

Advanced Power BI (Data)

Sample manual - first two chapters



TABLE OF CONTENTS (1 of 4)

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

2	CREATING COLUMNS	Page
2.1	Ways to Create New Columns	15
2.2	Columns from Examples	16
	<i>Step 1 – Start the Feature</i>	16
	<i>Step 2 – Show some Examples</i>	17
	<i>Step 3 – Confirm the Formula</i>	17
	<i>Step 4 – Review your Formula</i>	17
	<i>Step 5 – Understanding your Formula</i>	18
2.3	Built-up Columns	19
2.4	Custom Columns in M	20
	<i>M Prefixes</i>	20
	<i>Our Example – Elapsed Days</i>	20
	<i>Creating a Custom Column</i>	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
	<i>Setting the Folder</i>	25
	<i>Combining the Files</i>	26
	<i>Setting the Template for Import</i>	26
	<i>Tidying Up the Results</i>	27
	<i>An Outline of How it Works</i>	27
3.2	Appending Queries	28
	<i>Combining Queries using Append</i>	28
	<i>Tidying up your Results</i>	29
3.3	Merging Data	30
	<i>Loading the Tables to Merge</i>	30
	<i>Merging the Data</i>	31
	<i>Flattening the Combined Table</i>	32
	<i>Final Touches</i>	32
3.4	Fuzzy Merging	32
	<i>Setting a Similarity Threshold</i>	34
	<i>Enforcing Case Sensitivity</i>	34
	<i>Using a Transformation Table</i>	34

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

TABLE OF CONTENTS (2 of 4)

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

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

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

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

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

10	ROW-LEVEL SECURITY	Page
10.1	Overview of Row-Level Security	64
	<i>Limitations and Constraints</i>	64
10.2	Roles in Power BI Desktop	64
	<i>Creating Roles</i>	65
	<i>Testing Roles</i>	66
10.3	Roles in Power BI Service	67
	<i>Configuring the App Workspace</i>	67
	<i>Assigning Users to Roles</i>	68
	<i>Testing your Roles (or Users)</i>	69
10.4	Dynamic Security	70
	<i>Creating a Permissions Table</i>	70
	<i>Creating a Role</i>	71
10.5	Creating Groups of Users	72
	<i>Creating the Tables</i>	72
	<i>Linking the Tables</i>	73
	<i>Creating the Role</i>	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
	<i>Showing the Advanced Editor</i>	76
	<i>The Excel Query</i>	76
	<i>Creating a SQL Server Version of our Table</i>	77
	<i>Switching the Data Source</i>	77
	<i>Resolving Problems</i>	78
	<i>Viewing the Results</i>	78

TABLE OF CONTENTS (3 of 4)

12	SIMPLE PARAMETERS	Page
12.1	Parameters, Slicers and Filters	79
	<i>Parameters Needed for this Example</i>	79
12.2	Working with Parameters	80
	<i>Accessing Parameters</i>	80
	<i>Creating Parameters</i>	80
	<i>Managing Parameters</i>	81
	<i>Loading Parameters into Power BI</i>	81
12.3	Entering Values for Parameters	81
	<i>Entering Parameter Values in Query Editor</i>	82
	<i>Entering Parameter Values in Power BI Desktop</i>	82
12.4	Using Parameters	82
	<i>Filtering by Parameter Values</i>	83
	<i>the Parameter Values Chosen</i>	84
12.5	Using Parameters with Templates	85
	<i>Creating a Template</i>	85
	<i>Opening (Using) a Template</i>	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
	<i>Step 1 – Create a Query Giving a Single Column of Data</i>	90
	<i>Step 2 – Make this a List Query</i>	90
	<i>Step 3 – Creating the Dropdown Parameter</i>	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
	<i>Creating the Parameters</i>	95
	<i>Referencing the Parameters in the Advanced Editor</i>	95
	<i>Editing Parameter Values</i>	97
	<i>Displaying the Parameters in your Report</i>	97

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

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

TABLE OF CONTENTS (4 of 4)

