

## Глава 17. Формулы Power Query

Это продолжение перевода книги Кен Пульс и Мигель Эскобар. Язык M для Power Query. Главы не являются независимыми, поэтому рекомендую читать последовательно.

[Предыдущая глава](#) [Содержание](#) [Следующая глава](#)

Пользовательский интерфейс Power Query позволяет выполнять огромное число операций. Но наверняка возникнут моменты, когда вам потребуется что-то сделать, что не встроено в интерфейс. Вот мы и добрались до языка программирования Power Query: M.

Хотя некоторые аспекты M могут быть довольно сложными, есть простой способ получить представление о языке, начав с формул в пользовательских столбцах. Поскольку Power Query был создан для профессионалов Excel, можно было ожидать, что его язык будет подобен языку формул Excel. К сожалению, это не так.

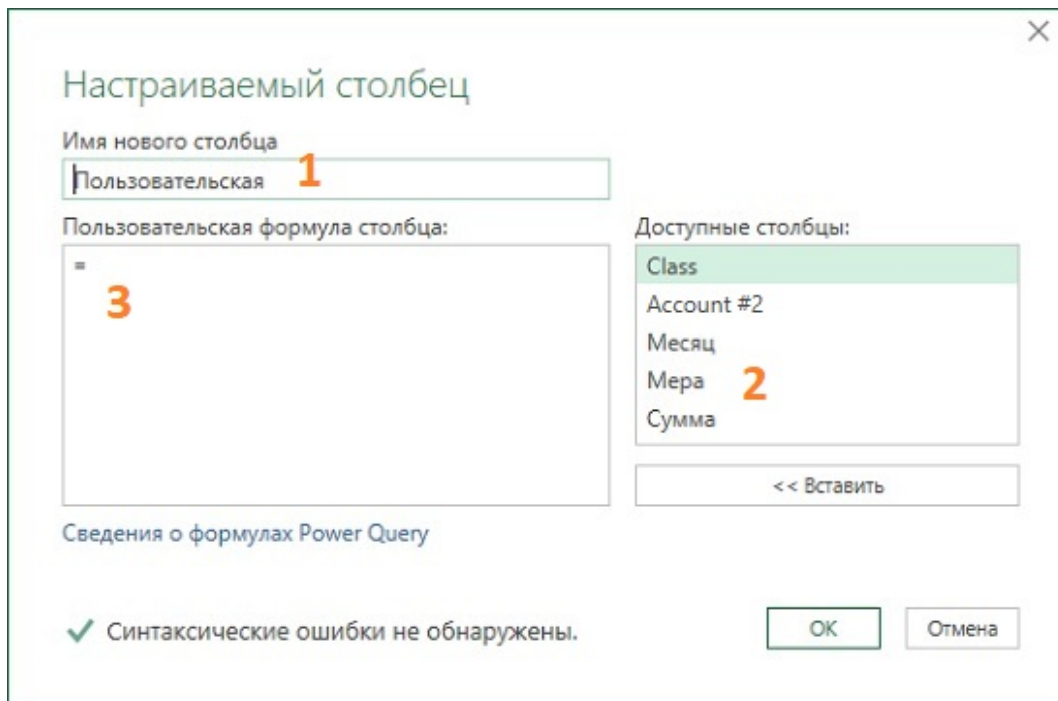


Рис. 17.1. Интерфейс создания пользовательского столбца

### Создание пользовательских столбцов

Окно создания настраиваемого столбца содержит три важные части (см. 17.1):

1. Имя столбца
2. Доступные столбцы: здесь перечислены имена всех столбцов в запросе. Двойной щелчок любого элемента в этом поле помещает его в область формулы с правильным синтаксисом для ссылки на поле.
3. Пользовательская формула столбца – место, где вы записываете формулу.

Некоторые функции, доступные в Excel, могут использоваться и здесь. Например, чтобы объединить два текстовых поля, постройте формулу следующим образом:

- Дважды щелкните *Class* в списке *Доступные столбцы* (или выделите строку *Class* и нажмите кнопку *Вставить*)
- Введите символ `&`
- Дважды щелкните столбец *Account #2* в списке *Доступные столбцы*

Power Query построит формулу: `=[Class]&[#"Account #2"]`

Самое замечательное в использовании интерфейса двойного щелчка заключается в том, что для вас не имеет значения, что синтаксис *Class* и *Account #2* должен обрабатываться по-разному. Этот специфический синтаксис объясняется в главах 19 и 20. Вы также можете выполнить все четыре арифметических действия: `-`, `+`, `/`, `*`. В то же время для возведения в степень нужно применить формулу: `=Number.Power([Column1],[Column2])`.

