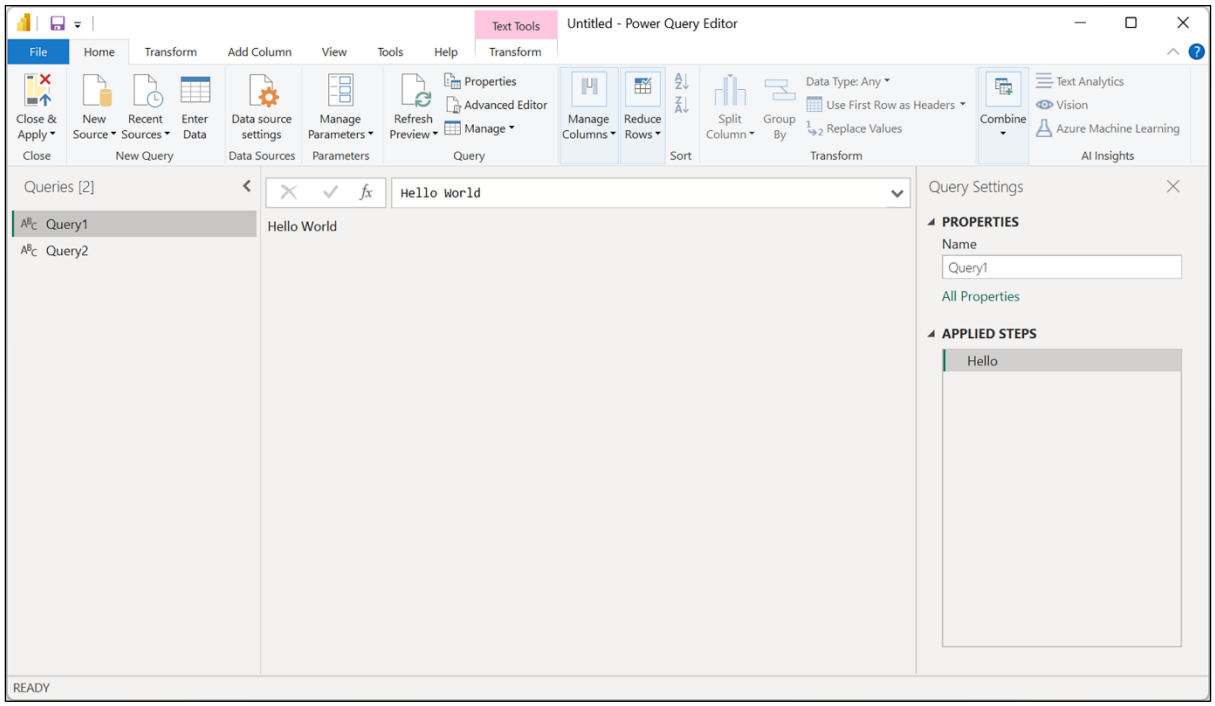
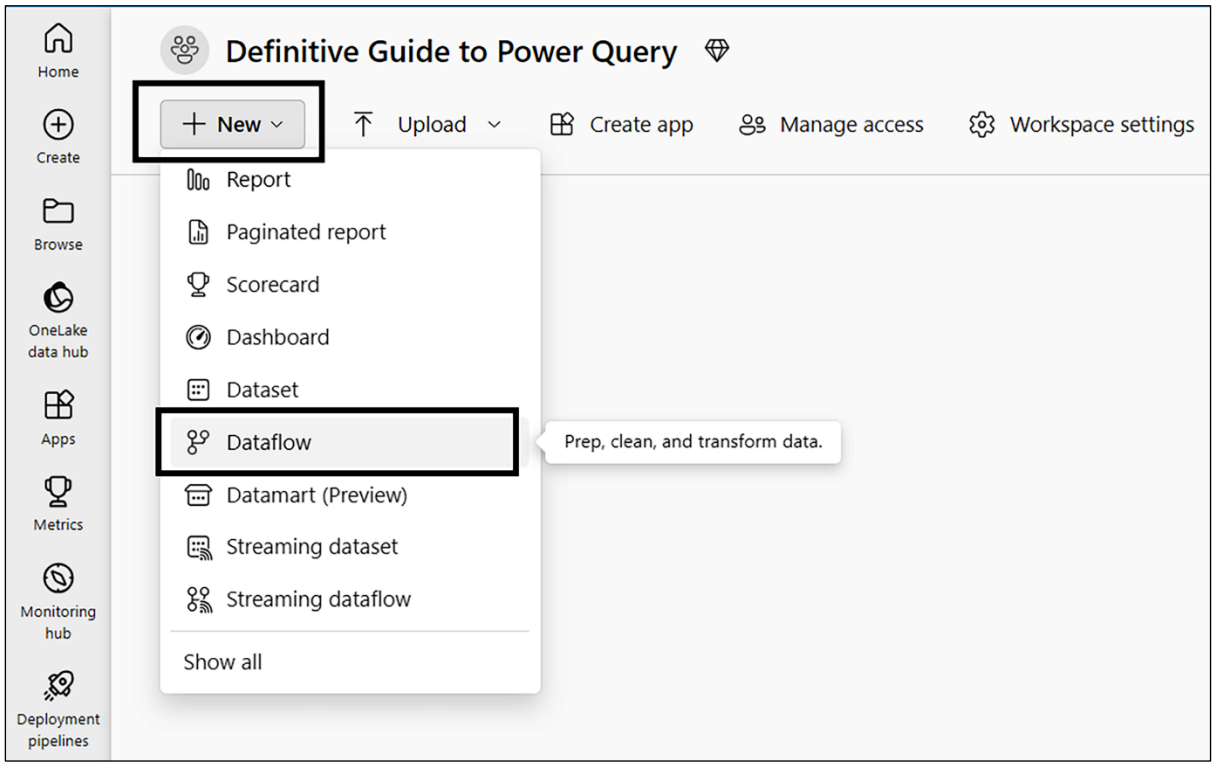
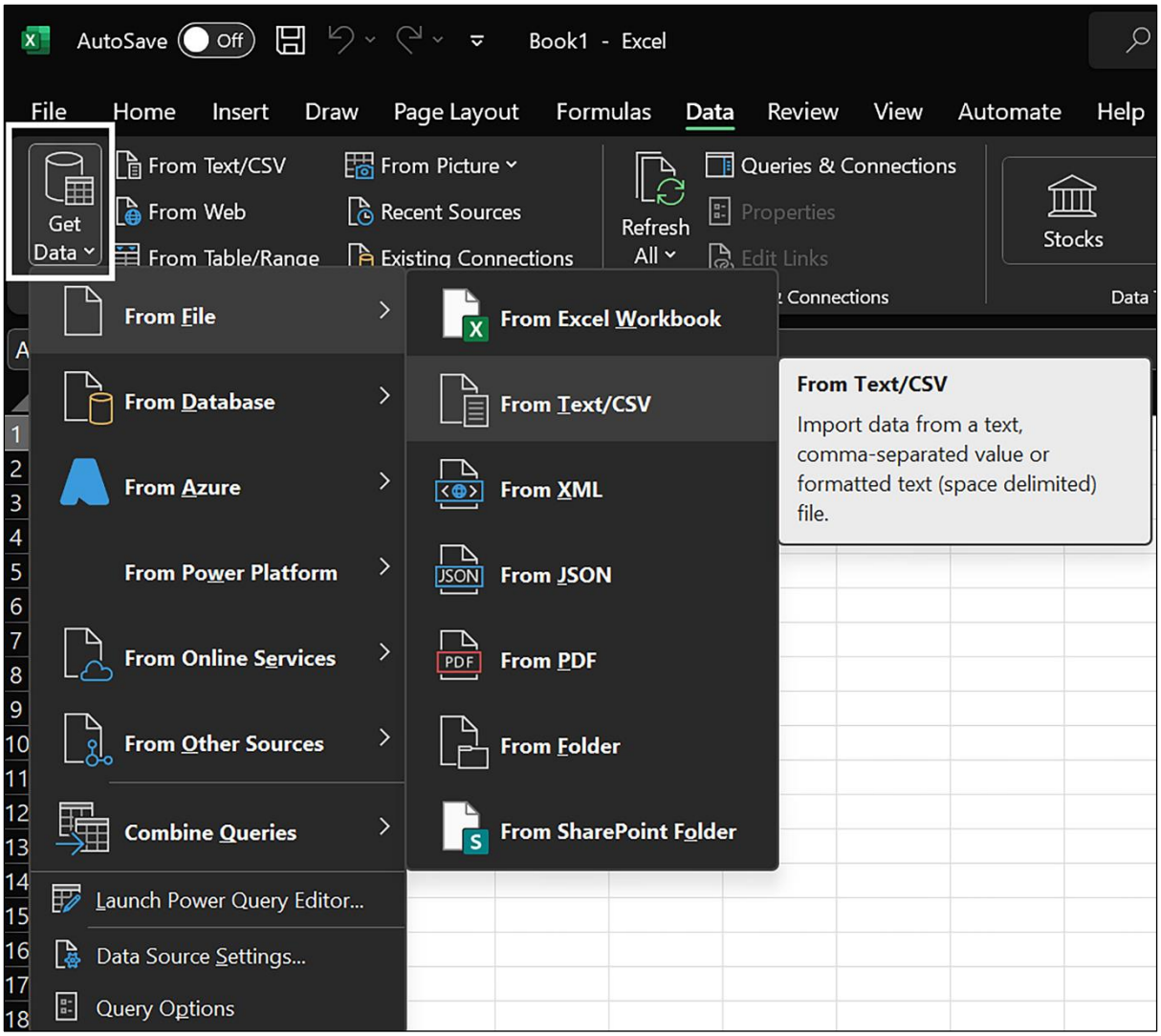
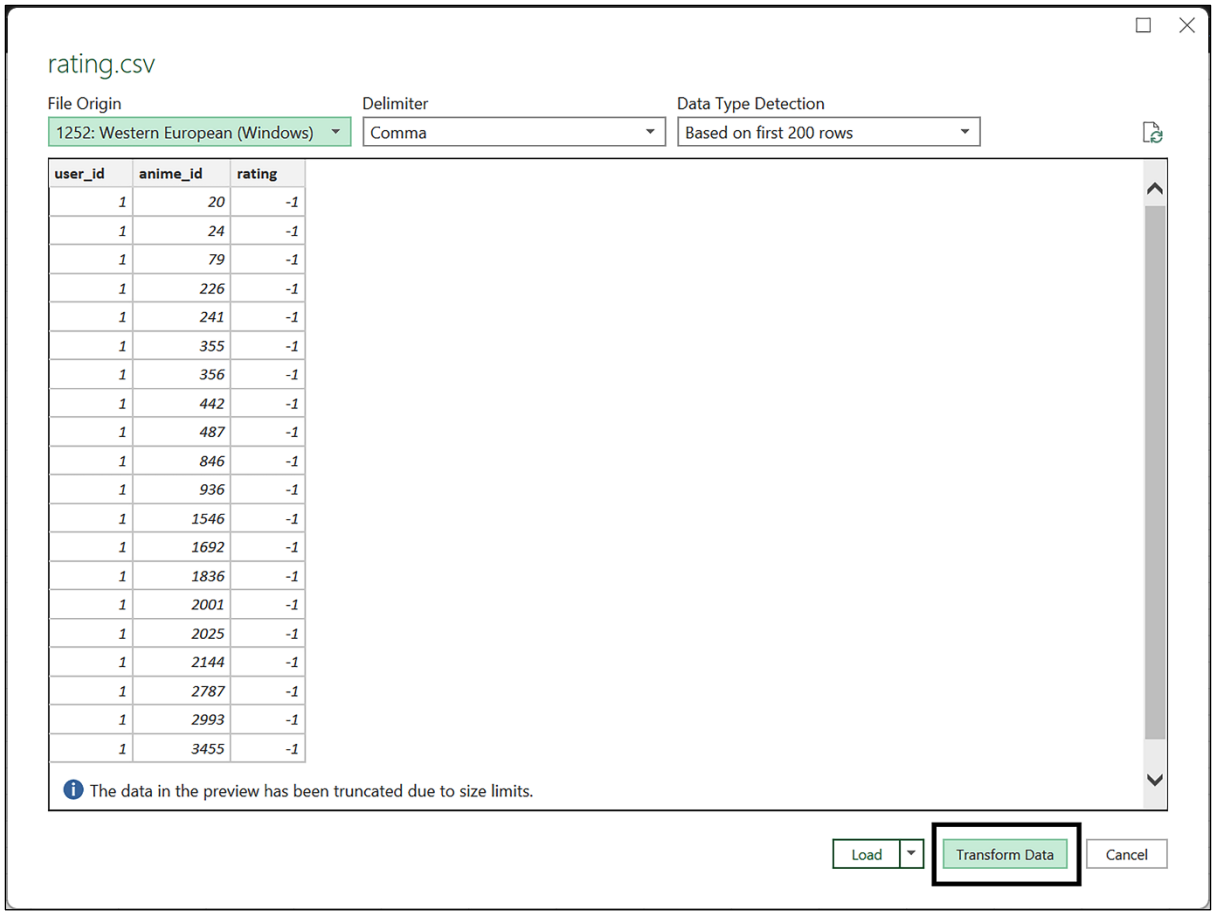
Chapter 1: Introducing M

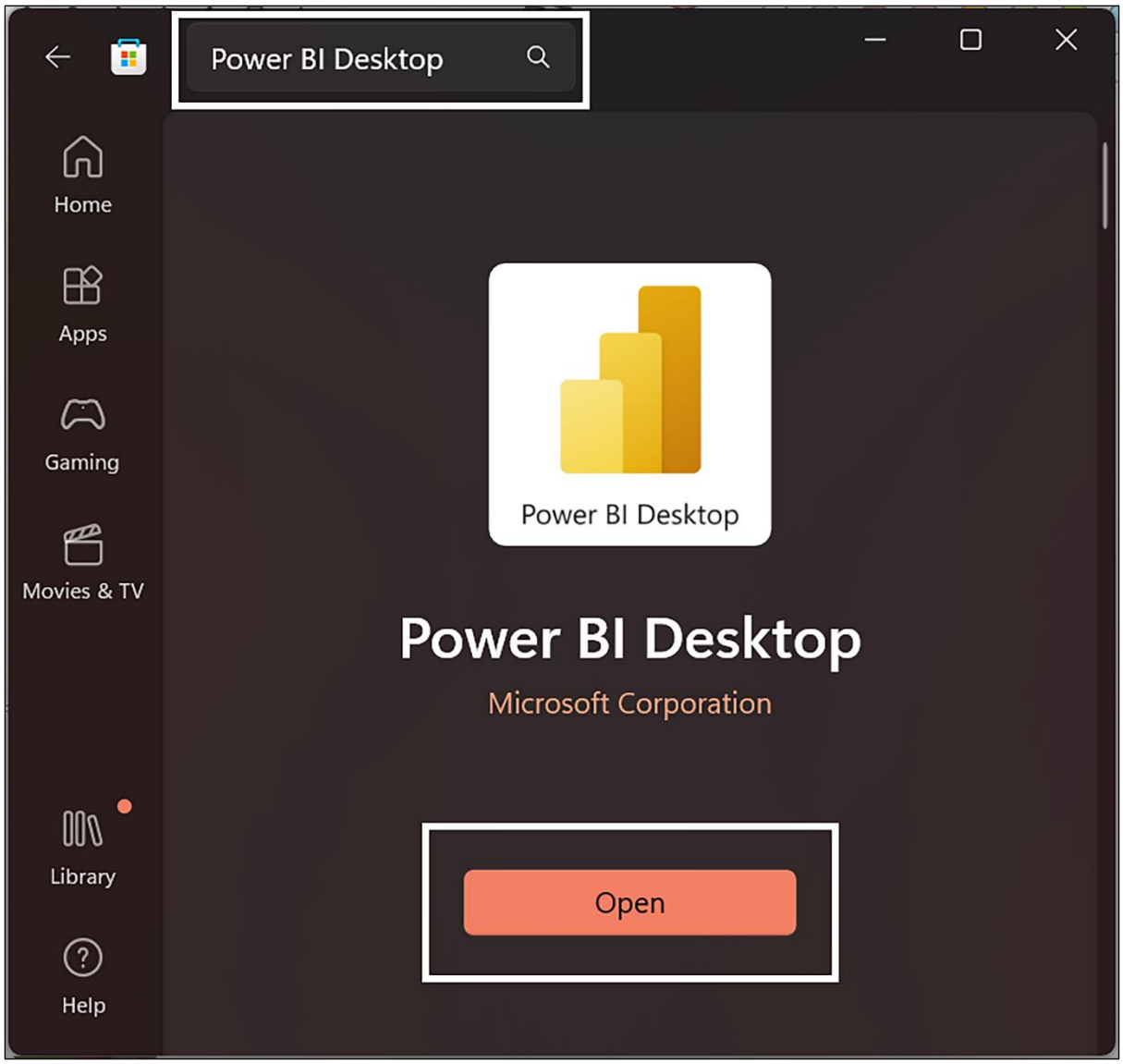


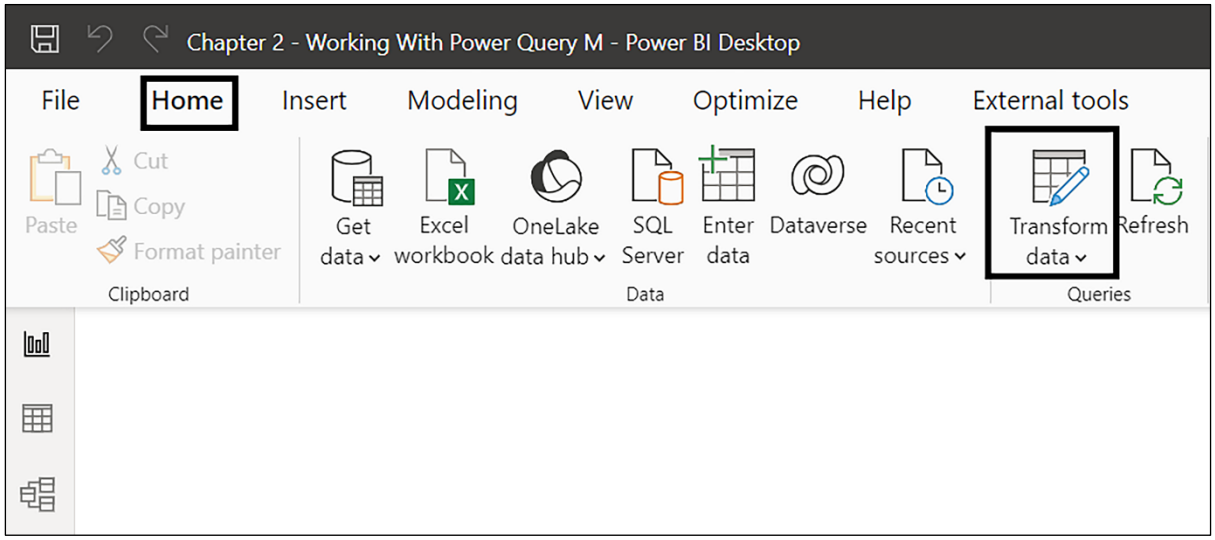


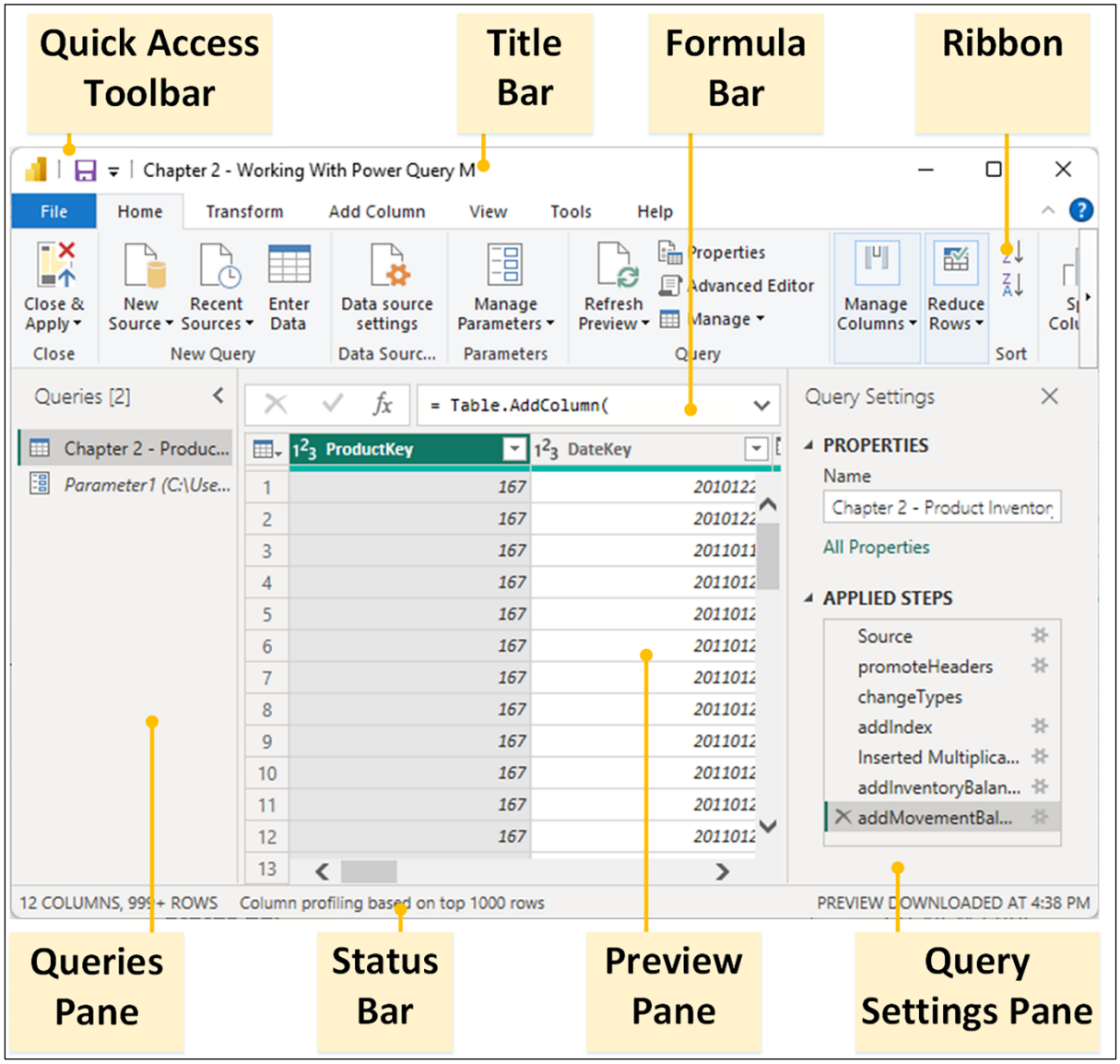

AutoSave
Off
Book1 - Excel
File
Home
Insert
Draw
Page Layout
Formulas
Data
Review
View
Automate
Help
Get Data
From Text/CSV
From Web
From Table/Range
From Picture
Recent Sources
Existing Connections
Refresh All
Queries & Connections
Properties
Edit Links
Stocks
From File
From Database
From Azure
From Power Platform
From Online Services
From Other Sources
Combine Queries
Launch Power Query Editor...
Data Source Settings...
Query Options
From Excel Workbook
From Text/CSV
From XML
From JSON
From PDF
From Folder
From SharePoint Folder
From Text/CSV
Import data from a text, comma-separated value or formatted text (space delimited) file.

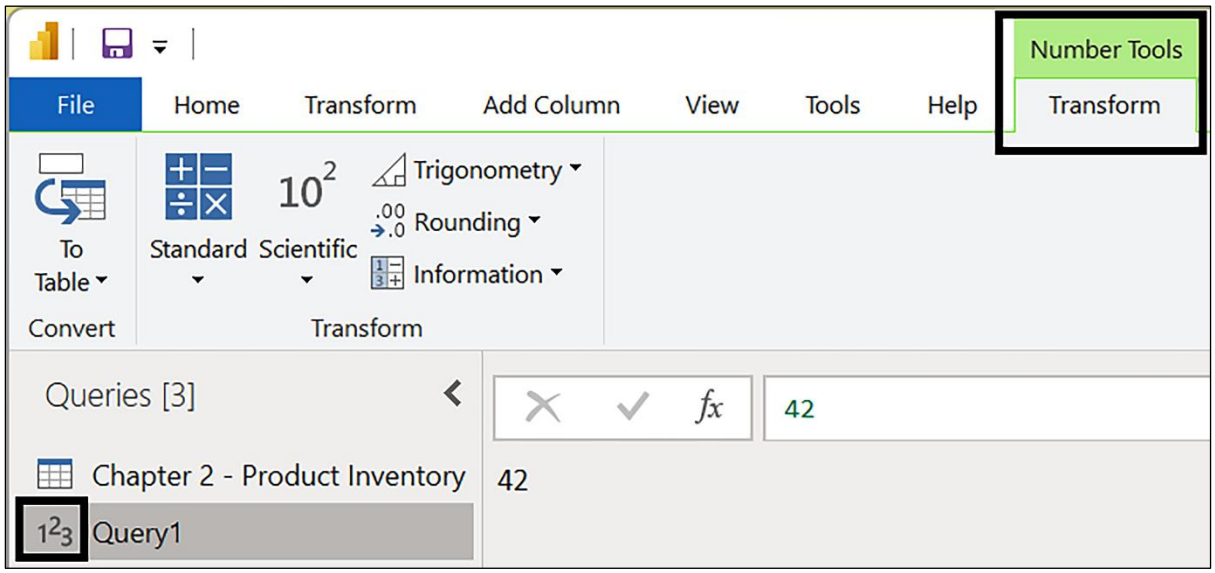

rating.csv
File Origin
1252: Western European (Windows)
Delimiter
Comma
Data Type Detection
Based on first 200 rows
user_id anime_id rating
1 20 -1
1 24 -1
1 79 -1
1 226 -1
1 241 -1
1 355 -1
1 356 -1
1 442 -1
1 487 -1
1 846 -1
1 936 -1
1 1546 -1
1 1692 -1
1 1836 -1
1 2001 -1
1 2025 -1
1 2144 -1
1 2787 -1
1 2993 -1
1 3455 -1
The data in the preview has been truncated due to size limits.
Load
Transform Data
Cancel

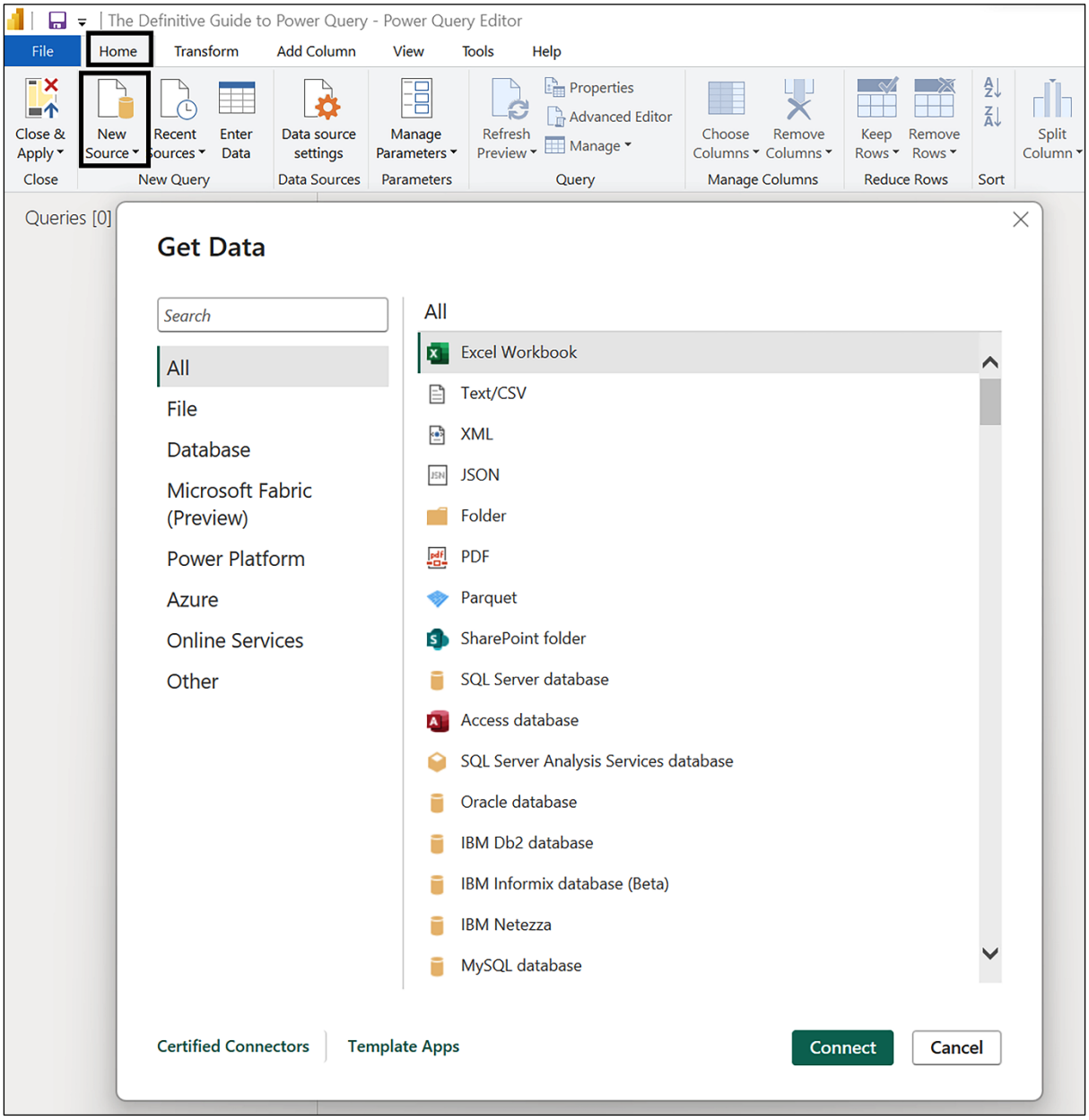
Chapter 2: Working with Power Query/M

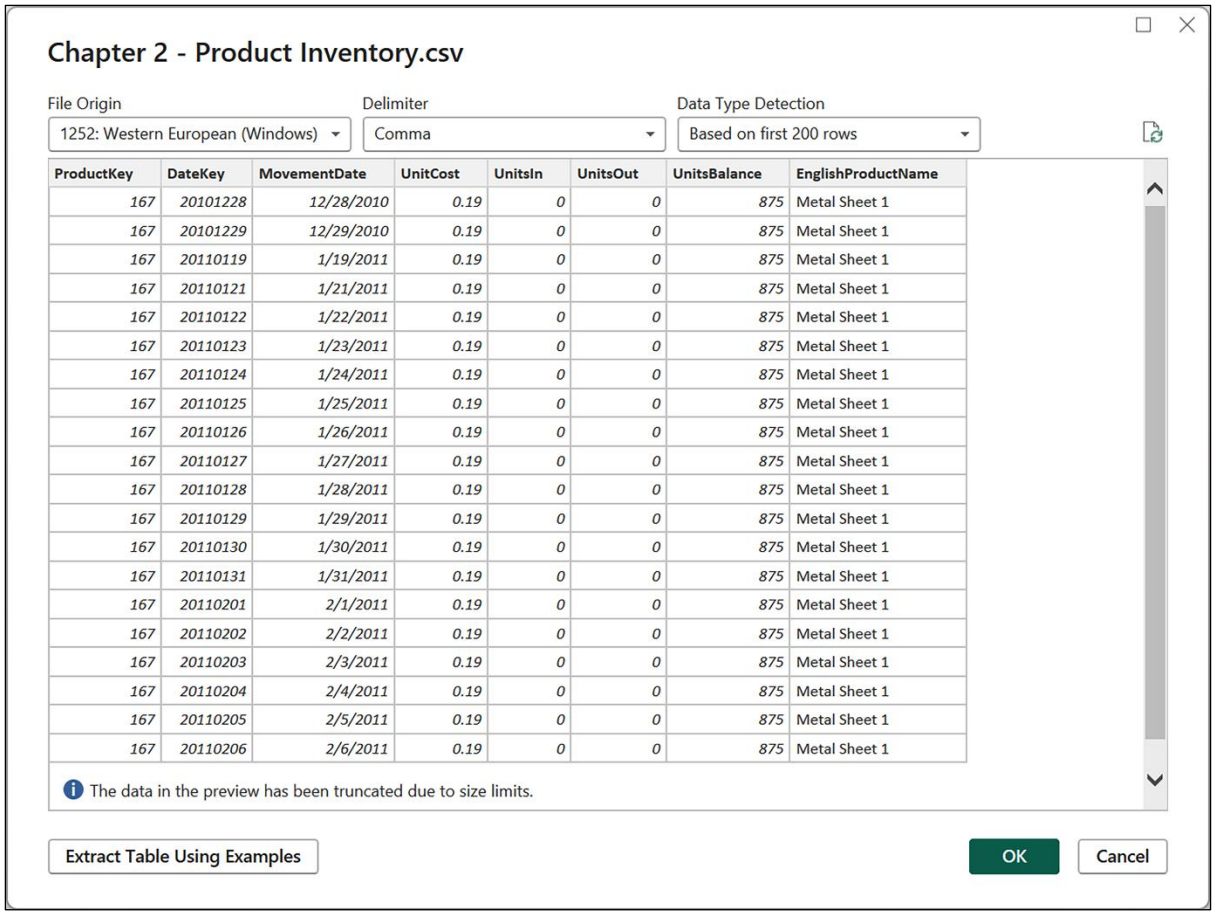


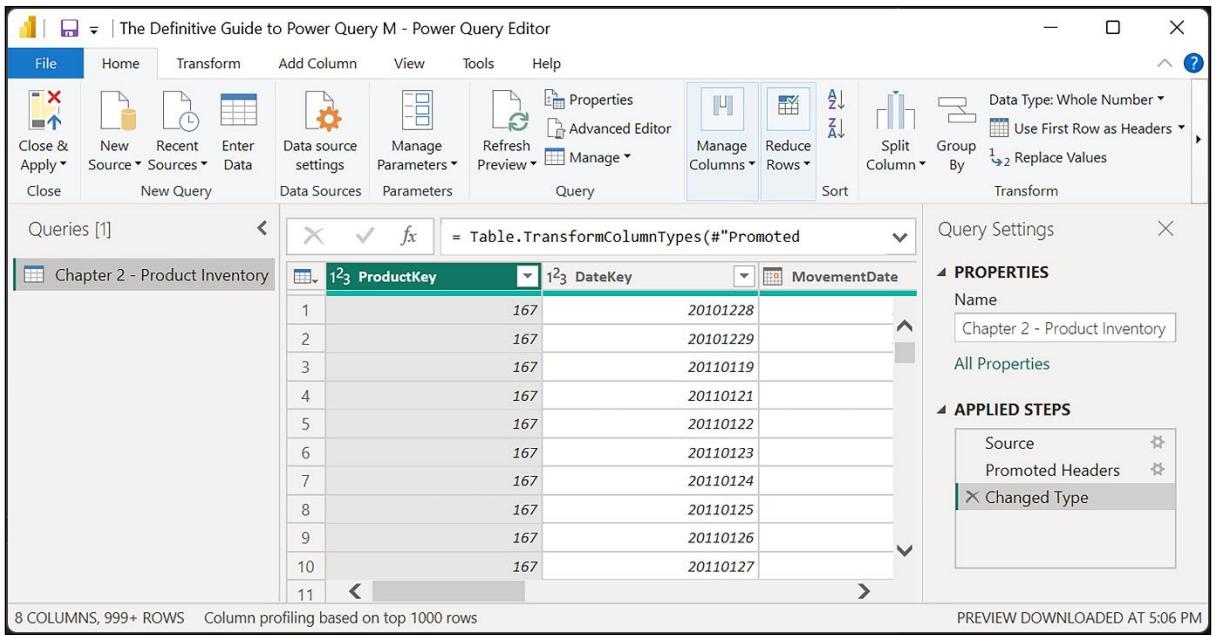


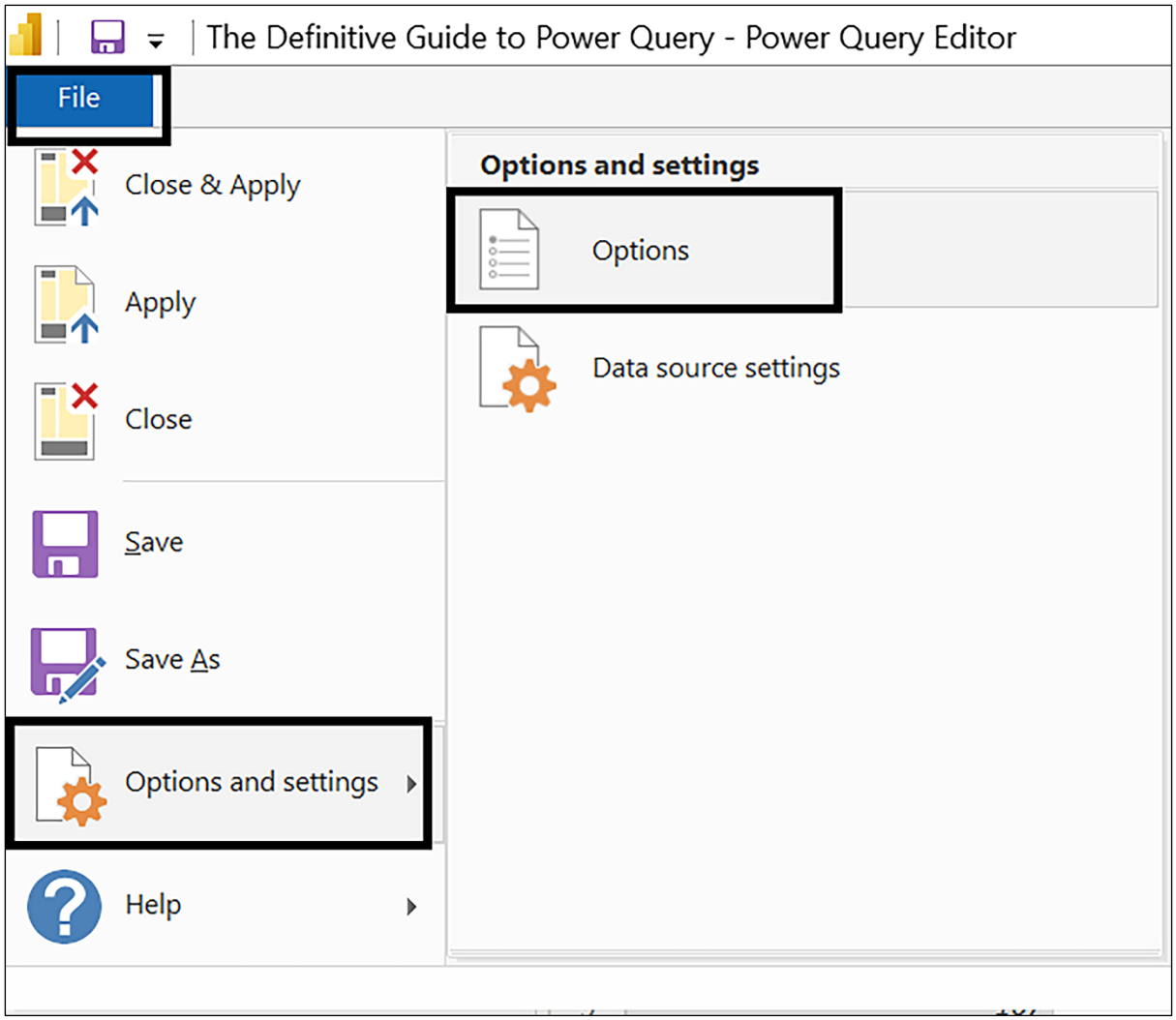

Quick Access Toolbar
Title Bar
Formula Bar
Ribbon
Chapter 2 - Working With Power Query M
File
Home
Transform
Add Column
View
Tools
Help
Close & Apply
New Source
Recent Sources
Enter Data
Data source settings
Manage Parameters
Refresh Preview
Properties
Advanced Editor
Manage
Manage Columns
Reduce Rows
Close
New Query
Data Sourc...
Parameters
Query
Sort
Queries [2]
Chapter 2 - Produc...
Parameter1 (C:\Use...
= Table.AddColumn(
ProductKey
DateKey
Query Settings
PROPERTIES
Name
Chapter 2 - Product Inventor
All Properties
APPLIED STEPS
Source
promoteHeaders
changeTypes
addIndex
Inserted Multiplica...
addInventoryBalan...
addMovementBal...
12 COLUMNS, 99+ ROWS
Column profiling based on top 1000 rows
PREVIEW DOWNLOADED AT 4:38 PM
Queries Pane
Status Bar
Preview Pane
Query Settings Pane

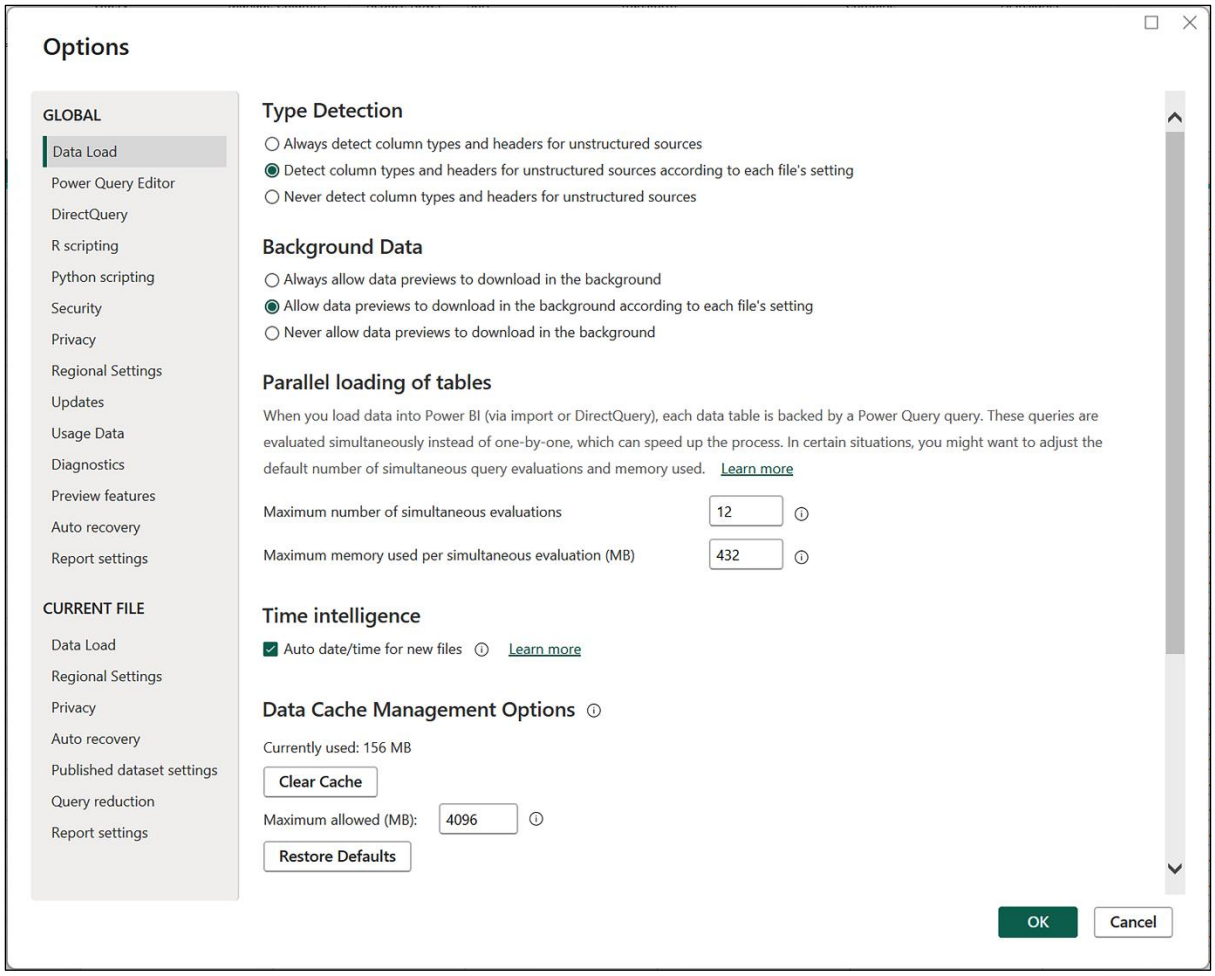

Number Tools
File
Home
Transform
Add Column
View
Tools
Help
Transform
To Table
Standard
Scientific
Trigonometry
Rounding
Information
Convert
Transform
Queries [3]
Chapter 2 - Product Inventory
Query1
42
42