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

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

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

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

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

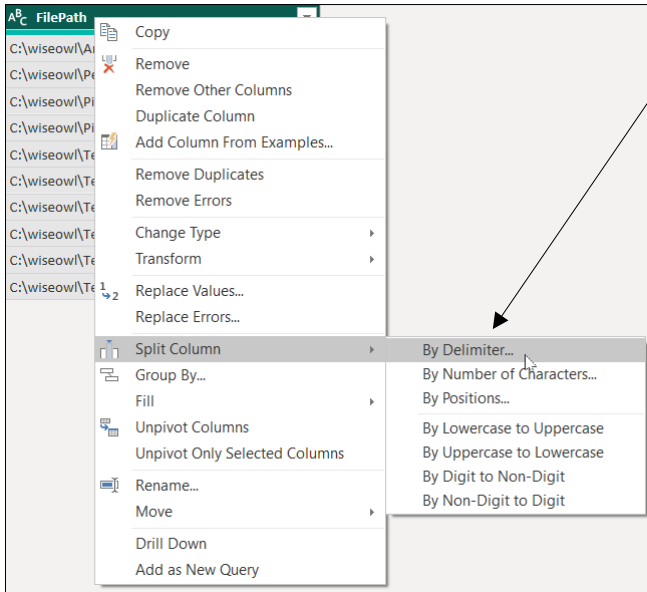
22	POWER BI IN POWER APPS	Page
22.1	Overview	152
22.2	Creating the Power BI Report	153
	<i>Setting a Custom Page Size</i>	153
	<i>Publishing the Page</i>	153
22.3	Creating your Dashboard	154
	<i>Pinning a Live Page to a Dashboard</i>	154
	<i>Hiding the Tile's Title and Subtitle</i>	155
22.4	Showing the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

CHAPTER 1 - MANIPULATING COLUMNS

1.1 Splitting Columns

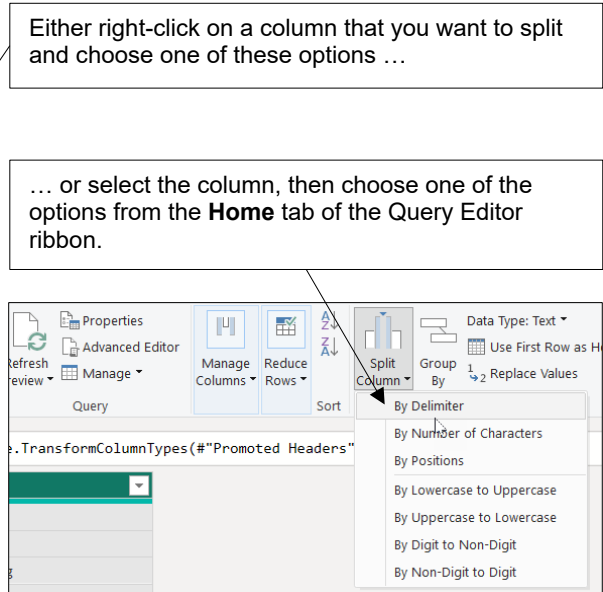
Accessing Split Column Menu

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:

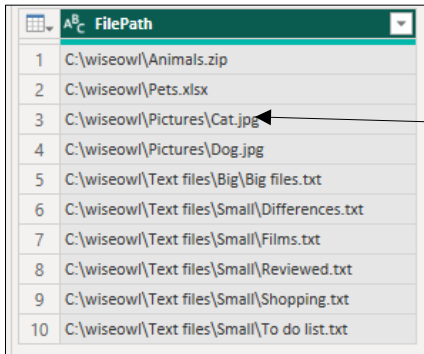
Option	What it does
<i>By Delimiter...</i>	Splits a string of text when Query Editor encounters a particular character.
<i>By Number of Characters...</i>	Splits a string of text after a given number of characters.
<i>By Position...</i>	Splits a string of text at certain pre-defined character positions.
<i>By Lowercase to Uppercase</i> <i>By Uppercase to Lowercase</i>	Splits a string of text whenever the case changes.
<i>By Digit to Non-Digit</i> <i>By Non-Digit to Digit</i>	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!

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:



a) Select the column you want to split by, and choose to split by delimiter:

By Delimiter...
 By Number of Characters...
 By Positions...

b) Choose a character to split by.

c) Choose the type of delimiter (see below for examples).

