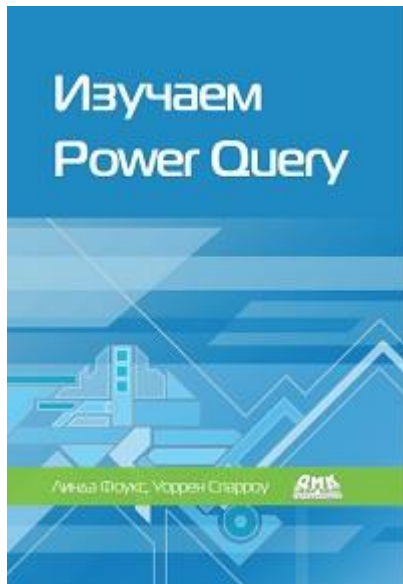


Линда Фоукс, Уоррен Спарроу. Изучаем Power Query

Power Query – инструмент Excel, позволяющий подключаться к источникам данных, подготавливать и обрабатывать информацию для ее дальнейшего анализа. В этой книге вы найдете все, что нужно для работы с Power Query. Отдельная часть книги посвящена языку формул M, который обеспечивает новый подход к агрегированию данных. В завершение вы научитесь строить дашборды и многомерные отчеты.

Линда Фоукс, Уоррен Спарроу. Изучаем Power Query. – М.: ДМК Пресс, 2021. – 378 с.



Файлы с примерами можно загрузить на [сайте](#) издательства.

Глава 1. Установка и настройка

Power Query и Power Pivot интегрированы во все современные версии Excel. По умолчанию Power Pivot не активирован. Чтобы активировать надстройку пройдите *Файл* → *Параметры* → *Надстройки*. В поле *Управление* выберите *Надстройки COM*. В открывшемся окне поставьте галочку напротив *Microsoft Power Pivot for Excel*.

Power BI можно бесплатно загрузить с Microsoft Store: <https://aka.ms/pbidesktopstore>.

Если у вас установлен Office 365, 2019 или 2016 вы найдете Power Query на вкладке *Данные* → *Получить и преобразовать данные*.

Чтобы получить доступ к Power Query из Power BI, запустите программу Power BI Desktop. На левой панели окна загрузки кликните *Получить данные*. В окне *Получить данные* выберите источник. Например, можно выбрать *Другое* → *Интернет*. Нажмите кнопку *Подключить*. В окне *Из Интернета* введите адрес.

Если же вы хотите получить доступ к Power Query находясь внутри Power BI, пройдите *Главная* → *Данные* → *Получить данные*.

Глава 2. Основы Power Pivot, его недостатки и управление данными

Кратко описано создание сводных таблиц, срезов, умных таблиц и модели данных. Показано, как в модели данных связать несколько таблиц, добавить вычисляемое поле и меру. Упомянут контекст строки и контекст фильтра. См. по теме: [Билл Джелен, Майкл Александер. Сводные таблицы в Microsoft Excel 2013](#), [Зак Барресс и Кевин Джонс. Таблицы Excel: Полное руководство для создания, использования и автоматизации списков и таблиц](#), [Альберто Феррари, Марко Руссо. Анализ данных при помощи Microsoft Power BI и Power Pivot для Excel](#).

Моделирование данных – процесс объединения таблиц в единую базу данных с определением правил хранения, обработки и обновления данных. И хотя наиболее популярным видом остается реляционная модель данных, существуют и другие, например иерархическая или сетевая модель.

Сводные таблицы, созданные из Power Pivot, обладают рядом недостатков. Во-первых, невозможно сгруппировать даты по кварталам или годам. Вторая проблема связана с тем, что

если в обычной сводной таблице вы всегда можете дважды щелкнуть по ячейке и увидеть предпросмотр по всем строкам, составляющим содержимое ячейки, то в сводной таблице, созданной в Power Pivot, предпросмотр ограничен 1000 записей. В сводной на основе модели данных невозможно создавать вычисляемые поля и элементы, хотя в обычных сводных таблицах такая возможность присутствует.

Глава 3. Введение в интерфейс Power Query

Чтобы изменить тип данных столбца щелкните по иконке рядом с названием столбца. В контекстном меню выберите тип данных для столбца.

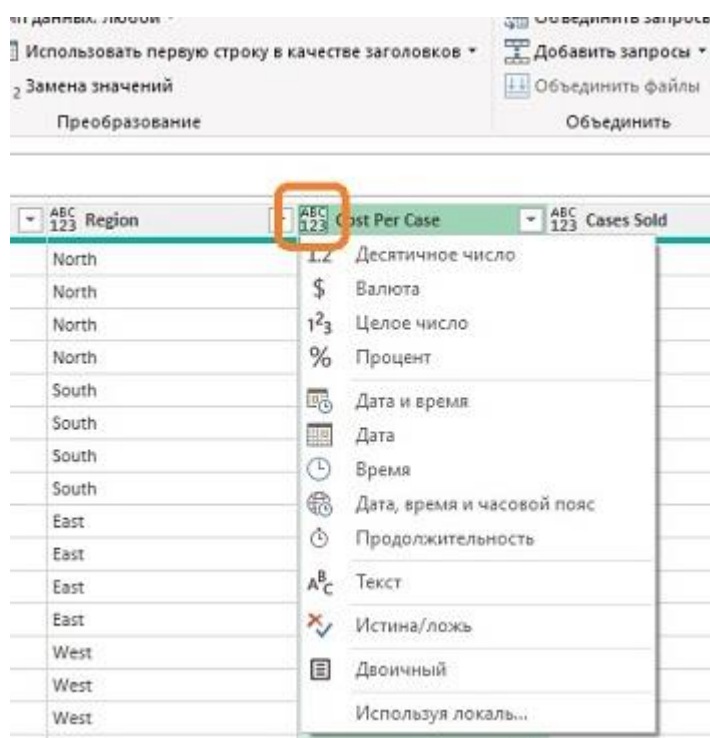


Рис. 1. Контекстное меню выбора типа данных

Если нужно удалить множество шагов на панели ПРИМЕНЕННЫЕ ШАГИ, кликните правой кнопкой мыши на шаг, начиная с которого хотите всё удалить и выберите *Удалить до конца*:

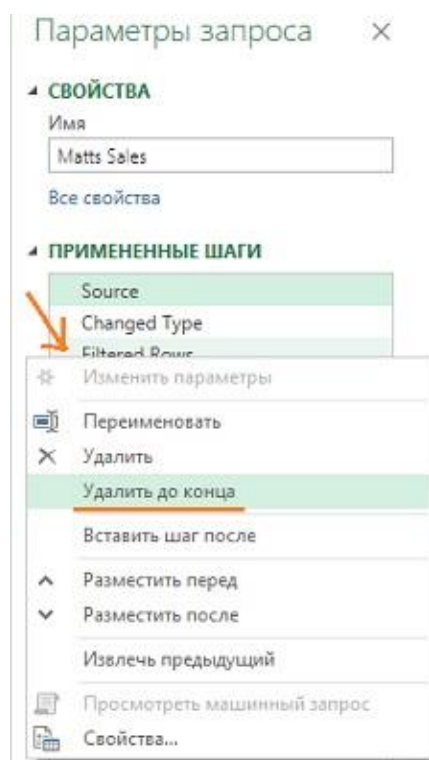


Рис. 2. Удаление нескольких шагов запроса одновременно

Очень полезно документировать шаги запроса. Чтобы сделать это кликните правой кнопкой мыши на шаг запроса в панели ПРИМЕНЕННЫЕ ШАГИ, выберите *Свойства*. Введите текст в поле *Описание*. Также важно, чтобы можно было легко понять предназначение того или иного шага при беглом взгляде на список шагов. Давайте шагам говорящие названия.

При работе в редакторе Power Query часто удобно использовать моноширинный шрифт. Чтобы включить его пройдите *Просмотр* → *Предварительный просмотр данных* → *Моноширинный*.

Профилирование данных на вкладке *Просмотр* редактора Power Query позволяет получить статистику о столбце, увидеть и исправить ошибки.

Глава 5. Преобразование данных посредством Power Query

В Power Query существует возможность установки опции автоматического обновления. Откройте файл Excel, сделайте активным запрос. Пройдите *Данные* → *Запросы и подключения*. На кнопке *Обновить все* нажмите стрелку вниз. Выберите *Свойства подключения*.

