment

(Source) => let

#"Removed Other Columns" = Table.SelectColumns(Source,

{"id", "text"}),

#"Added Custom" = Table.AddColumn(#"Removed Other Columns", "language", each "en"),

JSONRequest = Json.FromValue([documents = #"Added Custom"]), JSONResponse = Json.Document(

Web.Contents( "https://westus.api.cognitive.microsoft.com/text/analytics/ v2.0/sentiment", [

Headers = [

#"Ocp-Apim-Subscription-Key" = TextAnalyticsAPIKey

],

Content = JSONRequest

]

)

), documents = JSONResponse[documents], #"Converted to Table" = Table.FromList(

documents, Splitter.SplitByNothing(), null, null, ExtraValues.Error ),

SentimentScores = Table.ExpandRecordColumn(

#"Converted to Table", "Column1", {"score", "id"}, {"score", "id"} ) in

SentimentScores

СОВЕТ

Приведенная настраиваемая функция была создана за несколько следующих шагов. И хотя можно легко копировать и вставлять настраиваемые функции, которые созданы другими пользователями, а затем применять их повторно, следует учитывать, что многие сложные функции могут создаваться пошагово с использованием пользовательского интерфейса для большей части логики. Это упражнение предоставляет возможность попрактиковаться в создании расширенных настраиваемых функций при небольшой помощи пользовательского интерфейса редактора Power Query Editor, который улучшит ваши способности по созданию простых строительных блоков M-кода, которые в конечном итоге привлекаются и для сложных сценариев.

Часть 1. Загрузка данных

Для нашего упражнения воспользуемся данными из постов Microsoft Press Facebook, приведенными в главе 11 (обратитесь на сайт https://www.facebook.com/ microsoftpress).

ПРИМЕЧАНИЕ

Это упражнение выполняется только для небольшого набора данных из 10 сообщений. В противном случае может потребоваться плата за использование большого объема транзакций службой Text Analytics или можно достичь ограничения на уровне бесплатного пользования. Просмотрите еще раз раздел "Плюсы и минусы доступа к Cognitive Services, получаемого через Power Query" в начале этой главы.

Чтобы проанализировать сообщения Microsoft Press в Facebook, необходимо загрузить исходную рабочую книгу C11E01.xlsx с сайта https://aka.ms/DataPwrBIPivot/ downloads и сохранить ее в папке C:\Data\C11\.

1. Откройте пустую рабочую книгу или новый отчет Power BI Desktop и импортируйте рабочую книгу C11E01.xslx из папки C:\Data\C11\ в окно редактора Power Query.

2. В окне диалога Навигатор (Navigator) вберите вкладку Sheet1 и щелкните мы-шью на кнопке Изменить (Edit).

3. После открытия окна Power Query переименуйте запрос на Posts.

Обратите внимание, что сообщения Microsoft Press Facebook хранятся в столбце Message (Сообщение), а уникальный идентификатор — в столбце Post ID (Идентификатор сообщения). Таблица содержит более 10 строк, а вам нужно сохранить только первые 10 строк.

4. На вкладке Главная (Home) выберите команду Сохранить строки (Keep Rows), затем выберите команду Сохранить верхние строки (Keep Top Rows). После открытия окна диалога Сохранить верхние строки (Keep Top Rows) полю Количество строк (Number of Rows) присвойте значение 10 и щелкните на кнопке OK. Не пропустите этот шаг. Учтите, что исходный набор данных значительно превышает 10 строк.

Г") СОВЕТ

Использование операции Сохранить верхние строки (Keep Top Rows) в ваших запросах очень полезно в сценариях, выходящих за рамки этого упражнения. Такой шаг помогает ограничить набор данных, загружаемых на панель предварительного просмотра, и сократить время разработки. После завершения основных усилий по разработке можно удалить этот шаг и убедиться, что запросы работают.

Часть 2. Подготовка сообщений

Теперь покажем, как подготовить сообщения, прежде чем отправлять их службе Sentiment Analysis (Анализ настроений). Для продолжения упражнения 12.3 выполните следующие шаги:

5. В панели Запросы (Queries) правой кнопкой мыши щелкните на запросе Posts и в контекстном меню выберите команду Ссылка (Reference).

6. Переименуйте новый запрос в GetSentiment.

7. При выбранном запросе GetSentiment переименуйте столбец Message в text, а Post ID — в id.

8. Проверьте следующее:

а. Убедитесь, что текстовый столбец не содержит пустых значений. Вы можете использовать фильтр управления для его проверки. Если обнаружены пустые или нулевые значения, пробелы следует удалить.

б. Убедитесь, что в столбце id нет повторяющихся значений. Если они имеются, можно удалить их командой Удалить дубликаты (Remove Duplicates), находящейся в подменю Удалить строки (Remove Rows) на вкладке Главная (Home).

Шаги 7 и 8 очень важны. Если не переименовать эти столбцы в text и id или при пересылке пустых значений в text или повторяющихся значений в id, служба Sentiment Analysis API не будет обрабатывать сообщения.

Г") СОВЕТ

В будущем, при обработке собственных данных, при отсутствии уникальных иденти-фикаторов для ваших сообщений, можно формировать столбец индекса на вкладке Добавление столбца (Add Column) как столбец id.

9. На вкладке Главная (Home) выберите команду Выбор столбцов (Choose Columns), в открывшемся окне диалога Выбор столбцов (Choose Columns) выберите столбцы text и id. Затем щелкните мышью на кнопке OK.

Пришло время добавить столбец языка. Это требование не является обязательным, поэтому при наличии смешанных языков можно пропустить следующий шаг.

Для простоты в этом упражнении предположим, что все сообщения на английском языке. Поэтому зададим код en. В последнем разделе этой главы показано, какие коды стран доступны и как можно динамически применять код языка при наличии в столбце text нескольких разных языков.

10. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). После открытия окна диалога Настраивае-мый столбец (Custom Column) введите language в поле Имя нового столбца (New Column Name); затем в поле Пользовательская формула столбца (Custom Column Formula) введите "en". После этого щелкните мышью на кнопке OK.

Как показано на рис. 12.10, в панели предварительного просмотра редактора Power Query отображаются три столбца, которые необходимы в вызове API в формате JSON.

Рис. 12.10. Для выполнения анализа настроений необходимо располагать таблицей со столбцами id, text и language, прежде чем преобразовывать их в JSON и загружать в Cognitive Services

Часть 3. Конвертирование таблицы в формат JSON

С помощью описанных далее операций Power Query конвертирует вашу таблицу в формат JSON. Для продолжения упражнения 12.3 выполните следующие шаги:

11. Для преобразования таблицы в формат JSON необходимо придерживаться требований API и включать следующий элемент в запрос JSON:

{documents = ...}

Напомним, что в упражнении 12.1 было показано, что записи JSON определяются фигурными скобками и могут быть созданы в языке M с помощью функции Json.FromValue. Для формирования записи для документов можно заключать документы в квадратные скобки и назначать таблицу сообщений в качестве значения документов. Для этого выберите элемент управления fx рядом со строкой формул. Новый шаг, Custom1, добавляется в качестве последнего шага запроса на панели Примененные шаги (Applied Steps). Получим следующую формулу:

= #"Added Custom"

Измените формулу так:

= [documents = #"Added Custom"]

12. Для преобразования записи в файл JSON измените предыдущую формулу в строке формул, обернув ее функцией Json.FromValue. Вот модифицированная формула:

= Json.FromValue([documents = #"Added Custom"])

В панели предварительного просмотра редактора Power Query появится значок JSON, представляющий двоичный контент JSON, который только что создан. Далее будет показано, как отослать содержимое JSON.

Часть 4. Установка API Key

Теперь сформируем параметр для API Key. Для продолжения упражнения 12.3 выполните следующие шаги:

13. Перейдите на портал Azure и откройте ресурс Text Analytics. Скопируйте Key1 или Key2 из представления Keys.

14. На вкладке Главная (Home) откройте раскрывающееся меню Управление па-раметрами (Manage Parameters) и выберите команду Создать параметр (New Parameter). В открывшееся окно диалога Параметры (Parameters) внесите сле-дующие изменения:

а. В поле Имя (Name) введите TextAnalyticsAPIKey.

б. В поле Текущее значение (Current Value) вставьте ключ API, полученный на шаге 13.

в. Щелкните на кнопке OK для закрытия окна диалога Параметры (Para-meters).

Часть 5. Отправка веб-запроса

Теперь необходимо отправить веб-запрос с текстовыми сообщениями в формате JSON в службу API. Чтобы продолжить выполнение упражнения 12.3, выполните следующие шаги:

15. В окне редактора Power Query выберите запрос GetSentiment на панели Запросы (Queries) и щелкните на значке fx, под строкой формул. В результате будет создан шаг Custom2 и появится следующая формула:

= Custom1

16. В строке формул замените формулу M на следующий код:

= Web.Contents(

"https://westus.api.cognitive.microsoft.com/text/ analytics/v2.0/sentiment", [

Headers = [

#"Ocp-Apim-Subscription-Key" = TextAnalyticsAPIKey

],

Content = Custom1

]

ПРИМЕЧАНИЕ

Если ресурс Text Analytics сконфигурирован в ином месте, чем U.S. West (Запад США), замените westus в предыдущей формуле кодом своего местоположения. Код местоположения можно найти на панели обзора ресурса Text Analytics на портале Azure.

Рассмотрим код, который выведен в окне Power Query.

Предыдущая формула передает конечную точку (endpoint) Sentiment Analysis (URL-адрес) в качестве первого аргумента. Второй аргумент — это запись с полями Headers (Заголовки) и Content (Содержимое). В поле Headers передан ключ API через Ocp-Apim-Subscription-Key, как это выполнено в предыдущих упражнениях. Атрибут Content получает значение на основе содержимого JSON, сформированного на шаге Custom1.

17. Если на панели предварительного просмотра появится панель уведомлений, выберите окно Изменить учетные данные (Edit Credentials). После этого от-кроется окно диалога Веб-доступ (Web Access). Убедитесь, что установлен переключатель Анонимно (Anonymous), и щелкните на кнопке Подключиться (Connect).

18. Если увидите следующее уведомление (которое вы можете получить или не получить, в зависимости от ваших настроек конфиденциальности):

Information is required about data privacy

(Нужна информация о конфиденциальности данных).

Щелкните мышью на кнопке Продолжить (Continue) и в открывшемся окне диалога Уровни конфиденциальности (Privacy Levels) установите уровни конфиденциальности для источников данных в виде Организационный (Organizational), аналогично шагу 17 из упражнения 12.2.

19. Если вам встретилась следующая ошибка, задайте в книге или отчете Power BI игнорирование уровней конфиденциальности:

Formula.Firewall: Query 'Query1' (step 'Invoked Custom Function') references other queries or steps, so it may not directly access a data source. Please rebuild this data combination

(Запрос 'Query1' ссылается на другие запросы или шаги, поэтому не может получить непосредственный доступ к источнику данных. Пожалуйста, восстановите эту комбинацию данных)

Затем выполните следующие действия:

а. В окне редактора Power Query на вкладке Файл (File) выберите команду Параметры и настройки (Options and Settings. Затем выполните команду Параметры запроса (Query Options) в Excel или Параметры (Options) в Power BI.

б. В разделе Текущая книга (Current Workbook) в Excel или Текущий файл (Current File) в Power BI перейдите к разделу Конфиденциальность (Privacy).

в. Установите переключатель Игнорировать уровни конфиденциальности (Ignore the Privacy Levels) и щелкните мышью на кнопке OK.

г. Выберите команду Обновить просмотр (Refresh Preview) в окне редактора Power Query.

ПРИМЕЧАНИЕ

При работе с этим сценарием может потребоваться игнорирование уровней конфи-денциальности. И хотя существуют некоторые усовершенствованные способы перестройки запроса для предотвращения предыдущей ошибки, описание этой методики выходит за рамки данной книги, и усилия, необходимые для выполнения подобных модификаций, могут быть неэффективными с точки зрения затрат.

Часть 6. Обработка ответа

Теперь обратите внимание на информацию, отображаемую на панели предвари-тельного просмотра, например [location].cognitive.microsoft.com (где [location] пред-ставляет местоположение вашего ресурса Text Analytics, например westus). Это ссылка на контент в формате JSON, который возвращается API. Как будет показано далее, формат JSON легко обрабатывается в редакторе Power Query. Для продолжения упражнения 12.3 выполните следующие шаги:

20. Дважды щелкните мышью на ссылке в панели предварительного просмотра.

В панели предварительного просмотра появится запись, с полями документов и ошибок и с объектами List в качестве значений. В данный момент можно отследить, были ли возвращены какие-либо определенные ошибки Sentiment Analysis API (API анализа настроений), и перейти к ошибкам, щелкнув на соответствующие гиперссылки (List hyperlink). Предположим, что в данном упражнении ошибок нет. Позже покажем, как можно избежать распространенных ошибок.

21. Выберите объект List (список) в поле документов. Запись преобразуется в список из 10 объектов записи.

22. В группе вкладок Инструменты списка (List Tools), на вкладке Преобразование (Transform), выберите команду В таблицу (To Table). После открытия окна диалога В таблицу (To Table) сохраните выбранные настройки и щелкните на кнопке OK.

23. Расширьте столбец Column1, как показано на рис. 12.11. В панели Расширить (Expand) удостоверьтесь, что выбраны поля score и id, и отмените установку флажка Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix). Затем щелкните мышью на кнопке OK.

В результате вы получили оценки настроений и связанные с ними идентификаторы сообщений.

Часть 7. Создание настраиваемой функции

В данном разделе на основе запроса GetSentiment сформируем настраиваемую функцию.

Для продолжения упражнения 12.3 выполните следующие шаги:

24. В панели Запросы (Queries) правой кнопкой мыши щелкните на запросе GetSentiment и в контекстном меню выберите функцию Дублировать (Dupli-cate). В результате будет создан новый запрос.

Переименуйте его в FnGetSentiment.

25. На панели Запросы (Queries) выделите запрос FnGetSentiment. На вкладке Главная (Home) выберите команду Расширенный редактор (Advanced Editor).

26. В окне расширенного редактора для FnGetSentiment обратите внимание на следующий код (в листинге 12.2 код немного изменен для соответствия формату книги, но возвращает тот же результат).

Листинг 12.2. Исходный код для создания настраиваемой функции let

Source = Posts,

#"Renamed Columns" = Table.RenameColumns(

Source,{{"Post ID", "id"}, {"Message", "text"}}

),

#"Removed Duplicates" = Table.Distinct(#"Renamed Columns", {"id"}),

#"Filtered Rows" = Table.SelectRows(

#"Removed Duplicates", each [text] <> null and [text] <> ""

),

#"Removed Other Columns" = Table.SelectColumns(

#"Filtered Rows",{"id", "text"}

),

#"Added Custom" = Table.AddColumn(

#"Removed Other Columns", "language", each "en"

),

Custom1 = Json.FromValue([documents = #"Added Custom"]),

Custom2 = Json.Document(

Web.Contents(

"https://westus.api.cognitive.microsoft.com/" & "text/analytics/v2.0/sentiment", [

Headers = [

#"Ocp-Apim-Subscription-Key" = TextAnalyticsAPIKey

],

Content = Custom1

]

)

),

documents = Custom2[documents],

#"Converted to Table" = Table.FromList(

documents, Splitter.SplitByNothing(), null, null, ExtraValues.Error

),

#"Expanded Column1" = Table.ExpandRecordColumn(

#"Converted to Table", "Column1", {"score", "id"}, {"score", "id"}

)

in

#"Expanded Column1"

27. Чтобы преобразовать этот код в настраиваемую функцию, удалите первые две строки после let. Эти строки относятся к текущему набору данных и не будут нужны вам в функции. Настраиваемая функция должна отлично работать с любой таблицей с текстовой подачей материала и уникальными идентификаторами. Поэтому можно удалить ссылку на Posts (Сообщения) и выполнить переименование столбцов. При вызове функции для других наборов данных переименование выполняется вне функции.