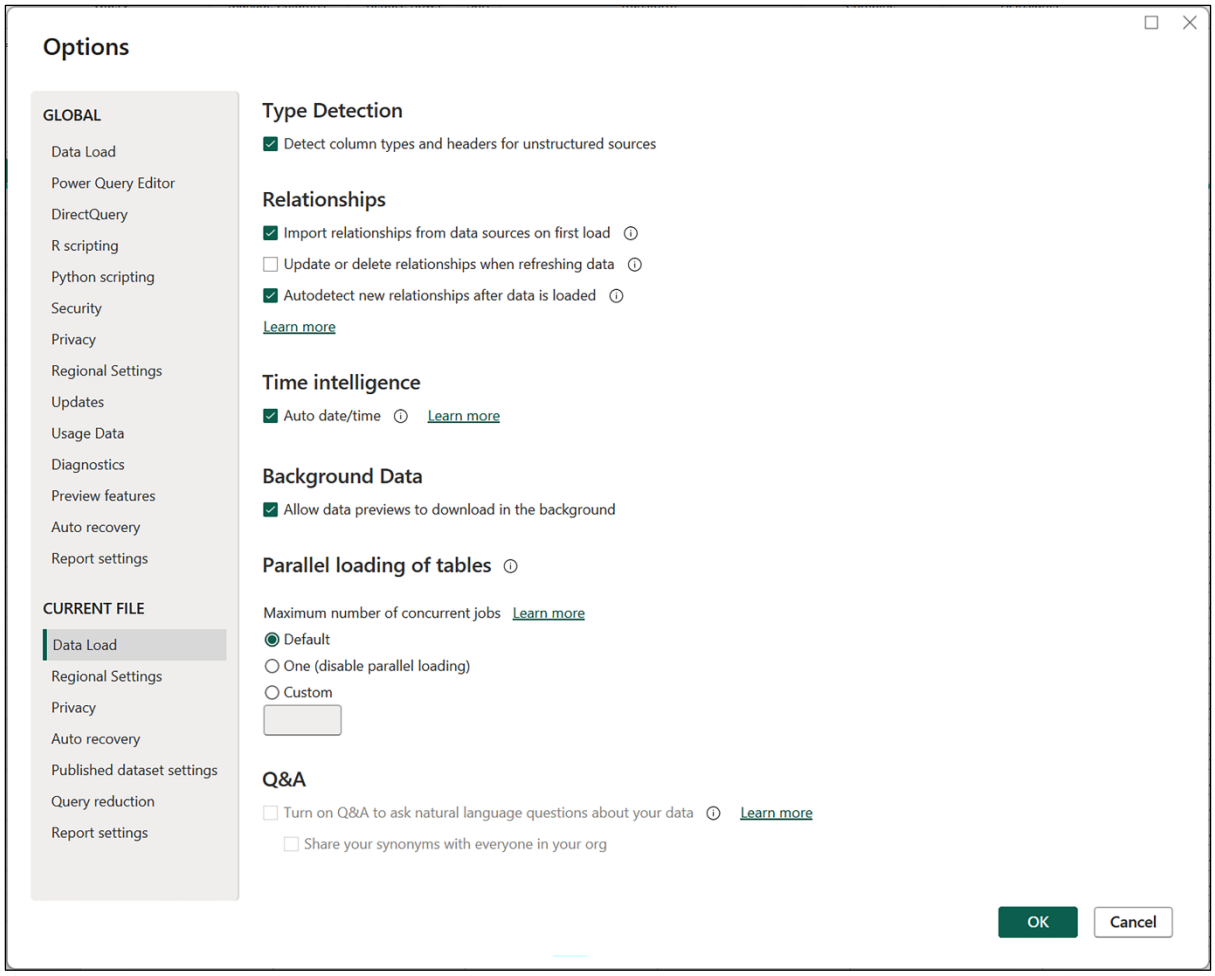

The Definitive Guide to Power Query - Power Query Editor
File
Home
Transform
Add Column
View
Tools
Help
Close & Apply
New Source
Recent Sources
Enter Data
Data source settings
Manage Parameters
Refresh Preview
Properties
Advanced Editor
Manage
Choose Columns
Remove Columns
Keep Rows
Remove Rows
Split Column
Close
New Query
Data Sources
Parameters
Query
Manage Columns
Reduce Rows
Sort
Queries [0]
Get Data
Search
All
File
Database
Microsoft Fabric (Preview)
Power Platform
Azure
Online Services
Other
All
Excel Workbook
Text/CSV
XML
JSON
Folder
PDF
Parquet
SharePoint folder
SQL Server database
Access database
SQL Server Analysis Services database
Oracle database
IBM Db2 database
IBM Informix database (Beta)
IBM Netezza
MySQL database
Certified Connectors
Template Apps
Connect
Cancel

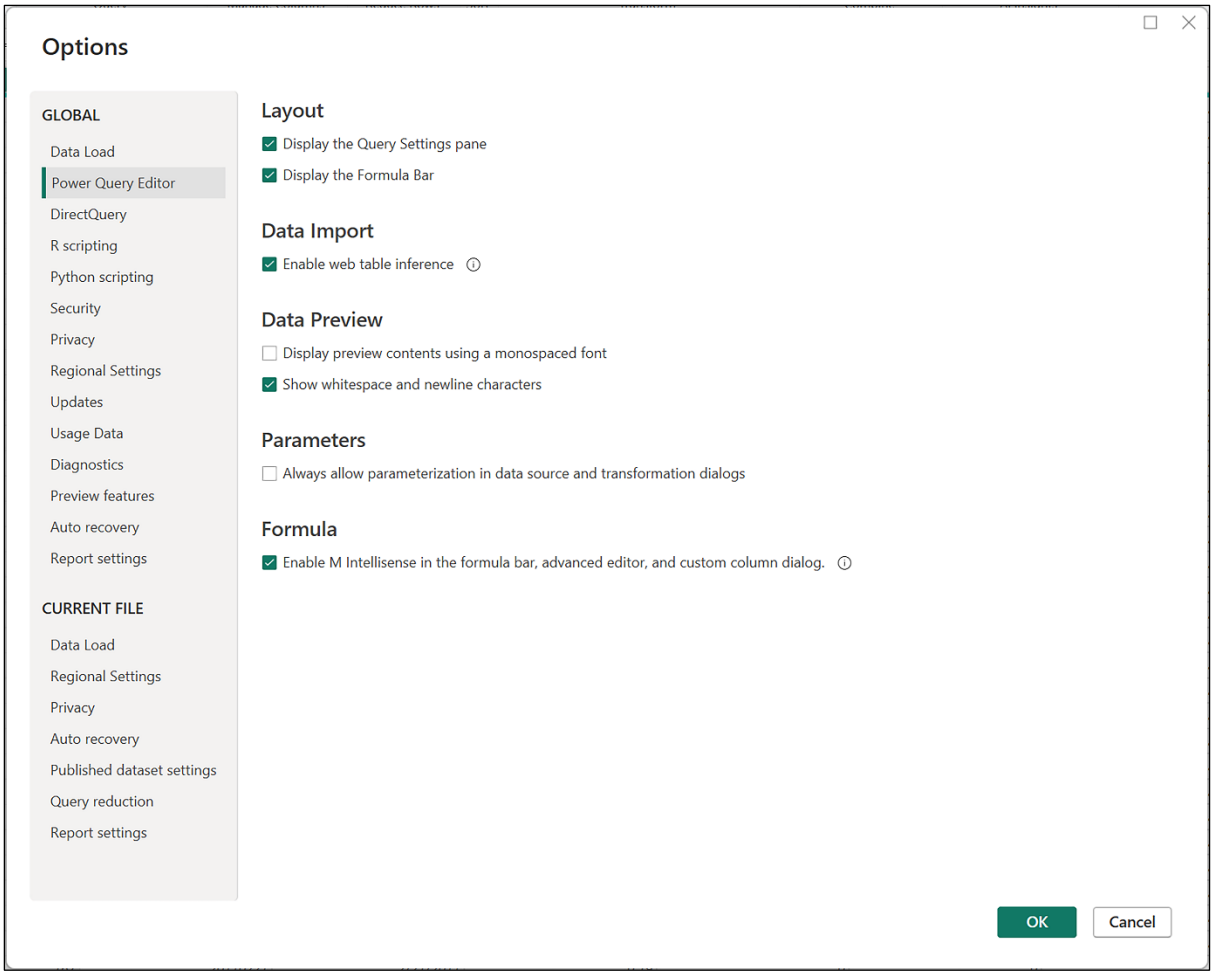

Chapter 2 - Product Inventory.csv
File Origin
1252: Western European (Windows)
Delimiter
Comma
Data Type Detection
Based on first 200 rows
ProductKey DateKey MovementDate UnitCost UnitsIn UnitsOut UnitsBalance EnglishProductName
167 20101228 12/28/2010 0.19 0 0 875 Metal Sheet 1
167 20101229 12/29/2010 0.19 0 0 875 Metal Sheet 1
167 20110119 1/19/2011 0.19 0 0 875 Metal Sheet 1
167 20110121 1/21/2011 0.19 0 0 875 Metal Sheet 1
167 20110122 1/22/2011 0.19 0 0 875 Metal Sheet 1
167 20110123 1/23/2011 0.19 0 0 875 Metal Sheet 1
167 20110124 1/24/2011 0.19 0 0 875 Metal Sheet 1
167 20110125 1/25/2011 0.19 0 0 875 Metal Sheet 1
167 20110126 1/26/2011 0.19 0 0 875 Metal Sheet 1
167 20110127 1/27/2011 0.19 0 0 875 Metal Sheet 1
167 20110128 1/28/2011 0.19 0 0 875 Metal Sheet 1
167 20110129 1/29/2011 0.19 0 0 875 Metal Sheet 1
167 20110130 1/30/2011 0.19 0 0 875 Metal Sheet 1
167 20110131 1/31/2011 0.19 0 0 875 Metal Sheet 1
167 20110201 2/1/2011 0.19 0 0 875 Metal Sheet 1
167 20110202 2/2/2011 0.19 0 0 875 Metal Sheet 1
167 20110203 2/3/2011 0.19 0 0 875 Metal Sheet 1
167 20110204 2/4/2011 0.19 0 0 875 Metal Sheet 1
167 20110205 2/5/2011 0.19 0 0 875 Metal Sheet 1
167 20110206 2/6/2011 0.19 0 0 875 Metal Sheet 1
The data in the preview has been truncated due to size limits.
Extract Table Using Examples
OK
Cancel

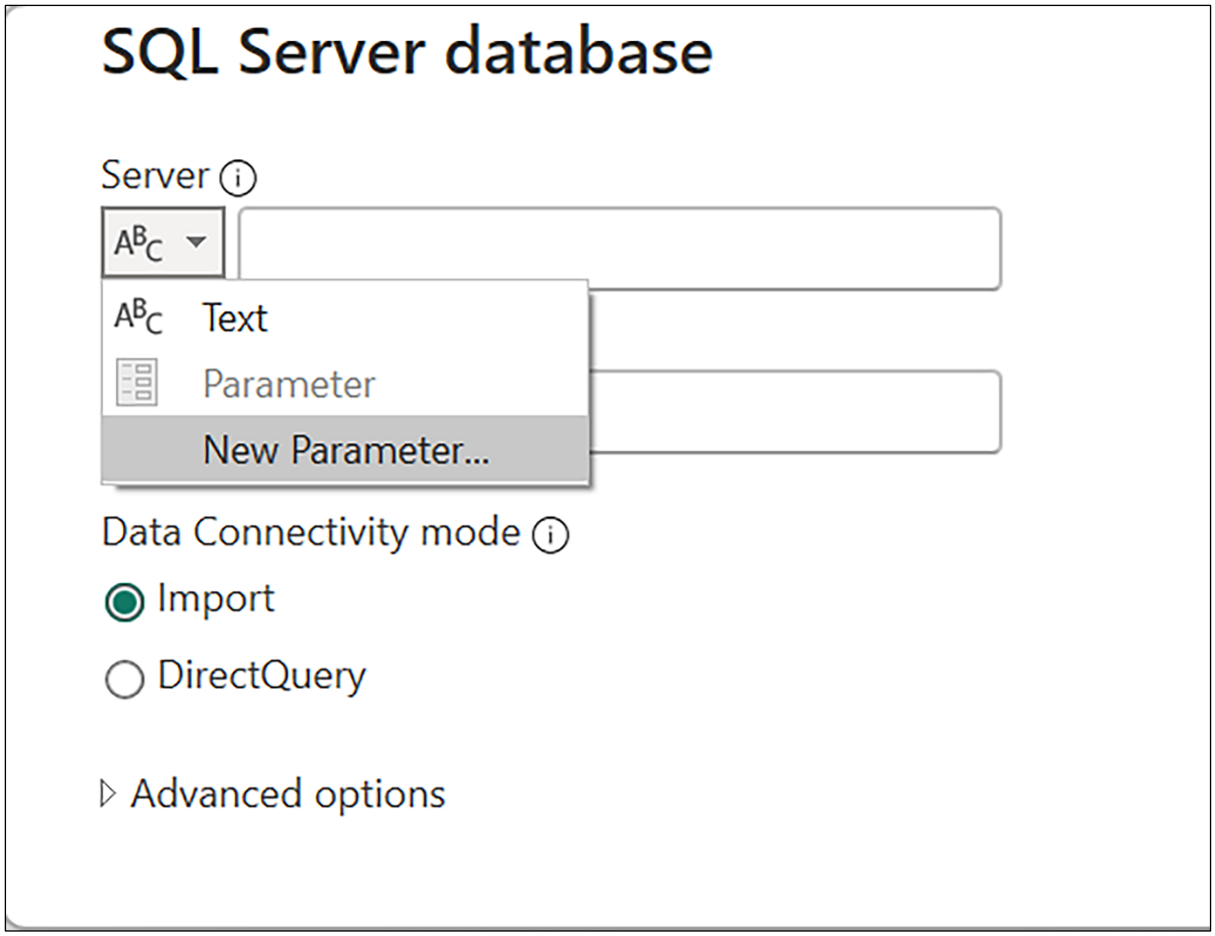

The Definitive Guide to Power Query M - Power Query Editor
File
Home
Transform
Add Column
View
Tools
Help
Close & Apply
New Source
Recent Sources
Enter Data
Data source settings
Manage Parameters
Refresh Preview
Properties
Advanced Editor
Manage
Manage Columns
Reduce Rows
Split Column
Group By
Data Type: Whole Number
Use First Row as Headers
Replace Values
Close
New Query
Data Sources
Parameters
Query
Sort
Transform
Queries [1]
Chapter 2 - Product Inventory
= Table.TransformColumnTypes(#"Promoted
ProductKey
DateKey
MovementDate
1 167 20101228
2 167 20101229
3 167 20110119
4 167 20110121
5 167 20110122
6 167 20110123
7 167 20110124
8 167 20110125
9 167 20110126
10 167 20110127
11
Query Settings
PROPERTIES
Name
Chapter 2 - Product Inventory
All Properties
APPLIED STEPS
Source
Promoted Headers
Changed Type
8 COLUMNS, 999+ ROWS
Column profiling based on top 1000 rows
PREVIEW DOWNLOADED AT 5:06 PM

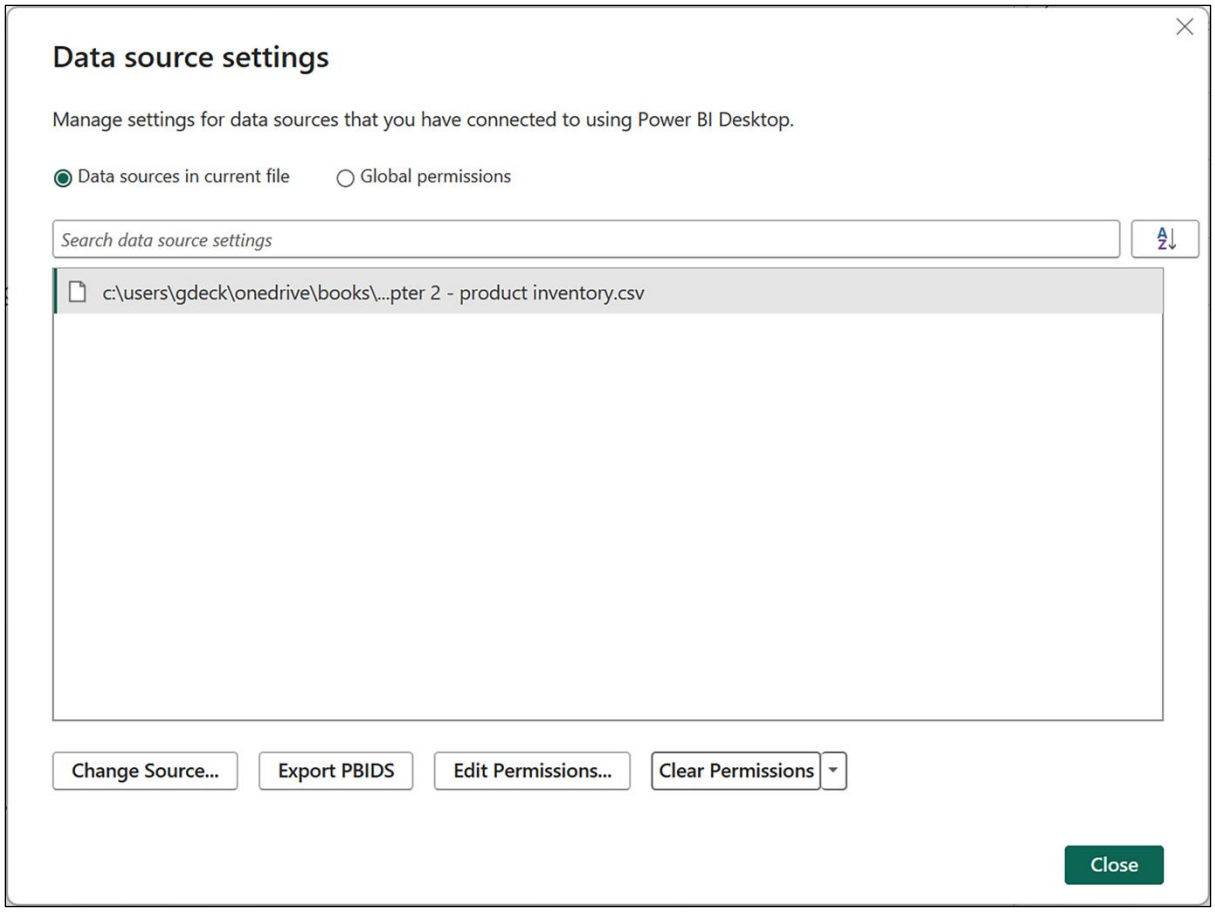

The Definitive Guide to Power Query - Power Query Editor
File
Close & Apply
Apply
Close
Save
Save As
Options and settings
Help
Options and settings
Options
Data source settings

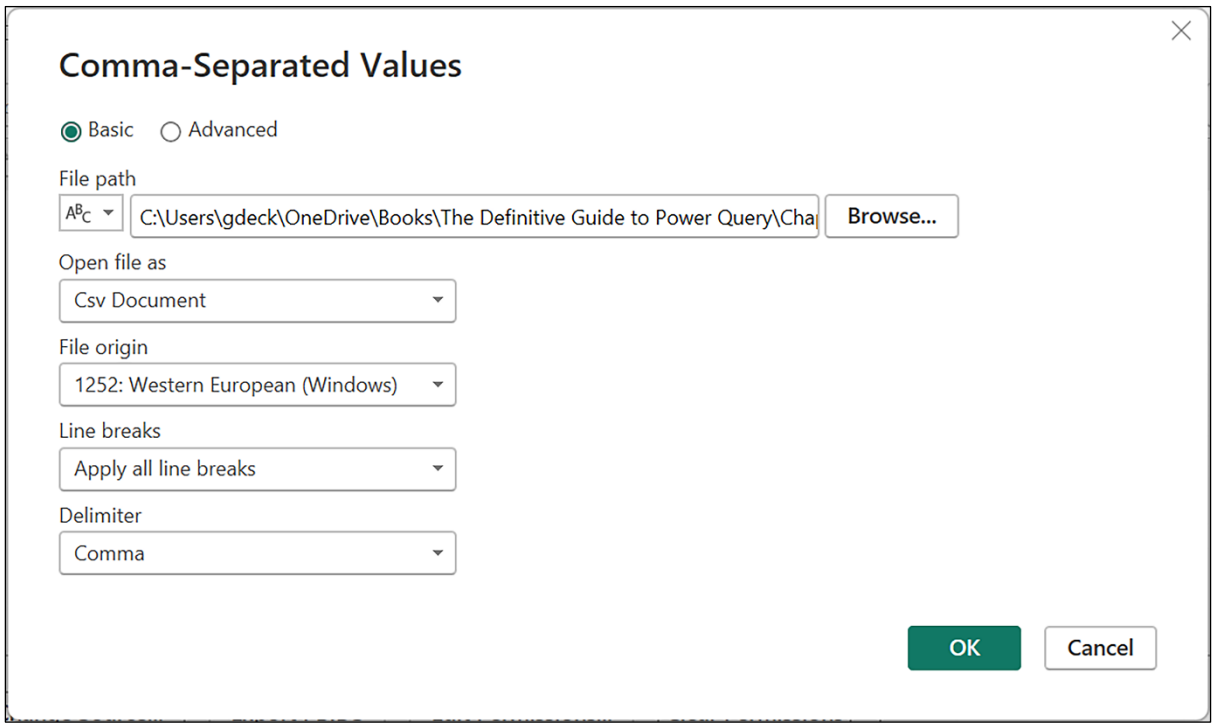

Options
GLOBAL
Data Load
Power Query Editor
DirectQuery
R scripting
Python scripting
Security
Privacy
Regional Settings
Updates
Usage Data
Diagnostics
Preview features
Auto recovery
Report settings
CURRENT FILE
Data Load
Regional Settings
Privacy
Auto recovery
Published dataset settings
Query reduction
Report settings
Type Detection
Always detect column types and headers for unstructured sources
Detect column types and headers for unstructured sources according to each file's setting
Never detect column types and headers for unstructured sources
Background Data
Always allow data previews to download in the background
Allow data previews to download in the background according to each file's setting
Never allow data previews to download in the background
Parallel loading of tables
When you load data into Power BI (via import or DirectQuery), each data table is backed by a Power Query query. These queries are evaluated simultaneously instead of one-by-one, which can speed up the process. In certain situations, you might want to adjust the default number of simultaneous query evaluations and memory used. Learn more
Maximum number of simultaneous evaluations
12
Maximum memory used per simultaneous evaluation (MB)
432
Time intelligence
Auto date/time for new files
Learn more
Data Cache Management Options
Currently used: 156 MB
Clear Cache
Maximum allowed (MB):
4096
Restore Defaults
OK
Cancel

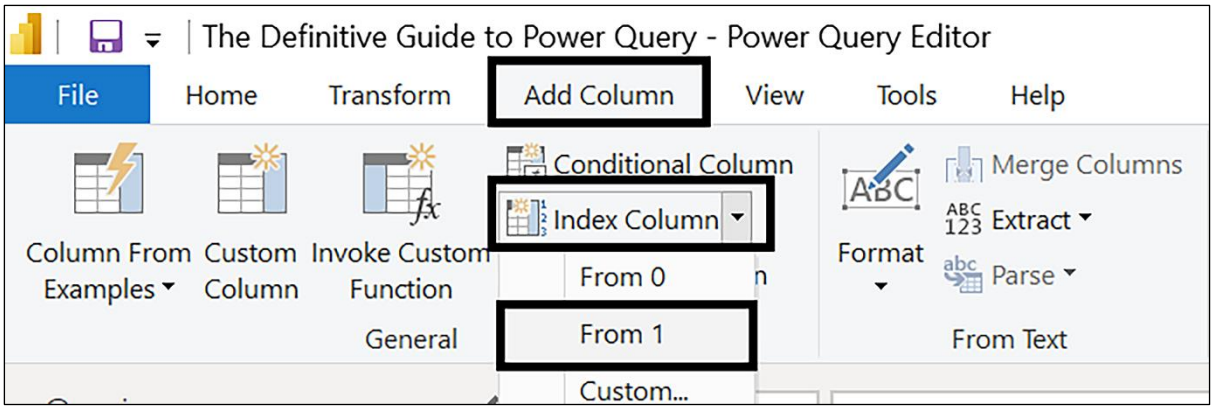

Options
GLOBAL
Data Load
Power Query Editor
DirectQuery
R scripting
Python scripting
Security
Privacy
Regional Settings
Updates
Usage Data
Diagnostics
Preview features
Auto recovery
Report settings
CURRENT FILE
Data Load
Regional Settings
Privacy
Auto recovery
Published dataset settings
Query reduction
Report settings
Type Detection
Detect column types and headers for unstructured sources
Relationships
Import relationships from data sources on first load
Update or delete relationships when refreshing data
Autodetect new relationships after data is loaded
Learn more
Time intelligence
Auto date/time
Learn more
Background Data
Allow data previews to download in the background
Parallel loading of tables
Maximum number of concurrent jobs
Learn more
Default
One (disable parallel loading)
Custom
Q&A
Turn on Q&A to ask natural language questions about your data
Learn more
Share your synonyms with everyone in your org
OK
Cancel

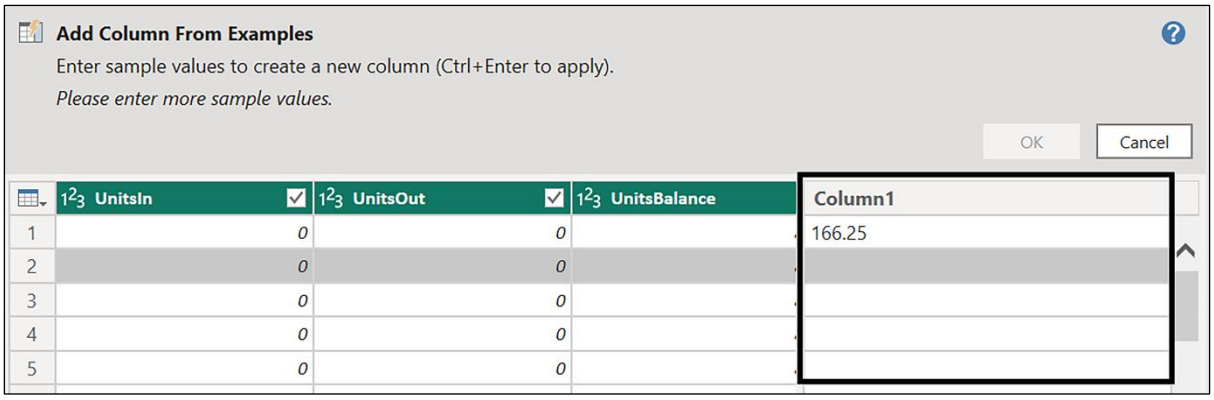

Options
GLOBAL
Data Load
Power Query Editor
DirectQuery
R scripting
Python scripting
Security
Privacy
Regional Settings
Updates
Usage Data
Diagnostics
Preview features
Auto recovery
Report settings
CURRENT FILE
Data Load
Regional Settings
Privacy
Auto recovery
Published dataset settings
Query reduction
Report settings
Layout
Display the Query Settings pane
Display the Formula Bar
Data Import
Enable web table inference
Data Preview
Display preview contents using a monospaced font
Show whitespace and newline characters
Parameters
Always allow parameterization in data source and transformation dialogs
Formula
Enable M Intellisense in the formula bar, advanced editor, and custom column dialog.
OK
Cancel

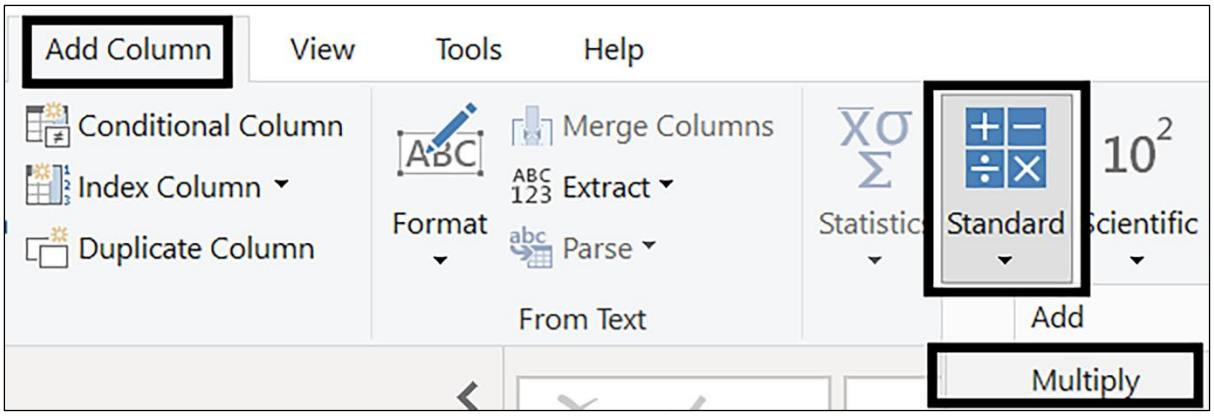

SQL Server database
Server
Text
Parameter
New Parameter...
Data Connectivity mode
Import
DirectQuery
Advanced options

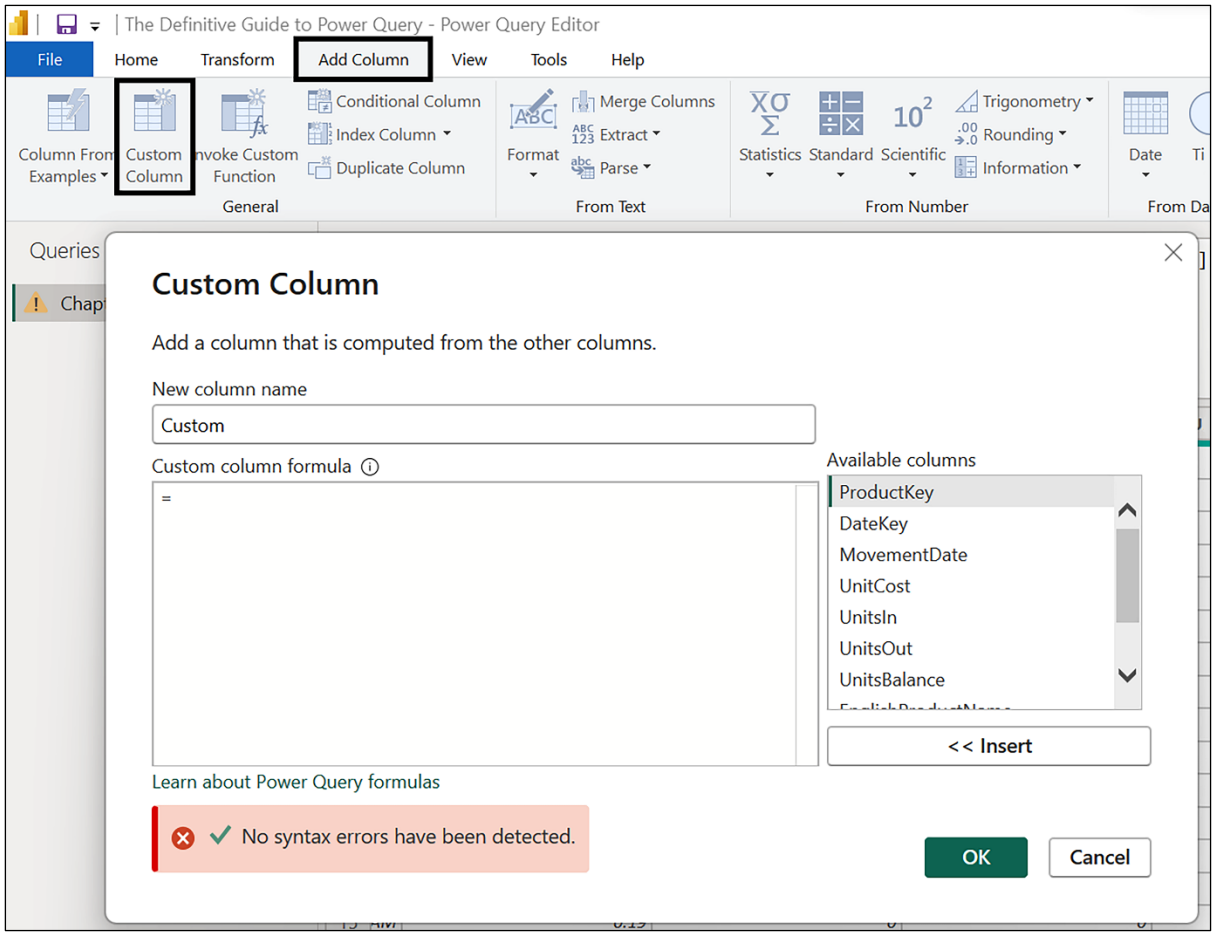

Data source settings
Manage settings for data sources that you have connected to using Power BI Desktop.
Data sources in current file
Global permissions
Search data source settings
c:\users\gdeck\onedrive\books\...pter 2 - product inventory.csv
Change Source...
Export PBIDS
Edit Permissions...
Clear Permissions
Close

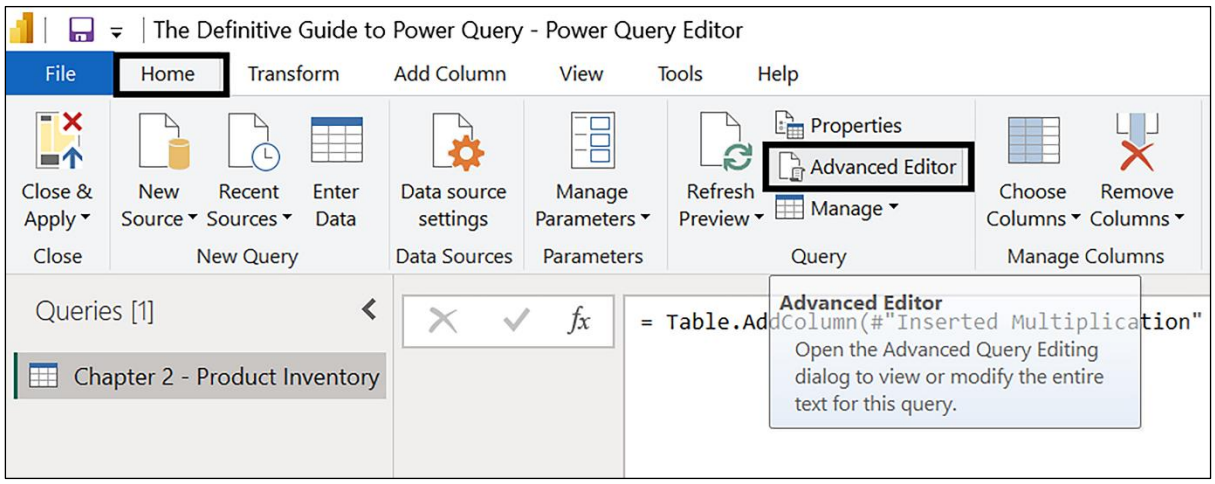

Comma-Separated Values
Basic
Advanced
File path
C:\Users\gdeck\OneDrive\Books\The Definitive Guide to Power Query\Cha
Browse...
Open file as
Csv Document
File origin
1252: Western European (Windows)
Line breaks
Apply all line breaks
Delimiter
Comma
OK
Cancel

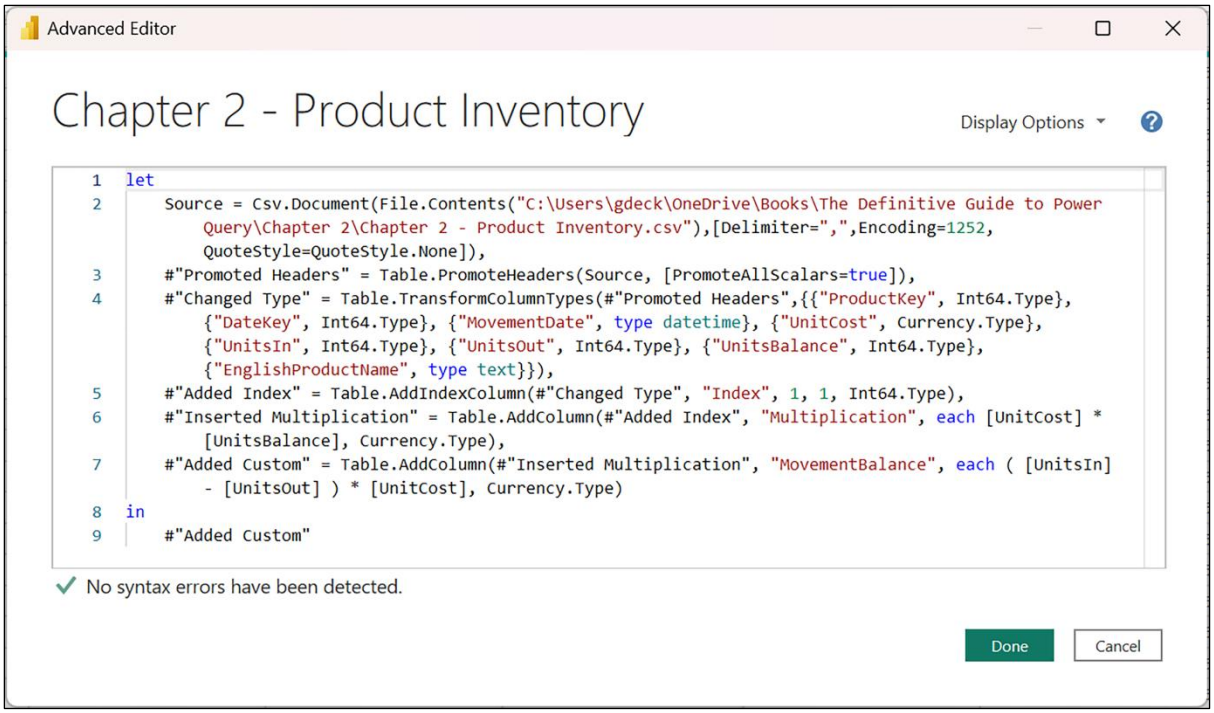

The Definitive Guide to Power Query - Power Query Editor
File
Home
Transform
Add Column
View
Tools
Help
Column From Examples
Custom Column
Invoke Custom Function
General
Conditional Column
Index Column
From 0
From 1
Custom...
Format
Merge Columns
Extract
Parse
From Text


Add Column From Examples
Enter sample values to create a new column (Ctrl+Enter to apply).
Please enter more sample values.
OK
Cancel
UnitsIn
UnitsOut
UnitsBalance
Column1
166.25


Add Column
View
Tools
Help
Conditional Column
Index Column
Duplicate Column
Format
Merge Columns
Extract
Parse
From Text
Statistic
Standard
Scientific
Add
Multiply


The Definitive Guide to Power Query - Power Query Editor
File
Home
Transform
Add Column
View
Tools
Help
Column From Examples
Custom Column
Invoke Custom Function
Conditional Column
Index Column
Duplicate Column
General
Format
Merge Columns
Extract
Parse
From Text
Statistics
Standard
Scientific
Trigonometry
Rounding
Information
From Number
Date
From Da
Queries
Custom Column
Add a column that is computed from the other columns.
New column name
Custom
Custom column formula
=
Available columns
ProductKey
DateKey
MovementDate
UnitCost
UnitsIn
UnitsOut
UnitsBalance
<< Insert
Learn about Power Query formulas
No syntax errors have been detected.
OK
Cancel


The Definitive Guide to Power Query - Power Query Editor
File
Home
Transform
Add Column
View
Tools
Help
Close & Apply
New Source
Recent Sources
Enter Data
Close
New Query
Data source settings
Data Sources
Manage Parameters
Parameters
Refresh Preview
Properties
Advanced Editor
Manage
Query
Choose Columns
Remove Columns
Manage Columns
Queries [1]
Chapter 2 - Product Inventory
= Table.AddColumn(#"Inserted Multiplication"
Advanced Editor
Open the Advanced Query Editing dialog to view or modify the entire text for this query.


Advanced Editor
Chapter 2 - Product Inventory
Display Options
let
Source = Csv.Document(File.Contents("C:\Users\gdeck\OneDrive\Books\The Definitive Guide to Power Query\Chapter 2\Chapter 2 - Product Inventory.csv"),[Delimiter=",",Encoding=1252, QuoteStyle=QuoteStyle.None]),
#"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
#"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"ProductKey", Int64.Type}, {"DateKey", Int64.Type}, {"MovementDate", type datetime}, {"UnitCost", Currency.Type}, {"UnitsIn", Int64.Type}, {"UnitsOut", Int64.Type}, {"UnitsBalance", Int64.Type}, {"EnglishProductName", type text}}),
#"Added Index" = Table.AddIndexColumn(#"Changed Type", "Index", 1, 1, Int64.Type),
#"Inserted Multiplication" = Table.AddColumn(#"Added Index", "Multiplication", each [UnitCost] * [UnitsBalance], Currency.Type),
#"Added Custom" = Table.AddColumn(#"Inserted Multiplication", "MovementBalance", each ([UnitsIn] - [UnitsOut]) * [UnitCost], Currency.Type)
in
#"Added Custom"
No syntax errors have been detected.
Done
Cancel


Advanced Editor
Chapter 2 - Product Inventory
Display Options
let
Source =
Csv.Document(
File.Contents(Parameter1),
[Delimiter=",",Encoding=1252, QuoteStyle=QuoteStyle.None]
),
promoteHeaders = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
changeTypes =
Table.TransformColumnTypes(
promoteHeaders,
{
{"ProductKey", Int64.Type},
{"DateKey", Int64.Type},
{"MovementDate", type datetime},
{"UnitCost", Currency.Type},
{"UnitsIn", Int64.Type},
{"UnitsOut", Int64.Type},
{"UnitsBalance", Int64.Type},
{"EnglishProductName", type text}
}
),
addIndex = Table.AddIndexColumn(changeTypes, "Index", 1, 1, Int64.Type),
#"Inserted Multiplication" = Table.AddColumn(addIndex, "Multiplication",
each [UnitCost] * [UnitsBalance], type number),
addInventoryBalanceColumn =
Table.AddColumn(
#"Inserted Multiplication",
"InventoryBalance",
each [UnitCost] * [UnitsBalance],
Currency.Type
),
addMovementBalanceColumn =
Table.AddColumn(
addInventoryBalanceColumn,
"MovementBalance",
each ([UnitsIn] - [UnitsOut]) * [UnitCost],
Currency.Type
)
in
addMovementBalanceColumn
No syntax errors have been detected.
Done
Cancel

Chapter 3: Accessing and Combining Data

File.Contents

Returns the contents of the file, `path`, as binary. The `options` parameter is currently intended for internal use only.