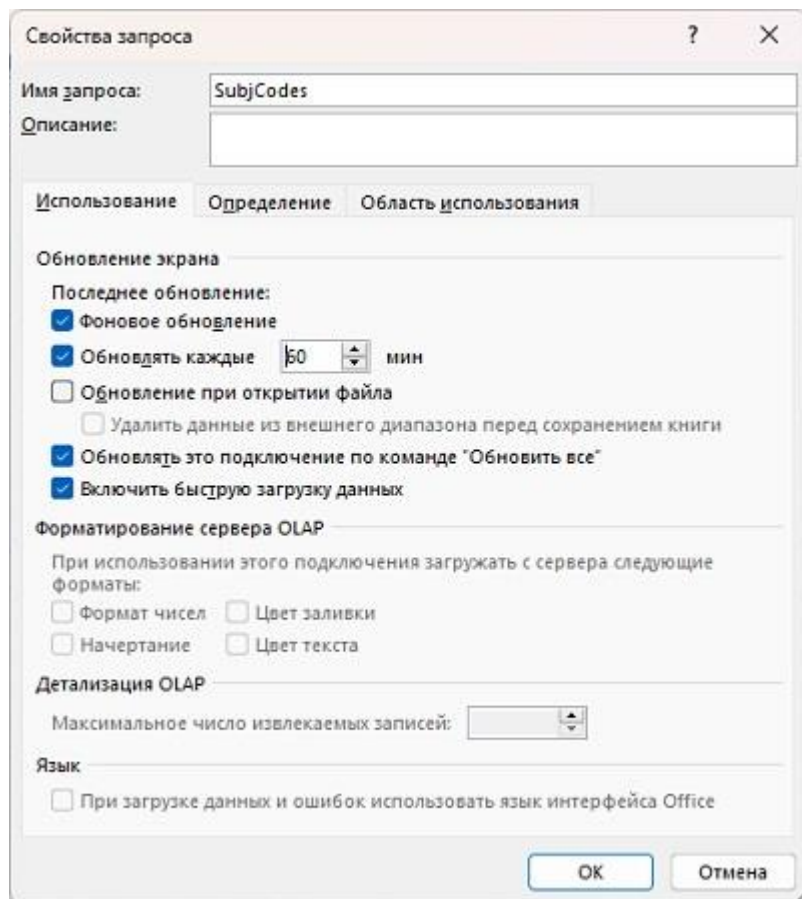


Рис. 3. Свойства обновления

В окне *Свойства запроса* на вкладке *Использование* обратите внимание на раздел *Обновление экрана*. Убедитесь, что флажок *Фоновое обновление* установлен. Выберите требуемую частоту обновлений в минутах. У вас есть возможность обновлять данные при открытии файла и включать или не включать запрос в число обновляемых по кнопке *Обновить все*.

Глава 7. Автоматизация отчетов в Power Query

Обычно, чтобы увидеть изменения, произошедшие в источнике, вам необходимо заново импортировать данные в Power BI. Однако существует и режим обращения к данным в источнике напрямую. При использовании Power BI Desktop (но не Excel) есть возможность управлять режимами хранения информации.

В Power BI настройку режима хранения можно установить отдельно для каждой таблицы модели данных. На левой панели Power BI Desktop кликните по иконке модели данных (1), выберите таблицу, для которой хотите изменить режим хранения (2), и требуемый режим (3). Доступно три варианта хранения данных: *Импорт*, *DirectQuery* и *Двойной*:

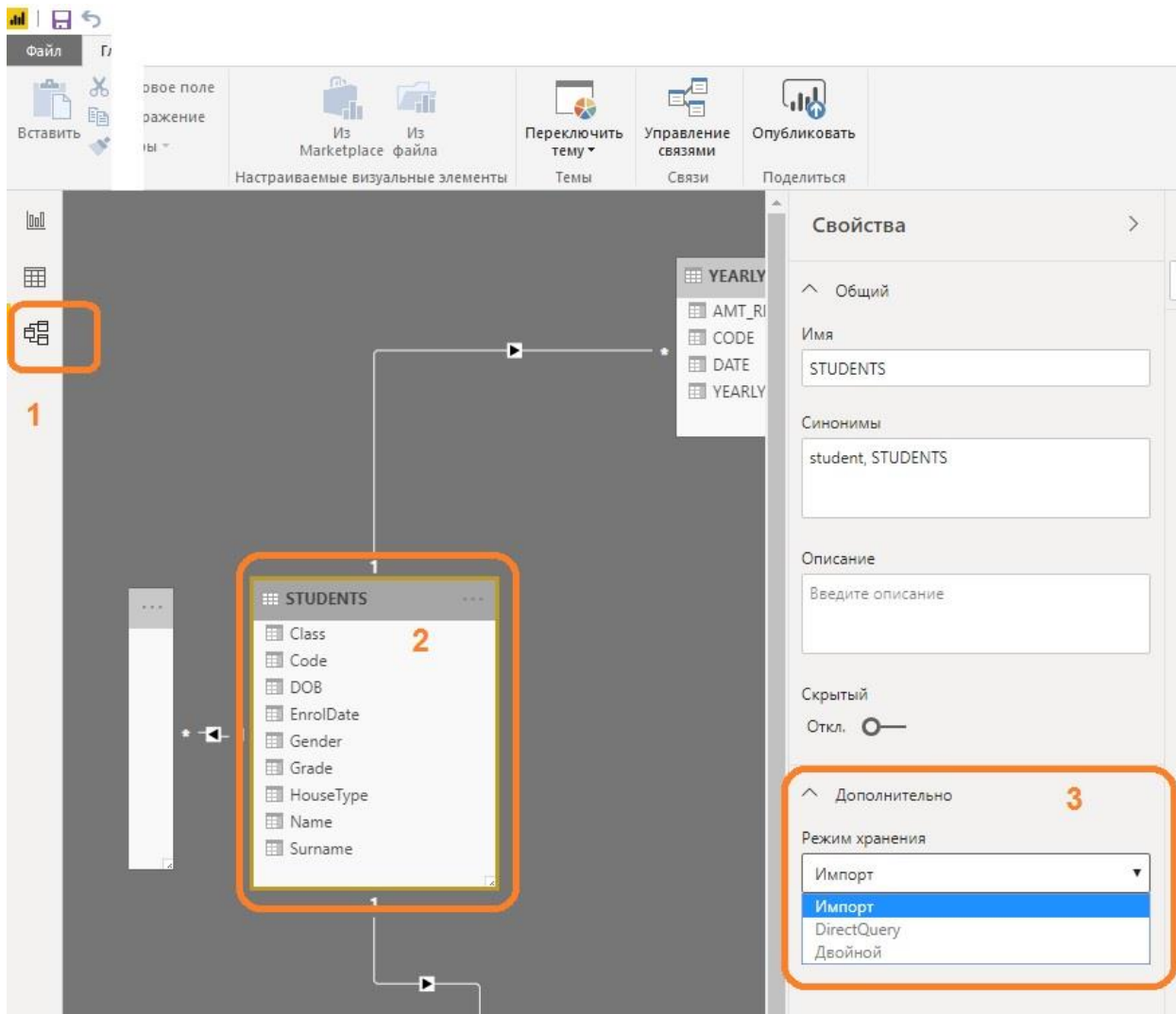


Рис. 4. Выбор режима хранения данных

В режиме *Импорт* все таблицы кешируются и физически загружаются в модель данных. Запрос к источнику направляется в модель данных. Режим импорта – единственный тип подключения, доступный вне зависимости от типа источника данных. Выбрав эту опцию, вы не сможете отменить свое действие. Т.е. другие режимы хранения будут недоступны (на рис. 4 они серые).

В режиме *DirectQuery* таблицы не кешируются, и каждый запрос к данным, выполненный в Power BI, транслируется в язык запросов, характерный для конкретного источника. Этот режим может быть более медленным. Плюсом же является то, что источник данных не нуждается в обновлении. С другой стороны вы будете ограничены по части применения языка DAX в плане моделирования данных и применения преобразований в Power Query.

Двойной режим позволяет применять кеширование или отказаться от него, например, по разным источникам данных. Двойной режим может быть использован с целью повышения производительности и ограничения количества слабых связей, возникающих в модели данных при наличии таблиц *DirectQuery*. Модели данных, в которых насчитывается больше одного режима подключения, называются составными.

