Advanced Power BI (Data)

Sample manual - first two chapters

Wise Owl Training

TABLE OF CONTENTS (1 of 4)

1	MANIPULATING COLUMNS	Page
1.1	Splitting Columns	6
	Accessing Split Column Menu	6
	Splitting by Delimiter	7
	Splitting by Number of Characters	8
	Splitting by Positions	9
	Splitting by Change in Case or Character Type	9
	Splitting into Rows	10
	Retaining Quotation Marks	11
1.2	Merging Columns	12
1.3	Extracting Data	13
	Replacing or Adding Columns	13
	An Example	13
	The Possible Options	14

2	CREATING COLUMNS	Page
2.1	Ways to Create New Columns	15
2.2	Columns from Examples	16
	Step 1 – Start the Feature	16
	Step 2 – Show some Examples	17
	Step 3 – Confirm the Formula	17
	Step 4 – Review your Formula	17
	Step 5 – Understanding your Formula	18
2.3	Built-up Columns	19
2.4	Custom Columns in M	20
	M Prefixes	20
	Our Example – Elapsed Days	20
	Creating a Custom Column	21
2.5	Special Case - Last Refresh Date/Time	22
2.6	Conditional Columns	23
2.7	Indexing Columns	24

3	COMBINING QUERIES	Page
3.1	Loading Multiple Files from a Folder	25
	Setting the Folder	25
	Combining the Files	26
	Setting the Template for Import	26
	Tidying Up the Results	27
	An Outline of How it Works	27
3.2	Appending Queries	28
	Combining Queries using Append	28
	Tidying up your Results	29
3.3	Merging Data	30
	Loading the Tables to Merge	30
	Merging the Data	31
	Flattening the Combined Table	32
	Final Touches	32
3.4	Fuzzy Merging	32
	Setting a Similarity Threshold	34
	Enforcing Case Sensitivity	34
	Using a Transformation Table	34

4	MANIPULATION TRANSFORMS	Page
4.1	Unpivoting Data	35
	Loading the Original Data	35
	Unpivoting the Data	36
	Renaming your Columns	36
4.2	Grouping Data	37
	Optional Step – Remove Excess Columns	37
	Starting Grouping	38
	Specifying How to Group Data	38
4.3	Pivoting Columns	39
	Remember Matrices	39
	Pivoting: Creating the Necessary New Columns	39
	Pivoting by Columns	40
	Using our Pivoted Columns	40
4.4	Transposing Data: Replacing Text	41
4.5	Transposing Data: Joining Rows Together	42
	Step 1 – Transposing the Data	42
	Step 2 – Reversing your Columns	42
	Step 3 – Merging your Columns	43
	Step 4 – Transposing Back	43

© Wise Owl Business Solutions Ltd

TABLE OF CONTENTS (2 of 4)

5	NORMALISING DATA	Page
5.1	Our Example	44
5.2	Extracting Unique Lists	45
	Duplicating the Original Table	45
	Creating a Unique List of Directors/Genres	45
5.3	Numbering the Lists	46
5.4	Storing Id Numbers not Names	47
	Merging the ID Column	47
	Expanding the Merged Columns	48
	Removing Unnecessary Additional Columns	48
	Making Sure the Right Table Loads	48
	The Final Relationships	49
	Finishing your Data Model	49

6	DATA PROFILING	Page
6.1	Showing Column Profiles	50
	Showing Pop-up Profile Data	51
	Grouping Profiled Data	51

7	FILTERING QUERIES	Page
7.1	Removing Nulls	52
7.2	Filtering Dates	53
	Previous or Next Time Periods	53
	Matching the Earliest/Latest Date	54
	Specific Time Periods	54

8	ERROR-HANDLING	Page
8.1	Default Treatment of Errors	55
	Errors in the First 1000 Rows	55
	Errors in Subsequent Rows	56
8.2	Fixing Errors in a Separate Table	57
	Step 1 – Duplicate the Table	57
	Step 2 – Remove the Errors from One Table	57
	Step 3 – Correct the Errors in the Second Table	58
	Step 4 - Merging the Data Back Together	58
8.3	Fixing Errors In Situ	59
	Replacing Errors	59
	Handling with a Custom Column	59

9	DATAFLOWS	Page
9.1	Creating Dataflows	60
	Step 1 – Start the Dataflow	60
	Step 2 - Create your Dataflow's Transforms	61
	Step 3 – Save your Dataflow	62
9.2	Basing Reports on Dataflows	63

10	ROW-LEVEL SECURITY	Page
10.1	Overview of Row-Level Security	64
	Limitations and Constraints	64
10.2	Roles in Power BI Desktop	64
	Creating Roles	65
	Testing Roles	66
10.3	Roles in Power BI Service	67
	Configuring the App Workspace	67
	Assigning Users to Roles	68
	Testing your Roles (or Users)	69
10.4	Dynamic Security	70
	Creating a Permissions Table	70
	Creating a Role	71
10.5	Creating Groups of Users	72
	Creating the Tables	72
	Linking the Tables	73
	Creating the Role	73

11	SWITCHING DATA SOURCES	Page
11.1	Our Example	74
11.2	Changing the Data Source Manually	75
11.3	Switching the Data Source using M	76
	Showing the Advanced Editor	76
	The Excel Query	76
	Creating a SQL Server Version of our Table	77
	Switching the Data Source	77
	Resolving Problems	78
	Viewing the Results	78

© Wise Owl Business Solutions Ltd

TABLE OF CONTENTS (3 of 4)

12	SIMPLE PARAMETERS	Page
12.1	Parameters, Slicers and Filters	79
	Parameters Needed for this Example	79
12.2	Working with Parameters	80
	Accessing Parameters	80
	Creating Parameters	80
	Managing Parameters	81
	Loading Parameters into Power BI	81
12.3	Entering Values for Parameters	81
	Entering Parameter Values in Query Editor	82
	Entering Parameter Values in Power BI Desktop	82
12.4	Using Parameters	82
	Filtering by Parameter Values	83
	the Parameter Values Chosen	84
12.5	Using Parameters with Templates	85
	Creating a Template	85
	Opening (Using) a Template	86
12.6	Query Dependencies	87

13	DROPDOWN PARAMETERS	Page
13.1	Our Example	88
13.2	Creating a Simple List Parameter	89
13.3	Using List Queries for Parameters	90
	Step 1 – Create a Query Giving a Single Column of Data	90
	Step 2 – Make this a List Query	90
	Step 3 – Creating the Dropdown Parameter	91
13.4	Referencing the Dropdown Parameter	92

14	PARAMETERS AND STORED PROCEDURES	Page
14.1	Overview	93
14.2	Loading Data from a Stored Procedure	94
14.3	Making Parameters Dynamic	95
	Creating the Parameters	95
	Referencing the Parameters in the Advanced Editor	95
	Editing Parameter Values	97
	Displaying the Parameters in your Report	97

15	DYNAMIC CONNECTIONS	Page
15.1	SQL Server Dynamic Connections	98
	First Enable Parameterisation of Data Sources	98
	Creating a Parameter	99
	Referencing your Parameter	99
	Changing the Server or Database	99
	The Query Editor Step Created	100
15.2	Dynamic Connections to Excel	100
	Step 1 – Create a Connection (Get Data)	101
	Step 2 – Create a Parameter	102
	Step 3 – Parameterising the Connection Used	102
	Step 4 - Changing the Worksheet Used	102

16	CUSTOM FUNCTIONS	Page
16.1	Overview	104
16.2	Example – Splitting Postcodes	105
	Step 1 – Getting the Initial Data	105
	Step 2 - Creating a Parameter	105
	Step 3 - Duplicate your Parameter	105
	Step 4 – Create a Query Based on the Parameter Value	106
	Step 5 - Turning this One-Off Query into a Function	106
	Step 6 - Understanding the Resulting List of Queries	107
	Step 7 – Test your Function	107
	Step 8 – Invoke your Function for Each Table Row	108
	Step 9 – Expand the Results	108
16.3	Example - Combining Selected Worksheets	109
	Step 1 - Set Up your Base Query and Parameter	109
	Step 2 - Link the Query to the Parameter	109
	Step 3 - Create your Custom Function	110
	Step 4 – Create a Table of Years	111
	Step 5 - Invoke your Custom Function	112
	Step 6 – Expand the Results	112
16.4	Combining Tables from Web Pages	112
	Step 1 – Get a List of Pages to Visit	113
	Step 2 – Create a Parameter	114
	Step 3 – Load the Data for One Page	114
	Step 4 – Parameterise your Query	115
	Step 5 – Create a Custom Function	115
	Step 6 – Invoking the Custom Function	116
	Step 7 – Expand your Results	116