Enter Parameters

path

Example: abc

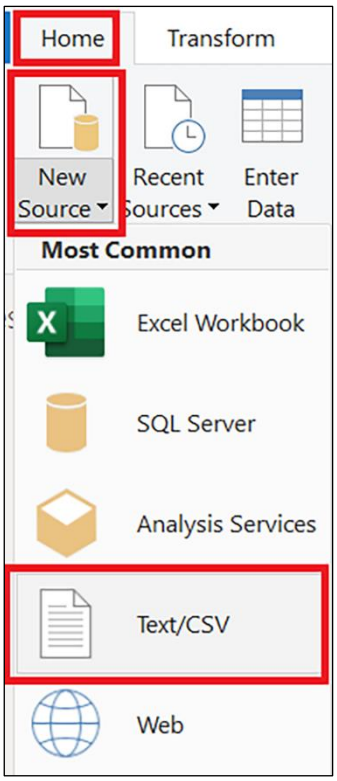
options (optional)

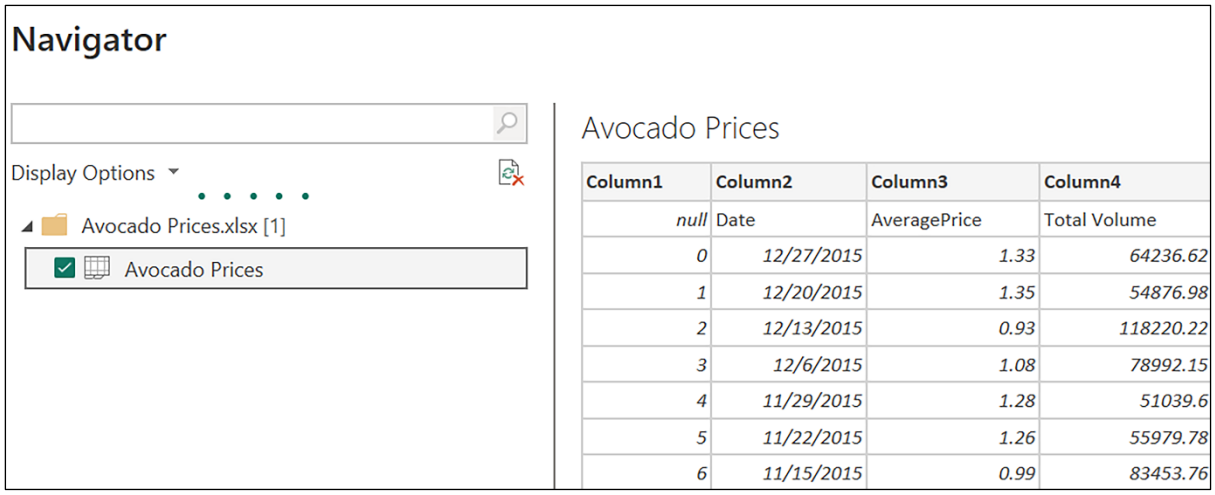
PreserveLastAccessTimes (optional)

Example: true

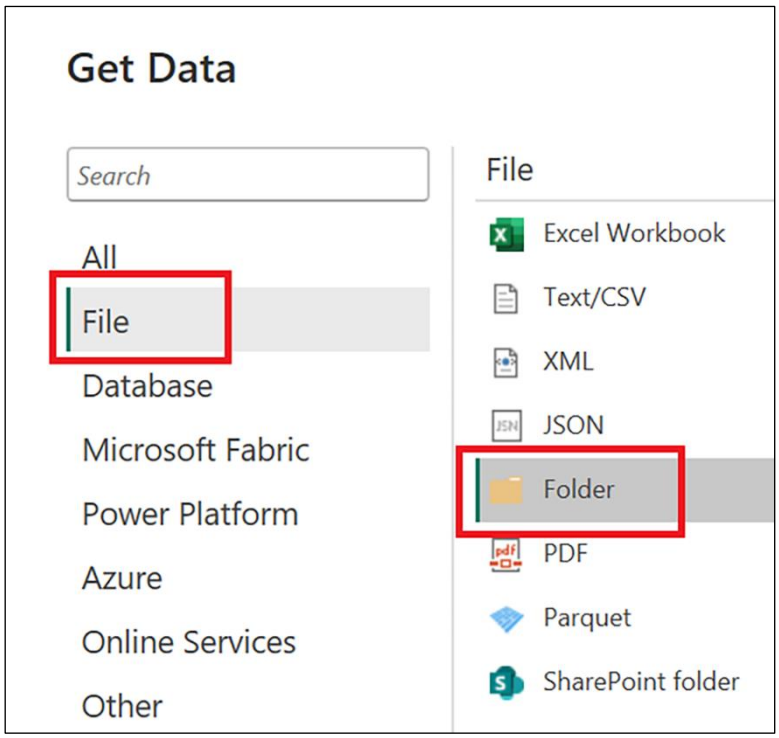
Invoke Clear

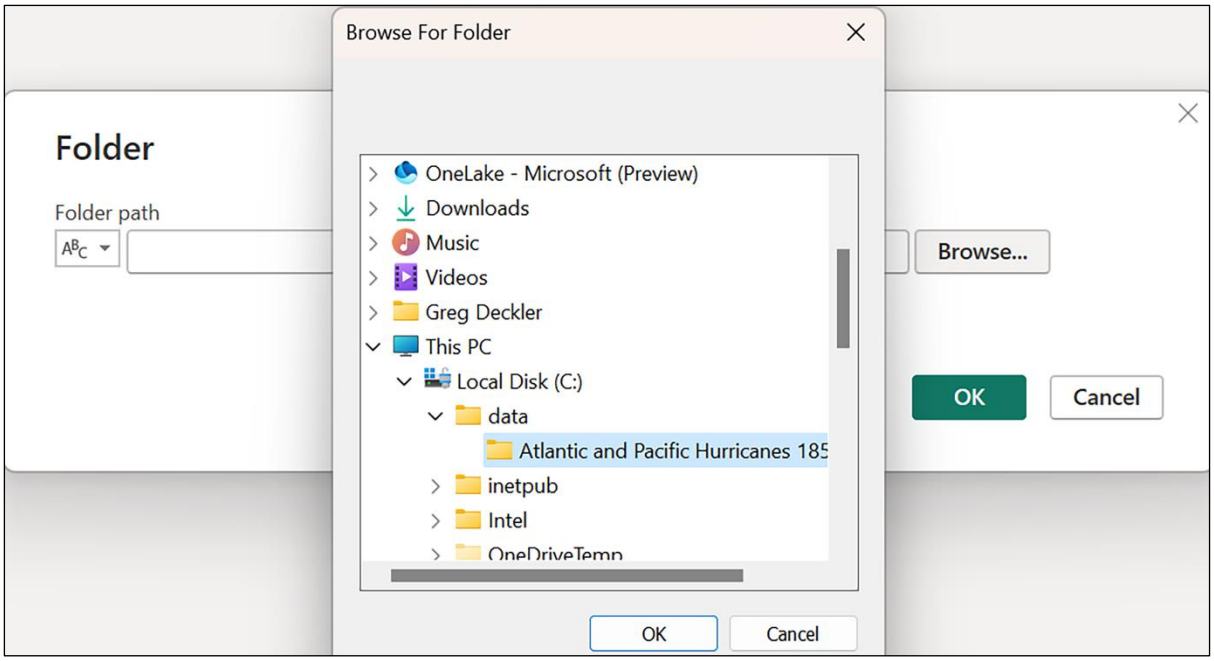
function (`path` as text, *optional* `options` as nullable record) as binary

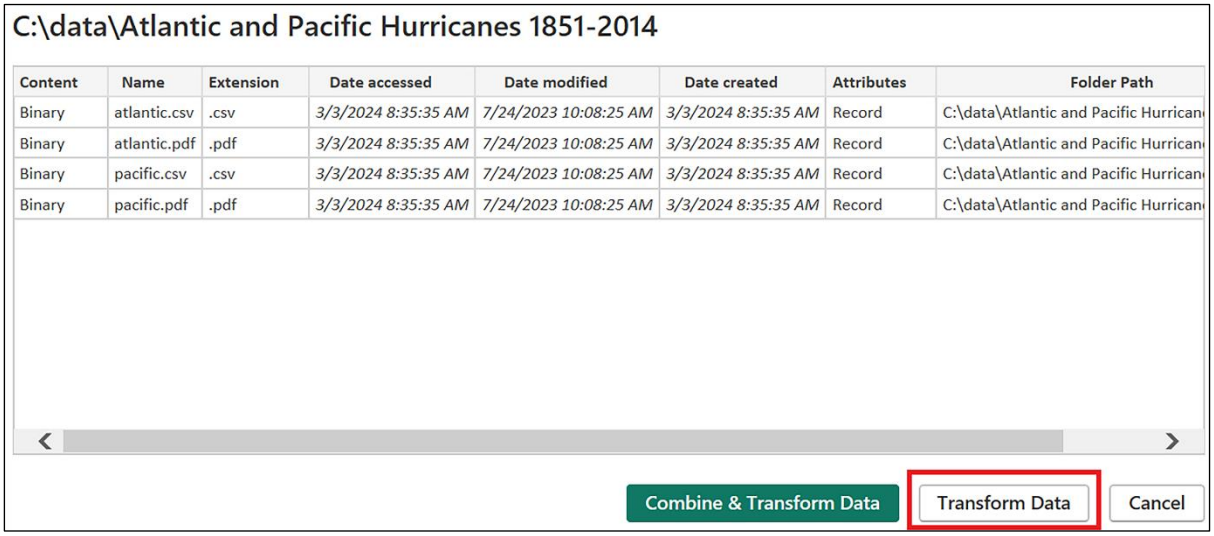


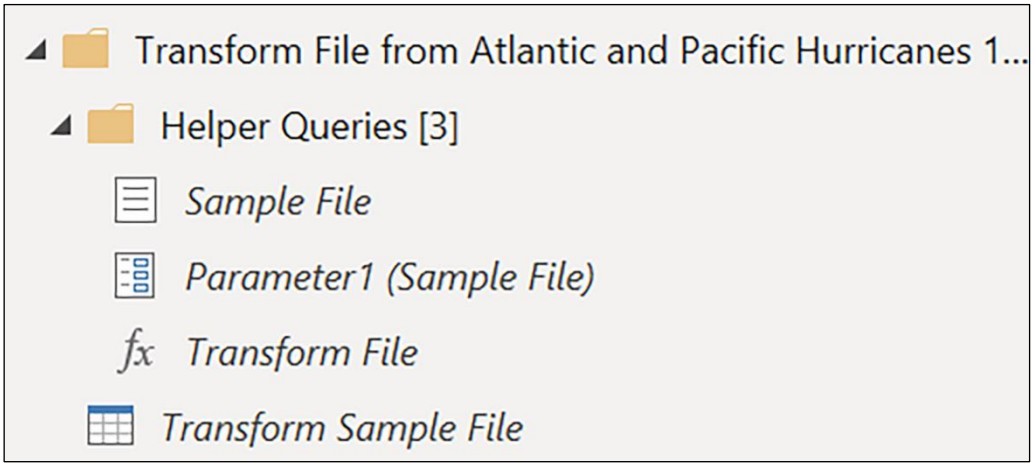


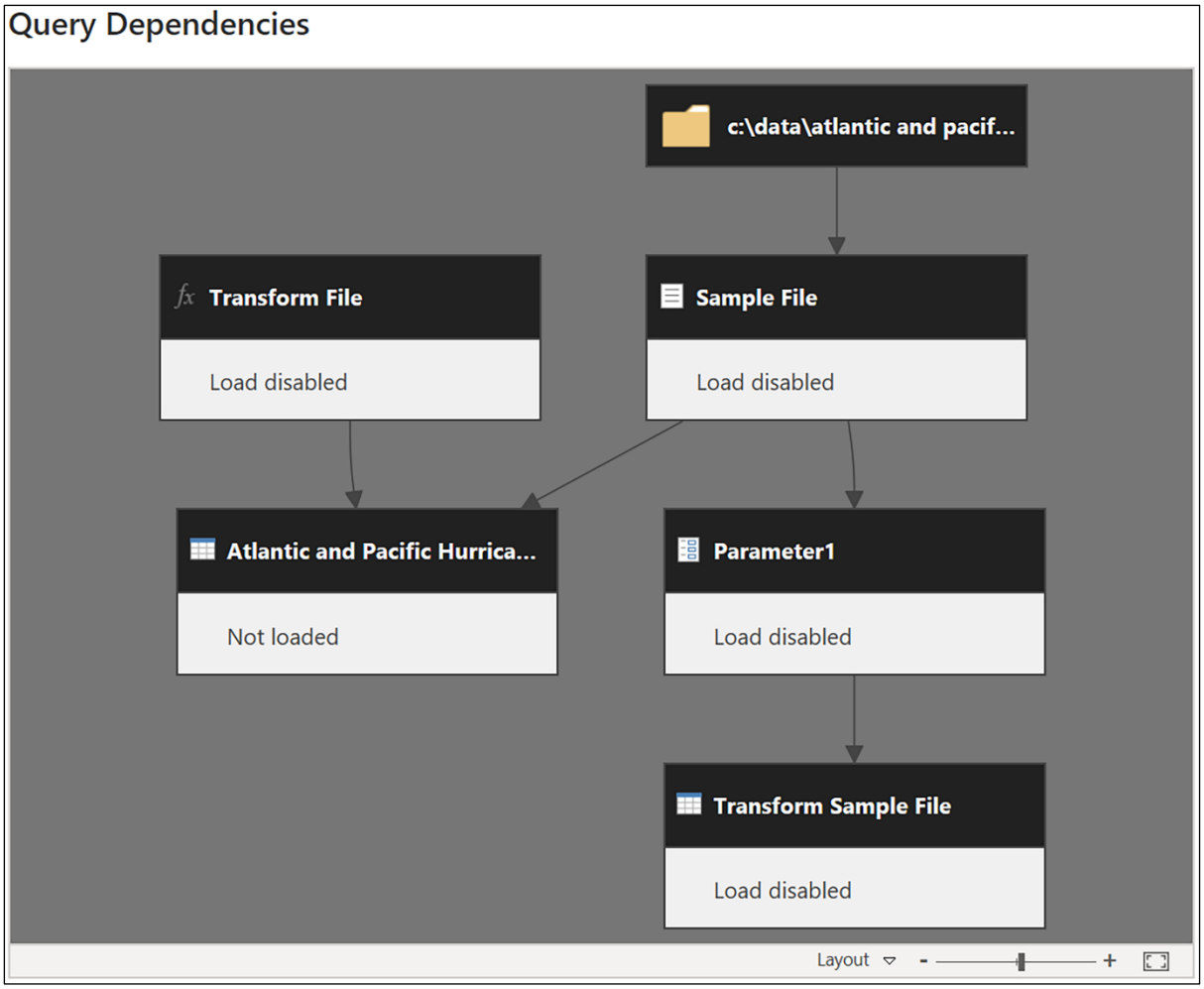
Column1	Column2	Column3	Column4
null	Date	AveragePrice	Total Volume
0	12/27/2015	1.33	64236.62
1	12/20/2015	1.35	54876.98
2	12/13/2015	0.93	118220.22
3	12/6/2015	1.08	78992.15
4	11/29/2015	1.28	51039.6
5	11/22/2015	1.26	55979.78
6	11/15/2015	0.99	83453.76

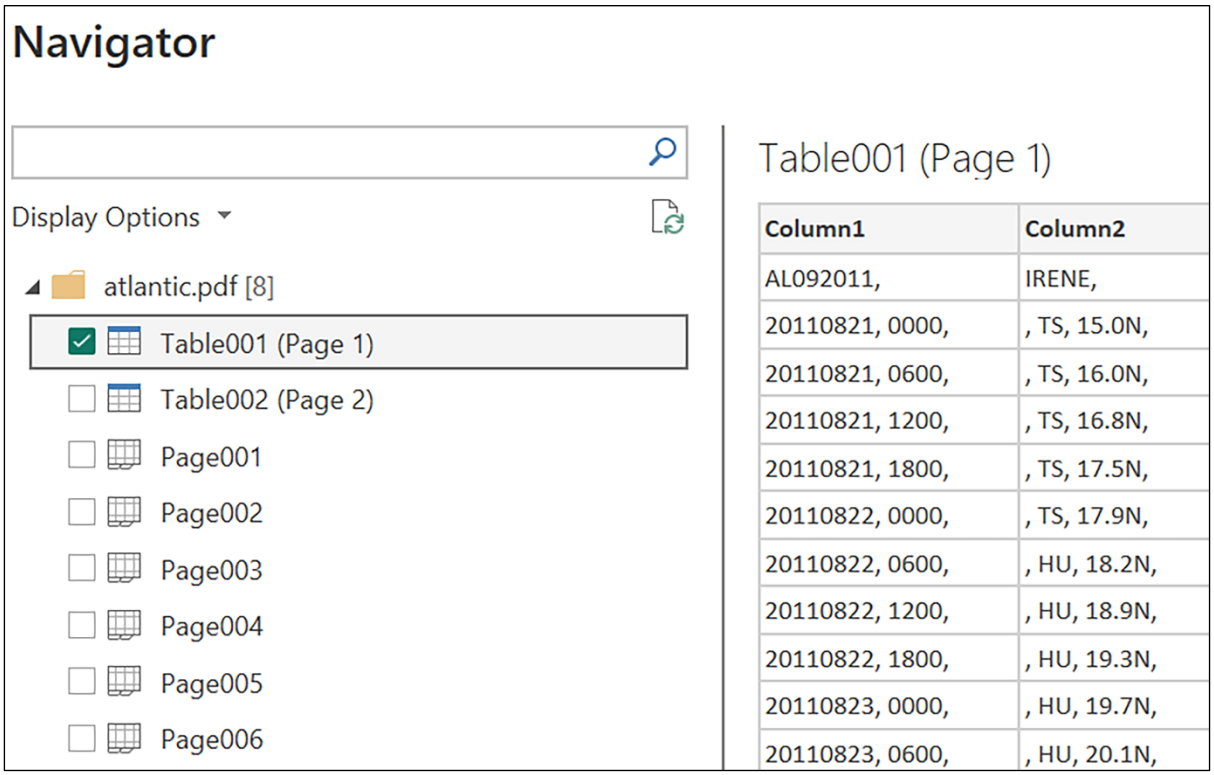



Browse For Folder
Folder
Folder path
Browse...
OK
Cancel
OneLake - Microsoft (Preview)
Downloads
Music
Videos
Greg Deckler
This PC
Local Disk (C:)
data
Atlantic and Pacific Hurricanes 185
inetpub
Intel
OneDriveTemp
OK
Cancel

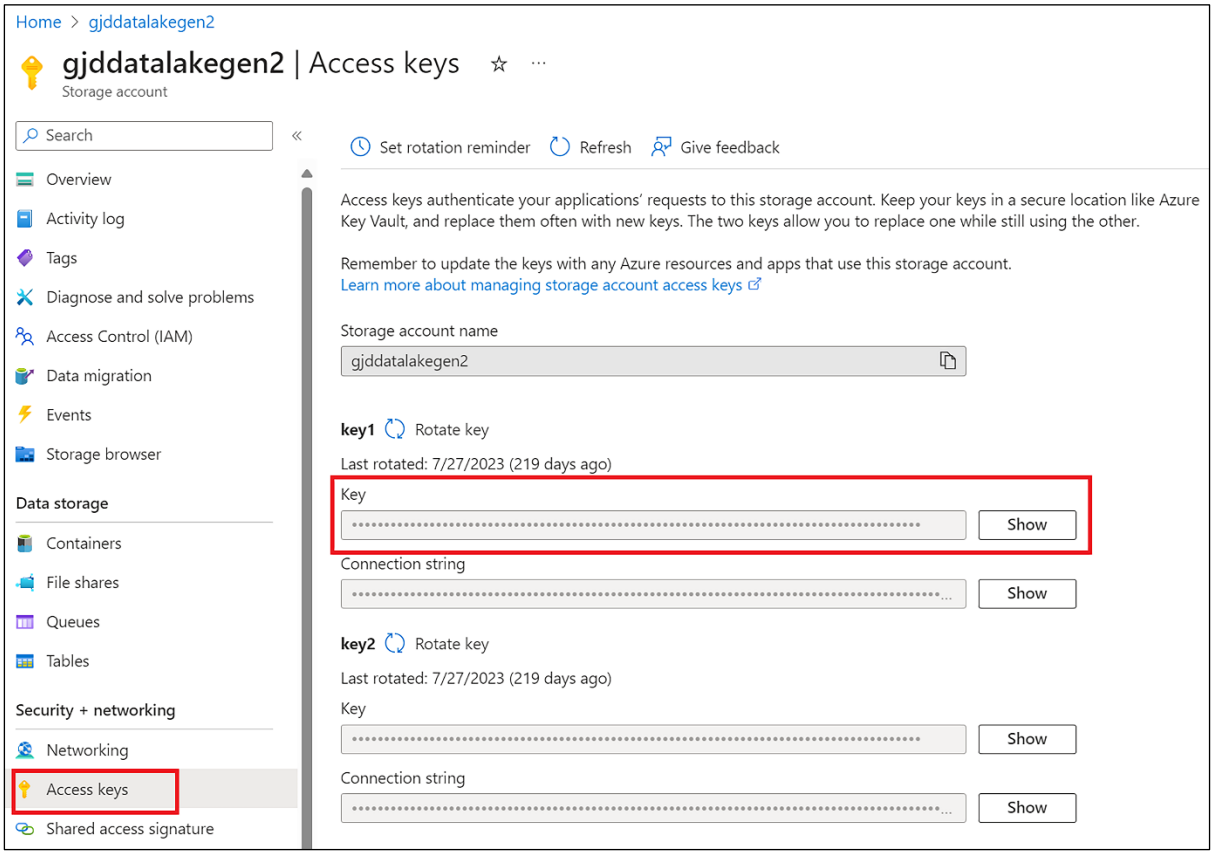

C:\data\Atlantic and Pacific Hurricanes 1851-2014
Content Name Extension Date accessed Date modified Date created Attributes Folder Path
Binary atlantic.csv .csv 3/3/2024 8:35:35 AM 7/24/2023 10:08:25 AM 3/3/2024 8:35:35 AM Record C:\data\Atlantic and Pacific Hurrican
Binary atlantic.pdf .pdf 3/3/2024 8:35:35 AM 7/24/2023 10:08:25 AM 3/3/2024 8:35:35 AM Record C:\data\Atlantic and Pacific Hurrican
Binary pacific.csv .csv 3/3/2024 8:35:35 AM 7/24/2023 10:08:25 AM 3/3/2024 8:35:35 AM Record C:\data\Atlantic and Pacific Hurrican
Binary pacific.pdf .pdf 3/3/2024 8:35:35 AM 7/24/2023 10:08:25 AM 3/3/2024 8:35:35 AM Record C:\data\Atlantic and Pacific Hurrican
Combine & Transform Data
Transform Data
Cancel


Transform File from Atlantic and Pacific Hurricanes 1...
Helper Queries [3]
Sample File
Parameter1 (Sample File)
Transform File
Transform Sample File

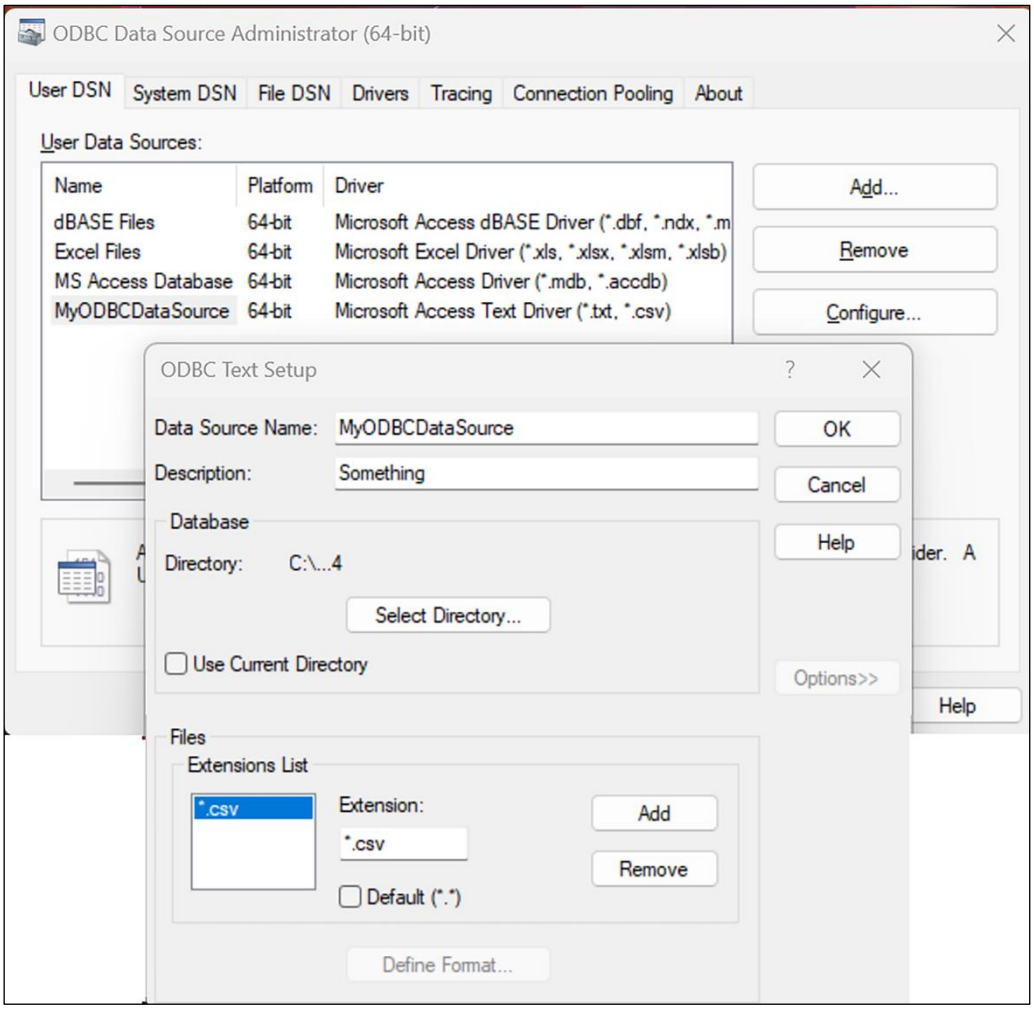


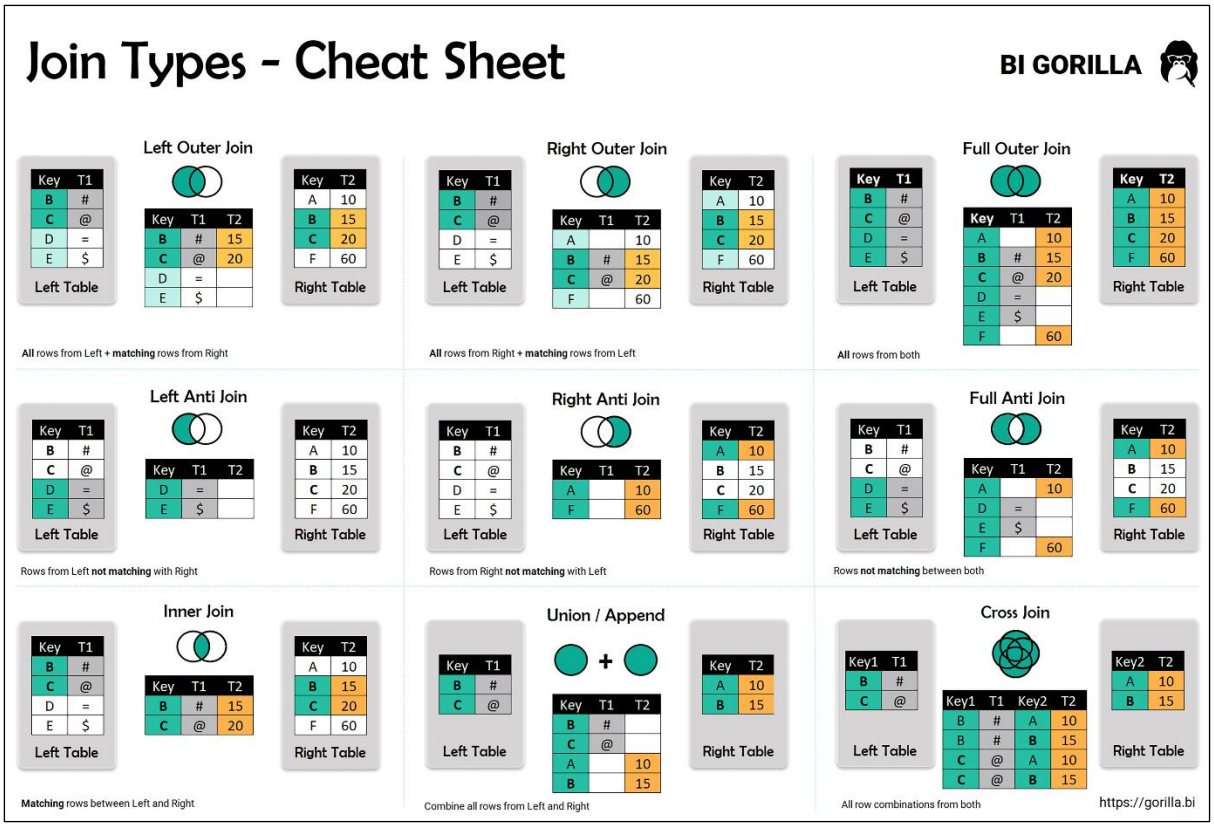


Column1	Column2
AL092011,	IRENE,
20110821, 0000,	, TS, 15.0N,
20110821, 0600,	, TS, 16.0N,
20110821, 1200,	, TS, 16.8N,
20110821, 1800,	, TS, 17.5N,
20110822, 0000,	, TS, 17.9N,
20110822, 0600,	, HU, 18.2N,
20110822, 1200,	, HU, 18.9N,
20110822, 1800,	, HU, 19.3N,
20110823, 0000,	, HU, 19.7N,
20110823, 0600,	, HU, 20.1N,



	List
1	{"Column1": "One", "Column2": "1"}
2	{"Column1": "Two", "Column2": "2"}
3	{"Column1": "Three", "Column2": "3"}


ODBC Data Source Administrator (64-bit)
User DSN
System DSN
File DSN
Drivers
Tracing
Connection Pooling
About
User Data Sources:
Name
Platform
Driver
dBASE Files
64-bit
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.m
Excel Files
64-bit
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)
MS Access Database
64-bit
Microsoft Access Driver (*.mdb, *.accdb)
MyODBCDataSource
64-bit
Microsoft Access Text Driver (*.txt, *.csv)
Add...
Remove
Configure...
ODBC Text Setup
Data Source Name:
MyODBCDataSource
Description:
Something
OK
Cancel
Help
Database
Directory:
C:\...4
Select Directory...
Use Current Directory
Options>>
Help
Files
Extensions List
*.csv
Extension:
*.csv
Add
Remove
Default (*.*)
Define Format...



	Column0	atlanticPDFTable1
1	AL092011,	Table
2	20110821, 0000,	Table
3	20110821, 0600,	Table
4	20110821, 1200,	Table
5	20110821, 1800,	Table
6	20110822, 0000,	Table
7	20110822, 0600,	Table
8	20110822, 1200,	Table
9	20110822, 1800,	Table
10	20110823, 0000,	Table

	Column0	Column1	Column2	Column3	Column4	Co
1	AL092011,	AL092011,	IRENE,	39,	*null*	
2	20110821, 0000,	20110821, 0000,	, TS, 15.0N,	59.0W,	*45*	
3	20110821, 0600,	20110821, 0600,	, TS, 16.0N,	60.6W,	*45*	
4	20110821, 1200,	20110821, 1200,	, TS, 16.8N,	62.2W,	*45*	
5	20110821, 1800,	20110821, 1800,	, TS, 17.5N,	63.7W,	*50*	
6	20110822, 0000,	20110822, 0000,	, TS, 17.9N,	65.0W,	*60*	
7	20110822, 0600,	20110822, 0600,	, HU, 18.2N,	65.9W,	*65*	
8	20110822, 1200,	20110822, 1200,	, HU, 18.9N,	67.0W,	*70*	
9	20110822, 1800,	20110822, 1800,	, HU, 19.3N,	68.0W,	*75*	
10	20110823, 0000,	20110823, 0000,	, HU, 19.7N,	68.8W,	*80*	

Chapter 4: Understanding Values and Expressions

	Author	Country
1	Brian	USA
2	Greg	USA
3	Melissa	NED
4	Rick	NED

```
let
    Source = Table.FromRows(Json.Document(Binary.Decompress(Binary.FromText
        ("i45WcirKTMxT0lEKDXZUitWJVnIvSk1H4vqm5mQWFycCRfxcXcAiQZnJ2TBuLAA=", BinaryEncoding.Base64),
        Compression.Deflate)), let _t = ((type nullable text) meta [Serialized.Text = true]) in type table [Author
        = _t, Country = _t])
in
    Source
```

```
= #binary({211, 93, 116}) = #binary("0110")
```

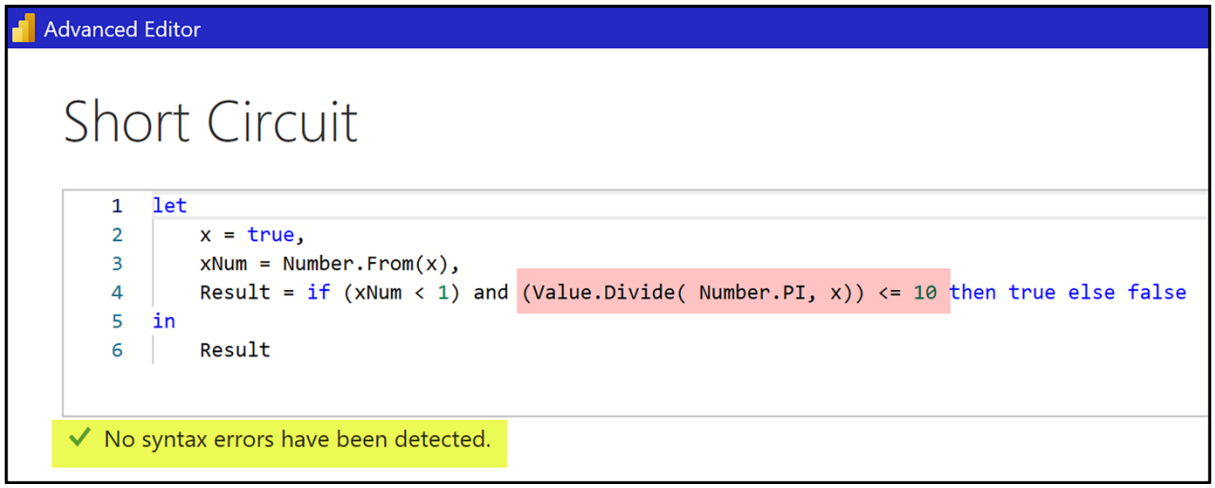
TRUE

```
= {#date(2023, 1, 1)..#date(2023, 1, 7)}
```

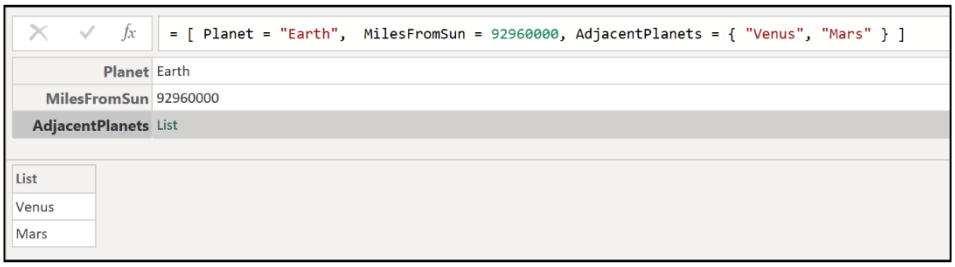
Expression.Error: We cannot apply operator .. to types Date and Date.
Details:
Operator=..
Left=1/1/2023
Right=1/7/2023

```
= List.Transform( { Number.From( #date(2023, 1, 1)).. Number.From( #date(2023, 1, 7) ) }, each Date.From(_) )
```

	List
1	1/1/2023
2	1/2/2023
3	1/3/2023
4	1/4/2023
5	1/5/2023
6	1/6/2023
7	1/7/2023



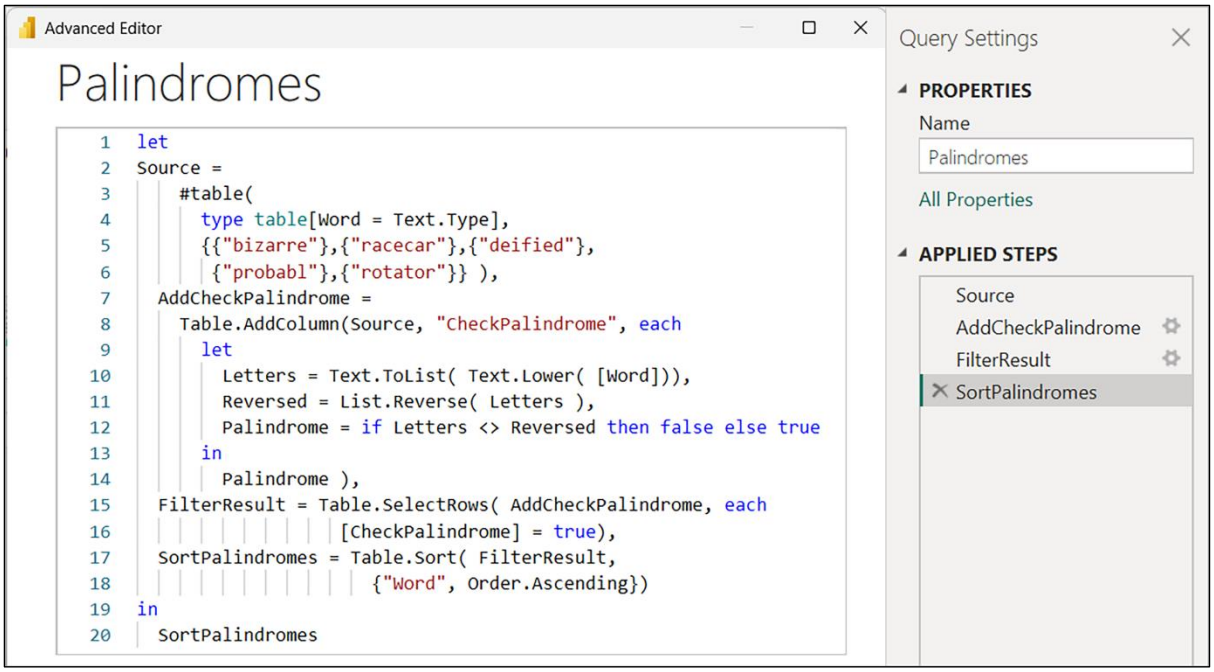
Expression	Description	Result
2.998	Fractional number	*2.998*
-3.2	Fractional number	*-3.2*
1.00e+3	Fractional number with exponent	*1000*
1.0e-3	Fractional number wiht exponent	*0.001*
36	Whole number	*36*
2e4	Whole number with exponent	*20000*
0x62	Whole number in hex	*98*

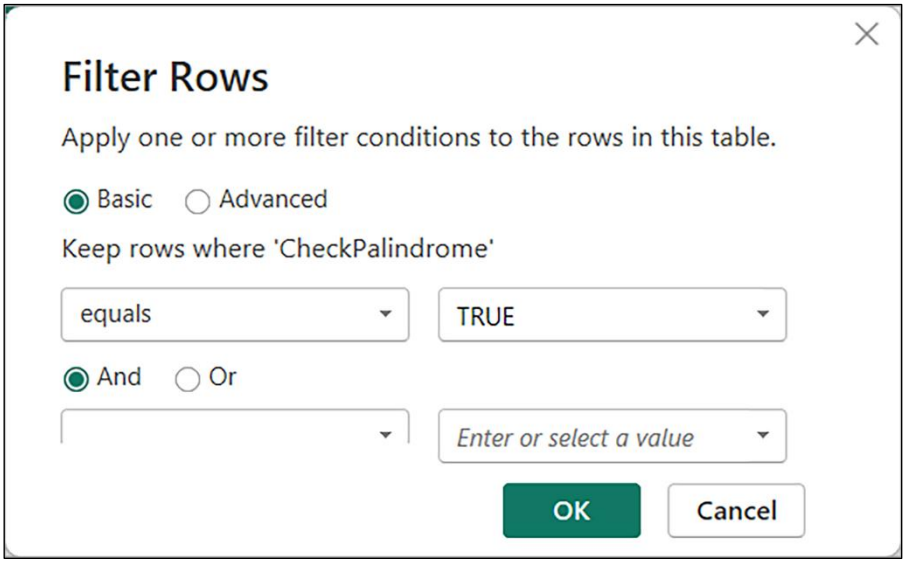


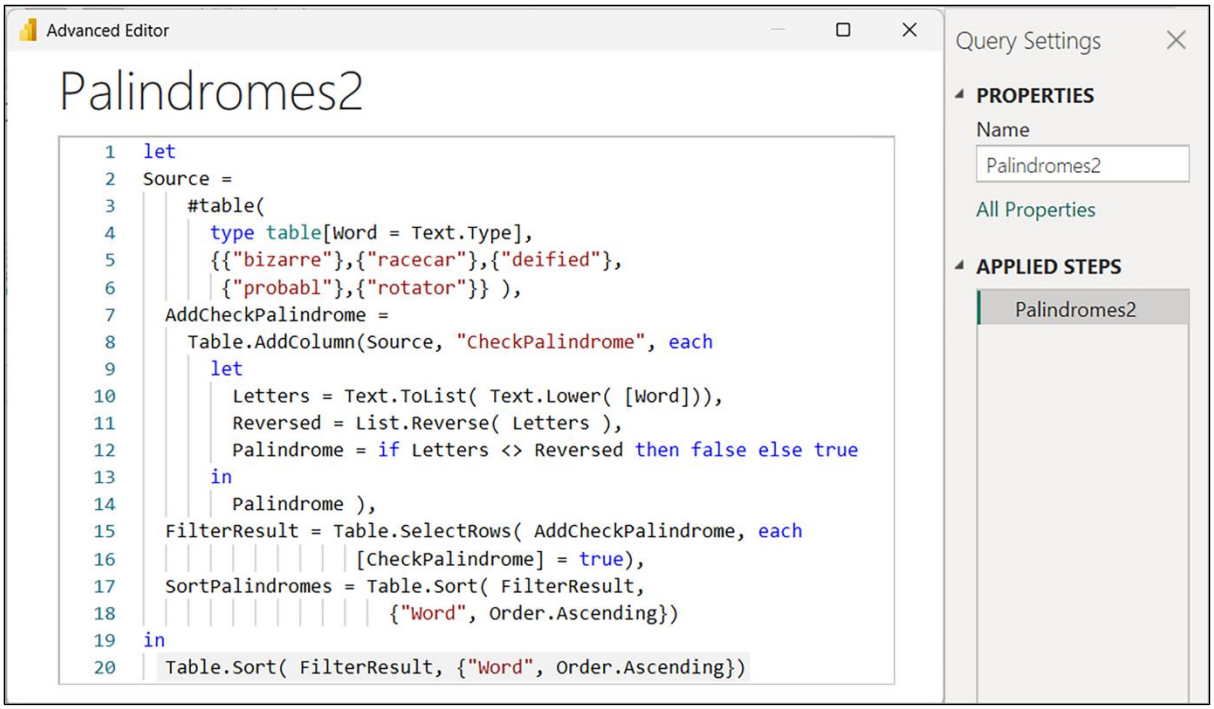
	Operators																					
	Comparison						Arithmetic						Logical			Coal.	Conc	List	Rec.	Func.	Meta	Type	
Value Type	=	<>	>	>=	<	<=	+	-	*	/	+x	-x	and	or	not	??	&	{}	[]	=>	Meta	is	as
Primitive Values																							
Null	●	●	●	●	●	●										●					●	○	○
Logical	●	●	●	●	●	●							●	●	●						●	○	○
Number	●	●	●	●	●	●	●	●	◐	◐	●	●									●	○	○
Time	●	●	●	●	●	●	○	◐									○				●	○	○
Date	●	●	●	●	●	●	○	◐									○				●	○	○
DateTime	●	●	●	●	●	●	○	◐													●	○	○
DateTimeZone	●	●	●	●	●	●	○	◐													●	○	○
Duration	●	●	●	●	●	●	◐		◐	◐	●	●									●	○	○
Text	●	●	●	●	●	●		◐									●				●	○	○
Binary	●	●	●	●	●	●															●	○	○
Structured Values																							
List	●	●															●	●			●	○	○
Record	●	●															●		●		●	○	○
Table	●	●															●		●		●	○	○
Function Values																							
Function	●	●																		●		○	○
Type Values																							
Type	●	●																			●	◐	◐

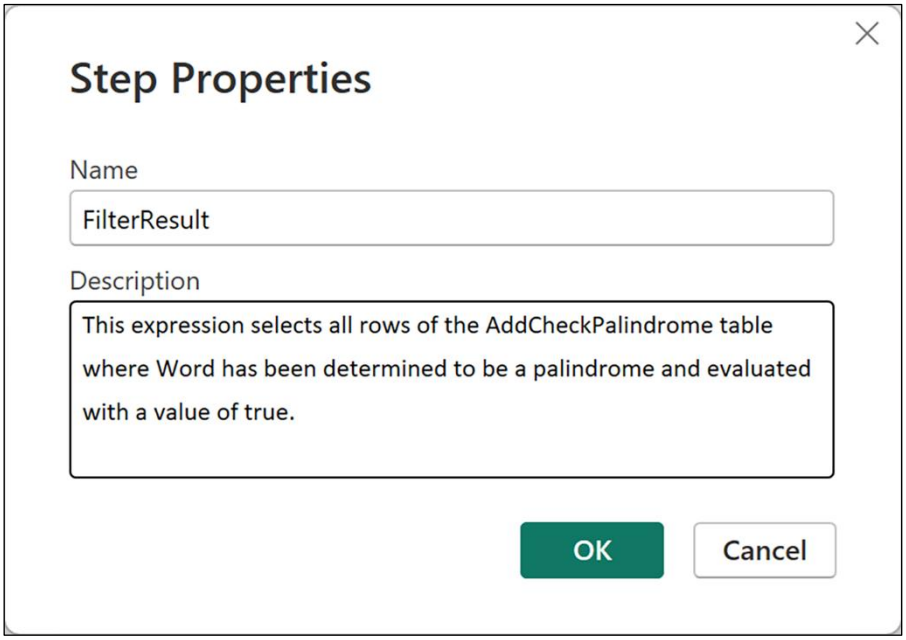
● **Self-Only** Operates exclusively with its own value type.
◐ **Self+Other** Operates with its own and some other value types.
○ **Other-Only** Operates exclusively with other value types.