Выбор режима хранения информации – важный шаг, и нужно с самого начала определиться с тем, какой тип подключения будет использоваться. Это позволит избежать ненужных хлопот с повторным импортом и установкой подключения по ходу работы. В Power BI режим хранения данных не может быть изменен «на лету».

Где Power BI хранит данные

Когда отчет открыт в Power BI Desktop, набор данных физически хранится в памяти компьютера, на котором запущен отчет. После публикации отчета на сайте Power BI информация перетекает в облако. При использовании режима подключения *Импорт* данные хранятся в службе Microsoft

SQL Server Analysis Services. Чтобы увидеть запущена ли эта служба, кликните правой кнопкой мыши по кнопке *Пуск* панели задач Windows. Выберите *Диспетчер задач*. Раскройте список опций Power BI Desktop. Найдите процесс Microsoft SQL Server Analysis Services.

Глава 8. Создание дашбордов при помощи Power Query

Процесс включает несколько шагов: подключение к источнику данных, обработка данных в Power Query, связывание таблиц в модели данных, дополнительные вычисления с помощью новых столбцов и мер, формирование отчетов и визуализация. Все эти шаги выполнены на учебном примере.

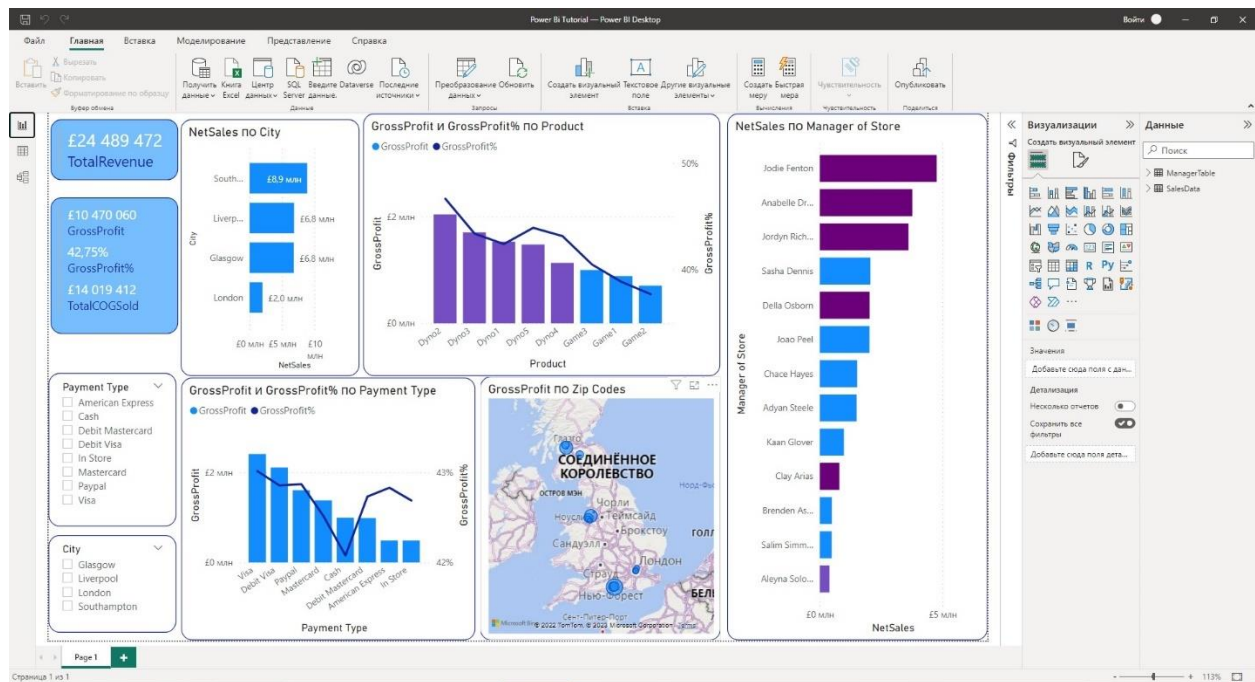


Рис. 5. Финальный вид учебного примера

Глава 9. Работаем с языком M

Язык программирования, использующийся в Power Query, получил имя M, поскольку его рабочее название было Mashup (смешивание, *англ.*). M является функциональным языком. Он чувствителен к регистру символов как в отношении имен переменных, так и в именах функций.

В Excel пройдите *Данные* → *Получить данные* → *Из других источников* → *Пустой запрос*. В окне редактора Power Query пройдите *Главная* → *Расширенный редактор*. По умолчанию открывшийся фрагмент кода будет иметь следующий вид:

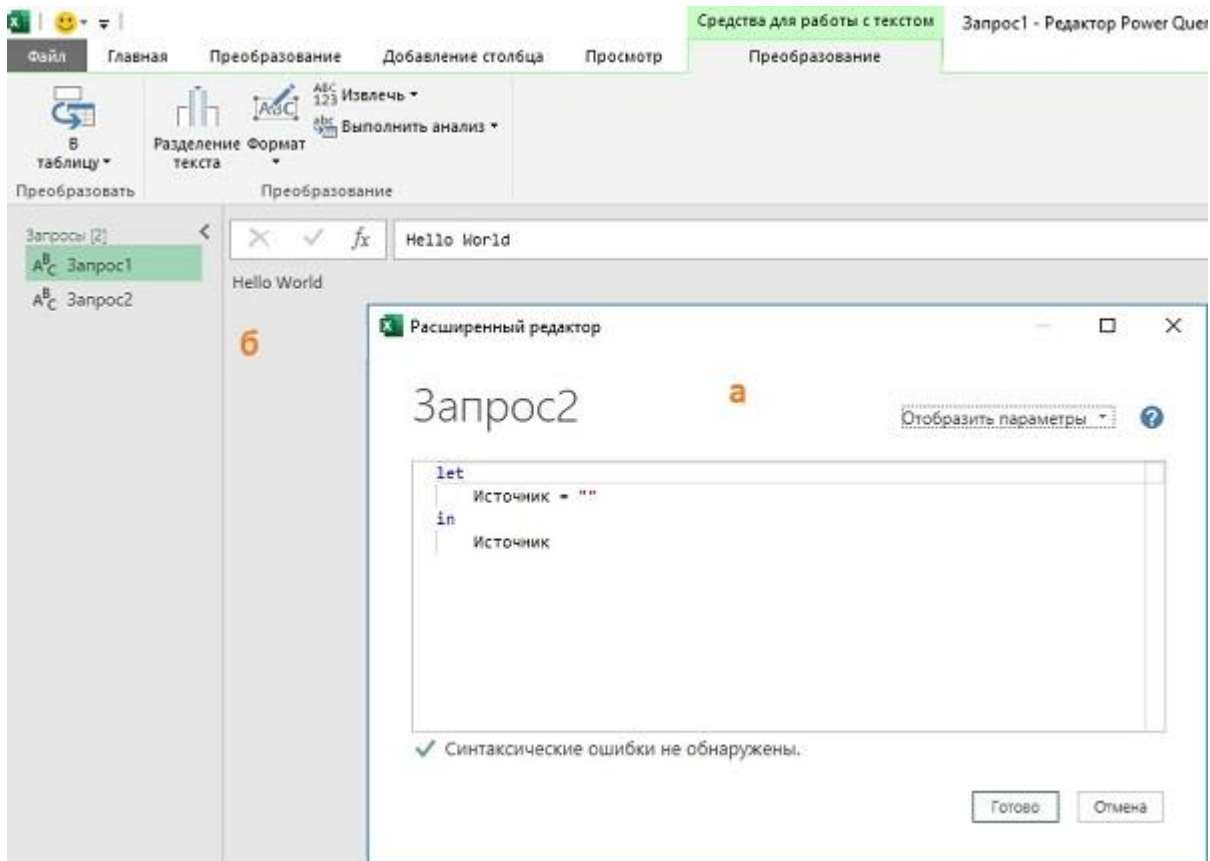


Рис. 6. (а) пустой запрос в Расширенном редакторе; (б) результат запроса 1 Hello World

Блок выражений `let` содержит программный код, он же набор процедурных шагов. Обычно каждый последующий шаг выводится из предыдущего, хотя это не всегда так. Каждый запрос в языке M представляет – одно выражение `let`. Каждый процедурный шаг соответствует одной переменной. Блок выражений `in` возвращает значение переменной, основываясь на блоке `let`.