TABLE OF CONTENTS (4 of 4)

17	HTML AND JSON	Page
17.1	Our Example	117
17.2	Extracting Text from HTML	118
	Starting our Example	118
	Creating an HTML.Table Function	118
	Creating a Column to Parse HTML	119
	Expanding the Table Returned	119
	Getting the Readable Text Only	120
17.3	Transforming JSON	121
	Parsing JSON (or XML)	121
	Expanding the Resulting List as Rows	122
	Expanding the Resulting List as a Single String	122
	Extracting the List of Topics	123
	Creating the Relationship	123
17.4	Getting External Links	124
	Preparing the Way	124
	Loading HTML from One Website	124
	Turning your HTML into a Table	125
	Extracting Links from HTML	125
	Parametrise your Query	126
	Creating a Custom Function	126
	Invoking this Custom Function	126

18	USING APIS	Page
18.1	Introducing APIs	127
18.2	Some Simple Examples	127
	Random Dog Pictures	128
	XKCD Cartoons	128
18.3	A Digression: Displaying Images Properly	130
18.4	Case Study – Exchange Rates	130
	Finding an API	131
	Getting the URL Needed	131
	Turning the Results into a Usable Table	131
	Adding Country Codes	132
	Getting the UK Exchange Rate	133
18.5	Multiple Table APIs	134
	Importing individual Tables	134
	Simplifying Column Titles	135
	The Need for a Bridging Table	135
	Starting the Bridging Table	136
	Removing Non-Essential Columns	136
	Expanding the List of Films	136
	Creating the Bridging Table	137
	Changing the Relationship Directions	137
	Using your Tables	137

19	DATA INSIGHTS	Page
19.1	Overview	138
	Three Warnings	138
19.2	Text Analytics	139
	Detecting Language	139
	Extracting Key Phrases	139
	Scoring Sentiment	140
19.3	Tagging Images	141
	Our Example	141
	Asking to Tag Images	141
	Interpreting the Results	142

20	LINKING TO POWER AUTOMATE	Page
20.1	Our Example	143
20.2	Creating the Button and Flow	144
	Step 1 – Create the Button	144
	Step 2 – Create your Flow	144
	Step 3 – Compose your Data	145
	Step 4 - Format your Table as HTML	146
	Step 5 – Send your Email	146
	Step 6 – Saving and Assigning your Flow	147
	Step 7 – Formatting your Button	147
20.3	Testing your Button	148
	Some Limitations of Running Flows from Power BI Desktop	148

21	POWER APPS IN POWER BI	Page
21.1	Overview	149
	Sharing Data Sources	149
21.2	Embedding an App	150
	Step 1 - Create a Power BI Report	150
	Step 2 – Adding an App Visual	150
	Step 3 – Creating or Choosing the App to Show	151

22	POWER BI IN POWER APPS	Page
22.1	Overview	152
22.2	Creating the Power BI Report	153
	Setting a Custom Page Size	153
	Publishing the Page	153
22.3	Creating your Dashboard	154
	Pinning a Live Page to a Dashboard	154
	Hiding the Tile's Title and Subtitle	155
22.4	Showing the Live Page in Power Apps	156

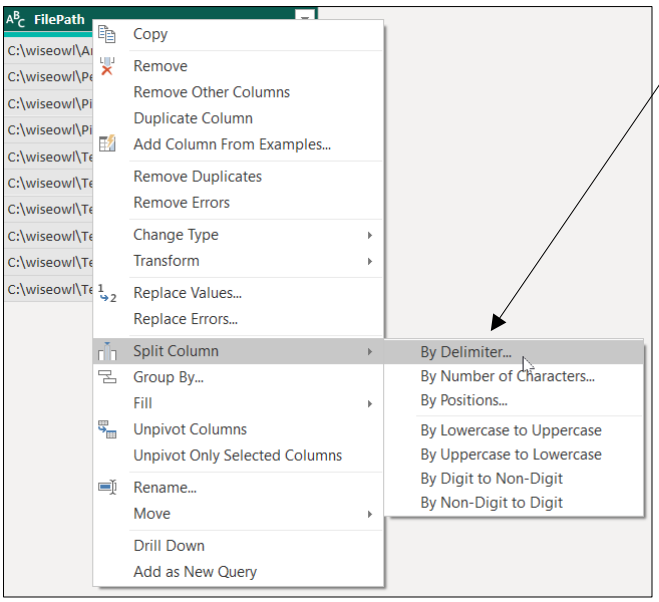
© Wise Owl Business Solutions Ltd

CHAPTER 1 - MANIPULATING COLUMNS

1.1 Splitting Columns

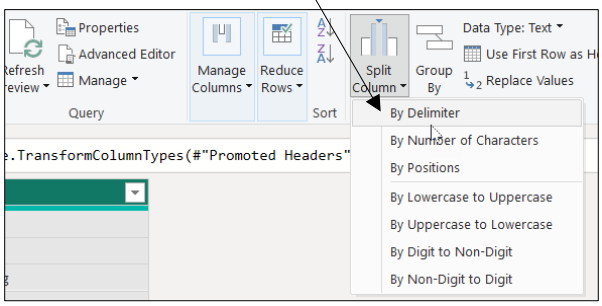
Accessing Split Column Menu

You can split columns using either the right mouse button menu or the ribbon:



Either right-click on a column that you want to split and choose one of these options …

… or select the column, then choose one of the options from the **Home** tab of the Query Editor ribbon.



The menu options are as follows:

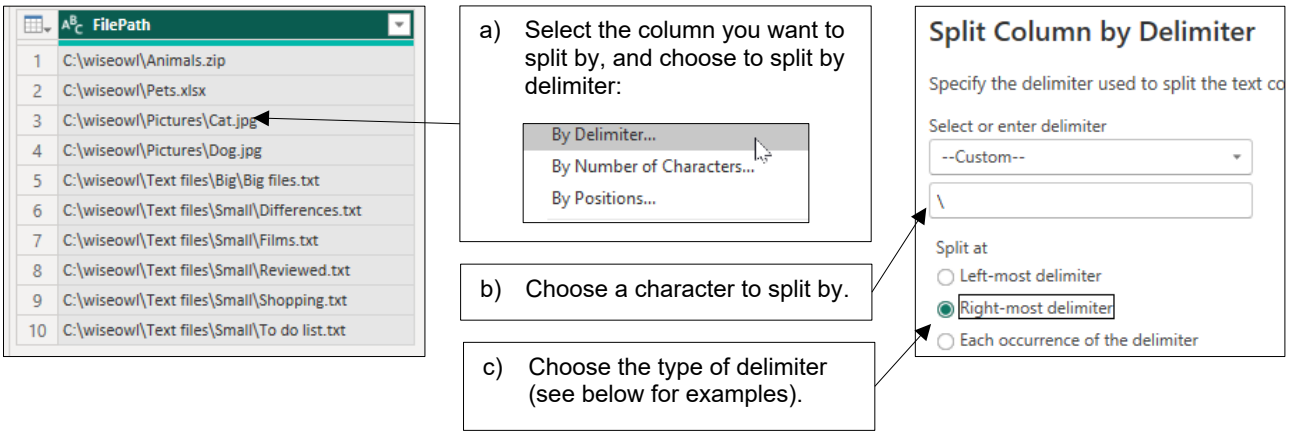
Option	What it does
By Delimiter…	Splits a string of text when Query Editor encounters a particular character.
By Number of Characters…	Splits a string of text after a given number of characters.
By Position…	Splits a string of text at certain pre-defined character positions.
By Lowercase to Uppercase *By Uppercase to Lowercase*	Splits a string of text whenever the case changes.
By Digit to Non-Digit *By Non-Digit to Digit*	Splits a string of text whenever it changes from numbers to letters or from letters to numbers.

All of these options are shown – in this order – in the following pages, along with a couple of other column splitting methods as a bonus!