Operand LHS => RHS	**Date**	**DateTime**	**DateTimeZone**	**Duration**	**Time**	**Number**
Date	All comparison - = duration			+, - = Date	& = DateTime	
DateTime		All comparison - = duration		+, - = DateTime		
DateTimeZone			All comparison - = duration	+, - = DateTimeZone		
Duration	+, - = Date	+, - = DateTime	+, - = DateTimeZone	All comparison +, - = duration	- = Time	*, / = Duration
Time	& = DateTime			+, - = Time	All comparison - = duration	
Number				* = duration		All comparison +, -, *, / = Number


Advanced Editor
Palindromes
let
Source =
#table(
type table[Word = Text.Type],
{{"bizarre"},{"racecar"},{"deified"},
{"probabl"},{"rotator"}}),
AddCheckPalindrome =
Table.AddColumn(Source, "CheckPalindrome", each
let
Letters = Text.ToList(Text.Lower([Word])),
Reversed = List.Reverse(Letters),
Palindrome = if Letters <> Reversed then false else true
in
Palindrome),
FilterResult = Table.SelectRows(AddCheckPalindrome, each
[CheckPalindrome] = true),
SortPalindromes = Table.Sort(FilterResult,
{"Word", Order.Ascending})
in
SortPalindromes
Query Settings
PROPERTIES
Name
Palindromes
All Properties
APPLIED STEPS
Source
AddCheckPalindrome
FilterResult
SortPalindromes


Filter Rows
Apply one or more filter conditions to the rows in this table.
Basic
Advanced
Keep rows where 'CheckPalindrome'
equals
TRUE
And
Or
Enter or select a value
OK
Cancel


Advanced Editor
Palindromes2
let
Source =
#table(
type table[Word = Text.Type],
{{"bizarre"},{"racecar"},{"deified"},
{"probabl"},{"rotator"}}),
AddCheckPalindrome =
Table.AddColumn(Source, "CheckPalindrome", each
let
Letters = Text.ToList(Text.Lower([Word])),
Reversed = List.Reverse(Letters),
Palindrome = if Letters <> Reversed then false else true
in
Palindrome),
FilterResult = Table.SelectRows(AddCheckPalindrome, each
[CheckPalindrome] = true),
SortPalindromes = Table.Sort(FilterResult,
{"Word", Order.Ascending})
in
Table.Sort(FilterResult, {"Word", Order.Ascending})
Query Settings
PROPERTIES
Name
Palindromes2
All Properties
APPLIED STEPS
Palindromes2


Step Properties
Name
FilterResult
Description
This expression selects all rows of the AddCheckPalindrome table where Word has been determined to be a palindrome and evaluated with a value of true.
OK
Cancel

Advanced Editor

Fizzbuzz

```
let
  Source = Table.FromList({1..100}, Splitter.SplitByNothing(), {"Numbers"},
    ExtraValues.Error),
  AddFizzBuzz = Table.AddColumn(Source, "FizzBuzz", each
    if Number.Mod([Numbers], 3) = 0 and Number.Mod([Numbers], 5) = 0 then "FizzBuzz" else
    if Number.Mod([Numbers], 3) = 0 then "Fizz" else
    if Number.Mod([Numbers], 5) = 0 then "Buzz" else
      [Numbers],
    Any.Type)
in
    AddFizzBuzz
```

	Numbers	FizzBuzz
1	1	1
2	2	2
3	3	Fizz
4	4	4
5	5	Buzz
6	6	Fizz
7	7	7
8	8	8
9	9	Fizz
10	10	Buzz
11	11	11
12	12	Fizz
13	13	13
14	14	14
15	15	FizzBuzz

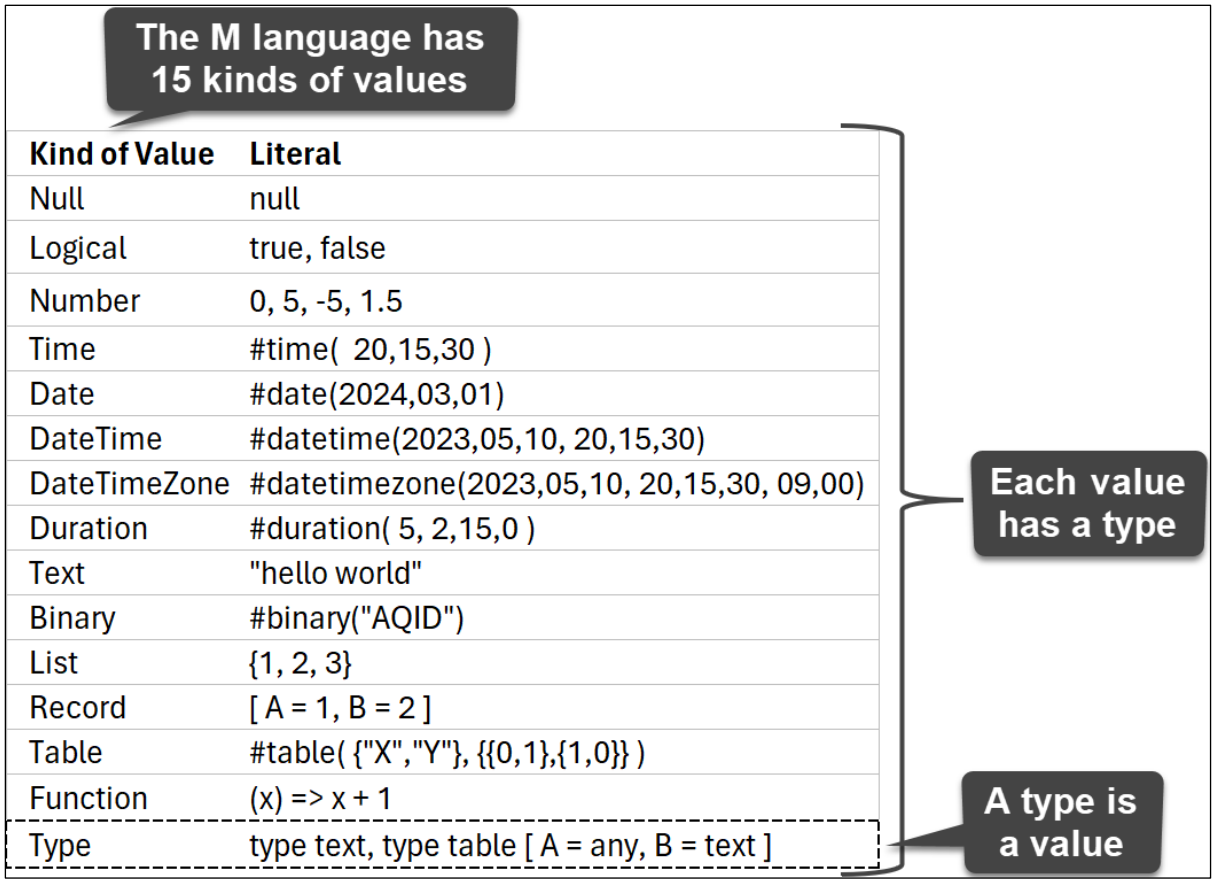
```
= Table.Sort( FilterResult, {"Word", 0})
```

	Word	CheckPalindrome
1	deified	TRUE
2	racecar	TRUE
3	rotator	TRUE

Name	Value	Description
Order.Ascending	0	Sorts the values in ascending order.
Order.Descending	1	Sorts the values in descending order.

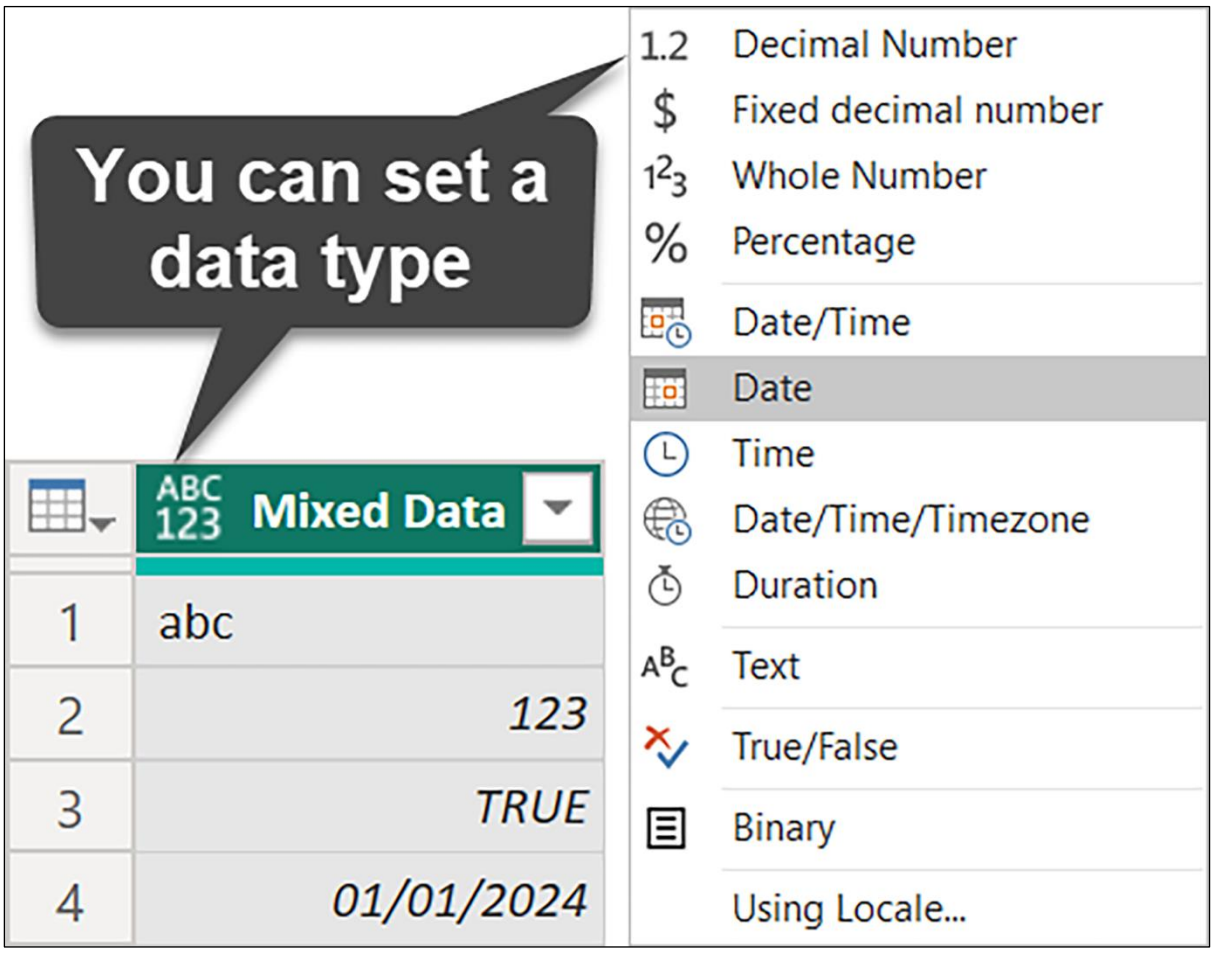
Enumeration	Description
BinaryOccurrence.Type	Specifies how many times the item is expected to appear in the group.
Occurrence.Type	Specifies the occurrence of an element in a sequence.
Order.Type	Specifies the direction of sorting.
PercentileMode.Type	Specifies the percentile mode type.
Precision.Type	Specifies the precision of comparison.
RankKind.Type	Specifies the type of ranking.
RoundingMode.Type	Specifies rounding direction when there is a tie between the possible numbers to round to.
AccessControlKind.Type	Specifies the kind of access control. This enumeration is not currently used in any function.
ODataOmitValues.Type	Specifies the kinds of values an OData service can omit.
SapBusinessWarehouseExe...	Valid options for SAP Business Warehouse execution mode option.
SapHanaDistribution.Type	Valid options for SAP HANA distribution option.
SapHanaRangeOperator.Type	A range operator for SAP HANA range input parameters.
BinaryEncoding.Type	Specifies the type of binary encoding.
Compression.Type	Specifies the type of compression.
ExtraValues.Type	Specifies the expected action for extra values in a row that contains columns more than expected.
MissingField.Type	Specifies the expected action for missing values in a row that contains columns less than expected.
BufferMode.Type	Describes the type of buffering to be performed.
ByteOrder.Type	Specifies the byte order.
CsvStyle.Type	Specifies the significance of quotes in Csv documents.
LimitClauseKind.Type	Describes the type of limit clause supported by the SQL dialect used by this data source.
QuoteStyle.Type	Specifies the quote style.
RelativePosition.Type	Indicates whether indexing should be done from the start or end of the input.
TextEncoding.Type	Specifies the text encoding type.
GroupKind.Type	Specifies the kind of grouping.
JoinAlgorithm.Type	Specifies the join algorithm to be used in the join operation.
JoinKind.Type	Specifies the kind of join operation.
JoinSide.Type	Specifies the left or right table of a join.
Day.Type	Specifies a day of week.
TraceLevel.Type	Specifies the trace level.
WebMethod.Type	Specifies an HTTP method.

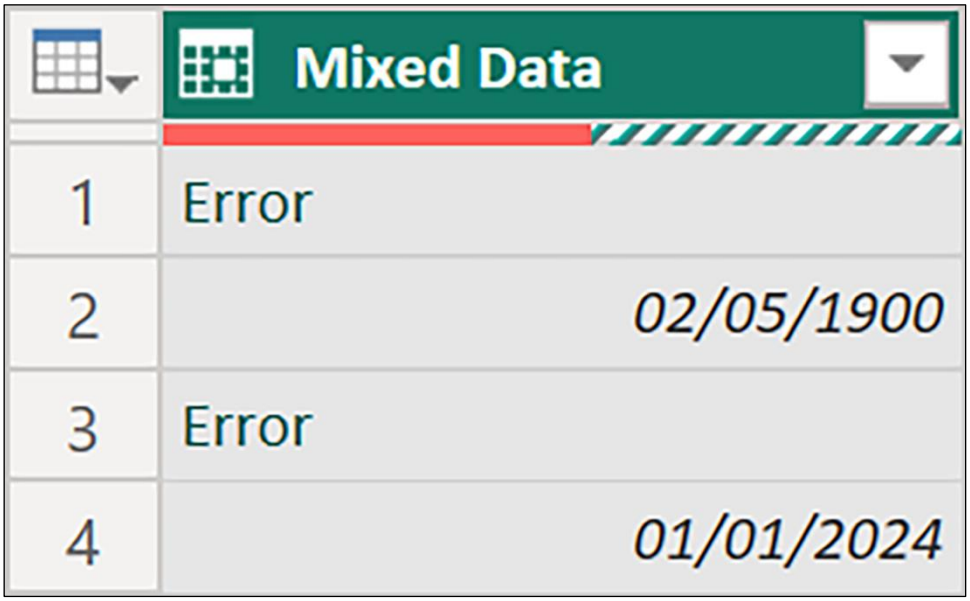
Chapter 5: Understanding Data Types



Kind of Value	Literal
Null	null
Logical	true, false
Number	0, 5, -5, 1.5
Time	#time(20,15,30)
Date	#date(2024,03,01)
DateTime	#datetime(2023,05,10, 20,15,30)
DateTimeZone	#datetimezone(2023,05,10, 20,15,30, 09,00)
Duration	#duration(5, 2,15,0)
Text	"hello world"
Binary	#binary("AQID")
List	{1, 2, 3}
Record	[A = 1, B = 2]
Table	#table({"X","Y"}, {{0,1},{1,0}})
Function	(x) => x + 1
Type	type text, type table [A = any, B = text]

	Mixed Data
1	abc
2	123
3	TRUE
4	01/01/2024





	Mixed Data
1	Error
2	02/05/1900
3	Error
4	01/01/2024

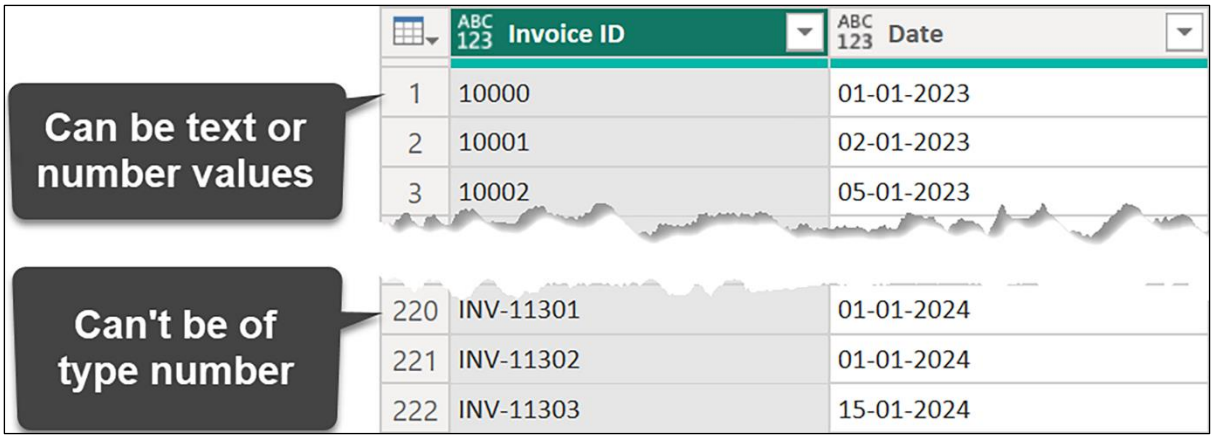
```
= Table.AddColumn(#"Changed Type", "Discount", each 0.05)
```

	Invoice ID	Date	Sales Amount	Discount
1	INV-11302	01/01/2024	195	0.05
2	INV-11303	15/01/2024	925	0.05
3	INV-11304	31/01/2024	250	0.05
4	INV-11305	01/02/2024	500	0.05

```
= Table.AddColumn(#"Added Custom", "Type", each Value.Type( [Discount] ) )
```

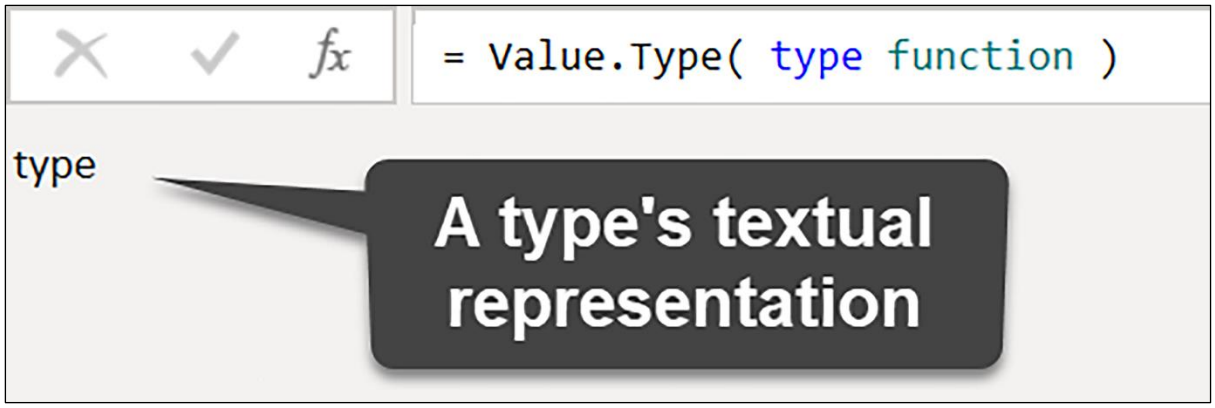
	Invoice ID	Date	Sales Amount	Discount	Type
1	INV-11302	01/01/2024	195	0.05	Type
2	INV-11303	15/01/2024	925	0.05	Type
3	INV-11304	31/01/2024	250	0.05	Type
4	INV-11305	01/02/2024	500	0.05	Type

number


Can be text or number values
Can't be of type number
Invoice ID
Date
1
10000
01-01-2023
2
10001
02-01-2023
3
10002
05-01-2023
220
INV-11301
01-01-2024
221
INV-11302
01-01-2024
222
INV-11303
15-01-2024

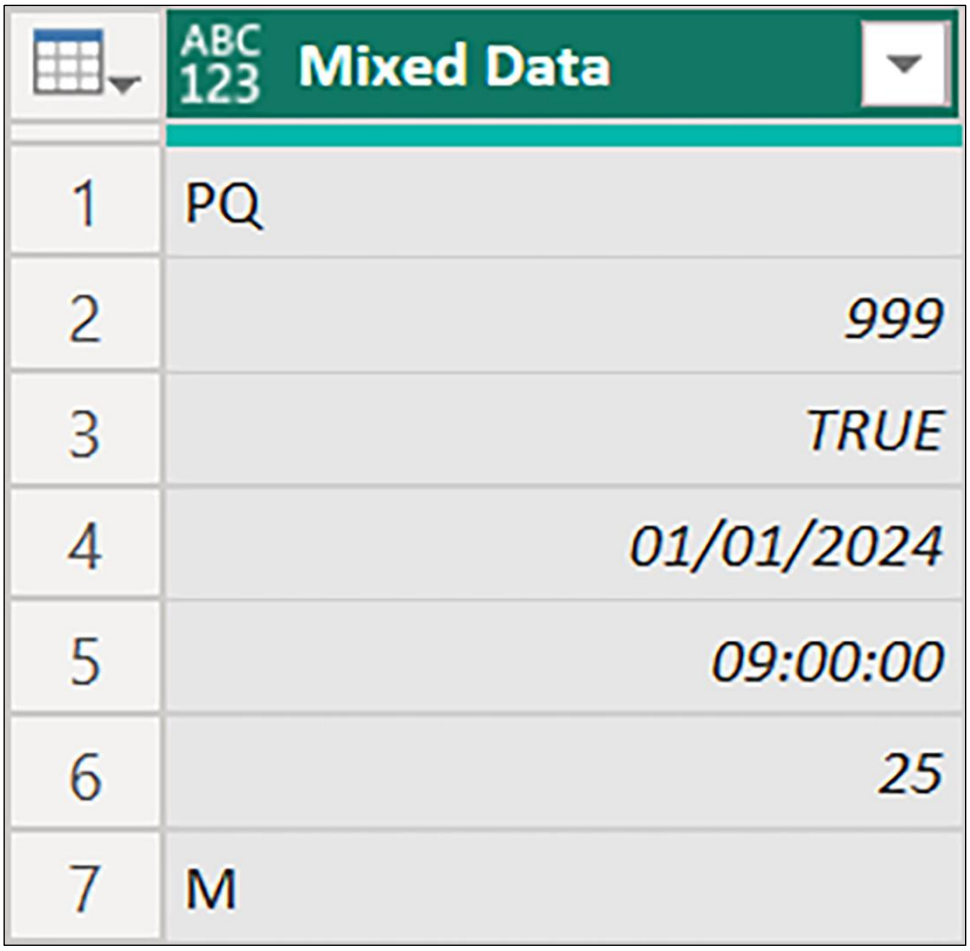
Primitive Types	Type Qualifiers	Value
binary		Binary
date		Date
datetime		DateTime
datetimezone		DateTimeZone
duration		Duration
list	Custom	List
logical		Logical
null		Null
number		Number
record	Abstract, Custom	Record
text		Text
time		Time
type		Type
function	Abstract, Custom	Function
table	Abstract, Custom	Table
any	Abstract	
anynonnull	Abstract	
none	Abstract	

Abstract: type is considered an abstract type
Custom: type can also be used as custom type

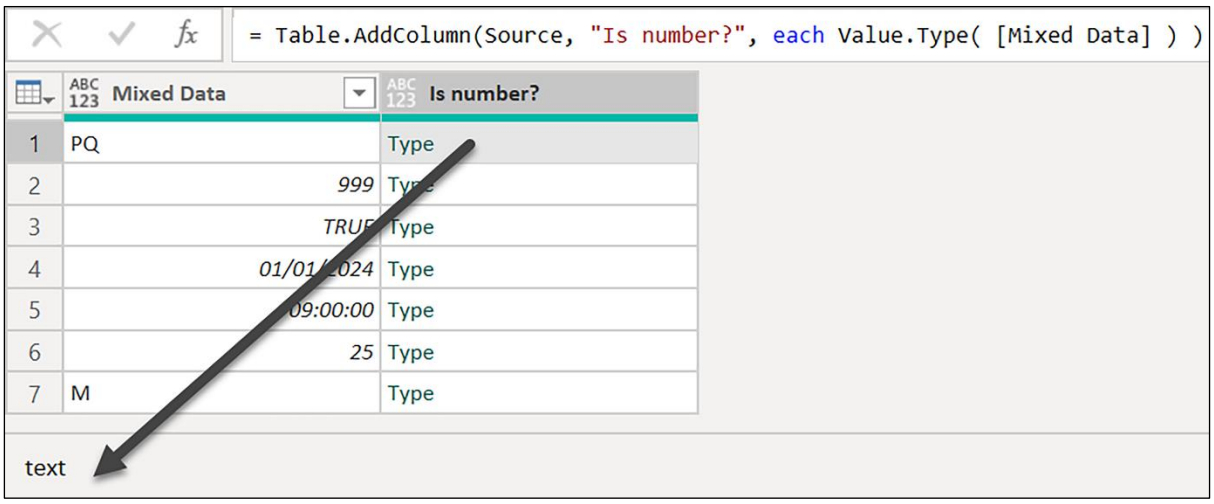


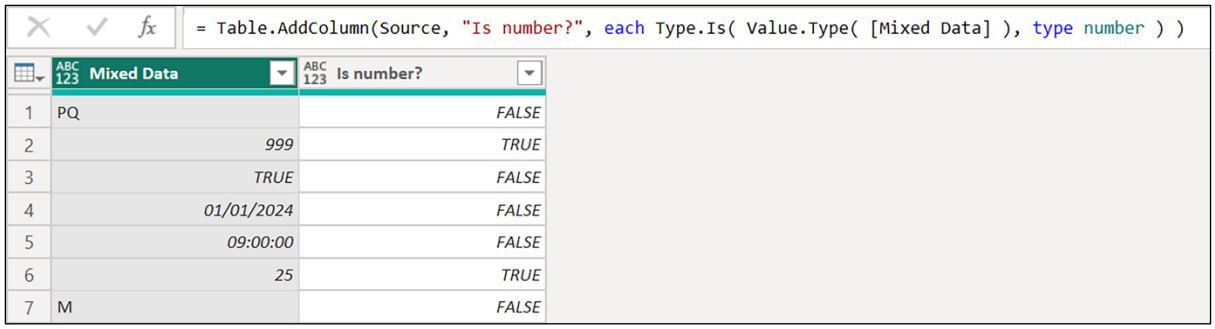
	Amount	Number.IsOdd	Number.Sign
1	*-10*	*FALSE*	*-1*
2	*-5*	*TRUE*	*-1*
3	*null*	Error	*null*
4	*0*	*FALSE*	*0*
5	*null*	Error	*null*
6	*3*	*TRUE*	*1*
7	*8*	*FALSE*	*1*

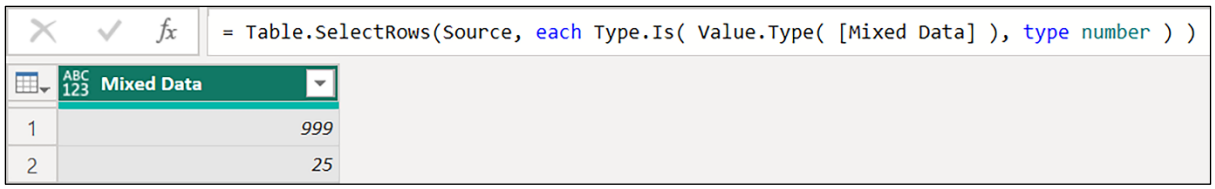
	Date	Product	Sales
1	2024-01-05	Bread	2.50
2	2024-02-10	Milk	1.99
3	2024-03-15	Cereal	3.75
4	2024-04-20	Pasta	1.29

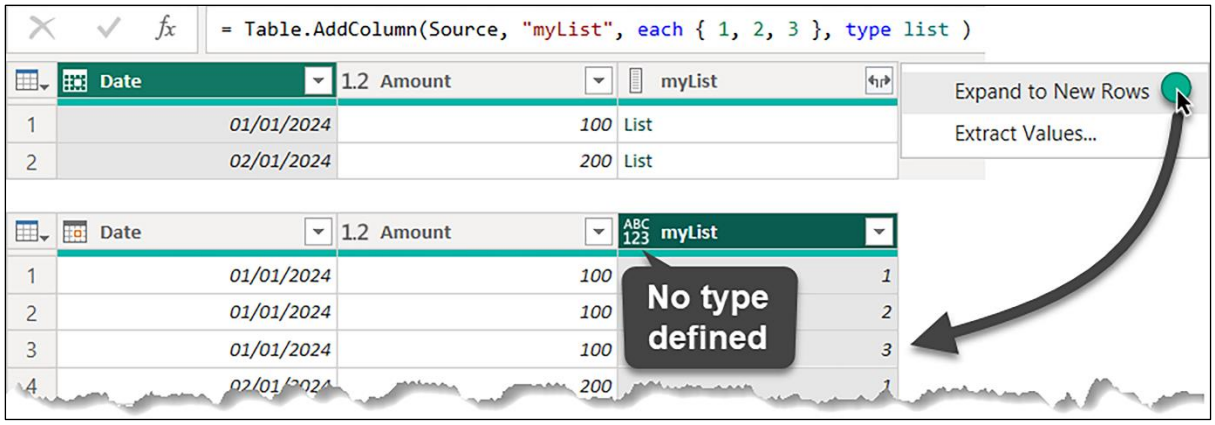


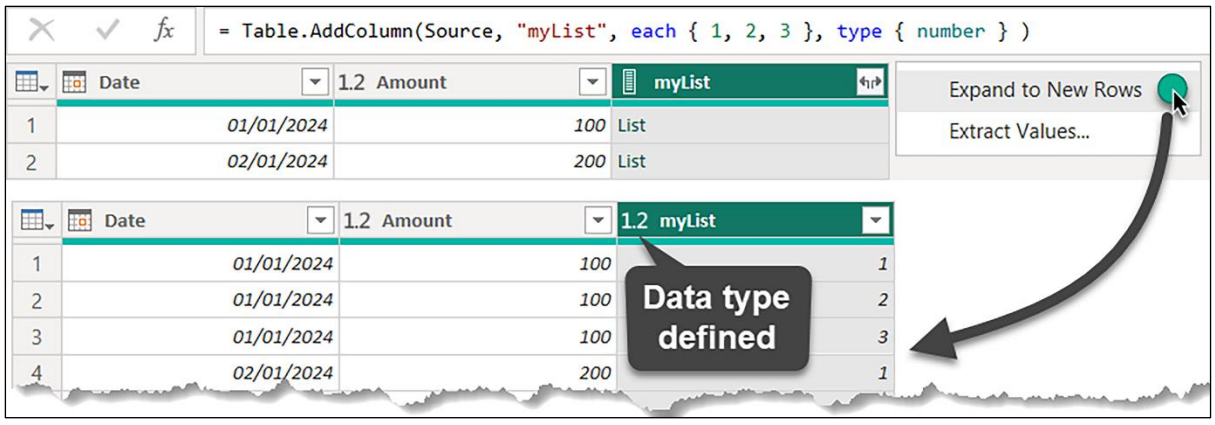
	Mixed Data
1	PQ
2	999
3	TRUE
4	01/01/2024
5	09:00:00
6	25
7	M

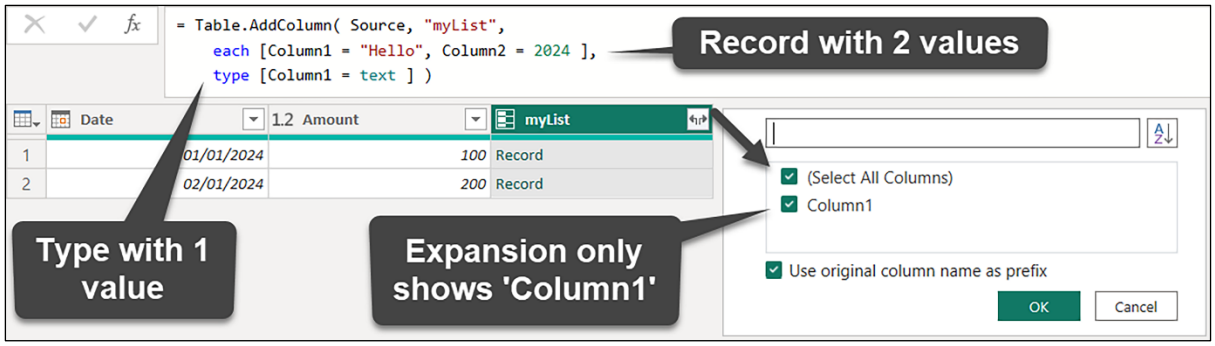



= Table.AddColumn(Source, "Is number?", each Type.Is(Value.Type([Mixed Data]), type number))
Mixed Data
Is number?
1 PQ FALSE
2 999 TRUE
3 TRUE FALSE
4 01/01/2024 FALSE
5 09:00:00 FALSE
6 25 TRUE
7 M FALSE

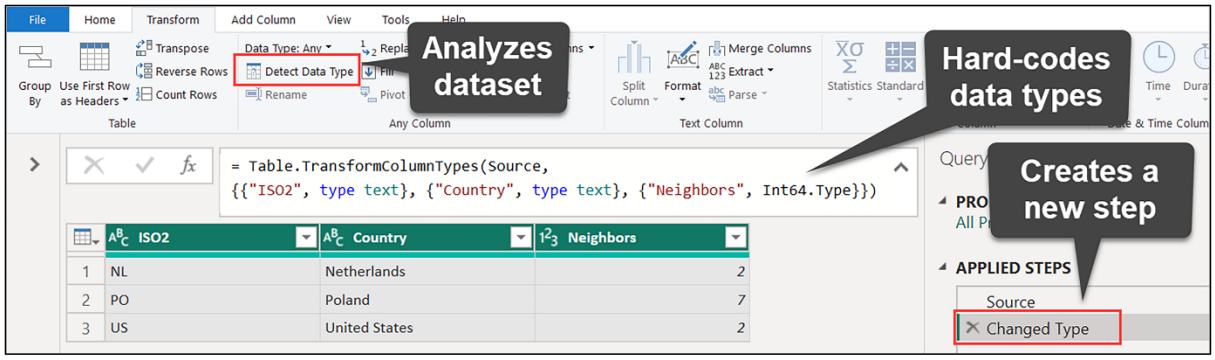

= Table.SelectRows(Source, each Type.Is(Value.Type([Mixed Data]), type number))
Mixed Data
1 999
2 25

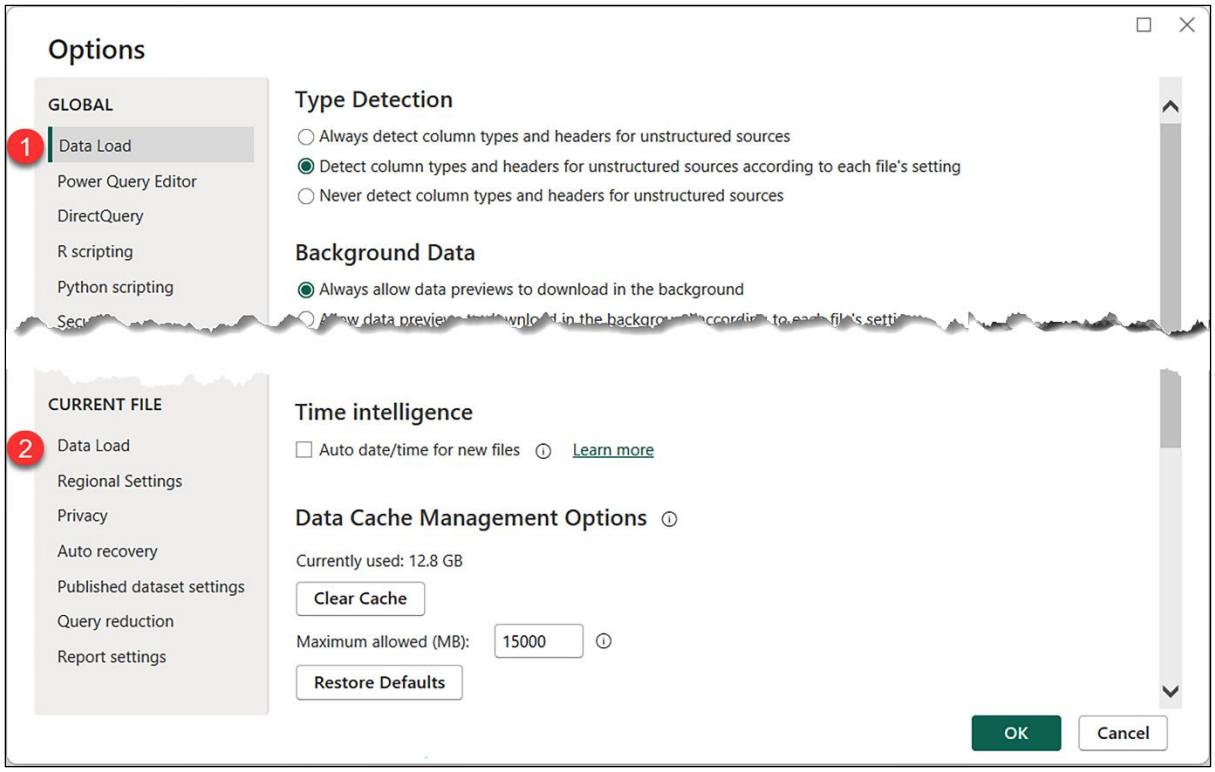

= Table.AddColumn(Source, "myList", each { 1, 2, 3 }, type list)
Date
1.2 Amount
myList
1 01/01/2024 100 List
2 02/01/2024 200 List
Expand to New Rows
Extract Values...
Date
1.2 Amount
myList
1 01/01/2024 100 1
2 01/01/2024 100 2
3 01/01/2024 100 3
4 02/01/2024 200 1
No type defined


= Table.AddColumn(Source, "myList", each { 1, 2, 3 }, type { number })
Date
1.2 Amount
myList
1 01/01/2024 100 List
2 02/01/2024 200 List
Expand to New Rows
Extract Values...
Date
1.2 Amount
1.2 myList
1 01/01/2024 100 1
2 01/01/2024 100 2
3 01/01/2024 100 3
4 02/01/2024 200 1
Data type defined



	ISO2	Country	Neighbors
1	NL	Netherlands	2
2	PO	Poland	7
3	US	United States	2





	Date	Product Name	Price
1	01/01/2024	Chef's Knife	45.00
2	02/01/2024	Cast Iron Skillet	30.50
3	03/01/2024	Cutting Board	15.75
4	04/01/2024	Digital Kitchen Scale	22.00
5	05/01/2024	Dutch Oven	55.99
6	06/01/2024	Silicone Spatula Set	10.25
7	07/01/2024	Stainless Steel Cookware Set	120.00
8	08/01/2024	Programmable Slow Cooker	75.00

```
= Table.AddColumn(#"Changed Type", "Custom",
each "On " & Text.From( [Date] ) & " we sold " & Text.From( [Price] ) & " worth of "& [Product Name] & "s"  )
```

	Date	Product Name	Price	Custom
1	01/01/2024	Chef's Knife	45.00	On 01/01/2024 we sold 45 worth of Chef's Knifes
2	02/01/2024	Cast Iron Skillet	30.50	On 02/01/2024 we sold 30.5 worth of Cast Iron Skillets
3	03/01/2024	Cutting Board	15.75	On 03/01/2024 we sold 15.75 worth of Cutting Boards
4	04/01/2024	Digital Kitchen Scale	22.00	On 04/01/2024 we sold 22 worth of Digital Kitchen Scales
5	05/01/2024	Dutch Oven	55.99	On 05/01/2024 we sold 55.99 worth of Dutch Ovens
6	06/01/2024	Silicone Spatula Set	10.25	On 06/01/2024 we sold 10.25 worth of Silicone Spatula Sets
7	07/01/2024	Stainless Steel Cookware Set	120.00	On 07/01/2024 we sold 120 worth of Stainless Steel Cookware Sets
8	08/01/2024	Programmable Slow Cooker	75.00	On 08/01/2024 we sold 75 worth of Programmable Slow Cookers