Split Column by Delimiter

Specify the delimiter used to split the text column

Select or enter delimiter

--Custom--

\

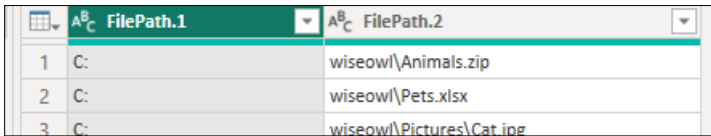
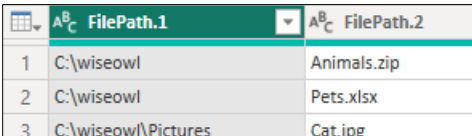
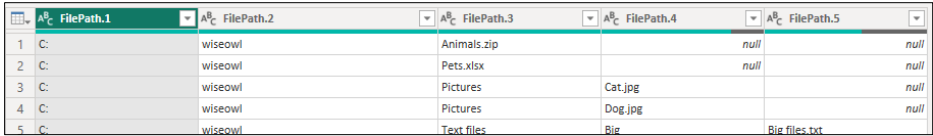
Split at

☐ Left-most delimiter

☒ **Right-most delimiter**

☐ Each occurrence of the delimiter

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

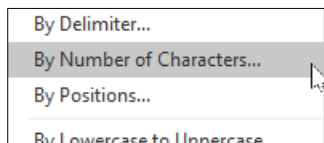
Option	Results	Notes
<i>Left-most delimiter</i>		Split by the first backslash
<i>Right-most delimiter</i>		Split by the last backslash
<i>Each occurrence of the delimiter</i>		Split by every backslash

Splitting by Number of Characters

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

	A ⁸ 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
5	ESQ-0031	20.87047060

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



Split Column by Number of Characters

Specify the number of characters used to split the text column.

Number of characters

Split

- ☒ Once, as far left as possible
☐ Once, as far right as possible
☐ Repeatedly

- 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**).

	A ⁸ Product Code.1	1 ² 3 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.

Splitting by Positions

Use this method to extract a substring of text:

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

By Delimiter...
By Number of Characters...
By Positions...
By Lowercase to Uppercase
By Uppercase to Lowercase

	A ^B C 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	
5	NHS-91/03-10	

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

Split Column by Positions

Specify the positions at which to split the text column.

Positions

0,4,6

	A ^B C Product code.1	1,2,3 Product code.2	A ^B C Product code.3
1	VPW-	85	/L9-50
2	COS-	84	/G6-98
3	MZZ-	45	/A3-22
4	VWO-	51	/P3-17
5	NHS-	91	/03-10

- 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:

	A ^B C Shopping centre	A ^B C PostCode	1,2,3 Num
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 Delimiter...
By Number of Characters...
By Positions...
By Lowercase to Uppercase
By Uppercase to Lowercase
By Digit to Non-Digit
By Non-Digit to Digit

By repeated application of these two transforms, for example ...

A ^B C PostCode.1	A ^B C PostCode.2	A ^B C PostCode.3.1	A ^B C PostCode.3.2
EN	8	7	BZ
SM	1	1	LF
NR	32	2	ED
CB	3	8	EL
PL	6	5	BL
PO	15	7	LJ
CV	4	7	EH
OL	7	0	DN
BS	34	7	JL
SS	6	7	TR

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

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!):

Split Column by Delimiter

Specify the delimiter used to split the text column.

Select or enter delimiter
 --Custom--
 -

Split at
☐ Left-most delimiter
☐ Right-most delimiter
☒ Each occurrence of the delimiter

Advanced options
 Split into
☐ Columns
☒ Rows

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

	A ^B C 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

... 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).

	A ^B C Product Code	1.2 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

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.