© Copyright 2026

Splitting by Delimiter

For the example below, you might want to extract the file name or file path, so you'd probably split at the last `\` character:



Here are the first couple of rows you'd get for each of the 3 **Split at** options above:

Option	Results	Notes
Left-most delimiter	FilePath.1 / FilePath.2 1 C: wiseowl\Animals.zip 2 C: wiseowl\Pets.xlsx 3 C: wiseowl\Pictures\Cat.jpg	Split by the first backslash
Right-most delimiter	FilePath.1 / FilePath.2 1 C:\wiseowl Animals.zip 2 C:\wiseowl Pets.xlsx 3 C:\wiseowl\Pictures Cat.jpg	Split by the last backslash
Each occurrence of the delimiter	FilePath.1 / FilePath.2 / FilePath.3 / FilePath.4 / FilePath.5 1 C: wiseowl Animals.zip null null 2 C: wiseowl Pets.xlsx null null 3 C: wiseowl Pictures Cat.jpg null 4 C: wiseowl Pictures Dog.jpg null 5 C: wiseowl Text files Big Big files.txt	Split by every backslash

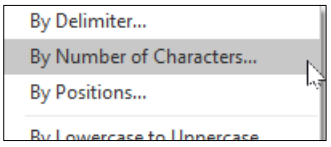
© Copyright 2026

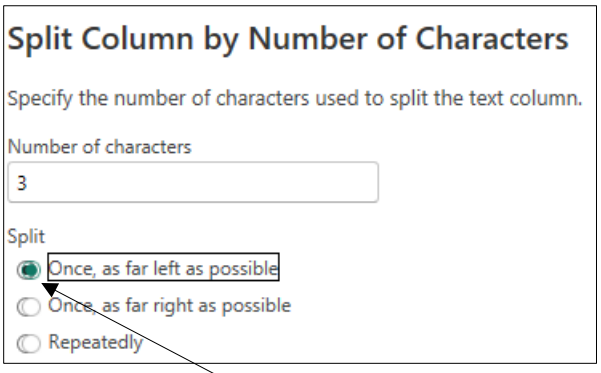
Splitting by Number of Characters

If you know how many characters you want to extract, you can use this method:

	Product Code	Sales
1	MPB-4242	24.49703694
2	HQC-0233	44.9529665
3	ZND-9682	91.64723297
4	PGL-4142	30.60766405

a) Select the column that you want to split, and choose to split it by the number of characters:





b) Choose to take the first N characters, the last N characters or a series of N-character chunks. To get the first 3 letters in this case we would choose the first option (**Once, as far left as possible**).

Product Code.1	Product Code.2
MPB	-4242
HQC	-233
ZND	-9682

c) For this example Query Editor would create these two columns, which you could then rename.

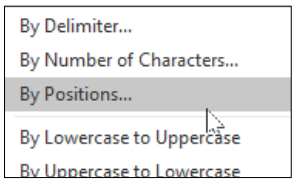
For the above example, you could instead split the product code by a delimiter (in this case, the - character) to get the letters and numbers separately.

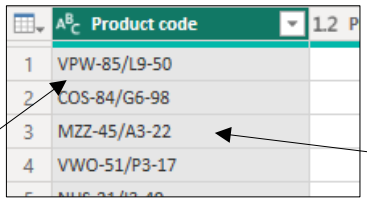
© Copyright 2026

Splitting by Positions

Use this method to extract a substring of text:

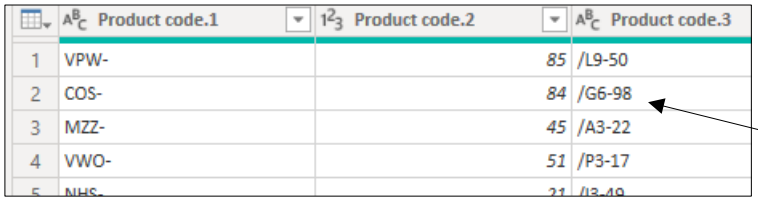
a) Choose this option to split this column by positions:





	Product code	1.2 P
1	VPW-85/L9-50	
2	COS-84/G6-98	
3	MZZ-45/A3-22	
4	VWO-51/P3-17	

b) To extract the two-digit number after the first dash from each product code you could use this transform:



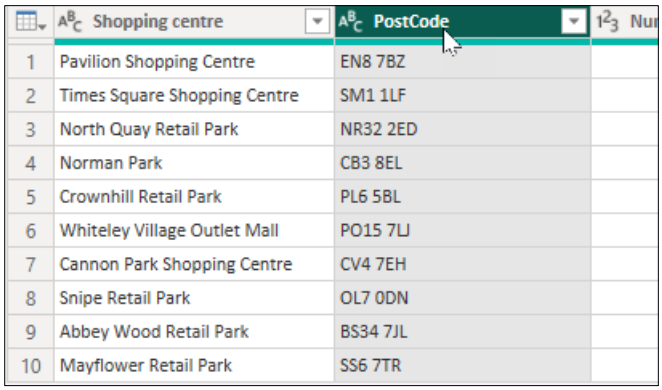
	Product code.1	Product code.2	Product code.3
1	VPW-	85	/L9-50
2	COS-	84	/G6-98
3	MZZ-	45	/A3-22
4	VWO-	51	/P3-17

c) This would give these 3 columns (everything from character 0 to 3, 4 to 5 and 6 onwards).

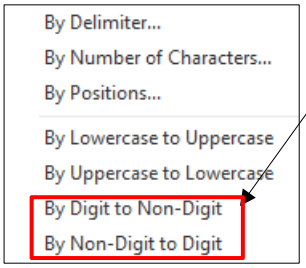
*As with most things in Power BI, characters are numbered from 0, not 1. To omit the first column you could just put **4, 6**, to start extracting characters at position 4.*

Splitting by Change in Case or Character Type

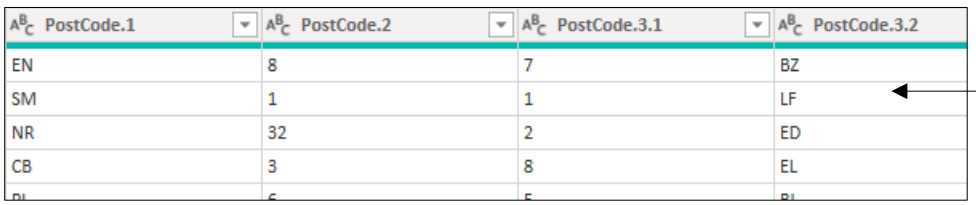
Codes are often divided into a predictable combination of letters and numbers (or of upper and lower case characters). Here's how to split up some well-behaved UK postcodes:



	Shopping centre	PostCode	Nu
1	Pavilion Shopping Centre	EN8 7BZ	
2	Times Square Shopping Centre	SM1 1LF	
3	North Quay Retail Park	NR32 2ED	
4	Norman Park	CB3 8EL	
5	Crownhill Retail Park	PL6 5BL	
6	Whiteley Village Outlet Mall	PO15 7LJ	
7	Cannon Park Shopping Centre	CV4 7EH	
8	Snipe Retail Park	OL7 0DN	
9	Abbey Wood Retail Park	BS34 7JL	
10	Mayflower Retail Park	SS6 7TR	



By repeated application of these two transforms, for example …



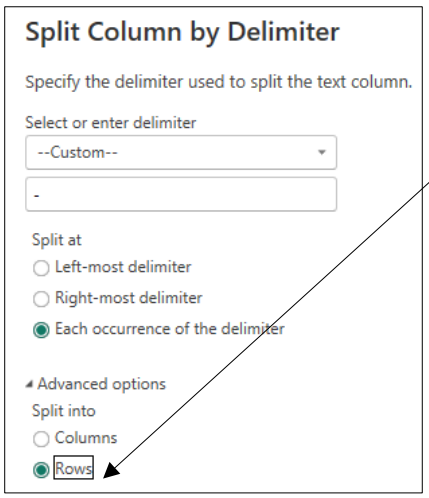
PostCode.1	PostCode.2	PostCode.3.1	PostCode.3.2
EN	8	7	BZ
SM	1	1	LF
NR	32	2	ED
CB	3	8	EL

… you could split each postcode into the letter(s) and number(s) before the space and the letter(s) and number(s) after it.

© Copyright 2026

Splitting into Rows

Sometimes you may want to create one row for each constituent part of a string of text (although it's hard to think of an example!):



If you expand the advanced options when splitting a column by delimiter and choose to generate extra rows, not columns for this example …

	Product Code	Sales
1	MPB-4242	24.49703694
2	HQC-0233	44.9529665
3	ZND-9682	91.64723297
4	PGL-4142	30.60766405

… you'll get multiple rows for each original one (notice that the values of the other fields - **Sales** in this case - will be duplicated for each of these rows, making the data hard to interpret in most cases).