28. Удалите также следующие две строки. Функции Table.Distinct и Table.SelectRows применялись для проверки, что позволит гарантировать, что данные не содержат повторяющихся идентификаторов или пустых текстовых значений, что приведет к ошибкам.

Теперь, когда первые четыре строки после let удалены, первая строка вашего кода будет иметь следующий вид:

# "Removed Other Columns" = Table.SelectColumns(#"Filtered Rows",{"id", "text"}),

29. Замените #”Filtered Rows” на Source и добавьте заголовок функции (Source)=> перед let. Внесите еще несколько небольших изменений:

а. Замените два экземпляра #”Expanded Column1” (выше и ниже строки in) на SentimentScores.

б. Замените два экземпляра Custom1 на JSONRequest.

в. Замените два экземпляра Custom2 на JSONResponse.

В листинге 12.3 приведена окончательная настраиваемая функция (с неболь-шими модификациями, которые выполнены с учетом формата книги).

Листинг 12.3. Окончательный код настраиваемой функции

(Source) =>

let

#"Removed Other Columns" = Table.SelectColumns(Source, {"id", "text"}),

#"Added Custom" = Table.AddColumn(

#"Removed Other Columns", "language", ach "en"),

JSONRequest = Json.FromValue([documents = #"Added Custom"]),

JSONResponse = Json.Document(

Web.Contents(

"https://westus.api.cognitive.microsoft.com" & "/text/analytics/v2.0/sentiment", [

Headers = [

#"Ocp-Apim-Subscription-Key" = TextAnalyticsAPIKey],

Content = JSONRequest

]

)

),

documents = JSONResponse[documents],

#"Converted to Table" = Table.FromList(

documents, Splitter.SplitByNothing(), null, null, ExtraValues.Error),

SentimentScores = Table.ExpandRecordColumn(

#"Converted to Table", "Column1", {"score", "id"}, {"score", "id"}) in

SentimentScores

Теперь осталось только найти применение этой функции.

Часть 8. Вызов функции FnGetSentiment

Далее показано, как вызывать настраиваемую функцию и получить оценки на-строений. Для продолжения упражнения 12.3 выполните следующие шаги:

30. Для вызова настраиваемой функции создайте новый ссылочный запрос из запроса Posts и переименуйте его как Sentiment Scores.

31. В новом запросе переименуйте столбцы из Message в text и Post ID в id. Далее, вызовите пользовательскую функцию, щелкнув мышью на кнопке fx рядом со строкой формулы и вставив следующую формулу:

= FnGetSentiment(#"Renamed Columns").

32. Перед загрузкой запроса Sentiment Scores в отчет удостоверьтесь, что тип столбца оценки (столбец score) изменен на Десятичное число (Decimal Number). Переименуйте score в Sentiment Score и id в Post ID.

33. При выбранном запросе Sentiment Scores на панели Запросы (Queries) выпол-ните команду Объединить запросы (Merge Queries) на вкладке Главная (Home) и воспользуйтесь следующими шагами для объединения запросов Sentiment Scores и Posts, как показано на рис. 12.12:

а. Откройте окно диалога Слияние (Merge). Выберите столбец Post ID в таб-лице Sentiment Scores.

б. Выберите Posts в качестве второго запроса.

в. Выберите столбец Post ID в таблице Posts.

г. В раскрывающемся меню Тип соединения (Join Kind) выберите опцию Внешнее соединение справа (все из второй таблицы, соответствие из первой таблицы) Right Outer (All from Second, Matching from First) и щелкните мышью на OK.

Слияние

Выберите таблицу и совпадающие столбцы для создания объединенной таблицы.

Sentiment Scores

Post ID Message Date URL

173252709421\_10155401936674422 New book! Exam Ref 70-347 Enabling Office 365 Servic... 02.02.2018 https://facebook.c

173252709421\_10155401826954422 ICYMI: New book! Windows 10 Step by Step, 2nd Editio... 02.02.2018 https://facebook.c

173252709421\_10155399649624422 The official study guide for the Microsoft 70-767 exam i.„ 01.02.2018 https://facebook.c

173252709421\_10155397501779422 New book! Exam Ref 70-533 Implementing ttMicrosoft... 31.01.2018 https://facebook.c

173252709421\_10155397053264422 Discover the benefits of the User Group Program. Ask y... 31.01.2018 https://facebook.c

< >

Posts

Post ID Message Date URL

173252709421\_10155401936674422 New book! Exam Ref 70-347 Enabling Office 365 Servic... 02.02.2018 https://facebook.c

173252709421\_10155401826954422 ICYMI: New book! Windows 10 Step by Step, 2nd Editio... 02.02.2018 https://facebook.c

173252709421\_10155399649624422 The official study guide for the Microsoft 70-767 exam i\_ 01.02.2018 https://facebook.c

173252709421\_10155397501779422 New book! Exam Ref 70-533 Implementing #Microsoft.\_ 31.01.2018 https://facebook.c

173252709421\_10155397053264422 Discover the benefits of the User Group Program. Ask y\_ 31.01.2018 https://facebook.c

< >

Тип соединения

Внешнее соединение справа (все из второй таблиц...

Рис. 12.12. Объедините посты с оценками настроений

Правильное внешнее объединение гарантирует, что вы не потеряете ни одной записи в данных, если служба не будет указывать в своем ответе некоторые ошибочные текстовые сообщения. (Более подробная информация о подобных потенциальных ошибках описана в следующем разделе.)

34. После объединения расширьте новый столбец Posts (Сообщения). На панели Развернуть (Expand) выберите все столбцы из таблицы Posts, кроме Post ID, который уже доступен в объединенной таблице. Теперь можно загрузить таблицу Sentiment Scores в отчет и выгрузить таблицу Posts. Сохраните отчет, поскольку он понадобится в следующем упражнении.

Файлы решения C12E03 - Solution.xlsx и C12E03 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Как избежать ошибок

Чтобы избежать ошибок при вызовах Text Analytics Sentiment API, воспользуйтесь следующими шагами перед выполнением функции FnGetSentiment для ваших наборов данных:

Убедитесь, что ваш столбец уникального идентификатора назван id.

1. Убедитесь, что столбец сообщения назван text.

2. Удалите пустые значения и нули из текстового столбца. Нет смысла отправлять пустые значения. Если вы отправляете пустые значения вместе с непустыми, то служба удалит пустые сообщения из списка результирующих документов (который расширен на шаге 23 упражнения 12.3, часть 6) и включит ошибку в список ошибок. Отправка пустых значений — излишний вызов транзакции.

3. Удалите дубликаты из столбца id, как показано на шаге 8 из упражнения 12.3, часть 8. Сохранение дубликатов идентификаторов приведет к общей ошибке для всего вызова API. Все сообщения не будут обработаны. Далее приведена ошибка, которую вы получите в связи с повторяющимися значениями в столбце id:

DataSource.Error: Web.Contents failed to get contents from 'https://westus.api.cognitive.microsoft.com/text/analytics/v2.0/sentiment' (400): Bad Request

4. Убедитесь, что ваши сообщения не содержат более 5000 символов. Более длинные сообщения приведут к отображению следующей ошибки в списке ошибок (который может быть расширен на шаге 20 упражнения 12.3, часть 6):

A document within the request was too large to be processed. Limit document size to: 5120 characters

(Документ в запросе был слишком большим для обработки. Ограничьте размер документа до 5120 символов).

Для того чтобы сократить длинные сообщения до 5000 символов, выполните следующие шаги:

а. Выделите столбец text.

б. На вкладке Главная (Home) выберите команду Разделить столбец (Split Column). Затем выполните команду По количеству символов (By Number of Characters).

в. В открытом окне диалога Разделить столбец по количеству символов (Split Column by Number of Characters), в поле Количество символов (Number of Characters) введите 5000.

г. В группе переключателей Разделить (Split) выберите переключатель Одно-кратно, как можно левее (Once, As Far Left As Possible).

5. Не передавайте более 1000 строк сообщений за вызов API. Если передадите более 1000 сообщений, получите ошибку Bad Request (Плохой запрос) для всего вы

зова. И хотя можно применить методику Сохранить верхние строки (Keep Top Rows) для ограничения числа сообщений, в следующем разделе показано, каким образом можно разделить большие наборы данных на группы по 1000 строк в каждой и выполнить анализ настроений для этих групп.

УПРАЖНЕНИЕ 12.4. Выполнение анализа настроений для больших наборов данных

Поскольку Sentiment Analysis API (API анализа настроений) может обрабатывать только 1000 сообщений за транзакцию, в этом упражнении показано, как можно запустить настраиваемую функцию FnGetSentiment для нескольких подмножеств ва-ших данных. Эта методика пригодится для многих других сценариев, в которых нужно разбивать большие наборы данных на меньшие по размеру группы и применять к подмножествам пользовательскую функцию.

Можно начать это упражнение с того места, где мы остановились в упражнении 12.3, или открыть рабочую книгу Excel C12E04.xlsx или отчет Power BI C12E04.pbix, которые доступны для загрузки на сайте https://aka.ms/DataPwrBIPivot/ downloads.

ПРИМЕЧАНИЕ

Чтобы свести к минимуму количество вызовов API, можно работать с небольшим набором данных из 100 сообщений и группировать их в 10 блоков по 10 сообщений. При реализации этого решения для своих собственных данных вы примените его к большим наборам данных и сгруппируете данные в блоки по 1000 сообщений.

6. Если вы не сохранили свою работу после выполнения упражнения 12.3, то можете воспользоваться ярлыком и открыть рабочую книгу Excel C12E04.xlsx или отчет Power BI C12E04.pbix и запустить редактор Power Query.

7. На панели Запросы (Queries) выберите запрос TextAnalyticsAPIKey, затем в панели предварительного просмотра вставьте ваш собственный ключ Sentiment Analysis API в поле Текущее значение (Current Value).

8. Выберите запрос Posts, и на панели Примененные шаги (Applied Steps) выбе-рите элемент управления настройками на шаге Kept First Rows (Сохранить первые строки). В открытом окне диалога Сохранить первые строки (Keep First Rows) измените число строк с 10 на 100 и щелкните мышью на кнопке OK.

9. На панели Запросы (Queries) щелкните правой кнопкой мыши на запросе Posts и в контекстном меню выберите команду Ссылка (Reference). Переименуйте новый запрос на Scored Posts.

10. Переименуйте столбец Post ID в id и столбец Message в text.

11. Удалите пустые значения из столбца text и повторяющиеся значения из столбца id.

12. На вкладке Добавление столбца (Add Column) выберите команду Столбец ин-декса (Index Column).

13. Выберите столбец индекса. На вкладке Преобразование (Transform) выберите команду Стандартный (Standard), затем выполните команду Целочисленное деление (Integer-Divide). После открытия окна диалога Целочисленное деление (Integer-Divide) полю Значение (Value) присвойте значение 10 и щелкните мышью на кнопке OK.

Можно заметить, что первые 10 строк получили 0 в качестве индекса. Вторые 10 сообщений получили 1 в качестве индекса и т. д.

ПРИМЕЧАНИЕ

В дальнейшем при работе с большими наборами данных, превышающими 1000 сообщений, на шаге 8 следует установить значение "1000", а не "10", которое указано здесь только в целях обучения.

14. При выбранном столбце индекса выберите команду Группировать по (Group By) на вкладке Преобразование (Transform). Откроется окно диалога Группи-ровать по (Group By). Следуйте приведенным далее шагам для установки настроек в окне диалога, как показано на рис. 12.13:

а. В поле Имя нового столбца (New Column Name) введите Subset.

б. В раскрывающемся меню Операция (Operation) выберите команду Все строки (All Rows).

в. Щелкните мышью на кнопке OK для закрытия окна диалога Группировать по (Group By).

15. Теперь на панели предварительного просмотра появится новая таблица со столбцами Index (Индекс) и Subset (Поднабор). Всего 10 строк содержат подмножества в табличных объектах.

16. На вкладке Добавить столбец (Add Column) выберите команду Вызвать на-страиваемую функцию (Invoke Custom Function).

17. После открытия окна диалога Вызвать настраиваемую функцию (Invoke Custom Function) выберите столбец FnGetSentiment в качестве значения поля Запрос функции (Function Query).

18. Убедитесь, что выбрана опция Subset в раскрывающемся меню Источник (Source), и щелкните на кнопке OK.

19. Если это необходимо, выполните шаги 17-19 из упражнения 12.3, часть 5 для изменения учетных данных и установления уровней конфиденциальности.

20. Удалите столбцы Index (Индекс) и Subset (Поднабор).

21. Расширьте столбец FnGetSentiment. В панели Развернуть (Expand) убедитесь, что выбраны поля score и id, отмените установку флажка Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix). Затем щелкните мышью на кнопке OK.

22. Следуйте шагам 32-34 в упражнении 12.3 для переименования столбцов и объединения их с помощью запроса Posts.

Г") СОВЕТ

Если используется Power BI Desktop, можно оставить запрос Scored Posts со столбцами Sentiment Score и Post ID без слияния. Загрузка их в отчет создаст автоматическое взаимно-однозначное соотношение между Posts и Sentiment Score и позволит установить связь между постами и соответствующими оценками настроения.

Теперь можно загрузить запрос Scored Posts в отчет и приступить к анализу постов на основе оценок настроений.

Можно просмотреть решения для упражнения 12.4, C12E04 - Solution.xlsx и C12E04 - Solution.pbix, которые доступны для загрузки на сайте https://aka.ms/ DataPwrBIPivot/downloads.

Сохраните отчет. Далее вы узнаете, как извлекать ключевые фразы из ваших сообщений.

Извлечение ключевых фраз

Интерфейс Azure Text Analytics API позволяет повторно использовать подходы из упражнений 12.3 и 12.4 для анализа сообщений и возврата ключевых фраз вместо оценок настроений. Синтаксис API аналогичен, и поэтому можно пропустить вводные шаги в этом разделе.

См. также

Дополнительные сведения о ссылках для Key Phrase API содержатся на сайте https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/ text-analytics-how-to-keyword-extraction.

В этом разделе показано, как изменить функцию запроса FnGetSentiment для извлечения ключевых фраз.

В главе 11 было рассмотрено, как разделять слова и анализировать количество упоминаний для слов или тем. Обращение к простому способу по применению искус

ственного интеллекта для извлечения ключевых фраз из текстовых информационных каналов позволяет вам сделать крупный шаг вперед в деле извлечения реальной сущности и значений из ваших текстовых данных.