	A ^B _C 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																				
<i>Remove quote marks</i>	<div><h3>Split Column by Delimiter</h3><p>Specify the delimiter used to split the text column.</p><p>Select or enter delimiter</p><div>Comma</div><p>Split at</p><p><input type="radio"/> Left-most delimiter</p><p><input type="radio"/> Right-most delimiter</p><p><input checked="" type="radio"/> Each occurrence of the delimiter</p><p>Advanced options</p><p>Quote Character</p><div>"</div></div>	<table><tr><th></th><th>A^B_C Pet.1</th><th>A^B_C Pet.2</th><th>I²₃ Pet.3</th></tr><tr><td>1</td><td>Annie</td><td>Cat</td><td>4</td></tr><tr><td>2</td><td>Neo</td><td>Cat</td><td>4</td></tr><tr><td>3</td><td>Tommy</td><td>Tortoise</td><td>4</td></tr><tr><td>4</td><td>Pogba</td><td>Parrot</td><td>2</td></tr></table>		A ^B _C Pet.1	A ^B _C Pet.2	I ² ₃ Pet.3	1	Annie	Cat	4	2	Neo	Cat	4	3	Tommy	Tortoise	4	4	Pogba	Parrot	2
	A ^B _C Pet.1	A ^B _C Pet.2	I ² ₃ Pet.3																			
1	Annie	Cat	4																			
2	Neo	Cat	4																			
3	Tommy	Tortoise	4																			
4	Pogba	Parrot	2																			
<i>Keep quote marks</i>	<div><h3>Split Column by Delimiter</h3><p>Specify the delimiter used to split the text column.</p><p>Select or enter delimiter</p><div>Comma</div><p>Split at</p><p><input type="radio"/> Left-most delimiter</p><p><input type="radio"/> Right-most delimiter</p><p><input checked="" type="radio"/> Each occurrence of the delimiter</p><p>Advanced options</p><p>Quote Character</p><div>None</div></div>	<table><tr><th></th><th>A^B_C Pet.1</th><th>A^B_C Pet.2</th><th>I²₃ Pet.3</th></tr><tr><td>1</td><td>"Annie"</td><td>"Cat"</td><td>4</td></tr><tr><td>2</td><td>"Neo"</td><td>"Cat"</td><td>4</td></tr><tr><td>3</td><td>"Tommy"</td><td>"Tortoise"</td><td>4</td></tr><tr><td>4</td><td>"Pogba"</td><td>"Parrot"</td><td>2</td></tr></table>		A ^B _C Pet.1	A ^B _C Pet.2	I ² ₃ Pet.3	1	"Annie"	"Cat"	4	2	"Neo"	"Cat"	4	3	"Tommy"	"Tortoise"	4	4	"Pogba"	"Parrot"	2
	A ^B _C Pet.1	A ^B _C Pet.2	I ² ₃ 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.

1.2 Merging Columns

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

	A ^B C Product Code.1	A ^B C Product Code.2	1.2 Sales
1	MPB	4242	
2	HQC	0233	
3	ZND	9682	
4	PGL	4142	
5	ESQ	9921	
6	SDB	9053	
7	CNA	4049	
8	UVU	0852	
9	CPU	4615	
10	PCS	2315	
11	YKF	3883	
12	TMZ	2076	

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).

	A ^B C 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.

Merge Columns

Choose how to merge the selected columns.

Separator

--Custom--

-

New column name (optional)

Original product code



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

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:

The top screenshot shows the **Transform** tab selected. The **Extract** dropdown menu is open, showing options: Length, First Characters, Last Characters, Range, Text Before Delimiter, Text After Delimiter, and Text Between Delimiters. A callout box points to this menu with the text: "If you choose this version on the **Transform** tab of the ribbon you will replace a column ...".

The bottom screenshot shows the **Add Column** tab selected. The **Extract** dropdown menu is open, showing the same options. A callout box points to this menu with the text: "... 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.

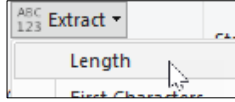
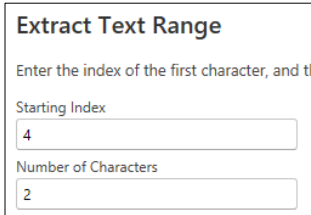
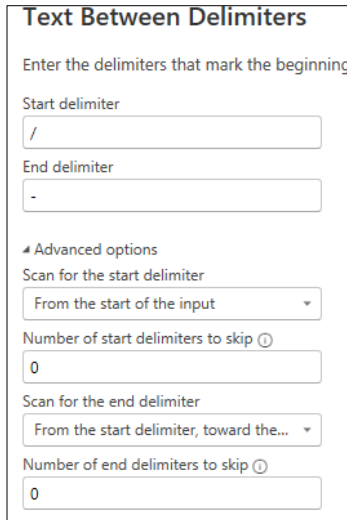
c) Specify how many characters you want to extract (here we've gone for the first 3).

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

The final result shows the **Product Code** column with values: 1 MPB, 2 HQC, 3 ZND, 4 PGL. The **Sales** column values remain: 24.49703694, 44.9529665, 91.64723297, 30.60766405.