	Product Code	Sales
1	MPB	24.49703694
2	4242	24.49703694
3	HQC	44.9529665
4	0233	44.9529665
5	ZND	91.64723297
6	9682	91.64723297
7	PGL	30.60766405
8	4142	30.60766405

© Copyright 2026

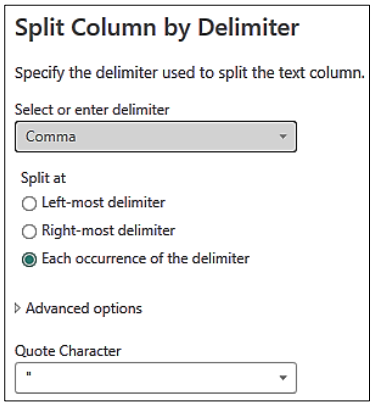
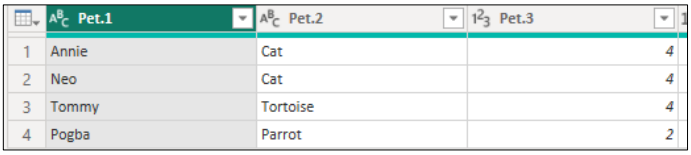
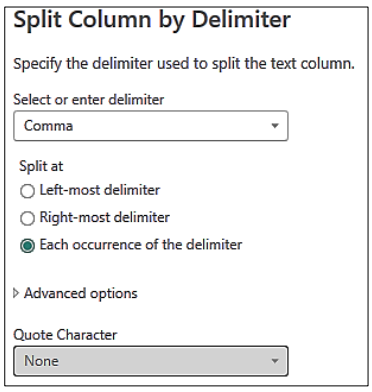
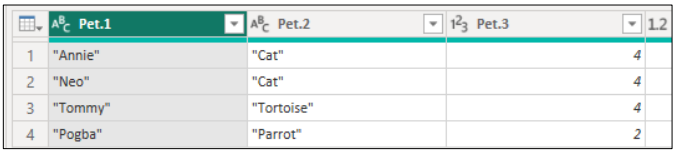
Retaining Quotation Marks

When splitting text, Power Query will automatically remove any quotation marks, but you can turn this option off as shown below.

The examples below use this as an example source query.

	Pet
1	"Annie","Cat",4,8.3
2	"Neo","Cat",4,8.5
3	"Tommy","Tortoise",4,6.2
4	"Pogba","Parrot",2,7.1

Here are the two options:

Option	Dialog box	Results for our example
Remove quote marks		
Keep quote marks		

Pet.1	Pet.2	Pet.3
1 Annie	Cat	4
2 Neo	Cat	4
3 Tommy	Tortoise	4
4 Pogba	Parrot	2

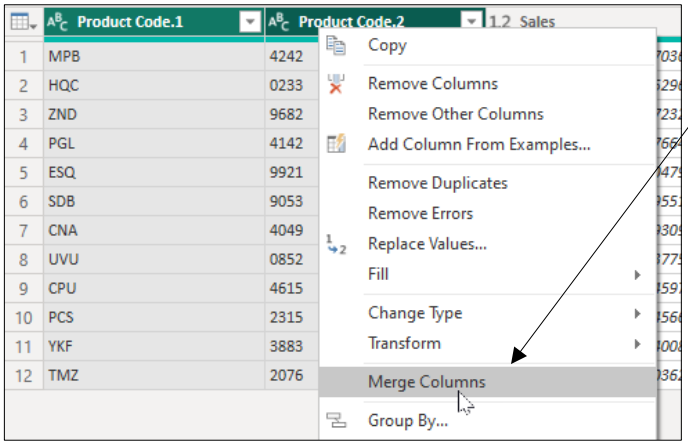
Pet.1	Pet.2	Pet.3
1 "Annie"	"Cat"	4
2 "Neo"	"Cat"	4
3 "Tommy"	"Tortoise"	4
4 "Pogba"	"Parrot"	2

There doesn't seem to be any option for automatically removing single quotation marks.

© Copyright 2026

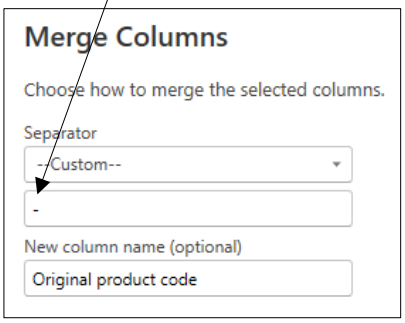
1.2 Merging Columns

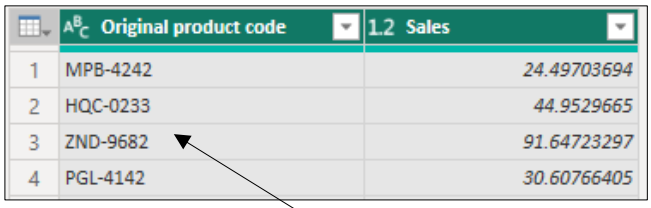
The opposite of splitting columns is *merging* them. Here's an example:



a) Suppose you want to join the parts of the product code that we split earlier in this chapter back together again! To do this, first select the columns that you want to join together, then right-click on them and choose **Merge Columns**.

b) Choose the glue you want to use to join the values together for each row (here we've gone for a dash).





	Original product code	1.2 Sales
1	MPB-4242	24.49703694
2	HQC-0233	44.9529665
3	ZND-9682	91.64723297
4	PGL-4142	30.60766405

c) Power Query has combined the columns together into a single column.

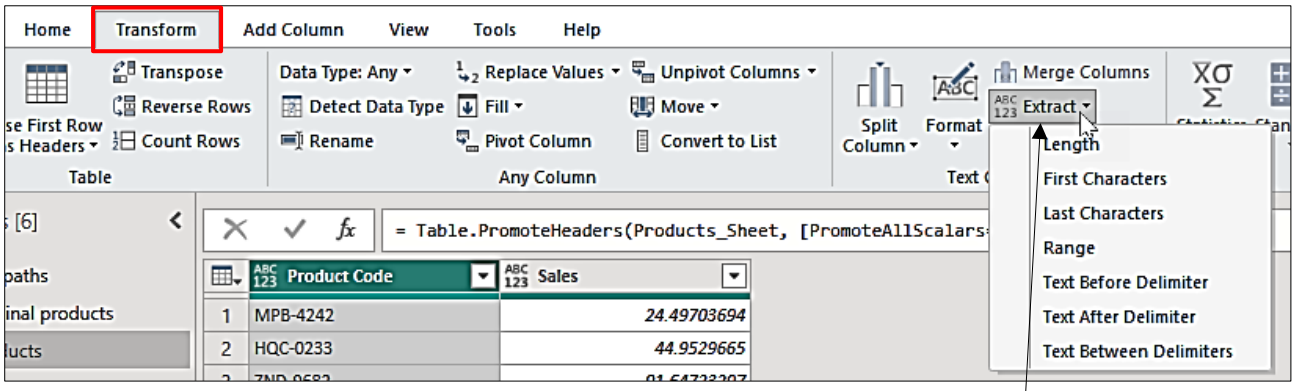
You're not limited to just two columns: you can merge as many columns as you like into a single one.

© Copyright 2026

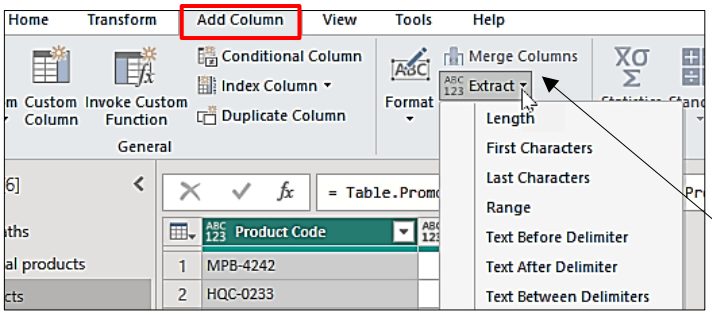
1.3 Extracting Data

Replacing or Adding Columns

There are two versions of the **Extract** tool – be careful to choose the right one for your task:



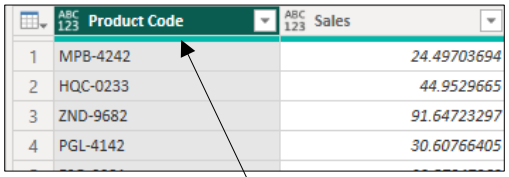
If you choose this version on the **Transform** tab of the ribbon you will replace a column …



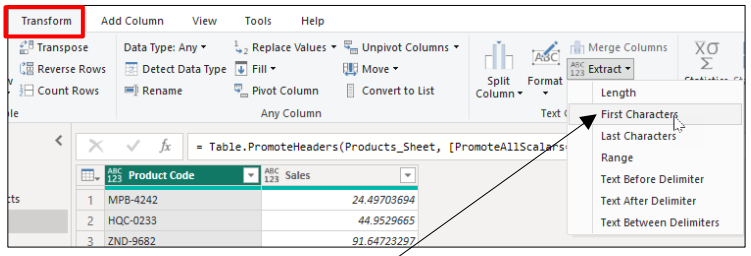
… whereas if you choose this version on the **Add Column** tab of the ribbon you will keep your existing column and add a new one to it.