УПРАЖНЕНИЕ 12.5. Преобразование логики настроений в ключевые фразы

Для выполнения этого упражнения можно начать с решения, полученного в упражнении 12.4, или обратиться к ярлыку и начать с рабочей книги Excel C12E04 - Solution.xlsx или с отчета Power BI C12E04 - Solution.pbix, которые доступны для загрузки на сайте https://aka.ms/DataPwrBIPivot/downloads.

1. Откройте файл C12E04 - Solution.xlsx или C12E04 - Solution.pbix и запустите редактор Power Query.

2. На панели Запросы (Queries) правой кнопкой мыши щелкните на запросе FnGetSentiment и в контекстном меню выберите команду Дублировать (Dup-licate).

3. Переименуйте новый запрос в FnGetKeyPhrases и выберите команду Расши-ренный редактор (Advanced Editor).

4. В окне расширенного редактора замените sentiment в следующей URL-ссылке:

https://...api.cognitive.microsoft.com/text/analytics/v2.0/sentiment

на KeyPhrases, как показано здесь:

https://westus.api.cognitive.microsoft.com/text/analytics/v2.0/KeyPhrases

5. Замените score дважды в следующей строке:

SentimentScores = Table.ExpandRecordColumn(

#"Converted to Table", "Column1",

{"score", "id"}, {"score", "id"})

на keyPhrases, получите следующую обновленную строку:

SentimentScores = Table.ExpandRecordColumn(

#"Converted to Table", "Column1",

{"keyPhrases", "id"}, {"keyPhrases", "id"})

ПРИМЕЧАНИЕ

Язык M чувствителен к регистру символов. Убедитесь, что keyPhrases начинается с буквы k в нижнем регистре.

6. Замените идентификатор SentimentScores на KeyPhrases в строке, показанной на ша-ге 5, а также в последней строке функции.

Обновленная настраиваемая функция FnGetKeyPhrases приведена в листинге 12.4. Откройте окно расширенного редактора и убедитесь, что получили идентичное выражение.

Листинг 12.4. Модифицированная настраиваемая функция FnGetKeyPhrases

(Source) =>

let

#"Removed Other Columns" = Table.SelectColumns(Source, {"id", "text"}),

#"Added Custom" = Table.AddColumn(#"Removed Other Columns", "language", each "en"),

JSONRequest = Json.FromValue([documents = #"Added Custom"]), JSONResponse = Json.Document(

Web.Contents(

"https://westus.api.cognitive.microsoft.com" & "/text/analytics/v2.0/KeyPhrases", [

Headers = [

#"Ocp-Apim-Subscription-Key" = TextAnalyticsAPIKey

],

Content = JSONRequest

]

)

),

documents = JSONResponse[documents], #"Converted to Table" = Table.FromList( documents, Splitter.SplitByNothing(), null, null,

ExtraValues.Error),

KeyPhrases = Table.ExpandRecordColumn(

#"Converted to Table", "Column1", {"keyPhrases", "id"}, {"keyPhrases", "id"}) in

KeyPhrases

7. В панели Запросы (Queries) щелкните правой кнопкой мыши на запросе Scored Posts и в контекстном меню выберите команду Дублировать (Duplicate).

8. Переименуйте продублированный запрос на Key Phrases. При выбранном за-просе Key Phrases (Ключевые слова) удалите все шаги после Group Rows (Группировать строки) на панели Примененные шаги (Applied Steps). Теперь вы вполне готовы применить новую настраиваемую функцию к сгруппированной таблице.

9. На вкладке Добавить столбец (Add Column) выберите команду Вызвать на-страиваемую функцию (Invoke Custom Function).

10. Если окно диалога Вызвать настраиваемую функцию (Invoke Custom Function) открыто, в поле Запрос функции (Function Query) введите FnGetKeyPhrases.

11. Переименуйте столбец Source в Subset и щелкните мышью на кнопке OK.

12. Удалите столбцы Index и Subset.

13. Расширьте столбец FnGetKeyPhrases. На панели Развернуть (Expand) удосто-верьтесь, что столбцы keyPhrases и id выделены, и отмените установку флажка Использовать исходное имя столбца в качестве префикса (Use Original Column Name As Prefix). Затем щелкните мышью на кнопке OK.

14. Удостоверьтесь, что ваши учетные данные и уровни конфиденциальности корректны (как это вы делали, выполняя шаги 17-19 в упражнении 12.3, часть 5).

15. Расширьте список результатов.

16. На панели предварительного просмотра убедитесь, что имеется столбец keyPhrases с объектами List и столбцом id. Расширьте столбец keyPhrases до новых строк, что позволит извлечь ключевые фразы из объектов List. Каждая строка будет включать комбинацию ключевой фразы и ее идентификатора.

17. Если используется Power BI Desktop, загрузите таблицу Key Phrases (Ключевые фразы) в визуальный объект Word Cloud, как показано на рис. 12.14. (Для получения дополнительной информации о том, как загрузить пользовательский визуальный элемент Word Cloud, обратитесь к упражнению 11.7 в главе 11.)

Г") СОВЕТ

Для отображения фраз из нескольких слов в облаке слов можно отключить опцию Разбиение по словам (Word-Breaking) в разделе Общие (General) представления Формат (Format) и отключить стоп-слова (Stop Words). Кроме того, убедитесь, что таблица Key Phrases (Ключевые фразы) связана с Scored Posts через общий столбец Post ID (Идентификатор сообщения). Затем измените связь, чтобы она была двунаправленной, выбрав опцию Оба (Both) в качестве Направление перекрестного фильтра (Cross Filter Direction) в окне диалога Изменить связь (Edit Relationship).

Рис. 12.14. Используйте визуальный элемент Word Cloud в Power BI, чтобы показать извлеченные ключевые фразы с учетом числа повторений

Можно просмотреть файл решения C12E05 - Solution.pbix, загрузив его с сайта https://aka.ms/DataPwrBIPivot/downloads. Убедитесь, что этот отчет будет настро-ен на игнорирование уровней конфиденциальности, как описано в упражнении 12.4, что позволит поддержать его обновление после предоставления ключа API.

Поддержка нескольких языков

На момент публикации книги вызовы API Sentiment Analysis и Key Phrase поддерживали ограниченный диапазон языков. Для просмотра полного списка поддерживаемых языков обратитесь на сайт https://docs.microsoft.com/en-us/azure/cognitive- services/text-analytics/text-analytics-supported-languages.

Если требуемый язык не поддерживается, можно с помощью API-интерфейса Translate Text API, который применялся в первой части этой главы, перевести сообщения на один из поддерживаемых языков.

Замена языкового кода

Если весь набор данных состоит из сообщений одного поддерживаемого языка, то можно задать поддерживаемый языковой код, выполнив следующие действия:

1. Найдите код вашего языка на странице поддерживаемых языков на сайте https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/text- analytics-supported-languages.

2. Откройте окно расширенного редактора для изменения настраиваемых функций FnGetSentiment и FnGetKeyPhrases.

3. Замените en кодом вашего языка в следующей строке:

#"Added Custom" = Table.AddColumn(#"Removed Other Columns", "language", each "en"),

4. Если вы в конечном итоге будете работать с наборами данных, где применяется несколько языков, то добавление настраиваемого столбца с одним значением приведет к ложному анализу настроений и пропуску ключевых фраз. Для разрешения этой проблемы можно удалить из кода строку #"Added Custom".

Для этого просмотрите следующие три фрагмента кода:

#"Removed Other Columns" = Table.SelectColumns(Source, {"id", "text"}),

#"Added Custom" = Table.AddColumn(#"Removed Other Columns", "language", each "en"),

JSONRequest = Json.FromValue([documents = #"Added Custom"]),

Удалите шаг #”Added Custom” и не забудьте "переписать" следующую строку с помощью предыдущей. Замените #”Added Custom” на #”Removed Other Columns”, как указано в следующих двух строках:

#"Removed Other Columns" = Table.SelectColumns(Source, {"id", "text"}),

JSONRequest = Json.FromValue([documents = #"Removed Other Columns"]),

Динамическое обнаружение языков

Если в текстовых информационных источниках имеются различные языки и если метод, примененный на шаге 4 в предыдущем разделе, не работает должным образом, то можно использовать вызов Language Detection API в Text Analytics для динамического определения языков отдельных сообщений, профильтровать сообщения, которые представлены с помощью поддерживаемых языков, и направить их службам Sentiment Analysis или Key Phrases API.

Служба Language Detection API описана на сайте https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/ text-analytics-how-to-language-detection.

УПРАЖНЕНИЕ 12.6. Преобразование логики настроений в обнаружение языка

Синтаксис вызова Language Detection API аналогичен вызовам Sentiment Analysis и Key Phrases API.

Воспользуйтесь такими же шагами, как в упражнении 12.5, для изменения исходной функции FnGetSentiment на новую настраиваемую функцию FnDetectLangauges (листинг 12.5).

Листинг 12.5. Код настраиваемой функции FnDetectLangauges

(Source) =>

let

#"Removed Other Columns" = Table.SelectColumns(Source,

{"id", "text"}),

JSONRequest = Json.FromValue([documents = #"Removed Other Columns"]),

JSONResponse = Json.Document(

Web.Contents(

"https://westus.api.cognitive.microsoft.com" &

"/text/analytics/v2.0/languages", [

Headers = [

#"Ocp-Apim-Subscription-Key" = TextAnalyticsAPIKey

],

Content = JSONRequest

]

)

),

documents = JSONResponse[documents],

#"Converted to Table" = Table.FromList(

documents, Splitter.SplitByNothing(), null, null, ExtraValues.Error),

DetectedLanguages = Table.ExpandRecordColumn(

#"Converted to Table", "Column1", {"detectedLanguages", "id"},

{"detectedLanguages", "id"})

in

DetectedLanguages

Поскольку три API-интерфейса Text Analytics схожи, не будем подробно останавливаться на этом упражнении для определения языка. Вместо этого можно перейти к файлам решений C12E06 - Solution.xlsx и C12E06 - Solution.pbix, которые доступны на сайте https://aka.ms/DataPwrBIPivot/ downloads. В этих файлах определяется язык многоязычных сообщений "Hello World" из упражнения 12.1 и возвращается перечень обнаруженных языков.

Поздравляем! Вы можете выполнить следующие шаги, что позволит объединить все инструменты, рассмотренные в этой главе, и выполнить расширенный анализ текста анализа настроений, а также обнаруживать ключевые фразы в многоязычных текстовых информационных каналах.

1. Определите язык каждого сообщения с помощью Detect Languages API (может применяться к блокам, каждый из которых состоит из 1000 сообщений).

2. Создайте список поддерживаемых языков.

3. Используя методы, описанные в главе 11, разбейте сообщения на промежуточные запросы:

• Сформируйте запрос со всеми сообщениями поддерживаемых языков, включая обнаруженные языки.

• Создайте запрос со всеми неподдерживаемыми языками.

4. Запустите Translation to English (перевод на английский) (или другой поддерживаемый язык) для второго запроса и добавьте результаты к первому запросу.

5. Запустите Sentiment Analysis и Key Phrase Detection (Анализ настроений и обнаружение ключевых фраз) для прилагаемых результатов.

Резюме

В этой главе показано, как переводить текст, формировать оценки настроений, извлекать ключевые фразы и обнаруживать языки в Excel и Power BI, при небольшой помощи со стороны Cognitive Services в Azure. После применения этих методов на практике к реальным данным сила искусственного интеллекта позволит извлечь новое значение из текстовых информационных каналов.

Прежде чем приступить к работе с реальными наборами данных, просмотрите раздел "Плюсы и минусы доступа к Cognitive Services, получаемого через Power Query" в начале этой главы и следуйте советам, что позволит уменьшить число чрезмерных вызовов API. Убедитесь, что ключ API не будет передан другим пользователям, и помните, что это решение, скорее всего, не будет обновлено при запланированном обновлении сервиса Power BI (если только Microsoft не сформирует его поддержку на момент прочтения данной главы). Вы по-прежнему сможете обновлять отчет через Power BI Desktop и Excel.

Возможности анализа настроений и обнаружения ключевых фраз можно увеличить на порядок, если объединить их с текущими наборами данных и сопоставить с существующими числовыми атрибутами и показателями. Благодаря этому можно по-новому взглянуть на поведение и мнения людей, узнать больше о том, какое влияние они оказывают на эффективность работы вашей компании.

В главе 13 выполняется обещание, данное в главе 11, а именно исследуется еще один аспект поведения людей — социальный — рассматривается применение данных из Facebook и их анализ в Excel и Power BI.

Глава 13

Аналитика социальных сетей

Facebook имеет уникальную возможность отвечать на вопросы людей, например: в какие суши- рестораны ходили мои друзья в Нью-Йорке в последнее время и нравились ли им они? ...Это запросы, которые можно потенциально делать с помощью Facebook, но без него делать не могли...

- Марк Цукерберг

В ЭТОЙ ГЛАВЕ ВЫ...

■ Рассмотрите, как подключиться к Facebook в Excel и Power BI для загрузки широкого диапазона сведений.

■ Выясните, как импортировать заинтересовавшие вас страницы Facebook и узнать, что они вам понравились.

■ Научитесь получать сведения о том, кто именно в вашей сети Facebook использует Power BI и подключен к Facebook через Excel и Power BI.

■ Рассмотрите, как обойти социальный график.

■ Выполните анализ произвольной страницы в Facebook и измерите уровни вовлеченности с помощью комментариев и сообщений.

■ Проанализируйте несколько страниц Facebook, применяя настраиваемые функции.

Имеется большое число источников данных, к которым можно подключиться с помощью меню Получить данные (Get Data) в Excel и Power BI. Каждый из подоб-ных источников данных может расширять ваши возможности для исследования. От простых файлов до хранилищ данных, от организационных репозиториев до онлайновых веб-сервисов — благодаря вашим новым навыкам обработки данных Power Query нет предела совершенствованию.

По мере приближения к завершению чтения книги пришло время узнать, как ис-пользовать богатый ресурс публичной информации для отработки новых навыков. Работа с реальными и разнообразными наборами данных может приводить к новым возможностям по обучению. Изучение неизвестных общедоступных данных, которые отличаются от ваших обычных рабочих данных, позволяет открывать новые методы по знакомству с проблемами данных, которые могут выходить за рамки текущего опыта.

Эта глава посвящена анализу социальных сетей, в частности Facebook. Power Query включает соединитель Facebook, который позволяет подключаться к вашему профилю или любой общедоступной странице, которую необходимо проанализировать. Информация доступна для всех, и она поможет вам попрактиковаться в подготовке неструктурированных наборов данных, в обработке разнообразного набора таблиц и необработанных данных, взятых из разнообразных и обширных наборов.

Помимо учебного потенциала, если вы являетесь аналитиком социальных сетей, для вас данная глава полезна, поскольку предлагает ознакомиться с уровнями вовлеченности пользователей и выявить тенденции социального поведения. Если вы не являетесь аналитиком социальных сетей, то наверняка вас заинтересуют иные аспекты: в этой главе вы опробуете несколько отличных инструментов для изучения брендов и конкурентов вашей компании или же методов формирования замечательных презентаций, которые можно показать коллегам.

Начало работы с соединителем Facebook

В данном разделе можно познакомиться с основами применения соединителя Facebook и выяснить, как подключаться к вашему профилю на Facebook.