К сожалению, Power Query не имеет подсказок, чтобы выяснить, какие функции можно использовать. Но... в окне *Настраиваемый столбец* есть гиперссылка *Сведения о формулах Power Query* (см. рис. 17.1). Щелкнув на нее, вы попадете на страницу с подробным каталогом функций, правда, на английском языке. В отличие от Excel, но аналогично Power Pivot, функции Power Query не русифицированы.

## Подводные камни формул на языке M

Power Query и Excel существенно различаются в том, как они обрабатывают входные данные.

| Excel  | Power Query                              |
|--|--|
| Формулы не чувствительны к регистру            | Формулы чувствительны к регистру         |
| Счет начинается с 1                            | Счет начинается с 0                      |
| Преобразования типов данных выполняется неявно | Требует строгого определения типа данных |

Рис. 17.2. Различия Power Query и Excel в обработке данных

**Чувствительность к регистру.** Запомните, что в 99% случаев первая буква каждого слова в формуле на языке M – заглавная, а остальные – строчные. В то время как Excel не заботится, какие буквы вы используете и преобразует формулы в верхний регистр по умолчанию, Power Query просто возвращает ошибку.

**База 0 против базы 1.** Если бы вас спросили о номере позиции буквы x в слове Excel, вы сказали бы 2. Это логично, и так считает программа MS Excel. Но Power Query скажет, что буква x в слове Excel занимает позицию 1.

**Преобразование типов данных.** В Excel вы можете добавить единицу к дате, что изменит дату на один день. В Power Query, если дата отформатирована с типом *Даты*, необходимо использовать специальную формулу чтобы добавить к ней день. Если вы попытаетесь использовать ту же формулу для добавления дней к числу, Power Query вернет ошибку. Это означает, что перед использованием столбцов в формулах необходимо явно задавать в них тип данных.

В Excel вы можете объединить две ячейки вместе с помощью функции конкатенации &. Содержат ли ячейки текст или числа, не важно. Excel неявно преобразует их в текст, а затем объединяет вместе:

|   | Столбец1 | Столбец2 | Столбец3 |
|---|----------|----------|----------|
| 1 |          |          |          |
| 2 | 1234     | абв      | 1234абв  |

Рис. 17.3. Неявное преобразование данных в Excel: число и текст, преобразованные в текст

Создайте на основе двух первых столбцов *Таблицы* Excel запрос, а затем внутри Power Query создаете пользовательский столбец, используя формулу: `= [Column1]&[Column2]`:

| 1 | Столбец1 | Столбец2 | Пользовательская |
|---|----------|----------|------------------|
| 1 | 1234     | абв      | Error            |

Expression.Error: Не удастся применить оператор & к типам Number и Text.  
 Сведения:  
 Operator=&  
 Left=1234  
 Right=абв

Рис. 17.4. Power Query не может соединить число и текст вместе

Чтобы устранить эту проблему, необходимо сначала преобразовать тип данных *Столбец1* в текст, а уже затем создать пользовательский столбец:

|   | А <sup>В</sup> С Столбец1 | А <sup>В</sup> С Столбец2 | А <sup>В</sup> С 123 Пользовательская |
|---|---------------------------|---------------------------|---------------------------------------|
| 1 | 1234                      | абв                       | 1234абв                               |

Рис. 17.5. Два текстовых столбца объединить можно

При явном преобразовании данных в столбце 1 в текстовое значение конкатенация будет работать так, как вы изначально предполагали:

На самом деле существует два способа работы с типами данных в Power Query:

- Установите тип данных для столбца, на который вы ссылаетесь, прежде чем использовать его в пользовательской функции.
- Используйте функцию преобразования типа данных, чтобы принудительно преобразовать входные данные в требуемый тип.

### Функции преобразования типов данных

Существует несколько функций преобразования типов данных.

**Преобразование в текст.** Если вам нужно преобразовать значения в столбце в текст, можно использовать универсальную функцию `Text.From()`. Если же вы хотите подчеркнуть тип преобразуемых данных, также в вашем распоряжении есть: `Date.ToText()`, `Time.ToText()`, `Number.ToText()`. Имейте в виду, что `Text.From()` преобразует любой тип данных в текст, в то время как `Date.ToText()` не преобразует число в текст.