An Example

The example below would extract the first 3 characters from a product code:

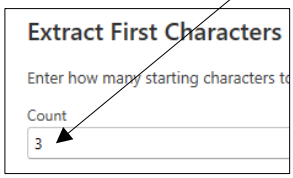


a) Select the column (or columns) from which you want to extract data.

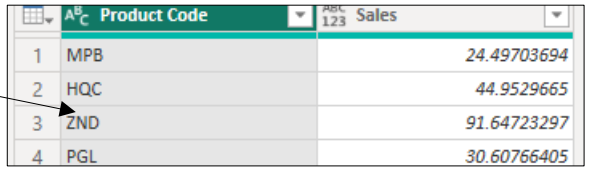


b) Here we're choosing to get the first N characters, replacing the current column.

c) Specify how many characters you want to extract (here we've gone for the first 3).



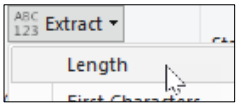
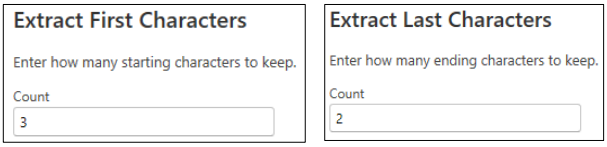
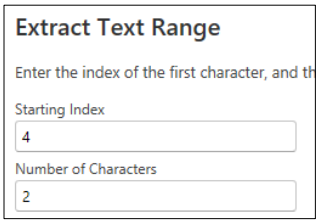
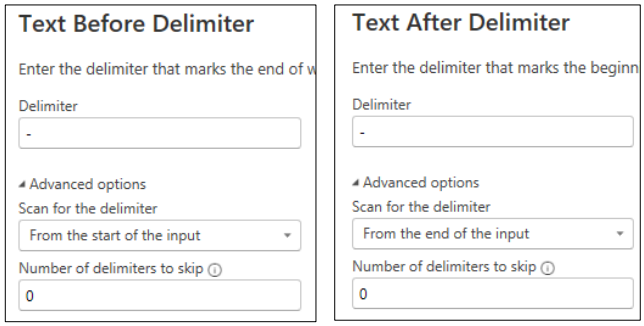
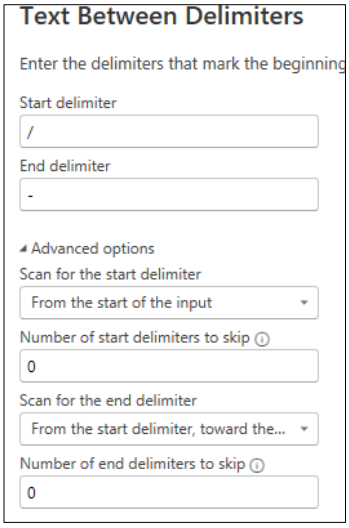
d) Power Query will replace your column with the first 3 letters for each product code.



© Copyright 2026

The Possible Options

Here's what the options would give for a field containing the product code **VPW-85/L9-50**:

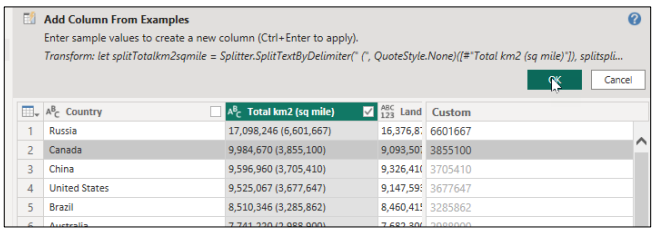
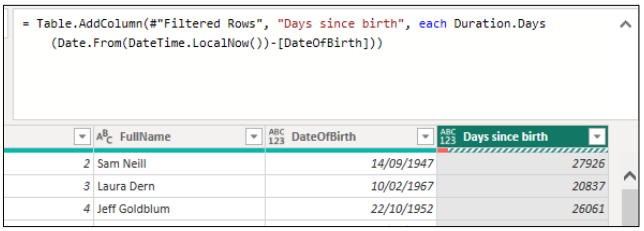
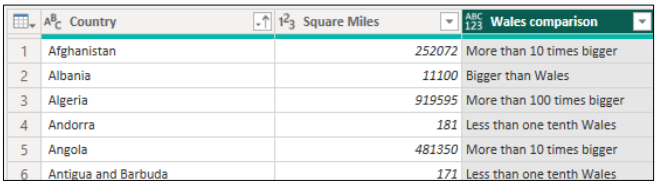
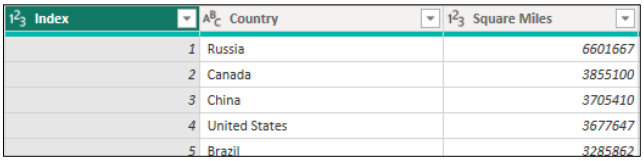
Option	Choices	Results	Notes
Length		12	This is a quick way to find how many characters there are for each value in a column.
First Characters *Last Characters*		VPW 50	The (in this case) 3 left-most characters. The (in this case) 2 right-most characters.
Range		85	The 2 characters starting at position number 4 (remember that characters are numbered from 0, not 1).
Text Before Delimiter *Text After Delimiter*		VPW 50	Notice that you can start scanning for a delimiter from the left or right with these options, and also choose not to take the first delimiter you find.
Text Between Delimiters		L9	This powerful option allows you to pick out text between any two (possibly different) characters. You can even do things like pick out the text between the second dash and fourth backslash from the end.

© Copyright 2026

CHAPTER 2 - CREATING COLUMNS

2.1 Ways to Create New Columns

There are a variety of ways of creating columns in Power Query, including the following (covered in this chapter):

Method	What it does	Example
Column from Examples	Use AI to guess what formula you want to apply to all rows of a table based on one or two sample training values.	
Built-up Columns	Use built-in Power Query transforms in sequence to produce complicated effects (you can then delete any intermediate columns created).	
Custom Columns	Return the value - for each row of a table - of an expression using the M Power Query Formula language.	
Conditional Columns	Use rules to divide rows up into a finite number of discrete categories.	
Indexing Columns	Apply numbering to the rows in a table.	

Columns that you create in Power Query will take up more space in your model (since all of the row values have to be pre-calculated when you load your data), but will run more quickly thereafter (since Power BI doesn't need to calculate DAX columns on the fly). In practice you're unlikely to notice the difference!

© Copyright 2026

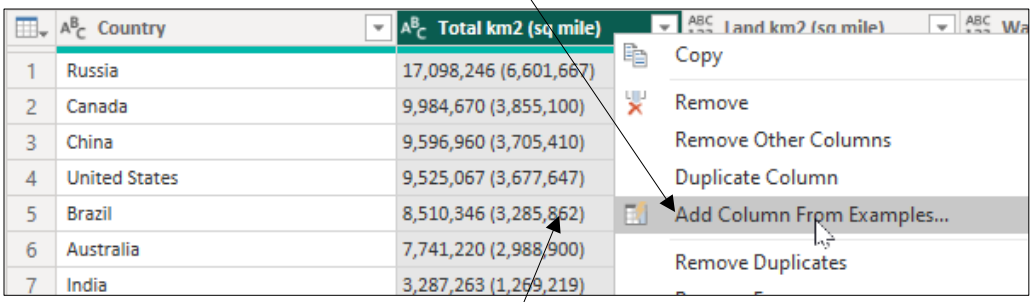
2.2 Columns from Examples

Power BI has got an excellent feature called *Columns from Examples*, which allows you to create formulae by typing in a few results. To create a column like this follow the numbered steps below.