Facebook предоставляет разработчикам богатый набор API-интерфейсов для взаимодействия с социальной сетью, получения информации и привлечения пользователей. Основной API, Facebook Graph API, разрешает приложениям использовать социальный граф Facebook (обратитесь на сайт https://developers.facebook.com/ docs/graph-api/). Как и сама социальная сеть, социальный граф позволяет приложениям проходить узлы и ребра в сети, выполнять считывание и запись данных от имени пользователей, страниц или групп и в соответствии с разрешениями, предоставленными пользователями или администраторами страниц и групп. И хотя некоторые конечные точки в социальном графе защищены от посторонних глаз, многие данные открыты для всех.

Чтобы подключиться к данным Facebook через приложение, необходимо войти в свою учетную запись Facebook и предоставить приложению разрешение на чтение и/или запись данных от вашего имени. В следующем упражнении вы выполните простые шаги для подключения к профилю Facebook с помощью соединителя Facebook в Power Query, который является одним из многих важных приложений Facebook.

УПРАЖНЕНИЕ 13.1.

Поиск страниц, которые вам понравились

В данном упражнении вы импортируете понравившиеся страницы Facebook и находите нужные даты. Затем показано, как извлекать URL профиля и изображения из идентификатора Facebook каждой из рассматриваемых страниц. Чтобы начать упражнение, необходима учетная запись Facebook, если она отсутствует, то предварительно создайте ее.

ПРИМЕЧАНИЕ

Если у вас нет учетной записи Facebook, данное упражнение выполнить невозможно. Пожалуйста, зарегистрируйтесь на Facebook для продолжения работы.

1. Установите связь Facebook с помощью либо Excel, либо Power BI Desktop (рис. 13.1):

В Excel: откройте пустую рабочую книгу. На вкладке Данные (Data) выполните команды Получить данные (Get Data) Из веб-служб (From Online Services) Из Facebook (From Facebook).

В Power BI Desktop: откройте пустой отчет. На вкладке Главная страница (Home) выберите команду Получить данные (Get Data). После открытия окна диалога Получить данные (Get Data) выберите команду Веб-службы (Online Services); затем выполните прокрутку вниз, выберите команды Facebook Подключиться (Connect). Также можно ввести Facebook в поле поиска, затем выбрать команду Facebook и щелкнуть мышью на кнопке Подключиться (Connect).

2. Если вы впервые подключаетесь к Facebook и видите окно диалога с предупреждением о потенциальных изменениях, которые могут произойти в API, то установите флажок Не предупреждать меня больше об этом соединителе (Don't Warn Me Again for This Connector) и щелкните на кнопке Продолжить (Continue).

3. Откроется окно диалога Facebook. Здесь можно выбрать сущность Facebook, которую необходимо импортировать. По умолчанию, me является предлагаемой

сущностью. Это ярлык для вашего собственного профиля в Facebook. Можно установить различные имена объектов Facebook в данном поле. Например, microsoftpress — имя объекта для страницы Microsoft Press в Facebook. Пока ос-тавьте значением по умолчанию me и щелкните мышью на кнопке OK.

Г") СОВЕТ

Окно диалога Facebook включает раскрывающееся меню Подключение (Connection), которое позволяет импортировать определенные части в график (graph), такие как друзья, посты или предпочтения (лайки). Вследствие постоянных изменений в API Facebook не все подобные преимущества полезны. Например, хотя и можно применять комбинацию "я" и "лайки", как описано в этом упражнении, попытка импортировать "лайки" в другие имена объектов Facebook приведет к пустым данным.

4. Откроется окно диалога Доступ к Facebook (Facebook Access). Если вы впервые подключаетесь к Facebook с помощью Excel или Power BI, выполните следующие шаги:

а. Выберите команду Вход (Sign In), и если вас попросят предоставить ваши учетные данные Facebook, сделайте это.

б. В открывшемся окне диалога Facebook рассмотрите тип разрешений, которые получит приложение Microsoft Power BI: ваш общедоступный профиль, список друзей, сообщения о временной шкале, обновления статуса, личное описание, лайки и адрес электронной почты. Измените разрешения, если хотите, и щелкните на кнопке Продолжить (Continue).

ПРИМЕЧАНИЕ

В случае общедоступных публикаций книг или отчетов Power BI ваши учетные данные Facebook не сохраняются в отчете. При следующем обновлении отчета с нового компьютера пользователю будет предложено войти в Facebook. Если отчет публикуется в службе Power BI и необходимо обновить его через службу Power BI, необходимо в настройках набора данных указать учетные данные Facebook.

в. Если вы располагаете страницей или приложением Facebook, имеете права доступа администратора, и открывается вторая подсказка с запросом дополнительных разрешений, то щелкните мышью на кнопке ОК.

5. В окне диалога доступа к Facebook щелкните на кнопке Подключение (Con-nect).

6. После открытия редактора Power Query на панели предварительного просмотра обратите внимание на запись со следующими ключами и соответствующими значениями: name, connections и id (имя, соединения и идентификатор), рис. 13.2.

Несмотря на то что данная запись пока не раскрывает ценных сведений, необходимо обсудить несколько важных элементов, доступных на этом этапе. В строке формул обратите внимание на следующую строку:

= Facebook.Graph("https://graph.facebook.com/v2.8/me")

M-функция Facebook.Graph получает один аргумент, URL. Для того чтобы узнать, как можно манипулировать этой URL-ссылкой, необходимо изучить несколько основных принципов Graph API. (Имеется ряд других интересных моментов, которые необходимо знать о Graph API, но они выходят за рамки этой главы.)

Важной первой частью URL является версия API (в данном случае версия 2.8). Для многих случаев Facebook предоставляет обновленный контент при выпуске новых версий API. Можно попытаться изменить номер версии в URL для получения доступа к этим новым элементам. Также важно отметить, что версия API ограничена во времени. Чтобы узнать, когда истечет срок действия вашей версии, перейдите на официальную страницу Facebook для Graph API Changelog: https://developers.facebook.com/docs/graph-api/changelog.

Рис. 13.2. Результаты API Graph Facebook изначально отображаются в виде записи в редакторе Power Query

СОВЕТ

Если планируется публикация обновляемых отчетов с помощью соединителя Facebook, следует внимательно просмотреть страницу Changelog. К версии, использованной в этой главе, а именно к версии 2.8, доступ прекратился 18 апреля 2019 года. Чтобы гарантировать обновление ваших отчетов после этой даты, можно обновить строку формул с последней версией и проверить, что измененный запрос работает как поло-жено.

Вторая важная часть в URL-ссылке — это конечная точка (endpoint) Graph API — в данном случае me. Эта конечная точка возвращает информацию о пользователе. Также можно формировать более сложные вызовы Graph API, применять параметры для уточнения вызовов и извлекать конкретные наборы данных. Несколько общих примеров приведены далее в этой главе, в упражнении 13.2.

7. На панели предварительного просмотра найдите ссылку Record (Запись) для полей соединений и щелкните на ней мышью.

Панель предварительного просмотра содержит следующие ключи и соответствующие им объекты таблицы: книги, каналы, друзья, игры, лайки, фильмы, разрешения, посты и телевидение. Каждый из данных объектов называется ребром

в Facebook Graph API. Выбор ссылки любого из этих объектов таблицы позволяет выполнять детализацию до соответствующих ребер и загружать соответствующие данные.

Facebook Graph API имеет большее число ребер, чем можно видеть на панели предварительного просмотра, на шаге 7.

Полный список доступен на сайте https://developers.facebook.com/docs/graph- api/reference/v3.1/user.

Вспомните, что на шаге 4б предоставлены специальные разрешения для приложения Power BI при доступе к своему профилю. Приложение запрашивает не все разрешения. В результате вы не сможете прочитать все ребра, которые доступны на Facebook. Следующие шаги демонстрируют это ограничение.

8. Удалите последний шаг на панели Примененные шаги (Applied Steps) и изме-ните формулу в строке формул, добавляя /family в URL-ссылку:

= Facebook.Graph("https://graph.facebook.com/v2.8/me/family")

Теперь после нажатия клавиши <Enter> на панели предварительного просмотра будет отображен пустой результат. Причина в том, что приложение (app) Power BI не запрашивает разрешения user\_relationship. В результате будет получено ребро семьи (family edge).

9. Измените строку формул с учетом первоначального состояния:

= Facebook.Graph("https://graph.facebook.com/v2.8/me")

Повторите шаг 7 для перехода к объекту Table из поля connections (подключения).

10. Для перехода к понравившимся страницам Facebook выберите гиперссылку Table рядом с лайками, как показано на рис. 13.3. На панели предварительного просмотра отобразятся понравившиеся вам страницы Facebook, включая для каждой из страниц столбец id (идентификатор Facebook) и столбец created\_ time (который указывает, что страница понравилась).

Из столбца created\_time можно извлечь дату, время и час, что полезно пред-ставить в виде отдельных столбцов.

11. Для добавления нового столбца Date (Дата), созданного на основании столбца created\_time, сначала измените его тип на Date/Time/Timezone (Дата/ Время/Часовой пояс) и при выбранном столбце перейдите на вкладку Добавление столбца (Add Column) и выберите раскрывающееся меню Дата (Date). Затем выполните команду Только дата (Date Only) для добавления нового столбца Date.

12. Теперь, при наличии нового столбца Date (Дата), измените тип столбца created\_time на Время (Time). Теперь можно извлекать часть часа из столбца Время (Time). Выберите столбец create\_time, а затем на вкладке Добавление столбца (Add Column) выполните команды Время (Time) Час (Hour) Час (Hour). В новом столбце Час (Hour) будет отображаться время суток, что позволит создавать диаграммы, которые показывают наиболее активные часы дня, когда вам больше всего нравятся новые страницы Facebook. (Этот анализ можно увидеть в файле шаблона Power BI, который предоставляется в конце как решение данного упражнения.)

13. Удалите столбец object\_link и переименуйте первые три столбца Facebook Facebook Page, ID и Time.

Г") СОВЕТ

Объект Facebook ID теперь сохраняется в столбце ID, что довольно полезно. Это уни-кальный идентификатор, который можно применять для создания связей между таб-лицами и формирования пользовательских столбцов, которые обогащают отчеты дополнительной информацией. Например, можно добавить пользовательский столбец URL страницы в редакторе Power Query с помощью следующей формулы:

= "https://www.facebook.com/" & [ID].

14. Если применяется Power BI, следуйте предыдущему совету для добавления столбца Page URL (URL страницы). Также добавьте URL-адрес изображения страницы, выполняя следующие шаги:

а. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). Откроется окно диалога Настраиваемый столбец (Custom Column).

б. В поле Имя нового столбца (New Column Name) введите Picture.

в. В окно Формула пользовательского столбца (Custom Column Formula) введите следующую формулу и щелкните мышью на кнопке OK для закрытия окна диалога:

= "https://graph.facebook.com/" & [ID] & "/picture?type=large"

Г") СОВЕТ

В Power BI Desktop можно классифицировать столбец URL-адреса как веб-изображение или веб-URL, а также использовать визуальные элементы, такие как срез, таблица или матрица, для отображения фактических изображений и использования гиперссылок в отчете.

15. Переименуйте запрос на Facebook Pages I Like и загрузите его в рабочую книгу или отчет Power BI. Можно воспользоваться сводной диаграммой в Excel или графиком Power BI для просмотра числа страниц, которые вам понравились за все время. Данные в этом отчете эквивалентны данным, которые можно видеть при переходе на сайт https://www.facebook.com/me/likes.

Файлы решения C13E01 - Solution.xlsx и C13E01 - Solution.pbit можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Анализ предпочтений ваших друзей

В данном разделе кратко рассматривается один из самых популярных сценариев для анализа в Facebook: возможность анализировать своих друзей. В первом упражнении показано, как импортировать своих друзей и их друзей, и формировать запросы, которые позволят визуализировать социальные графики. Во втором упражнении показано, как импортировать страницы, которые нравятся вашим друзьям.

В Facebook Graph API можно использовать ребро (edge) друзей, что позволит рассмотреть социальный график. К сожалению, для вас, как для аналитика (и, к счастью, для любого пользователя, который заинтересован в конфиденциальности), приложение Facebook может проходить только через график Facebook, начиная с друзей, которые одобрили приложение, а затем переходить уже к их друзьям, которые также одобрили приложение. В упражнении 13.2 рассмотрен именно такой сценарий.

УПРАЖНЕНИЕ 13.2.

В поисках друзей Power BI и их друзей

Выполните приведенные в предлагаемом упражнении шаги, где соединитель Facebook служит для отыскания ваших друзей в Facebook, а также их друзей и формирования визуальных элементов, как показано на рис. 13.4 (в конце данного упражнения доступно решение Power BI).

Как будет показано, отображаются только ваши друзья, которые импортировали данные Facebook через Power Query (в Excel или Power BI).

1. Откройте пустую новую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из веб-служб (From Online Services) Из Facebook (From Facebook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data). После открытия окна диалога Получить данные (Get Data) выберите команду Веб-службы (Online Services); затем выполните прокрутку вниз, выберите команду Facebook и щелкните на кнопке Подключе-ние (Connect). Также можно ввести Facebook в поле поиска и затем выберите Facebook и щелкните мышью на кнопке Подключение (Connect).

2. В окне диалога Facebook сохраните me в первом поле text, в раскрывающемся меню Подключение (Connection) выберите команду Друзья (Friends) и щелкни-те на кнопке OK. Если отобразится окно диалога с предварительным просмотром данных, щелкните на кнопке Изменить (Edit). Теперь можно увидеть список друзей на панели предварительного просмотра.

ПРИМЕЧАНИЕ

В редакторе Power Query если видите пустую таблицу на панели предварительного просмотра, то не сможете продолжить это упражнение. Сначала следует подключиться к Facebook с пользователями, которые задействовали соединитель Facebook в Excel или Power BI. Следующие шаги предполагают, что такие друзья на Facebook имеются. Если вы этого не сделаете, попросите друзей выполнить одно из упражнений из данной главы, и вы увидите их на панели предварительного просмотра.

3. Для отыскания друзей своих друзей, использующих соединитель Facebook, разверните столбец object\_link. Этот столбец показывает запись элементов, которую соединитель Facebook позволяет извлечь. Столбец connections содержит все ребра Graph API, которые можно импортировать из пользовательского интерфейса.

На панели Развернуть (Expand) отмените выбор всех столбцов, кроме connections. Удостоверьтесь, что флажок Использовать исходное имя столбца в качестве префикса (Use Original Column Name As Prefix) не установлен, и щелкните мышью на кнопке OK.

4. Разверните столбец connections. На панели Развернуть (Expand) отмените вы-бор всех столбцов, кроме friends. Удостоверьтесь, что флажок Использовать исходное имя столбца в качестве префикса (Use Original Column Name As Prefix) не установлен, и щелкните на кнопке OK.

5. Разверните столбец friends. На панели Развернуть (Expand) отмените выбор всех столбцов, кроме name и id. Удостоверьтесь, что флажок Использовать ис-ходное имя столбца в качестве префикса (Use Original Column Name As Prefix) не установлен, и щелкните на кнопке OK.

Теперь на панели предварительного просмотра появились два новых столбца: name.1 и id.1. Для каждого из ваших друзей в столбце name теперь может быть несколько строк, каждая с уникальным user id — идентификатором пользователя (в столбце id.1) для представления друзей вашего друга. Вследствие того, что через ребро друзей могут отображаться только пользователи приложения, в двух новых столбцах содержатся только друзья друзей, которые используют коннектор Facebook. Также можно видеть друзей первой степени в столбце name.1.