В пустом запросе основа для блоков `let` и `in` уже готова, остается наполнить их. Если вставить текст `Hello World` в блок `let` внутри кавычек, то мы получим соответствующий текст.

Каждая строка в блоке `let` должна заканчиваться запятой, кроме последней строки, стоящей перед `in`. Имена переменных пишутся без пробелов в одно слово с использованием прописных букв или символа подчеркивания `_` в начале каждого слова. Имена переменных с пробелами предваряются символом `#` и берутся в кавычки, например, `#"This is the long variable name"`.

Имена переменных могут также включать специальные символы, такие как `%` или `&`. Имя переменной показывается на панели ПРИМЕНЕННЫЕ ШАГИ в качестве названия шага. Для размещения комментариев в начале строки можно используйте символы `//`. Круглые скобки `()` используются для передачи параметров функциям, квадратные `[]` – для записей, а фигурные `{}` – для списков.

Большинство функций в языке M являются частью класса *object*, с синтаксисом: `ObjectClass.Function()`, `ObjectClass.Function(Parameter)`, `ObjectClass.Function(Parameter1, parameter2)`. Несколько функций не принадлежат базовому классу *object*, например, `date`. Чтобы использовать ее, следует написать `#date()`.

Чтобы легче ориентироваться в коде на языке M, обращайтесь внимание на скобки, дабы понимать, с чем мы имеем дело – с переменными, списками или набором записей.

Текстовый тип данных

Иногда нет необходимости явно указывать тип данных, поскольку он может быть определен автоматически. С другой стороны функция `Text.format()` позволяет вставлять значения внутри текстовой строки:

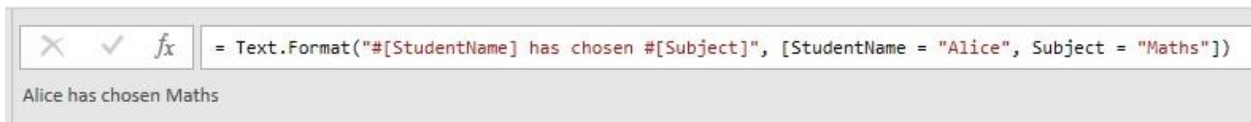


Рис. 7. Функция Text.format()

Эту же функцию можно использовать и для передачи целого списка:

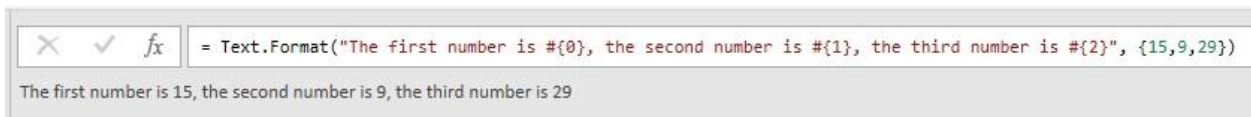


Рис. 8. Функция Text.format() может передавать список

Числовой тип данных

Анализатор кода языка M автоматически присваивает числовой тип данных (Number) вычисляемым переменным, когда это возможно. Создайте пустой запрос и вставьте в него код:

Запрос 1¹

```
let
    variable1 = 5,
    variable2 = 10,
    variable3 = variable1 + variable2
in
    variable3
```

На панели *Примененные шаги* в правой части экрана наши переменные преобразились в шаги запроса:

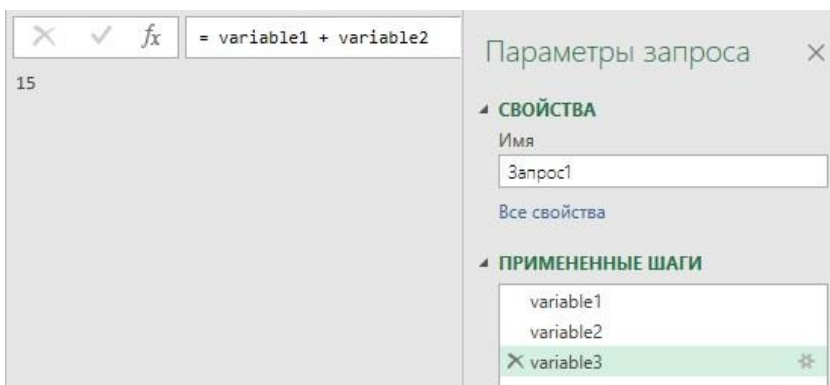


Рис. 9. Шаги запроса

Числовой тип данных включает в себя целые значения, числа с десятичными знаками и валюту.

Списки

Список (List) представляют собой отсортированный набор значений. Чтобы создать список в языке M, откройте редактор запросов Power Query и введите:

Запрос 2

```
let
    fruit={"Apple","Grapes","Pear"}
in
    fruit
```

Слева от названия запроса на левой панели мы увидим иконку списка, а в меню автоматически появится вкладка Средства для списков. Вы можете преобразовывать список в таблицу.

¹ По номеру запроса код можно найти в приложенном Excel-файле

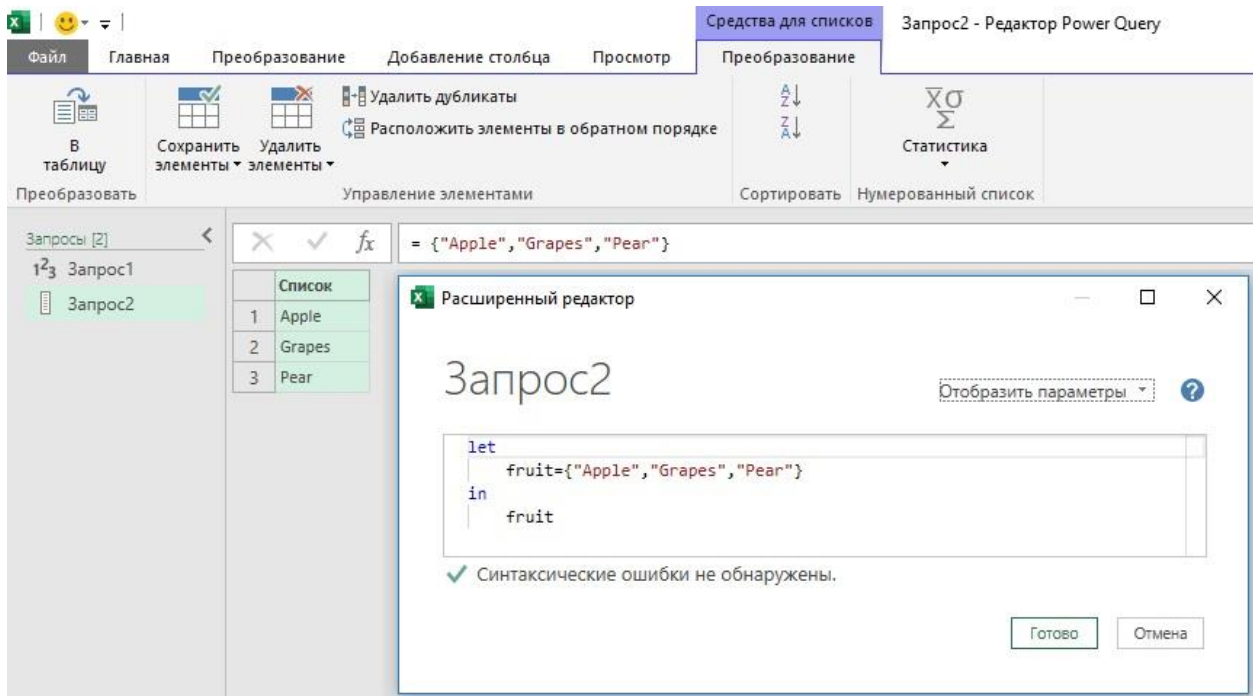


Рис. 10. Код создания списка и меню списка

Можно создавать вложенные списки с использованием конструкций вида `{{},{}}`:

Запрос 3

```
let
    fruitNested = {"Apple", "Grapes"}, {"Limes", "Lemons"}
in
    fruitNested
```

Результат вывода – два списка. Кликнув справа от списка (но не в слово List), в нижней части экрана вы увидите из каких элементов он состоит.