**Даты.** Данные, похожие на даты, могут поступать в формате чисел или текста. Для их преобразования есть две функции: `Date.From()` и `Date.FromText()`. Опять же, `Date.From()` справится с преобразованием в формат даты, как чисел, так и текста.

**Время.** Значения времени могут поступать как в виде чисел, так и в виде текста. Опять же, есть две функции для них: `Time.From()` и `Time.FromText()`.

**Длительность** – это разница между двумя значениями даты/времени: `Duration.From()` и `Duration.FromText()`.

**Числа.** Имеется универсальная функция `Number.From()` и несколько специальных. Для чисел из текста `Number.FromText()`, для десятичных чисел `Decimal.From()`, целых чисел `Int64.From()`, валюты `Currency.From()`.

### Сравнение текстовых функций Excel и Power Query

Если вы работали с текстовыми функциями Excel, то привыкли использовать их для извлечения элементов текста из данных. В Power Query текстовые функции работают иначе. Рассмотрим пять наиболее часто используемых текстовых функций Excel, и их аналоги в Power Query. Откройте файл `5 Useful Text Functions.xlsx`. Каждый из примеров в этом разделе начинается с набора данных:

|   | А          | В              |
|---|------------|----------------|
| 1 | Слово      | ЛЕВСИМВ(...;4) |
| 2 | Football   | Foot           |
| 3 | Doorknob   | Door           |
| 4 | Bookkeeper | Book           |
| 5 | Automobile | Auto           |
| 6 | Mountain   | Moun           |
| 7 | Truck      | Truc           |
| 8 | Car        | Car            |

Рис. 17.6. Пример данных

В августе 2015 года команда Power Query добавила возможность извлечения первого, последнего и ряда символов на вкладку *Преобразование*. Несмотря на это, ниже рассматривается процесс извлечения текста с помощью языка M, что позволит глубже познакомиться с языком, и создавать более надежные решения, чем те, что могут быть созданы с помощью команд пользовательского интерфейса.

Итак, что поместить данные, представленные на рис. 17.6, в Power Query, кликните на любой ячейке в диапазоне A1:B8 → *Данные* → *Из таблицы/диапазоне*. Подтвердите создание *Таблицы* с заголовком. В окне редактора Power Query переименуйте запрос *pqLeft*. Перейдите на вкладку *Добавление столбца* → *Настраиваемый столбец*. Назовите новый столбец *pqLeft(x,4)*. Введите формулу: `=LEFT([Слово],4)`. Вроде бы, это должно сработать:

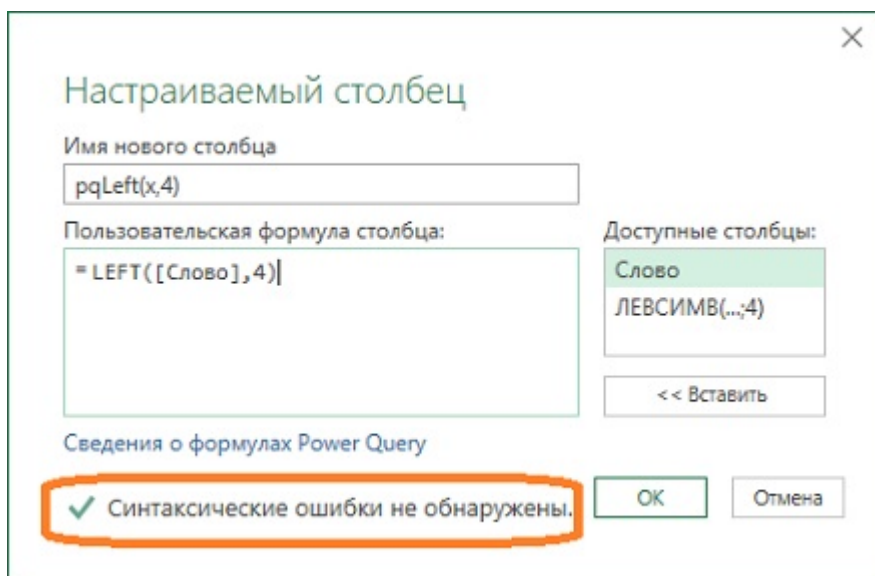


Рис. 17.7. Power Query не находит синтаксических ошибок

Однако, после нажатия *Ok*, появляется ошибка:

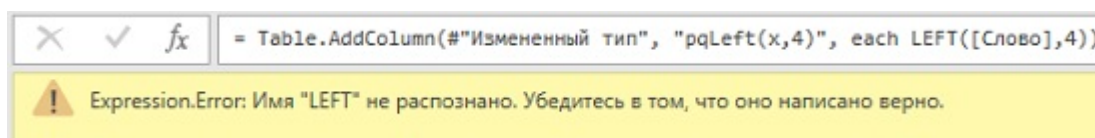


Рис. 17.8. Формула `=LEFT([Слово],4)` не работает

В Power Query используется иной синтаксис `=Text.Start(text,num_chars)`. Отредактируйте формулу. В области ПРИМЕНЕННЫЕ ШАГИ кликните на шестеренку справа от строки *Добавлен пользовательский столбец*, и в окне *Настраиваемый столбец* введите формулу: `=Text.Start([Слово],4)`. Не забывайте, что формулы в Power Query чувствительны к регистру: `Text.start` и `TEXT.START` вернут ошибку. Нажмите *Ok*:

|   | Слово      | ЛЕВСИМВ(...;4) | pqLeft(x,4) |
|---|------------|----------------|-------------|
| 1 | Football   | Foot           | Foot        |
| 2 | Doorknob   | Door           | Door        |
| 3 | Bookkeeper | Book           | Book        |
| 4 | Automobile | Auto           | Auto        |
| 5 | Mountain   | Moun           | Moun        |
| 6 | Truck      | Truc           | Truc        |
| 7 | Car        | Car            | Car         |

Рис. 17.9. Функции `ЛЕВСИМВ()` в Excel соответствует `Text.Start()` в Power Query

Теперь вы можете завершить запрос: *Главная* → стрелочка вниз возле кнопки *Закреть и загрузить* → *Закреть и загрузить в...* → *Только создать подключение*.

Никогда не используйте имя функции Excel в качестве имени запроса Power Query. Если бы вы назвали запрос `ЛЕВСИМВ`, вы бы получили ошибки в формулах Excel исходной *Таблицы*. Имена таблиц обрабатываются перед функциями.



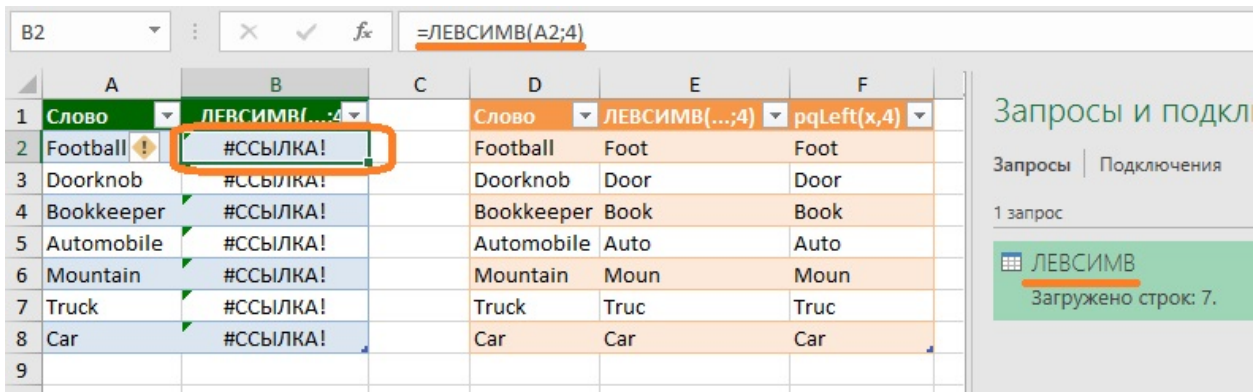


Рис. 17.10. Не давайте запросам имена, совпадающие с именами функций Excel

| Excel   | Power Query                        |
|---|------------------------------------|
| =ПРАВСИМВ(текст;[число_знаков])                           | =Text.End(text,num_chars)          |
| =ДЛСТР(текст)   | =Text.Length(text)                 |
| =НАЙТИ(искомый_текст;просматриваемый_текст;[нач_позиция]) | =Text.PositionOf(text, find_text)  |
| =ПСТР(текст;начальная_позиция;число_знаков)               | =Text.Range(text, start,num_chars) |

Рис. 17.11. Соответствие текстовых функций Excel и Power Query

Первые две функции из таблицы аналогичны только что рассмотренной Text.Start(). Использование двух последних функций требует небольшой коррекции в связи с тем, что Power Query за точку начала отсчета берет ноль. Также обратите внимание на следующее различие: аргумент *искомый текст* является первым в функции НАЙТИ(), и – вторым в функции Text.PositionOf().