Почему вы видите друзей первой степени в столбце, который должен показы-вать только друзей второй степени? Ответ прост: для каждого друга первой степени, который имеет общих с вами друзей, вы увидите этих общих друзей в столбце name.1.

Поэтому имеется два интересных сценария для реализации. Во-первых, вы можете добавлять новый столбец, который будет отмечать общих друзей в столбце name.1. Этот шаг позволяет проанализировать, сколько общих друзей у каждого из ваших друзей, а затем перейти к этим общим друзьям. Во-вторых, если необходимо построить социальный график, то можно отфильтровать строки с общими друзьями. Эти строки избыточны, так как у вас уже есть другие строки в таблице, где общий друг из столбца name.1 находится в столбце name других строк.

6. Для идентификации общих друзей на вкладке Добавление столбца (Add Column) выберите команду Настраиваемый столбец (Custom Column). В окне диалога Настраиваемый столбец (Custom Column) выполните такие операции: а. Введите IsMutual в поле Имя нового столбца (New Column Name).

б. Введите приведенную далее формулу в поле Формула пользовательского столбца (Custom Column Formula) и щелкните мышью на кнопке OK для за-крытия окна диалога:

List.Contains(#"Expanded friends"[name], [name.1])

Обратите внимание, что новый столбец IsMutual включает значения TRUE или FALSE в панели предварительного просмотра. Для удаления лишних строк в отчете социального графа можно применить фильтр, чтобы в IsMutual оставались только строки с ложными значениями.

Кратко рассмотрим формулу в окне диалога Настраиваемый столбец (Custom Column). Функция List.Contains возвращает значение true, если элемент во втором аргументе найден в списке для первого аргумента. В главе 9 было показано, что столбцы могут быть представлены в виде списков, если обращаться к имени столбца в скобках после имени таблицы. В данном случае таблица в #"Expanded friends" возвращает столбец name в виде списка, используя аргумент #”Expanded

friends”[name]. Второй аргумент в функции List.Contains — [name.1] представляет имя друга второй степени, которое мы ищем в списке.

Несмотря на то что нельзя далее проходить по графику за пределами ваших друзей второй степени, это позволяет формировать социальные графы в Power BI, такие как на рис. 13.4.

Можно загрузить файл решения C13E02 - Solution.pbit (показан на рис. 13.4) или запрос в файле C13E02 - Solution.xlsx. Оба файла доступны на сайте https:// aka.ms/DataPwrBIPivot/downloads.

СОВЕТ

Имеются более эффективные способы обхода социального графа с помощью расши-ренных вызовов API. Например, при копировании в одну строку следующая функция языка M вызывает 100 ваших друзей и 50 первых друзей каждого из ваших друзей:

=Facebook.Graph("https://graph.facebook.com/v3.0/me?fields= friends.limit(100) {name,id,friends.limit(50){name,id}}").

УПРАЖНЕНИЕ 13.3. Найдите страницы, которые понравились вашим друзьям

Это упражнение дополняет упражнение 13.2. Вместо того чтобы импортировать друзей своих друзей, вы импортируете любимые страницы своих друзей. Следуя ограничениям из предыдущего упражнения, вы сможете импортировать только неполное подмножество своих друзей — будут отображаться только те, кто импортировал данные Facebook через Power Query. Но можно узнать, какие страницы им нравятся и каковы их общие интересы, что отражено на их страницах, как показано на рис. 13.5.

Рис. 13.5. Свою страницу можно дополнить фотографиями из профиля Facebook и гиперссылками. Ознакомьтесь с файлом решения C13E03 - Solution.pbit

7. Откройте пустую новую рабочую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните последовательность команд По-лучить данные (Get Data) Из веб-служб (From Online Services) Из Facebook (From Facebook).

В Power BI Desktop: на вкладке Главная (Home) выберите команду Получить данные (Get Data). После открытия окна диалога Получить данные (Get Data) выполните команду Веб-службы (Online Services); затем прокрутите вниз, вы-берите команды Facebook Подключение (Connect). Также можно ввести Facebook в поле поиска и затем выбрать Facebook и щелкнуть на кнопке Под-ключение (Connect).

8. В окне диалога Facebook сохраните me в первом текстовом поле, выберите вкладку Друзья (Friends) в раскрывающемся меню Подключение (Connection) и щелкните мышью на кнопке OK. После того как отобразится окно диалога, которое поддерживает предварительный просмотр данных, щелкните на кнопке Изменить (Edit).

9. Для определения страниц, которые предпочитают ваши друзья, разверните столбец object\_link. На панели Развернуть (Expand) отмените выбор всех столбцов, исключая столбец connections. Удостоверьтесь, что флажок Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix) не установлен, и щелкните на кнопке OK. Теперь разверните столбец connections.

На панели Развернуть (Expand) отмените выбор всех столбцов, кроме likes. Удостоверьтесь, что флажок Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix) не установлен, и щелкните на кнопке OK.

10. Переименуйте запрос Base, в панели Запросы (Queries) правой кнопкой мыши щелкните на запросе Base и в контекстном меню выберите команду Ссылка (Reference).

11. Переименуйте новый запрос в Friends и, пока запрос Friends выбран, удалите столбец likes.

12. На панели Запросы (Queries) правой кнопки мыши щелкните на запросе Base и в контекстном меню выберите команду Ссылка (Reference). Переименуйте новый запрос как Pages и при выбранном запросе Pages удалите столбцы name и id.

13. При выбранном запросе Pages разверните столбец likes. На панели Развернуть (Expand) отмените выбор всех столбцов, исключая name и id. Удостоверьтесь, что флажок Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix) не установлен, и щелкните на кнопке OK.

14. Чтобы избежать рассмотрения нескольких страниц, выберите столбец id и на вкладке Главная (Home) выполните команды Удалить строки (Remove Rows) Удалить дубликаты (Remove Duplicates).

Затем, поскольку у некоторых из ваших друзей нет любимых страниц, нужно удалить пустую строку.

15. Выберите элемент управления фильтром столбца id и выполните команду Удалить пустые (Remove Empty).

Пришло время создать новую таблицу, которая будет отображать ваших друзей и их любимые страницы.

16. На панели Запросы (Queries) правой кнопкой мыши щелкните на запросе Base и в контекстном меню выберите команду Ссылка (Reference). Переименуйте новый запрос в Friends and Pages и при выбранном запросе Friends and Pages удалите столбец name и переименуйте столбец id в friend id.

17. При выбранном запросе Friends and Pages разверните столбец likes. На панели Развернуть (Expand) отмените выбор всех столбцов, кроме id. Удостоверьтесь, что не установлен флажок Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix), и щелкните на кнопке OK.

18. Переименуйте столбец id в page id.

Теперь дополним запросы Pages and Friends профилем и изображением URL- ссылок, как показано в упражнении 13.1.

Вот формула для профиля URL:

"https://www.facebook.com/" & [id]

А так выглядит формула для URL рисунка:

"https://graph.facebook.com/" & [id] & "/picture?type=large"

19. Загрузите запросы в вашу рабочую книгу Excel или отчет Power BI и выполните следующие шаги:

а. Выгрузите запрос Base (как было показано в главе 3, упражнение 3.1).

б. В PowerPivot, в Excel или в представлении Связи (Relationships) в Power BI Desktop создайте связь между столбцом id в запросе Friends и столбцом friend id в запросе Friends and Pages.

Затем создайте связь между столбцом id в запросе Pages и страницей столбца id в запросе Friends and Pages.

Можно загрузить файл решения C13E03 - Solution.pbit (показан на рис. 13.5) или запросы из файла C13E03 - Solution.xlsx. Оба файла доступны на сайте https:// aka.ms/DataPwrBIPivot/downloads.

СОВЕТ

Имеются более эффективные способы обхода социального графа с применением расширенных вызовов API. Например, следующая функция языка M вызывает до 100 ваших друзей (которые использовали коннектор Facebook в Power Query) и до 50 любимых страниц (при использовании однострочного выражения):

=Facebook.Graph("https://graph.facebook.com/v3.0/me?fields= friends.limit(100) {name,id,likes.limit(50){name,id}}").

Анализ страниц Facebook

Один из наиболее распространенных бизнес-сценариев с соединителем Facebook в Excel и Power BI — анализ участия пользователей на страницах Facebook. В следующих упражнениях описано, как можно просматривать публичные посты с любой страницы Facebook и анализировать количество комментариев и публикаций для каждого из импортированных постов.

Начнем с базового упражнения по обработке сообщений и комментариев, которое позволяет извлекать дополнительную информацию из данных и комментариев пользователей и применять методы анализа текста, описанные в главе 11 и 12 для любой страницы Facebook. Далее в этом разделе показано, как можно эффективно извлекать комментарии и подсчитывать количество комментариев и публикаций для анализа вовлеченности пользователей. Эти инструменты вместе с уже освоенным методом обработки текста полезны при анализе общественного мнения и настроения бренда в отношении продуктов вашей компании или конкурентов, как показано на рис. 13.6.

Рис. 13.6. С помощью Power Query можно анализировать любую страницу Facebook и применять аналитику текста (см. главу 12) для получения оценки настроения, связанного с сообщениями и комментариями

УПРАЖНЕНИЕ 13.4. Извлечение сообщений и комментариев из страниц Facebook, основная возможность

В главе 11 были проанализированы сообщения на странице Microsoft Press в Facebook. В данном упражнении показано, как импортировать сообщения и ком

ментарии с любой страницы Facebook. Как известно, эти шаги неэффективны при анализе всех страниц и могут быть слишком медленными для загрузки страниц с большим количеством комментариев.

1. Откройте пустую новую рабочую книгу или новый отчет Power BI.

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get Data) Из веб-служб (From Online Services) Из Facebook (From Facebook).

В Power BI Desktop: на вкладке Главная страница (Home) выберите команду Получить данные (Get Data). После открытия окна диалога Получить данные (Get Data) выполните команду Веб-службы (Online Services); затем прокрутите вниз и выберите команды Facebook Подключение (Connect). Также можно ввести Facebook в поле поиска, выбрать Facebook и щелкнуть мышью на кнопке Подключение (Connect).

2. Откроется окно диалога Facebook. Введите microsoftpress в первое текстовое поле, выберите опцию Posts в раскрывающемся меню Подключение (Con-nection) и щелкните мышью на кнопке OK для закрытия этого окна диалога. В следующем окне предварительного просмотра щелкните на кнопке Изменить (Edit) для запуска редактора Power Query.

ПРИМЕЧАНИЕ

Можно повторить это упражнение на любой другой странице Facebook. На шаге 2 замените microsoftpress в первом текстовом поле на имя страницы Facebook. Чтобы найти имя страницы, откройте страницу Facebook в браузере и скопируйте имя корневой папки в URL. Например, URL официальной страницы Power BI на Facebook: https://www.facebook.com/microsoftbi.

Чтобы проанализировать эту страницу, необходимо ввести microsoftbi в текстовое поле шага 2. Также можно изменить имя страницы Facebook позже, изменяя URL- ссылку в функции Facebook.Graph в строке формул шага Source. Например: Facebook.Graph(“https://graph.facebook.com/v2.8/microsoftbi/posts”).

На панели предварительного просмотра отобразятся следующие столбцы: message, created\_time, id, object\_link и story.

3. Для извлечения столбцов Date, Time и Hour из столбца created\_time повторите шаги 11 и 12 из упражнения 13.1.

4. Для того чтобы получить удобные для пользователя имена столбцов, переименуйте столбцы message, created\_time, id и story в столбцы Message, Time, ID и Story, соответственно.

В шагах 6-14 показаны основные шаги для извлечения комментариев Facebook. И хотя эти шаги тривиальны в пользовательском интерфейсе, обнаружится, что они приводят к медленной загрузке и чрезмерным вызовам API в Facebook, особенно для активных страниц Facebook, таких как Microsoft Press. В упражнении 13.5 показано, как улучшить эти запросы. А пока рассмотрим базовые, но более медленные шаги. Следует предупредить вас, что вследствие чрезмерных вызовов Power Query к API Graph Facebook в данном методе можно получить следующую ошибку:

Facebook: (#17) User request limit reached

В этом случае необходимо прекратить работу над отчетом и подождать несколько часов. Facebook реализует механизм регулирования для блокировки чрезмерных вызовов API, сделанных пользователями. К счастью, после выполнения шага 5 ошибки будут предотвращены.

СОВЕТ

При работе с медленными запросами рекомендуется уменьшить число строк в качестве временного шага, что позволит минимизировать время загрузки запроса. Для этого примените окно диалога Сохранить верхние строки (Keep Top Rows). После завершения разработки можно удалить этот шаг и загрузить весь набор данных.

5. На вкладке Главная (Home) выберите команду Сохранить строки (Keep Rows) Сохранить верхние строки (Keep Top Rows). Введите 100 в текстовое поле Количество строк (Number of Rows) окна диалога Сохранить верхние строки (Keep Top Rows) и щелкните мышью на кнопке ОК. Теперь, при наличии лишь 100 строк, можно приступить к выполнению упражнения и предотвра-тить медленную загрузку и ошибки при обработке запроса. Помните, что затем нужно будет отменить этот шаг после завершения этапа подготовки данных.

6. Разверните столбец object\_link. В панели Развернуть (Expand) отмените выбор всех столбцов, кроме соединителей; убедитесь, что флажок Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix) не установлен, и щелкните мышью на кнопке ОК.

7. Столбец object\_link изменяется на столбец connections. Разверните его и на па-нели Развернуть (Expand) убедитесь, что выбрано поле comments. Убедитесь, что флажок Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix) не установлен, и щелкните на кнопке OK.

Теперь имеется новый столбец comments вместо исходного столбца connections. В каждой строке отображаются объекты Table, которые содержат таблицу comments, полученную каждым сообщением.

ПРИМЕЧАНИЕ

На шаге 7 на панели Развернуть (Expand) можно заметить, что поле likes также доступно. Если развернуть его, то Power Query вернет пустые объекты таблицы. Соединитель Facebook не может загрузить число лайков, полученных каждым сообщением. Для того чтобы просмотреть число лайков, необходимо разработать собственный пользовательский соединитель или использовать сложные M-запросы, которые не рассматриваются в этой книге.

Все повторяющиеся значения в столбце ID рекомендуется удалить. Далее в этом упражнении необходимо создать соответствие между запросами Posts и Comments, предполагая, что столбец ID в Posts содержит уникальные значения. Без следующего шага загрузка не удастся.

8. Выберите столбец ID и на вкладке Главная (Home) выполните команды Уда-лить строки (Remove Rows) Удалить дубликаты (Remove Duplicates).

9. Переименуйте запрос на Posts - Base.

10. На панели Запросы (Queries) правой кнопкой мыши щелкните на запросе Posts - Base и в контекстном меню выберите команду Ссылка (Reference). Переименуйте новый запрос на Posts.

11. Выберите запрос Posts на панели Запросы (Queries) и удалите столбец comments.

12. На панели Запросы (Queries) правой кнопкой мыши щелкните на запросе Posts - Base и в контекстном меню выберите команду Ссылка (Reference). Пе-реименуйте новый запрос в Comments.