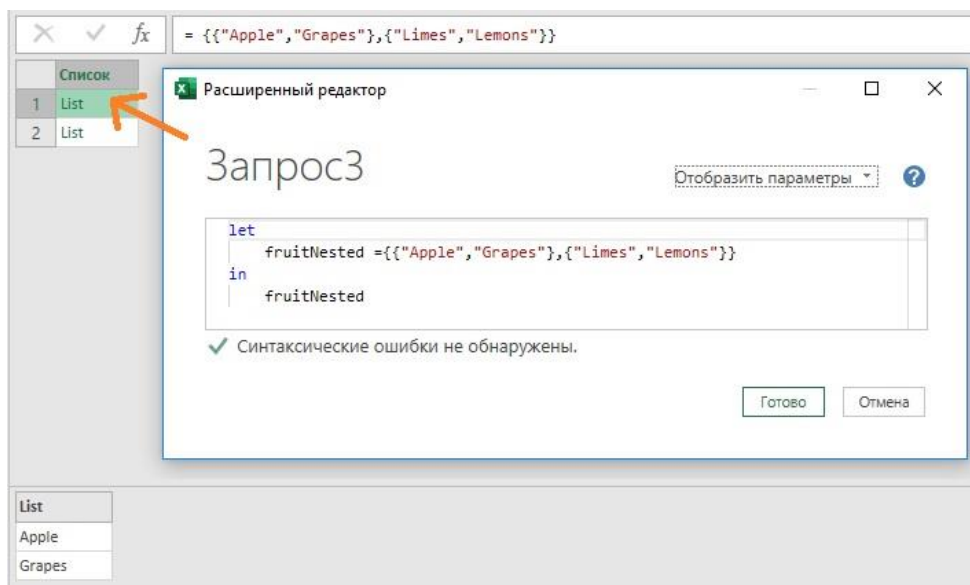


Рис. 11. Вложенные списки

Если объявить переменную `numbersToTen = {1..10}`, мы получим список из чисел: от 1 до 10. Две точки между числами задает целочисленную последовательность.

Список	
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Рис. 12. Числовой список

Записи

Запись (Record) представляет собой разделенный запятыми список, в котором значения ассоциируются с разными полями. Объявляют записи при помощи квадратных скобок [].

Запрос 5

```
let
    record = [firstName="Warren",surname="Sparrow",title="Mr"]
in
    record
```

Мы создали разделенный запятыми список полей: firstName, surname и title. Содержимое полей записано справа от их имен после знака равенства. Текст берется в кавычки. На ленте редактора Power Query появилась вкладка *Средства для записей*:

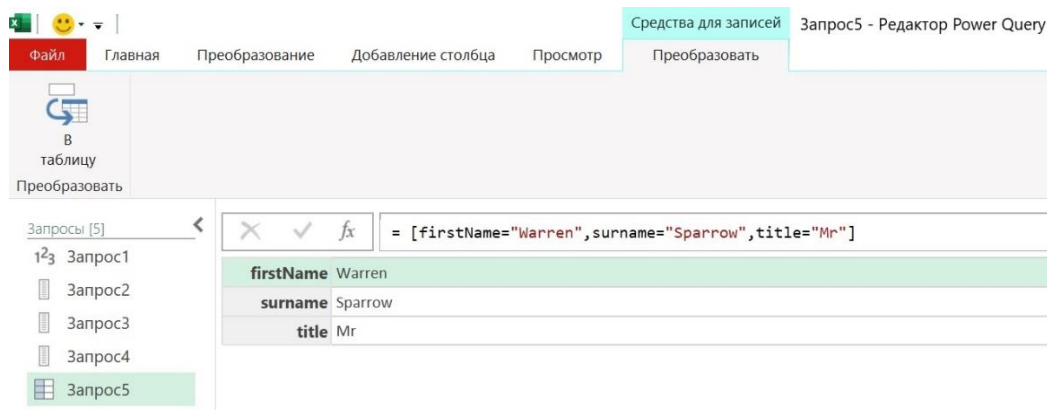


Рис. 13. Запись

Обратите внимание на иконку слева от имени запроса на левой панели.

Табличный тип данных

Табличный тип данных (Table) является наиболее важным из структурных типов в языке M. Его можно представить себе как комбинацию списков и записей:

Запрос 6

```
let
    Источник = #table({"A","B","C","D"},{{"1","2","3","4"},{"10","11","12","13"}})
in
    Источник
```

Функция #table создает таблицу. Таблица состоит из двух частей: списка имен столбцов в текстовом формате (A, B, C и D) и списка строк таблицы.

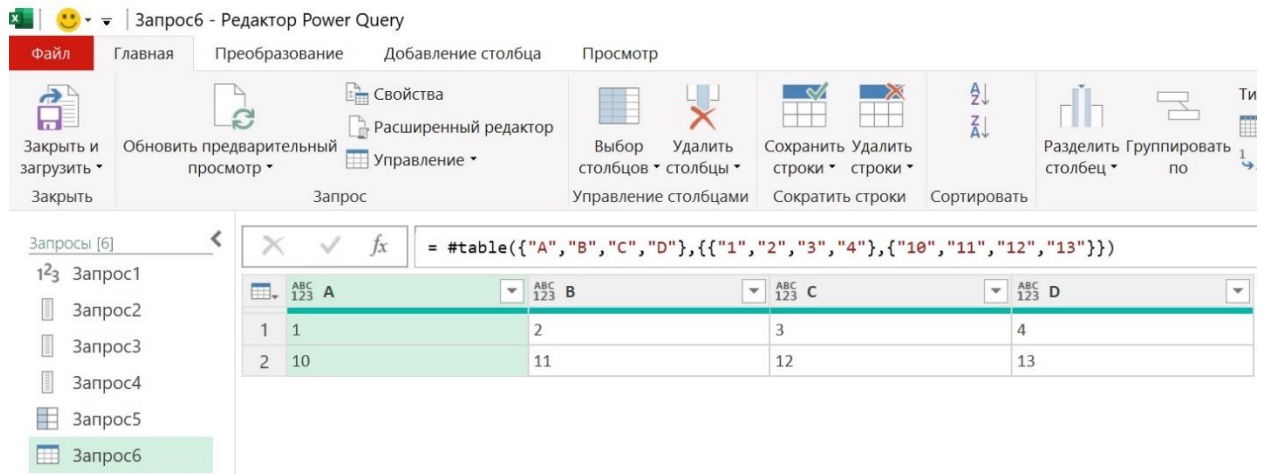


Рис. 14. Таблица

Иконка перед именем *Запрос6* в списке запросов слева соответствует типу *Таблица*.), чтобы мы понимали, что в нем представлена информация в табличном виде. Столбцы таблицы на рис. 14 имеют формат any. Можно задать числовой формат при создании таблицы:

Запрос 7

let

```
Источник = #table(type table[A=Int64.Type, B=Int64.Type, C=Int64.Type, D=Int64.Type],
  {{1,2,3,4},{100,101,102,103}})
```

in

Источник

Поиск данных

Мы можем выполнять поиск по спискам, записям и таблицам как по позиции (индексу), так и по вхождению. Помните, что в языке M первый индекс равен 0, а не 1.

Запрос 8

let

```
fruit={"Apple","Grapes","Pear"},
numbersToTen = {1..10}
```

in

```
numbersToTen{2}
```

Запрос8 вернет третье значение списка *numbersToTen* равное 3. Если изменить последнюю строку на *fruit{0}*, мы получим на выходе значение *Apple*. Запрос *fruit{6}*, а запрос *fruit{6}?* – null. Вопросительный знак указывает, что следует вернуть пустое значение при ошибке.

Запрос 9

let

```
Источник = #table({"A","B","C","D"},{{"1","2","3","4"}, {"10","11","12","13"}}),
Output = Источник[A]
```

in

```
Output
```

Здесь у нас таблица с заголовками столбцов A, B, C и D. Квадратные скобки [] возвращают поле из записи. Но у нас таблица, поэтому строка *Output = Source[A]* вернет список значений из колонки A. Если изменить [A] на {0}, поиск выдаст первую строку таблицы – 1, 2, 3, 4 (*Запрос10*). Допустимо комбинировать эти методы. Например, *Источник[A]{0}* вернет значение из первого столбца и первой строки (*Запрос11*).