Step 1 – Start the Feature

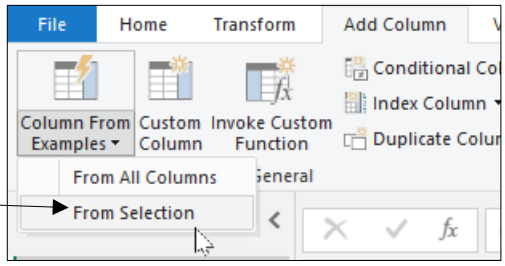
Begin by selecting the column or columns which contain the raw data from which Power Query can deduce the formula you want to apply, then right-click and choose this menu option:

For this example we want to show the total area of each country (the list comes from Wikipedia) in square miles. So for Russia this should give **6601667**, the figure in brackets.



Start by selecting the column or columns upon which Power Query should base its formula, then right-click and choose to add a column by showing some examples.

Note that you can also do this from the ribbon:

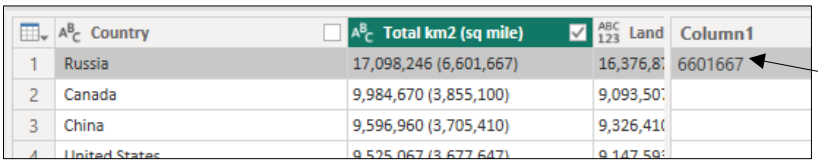


You can do the same thing by clicking on the drop-down next to the **Column From Examples** button on the **Add Column** tab of the Power Query ribbon.

© Copyright 2026

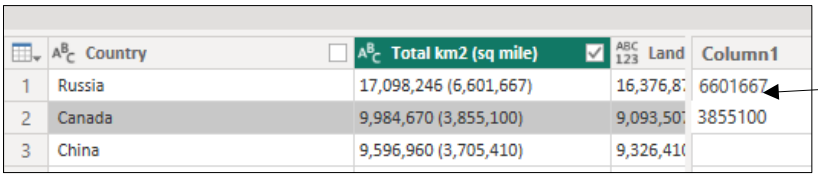
Step 2 – Show some Examples

You can now train the Power BI column formula generation algorithm:



	Country	Total km2 (sq mile)	Land	Column1
1	Russia	17,098,246 (6,601,667)	16,376,8	6601667
2	Canada	9,984,670 (3,855,100)	9,093,50	
3	China	9,596,960 (3,705,410)	9,326,41	
4	United States	9,525,067 (3,677,647)	9,147,59	

a) Type in the value you would expect a Query Editor formula to return for your first row of data, then press Ctrl + Enter.



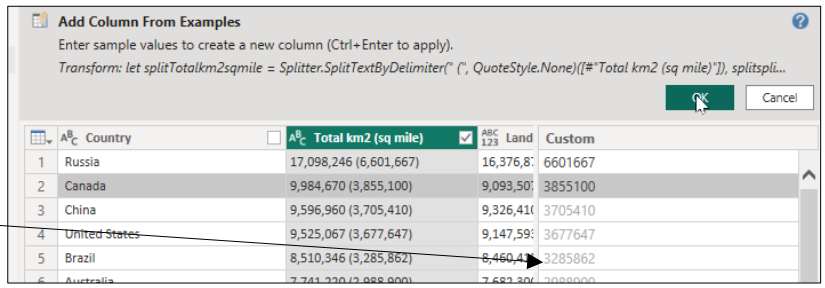
	Country	Total km2 (sq mile)	Land	Column1
1	Russia	17,098,246 (6,601,667)	16,376,8	6601667
2	Canada	9,984,670 (3,855,100)	9,093,50	3855100
3	China	9,596,960 (3,705,410)	9,326,41	

b) If you don't get any results for the first value, try a second (and even a third) until Power BI gets what you're trying to do, pressing press Ctrl + Enter each time.

Step 3 – Confirm the Formula

If you're happy with what Power BI has created, select OK; otherwise, choose Cancel:

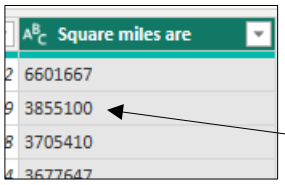
Incredibly, Power BI seems to have worked out exactly what you want to do.



	Country	Total km2 (sq mile)	Land	Custom
1	Russia	17,098,246 (6,601,667)	16,376,8	6601667
2	Canada	9,984,670 (3,855,100)	9,093,50	3855100
3	China	9,596,960 (3,705,410)	9,326,41	3705410
4	United States	9,525,067 (3,677,647)	9,147,59	3677647
5	Brazil	8,510,346 (3,285,862)	8,460,4	3285862
6	Australia	7,741,220 (2,988,900)	7,683,30	2988900

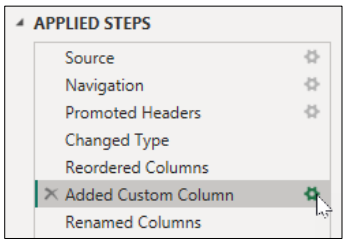
Step 4 – Review your Formula

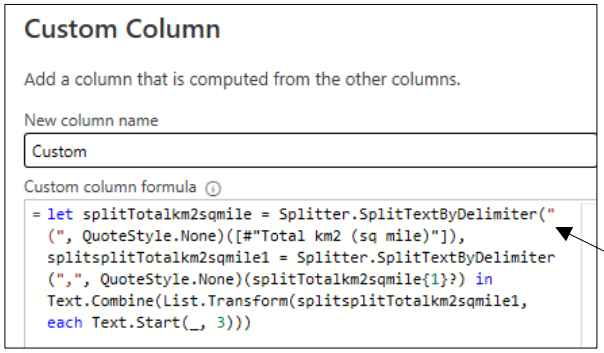
You can now have a look at the M formula created for your new column:



	Square miles are
2	6601667
9	3855100
8	3705410
4	3677647

a) Here we've renamed the column too. You can view the details of the transform step by clicking on the gear icon next to it:





b) This would show this formula, which is explained overleaf!)

© Copyright 2026

Step 5 – Understanding your Formula

The big disadvantage of creating columns by example is that you may find it hard to understand (and amend) the resulting formula.

```
= let splitTotalkm2sqmile = Splitter.SplitTextByDelimiter("
  (", QuoteStyle.None)([#"Total km2 (sq mile)"]),
  splitsplitTotalkm2sqmile1 = Splitter.SplitTextByDelimiter
  (",", QuoteStyle.None)(splitTotalkm2sqmile{1}?) in
  Text.Combine(List.Transform(splitsplitTotalkm2sqmile1,
  each Text.Start(_, 3)))
```

For our example, here's our formula.

Here's a rough breakdown of what this does:

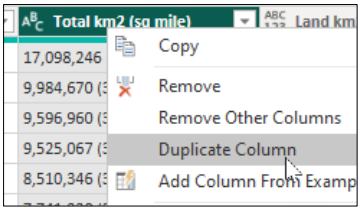
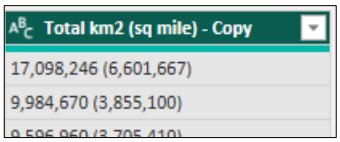
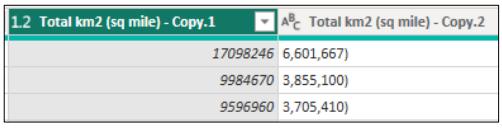
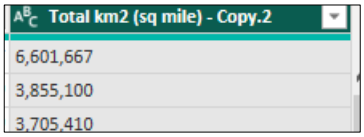
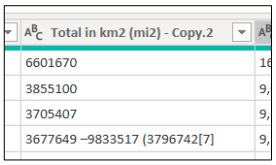
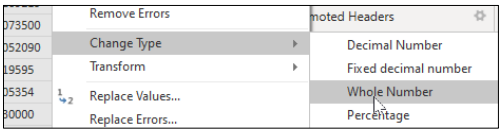
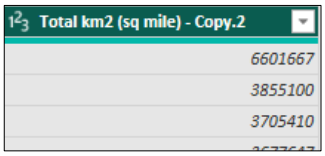
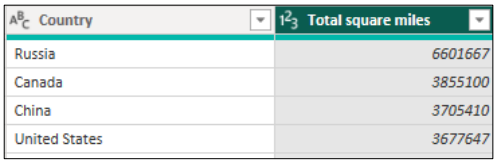
Stage	What it does	Returns for row 1
Before any formulae applied		**17,098,246 (6,601,667)**
`let splitTotalkm2sqmile = Splitter.SplitTextByDelimiter(" (", QuoteStyle.None)([#"Total km2 (sq mile)"])`	Creates a variable with a long unwieldy name (**splitTotalkm2sqmile**) to take the column called **Total km2 (sq mile)** and split it at the first `(` character. This will return a list of items.	A list containing two items: **17,098,246 (** and **6,601,667)**.
`splitsplitTotalkm2sqmile1 = Splitter.SplitTextByDelimiter(",", QuoteStyle.None)(splitTotalkm2sqmile{1}?)`	Takes the second element returned in this list of items (list elements are numbered from 0, so [1] means the second element), which is the bit after the `(` character, and splits this by the `,` character.	A list containing the items **6**, **601** and **667)**.
`let … in Text.Combine(List.Transform(splitsplitTotalkm2sqmile1, each Text.Start(_, 3)))`	Takes this list of numbers, and for each one picks out the first 3 characters. Combines this list of miniature strings of text into a single string.	**6601667**

The above shows two things: that the M language won't be that intuitive to learn, and that (like all Microsoft wizards) the formulae created for columns by example are not always written in the simplest way!