13. При выбранном запросе Comments в панели Запросы (Queries) выполните команду Выбор столбцов (Choose Columns) на вкладке Главная (Home). В окне диалога Выбор столбцов (Choose Columns) выберите столбцы ID и comments и щелкните мышью на кнопке OK.

14. Оставаясь в запросе Comments, разверните столбец comments и в панели Раз-вернуть (Expand) отмените выбор всех полей, кроме message и id. Убедитесь, что флажок Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix) не установлен (как на рис. 13.7), и щелкните на кнопке OK.

15. Переименуйте столбец id.1 в Comment ID и переименуйте столбец message в Comment.

Можно заметить, что столбцы Comment и Comment ID включают пустые зна-чения. Это результат того, что некоторые посты не имеют комментариев пользователей. Вы можете отфильтровать эти строки.

16. Выберите элемент управления фильтром для столбца Comment и выберите команду Удалить пустые (Remove Empty) для фильтрации строк без коммен-тариев.

Можно дополнить запрос Posts and Comments и включить в него URL-ссылку.

17. Выберите запрос Posts и следуйте таким шагам:

а. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). Откроется окно диалога Настраиваемый столбец (Custom Column).

б. В поле Имя нового столбца (New Column Name) введите Profile.

в. В поле Формула пользовательского столбца (Custom Column Formula) введите следующую формулу:

"https://www.facebook.com/" & [ID]

г. Щелкните мышью на кнопке OK для закрытия окна диалога.

18. Выберите запрос Comments и выполните шаги 17 а-в с одним исключением в строке формул:

замените ID на Comment ID:

"https://www.facebook.com/" & [Comment ID]

19. Перед загрузкой ваших запросов можно удалить шаг Сохранить первые стро-ки (Kept First Rows) на панели Примененные шаги (Applied Steps) для запроса Posts - Base. Напомним, что на шаге 5 сохранилось только 100 строк, что по-зволит избежать ошибок и медленной загрузки на панели предварительного просмотра во время разработки отчета.

20. Теперь вы можете загрузить запросы в свою книгу Excel или отчет Power BI, выполняя следующие шаги:

а. Выгрузите запрос Posts - Base (как показано в главе 3, упражнение 3.1).

б. В Power Pivot в Excel или в представлении Связи (Relationships) в Power BI Desktop создайте соответствие между столбцом ID в запросе Posts и столбцом ID в запросе Comments.

21. Сохраните отчет; вы будете использовать его в следующем разделе и узнаете, как ограничить отчет по времени и улучшить время загрузки.

Файлы решения C13E04 - Solution.xlsx и C13E04 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

Сокращение обзора:

фильтрация результатов по времени

В упражнении 13.4 показано, что время загрузки данных Facebook иногда велико, особенно если выполняется анализ активной страницы Facebook с большим количеством постов и комментариев. Для того чтобы ограничить отчет определенным периодом и уменьшить время загрузки, можно настроить запрос и задать парамет

ры "с тех пор и до тех пор" в URL Facebook.Graph. Рассмотрим это на примере файла из упражнения 13.4.

1. Откройте рабочую книгу, которая сохранилась после выполнения упражне-ния 13.4, или один из файлов решения C13E04 - Solution.xlsx или C13E04 - Solution.pbix на сайте https://aka.ms/DataPwrBIPivot/downloads.

2. Запустите редактор Power Query.

3. На панели Запросы (Queries) выберите запрос Posts - Base.

4. На панели Примененные шаги (Applied Steps) выберите шаг Source. Отобра-зится следующая формула:

= Facebook.Graph("https://graph.facebook.com/v2.8/microsoftpress/posts")

Для фильтрации результатов и отображения постов, начиная с 2017 года, до-бавьте URL-параметр since=2017-01-01, следом за знаком вопроса, как показано в формуле (которая скопирована как однострочное выражение):

= Facebook.Graph("https://graph.facebook.com/v2.8/microsoftpress/posts?since=2017-01-01")

Для фильтрации результатов и отображения постов только за 2017 год добавьте URL-параметр until=2017-12-31 как показано в формуле (скопирована как однострочное выражение):

= Facebook.Graph("https://graph.facebook.com/v2.8/microsoftpress/ posts?since=2017-01-01&until=2017-12-31")

5. В панели предварительного просмотра появятся отфильтрованные посты, начиная с 2017 года.

УПРАЖНЕНИЕ 13.5. Анализ вовлеченности пользователей путем подсчета комментариев и общих ресурсов

В упражнении проанализированы уровни вовлеченности для любой страницы Facebook и показан более удобный способ получения комментариев. В упражне-нии 13.4 были импортированы посты и комментарии для данной страницы Facebook. Следующий шаг — подсчет количества комментариев и публикация каждого полученного поста. В настоящее время, после недавних изменений в API Facebook, а также вследствие ограничений для коннектора Facebook, нельзя загружать число лайков, которые получает каждый пост. В качестве обходного пути в данном упражнении сосредоточимся на подсчете комментариев и общих ресурсов в качестве "показателей вовлеченности" (engagement metrics). Например, количество общих ресурсов (shares) для каждого поста, как показано на рис. 13.8, можно суммировать на протяжении некоторого интервала времени, чтобы отразить вовлечение аудитории.

ПРИМЕЧАНИЕ

Существуют современные методы извлечения числа лайков и эмоциональных реакций (Wow, Sad, Angry, Haha и Love) из постов на общедоступных страницах, но они выходят за рамки этой книги, поскольку для них необходим расширенный диапазон сведений о Graph API и определенные усилия, направленные на создание формул языка М.

1. Откройте рабочую книгу, которая сохранилась после выполнения упраж-нения 13.4, или один из файлов решения C13E04 - Solution.xlsx или C13E04 - Solution.pbix с сайта https://aka.ms/DataPwrBIPivot/downloads.

2. Запустите редактор Power Query.

3. На панели Запросы (Queries) выберите запрос Posts - Base.

4. На панели Примененные шаги (Applied Steps) выберите шаг Source. Отобра-зится следующая формула:

= Facebook.Graph("https://graph.facebook.com/v2.8/microsoftpress/posts")

5. Добавьте знак вопроса с последующим указанием этого параметра в URL в функцию Facebook.Graph:

fields=message,created\_time,comments,story,shares

Обновленная формула M примет следующий вид (копируйте ее в виде отдельной строки):

= Facebook.Graph("https://graph.facebook.com/v2.8/microsoftpress/ posts?fields=message,created\_time,comments,story,shares")

При изменении URL-ссылки Facebook Graph также поменяем версию с v2.8 на v3.0 или на последнюю версию, доступную в настоящее время на Facebook. Модифицированная формула M должна иметь следующий вид (как однострочное выражение):

= Facebook.Graph("https://graph.facebook.com/v3.0/microsoftpress/ posts?fields=message,created\_time,comments,story,shares")

Г") СОВЕТ

Можно изменить URL-ссылку, чтобы опробовать последнюю версию API Graph Facebook или получить дополнительные данные. Например, добавим поле общего доступа к URL-адресу, тогда через пользовательский интерфейс, по умолчанию, можно получать новую информацию, которая не предоставляется Power Query Editor.

Данная URL-ссылка импортирует сообщения с явными полями, которые указаны в параметре fields. При таком подходе необходимо указать все поля, требуемые в отчете, — в данном случае message, creation\_time, comments и story. Указание столбца лайков здесь не поможет, поскольку вы получите пустые результаты.

6. Поскольку вы явно импортировали поля на шаге 5, не нужно расширять их из столбца object\_link, поэтому в разделе Примененные шаги (Applied Steps) удалите шаги Expanded object\_link (Расширенные ссылки объекта) и Expanded connections (Расширенные соединения). При наличии шага Kept First Rows (Сохранить первую строку) также можете удалить и его.

7. На панели Запросы (Queries) выберите запрос Posts.

8. Удалите столбец object\_link.

9. Разверните столбец shares и на панели Развернуть (Expand) сохраните выде-ленным столбец count, установите флажок Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix) и щелкните мышью на кнопке OK.

10. Когда столбец shares преобразуется в shares.count, переименуйте его в Shares Count и измените его тип на Целое число (Whole Number).

Теперь, когда имеется число общих ресурсов на пост (shares) в поле Shares Count, обратимся к подсчету комментариев. Имеется несколько способов для подсчета числа комментариев на пост. Если вы знакомы с Power Pivot в Excel или с моделированием в Power BI, можно создать простую меру DAX для подсчета строк в таблице Comments. Если же предпочитаете считать число строк в Power Query, выполните следующие шаги.

11. При выбранном запросе Posts выберите шаг Source (Источник) на панели Примененные шаги (Applied Steps).

12. Убедитесь, что столбец комментариев содержит либо объект Table, либо нулевое значение в каждой ячейке. Можно создать пользовательский столбец для подсчета числа строк в каждой таблице или вернуть ноль, если значение в комментариях равно нулю, выполняя следующие шаги:

а. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). В окне диалога Вставить шаг (Insert Step) выберите Вставить (Insert) в окне диалога Insert Step (Шаг вставки). Откроется окно диалога Custom Column (Пользовательский столбец).

б. Введите Comments Count в поле Имя нового столбца (New Column Name).

в. В окно Формула пользовательского столбца (Custom Column Formula) введите следующую формулу:

if [comments] = null then 0 else Table.RowCount([comments])

г. Щелкните мышью на кнопке OK для закрытия окна диалога Настраиваемый столбец (Custom Column).

13. Измените тип столбца Comments Count на Целое число (Whole Number).

14. Сохраните эту рабочую книгу или отчет; далее он будет применяться в упражнении 13.6.

Теперь, когда появилась возможность подсчитать число постов и комментариев на пост, можно оценивать активность пользователей на любой общедоступной странице Facebook.

Файлы решения C13E05 - Solution.xlsx и C13E05 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads.

УПРАЖНЕНИЕ 13.6. Сравнение нескольких страниц

В данном упражнении показано, как превратить анализ одной страницы Facebook в многостраничный анализ. Будем выполнять данное упражнение на двух страницах Facebook:

■ Официальная страница Facebook для Microsoft Excel: https://www.facebook.com/microsoftexcel

■ Официальная страница Facebook для Power BI: https://www.facebook.com/microsoftbi

Начнем с отчета, сохраненного в конце упражнения 13.5. Также можно начать с одного из файлов решения, C13E05 - Solution.xlsx или C13E05 - Solution.pbix, которые доступны на сайте https://aka.ms/DataPwrBIPivot/downloads.

1. Откройте файл решения и запустите редактор Power Query.

Первая задача — создать таблицу с двумя именами объектов Facebook. В этом случае, просматривая URL-ссылки официальных страниц Facebook в Excel и Power BI, можно заметить, что названия страниц Facebook — это microsoftexcel и microsoftbi.

2. Создайте пустой запрос и вставьте следующую строку в строку формул, нажмите клавишу <Enter>:

= #table({"PageObjectName", "Page"}, {{"microsoftexcel", "Excel"},

{"microsoftbi", "Power BI"}})

3. Переименуйте запрос Query1 в Pages.

4. На панели Запросы (Queries) выберите запрос Posts - Base. Ваша цель на этом этапе в том, чтобы повторно использовать его шаги для каждой из страниц.

5. Правой кнопкой мыши щелкните на запросе Posts - Base и в контекстном меню выберите команду Дублировать (Duplicate). Переименуйте продублированный запрос на LoadPosts.

6. На вкладке Главная (Home) выберите раскрывающееся меню Управление па-раметрами (Manage Parameters) и затем выберите команду Создать параметр (New Parameter). После открытия окна диалога Параметры (Parameters) до-бавьте следующие изменения:

а. В поле Имя (Name) введите PageName.

б. В поле Текущее значение (Current Value) введите microsoftpress.

в. Щелкните мышью на кнопке OK для закрытия окна диалога.

ПРИМЕЧАНИЕ

В поле Текущее значение (Current Value) значение microsoftpress применяется как временное, чтобы сократить время загрузки предварительного просмотра на шаге 8, поскольку панель предварительного просмотра уже будет загружена результатами страницы Microsoft Press, которые хранятся в кэше Power Query. Вы не будете дли-

тельное время использовать это значение.

7. На панели Запросы (Queries) выберите запрос LoadPosts, а на панели Приме-ненные шаги (Applied Steps) выберите шаг Source.

8. В строке формул замените microsoftpress на " & PageName & ". Исходная формула выглядит так:

= Facebook.Graph("https://graph.facebook.com/v2.8/microsoftpress/ posts?fields=message,created\_time,comments,story,shares")

А вот ее обновленная версия:

= Facebook.Graph(https://graph.facebook.com/v2.8/" & PageName & " / posts?fields=message,created\_time,comments, story,shares")

9. Для формирования функции из запроса LoadPosts на панели Запросы (Queries) правой кнопкой мыши щелкните на LoadPosts и в контекстном меню выбе-рите команду Создать функцию (Create Function). В открывшееся окно диало-га Создать функцию (Create Function) введите FnLoadPostsByPage в поле Имя функции (Function Name) и щелкните на кнопке OK для закрытия окна диа-лога.

Как показано на рис. 13.9, запрос LoadPosts теперь является образцовым. Лю-бые изменения, примененные к LoadPosts, будут распространяться на настраиваемую функцию FnLoadPostsByPage. Не следует изменять какие-либо элементы в коде функции FnLoadPostsByPage.

10. Правой кнопкой мыши щелкните на запросе Pages и в контекстном меню выберите пункт Ссылка (Reference). Переименуйте новый запрос в Posts - All Pages.

11. На панели Запросы (Queries) выберите запрос Posts - All Pages. На вкладке Добавление столбца (Add Column) выберите пункт меню Вызвать настраи-ваемую функцию (Invoke Custom Function).

12. После открытия окна диалога Вызвать настраиваемую функцию (Invoke Custom Function) в раскрывающемся меню Запрос функции (Function Query) выберите пункт FnLoadPostsByPage. Удостоверьтесь, что опция PageObjectName выбрана в раскрывающемся меню PageName, и щелкните мышью на кнопке OK для закрытия окна диалога.

Новый столбец, FnLoadPostsByPage, добавляется следом за столбцом Page, с объектами Table. Каждый объект в строке представляет результаты со стра-ницы Facebook.

Queries [7] yjt 11 = (PageName as any) => let v

J ■ FnLoadPostsByPage [3]

Ll Load Posts

Ц PageName (microsoftpress) fx FnLoadPostsByPage

Enter Parameter

PageName (optional)

1 1

J Other Queries [4]

E Posts - Base Invoke Clear

ffl Posts

\_ „ function (PaqeName as any) as any

П Comments 4 7 J

S Pages

Рис. 13.9. Настраиваемая функция FnLoadPostsByPage, создана на шаге 9.

Теперь можно вызывать эту функцию на нескольких страницах Facebook.

Для того чтобы отредактировать функцию, внесите изменения в запрос Load Posts

13. Разверните столбец FnLoadPostsByPage. На панели Развернуть (Expand) удо-стоверьтесь, что активны все столбцы. Удостоверьтесь, что не установлен флажок Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix), и щелкните на кнопке OK.

Теперь в запросе Posts - All Pages имеются все необходимые сообщения.

14. К сожалению, на шаге 13 утерян столбец column, поэтому измените типы столбцов в запросе Posts - All Pages таким образом: тип столбца Time на Time, тип столбца Date на Date, а тип столбца Hour на Whole Number.