The Possible Options

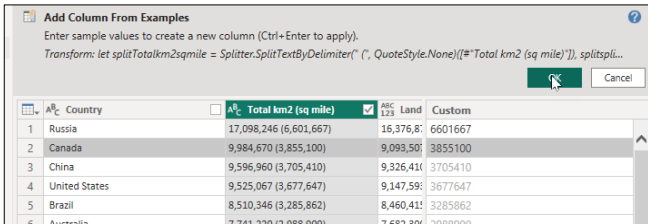
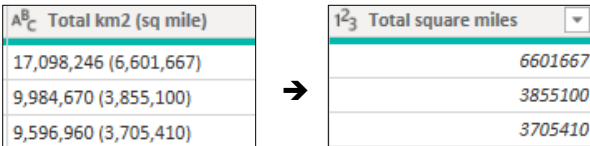
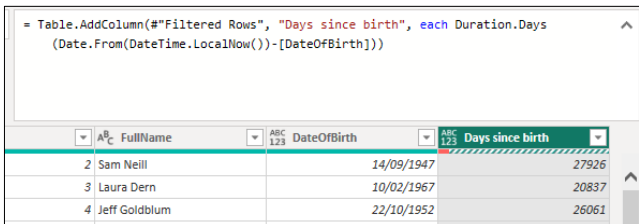
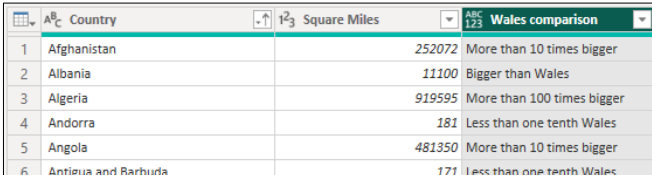
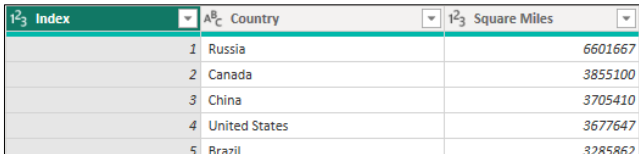
Here's what the options would give for a field containing the product code **VPW-85/L9-50**:

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	<div> Extract First Characters Enter how many starting characters to keep. Count <input type="text" value="3"/> </div> <div> Extract Last Characters Enter how many ending characters to keep. Count <input type="text" value="2"/> </div>	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	<div> Text Before Delimiter Enter the delimiter that marks the end of what you want. Delimiter <input type="text" value="-"/> Advanced options Scan for the delimiter From the start of the input Number of delimiters to skip ⁽ⁱ⁾ <input type="text" value="0"/> </div> <div> Text After Delimiter Enter the delimiter that marks the beginning of what you want. Delimiter <input type="text" value="-"/> Advanced options Scan for the delimiter From the end of the input Number of delimiters to skip ⁽ⁱ⁾ <input type="text" value="0"/> </div>	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.

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

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:

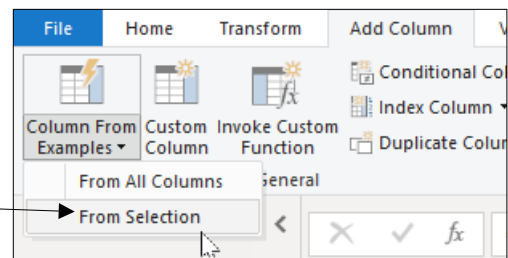
For this example we want to show the total area of each country (the list comes from Wikipedia) in square miles. So for Russia this should give **6601667**, the figure in brackets.

	Country	Total km2 (sq mile)	Land km2 (sq mile)	Water km2 (sq mile)
1	Russia	17,098,246 (6,601,667)		
2	Canada	9,984,670 (3,855,100)		
3	China	9,596,960 (3,705,410)		
4	United States	9,525,067 (3,677,647)		
5	Brazil	8,510,346 (3,285,862)		
6	Australia	7,741,220 (2,988,900)		
7	India	3,287,263 (1,269,219)		

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.