© Copyright 2026

2.3 Built-up Columns

An alternative to using columns by example is to create a new column one bit at a time. For our example (getting each country's total land area in square miles) this could be as follows:

Transform	How	Result
Duplicate column	Right-click on the total land area column and choose **Duplicate Column** to take a copy of it, so we don't lose the original. 	
Split by bracket	Right-click on this duplicated column and choose **Split Column** > **By Delimiter…**, then choose a (as your delimiter.	
Remove the end bracket	Right-click on the second column generated and choose **Replace Values…**, then replace every) with nothing.	
Remove the commas	Repeat the above step, replacing each , with nothing.	
Change the value to a number	Right-click on the column and choose to change the type to a number: 	
Tidy up the columns	Delete any intermediate columns generated, reorder your columns and rename this one.	

The great advantage of this approach is that you can understand how you got from A to B, reproduce the stages and even modify them should you require.

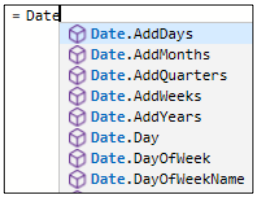
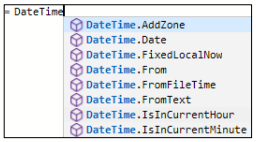
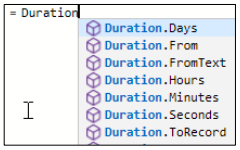
© Copyright 2026

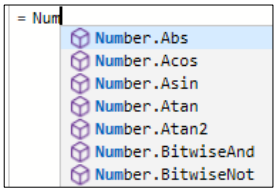
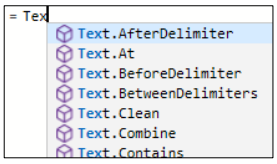
2.4 Custom Columns in M

Even if you don't know the M formula language you can sometimes guess your way round using it to create formulae for new columns based on existing ones.

M Prefixes

Here are some good prefixes to try in Intellisense:

Prefix	What it gives	Example
Date	Functions to do with dates	
DateTime	Functions to do with dates/times	
Duration	Functions giving the duration of time	

Prefix	What it gives	Example
Number	Functions giving things you can do to numbers	
Text	Functions to do with text	

Our Example – Elapsed Days

Suppose you have a table of actors, and want to find out how many days have elapsed since each was born:

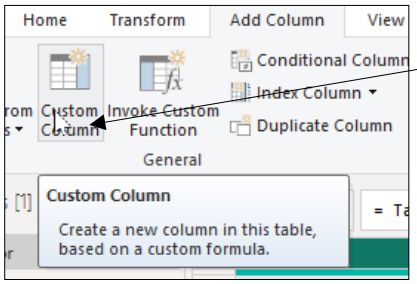
DateOfBirth	Days since birth
14/09/1947	27926
10/02/1967	20837
22/10/1952	26061
29/08/1923	36708

To understand how to create this formula, ask the question a different way: you want to take the duration in days of the period between the **[DateOfBirth]** column in this table and the current local date/time, expressed as a date.

© Copyright 2026

Creating a Custom Column

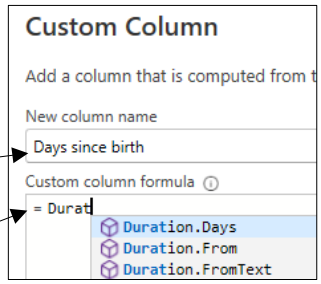
Here's how to create a column like this:

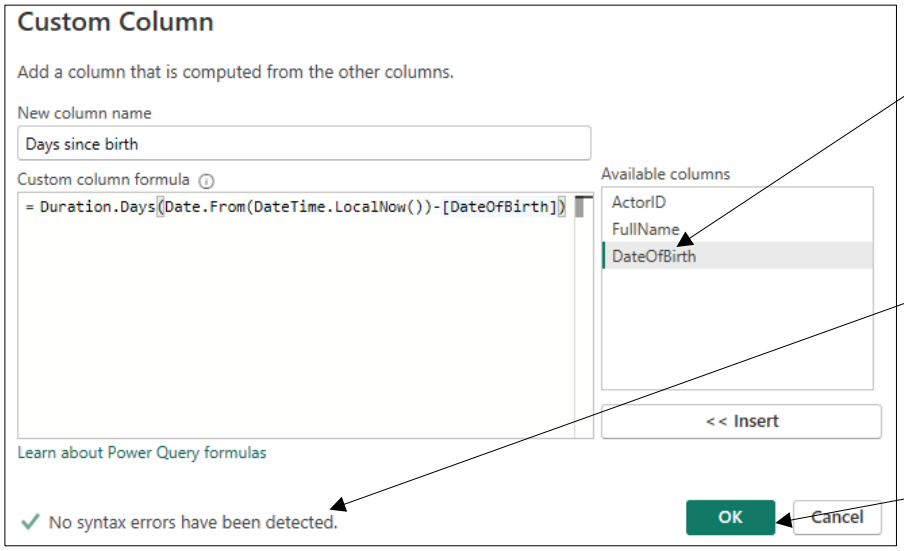


a) Click on tis icon on the **Add Column** menu to add a custom column to your query.

b) Give your new column a name.

c) Begin to create a formula.





d) Double-click on any column to insert it into your formula at any stage.

e) You won't be able to continue until you see this message confirming that your syntax is plausible.

f) When you've finished your formula, press OK to confirm it.

```
= Table.AddColumn(#"Filtered Rows", "Days since birth", each Duration.Days(Date.From(DateTime.LocalNow())-[DateOfBirth]))
```

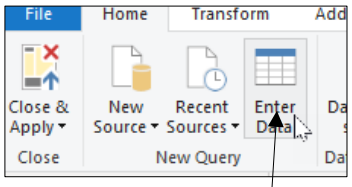
g) Power Query will incorporate this in a new query step (in this case showing for each row in the given table the number of days between each actor's birth date and today's date).

What the above example shows is that it's not easy to guess your way round the M language. Who would have thought it would be so hard to get at today's date, for example?

© Copyright 2026

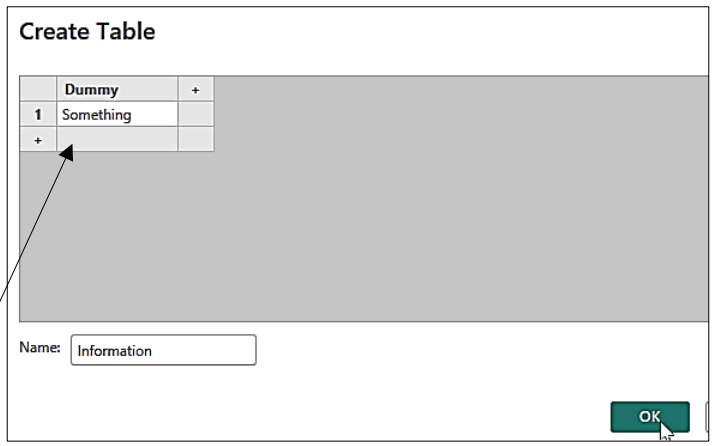
2.5 Special Case - Last Refresh Date/Time

This pages shows a way to show the last date/time when a report was refreshed:

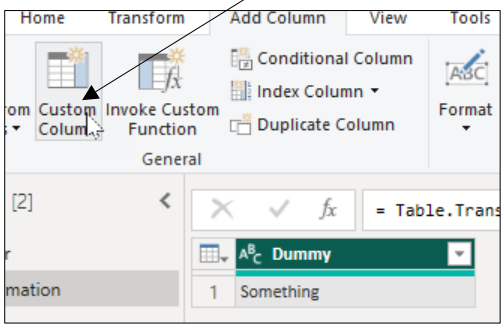


a) In Power Query, choose to create a new table by entering data for it.