Напомним, что запросы Posts (Сообщения) и Comments (Комментарии) по- прежнему ссылаются на Posts - Base, исходный запрос, который загружал по-сты Microsoft Press. Необходимо изменить их таким образом, чтобы они ссылались на запрос Posts - All Pages.

15. На панели Запросы (Queries) выберите запрос Posts. На панели Примененные шаги (In Applied Steps) выберите шаг Source. Измените строку в строку фор-мул следующим образом:

= #"Posts - All Pages"

16. На панели Запросы (Queries) выберите запрос Comments, на панели Приме-ненные шаги (Applied Steps) выберите шаг Source для обновления строки в строке формул с помощью формулы, показанной на шаге 15.

Теперь оба запроса (Posts и Comments) начинаются с базового запроса Posts - All Pages. В целях безопасности удалите запрос Posts - Base, поскольку он вам не понадобится.

17. Для загрузки обновленных запросов выполните такие действия:

В Excel: на вкладке Главная (Home) редактора Power Query выберите раскры-вающееся меню Закрыть и загрузить (Close & Load) и затем выберите коман-ды Закрыть и загрузить в (Close & Load To). В открывшемся окне диалога Импорт данных (Import Data) выполните команду Создать подключение (Create Connection) только для того, чтобы удостовериться, что только Posts и Comments загружены в модель данных.

В Power BI Desktop: чтобы удостовериться, что LoadPosts и Posts - All Pages не загружены в отчет, правой кнопкой мыши щелкните на каждом из этих за-просов в панели Запросы (Queries) и отмените выбор команды Активизиро-вать загрузку (Enable Load) для каждого из них.

18. Загрузите запрос Pages в модель данных в Excel или в отчет Power BI и создайте связь между столбцом PageObjectName в Pages и столбцом PageObjectName в Posts.

Теперь можно определить, какая страница Facebook получает наибольшее число shares (общедоступных ресурсов) за определенный период времени и какой продукт наиболее активен в Facebook, как показано на рис. 13.10.

Рис. 13.10. В файле решения C13E06 - Solution.pbix реализовано сравнение между Facebook-страницами Excel и Power BI в Power BI

Файлы решения C13E06 - Solution.xlsx и C13E06 - Solution.pbix можно загрузить с сайта https://aka.ms/DataPwrBIPivot/downloads .

Дополнительные сведения об анализе данных Facebook в Excel и Power BI и о том, каким образом визуализировать данные, можно получить, обратившись к моему блогу на сайте https://datachant.com/category/facebook-insights/.

Резюме

В данной главе рассматривается создание социального графика, показано, как импортировать данные из Facebook для получения ответов на большое число личных и деловых вопросов. Также показано, как подключиться к Facebook в Excel и Power BI для загрузки нескольких конечных точек (например, вашего профиля и любой страницы Facebook) и ребер (таких, как лайки страниц, друзья, записи и комментарии).

Рассмотрены несколько практических приемов использования соединителя Facebook и анализа как своего профиля, так и любой общедоступной страницы. В упражнении 13.1 выполнен импорт понравившихся вам страниц Facebook и их сортировка по времени просмотра.

В упражнении 13.2 показано, как пройти через социальный график и найти друзей первой и второй степеней, которые подключились к Facebook через соединитель Facebook. И хотя данный метод имеет ряд ограничений и не может загрузить весь список ваших друзей, он является хорошим инструментом для "шпионажа" в сети, позволяя оценить число пользователей вашей сети, воспринявших Power Query и Power BI. В упражнении 13.3 рассмотрены общие страницы, которые понравились вашим друзьям (опять же, для ограниченного списка друзей, использующих соединитель Power Query на Facebook).

В упражнении 13.4 рассмотрен основной метод извлечения постов и комментариев, а в упражнении 13.5 показано, как улучшить запросы и импортировать количество shares (общедоступных ресурсов) на пост. Наконец, в упражнении 13.6 показано, как расширить запросы для импорта нескольких страниц Facebook с помощью пользовательской функции. Теперь, после приобретения знаний об анализе страниц, постов и комментариев в Facebook, можно применять передовые методы анализа текста из глав 11 и 12 и создать мощный инструмент конкурентного анализа для любого бренда, представленного в Facebook.

В следующей, и последней, главе будем применять на практике полученные навыки по обработке данных в серии задач, охватывающих основные темы, представленные в книге. Я полагаю, что вы справитесь.

Глава 14

Заключительный проект: объединяем все вместе

Чтобы написать это, потребовалось три месяца; зачать это — три минуты; собирать данные — всю свою жизнь.

- Ф. Скотт ФицДжеральД

В ЭТОЙ ГЛАВЕ ВЫ...

■ Займете должность главного специалиста по данным вымышленной компании Wide World Importers.

■ Приступите к обработке сложных наборов данных и примените свои навыки для подготовки необработанных данных для анализа.

■ Научитесь объединять последовательности операций отмены сведения и сведения столбцов в сложный сценарий реальной жизни с несколькими файлами обобщенных данных.

■ Рассмотрите применение различных преобразований слияния для сравнения двух таблиц.

Примите поздравления. Вы приступаете к последней главе. Пришло время ответить на вызов и подтвердить наличие навыков по обработке данных, проверяя свои возможности в заключительном проекте.

Вы прошли долгий путь. В предыдущих главах книги было показано, как выпол-нять очистку необработанных наборов данных, объединять несколько таблиц, исправлять несоответствующие имена столбцов, реализовывать разворот и сведение данных, а также решать многие другие проблемы с данными. Пришло время проверить все эти знания на практике и применить новые возможности для обработки данных в крупномасштабной задаче.

УПРАЖНЕНИЕ 14.1.

Спасите данные компании Wide World Importers

Представьте, что вы занимаете должность главного специалиста по данным в Wide World Importers. В минувшие выходные компания подверглась массированной кибератаке, направленной на хранилище данных компании. Группа специалистов пытается оценить риски для данных о доходах, и вас попросили восстановить новый

сводный отчет о доходах компании на основе старых данных по экспорту, которые не затронуты кибератакой. Ваш предшественник вел отчеты о доходах компании в трех файлах Excel за 2015-2017 годы. В каждом из них содержалось 12 рабочих книг — по одной на каждый месяц года. Кроме того, выручка 2018 года недавно была экспортирована в новом сложном формате.

Можно ли импортировать все данные из каждого из 36 рабочих листов и объединить их с данными 2018 года для создания единой таблицы, аналогичной той, что показана на рис. 14.1?

Чтобы начать упражнение, загрузите zip-файл C14E01.zip и результирующую таблицу в файле C14E01 - Goal.xlsx. Оба файла доступны на сайте https://aka.ms/ DataPwrBIPivot/downloads. Извлеките четыре файла из ZIP-архива и поместите их в папку C:\Data\C14\C14E01.

Изучите каждую из рабочих книг 2015.xlsx, 2016.xlsx и 2017.xlsx, а также их рабочие листы. Каждый рабочий лист содержит сводную таблицу формата 2\*3 с категориями и городами клиентов в строках и цветами продуктов, категориями поставщиков и наименованиями поставщиков в столбцах.

Данные о доходах в 2018 году хранятся в файле со значениями, разделенными запятыми (CSV), со столбцами Attribute (Атрибут) и Value (Значение), как показано в нижней части, слева на рис. 14.1. Ваша цель состоит в объединении всех данных вместе для формирования целостного отчета доходов компании Wide World Importers, как показано в файле C14E01 - Goal.xlsx.

ПРИМЕЧАНИЕ

Вам предлагается приостановить чтение на данный момент. Попробуйте решить эту проблему самостоятельно. Потратьте пару часов, пытаясь это сделать. Если вам потребуется помощь, можно прочитать подсказки в следующем разделе.

Подсказки

Не знаете, как поступить? Вы можете разбить задачу на подзадачи:

1. Импортируйте папку и сохраните книги Excel.

2. Создайте новый столбец с 36 строками объектов Table (состоящими из 36 листов из трех книг с 12 таблицами доходов за каждый месяц 2015-2017 годов).

3. Для каждого листа пропустите первую строку и удалите столбец и строку общего итога (Grand Total).

4. Примените функцию FnUnpivotSummarizedTable для каждого рабочего листа. (Обра-титесь к главе 7.)

5. Включите файл 2018.csv после того, как будете выполнять операцию сведения столбца, которая рассмотрена в главе 7.

6. Присоедините доходы 2018 года к данным, для которых отменено сведение столбца. Обратите внимание, что имеются несоответствия между именами столбцов. Можно возвратиться к главе 3 для уточнения деталей процесса нормализации для имен столбцов.

ПРИМЕЧАНИЕ

Вам предлагается снова приостановить чтение главы на данный момент. Попробуйте решить эту проблему самостоятельно. Потратьте несколько часов, пытаясь это выполнить. Если нужна помощь, прочитайте главу 7 еще раз. Если потребуется дополнительная помощь, можно выполнить упражнение 14.1 шаг за шагом.

Часть 1. Приступим к решению

Для преобразования сведенных данных из 2015-2017 годов и присоединения их к многострочным записям за 2018 можно выполнить такие шаги в данном упражнении:

1. Загрузите zip-файл C14E01.zip с сайта https://aka.ms/DataPwrBIPivot/downloads и извлеките четыре файла в папку C:\Data\C14\C14E01.

2. Откройте пустую рабочую книгу или отчет Power BI:

В Excel: на вкладке Данные (Data) выполните команды Получить данные (Get

Data) Из файла (From File) Из папки (From Folder).

В Power BI Desktop: выполните команды Получить данные (Get Data) Файл

(File) Папка (Folder).

3. Выберите путь для папки C:\Data\C14\C14E01 и щелкните мышью на кнопке OK. В последующем окне диалога щелкните на кнопке Изменить (Edit). Откро-ется окно редактора Power Query.

4. Для выполнения фильтрации файла 2018.csv, формат которого отличается от других файлов (он обрабатывается позднее), выберите элемент управления фильтром для столбца Name и выберите файл 2018.csv. Затем щелкните на кнопке OK для закрытия панели Фильтр (Filter).

5. Переименуйте запрос из книги C14E01 на Revenues.

6. Щелкните на значке Объединить файлы (Combine Files) в заголовке столбца Content.

7. После открытия окна диалога Объединить файлы (Combine Files) правой кнопкой мыши щелкните на команде Sample File Parameter1 (Параметр1 об-разца файла) и выберите команду Изменить (Edit).

8. После открытия окна редактора Power Query и выбора вкладки Revenues уда-лите последние три столбца.

9. Удалите расширение xlsx из значений Source.Name для извлечения года и переименуйте столбец Year. (Имеется несколько способов удалить расширение файла. Если вы не уверены, как это сделать, вернитесь к главе 3.)

10. Переименуйте столбец с Name на Month.

В столбце Data получим объекты Table, каждый со сводными таблицами. Пе-ред применением последовательности отмены сведения столбца следует удалить первую строку, которая не содержит каких-либо ценных данных и не будет работать должным образом, если применить к ней последовательность отмены сведения столбца. Также следует удалить общий итог (Grand Totals) в последнем столбце и строке. На следующих шагах показано, как это сделать.

11. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). Откроется окно диалога Настраиваемый столбец (Custom Column). Внесите следующие изменения:

а. Введите Skip First Row в поле Имя нового столбца (New Column Name).

б. Введите следующую формулу в поле Формула пользовательского столбца (Custom Column Formula) и затем щелкните на кнопке OK:

= Table.Skip([Data], 1)

Новый столбец возвращает объект Table в столбце Data после столбца и пропуска первой строки.

в. Удалите столбец Data.

12. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). Откроется окно диалога Настраиваемый столбец (Custom Column). Внесите следующие изменения:

а. В поле Имя нового столбца (New Column Name) введите значение No Grand Total Column.

б. В окне Формула пользовательского столбца (Custom Column Formula) введите следующую формулу и щелкните мышью на кнопке OK:

= Table.RemoveColumns(

[Skip First Row], {

List.Last(Table.ColumnNames([Skip First Row]))

}

)

Эта формула удаляет последний столбец. С помощью функции Table. ColumnNames вы получаете имена столбцов каждого объекта таблицы в столбце Data, а с помощью List.Last — имя последнего столбца.

в. Удалите столбец Skip First Row.

13. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). Откроется окно диалога Настраиваемый столбец (Custom Column). Внесите следующие изменения:

а. В поле Имя нового столбца (New Column Name) введите значение Summarized Table.

б. Введите следующую формулу в окно Формула пользовательского столбца (Custom Column Formula) и затем щелкните мышью на кнопке OK:

в. = Table.RemoveLastN([No Grand Total Column], 1)

г. Удалите столбец No Grand Total Column.

ПРИМЕЧАНИЕ

Альтернативный способ очистки обобщенных таблиц вместо выполнения шагов 11-13 состоит в использовании одной функции. Можно создать пустой запрос с помощью следующего кода:

(inputTable) =>

let

SkipFirstRow = Table.Skip(inputTable, 1),

NoGrandTotalColumn = Table.RemoveColumns(

SkipFirstRow, {

List.Last(Table.ColumnNames(SkipFirstRow))

}

),

Result = Table.RemoveLastN(NoGrandTotalColumn, 1)

in

Result

Назовите запрос FnCleanSummarizedTable и затем вызовите его в запросе Revenues, выбирая команду Вызвать настраиваемый столбец (Invoke Custom Column) на вкладке Добавление столбца (Add Column) и применяя ее к столбцу Data.

Часть 2. Вызов функции отмены сведения столбца

В данной части решения для упражнения 14.1 применим пользовательскую функцию FnUnpivotSummarizedTable. Напомним, что в упражнении 7.3 эта функция была создана для выполнения отмены свертывания столбцов для таблиц суммирования. В упражнении 14.1 теперь можно выполнить операцию отмены свертывания столбцов для таблицы суммирования Wide World Importers (за 2015-2017 годы). Следуйте описанным далее шагам для продолжения упражнения 14.1.

14. Для копирования функции FnUnpivotSummarizedTable из упражнения 7.3 выполните такие шаги:

а. Сохраните текущий отчет и держите его открытым.

б. Откройте рабочую книгу C07E03 - Solution.xlsx на сайте https://aka.ms/ DataPwrBIPivot/downloads.

в. На вкладке Данные (Data) выберите команду Запросы и подключения (Queries & Connections).

г. На панели Запросы и подключения (Queries & Connections) правой кнопкой мыши щелкните на опции FnUnpivotSummarizedTable и выберите команду Копировать (Copy).

д. Вернитесь к текущей рабочей книге из упражнения 14.1 или отчету и запустите редактор Power Query.

е. Правой кнопкой мыши щелкните где-либо в фоновой области панели За-просы (Queries) и выберите команду Вставить (Paste) для копирования функции FnUnpivotSummarizedTable.

15. На панели Запросы (Queries) выберите запрос Revenues.

16. На вкладке Добавление столбца (Add Column) выберите команду Настраи-ваемый столбец (Custom Column). Откроется окно диалога Настраиваемый столбец (Custom Column). Внесите следующие изменения:

а. Введите название Unpivoted в поле Имя нового столбца (New Column Name).