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)	16,376,810	6601667
2	Canada	9,984,670 (3,855,100)	9,093,500	
3	China	9,596,960 (3,705,410)	9,326,410	
4	United States	9,525,067 (3,677,647)	9,147,590	

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

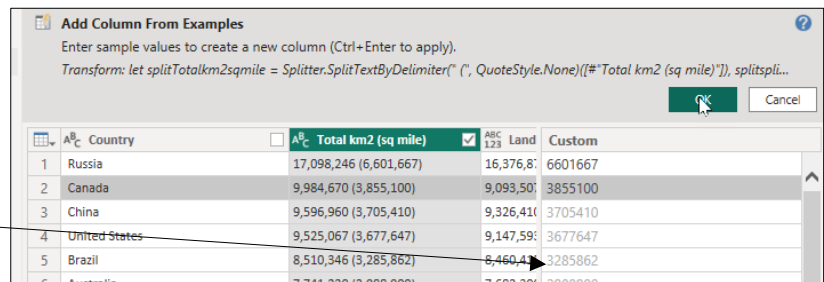
	Country	Total km2 (sq mile)	Land	Column1
1	Russia	17,098,246 (6,601,667)	16,376,810	6601667
2	Canada	9,984,670 (3,855,100)	9,093,500	3855100
3	China	9,596,960 (3,705,410)	9,326,410	

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 **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.

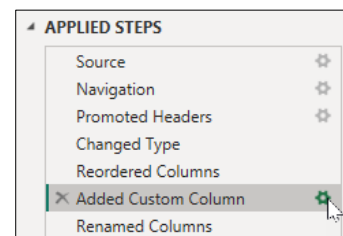


Step 4 – Review your Formula

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:



Custom Column

Add a column that is computed from the other columns.