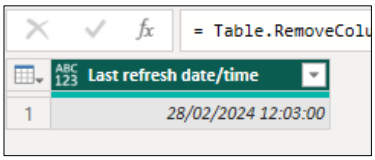
b) Make sure your table has one row in, since you want it to contain one bit of information only (it doesn't matter what this column is called, nor what it contains).

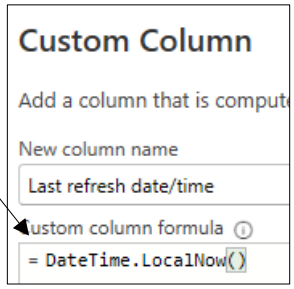


c) Choose to add a custom column from the **Add Column** tab of the ribbon.

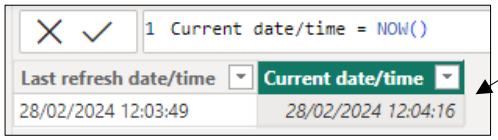


d) Create an expression giving the current date/time. You can then delete the original column to give your final table:

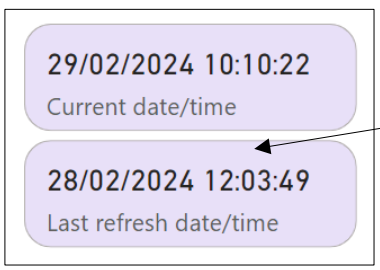




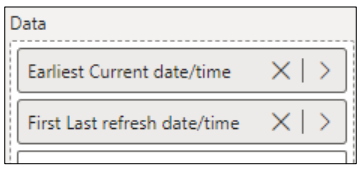
After loading your table, you could create a calculated column in it to show the current date and time, using the DAX **NOW()** function, then display both bits of information side by side:



You could add a column to your table in Power BI to show the latest date/time.



The card on the top is showing the date/time now; the one on the bottom is showing the date/time when you last refreshed your data (both shown below before we renamed them to remove the messy statistic prefix):



© Copyright 2026

2.6 Conditional Columns

You can use conditional columns to divide data up into discrete bands. In the following example we express each country's land area in units of the size of Wales (about 8,023 square miles).

In an ideal world the provider of your data would add this column into the underlying data source, to prevent you having to calculate it as you load the data.

To create a conditional column like this:

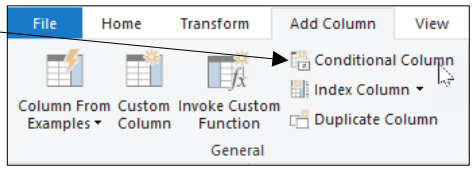
a) On the **Add Column** tab select to create a conditional column.

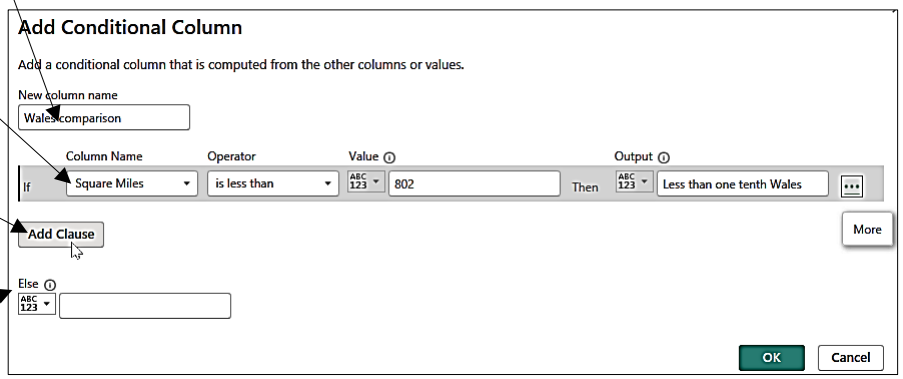
b) Give your new column a name (here we've called it **Wales comparison**).

c) Create your first condition.

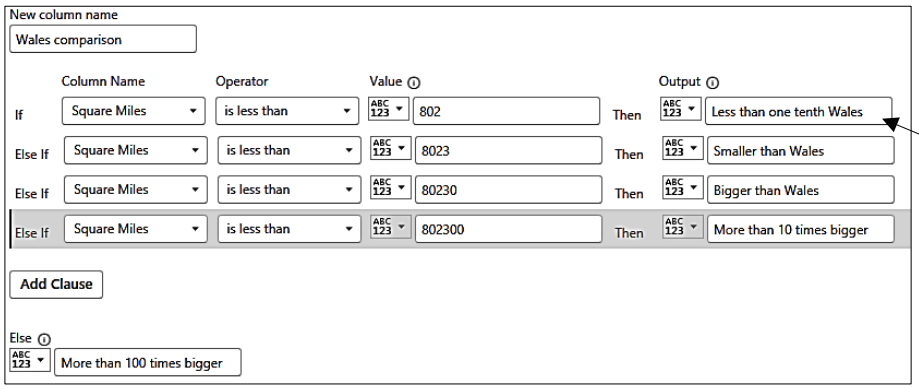
d) Click here to add other conditions.

e) Put here what should happen if all of the other conditions fail.





Continue adding conditions to get the final column:



Note that when writing numerical or date comparison conditions, you should start with the smallest/earliest number/date and work your way up (as here), or start with the biggest/latest one and work your way down.

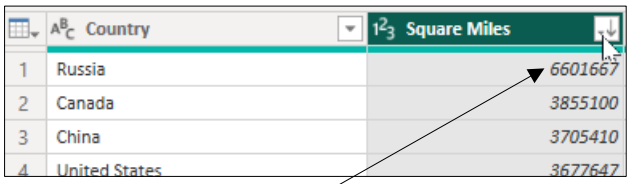
	Country	Square Miles	Wales comparison
1	Afghanistan	252072	More than 10 times bigger
2	Albania	11100	Bigger than Wales
3	Algeria	919595	More than 100 times bigger
4	Andorra	181	Less than one tenth Wales
5	Angola	481350	More than 10 times bigger
6	Antigua and Barbuda	171	Less than one tenth Wales

The first few countries listed in alphabetical order. The UK is more than 10 times bigger than Wales, if you're interested.

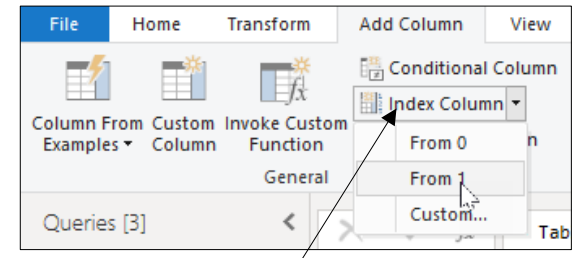
© Copyright 2026

2.7 Indexing Columns

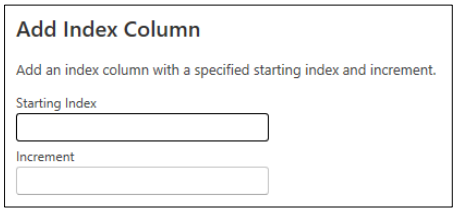
You can number rows in a table using an *index column*:

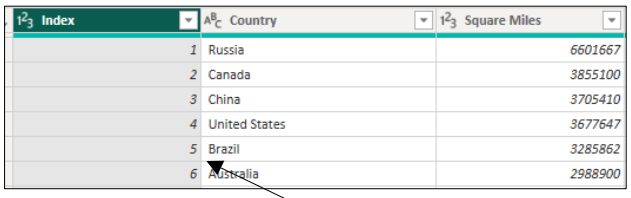


a) Sort your rows in the order in which you want them to be ordered (in this case, we'll order them by size, with the largest country first).



b) Add an index column from the **Add Column** menu, choosing to order the rows from 1, not 0. Note that you could instead choose **Custom…** to create any arithmetic series:





c) You can now use this column to show the original sort order (and to revert to it should you need to).

Our training and consultancy services

		Basic training	Advanced training	Systems / consultancy
Desktop	Microsoft Excel			
	VBA macros			
	Microsoft Access			
Business Intelligence	Power BI			
	Power Apps			
	Power Automate			
SQL Server	SQL			
	Reporting Services			
	Report Builder			
	Integration Services			
	Analysis Services			
Coding	Visual C# programming			
	VB programming			
	DAX			
	Python			