Рис. 14.2. Для выполнения операции отмены свертывания столбца для таблицы суммирования вызовите функцию FnUnpivotSummarizedTable в окне диалога Настраиваемый столбец

б. Как показано на рис. 14.2, введите следующую формулу в окно Формула пользовательского столбца (Custom Column Formula) и затем щелкните мышью на кнопке OK:

= FnUnpivotSummarizedTable([Summarized Table], {"Customer Category", "City"}, {"Color", "Supplier Category", "Supplier Name"}, "Revenue")

в. Удалите столбец Summarized Table.

17. Разверните столбец Unpivoted. На панели Развернуть (Expand) отмените уста-новку флажка Использовать имя исходного столбца в качестве префикса (Use Original Column Name As Prefix) и щелкните мышью на кнопке OK.

В результате в панели предварительного просмотра можно увидеть, что для всех рабочих листов операция отмены свертывания столбцов выполнена корректно. Теперь можно обработать доходы за 2018.

Часть 3. Последовательность свертывания столбцов для доходов 2018 года

Напомним, что на шаге 4 этого упражнения вы отфильтровали файл 2018.csv. Этот файл содержит разделенные запятыми значения пар атрибут/значение, и можно предположить, что каждая многострочная запись о доходах в файле 2018.csv начинается с атрибута Date. В этой части упражнения необходимо импортировать ее и применить последовательность свертывания столбцов, изученную в главе 7. Для продолжения упражнения 14.1 выполните следующие действия:

18. Импортируйте файл 2018.csv в окно редактора Power Query, выбирая команду Создать источник (New Source) на вкладке Главная (Home). Переименуйте новый запрос в 2018 Revenues.

19. На вкладке Преобразование (Transform) выберите команду Использовать первую строку в качестве заголовка (Use First Row as Headers).

20. На вкладке Добавление столбца (Add Column) выберите команду Столбец ин-декса (Index Column).

21. На вкладке Добавление столбца (Add Column) выберите команду Условный столбец (Conditional Column). После открытия окна диалога Добавление ус-ловного столбца (Add Conditional Column) внесите следующие изменения:

а. Введите Row ID в поле Имя нового столбца (New Column Name).

б. Установите значение Имя столбца (Column Name) как Attribute.

в. В качестве оператора выберите равенство.

г. Введите Date в поле Значение (Value).

д. В списке Выход (Output) выберите пункт Выберите столбец (Select a Column) и выберите столбец Индекс (Index).

е. Введите null в поле В противном случае (Otherwise).

ж. Щелкните мышью на кнопке OK.

22. Выберите столбец Row ID и на вкладке Преобразование (Transform) выполните команды Заполнить (Fill) Вниз (Down). Удалите столбец Индекс (Index).

23. Выберите столбец Attribute (Атрибут) и на вкладке Преобразование (Trans-form) выберите команду Столбец сведения (Pivot Column). После открытия окна диалога Столбец сведения (Pivot Column) внесите следующие изменения:

а. В столбце Столбец значений (Values Column) выберите Значение (Value).

б. Раскройте раздел Расширенные параметры (Advanced Options).

в. В списке Функция агрегированного значения (Aggregate Value Function) выберите значение Не агрегировать (Don't Aggregate).

г. Щелкните мышью на кнопке OK.

24. Удалите столбец Row ID.

Теперь после получения завершенной последовательности сведения столбца для доходов 2018 года можно объединить результаты.

Часть 4. Объединение доходов за 2018 и 2015-2017 годы

Шаги, которые необходимо предпринять в этой части упражнения, обсуждаются в главе 4. Теперь имеются две таблицы с несовпадающими именами столбцов. Обратите внимание, что запрос Revenues включает столбцы Year (Год) и Month (Месяц), а запрос 2018 Revenues включает столбец Date (Дата).

Выполните следующие действия для продолжения упражнения 14.1:

25. Выберите запрос Revenues. Выделите первые два столбца, Year и Month, и вы-берите команду Объединить столбцы (Merge Columns) на вкладке Преобразо-вание (Transform).

26. После открытия окна диалога Объединить столбцы (Merge Columns) выберите разделитель пробел, введите Date в поле Имя нового столбца (New Column Name) и щелкните мышью на кнопке OK.

27. На вкладке Главная (Home) выберите команду Объединить запросы (Append Queries). После открытия окна диалога Объединить (Append) выберите объ-единение с таблицей 2018 Revenues и щелкните мышью на кнопке OK.

ПРИМЕЧАНИЕ

На шаге 16 было объявлено, что столбец names применяется для отмены свертывания столбца доходов для 2015-2017 годов. При использовании других имен для аргументов, кроме полей Row или Column, в функции FnUnpivotSummarizedTable, появятся несоответствия в названиях столбцов на этом этапе. Для устранения этого задавайте те же названия столбцов, что и на шаге 16, или переименуйте 2018 Revenues для соответствия запросу Revenues перед добавлением.

28. Измените тип столбца Revenue на Денежный (Currency) или Десятичное число (Decimal Number) и измените тип столбца Date на Дата (Date).

29. Наконец, загрузите запрос Revenues в отчет и отключите загрузку для запроса 2018 Revenues:

В Excel: на вкладке Данные (Data) выберите команду Запросы и подключения (Queries & Connections). Правой кнопкой мыши щелкните на запросе 2018 Revenues и в контекстом меню выберите команду Загрузить в (Load To). В открывшемся окне диалога Импорт данных (Import Data) установите флажок Только создать подключение (Create Connection) и щелкните мышью на кнопке OK.

В Power BI Desktop: в окне редактора Power Query правой кнопкой мыши щелкните на запросе 2018 Revenues на панели Запросы (Queries) и отмените установку флажка Активизировать загрузку (Enable Load).

Файлы решения C14E01 - Solution.xlsx и C14E01 - Solution.pbix доступны на сайте https://aka.ms/DataPwrBIPivot/downloads.

УПРАЖНЕНИЕ 14.2. Сравнение таблиц

и отслеживание действий хакеров

Благодаря успешному выполнению упражнения 14.1 создан отчет о доходах, который помог руководству вашей компании Wide World Importers принять обоснованные решения для поддержки бизнеса после кибератаки. Согласно исследованию группы по кибербезопасности, злоумышленники подделали значения доходов в хранилище данных.

Ваша задача состоит в сравнении суммарных доходов, полученных в упражне-нии 14.1, с измененными данными в файле C14E02 - Compromied.xlsx, который доступен на сайте https://aka.ms/DataPwrBIPivot/downloads для определения, какие записи подделаны хакерами и скомпрометированы. Сможете ли вы найти запи-си, которые содержат измененные доходы? Сможете ли найти сообщение хакера в данных?

Попытайтесь выполнить эту задачу, создавая запрос, который сравнивает файлы C14E01 - Solution.xlsx и C14E02 - Compromised.xlsx. Хотя найти различия в таблицах можно с помощью сводных таблиц или мер DAX, сконцентрируйтесь на поиске решения с помощью редактора Power Query.

Подсказка

Если вы не знаете, как решить эту проблему, вот несколько подсказок, которые помогут вам начать работу:

1. Хакеры внесли изменения в столбец Revenue (Доход). Можно объединить две таблицы для нахождения несоответствующих доходов. Используйте все столбцы, кроме столбца Revenue, для слияния, а затем разверните столбец Revenue из второй таблицы. Затем примените фильтр к двум столбцам Revenue для нахождения различий.

2. Чтобы найти новые строки, добавленные хакерами, можно применить операцию Объединить столбцы (Merge Columns) с антисоединением справа между суммированной таблицей из упражнения 14.1 и скомпрометированной таблицей. Таким образом, будут обнаружены только те строки, которые имеются в скомпрометированной таблице, на этапе развертывания, следующем после объединения.

УПРАЖНЕНИЕ 14.2. Решение

Для того чтобы найти несовпадающие доходы, необходимо объединить две таблицы, используя все столбцы, кроме Revenue. Если вы не можете выяснить это сами, то выполните действия, описанные в этом упражнении.

1. Импортируйте таблицу Revenues из файла C14E01 - Solution.xlsx в редакторе Power Query из Excel или Power BI Desktop. Переименуйте запрос в Correct.

2. Импортируйте таблицу Revenues из файла C14E02 - Compromised.xlsx в окно редактора Power Query в Excel или Power BI Desktop.

3. На панели Запросы (Queries) выберите запрос Correct. Затем на вкладке Глав-ная (Home) выберите раскрывающееся меню Объединить запросы (Merge Queries) и после этого выберите команду Объединить запросы в новый (Merge Queries as New).

4. После открытия окна диалога Слияние (Merge) внесите следующие изменения:

а. В таблицу Correct, удерживая нажатой клавишу <Ctrl>, выбирайте столбцы Date, Color, Supplier Category, Supplier Name, Custom Category и City. Обяза-тельно сохраняйте порядок следования столбцов.

б. В раскрывающемся меню, под таблицей Correct, выберите запрос Compro-mised. В таблице Compromised, удерживая нажатой клавишу <Ctrl>, выби-райте поля Date, Color, Supplier Category, Supplier Name, Custom Category и City. Обязательно сохраняйте порядок следования столбцов.

в. В раскрывающемся меню Тип соединения (Join Kind) выберите опцию Внутреннее (только совпадающие строки) ((Inner (Only Matching Rows)), как показано на рис. 14.3, и щелкните мышью на кнопке OK для закрытия окна диалога Слияние (Merge).

5. Переименуйте новый запрос на Compromised Rows и разверните столбец Compromised. В панели Expand отмените выбор всех столбцов, кроме Revenue. С помощью исходного названия столбца в виде префикса переименуйте новый столбец в Compromised.Revenue.

Чтобы отфильтровать несоответствующие строки дохода, необходимо применить условие фильтра, которое недоступно в пользовательском интерфейсе. Можно создать временное состояние фильтра через пользовательский интерфейс, а затем изменить его в строке формул в соответствии со своими потребностями.

6. Щелкните мышью на элементе фильтра в заголовке столбца Revenue и выберите команды Числовые фильтры (Number Filters) Не равно (Does Not Equal). Введите 12345 в текстовое поле, следующее за раскрывающимся меню Не равно (Does Not Equal). В строке формул отобразится такое выражение:

= Table.SelectRows(#"Expanded Compromised", each [Revenue] <> 12345)

7. Для фильтрации строк, где значения столбца Revenue не равны значениям Compromised.Revenue, обновите формулу следующим образом:

= Table.SelectRows(#”Expanded Compromised”, each [Revenue] <> [Compromised.Revenue])

Теперь будут видны строки с различными значениями в Revenue и Compro- mised.Revenue.

X

Слияние

Выберите таблицу и совпадающие столбцы для создания объединенной таблицы.

Correct a

Date i Color 2 Supplier Category з Supplier Name 4 Customer Category s City 6 Revenue

01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Beaver Bay 4471,2

01.01.2015 Black Clothing Supplier Fabrikam, Inc Computer Store Caruthers 616,4

01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Conesus Lake 2235,6

01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Darling 201,25

01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Flowella 1490,4

Compromised a

Date i Color 2 Supplier Category з Supplier Name 4 Customer Category s City 6 Revenue

01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Beaver Bay 4471,2

01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Caruthers 616,4

01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Conesus Lake 2235,6

01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Darling 201,25

01.01.2015 Black Clothing Supplier Fabrikam, Inc. Computer Store Flowella 1490,4

Тип соединения

Внутреннее (только совпадающие строки)

\/ Выбор согласовал 96035 из первых строк (96035).

Рис. 14.3. Окно диалога Слияние (Merge) может использоваться для сравнения двух таблиц

и нахождения несоответствующих значений

Обнаружение следов деятельности хакеров

во "взломанной" таблице

На следующих шагах вы примените правильное антисоединение, что позволит обнаружить строки в скомпрометированной таблице, которые были добавлены хакерами и не найдены в исходной таблице. Выполните следующие шаги для продолжения упражнения 14.2:

8. На панели Запросы (Queries) выберите запрос Correct. Затем на вкладке Глав-ная (Home) выберите раскрывающееся меню Объединить запросы (Merge Queries) и после этого выберите команду Объединить запросы в новый (Merge Queries as New).

9. После открытия окна диалога Слияние (Merge) внесите следующие изменения:

а. В таблице Correct, удерживая нажатой клавишу <Ctrl>, выбирайте столбцы Date, Color, Supplier Category, Supplier Name, Custom Category и City. Обяза-тельно сохраняйте порядок следования столбцов.

б. В раскрывающемся меню под таблицей Correct выберите запрос Compro-mised. В таблице Compromised, удерживая нажатой клавишу <Ctrl>, выби-райте столбцы Date, Color, Supplier Category, Supplier Name, Custom Category и City. Обязательно сохраняйте порядок следования столбцов.

в. В раскрывающемся меню Тип соединения (Join Kind) выберите пункт Антисоединение справа (только строки во второй т таблице) (Right Anti (Rows Only in Second)) и щелкните мышью на кнопке OK для закрытия окна диалога Слияние (Merge).

10. Переименуйте новый запрос в Hacker's Instructions.

11. Выберите столбец Compromised и удалите все другие столбцы.

12. Извлеките все поля в столбце Compromised.

13. Найдите сообщение хакера.

14. Загрузите запрос Hacker's Instructions в вашу рабочую книгу Excel или Power BI Report.

Файлы решения C14E02 - Solution.xlsx и C14E02 - Solution.pbix доступны на сайте https://aka.ms/DataPwrBIPivot/downloads.

См. также

Дополнительные запросы по сравнению таблиц найдите в следующих постах блога: https://datachant.com/downloads/compare-tables-excel-power-bi/ https://datachant.com/2017/08/10/compare-tables-excel-power-bi-part-2/.

Резюме

Теперь, завершив финальный проект, вы официально стали опытным специалистом по обработке данных. С помощью редактора Power Query Editor вы сможете существенно влиять на ваш бизнес и даже за его пределами. В этом проекте вы задействовали ряд навыков, включая добавление нескольких рабочих книг и рабочих таблиц, выполнение расширенных операций по свертыванию и отмене свертывания столбцов, применение пользовательских функций и сравнение наборов данных с помощью операции слияния.

Эта книга рекомендуется в качестве справочного руководства. Сохраните образцы работ для дальнейшего использования.

Вы найдете их полезными в дальнейшем, когда будете решать новые проблемы. И хотя ваши наборы данных могут полностью отличаться от данных, приведенных в этой книге, вы постепенно поймете, как повторно использовать методы и расширить имеющиеся решения применительно к вашим конкретным проблемам с данными.

Изучение Power Query — это непрерывный процесс. Для того чтобы совершенствовать свои возможности по обработке данных, рекомендуется продолжать изучение неупорядоченных наборов данных в Power Query и ставить перед собой новые задачи. Сейчас можно вернуться к главе 9 и просмотреть, какие компоненты языка М или синтаксиса вы понимаете лучше и как можно использовать этот язык для расширения запросов на основе пользовательского интерфейса.

Со времени написания книги и до настоящего момента вы, скорее всего, обнаружили новые функции и возможности Power Query в Excel и Power BI, а также новые случаи использования, когда Power Query может сэкономить сотни часов. Настало время узнать, как можно воспользоваться новыми возможностями для обработки данных в таких продуктах, как Microsoft Analysis Services, Microsoft Flows и Power Apps, которые теперь также включают Power Query. Ваше путешествие только началось. Надеюсь, оно вам понравится.