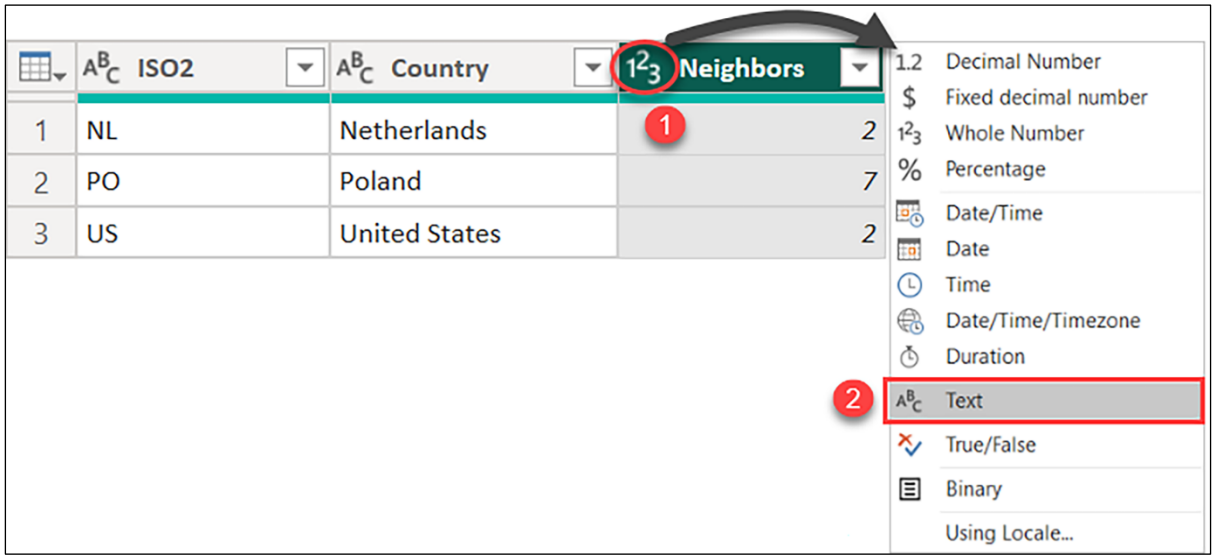
Data Types	1.2	$	123	%						ABC	
1.2 Decimal number		⊖	⊖	✓	✓	⊖	⚠	⊕	✓	✓	✓
$ Currency	✓		⊖	✓	✓	⊖	⚠	⊕	✓	✓	✓
123 Whole number	✓	✓		✓	✓	✓	⚠	⊕	✓	✓	✓
% Percentage	✓	⊖	⊖		✓	✓	✓	⊕	✓	✓	✓
Date/Time	✓	⊖	⊖	✓		⊖	⊖	⊕	⚠	✓	⚠
Date	✓	✓	✓	✓	✓		⚠	⊕	⚠	✓	⚠
Time	✓	✓	✓	✓	⊕	⚠		⊕	⚠	✓	⚠
Date/Time/Zone	✓	⊖	⊖	✓	⊖	⊖	⊖		⚠	✓	⚠
Duration	✓	⊖	⊖	✓	⚠	⚠	⚠	⚠		✓	⚠
ABC Text	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
True/False	✓	✓	✓	✓	⚠	⚠	⚠	⚠	⚠	✓	

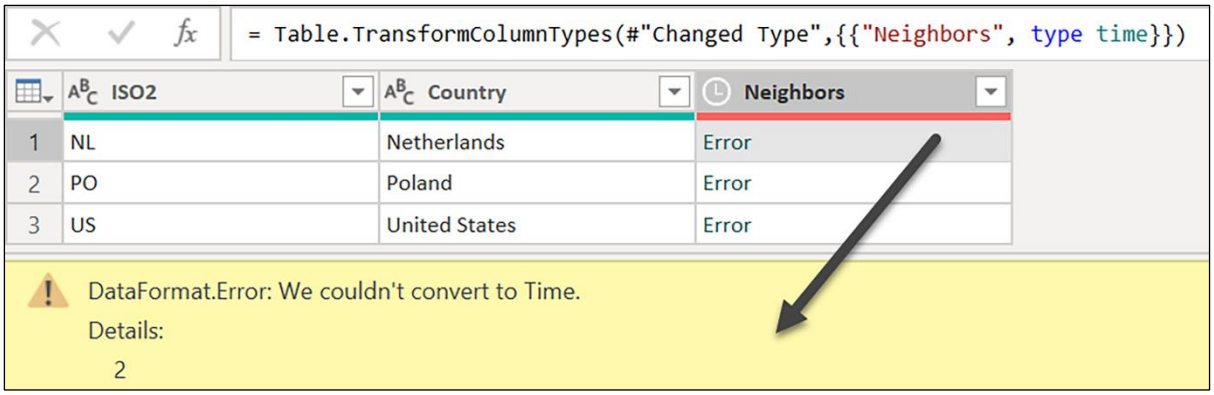
✓ Conversion possible

⚠ Conversion fails with error

⊕ Conversion possible, adds detail to the original value

⊖ Conversion possible, loses detail of the original value

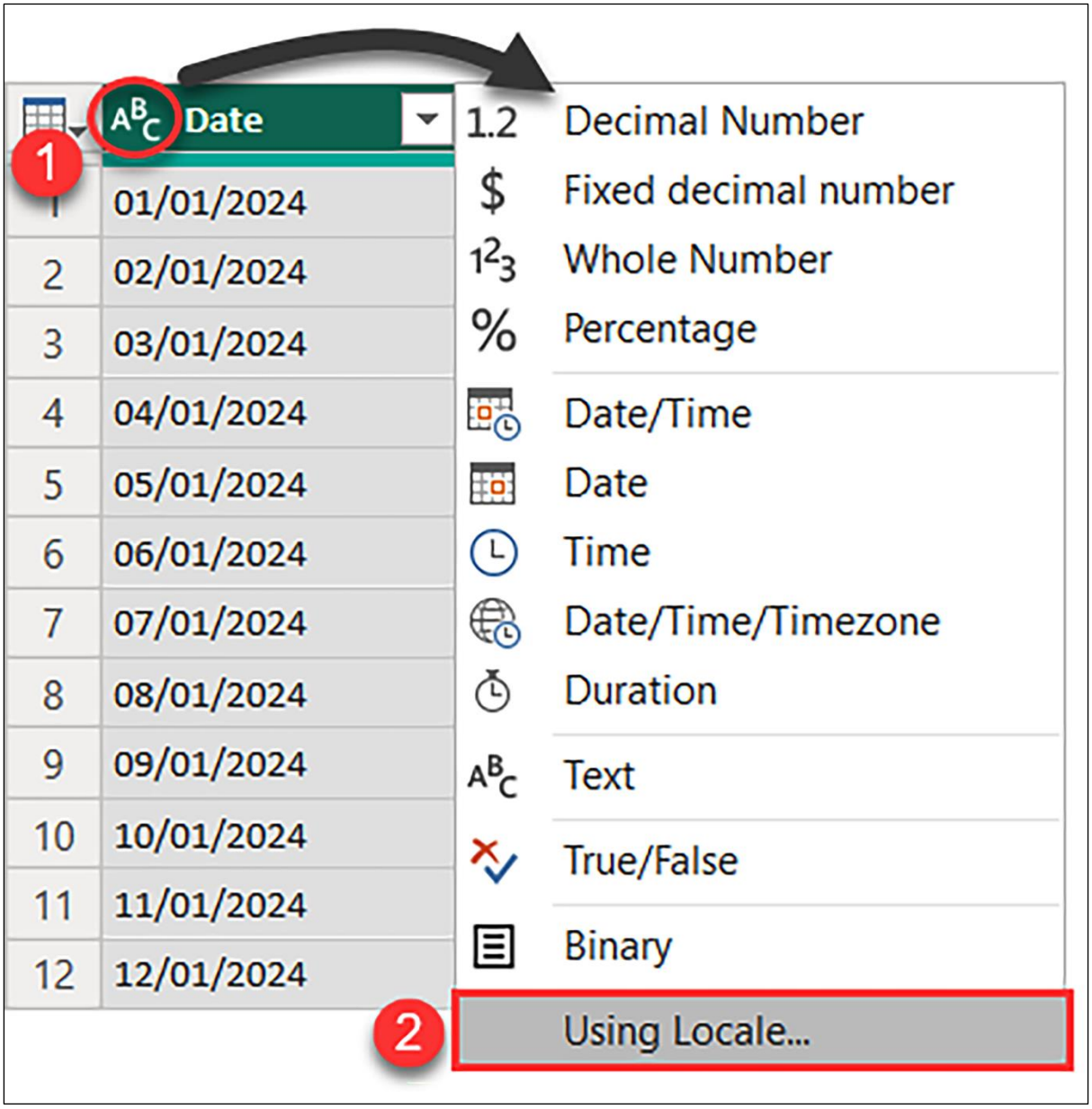


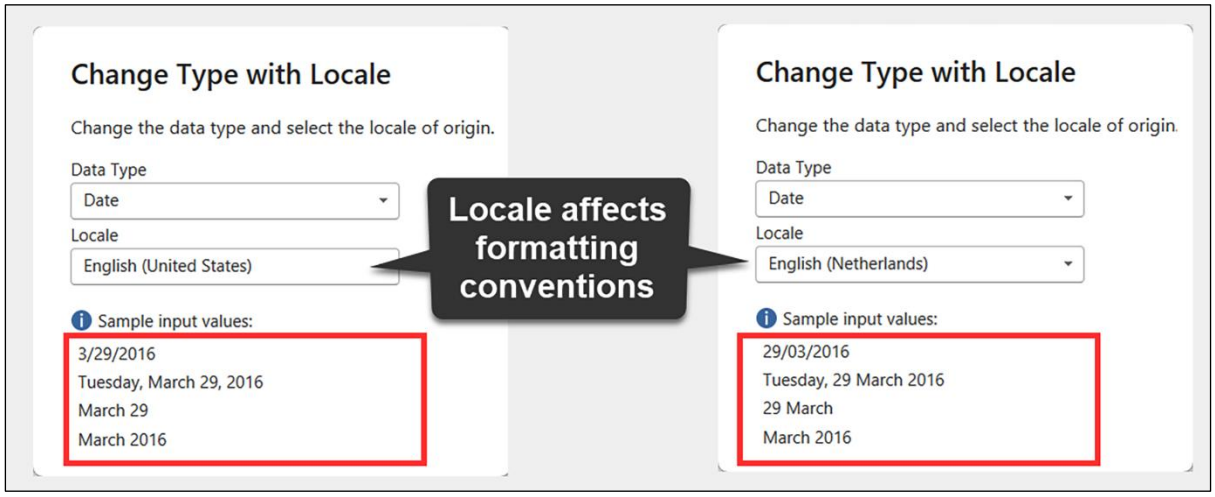


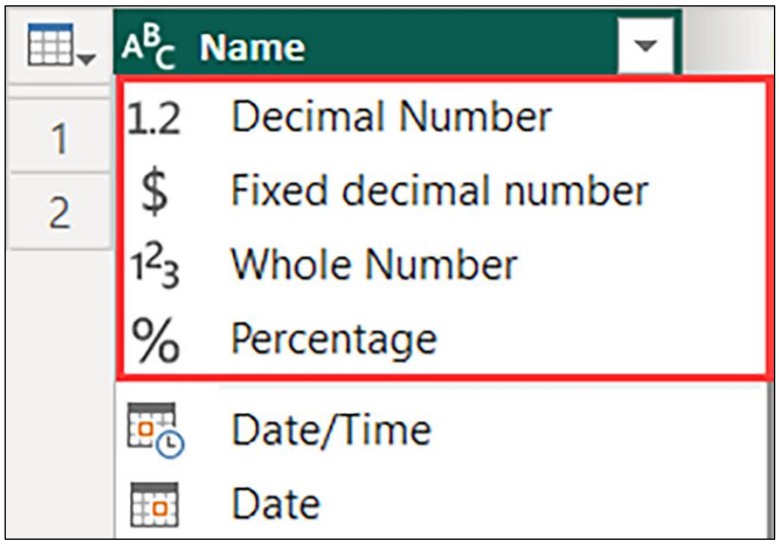
	Transaction ID	Meter Code	Transaction Date	Payment Mean	Amount Paid
1	1250162207	12028002	7/11/2023	PHONE	0.26
2	1250162278	19232002	7/11/2023	PHONE	1.52
3	1250131414	19127010	7/11/2023	PHONE	9.67
4	1250134465	5073002	7/11/2023	CREDIT CARD	4
5	1250134860	19161010	7/11/2023	PHONE	6.73
6	1250134876	5019002	7/11/2023	PHONE	0.3
7	1250134941	19126010	7/11/2023	PHONE	8.77

	Transaction ID	Meter Code	Transaction Date	Payment Mean	Amount Paid
1	1250162207	12028002	7/11/2023	PHONE	0
2	1250162278	19232002	7/11/2023	PHONE	2
3	1250131414	19127010	7/11/2023	PHONE	10
4	1250134465	5073002	7/11/2023	CREDIT CARD	4
5	1250134860	19161010	7/11/2023	PHONE	7
6	1250134876	5019002	7/11/2023	PHONE	0
7	1250134941	19126010	7/11/2023	PHONE	9

	Transaction ID	Meter Code	Transaction Date
1	1250162207	12028002	7/11/2023 12:00:00 AM
2	1250162278	19232002	7/11/2023 12:00:00 AM
3	1250131414	19127010	7/11/2023 12:00:00 AM
4	1250134465	5073002	7/11/2023 12:00:00 AM
5	1250134860	19161010	7/11/2023 12:00:00 AM
6	1250134876	5019002	7/11/2023 12:00:00 AM
7	1250134941	19126010	7/11/2023 12:00:00 AM




Change Type with Locale
Change the data type and select the locale of origin.
Data Type
Date
Locale
English (United States)
Sample input values:
3/29/2016
Tuesday, March 29, 2016
March 29
March 2016
Locale affects formatting conventions
Change Type with Locale
Change the data type and select the locale of origin.
Data Type
Date
Locale
English (Netherlands)
Sample input values:
29/03/2016
Tuesday, 29 March 2016
29 March
March 2016


Name
1
2
1.2 Decimal Number
$ Fixed decimal number
123 Whole Number
% Percentage
Date/Time
Date

```
= Type.Facets( addFacets )
```

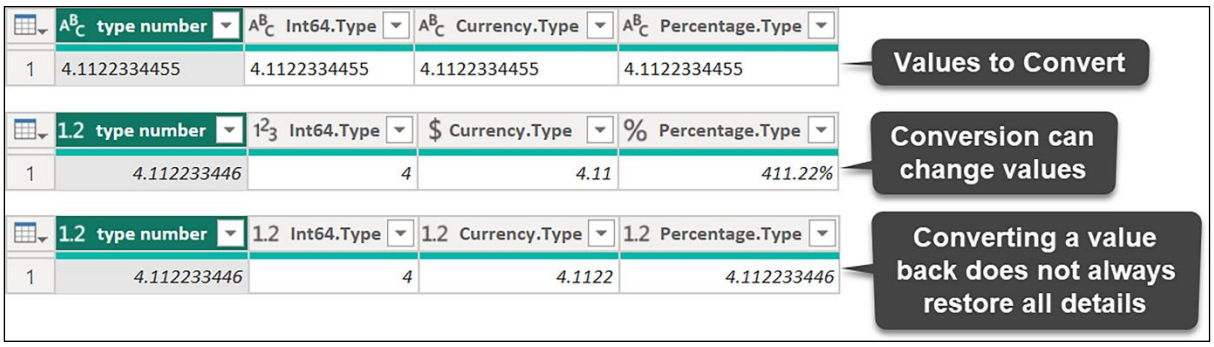
NumericPrecisionBase	10
NumericPrecision	null
NumericScale	null
DateTimePrecision	null
MaxLength	4
IsVariableLength	null
NativeTypeName	INTEGER
NativeDefaultExpression	null
NativeExpression	null

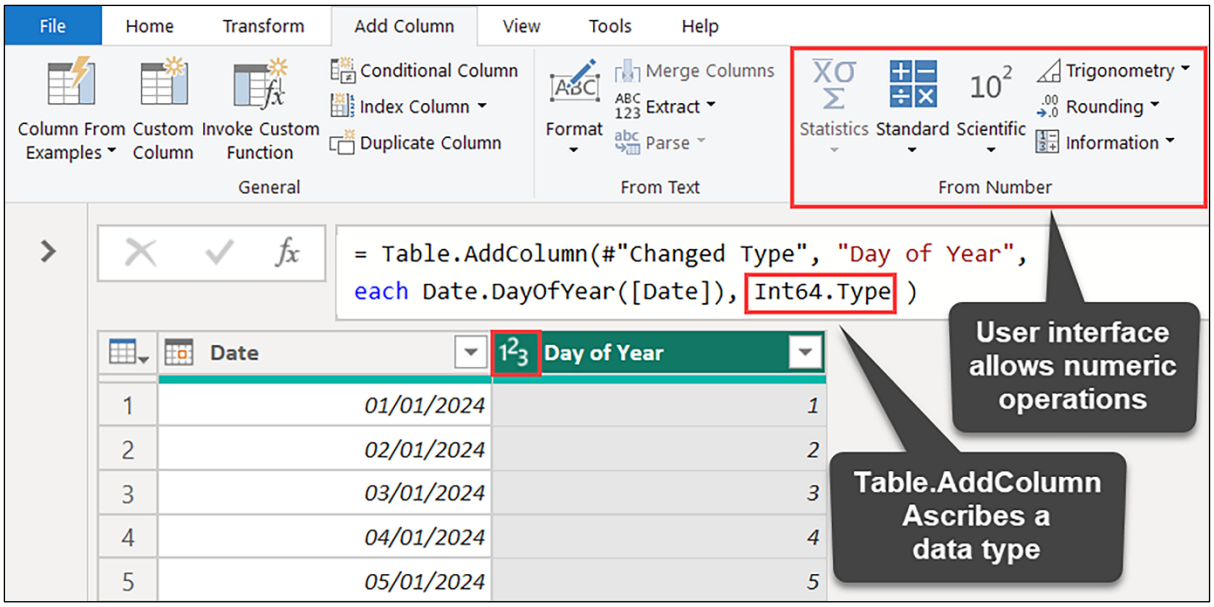
Type Claim Data Type

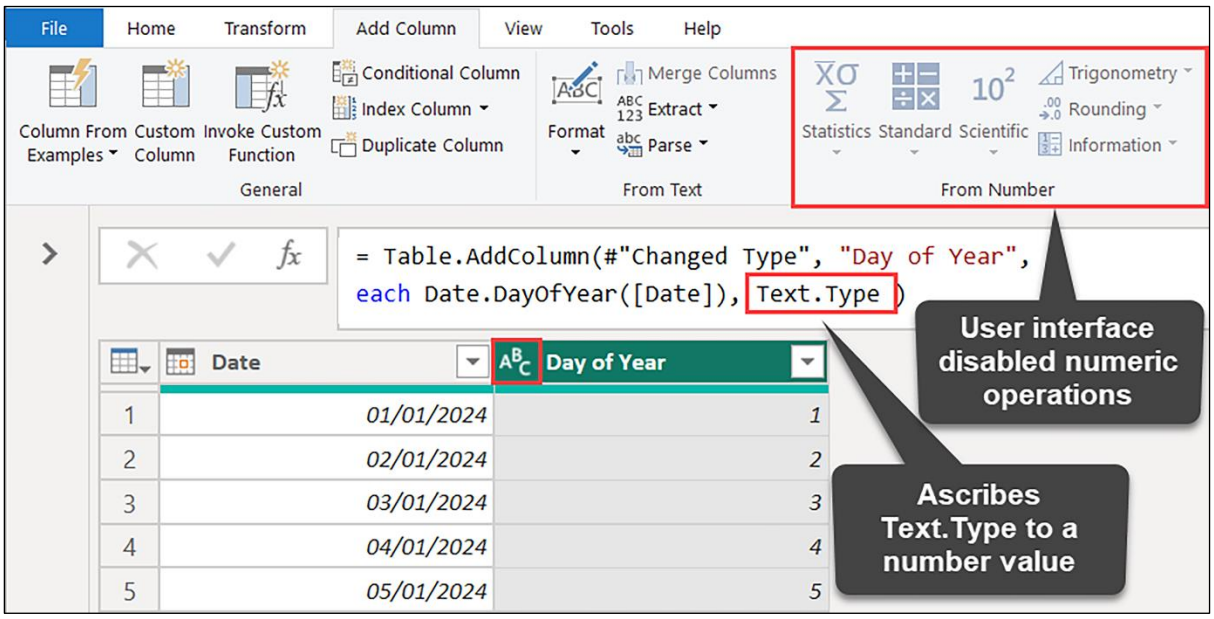
	Name	Position	TypeName	Kind	IsNullable	NumericPrecisionBase	NumericPrecision	NumericScale	DateTimePrecision
1	ProductKey	0	Int64.Type	number	FALSE	null	null	null	null
2	Product	1	Text.Type	text	FALSE	null	null	null	null

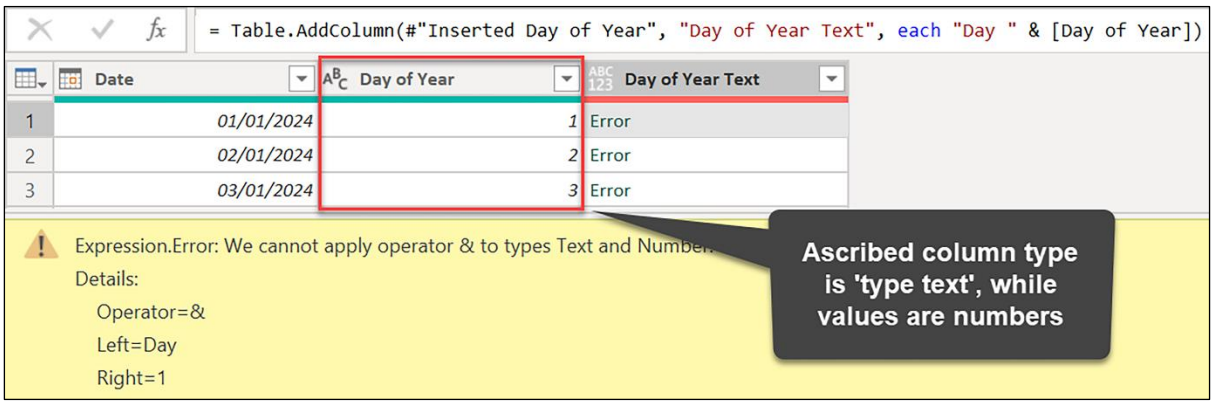
MaxLength	IsVariableLength	NativeTypeName	NativeDefaultExpression	NativeExpression	Description	IsWritable	FieldCaption
null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null

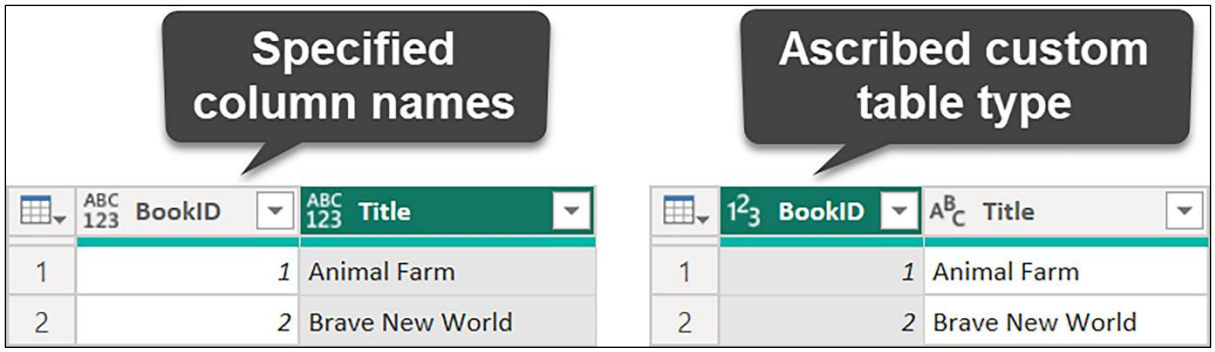
Type Claim	Base Type	Description
Any.Type	type any	Represents all values
Binary.Type	type binary	Represents all binary values
Date.Type	type date	Represents all date values
DateTime.Type	type datetime	Represents all date and time values
DateTimeZone.Type	type datetimezone	Represents all date and time values relative to a timezone
Duration.Type	type duration	Represents all duration values
Function.Type	type function	Represents all functions
List.Type	type list	Represents all lists
Logical.Type	type logical	Represents all logical values
None.Type	type none	Represents no values
Null.Type	type null	Represents null
Byte.Type	type number	Represents all bytes
Currency.Type	type number	Represents currency value
Decimal.Type	type number	Represents fixed-point decimal number
Double.Type	type number	Represents double precision floating point number
Int16.Type	type number	Represents signed 16 bit integer
Int32.Type	type number	Represents signed 32 bit integer
Int64.Type	type number	Represents signed 64 bit integer
Int8.Type	type number	Represents signed 8 bit integer
Number.Type	type number	Represents all numbers
Percentage.Type	type number	Represents percentage value
Single.Type	type number	Represents single precision floating point number
Record.Type	type record	Represents all records
Table.Type	type table	Represents all tables
Character.Type	type text	Represents all characters
Guid.Type	type text	Represents a GUID value
Password.Type	type text	Represents a text password
Text.Type	type text	Represents all text values
Uri.Type	type text	Represents a text URI
Time.Type	type time	Represents all time values
Type.Type	type type	Represents all types

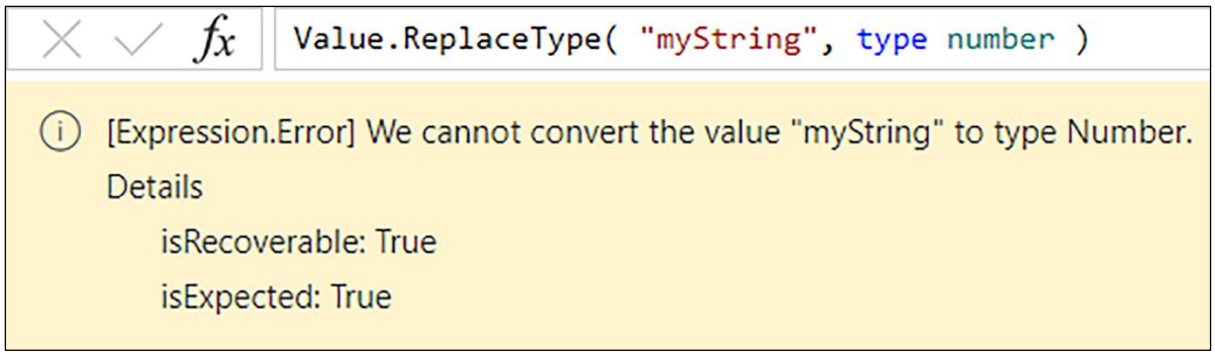



File
Home
Transform
Add Column
View
Tools
Help
Column From Examples
Custom Column
Invoke Custom Function
Conditional Column
Index Column
Duplicate Column
General
Format
Merge Columns
Extract
Parse
From Text
Statistics
Standard
Scientific
Trigonometry
Rounding
Information
From Number
= Table.AddColumn(#"Changed Type", "Day of Year",
each Date.DayOfYear([Date]), Int64.Type)
Date
Day of Year
1 01/01/2024 1
2 02/01/2024 2
3 03/01/2024 3
4 04/01/2024 4
5 05/01/2024 5
User interface allows numeric operations
Table.AddColumn Ascribes a data type

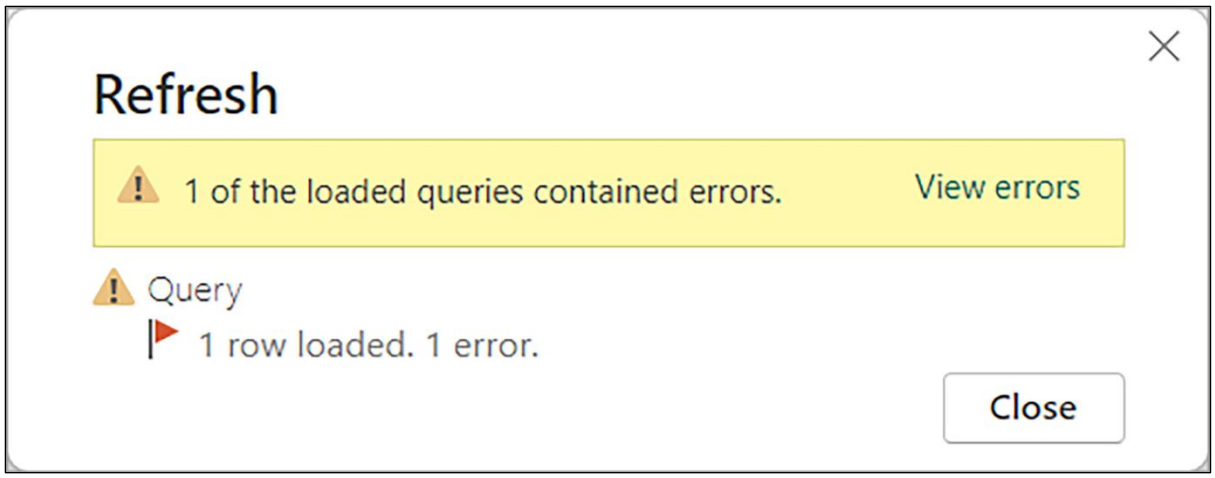

File
Home
Transform
Add Column
View
Tools
Help
Column From Examples
Custom Column
Invoke Custom Function
Conditional Column
Index Column
Duplicate Column
General
Format
Merge Columns
Extract
Parse
From Text
Statistics
Standard
Scientific
Trigonometry
Rounding
Information
From Number
= Table.AddColumn(#"Changed Type", "Day of Year",
each Date.DayOfYear([Date]), Text.Type)
Date
Day of Year
1 01/01/2024 1
2 02/01/2024 2
3 03/01/2024 3
4 04/01/2024 4
5 05/01/2024 5
User interface disabled numeric operations
Ascribes Text.Type to a number value


= Table.AddColumn(#"Inserted Day of Year", "Day of Year Text", each "Day " & [Day of Year])
Date
Day of Year
Day of Year Text
1 01/01/2024 1 Error
2 02/01/2024 2 Error
3 03/01/2024 3 Error
Expression.Error: We cannot apply operator & to types Text and Number.
Details:
Operator=&
Left=Day
Right=1
Ascribed column type is 'type text', while values are numbers


Specified column names
Ascribed custom table type
ABC 123 BookID
ABC 123 Title
1 2 3 BookID
A B C Title
1 Animal Farm
2 Brave New World


Value.ReplaceType("myString", type number)
[Expression.Error] We cannot convert the value "myString" to type Number.
Details
isRecoverable: True
isExpected: True

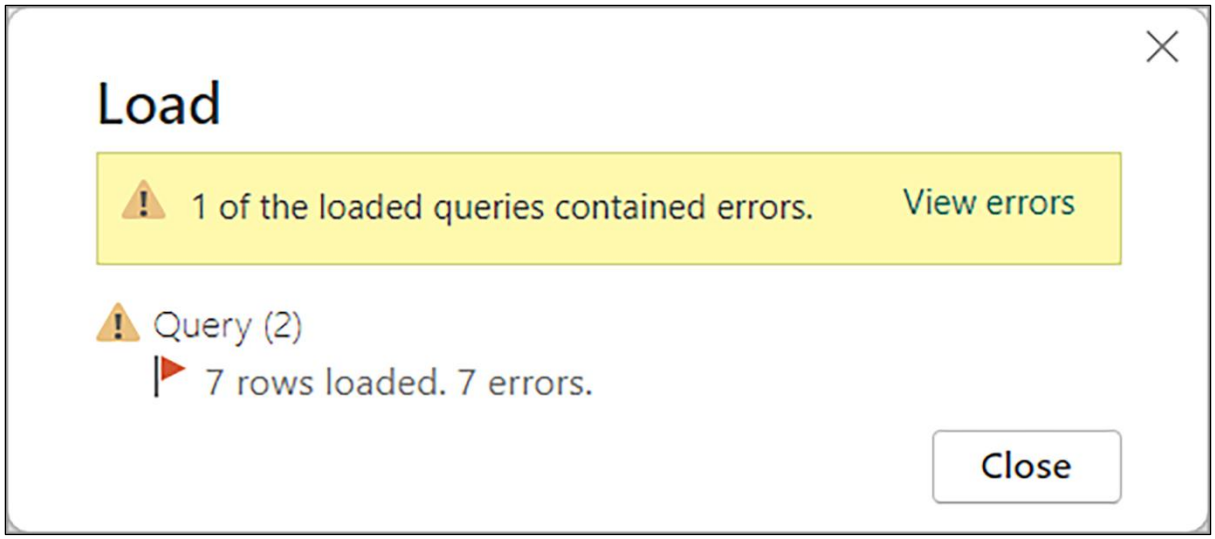

= Value.ReplaceType(5.33353 , Int64.Type)
5.33353
Has decimals
Ascribes a Whole Number


Refresh
1 of the loaded queries contained errors.
View errors
Query
1 row loaded. 1 error.
Close

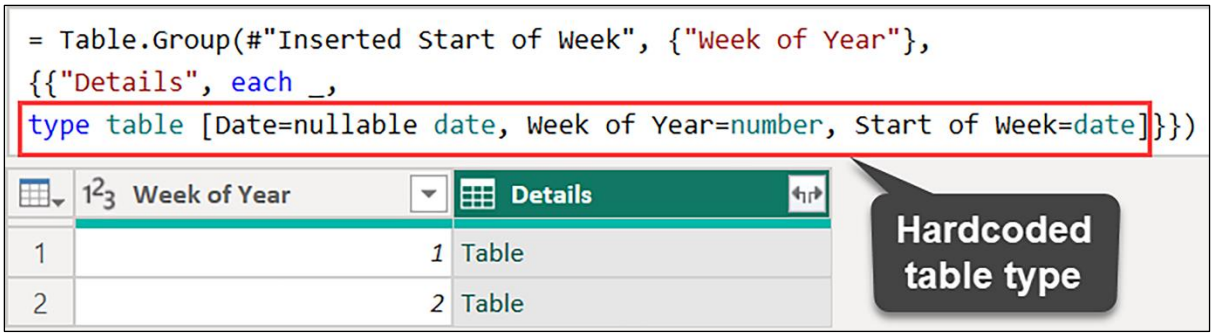
```
= Table.AddColumn(#"Inserted Day of Year", "Day Name",
each Date.DayOfWeekName([Date]), type number )
```

	Date	Day of Year	Day Name
1	01/01/2024	1	Monday
2	02/01/2024	2	Tuesday
3	03/01/2024	3	Wednesday
4	04/01/2024	4	Thursday
5	05/01/2024	5	Friday
6	06/01/2024	6	Saturday
7	07/01/2024	7	Sunday

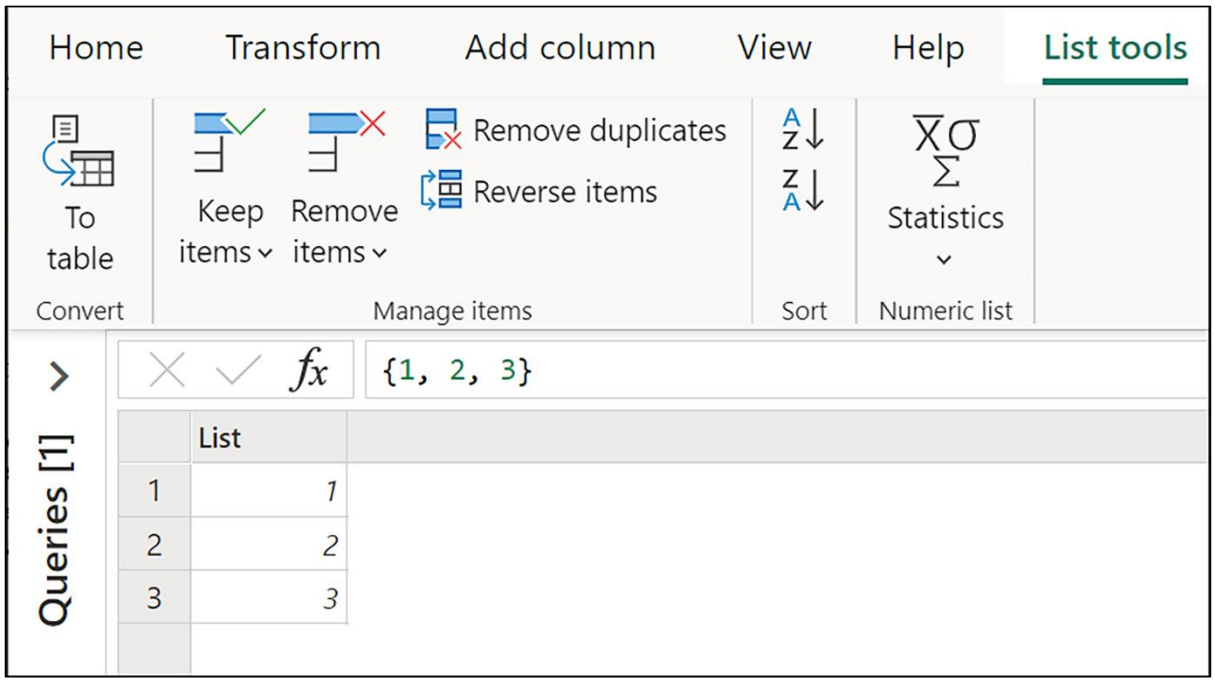
Ascribes an incompatible type to a table column



	Date	Week of Year	Start of Week
1	03/01/2024	1	01/01/2024
2	04/01/2024	1	01/01/2024
3	05/01/2024	1	01/01/2024
4	06/01/2024	1	01/01/2024
5	07/01/2024	1	01/01/2024
6	08/01/2024	2	08/01/2024
7	09/01/2024	2	08/01/2024


= Table.Group(#"Inserted Start of Week", {"Week of Year"},
{{"Details", each _,
type table [Date=nullable date, Week of Year=number, Start of Week=date]})
Week of Year
Details
1
1
Table
2
2
Table
Hardcoded table type

Chapter 6: Structured Values



	Product	Size	Category
1	Shoe	Medium	Clothes
2	Hat	Large	Clothes
3	Shirt	Medium	Clothes
4	Belt	Extra Small	Accessories

```
Table.AddColumn(Source, "Max", each List.Max( { [Column1], [Column2], [Column3] } ))
```

	Column1	Column2	Column3	Max
1	5	20	12	5
2	26	8	10	8
3	14	20	90	90

```
Table.AddColumn(Source, "IsMyColor", each List.Contains( { "Blue", "Green", "Orange" }, [Color] ))
```

	Product ID	Color	IsMyColor
1	1	Orange	TRUE
2	2	Pink	FALSE
3	3	Blue	TRUE
4	4	Green	TRUE
5	5	White	FALSE

```
Table.Group(    Source,
    {"Column1", "Column2"},
    {{"Count", each Table.RowCount(_), Int64.Type}})
```

	Column1	Column2	Count
1	A	Ecuador	3
2	B	Taiwan	3
3	B	Siberia	1

```
Table.AddColumn(#"Changed column type", "List", each { 1, 2, 3 } )
```

	ID	List
1	1	[List]
2	2	[List]
3	3	[List]

```
Table.ExpandListColumn(#"Added custom", "List")
```

	ID	List
1	1	1
2	1	2
3	1	3

```
Table.AddColumn(#"Changed column type", "List", each { 1, 2, 3 }, type {Int64.Type} )
```

	ID	List
1	1	[List]
2	2	[List]
3	3	[List]

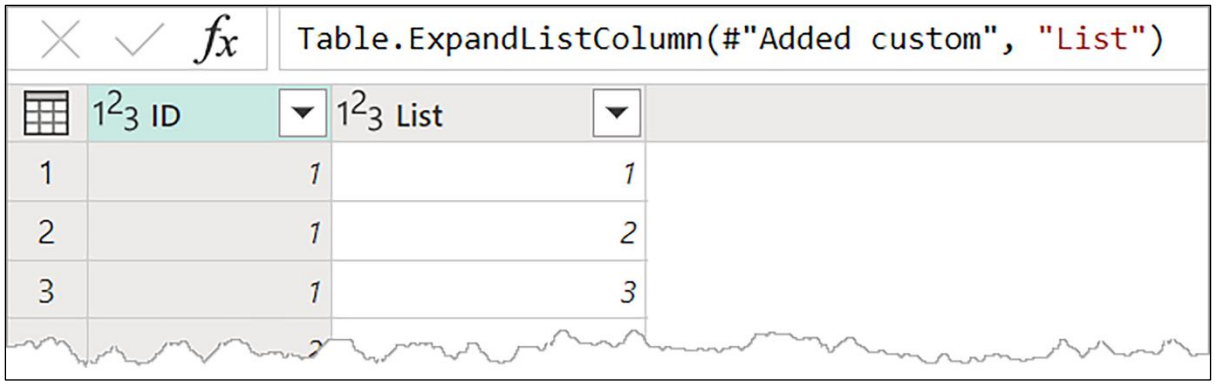
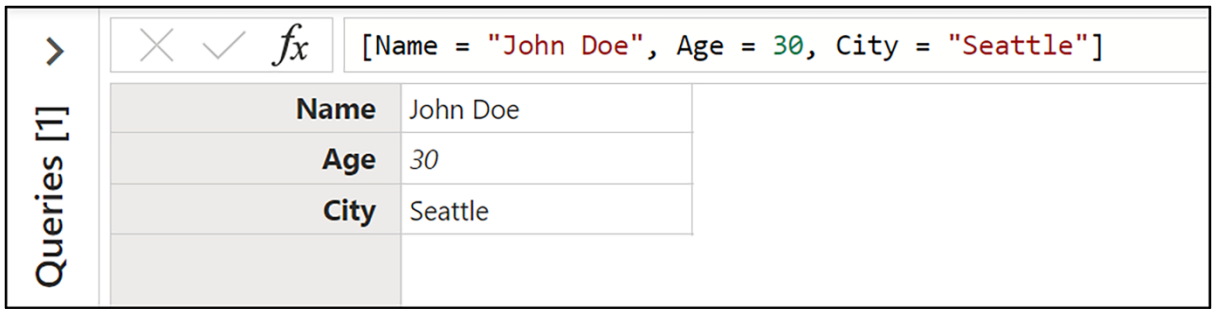
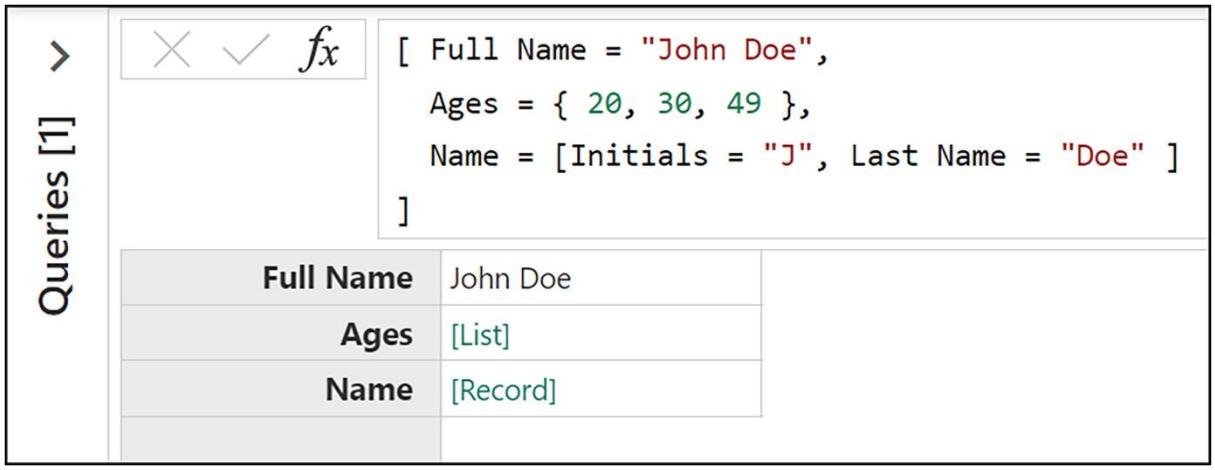


Table.ExpandListColumn(#"Added custom", "List")
ID
List
1 1 1
2 1 2
3 1 3


Queries [1]
[Name = "John Doe", Age = 30, City = "Seattle"]
Name John Doe
Age 30
City Seattle


Queries [1]
[Full Name = "John Doe",
Ages = { 20, 30, 49 },
Name = [Initials = "J", Last Name = "Doe"]
]
Full Name John Doe
Ages [List]
Name [Record]