Если в пустом запросе ввести *=#shared*, на экран в виде записи будет выведен полный список функций (*Запрос12*). Щелчок мыши по имени функции приведет к открытию документации по ней.

Глава 10. Примеры использования языка M

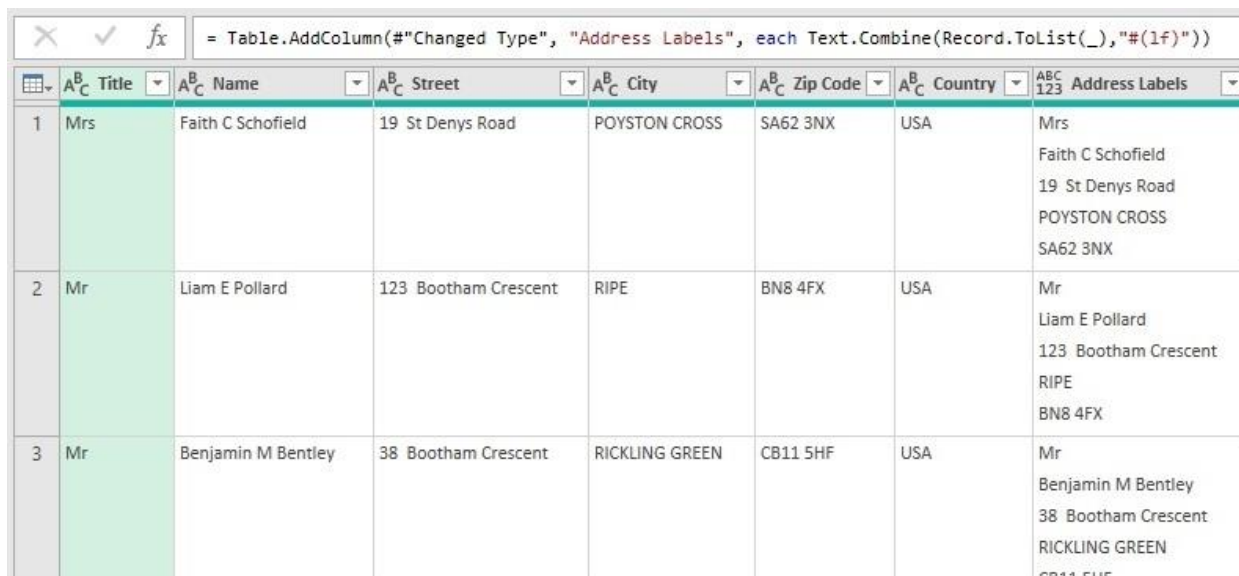
Преобразование типов данных

Для объединения полей несовместимых типов данных используют функцию Text.From. Она принимает в качестве параметра число, дату, время или двоичные данные и возвращает их текстовое представление. Например, формула Text.From(7) вернет "7".

Откройте файл Address.xlsx и перейдите в редактор запросов Power Query. В окне ПРИМЕНЕННЫЕ ШАГИ кликните правой кнопкой мыши на последнем шаге и выберите *Вставить шаг после*. Введите в строку формул:

```
= Table.AddColumn("#Changed Type", "Address Labels", each Text.Combine(Record.ToList(_), "#(lf)"))
```

Формула создаст новый столбец с именем Address Labels, в который соберет данные из всех полей слева направо. Выражение #(lf) означает переход строки.



	Title	Name	Street	City	Zip Code	Country	Address Labels
1	Mrs	Faith C Schofield	19 St Denys Road	POYSTON CROSS	SA62 3NX	USA	Mrs Faith C Schofield 19 St Denys Road POYSTON CROSS SA62 3NX
2	Mr	Liam E Pollard	123 Bootham Crescent	RIPE	BN8 4FX	USA	Mr Liam E Pollard 123 Bootham Crescent RIPE BN8 4FX
3	Mr	Benjamin M Bentley	38 Bootham Crescent	RICKLING GREEN	CB11 5HF	USA	Mr Benjamin M Bentley 38 Bootham Crescent RICKLING GREEN CB11 5HF

Рис. 15. Объединение столбцов

Подробно описаны:

- установка SQL Server версии Developer (это бесплатный вариант);
- использование параметров в запросе на языке T-SQL для подключения к той или иной базе данных и для фильтрации данных;
- использование параметров для изменения логики выражений; например, можно по параметру менять функцию SUM на AVG, тем самым влияя на агрегацию в выражении;
- использование параметров для изменения порядка сортировки.

В Power BI существует возможность задавать значения всех параметров без открытия окна редактора Power Query. В любом из представлений перейдите на вкладку *Главная*, далее *Преобразование данных* → *Изменить параметры*.

В начале главы мы кратко сравнили языки, присущие Excel и Power BI, на примере выполнения конкатенации данных. Мы поговорили об использовании оператора амперсанд (&) применительно к объединению строк, дат и других объектов, после чего посмотрели на сходства и различия этого оператора со связкой функций Text.From и Text.Combine.

Глава 11. Создание базовых пользовательских функций

Описано превращение запроса в функцию и использование функции для обработки всех файлов в папке. Ранее я описал эти шаги для похожей задачи – [извлечение данных из нескольких API-страниц](#).

Полный список функций языка M для работы с датой и временем можно найти по адресу: <https://learn.microsoft.com/ru-ru/powerquery-m/date-functions>. Удобство электронной документации Microsoft состоит в возможности копирования блоков кода и вставки в окно расширенного редактора Power Query или в строку формул.

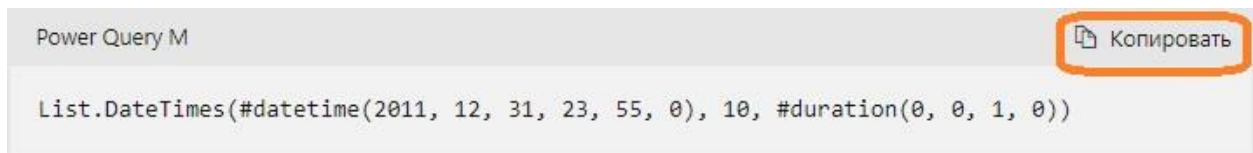


Рис. 16. Копирование кода с [сайта](#) Microsoft

Откройте файл SSGSalesD.xlsx. Кликните на любую ячейку с данными, и пройдите *Данные* → *Получить и преобразовать данные* → *Из таблицы/диапазона*. Для определения разницы в днях между двумя датами в редакторе Power Query выделите столбцы *Date Delivered* и *Date Sold*. Порядок выделения важен. Пройдите *Добавление столбца* → *Дата* → *Вычесть дни*. Результат будет выведен в отдельном столбце. Изучите код на языке M, сгенерированный для наших шагов.

	A ^B Region	A ^B Product	1.2 Sales	Date Sold	Date Delivered	1 ² Вычитание
1	West	Board5	53395,17732	31.12.2019 0:00:00	01.02.2020 0:00:00	32
2	West	Game6	116609,6948	31.10.2019 0:00:00	01.04.2020 0:00:00	153
3	South	Game1	72524,0953	30.12.2019 0:00:00	01.05.2020 0:00:00	123
4	South	Game5	22538,47873	30.09.2019 0:00:00	01.06.2020 0:00:00	245
5	North	Dyno4	45616,53282	29.12.2019 0:00:00	01.06.2020 0:00:00	155
6	North	Game5	81902,36089	29.11.2019 0:00:00	01.03.2020 0:00:00	93
7	South	Board5	72000,3749	20.10.2019 0:00:00	08.02.2020 0:00:00	111
8	North	Game10	99227,6778	01.10.2019 0:00:00	01.12.2019 0:00:00	61
9	West	Dyno1	93520,54806	15.11.2019 0:00:00	23.03.2020 0:00:00	129
10	South	Game5	77445,82665	29.09.2020 0:00:00	04.04.2020 0:00:00	-178
11	South	Dyno5				
12	South	Game2				
13	East	Dyno1				
14	South	Game1				
15	East	Game10				
16	West	Board2				

Расширенный редактор

Таблица1

```

let
    Источник = Excel.CurrentWorkbook(){[Name="Таблица1"]}[Content],
    #"Измененный тип" = Table.TransformColumnTypes(Источник, {
        {"Region", type text},
        {"Product", type text},
        {"Sales", type number},
        {"Date Sold", type datetime},
        {"Date Delivered", type datetime}
    }),
    #"Вычитание вставленных дат" = Table.AddColumn(
        #"Измененный тип",
        "Вычитание",
        each Duration.Days([Date Delivered] - [Date Sold]),
        Int64.Type
    )
in
    #"Вычитание вставленных дат"

```

Рис. 17. Добавление столбца с разницей двух дат

Второй способ заключается в использовании функции [List.DateTimes](#). Она формирует список дат или времен. Создайте новый пустой запрос с именем NumbDays. Откройте расширенный редактор и переименуйте *Источник* в *NumbDays*. Это заготовка функции, которую мы будем использовать позже в формуле List.DateTimes.