### Аналог функции НАЙТИ

В файле 5 Useful Text Functions.xlsx перейдите на лист FIND. Кликните на одной из ячеек таблицы, пройдите по меню *Данные* → *Из таблицы/диапазона*. В окне редактора Power Query переименуйте запрос *pqFind*. Перейдите на вкладку *Добавление столбца* → *Настраиваемый столбец*. Назовите новый столбец *pqFind(x,"o")*. Введите формулу: =Text.PositionOf([Word],"o"). Нажмите Ok.

= Table.RenameColumns("#Добавлен пользовательский объект",{{"pqFind", "pqFind(x,"o")"}})

|   | Word       | FIND("o",x) | pqFind(x,"o") |
|---|------------|-------------|---------------|
| 1 | Football   | 2           | 1             |
| 2 | Doorknob   | 2           | 1             |
| 3 | Bookkeeper | 2           | 1             |
| 4 | Automobile | 4           | 3             |
| 5 | Mountain   | 2           | 1             |
| 6 | Truck      | Error       | -1            |
| 7 | Car        | Error       | -1            |

Рис. 17.12. Результат не вполне согласуется с Excel

Возвращаемые значения, следуют базовому правилу. В первой строке буква F идет под номером 0. Измените формулу, добавив 1: =Text.PositionOf([Word],"o")+1.

|   | A <sup>B</sup> <sub>C</sub> Word | ABC <sub>123</sub> FIND("o",x) | ABC <sub>123</sub> pqFind(x,"o") |
|---|----------------------------------|--------------------------------|----------------------------------|
| 1 | Football                         | 2                              | 2                                |
| 2 | Doorknob                         | 2                              | 2                                |
| 3 | Bookkeeper                       | 2                              | 2                                |
| 4 | Automobile                       | 4                              | 4                                |
| 5 | Mountain                         | 2                              | 2                                |
| 6 | Truck                            | Error                          | 0                                |
| 7 | Car                              | Error                          | 0                                |

Рис. 17.13. Есть совпадение с Excel, а вместо ошибок выводится значение ноль

### Аналог функции ПСТР

В файле 5 Useful Text Functions.xlsx перейдите на лист *MID*. Кликните на одной из ячеек таблицы, пройдите по меню *Данные* → *Из таблицы/диапазона*. В окне редактора Power Query переименуйте запрос *pqMid*. Перейдите на вкладку *Добавление столбца* → *Настраиваемый столбец*. Назовите новый столбец *pqMid(x,5,4)*. Введите формулу: `=Text.Range([Word],5,4)`. Нажмите *Ок*. Результат не соответствует ожиданиям:

```
= Table.AddColumn("#Измененный тип", "pqMid(x,5,4)", each Text.Range([Word],5,4))
```

|   | A <sup>B</sup> <sub>C</sub> Word | A <sup>B</sup> <sub>C</sub> MID(5,4) | ABC <sub>123</sub> pqMid(x,5,4) |
|---|----------------------------------|--------------------------------------|---------------------------------|
| 1 | Football                         | ball                                 | Error                           |
| 2 | Doorknob                         | knob                                 | Error                           |
| 3 | Bookkeeper                       | keep                                 | eepe                            |
| 4 | Automobile                       | mobi                                 | obil                            |
| 5 | Mountain                         | tain                                 | Error                           |
| 6 | Truck                            | k                                    | Error                           |
| 7 | Car                              |                                      | Error                           |

Рис. 17.14. Несколько результатов адекватно, но почему не все?

Это немного удивляет. Вы ожидали, что результат не будет соответствовать Excel. Но что число ошибок будет таким большим!? Для начала исправим положение начального символа. На примере *Bookkeeper* вы ожидали увидеть *keep*, а появился *eepe*. Поскольку первый символ в слове имеет номер ноль, вам нужно исправить формулу на `=Text.Range([Word],5-1,4)`.

```
= Table.AddColumn("#Измененный тип", "pqMid(x,5,4)", each Text.Range([Word],4,4))
```

|   | A <sup>B</sup> <sub>C</sub> Word | A <sup>B</sup> <sub>C</sub> MID(5,4) | ABC <sub>123</sub> pqMid(x,5,4) |
|---|----------------------------------|--------------------------------------|---------------------------------|
| 1 | Football                         | ball                                 | ball                            |
| 2 | Doorknob                         | knob                                 | knob                            |
| 3 | Bookkeeper                       | keep                                 | keep                            |
| 4 | Automobile                       | mobi                                 | mobi                            |
| 5 | Mountain                         | tain                                 | tain                            |
| 6 | Truck                            | k                                    | Error                           |
| 7 | Car                              |                                      | Error                           |