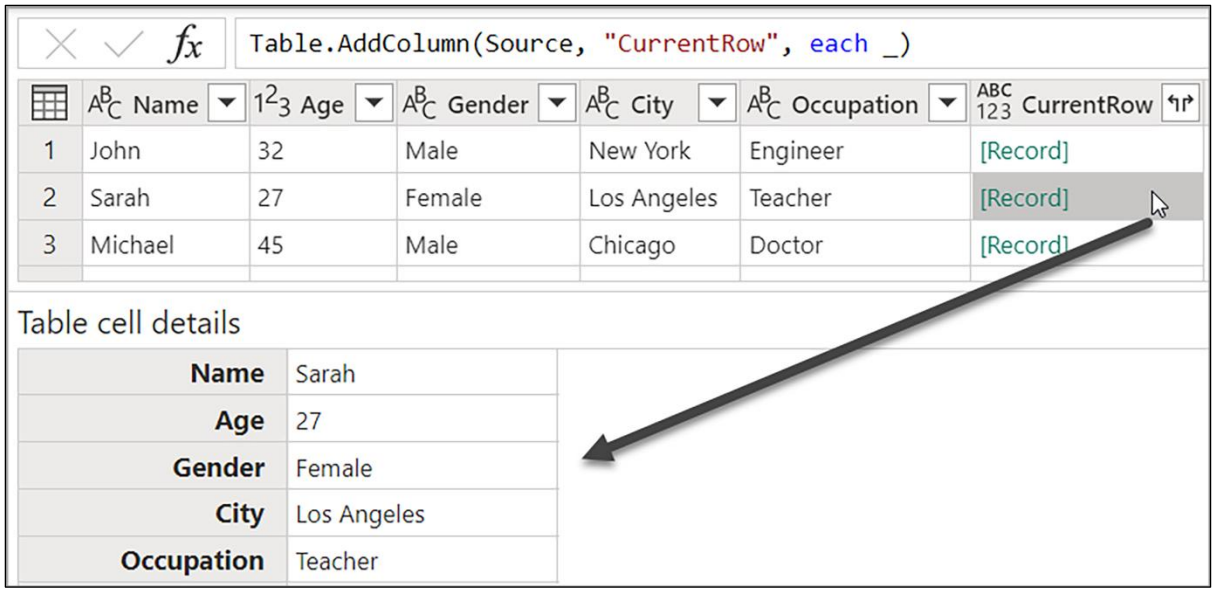
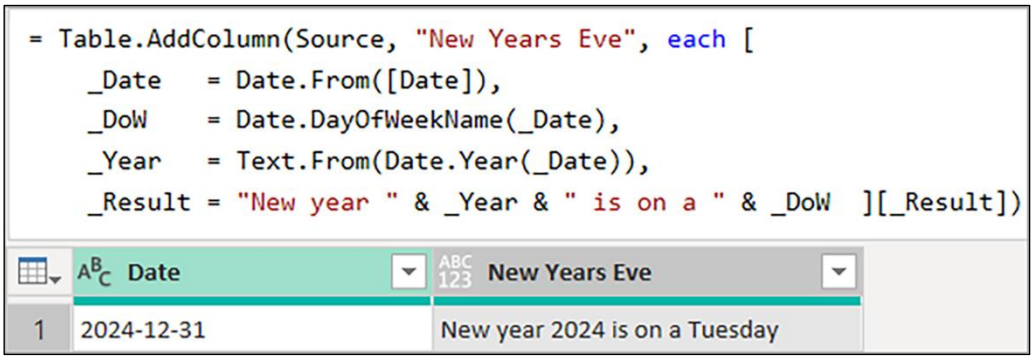
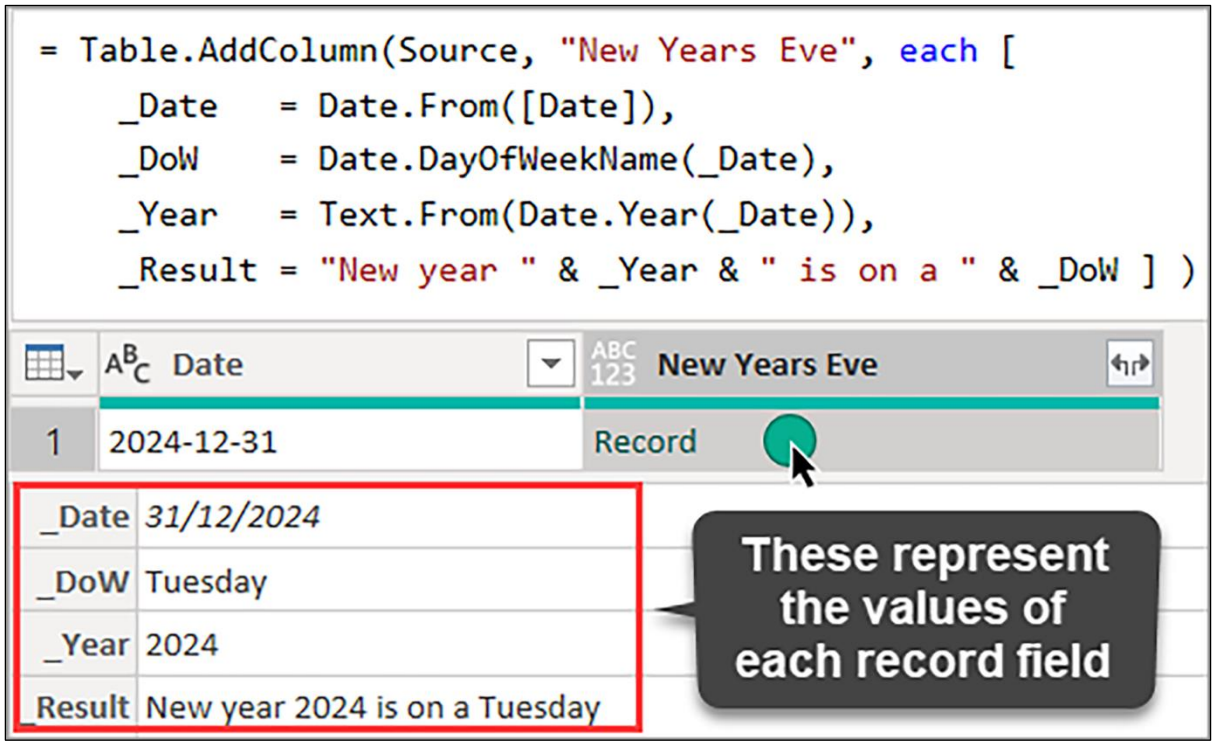


Table.AddColumn(Source, "CurrentRow", each _)
Name
Age
Gender
City
Occupation
CurrentRow
1 John 32 Male New York Engineer [Record]
2 Sarah 27 Female Los Angeles Teacher [Record]
3 Michael 45 Male Chicago Doctor [Record]
Table cell details
Name Sarah
Age 27
Gender Female
City Los Angeles
Occupation Teacher


= Table.AddColumn(Source, "New Years Eve", each [
_Date = Date.From([Date]),
_DoW = Date.DayOfWeekName(_Date),
_Year = Text.From(Date.Year(_Date)),
_Result = "New year " & _Year & " is on a " & _DoW][_Result])
Date
New Years Eve
1 2024-12-31 New year 2024 is on a Tuesday


= Table.AddColumn(Source, "New Years Eve", each [
_Date = Date.From([Date]),
_DoW = Date.DayOfWeekName(_Date),
_Year = Text.From(Date.Year(_Date)),
_Result = "New year " & _Year & " is on a " & _DoW])
Date
New Years Eve
1 2024-12-31 Record
_Date 31/12/2024
_DoW Tuesday
_Year 2024
_Result New year 2024 is on a Tuesday
These represent the values of each record field

	Store	January	February	March	April
1	Denver	501	490	477	522
2	Seattle	780	869	840	700
3	New York	1100	1250	1219	1846

```
Date.ToText(
  #date(2023, 12, 31),
  [Format="dd MMM yyyy", Culture="en-US"]
)
```

31 Dec 2023

```
Date.ToText(
  #date(2023, 12, 31),
  [Format="dd MMM yyyy", Culture="de-DE"]
)
```

31 Dez 2023

```
Table.AddColumn(ChangeColType, "MyRecord", each [ Key = [ID], Product = "Jeans" ] )
```

	ID	MyRecord
1	1	[Record]
2	2	[Record]
3	3	[Record]

```
Table.ExpandRecordColumn(Custom, "MyRecord", {"Key", "Product"}, {"Key", "Product"})
```

	ID	Key	Product
1	1	1	Jeans
2	2	2	Jeans
3	3	3	Jeans

```
Table.AddColumn( ChangeColType, "MyRecord",
each [ Key = [ID],       Product = "Jeans"   ],
type [ Key = Int64.Type, Product = Text.Type ] )
```

	ID	MyRecord
1	1	[Record]
2	2	[Record]
3	3	[Record]

```
Table.ExpandRecordColumn(AddRecord, "MyRecord", {"Key", "Product"}, {"Key", "Product"})
```

	ID	Key	Product
1	1	1	Jeans
2	2	2	Jeans
3	3	3	Jeans

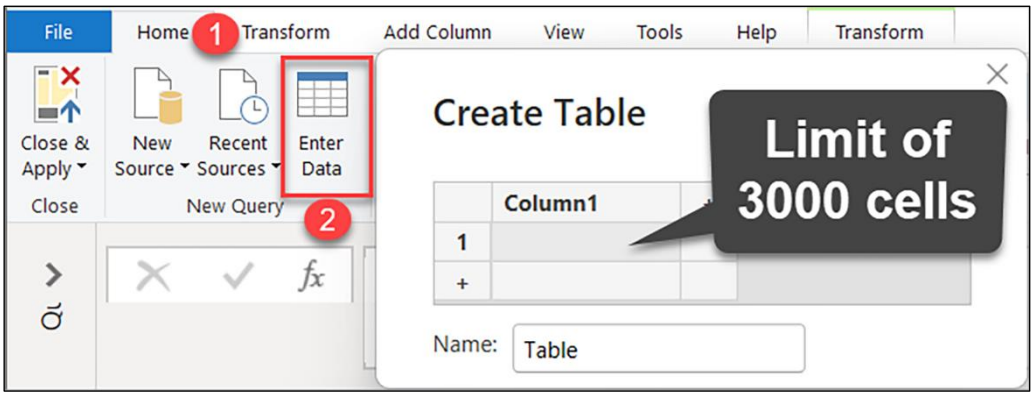
	ID	Name	Country
1	1	John Doe	USA
2	2	Jane Doe	USA
3	3	Jane Doe	Canada

```
#table( {"ID", "Name", "Country"}, {{1, "John Doe", "USA"}}) &
#table( {"ID", "Name", "Country"}, {{2, "Jane Doe", "USA"}})
```

	ID	Name	Country
1	1	John Doe	USA
2	2	Jane Doe	USA

```
#table( {"ID", "Name", "City"}, {{1, "John Doe", "Texas"}}) &
#table( {"ID", "Name", "Country"}, {{2, "Jane Doe", "USA"}})
```

	ID	Name	City	Country
1	1	John Doe	Texas	null
2	2	Jane Doe	null	USA

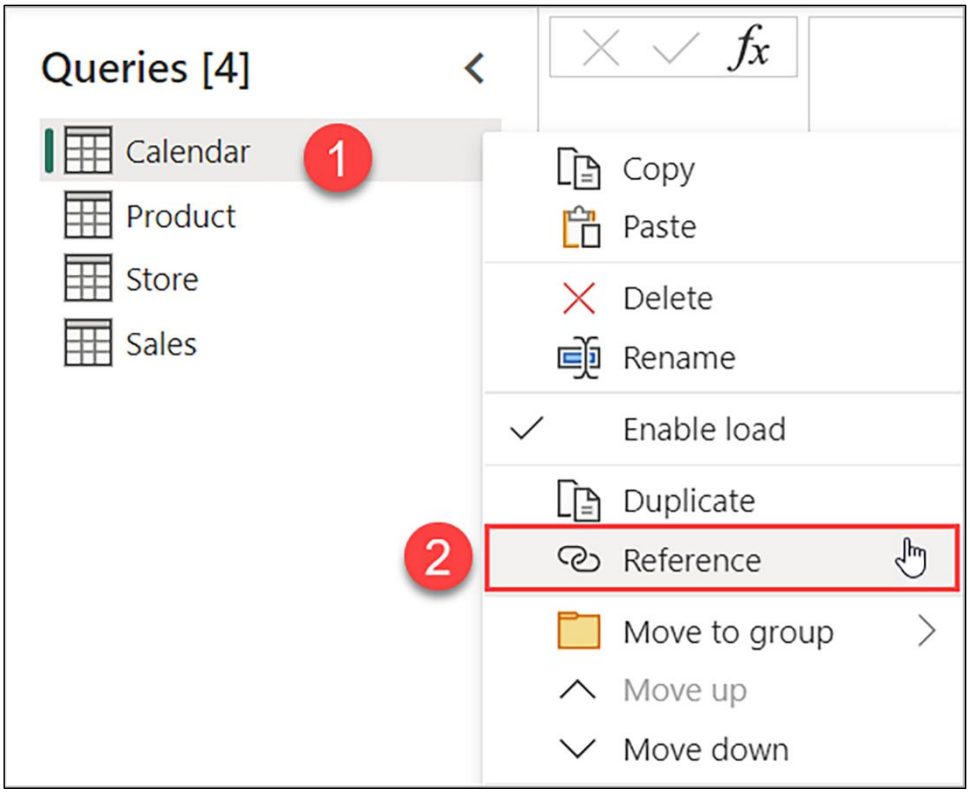


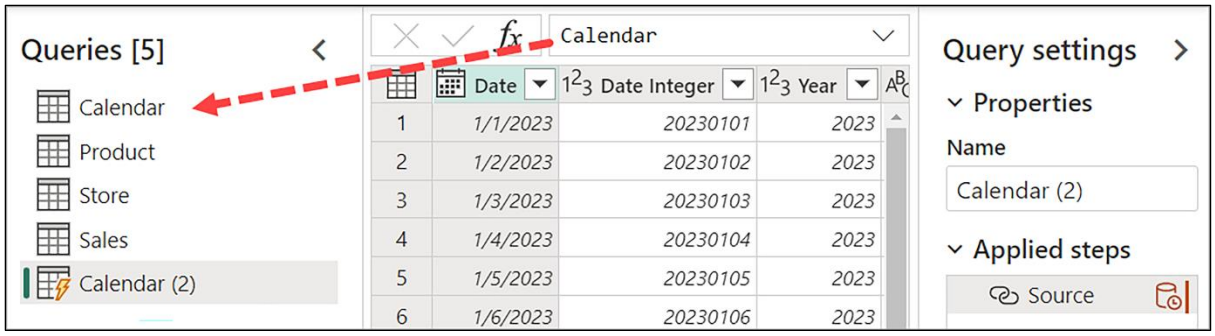
```
#table( { "ProductKey", "Product" },
        {{ 1, "Apple" }, { 2, "Prume" }} )
```

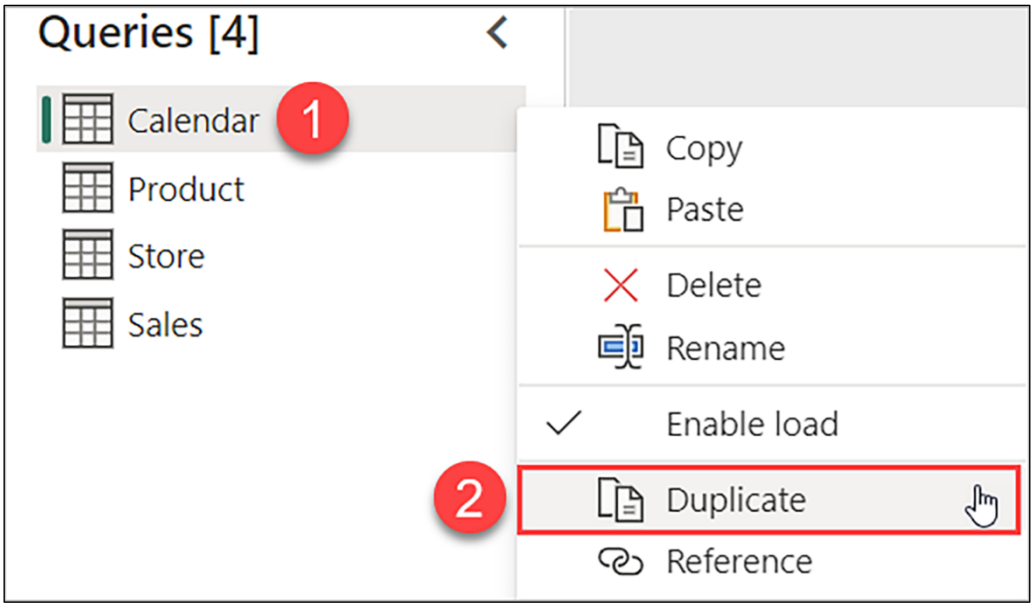
	ABC 123 ProductKey	ABC 123 Product
1	1	Apple
2	2	Prume

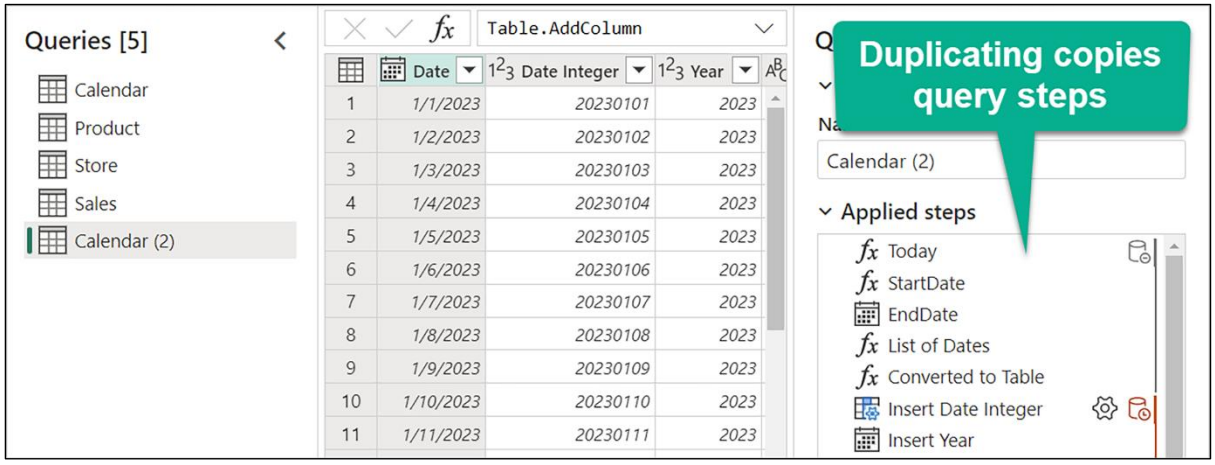
```
Table.FromRecords( {  [ ProductKey = 1, Product = "Apple" ] ,
                      [ ProductKey = 2, Product = "Prume" ]  } )
```

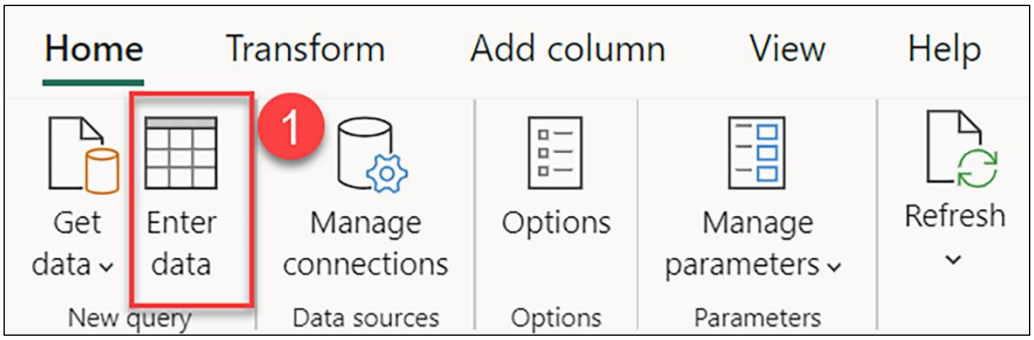
	ABC 123 ProductKey	ABC 123 Product
1	1	Apple
2	2	Prume

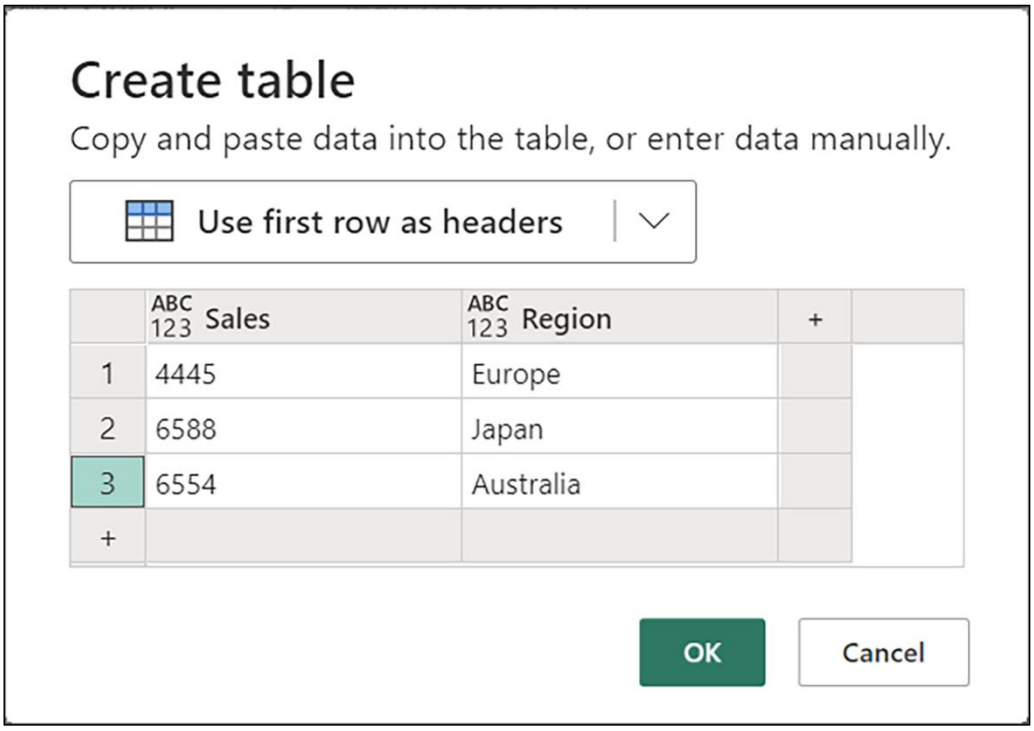


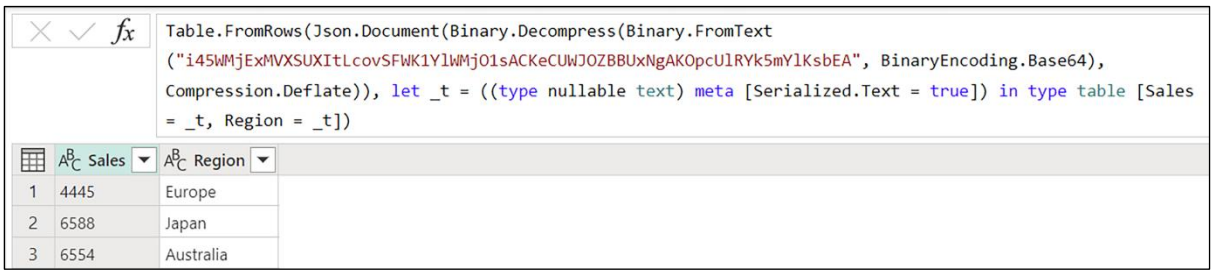

Queries [5]
Calendar
Product
Store
Sales
Calendar (2)
Calendar
Date
Date Integer
Year
1 1/1/2023 20230101 2023
2 1/2/2023 20230102 2023
3 1/3/2023 20230103 2023
4 1/4/2023 20230104 2023
5 1/5/2023 20230105 2023
6 1/6/2023 20230106 2023
Query settings
Properties
Name
Calendar (2)
Applied steps
Source


Queries [4]
Calendar
1
Product
Store
Sales
Copy
Paste
Delete
Rename
Enable load
2
Duplicate
Reference

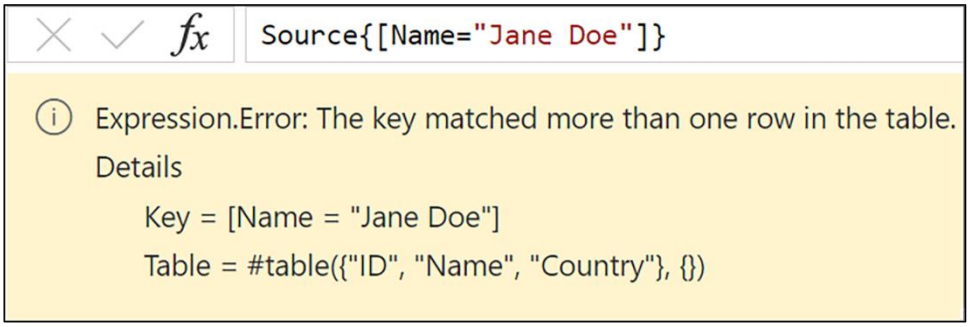

Queries [5]
Calendar
Product
Store
Sales
Calendar (2)
Table.AddColumn
Date
Date Integer
Year
1 1/1/2023 20230101 2023
2 1/2/2023 20230102 2023
3 1/3/2023 20230103 2023
4 1/4/2023 20230104 2023
5 1/5/2023 20230105 2023
6 1/6/2023 20230106 2023
7 1/7/2023 20230107 2023
8 1/8/2023 20230108 2023
9 1/9/2023 20230109 2023
10 1/10/2023 20230110 2023
11 1/11/2023 20230111 2023
Duplicating copies query steps
Calendar (2)
Applied steps
Today
StartDate
EndDate
List of Dates
Converted to Table
Insert Date Integer
Insert Year

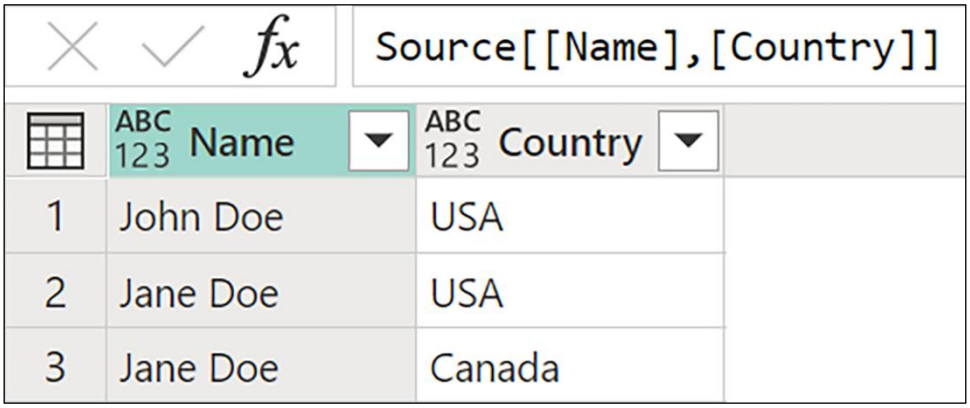




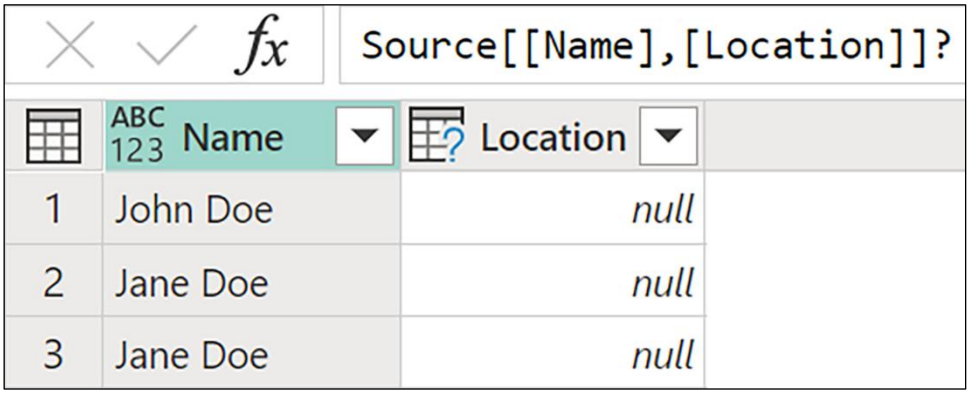


	ABC 123 ID	ABC 123 Name	ABC 123 Country
1	1	John Doe	USA
2	2	Jane Doe	USA
3	3	Jane Doe	Canada





	Name	Country
1	John Doe	USA
2	Jane Doe	USA
3	Jane Doe	Canada



	Name	Location
1	John Doe	*null*
2	Jane Doe	*null*
3	Jane Doe	*null*

```
Table.AddColumn(ChangeColType, "Table",
each #table( {"Key", "Product"}, {{ 1, "Apple" },{ 2, "Prume" }} ))
```

	ID	Table
1	1	[Table]
2	2	[Table]
3	3	[Table]

```
Table.AddColumn(ChangeColType, "Table",
each #table( {"Key", "Product"},
                {{ 1, "Apple" },{ 2, "Prume" }} ),
                type table[ Key = Int64.Type, Product = Text.Type ] )
```

	1²3 ID	Table
1	1	[Table]
2	2	[Table]
3	3	[Table]

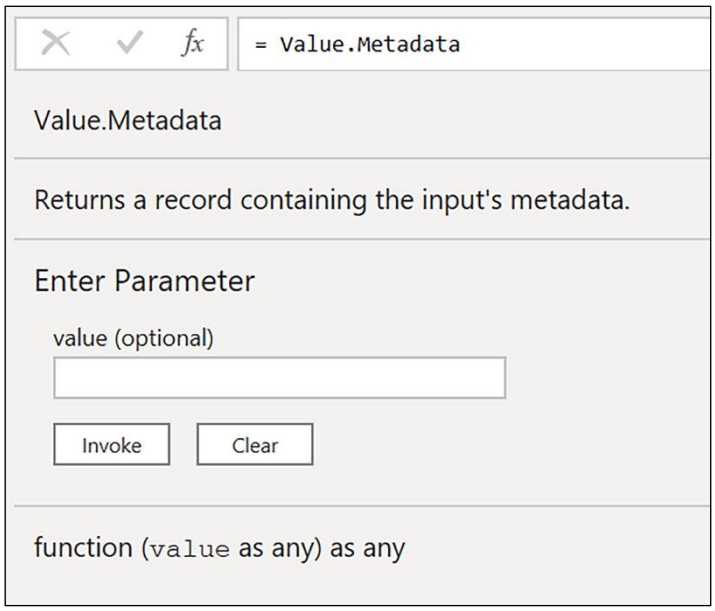
```
Table.ExpandTableColumn(#"Added Custom", "Table", {"Key", "Product"}, {"Key", "Product"})
```

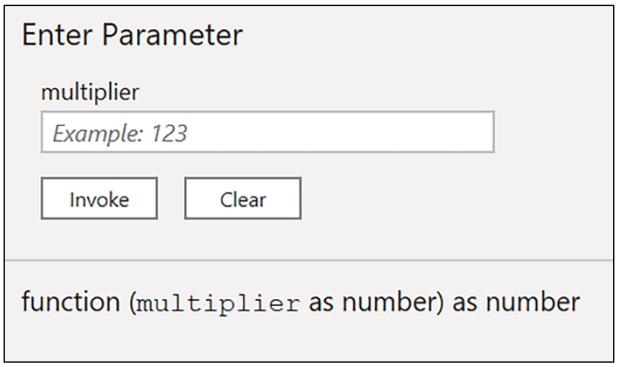
	1²3 ID	1²3 Key	ABC Product
1	1	1	Apple
2	1	2	Prume
3	2	1	Apple

```
Table.AddColumn(ChangeColType, "Custom",
each #table( type table[ Key = Int64.Type, Product = Text.Type ],
                {{ 1, "Apple" },{ 2, "Prume" }} ))
```

	1²3 ID	ABC 123 Custom
1	1	[Table]
2	2	[Table]
3	3	[Table]

Chapter 7: Conceptualizing M





multiplyFunction

Multiplies the number 10 by the multiplier.

Enter Parameter

multiplier

Example: 123

Invoke Clear

function (`multiplier` as number) as number

Example: Multiply by 1

Usage:

```
multiplyFunction(1)
```

Output:

```
10
```

Example: Multiply by 2

Usage:

```
multiplyFunction(2)
```

Output:

```
20
```

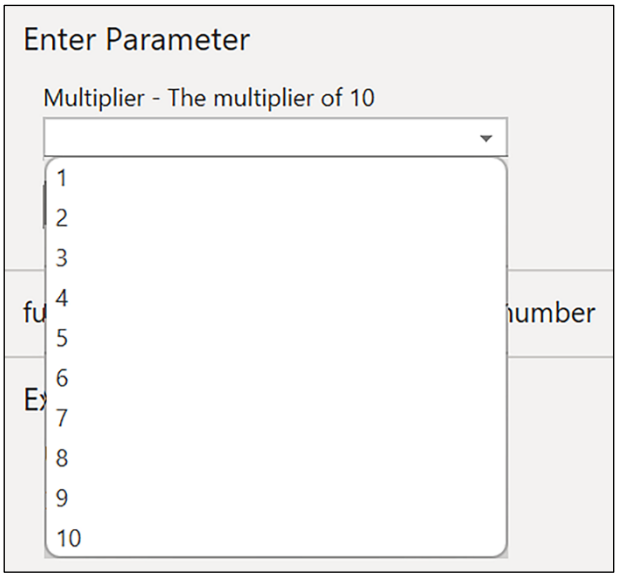
Example: Multiply by 3

Usage:

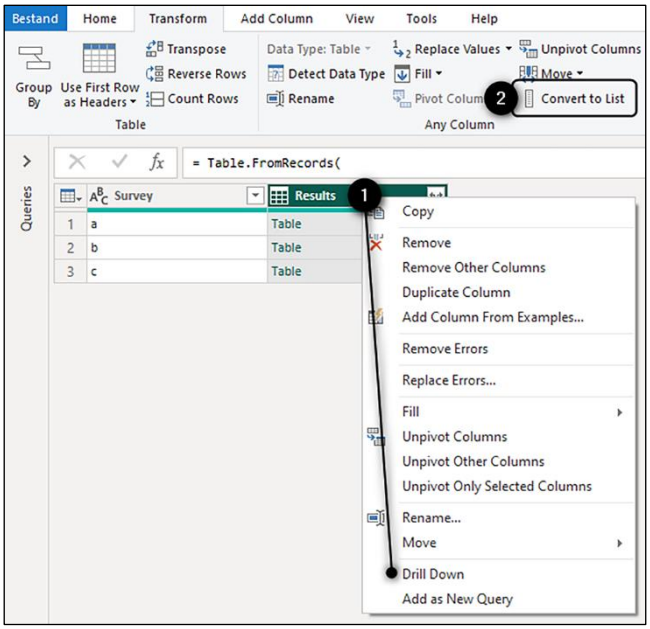
```
multiplyFunction(3)
```

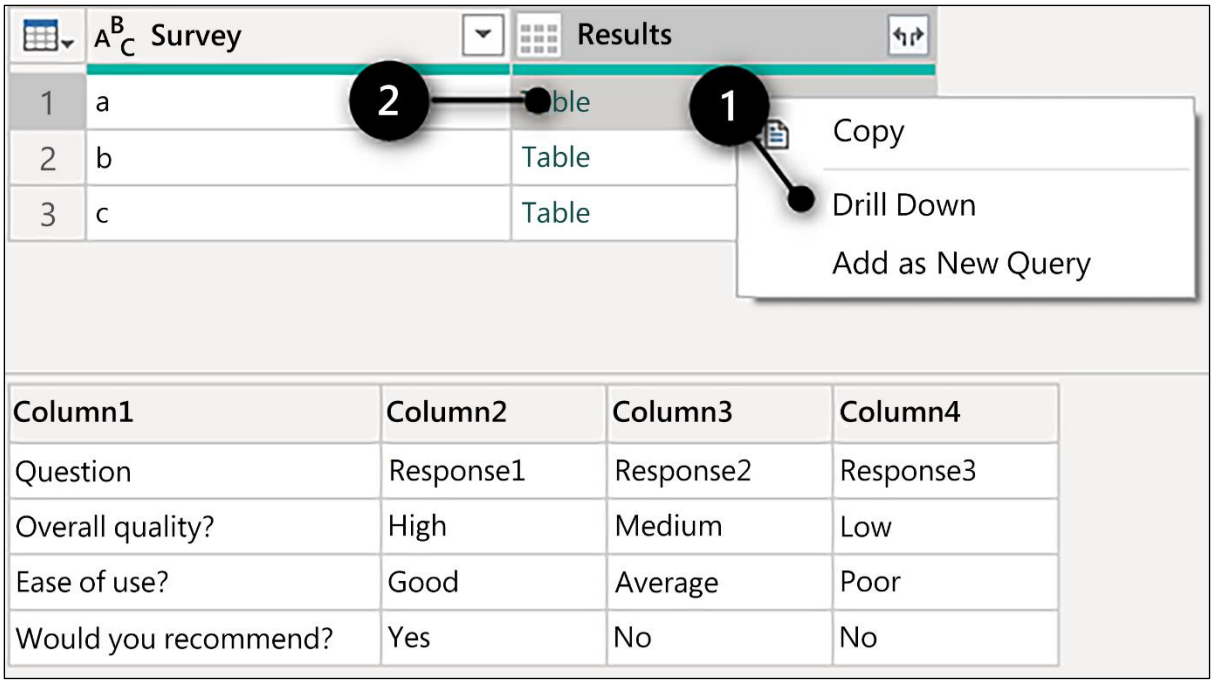
Output:

```
30
```

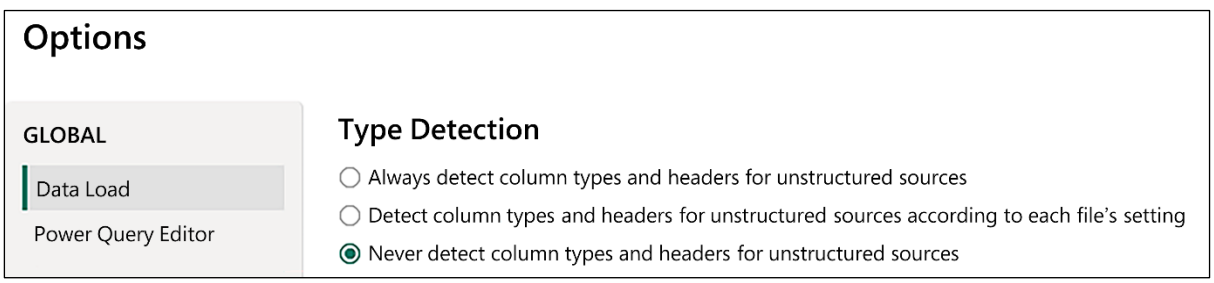

Enter Parameter
Multiplier - The multiplier of 10
1
2
3
4
5
6
7
8
9
10

Chapter 8: Working with Nested Structures

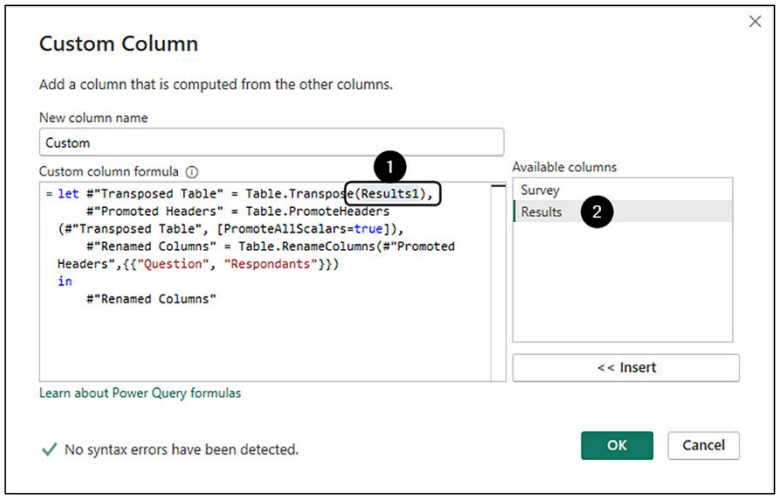




Column1	Column2	Column3	Column4
Question	Response1	Response2	Response3
Overall quality?	High	Medium	Low
Ease of use?	Good	Average	Poor
Would you recommend?	Yes	No	No

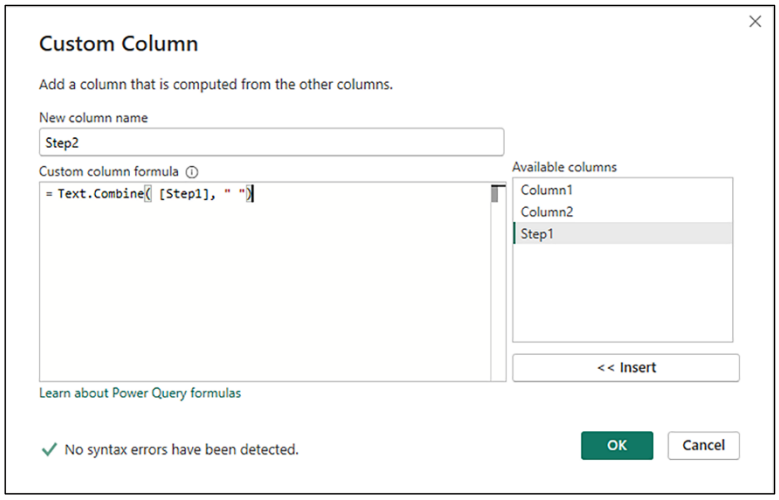


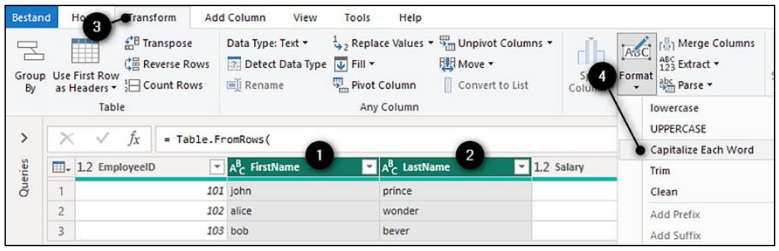
```
    Results1 = Source{0}[Results],
    #"Transposed Table" = Table.Transpose(Results1),
    #"Promoted Headers" = Table.PromoteHeaders(#"Transposed Table", [PromoteAllScalars=true]),
    #"Renamed Columns" = Table.RenameColumns(#"Promoted Headers",{{"Question", "Respondants"}})
in
    #"Renamed Columns"
```