Расширенный редактор

NumbDays

```

let
    NumbDays = ""
in
    NumbDays

```

Рис. 18. Переименование *Источника* в расширенном редакторе

В качестве вычисления (второй строки кода) введите:

NumbDays = #datetime(2020,06,17,00,00,00)-#datetime(2019,01,01,00,00,00)

Нажмите *Готово*. Отобразится разницу между двумя датами:

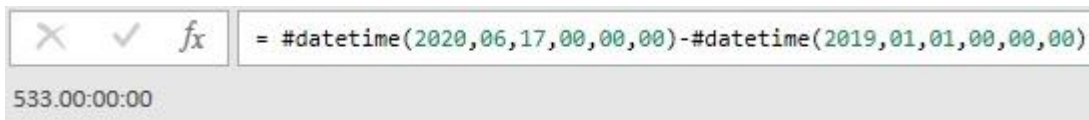


Рис. 19. Вычитание одной даты из другой для получения разницы в днях между ними

Нажмите f_x слева от строки формул, чтобы добавить шаг. Введите:

```
List.Datetimes(#datetime(2019, 01, 01, 00, 00, 0), 10, #duration(1, 0, 0, 0))
```

Эта функция сгенерирует список с десятью значениями с датой и временем начиная с 1 января 2019 года с интервалом в один день. Подобная инструкция может быть использована применительно к дням, часам, минутам и секундам.

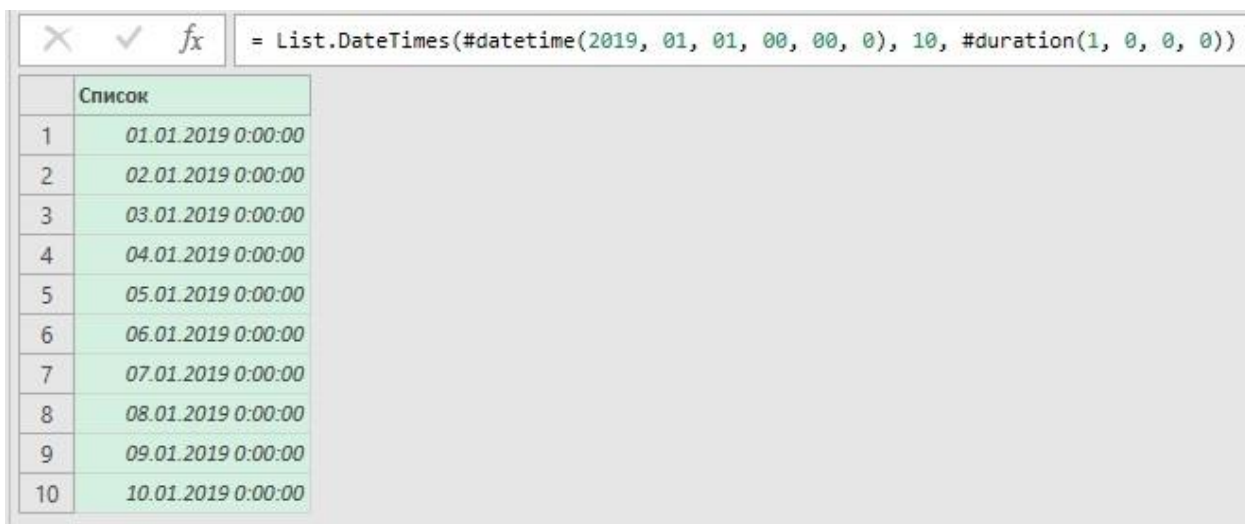


Рис. 20. Создание шага

Заменяем фиксированное значение 10 в формуле на переменную *NumbDays*. Поскольку *NumbDays* не является числом периодом, заменим шаг

```
NumbDays = #datetime(2020,06,17,00,00,00)-#datetime(2019,01,01,00,00,00)
```

на

```
NumbDays = Duration.TotalDays(#datetime(2020,06,17,00,00,00)-#datetime(2019,01,01,00,00,00))
```

Преобразуем список в таблицу. Пройдите *Преобразование* → *В таблицу*. В появившемся окне *В таблицу* оставьте значения по умолчанию. Нажмите *Ок*.

Вместо фиксированной даты окончания периода можно ввести текущую дату и время:

```
DateTime.LocalNow()
```

тогда календарь будет начинаться 01.01.2019, а заканчиваться текущей датой. Чтобы увидеть это отсортируем даты по убыванию:

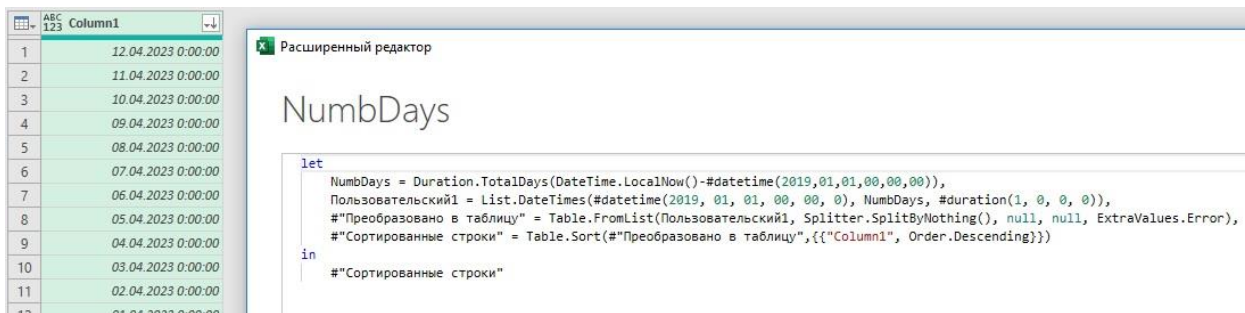


Рис. 21. Добавление ссылки на текущую дату в функцию

Глава 12. Различия между DAX и M

Язык M считается полноценным функциональным языком среды Power Query. Его официальным названием является Mashup или Power Query Formula Language (Язык формул Power Query), и он

используется для запросов к различным источникам и преобразования полученных данных. В то же время Data Analysis eXpressions (DAX) представляет собой набор функций для работы с данными, хранящимися в виде таблиц, подобно Excel.

DAX – язык формул, он не считается языком программирования в полном смысле слова, поскольку строится на основе пользовательских вычислений. DAX строится на следующих структурных единицах: операторы, константы, функции.

Структурно DAX состоит из запросов DAX (Analysis Services, DAX Studio и SQL Management Studio) и формул DAX (Power BI, Excel и Power Pivot). Функции DAX и Excel во многом схожи в плане поведения и семантики, но отличаются в синтаксисе. В Excel используются ссылки на ячейки и диапазоны, тогда как DAX оперирует в формулах столбцами и таблицами. Недопустимо смешивать функции Excel и DAX в одной формуле. DAX всегда требует наличия связей между таблицами при выполнении операций поиска.

Технология *IntelliSense* позволяет разработчику получить полный список допустимых функций и их параметров при написании кода на DAX (аналогично вводу формул в Excel). Работая в расширенном редакторе Power Query, слева от элемента вы увидите специальную иконку, представляющую собой куб, треугольник, квадрат или круг. Куб символизирует родные функции языка M, а треугольник, квадрат и круг применяются для описания переменных, шагов, параметров, констант, имен запросов и т.д.