New column name

Custom

Custom column 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)))
```

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

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:

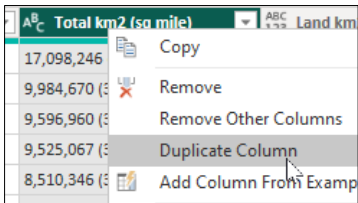
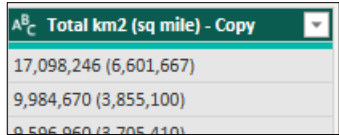
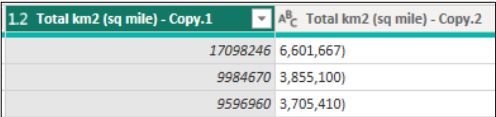

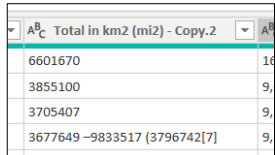
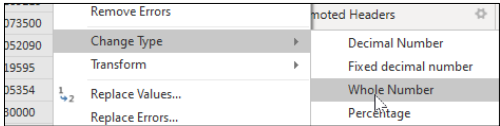
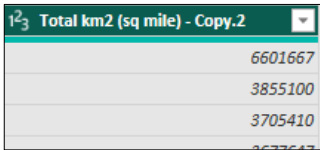
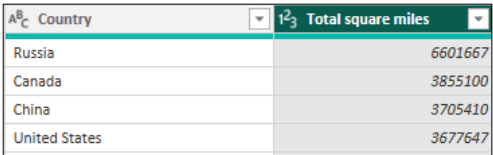
Stage	What it does	Returns for row 1
<i>Before any formulae applied</i>		17,098,246 (6,601,667)
<code>let splitTotalkm2sqmile = Splitter.SplitTextByDelimiter("(",", QuoteStyle.None)([#"Total km2 (sq mile)"])</code>	Creates a variable with a long unwieldy name (splitTotalkm2sqmile) to take the column 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) .
<code>splitsplitTotalkm2sqmile1 = Splitter.SplitTextByDelimiter("","", QuoteStyle.None)(splitTotalkm2sqmile{1}?)</code>	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) .
<code>let ... in Text.Combine(List.Transform(splitsplitTotalkm2sqmile1, each Text.Start(_, 3)))</code>	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!

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
<i>Duplicate column</i>	<p>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.</p> 	
<i>Split by bracket</i>	<p>Right-click on this duplicated column and choose Split Column > By Delimiter..., then choose a <code>()</code> as your delimiter.</p>	
<i>Remove the end bracket</i>	<p>Right-click on the second column generated and choose Replace Values..., then replace every <code>()</code> with nothing.</p>	
<i>Remove the commas</i>	<p>Repeat the above step, replacing each <code>,</code> with nothing.</p>	
<i>Change the value to a number</i>	<p>Right-click on the column and choose to change the type to a number:</p> 	
<i>Tidy up the columns</i>	<p>Delete any intermediate columns generated, reorder your columns and rename this one.</p>	



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.

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
<i>Date</i>	Functions to do with dates	<pre>= Date</pre> <ul style="list-style-type: none"> Date.AddDays Date.AddMonths Date.AddQuarters Date.AddWeeks Date.AddYears Date.Day Date.DayOfWeek Date.DayOfWeekName
<i>DateTime</i>	Functions to do with dates/times	<pre>= DateTime</pre> <ul style="list-style-type: none"> DateTime.AddZone DateTime.Date DateTime.FixedLocalNow DateTime.From DateTime.FromFileTime DateTime.FromText DateTime.IsInCurrentHour DateTime.IsInCurrentMinute
<i>Duration</i>	Functions giving the duration of time	<pre>= Duration</pre> <ul style="list-style-type: none"> Duration.Days Duration.From Duration.FromText Duration.Hours Duration.Minutes Duration.Seconds Duration.ToRecord

Prefix	What it gives	Example
<i>Number</i>	Functions giving things you can do to numbers	<pre>= Num</pre> <ul style="list-style-type: none"> Number.Abs Number.Acos Number.Asin Number.Atan Number.Atan2 Number.BitwiseAnd Number.BitwiseNot
<i>Text</i>	Functions to do with text	<pre>= Text</pre> <ul style="list-style-type: none"> Text.AfterDelimiter Text.At Text.BeforeDelimiter Text.BetweenDelimiters Text.Clean Text.Combine Text.Contains

Our Example – Elapsed Days

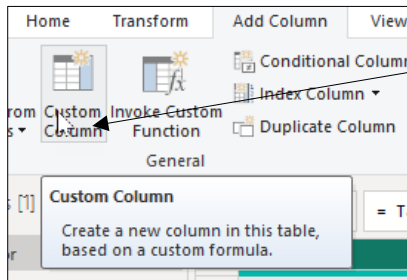
Suppose you have a table of actors, and want to find out how many days have elapsed since each was born:

ABC 123	DateOfBirth	ABC 123	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.

Creating a Custom Column

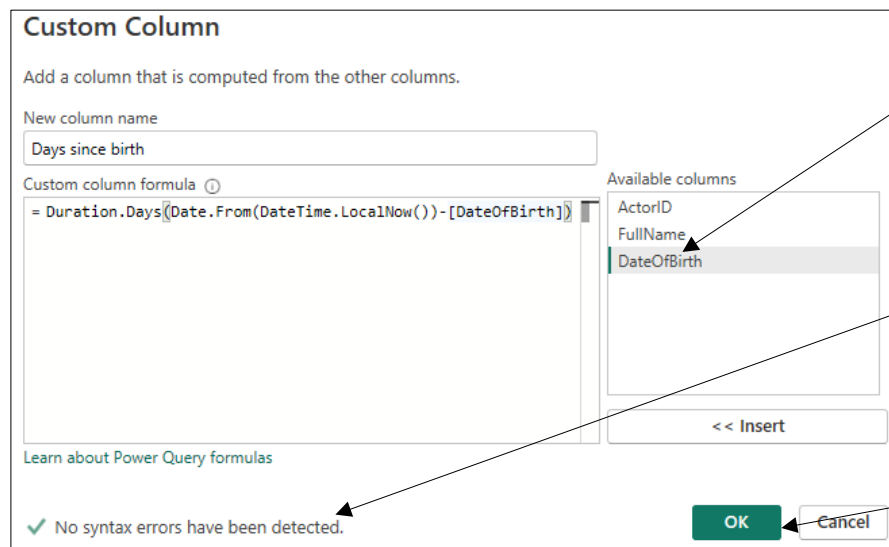
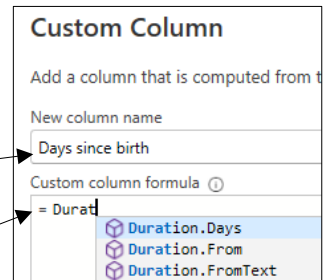
Here's how to create a column like this:



a) Click on this 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]))
```

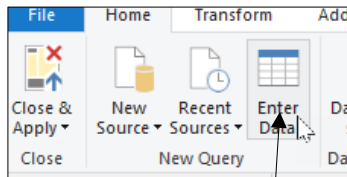
g) Power Query will incorporate this in a new query step (in this case showing for each row in the given table the number of days between each actor's birth date and today's date).



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?