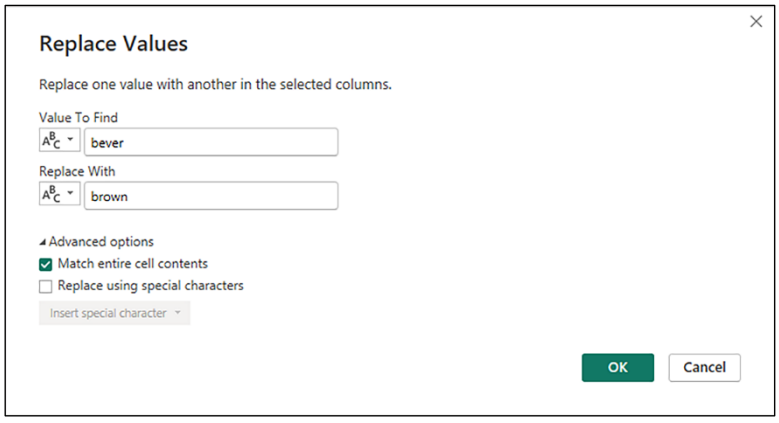


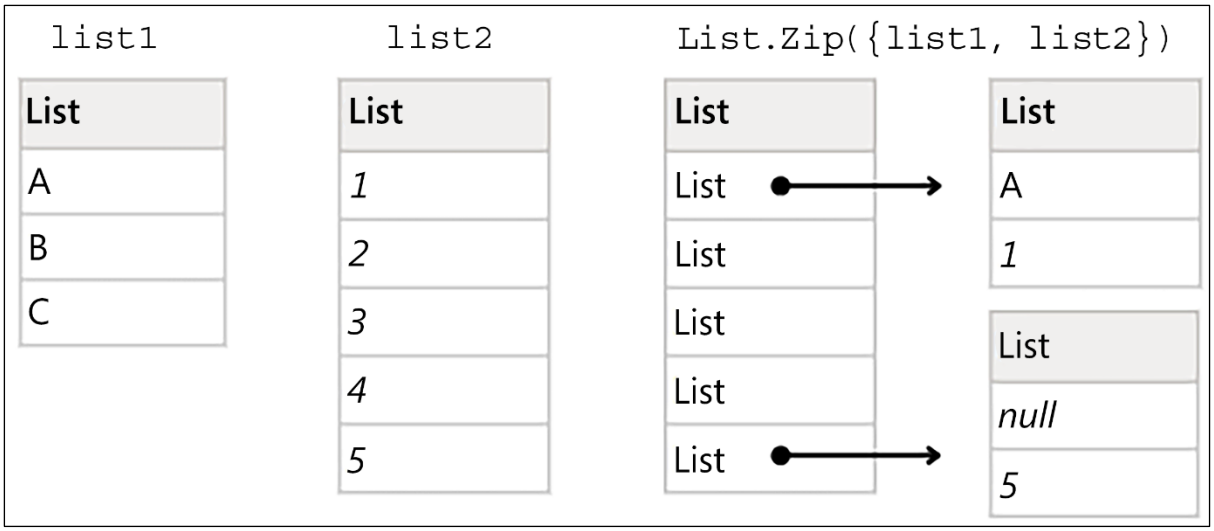
	Survey	Results	Custom
1	a	Table	Table
2	b	Table	Table
3	c	Table	Table

Respondants	Overall quality?	Ease of use?	Would you recommend?
Response1	High	Good	Yes
Response2	Medium	Average	No
Response3	Low	Poor	No


Custom Column
Add a column that is computed from the other columns.
New column name
Step2
Custom column formula
= Text.Combine([Step1], " ")
Available columns
Column1
Column2
Step1
<< Insert
Learn about Power Query formulas
No syntax errors have been detected.
OK
Cancel

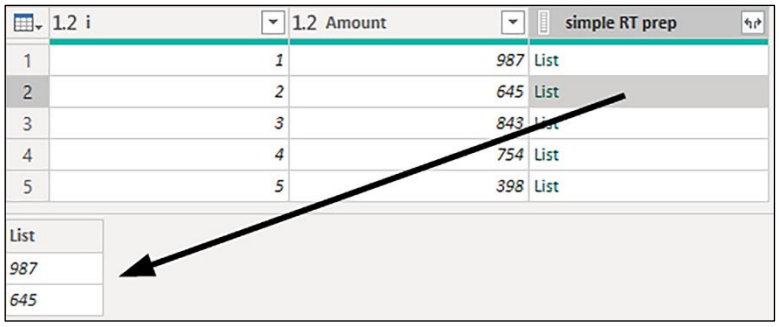

Bestand
Transform
Add Column
View
Tools
Help
Group By
Use First Row as Headers
Transpose
Reverse Rows
Count Rows
Table
Data Type: Text
Detect Data Type
Rename
Replace Values
Fill
Pivot Column
Unpivot Columns
Move
Convert to List
Any Column
Format
Merge Columns
Extract
Parse
lowercase
UPPERCASE
Capitalize Each Word
Trim
Clean
Add Prefix
Add Suffix
Queries
= Table.FromRows(
EmployeeID
FirstName
LastName
Salary
1 101 john prince
2 102 alice wonder
3 103 bob bever


Replace Values
Replace one value with another in the selected columns.
Value To Find
bever
Replace With
brown
Advanced options
Match entire cell contents
Replace using special characters
Insert special character
OK
Cancel



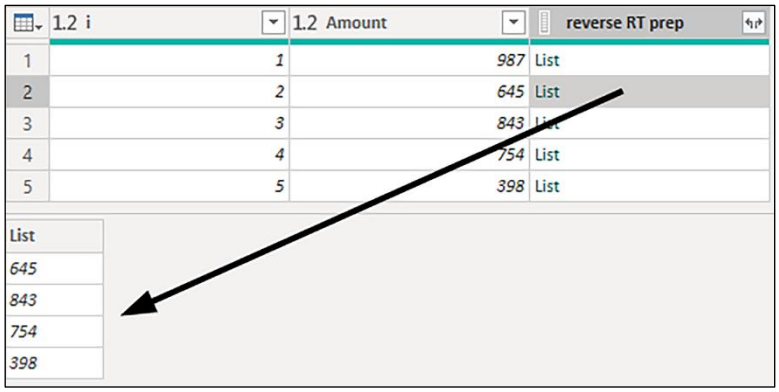
myEmptyList	List
listFirst	null
listLast	null
listSingle	Error
singleOrDefault	null
singleOrDefault2	99
myNonEmptyList	List
listFirst2	1
listLast2	3
listSingle2	Error
singleOrDefault3	Error

List
987
645
843
754
398



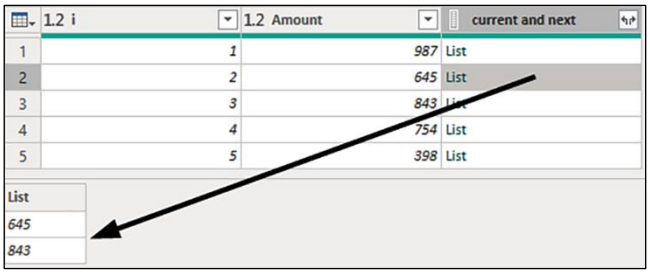
	1.2 i	1.2 Amount	simple RT prep
1	1	987	List
2	2	645	List
3	3	843	List
4	4	754	List
5	5	398	List

List
987
645



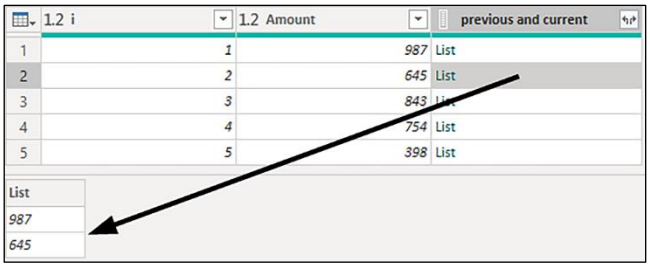
	1.2 i	1.2 Amount	reverse RT prep
1	1	987	List
2	2	645	List
3	3	843	List
4	4	754	List
5	5	398	List

List
645
843
754
398



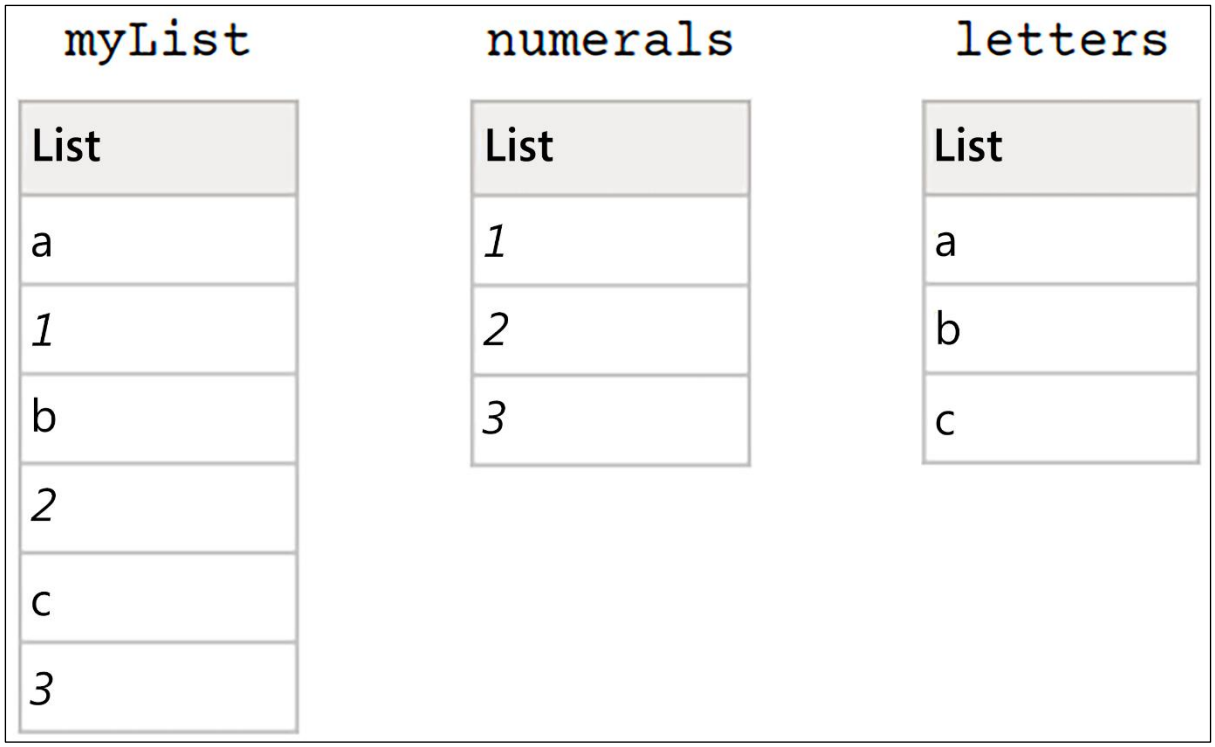
	1.2 i	1.2 Amount	current and next
1	1	987	List
2	2	645	List
3	3	843	List
4	4	754	List
5	5	398	List

List
645
843



	1.2 i	1.2 Amount	previous and current
1	1	987	List
2	2	645	List
3	3	843	List
4	4	754	List
5	5	398	List

List
987
645


myList
List
a
1
b
2
c
3
numerals
List
1
2
3
letters
List
a
b
c


myList
List
a
1
b
2
c
3
ba
ByType
List
1
2
3
IsOdd
List
1
3
ByLen
List
ba

tblColNames

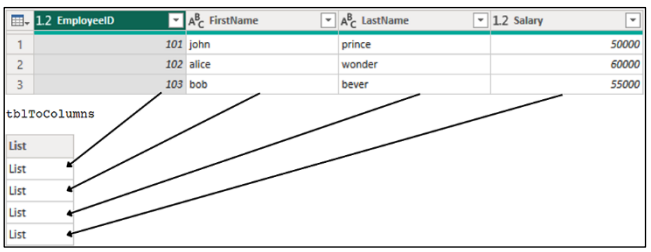
List
EmployeeID
FirstName
LastName
Salary

recFieldNames

List
EmployeeID
FirstName
LastName
Salary

tblColumnsOfType

List
FirstName
LastName



tblToRows	tblToList	tblToRecords	recFieldValues, recToList
List	List	List	List
List	101,john,prince,50000	Record	*101*
List	102,alice,wonder,60000	Record	john
List	103,bob,bever,55000	Record	prince
			50000

tblSplit

List
Table
Table
Table

lst

List
List
List
List

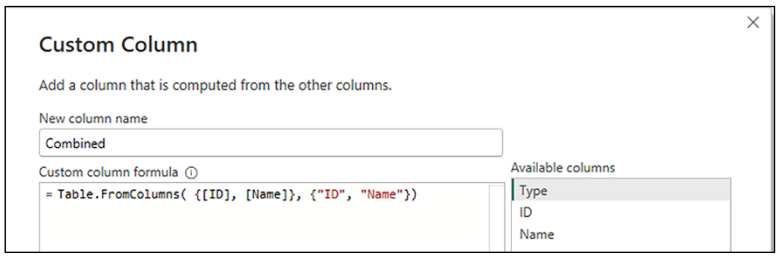
lstCombine

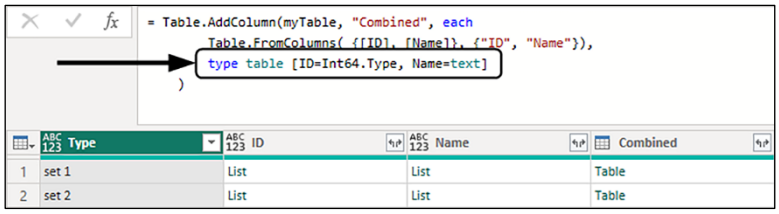
List
101
john
prince
50000
102
alice
wonder
60000
103
bob
bever
55000

lstSplit

List
List
List
List

	ABC 123 Type	ABC 123 ID	ABC 123 Name
1	set 1	List	List
2	set 2	List	List



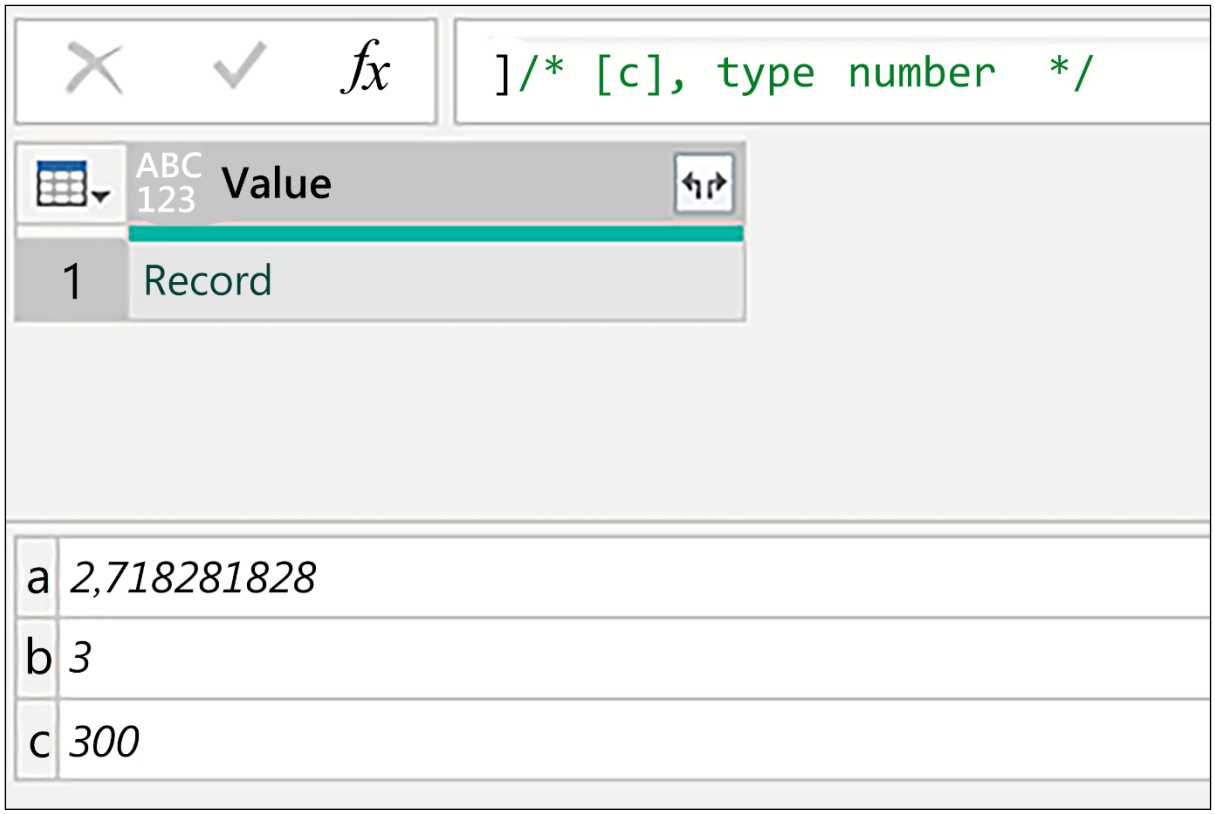


EmployeeID	102
FirstName	Alice
LastName	Wonder
Salary	6000.0

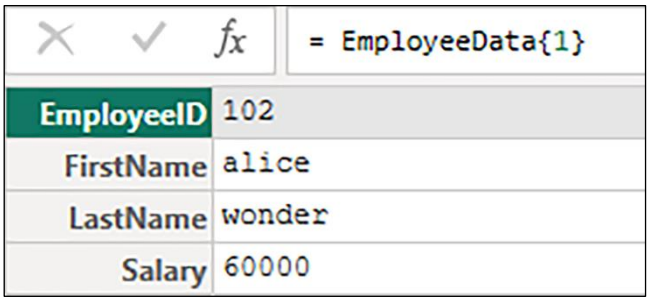
EmployeeID	102
FirstName	Alice
LastName	Wonder
Salary	6000

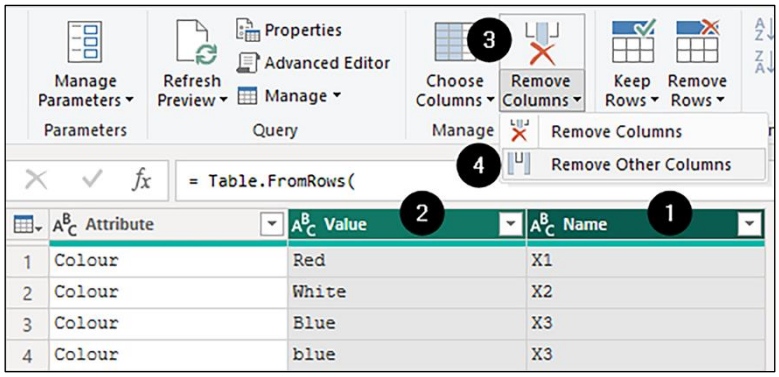
rec	Record
recField	Alice
recField2	Error
recFieldOrDefault	Alice
recFieldOrDefault2	null
recFieldOrDefault3	Unknown

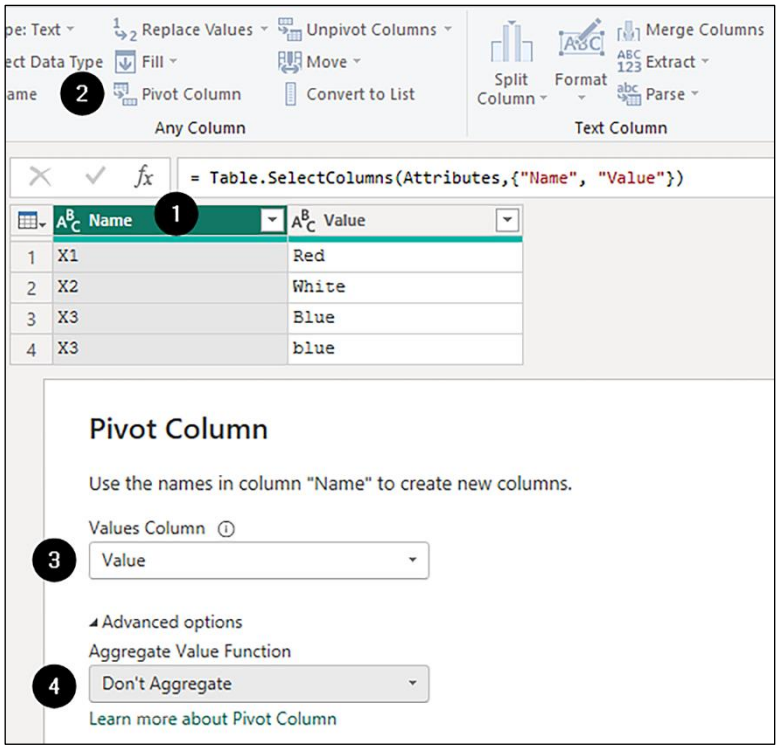
An error occurred in the '' query. Expression.Error: The field 'FirstNames' of the record wasn't found.



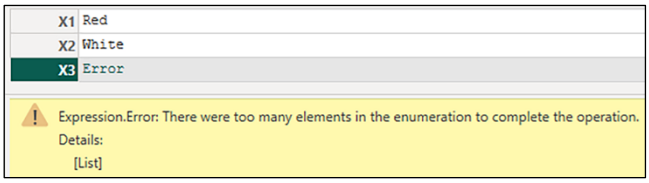
rec	Record
recRemoveFields	Record
recRemoveFields2	Record
recAddField	Record
recAddField2	Record
recCombine	Record
recSelectFields	Record
recSelectFields2	Record
EmployeeID	102
FirstName	Alice
LastName	Wonder
Salary	6000.0
fxGreeting	Hi!





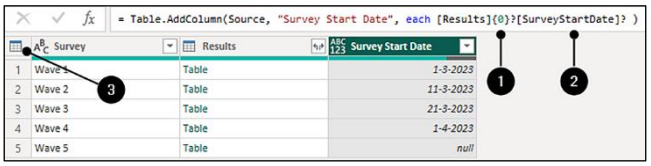



```
= Table.ToRecords( Table.Pivot(#"Removed Other Columns", List.Distinct(#"Removed Other Columns"[Name]), "Name", "Value"))
```



	Survey	Results
1	Wave 1	Table
2	Wave 2	Table
3	Wave 3	Table
4	Wave 4	Table
5	Wave 5	Table

RespondentID	Age	Gender	ExperienceLevel	OverallQ
456	30	Female	Intermediate	High
457	52	Male	Expert	Medium
458	24	Female	Beginner	High
459	43	Male	Intermediate	Low
460	28	Non-binary	Expert	High



Wave 1, first 2 cols

RespondentID	Age
456	30
457	52
458	24
459	43
460	28

NoCountOrCondition

RespondentID	Age
457	52
458	24
459	43
460	28

Count

RespondentID	Age
458	24
459	43
460	28

Condition

RespondentID	Age
457	52
458	24
459	43
460	28

Wave 1, col1

RespondentID
456
457
458
459
460

Step1, col V1

RespondentID
457
459

Step2, col V2

RespondentID
456
457
458
459
460

Step3, col V3

RespondentID
456

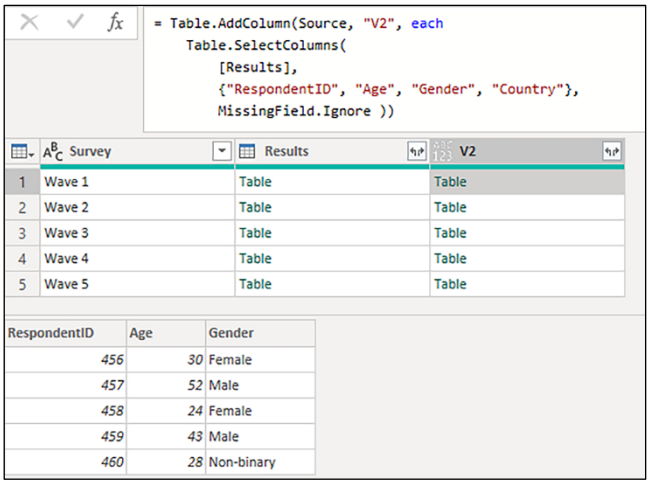
Step4, col V4

RespondentID
456
458
460

```
= Table.AddColumn(Source, "V1", each
    Table.SelectColumns(
        [Results],
        {"RespondentID", "Age", "Gender", "Country"} ))
```

	Survey	Results	V1
1	Wave 1	Table	Error
2	Wave 2	Table	Error
3	Wave 3	Table	Error
4	Wave 4	Table	Error
5	Wave 5	Table	Error

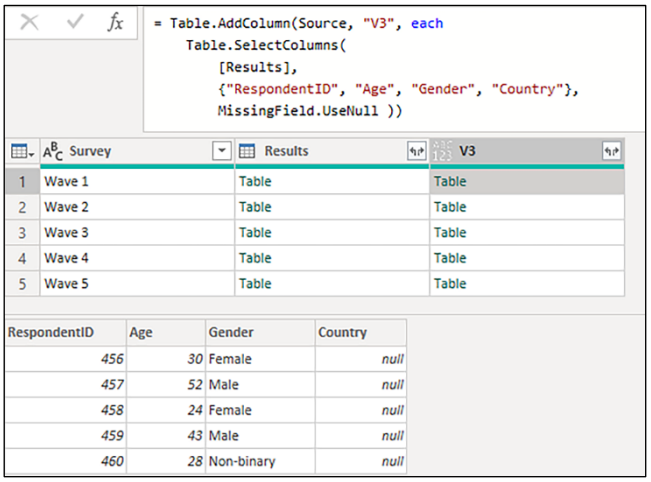
Expression.Error: The column 'Country' of the table wasn't found.
Details:
Country



```
= Table.AddColumn(Source, "V2", each
    Table.SelectColumns(
        [Results],
        {"RespondentID", "Age", "Gender", "Country"},
        MissingField.Ignore ))
```

	Survey	Results	V2
1	Wave 1	Table	Table
2	Wave 2	Table	Table
3	Wave 3	Table	Table
4	Wave 4	Table	Table
5	Wave 5	Table	Table

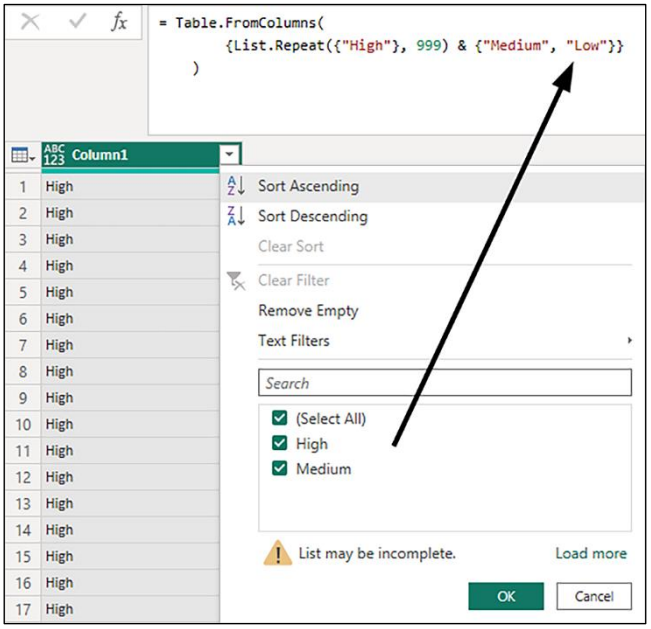
RespondentID	Age	Gender
456	30	Female
457	52	Male
458	24	Female
459	43	Male
460	28	Non-binary



```
= Table.AddColumn(Source, "V3", each
    Table.SelectColumns(
        [Results],
        {"RespondentID", "Age", "Gender", "Country"},
        MissingField.UseNull ))
```

	Survey	Results	V3
1	Wave 1	Table	Table
2	Wave 2	Table	Table
3	Wave 3	Table	Table
4	Wave 4	Table	Table
5	Wave 5	Table	Table

RespondentID	Age	Gender	Country
456	30	Female	null
457	52	Male	null
458	24	Female	null
459	43	Male	null
460	28	Non-binary	null



```
= Table.FromColumns(
        {List.Repeat({"High"}, 999) & {"Medium", "Low"}}
    )
```

	Column1
1	High
2	High
3	High
4	High
5	High
6	High
7	High
8	High
9	High
10	High
11	High
12	High
13	High
14	High
15	High
16	High
17	High

ExperienceLevel	OverallQuality	EaseOfUse	WouldRecommend
Intermediate	High	Good	Yes
Expert	High	Good	Yes

ExperienceLevel	OverallQuality	EaseOfUse	WouldRecommend	Validation
Intermediate	High	Good	Yes	TRUE
Expert	Medium	Good	Yes	FALSE
Beginner	High	Excellent	Yes	FALSE
Intermediate	Low	Average	No	FALSE
Expert	High	Good	Yes	TRUE

SalesData

	OrderID	Total Value
1	ORD-123	149
2	ORD-124	650
3	ORD-125	749
4	ORD-126	543
5	ORD-127	324
6	ORD-128	1685
7	ORD-129	750
8	ORD-130	999

DiscountRates

	Value	Discount
1	750	0,02
2	1500	0,05

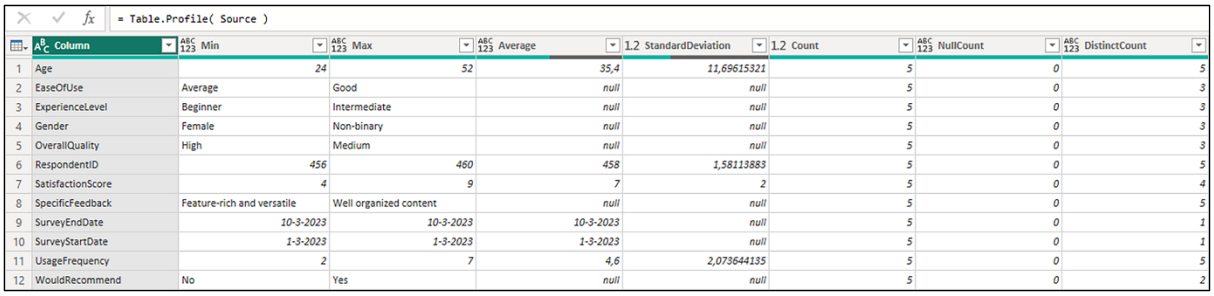
```
= Table.AddColumn(Source, "recToTable", each Record.ToTable(_))
```

	EmployeeID	FirstName	LastName	Salary	recToTable
1	101	john	prince	50000	Table
2	102	alice	wonder	60000	Table
3	103	bob	bever	55000	Table

Name	Value
EmployeeID	101
FirstName	john
LastName	prince
Salary	50000

```
= Table.Schema( Source )
```

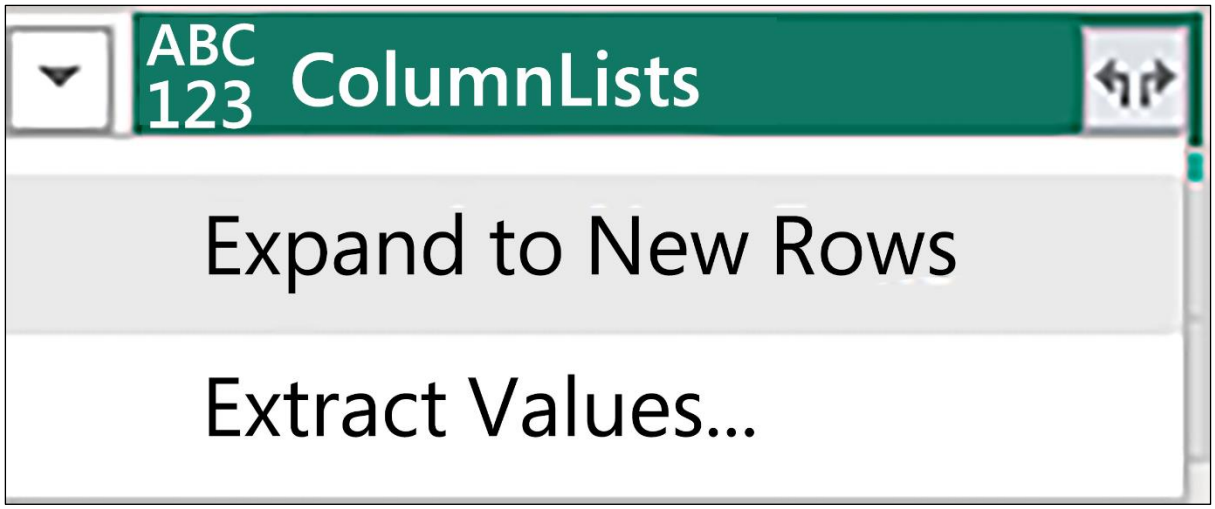
	Name	Position	TypeName	Kind	IsNullable
1	RespondentID	0	Int64.Type	number	TRUE
2	Age	1	Int64.Type	number	TRUE
3	Gender	2	Text.Type	text	TRUE
4	ExperienceLevel	3	Text.Type	text	TRUE
5	OverallQuality	4	Text.Type	text	TRUE
6	EaseOfUse	5	Text.Type	text	TRUE
7	WouldRecommend	6	Text.Type	text	TRUE
8	SpecificFeedback	7	Text.Type	text	TRUE
9	SurveyStartDate	8	Date.Type	date	TRUE
10	SurveyEndDate	9	Date.Type	date	TRUE
11	UsageFrequency	10	Int64.Type	number	TRUE
12	SatisfactionScore	11	Int64.Type	number	TRUE

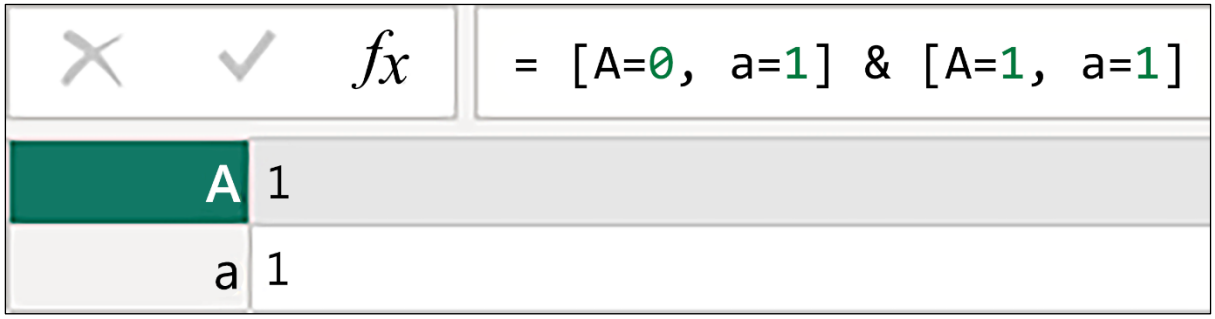


= Table.Profile(Source)

	Column	Min	Max	Average	StandardDeviation	Count	NullCount	DistinctCount
1	Age	24	52	35,4	11,69615321	5	0	5
2	EaseOfUse	Average	Good	null	null	5	0	3
3	ExperienceLevel	Beginner	Intermediate	null	null	5	0	3
4	Gender	Female	Non-binary	null	null	5	0	3
5	OverallQuality	High	Medium	null	null	5	0	3
6	RespondentID	456	460	458	1,58113883	5	0	5
7	SatisfactionScore	4	9	7	2	5	0	4
8	SpecificFeedback	Feature-rich and versatile	Well organized content	null	null	5	0	5
9	SurveyEndDate	10-3-2023	10-3-2023	10-3-2023	null	5	0	1
10	SurveyStartDate	1-3-2023	1-3-2023	1-3-2023	null	5	0	1
11	UsageFrequency	2	7	4,6	2,073644135	5	0	5
12	WouldRecommend	No	Yes	null	null	5	0	2

	Column1	Column2
1	Column	myNumber
2	Min	1
3	Max	text
4	Average	Error
5	StandardDeviation	Error
6	Count	204
7	NullCount	3
8	DistinctCount	202
9	count of Elements	204
10	count No Nulls	201
11	raise Type Error	Error
12	count of Numbers	200

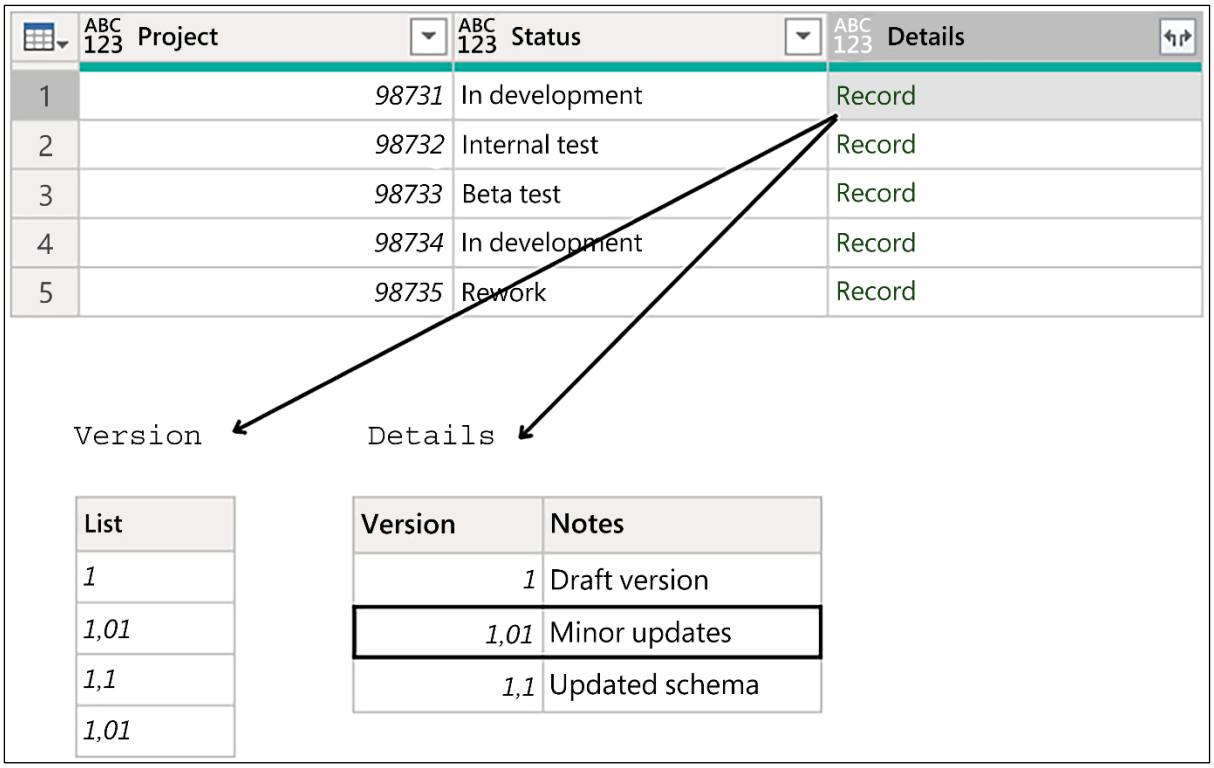




	Column1	Column2	Column3	Column4
1	List		List	List
2	List	List	Record	Record
3	21-1-2024		List	List
4	7-3-2024		List	List
5	List	List	Record	Record

	Column1	Column2	Column3	Column4
1	Let's, go		1, 2	3, 4, 5, 6, 7, 8, 9
2		true, true, false	a = 1, b = 2	a = 1, b = 2
3	21-1-2024		1, 2	3, 4, 5, 6, 7, 8, 9
4	7-3-2024		1, 2	3, 4, 5, 6, 7, 8, 9
5		true, true, false	a = 1, b = 2	a = 1, b = 2

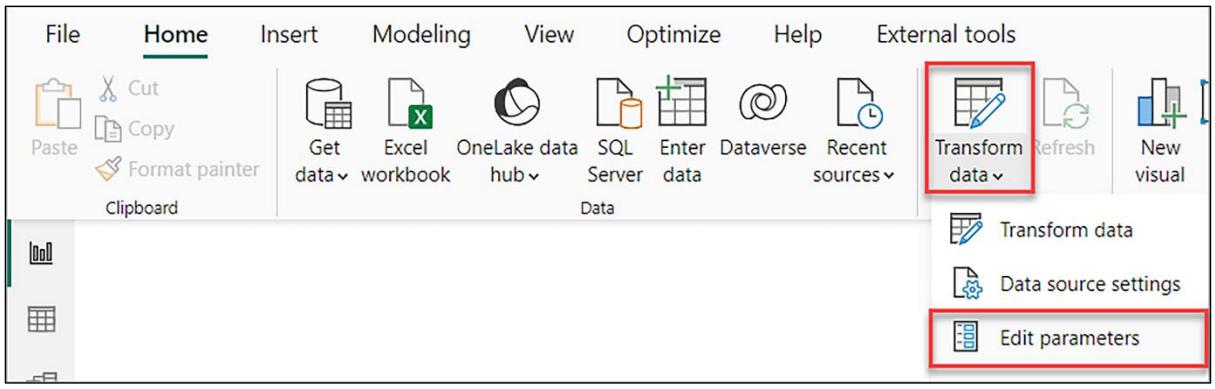
	ABC 123 Project	ABC 123 Status	ABC 123 Details
1	*98731*	In development	Record
2	*98732*	Internal test	Record
3	*98733*	Beta test	Record
4	*98734*	In development	Record
5	*98735*	Rework	Record

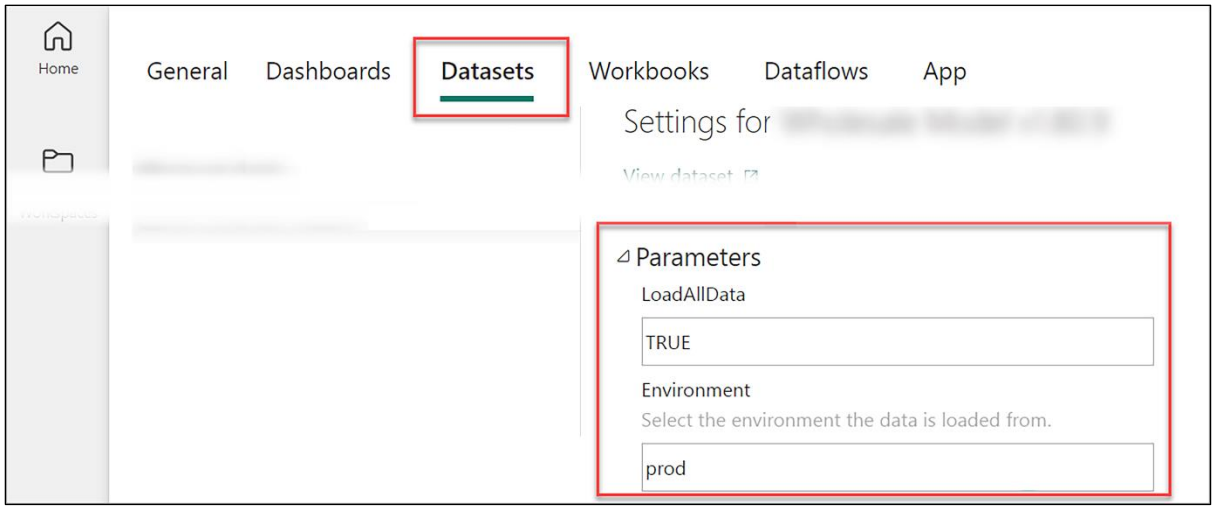


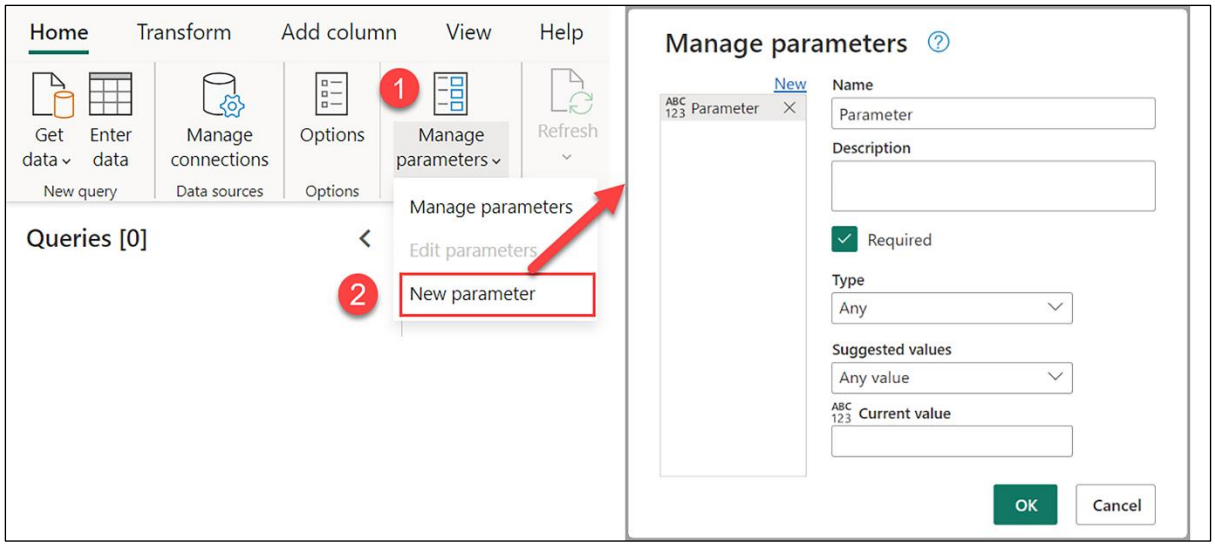
List
1
1,01
1,1
1,01

Version	Notes
1	Draft version
1,01	Minor updates
1,1	Updated schema

Chapter 9: Parameters and Custom Functions







```
#date(2024, 1, 31) meta [IsParameterQuery = true, IsParameterQueryRequired = true, Type = type date]
```