Рис. 17.15. Уже лучше, но всё еще ошибки в двух последних строках

Одна из замечательных особенностей функции *Mid* (ПСТР) Excel заключается в том, что вас не волнует, сколько символов осталось в текстовой строке. Если конечный параметр больше, чем количество оставшихся символов, он просто вернет все оставшиеся символы. Не таков Power Query. Вам нужно дополнить формулу проверкой: вы хотите вернуть четыре символа или меньше, до конца текстовой строки. Для этих целей подойдет функция *List.Min* (подробнее о ней вы узнаете из главы 20). Вместо того, чтобы пытаться встроить эту функцию в формулу столбца *pqMid(x,5,4)*, создайте еще один пользовательский столбец с формулой `=List.Min({Text.Length([Word])-(5-1),4})`.

= Table.AddColumn( "Пользовательская", each List.Min({Text.Length([Word])-(5-1),4}))

|   | Word       | MID(5,4) | pqMid(x,5,4) | Пользовательская |
|---|------------|----------|--------------|------------------|
| 1 | Football   | ball     | ball         | 4                |
| 2 | Doorknob   | knob     | knob         | 4                |
| 3 | Bookkeeper | keep     | keep         | 4                |
| 4 | Automobile | mobi     | mobi         | 4                |
| 5 | Mountain   | tain     | tain         | 4                |
| 6 | Truck      | k        | Error        | 1                |
| 7 | Car        |          | Error        | -1               |

Рис. 17.16. В отдельном столбце определено количество оставшихся символов

Несколько слов о том, как работает формула:

- Text.Length([Word])-(5-1) подсчитывает длину слова в столбце *Word* и вычитает начальную позицию. Вы использовали выражение (5-1), чтобы подчеркнуть, что хотели взять пятый символ, но исправили формулу для базы 0 (можно использовать и 4).
- Последняя четверка в формуле – максимальное количество символов, которые вы хотите вернуть
- Для того, чтобы использовать их в функции List.Min() они должны быть окружены фигурными скобками и разделены запятыми.

Теперь вы можете отредактировать формулу в столбце *pqMid(x,5,4)* =Text.Range([Word],5-1, List.Min({Text.Length([Word])-(5-1),4}))

|   | Word       | MID(5,4) | pqMid(x,5,4) | Пользовательская |
|---|------------|----------|--------------|------------------|
| 1 | Football   | ball     | ball         | 4                |
| 2 | Doorknob   | knob     | knob         | 4                |
| 3 | Bookkeeper | keep     | keep         | 4                |
| 4 | Automobile | mobi     | mobi         | 4                |
| 5 | Mountain   | tain     | tain         | 4                |
| 6 | Truck      | k        | k            | 1                |
| 7 | Car        |          | Error        | -1               |

Рис. 17.17. Всё верно, кроме последней строки

Теперь вы можете удалить вспомогательный столбец *Пользовательская* и загрузить запрос в *Таблицу* на лист Excel. А как же ошибка в последней строке. Не страшно. Потому что ошибки в Power Query будут показываться в Excel, как пустые ячейки:

|   | A          | B        | C            |
|---|------------|----------|--------------|
| 1 | Word       | MID(5,4) | pqMid(x,5,4) |
| 2 | Football   | ball     | ball         |
| 3 | Doorknob   | knob     | knob         |
| 4 | Bookkeeper | keep     | keep         |
| 5 | Automobile | mobi     | mobi         |
| 6 | Mountain   | tain     | tain         |
| 7 | Truck      | k        | k            |
| 8 | Car        |          |              |

Рис. 17.18. Ошибки исчезают при загрузке в *Таблицу*

Очень жаль, что не все функции в Power Query эквивалентны функциям в Excel. Особенно учитывая, что Power Query – это инструмент для пользователей Excel. Надеемся, что это изменится в будущих версиях. Мы хотели бы видеть новую библиотеку функций Power Query, которые позволяют переносить существующие знания из Excel в Power Query без необходимости изучать новые функции и новый синтаксис.