Написание формул DAX

Основные типы формул на языке DAX в Power BI Desktop:

- вычисляемые столбцы: используются для выполнения построчных вычислений путем создания нового столбца в таблице;
- быстрые меры: позволяют воспользоваться заранее определенными шаблонами вычислений, которые можно создать без знания синтаксиса языка DAX;
- вычисляемые меры: меры не занимают физическую память, а вычисляются на лету исходя из контекста. Такие вычисления всегда требуют определенного типа агрегации;
- вычисляемые таблицы: служат для создания новых таблиц с данными. Совсем скоро мы познакомимся с этими типами вычислений поближе.

За обработку выражений на языке DAX отвечают два движка: движок формул (FE) и движок хранилища (SE).

Создадим вычисляемый столбец. В Power BI откройте файл SSGThemePark.pbix. Формулы DAX допустимо создавать, находясь в любой вкладке, но я предпочитаю делать это на вкладке *Данные*. Удобно видеть заголовки столбцов при работе с формулами. Пройдите *Средства работы с таблицами* → *Создать столбец*. Введите формулу на языке DAX в строку формул. Например, вычислим заработную плату сотрудника, как произведение отработанных часов и ставки.

The screenshot shows the Power BI Desktop interface. The ribbon is set to 'Средства работы со столбцами' (Columns Tools). The formula bar contains the DAX formula: `1 SALARY = Table1[HRS]*Table1[HOURLY RATE]`. Below the formula bar, a table is displayed with columns: CODE, FIRST, SURNAME, EMP NO, DIVISION, DEPT, DATE OF HIRE, HRS, HOURLY RATE, and SALARY. The SALARY column is highlighted in green, indicating it is a calculated column. The data in the table is as follows:

CODE	FIRST	SURNAME	EMP NO	DIVISION	DEPT	DATE OF HIRE	HRS	HOURLY RATE	SALARY
69	Shirley	Dandrow	TBV45	View Tabue	Cobrella	14 марта 1991 г.	.35	12,1	423,5
12	Seth	Rose	TBV76	View Tabue	Cobrella	5 апреля 1990 г.	.32	5,5	176
47	Anne	Davidson	TBV23	View Tabue	Cobrella	6 апреля 1986 г.	.25	8,52	213
5	Colleen	Abel	TBV58	View Tabue	Mankay Falls	26 июля 1990 г.	.42	16,75	703,5
15	Theresa	TBVlifano	TBV19	View Tabue	Mankay Falls	26 февраля 1989 г.	.35	12,1	423,5
55	Jennifer	Snyder	TBV30	View Tabue	Mankay Falls	7 июня 1988 г.	.25	8,52	213
66	Chris	Reed	TBV03	View Tabue	Mankay Falls	27 марта 1984 г.	35.5	13.3	472.15

Рис. 22. Вычисляемый столбец в Power BI

Вычисляемые меры

Мера не может работать без применения к столбцу, на который ссылается. Мера включает в себя функцию. Функции называются агрегаторами, и без их применения столбец будет именоваться голым. При написании мер используется технология дополнения формул *IntelliSense*. Меры возвращают разные значения в зависимости от примененного к ним критерия фильтра. В Power BI меры создаются на вкладке *Отчет* или *Данные*.

Быстрые меры основаны на шаблонах, и не требуют знания DAX. Откройте файл SalesData.pbix. На вкладке *Представление данных* пройдите *Средства работы с таблицами* → *Вычисления* → *Быстрая мера*. В окне *Быстрая мера* в списке *Вычисление* выберите *Итого с начала года*. Раскройте список полей *Sheet1* справа, и перенесите поле *TOTAL SALES* в область *Базовое значение*. Можно выбрать тип агрегации значения из выпадающего списка. Перетащите поле *DATE SOLD* в область *Дата*.

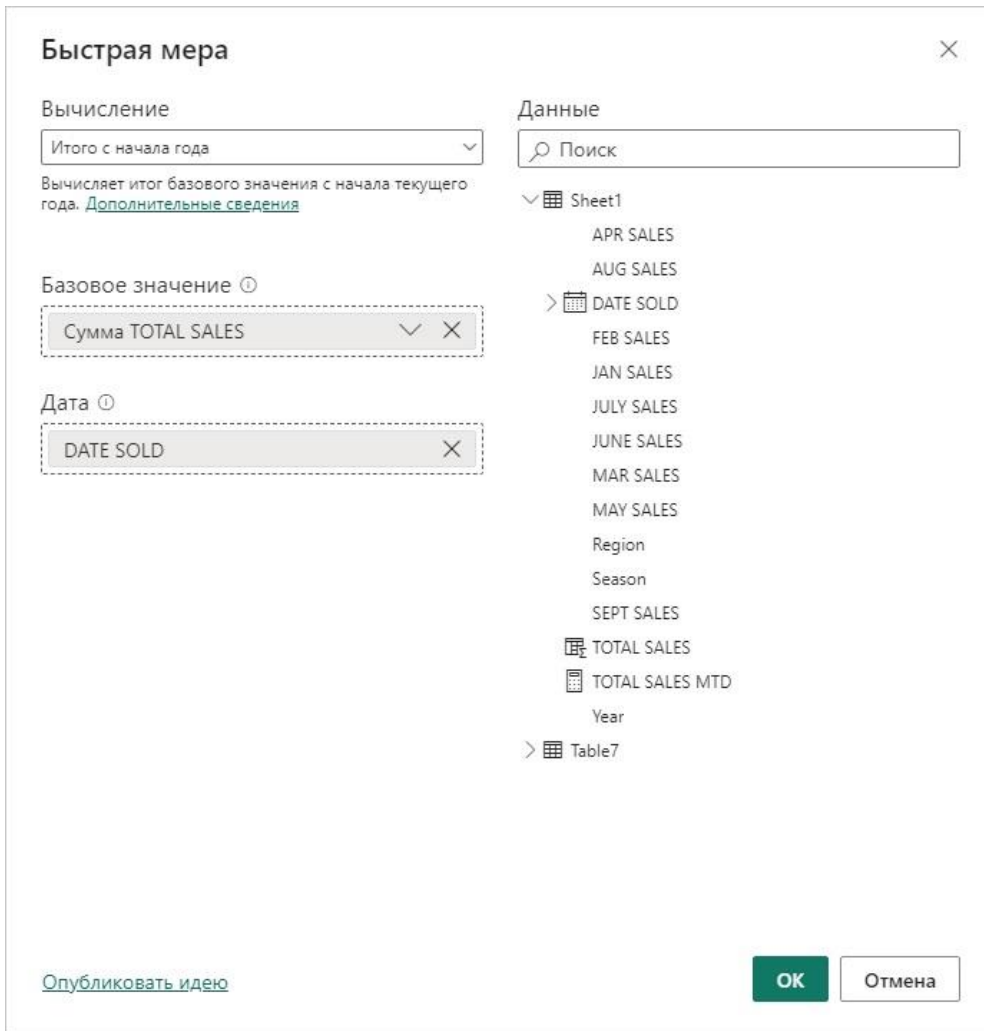


Рис. 23. Настройка быстрой меры

Нажмите *Ok*, чтобы Power BI сгенерировал формулу на языке DAX. Новая мера будет добавлена на панель *Данные*.

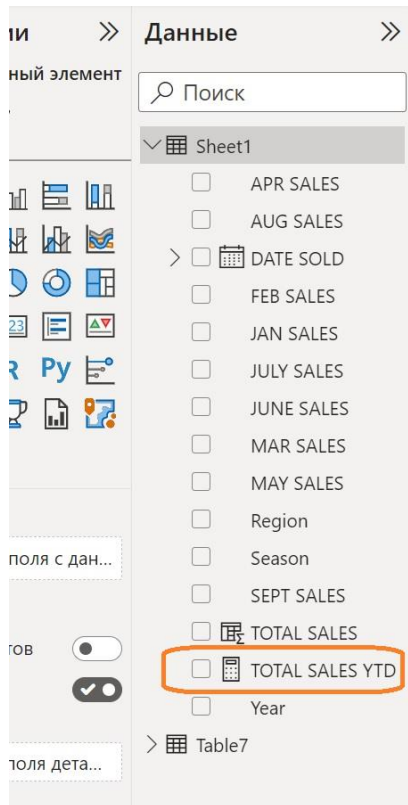


Рис. 24. Новая мера

Перетащите созданную меру TOTAL SALES YTD на дашборд с выведенной таблицей. Кликните на имени меры в поле *Данные*. Появится строка формул. Вы увидите, какой код на языке DAX был автоматически сгенерирован.