2.5 Special Case - Last Refresh Date/Time

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

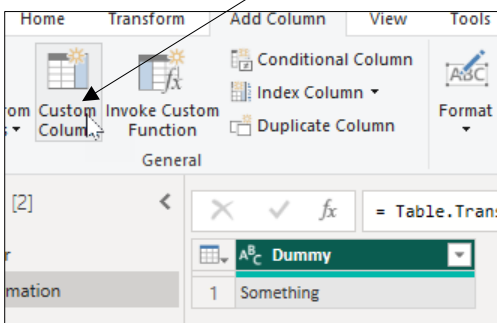


a) In Power Query, choose to create a new table by entering data for it.

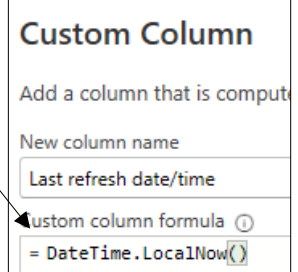
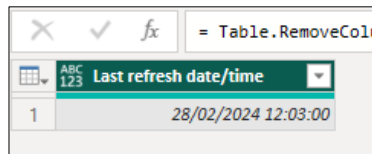
b) Make sure your table has one row in, since you want it to contain one bit of information only (it doesn't matter what this column is called, nor what it contains).



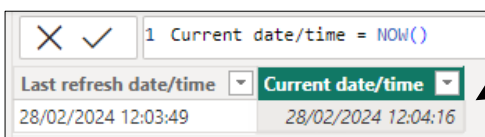
c) Choose to add a custom column from the **Add Column** tab of the ribbon.



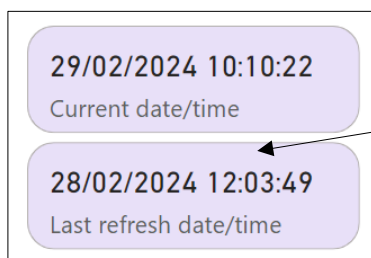
d) Create an expression giving the current date/time. You can then delete the original column to give your final table:



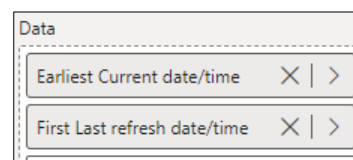
After loading your table, you could create a calculated column in it to show the current date and time, using the DAX **NOW()** function, then display both bits of information side by side:



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):



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:

New column name
Wales comparison

	Column Name	Operator	Value	Output
If	Square Miles	is less than	802	Less than one tenth Wales
Else If	Square Miles	is less than	8023	Smaller than Wales
Else If	Square Miles	is less than	80230	Bigger than Wales
Else If	Square Miles	is less than	802300	More than 10 times bigger

Add Clause

Else
More than 100 times bigger

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.

2.7 Indexing Columns

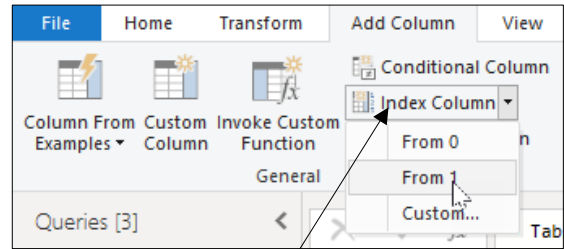
You can number rows in a table using an *index column*:

	A ^B Country	I ² Square Miles
1	Russia	6601667
2	Canada	3855100
3	China	3705410
4	United States	3677647

- 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).

I ² Index	A ^B Country	I ² Square Miles
1	Russia	6601667
2	Canada	3855100
3	China	3705410
4	United States	3677647
5	Brazil	3285862
6	Australia	2988900

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



- 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:

Add Index Column

Add an index column with a specified starting index and increment.

Starting Index

Increment

Handwriting practice lines consisting of 24 horizontal dotted lines.



Blank lined area for writing.



Blank lined area for writing.



Blank lined paper for writing.



Blank lined paper for writing.



Blank lined paper for writing.



WiseOwl
Training

Blank lined paper for writing.



WiseOwl
Training

Blank lined paper for writing.






































Handwriting practice lines consisting of 20 horizontal dotted lines.



Handwriting practice lines consisting of 20 horizontal dotted lines.



What we do!

		Basic training	Advanced training	Systems / consultancy
Office	Microsoft Excel			
	VBA macros			
	Office Scripts			
	Microsoft Access			
Power BI, etc	Power BI and DAX			
	Power Apps			
	Power Automate (both)			
SQL Server	SQL			
	Reporting Services			
	Report Builder			
	Integration Services			
	Analysis Services			
Coding	Visual C#			
	VB programming			
	MySQL			
	Python			



WiseOwl
Training