Current value

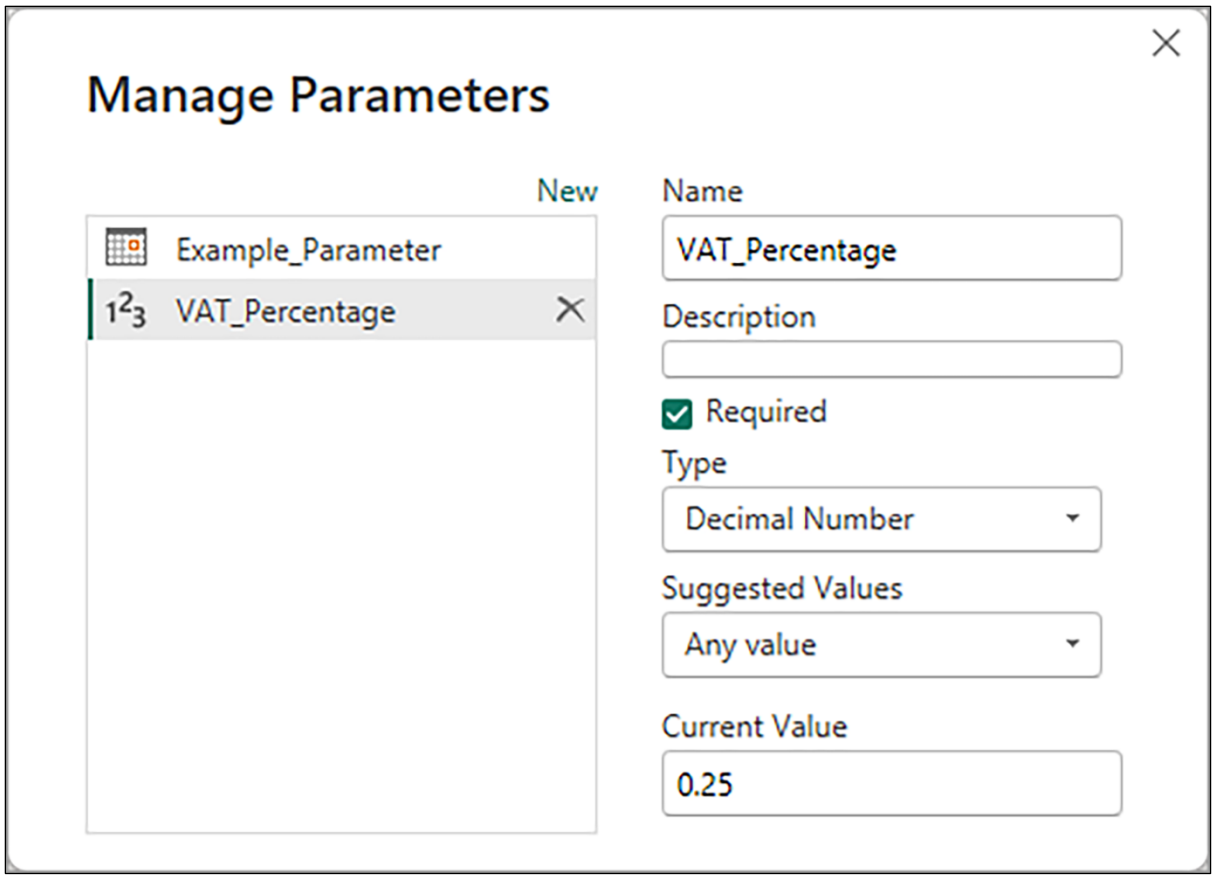
1/31/2024

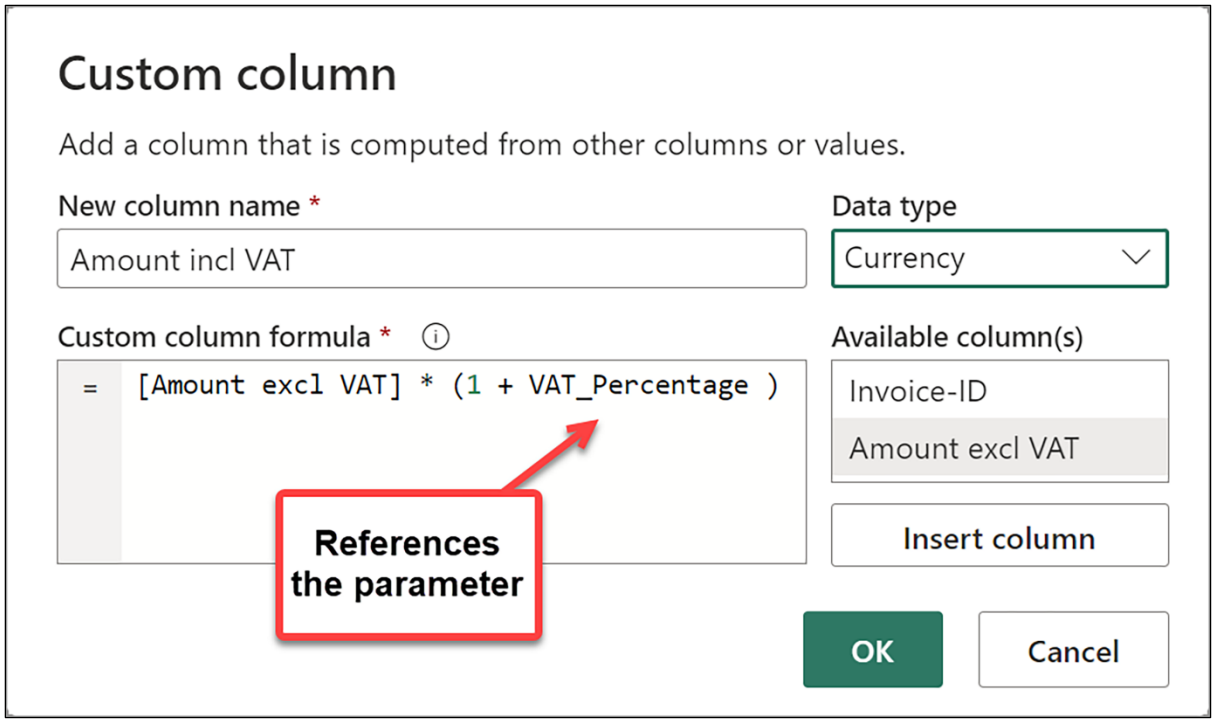
Manage parameter

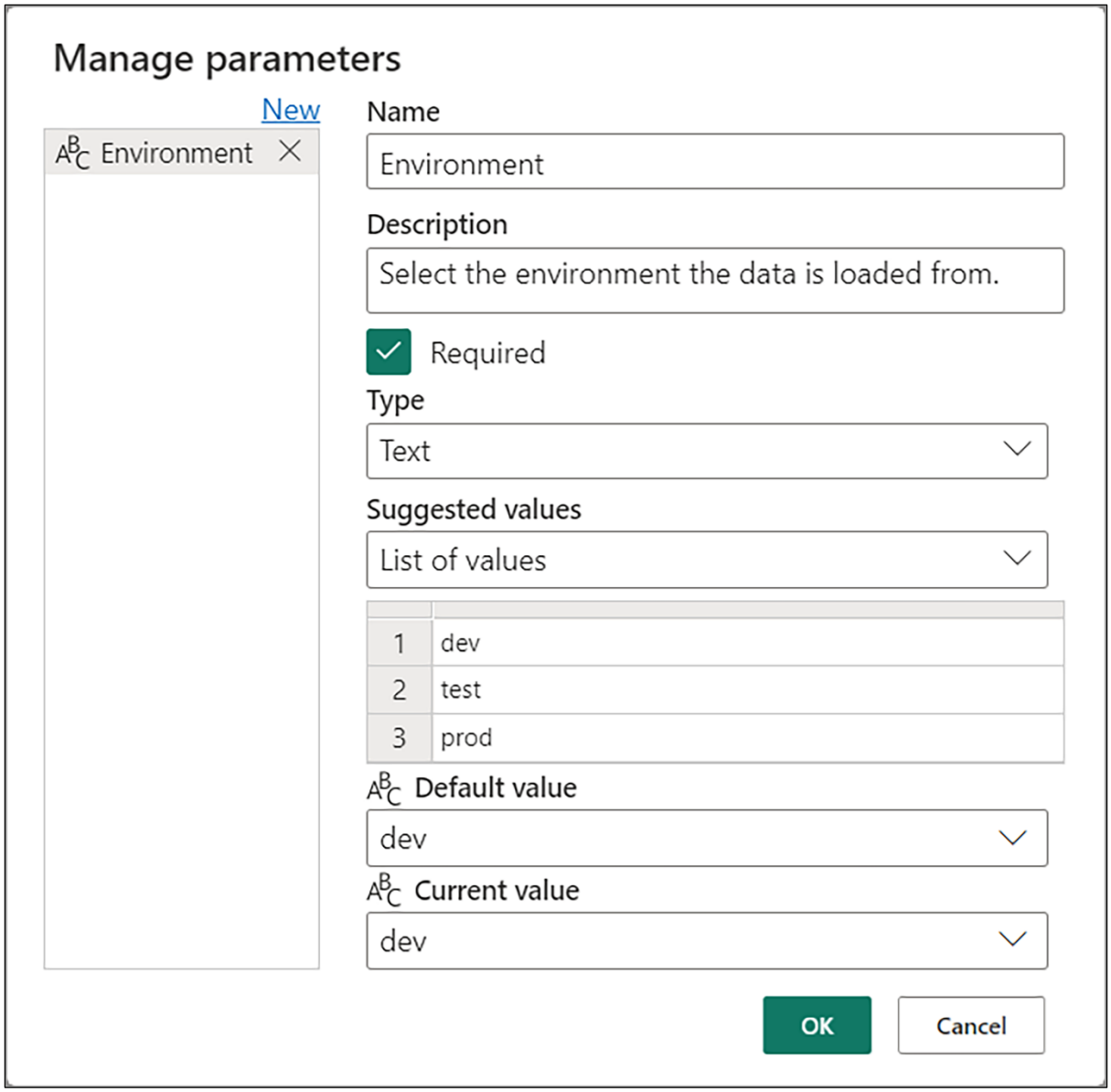
Transactions

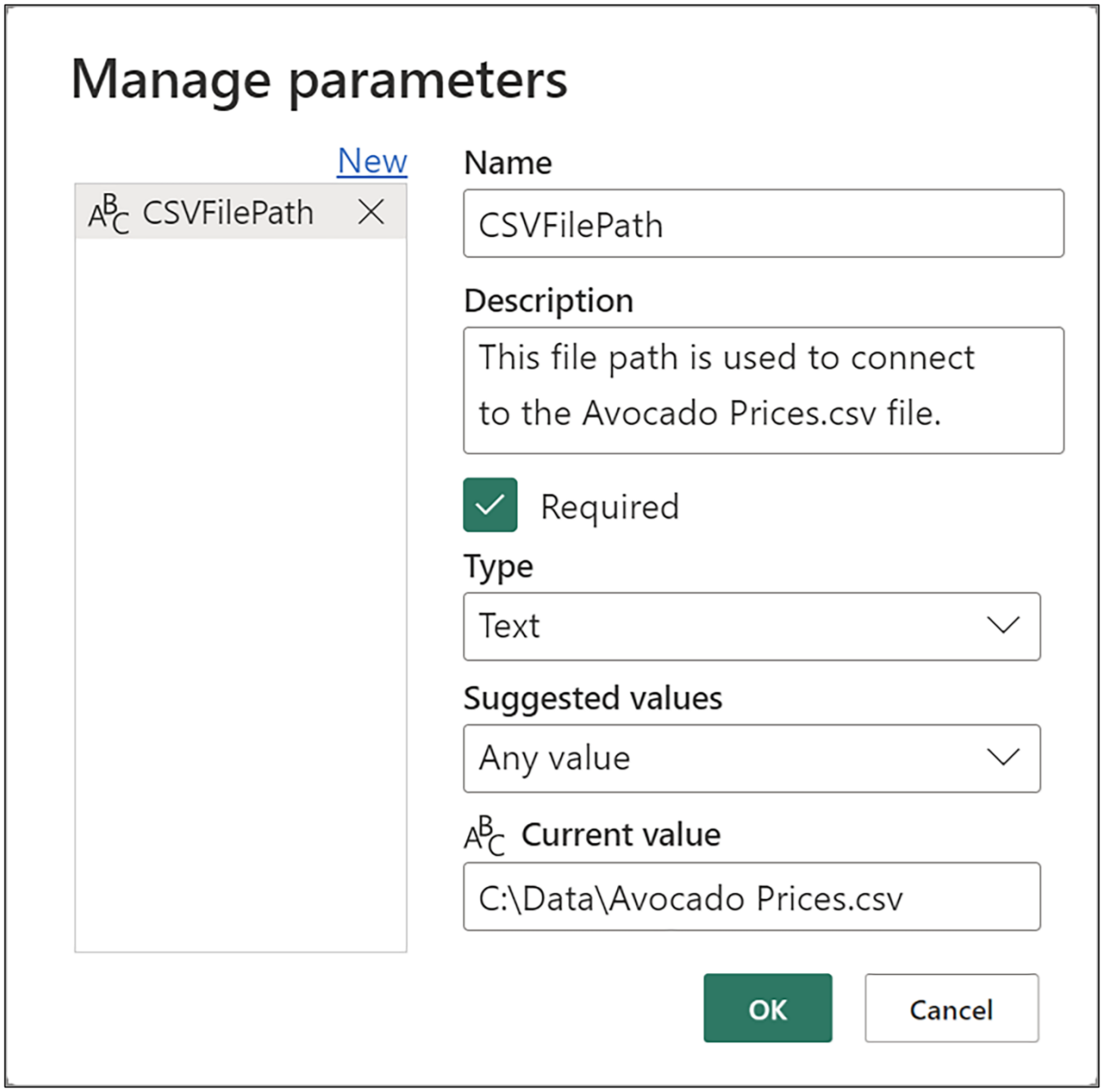
VAT_Percentage (0.25)

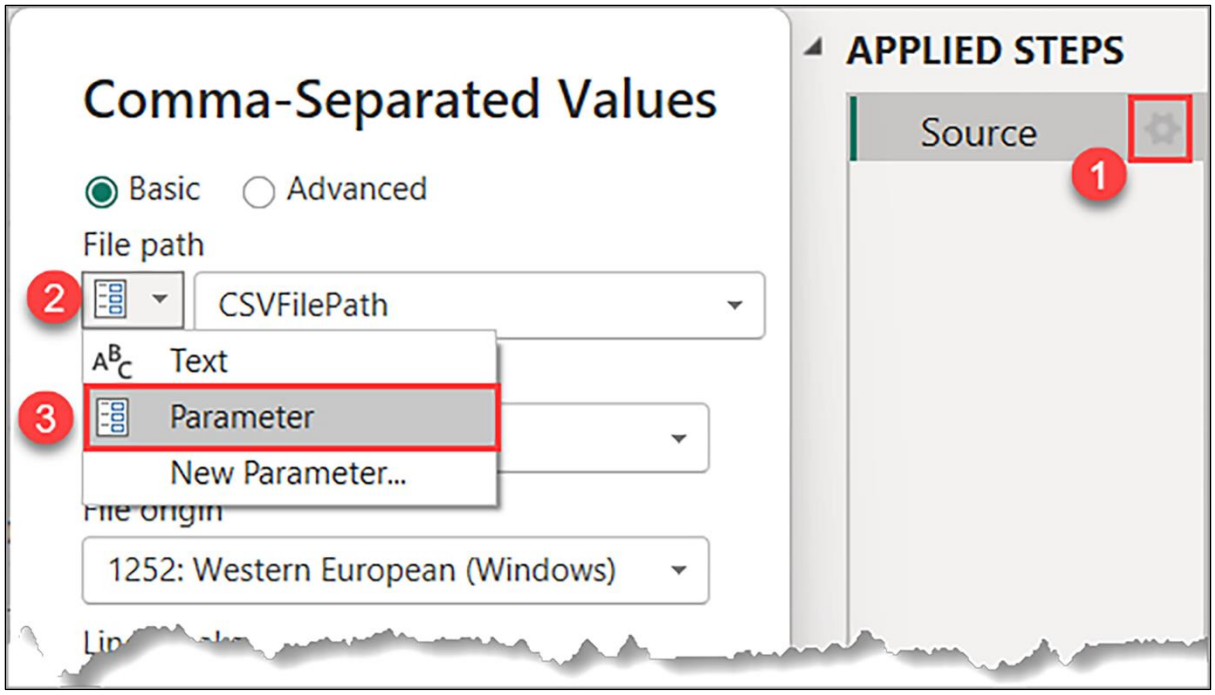
	Invoice-ID	Amount excl VAT
1	INV-12345	*150.00*
2	INV-67890	*250.00*
3	INV-23456	*75.00*
4	INV-78901	*500.00*
5	INV-34567	*200.00*
6	INV-89012	*350.00*




Custom column
Add a column that is computed from other columns or values.
New column name *
Amount incl VAT
Data type
Currency
Custom column formula *
= [Amount excl VAT] * (1 + VAT_Percentage)
References the parameter
Available column(s)
Invoice-ID
Amount excl VAT
Insert column
OK
Cancel

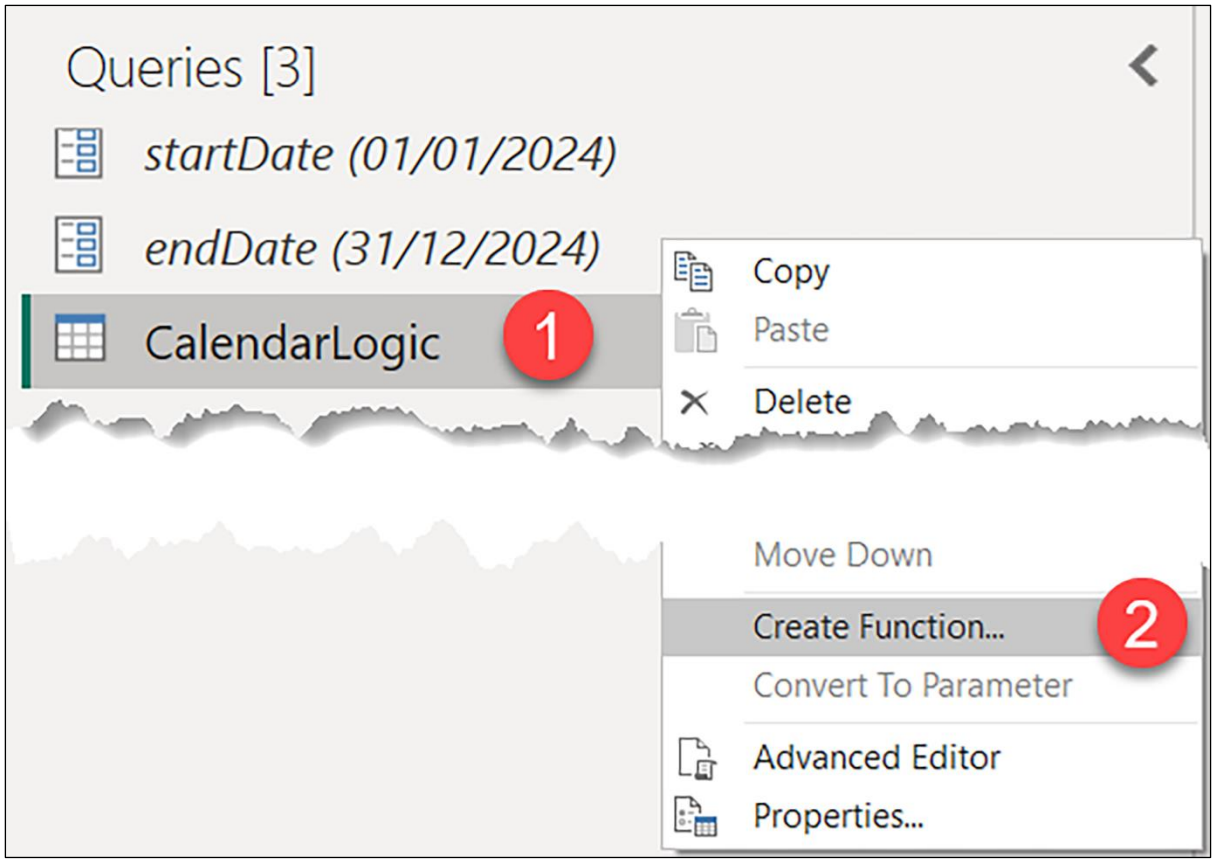

Manage parameters
New
Environment
Name
Environment
Description
Select the environment the data is loaded from.
Required
Type
Text
Suggested values
List of values
1 dev
2 test
3 prod
Default value
dev
Current value
dev
OK
Cancel

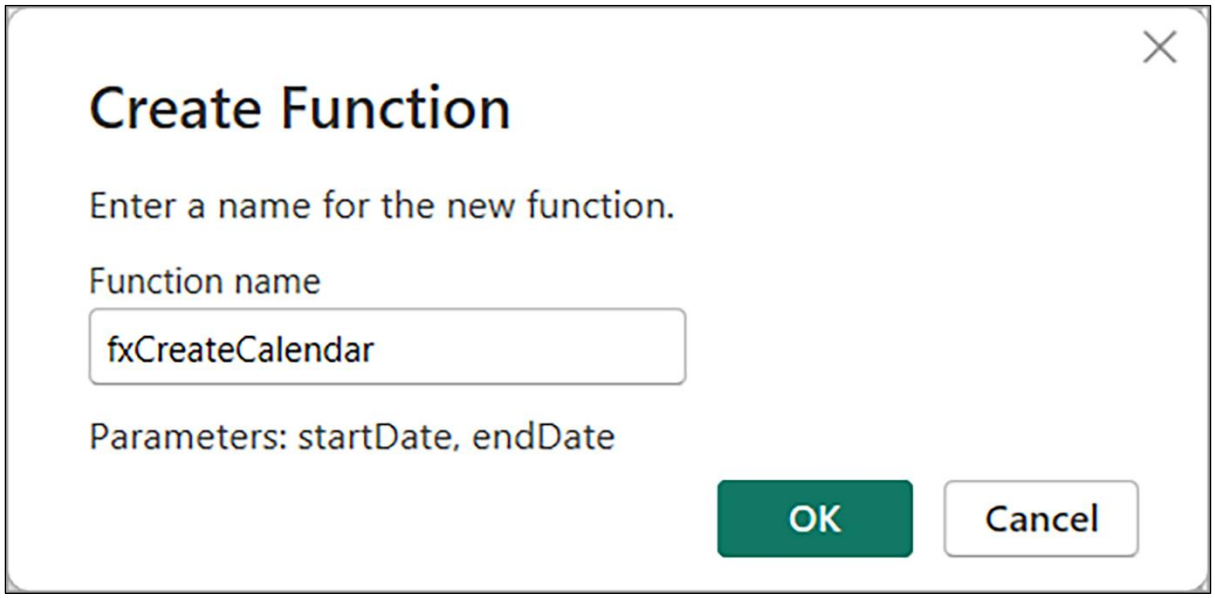

Manage parameters
New
CSVFilePath
Name
CSVFilePath
Description
This file path is used to connect to the Avocado Prices.csv file.
Required
Type
Text
Suggested values
Any value
Current value
C:\Data\Avocado Prices.csv
OK
Cancel

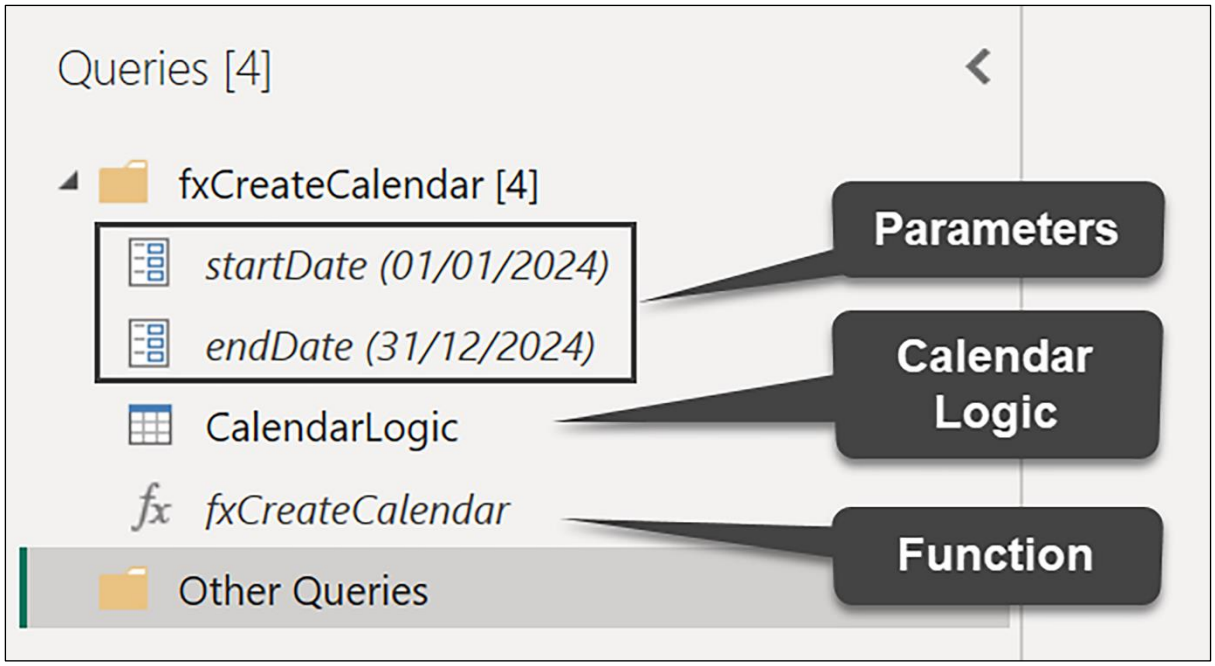


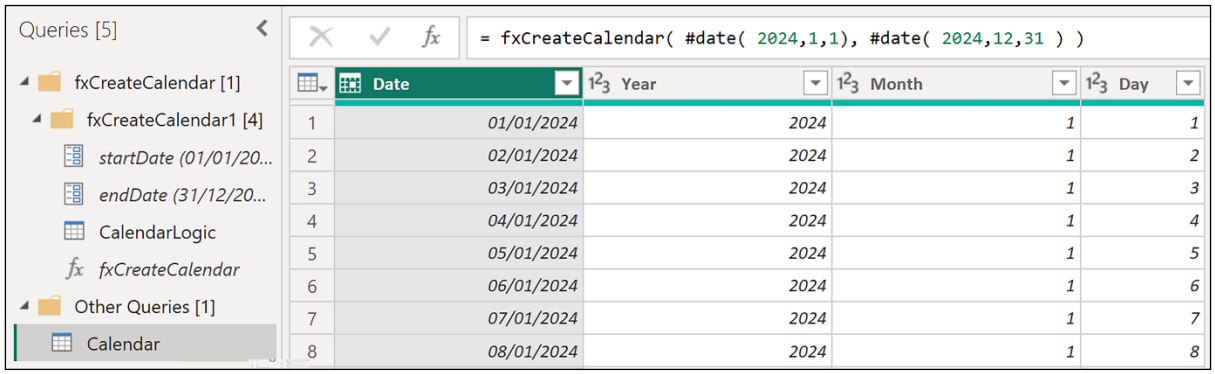

```
let
  Source = List.Dates(
             #date(2023,1,1),
             Duration.Days( #date(2023,10,1) - #date(2023,1,1) ) + 1,
             #duration( 1, 0, 0, 0 )
           ),
  ToTable = Table.FromList(Source, Splitter.SplitByNothing(), type table [Date = date], null, 1),
  AddYear = Table.AddColumn(ToTable, "Year", each Date.Year([Date]), Int64.Type),
  AddMonth = Table.AddColumn(AddYear, "Month", each Date.Month([Date]), Int64.Type)
in
  AddMonth
```

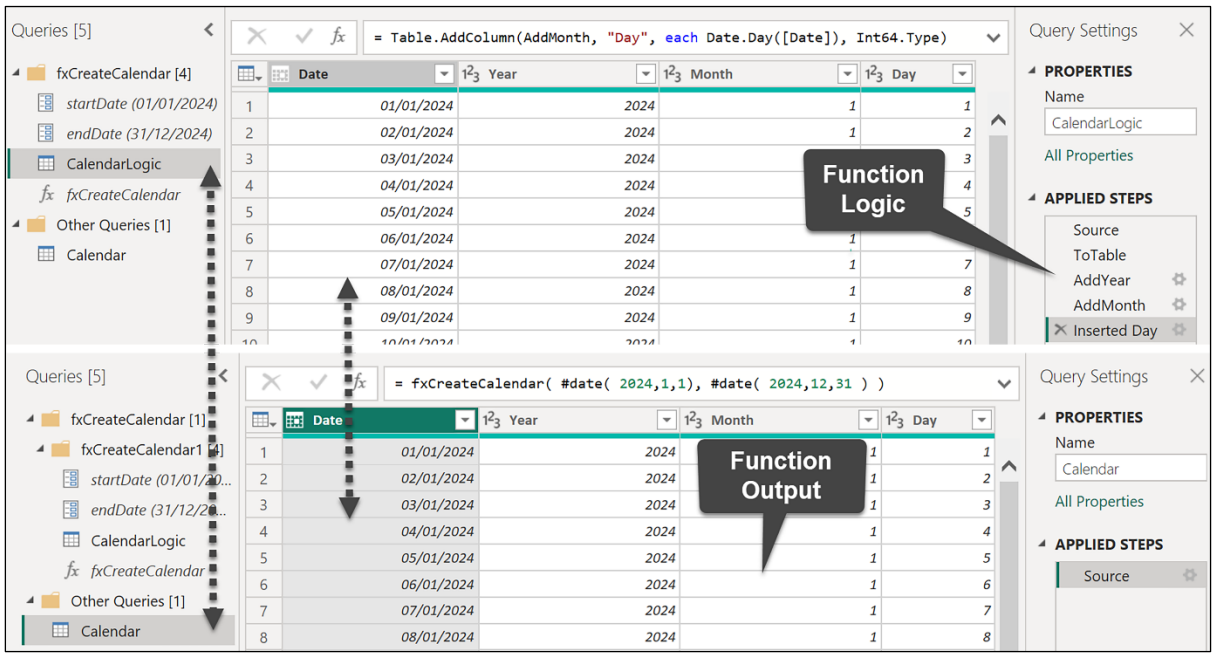
```
let
  Source = List.Dates(
             startDate,
             Duration.Days( endDate - startDate ) + 1,
             #duration( 1, 0, 0, 0 )
           ),
  ToTable = Table.FromList(Source, Splitter.SplitByNothing(), type table [Date = date], null, 1),
  AddYear = Table.AddColumn(ToTable, "Year", each Date.Year([Date]), Int64.Type),
  AddMonth = Table.AddColumn(AddYear, "Month", each Date.Month([Date]), Int64.Type)
in
  AddMonth
```

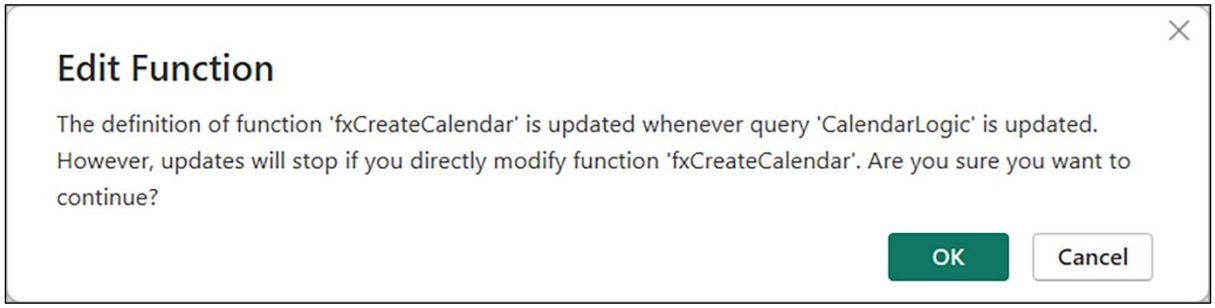

Queries [3]
startDate (01/01/2024)
endDate (31/12/2024)
CalendarLogic
1
Copy
Paste
Delete
Move Down
Create Function...
2
Convert To Parameter
Advanced Editor
Properties...

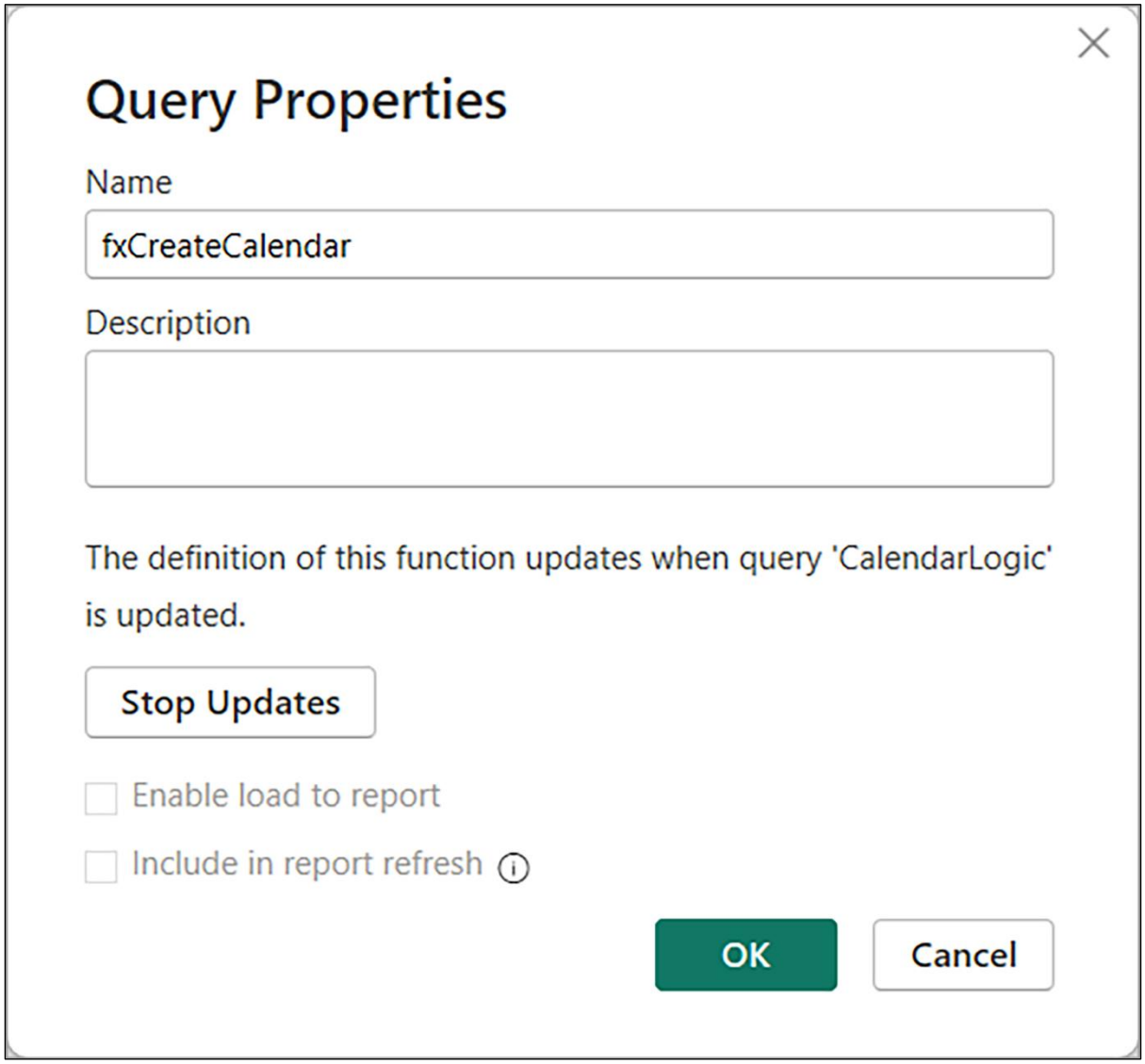

Create Function
Enter a name for the new function.
Function name
fxCreateCalendar
Parameters: startDate, endDate
OK
Cancel

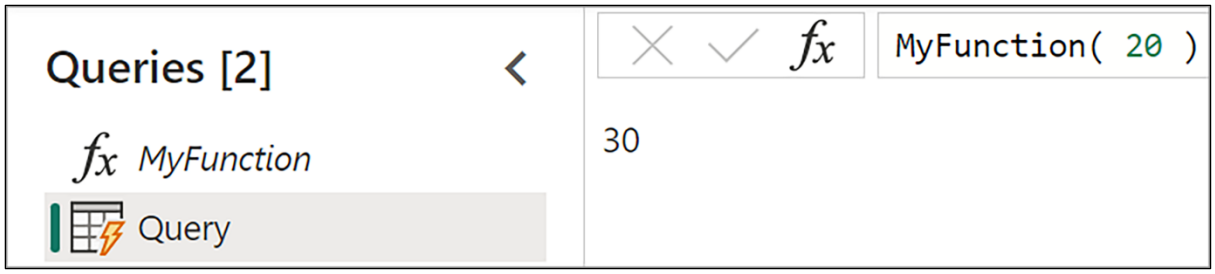

Queries [4]
fxCreateCalendar [4]
startDate (01/01/2024)
endDate (31/12/2024)
CalendarLogic
fxCreateCalendar
Other Queries
Parameters
Calendar Logic
Function

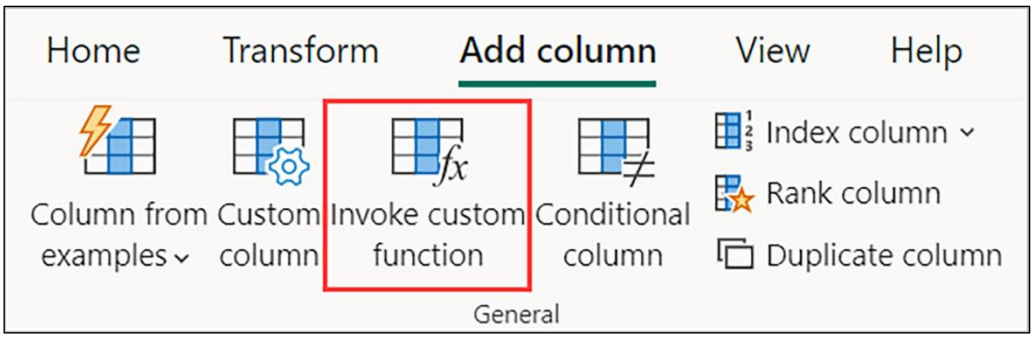

Queries [5]
fxCreateCalendar [1]
fxCreateCalendar1 [4]
startDate (01/01/20...
endDate (31/12/20...
CalendarLogic
fxCreateCalendar
Other Queries [1]
Calendar
= fxCreateCalendar(#date(2024,1,1), #date(2024,12,31))
Date
Year
Month
Day
1 01/01/2024 2024 1 1
2 02/01/2024 2024 1 2
3 03/01/2024 2024 1 3
4 04/01/2024 2024 1 4
5 05/01/2024 2024 1 5
6 06/01/2024 2024 1 6
7 07/01/2024 2024 1 7
8 08/01/2024 2024 1 8

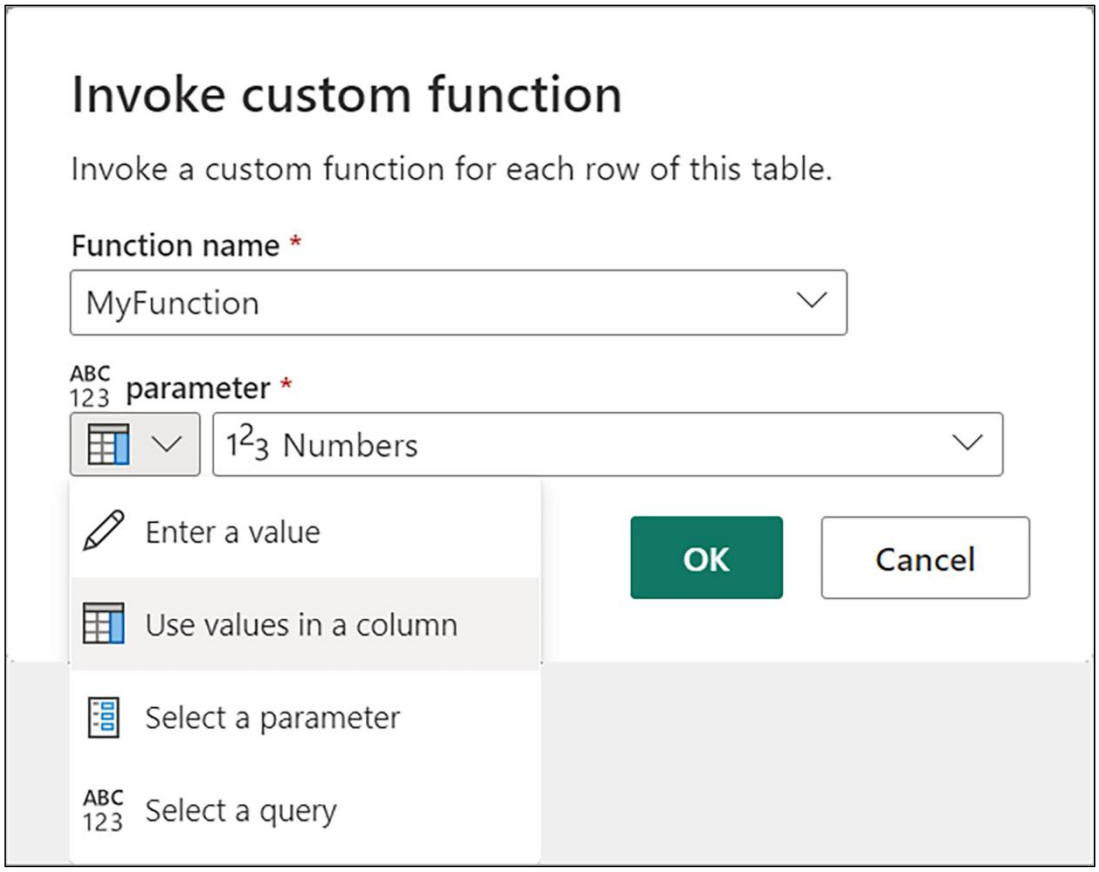

Queries [5]
fxCreateCalendar [4]
startDate (01/01/2024)
endDate (31/12/2024)
CalendarLogic
fxCreateCalendar
Other Queries [1]
Calendar
= Table.AddColumn(AddMonth, "Day", each Date.Day([Date]), Int64.Type)
Query Settings
PROPERTIES
Name
CalendarLogic
All Properties
APPLIED STEPS
Source
ToTable
AddYear
AddMonth
Inserted Day
Function Logic
Queries [5]
fxCreateCalendar [1]
fxCreateCalendar1 [4]
startDate (01/01/20...
endDate (31/12/20...
CalendarLogic
fxCreateCalendar
Other Queries [1]
Calendar
= fxCreateCalendar(#date(2024,1,1), #date(2024,12,31))
Query Settings
PROPERTIES
Name
Calendar
All Properties
APPLIED STEPS
Source
Function Output

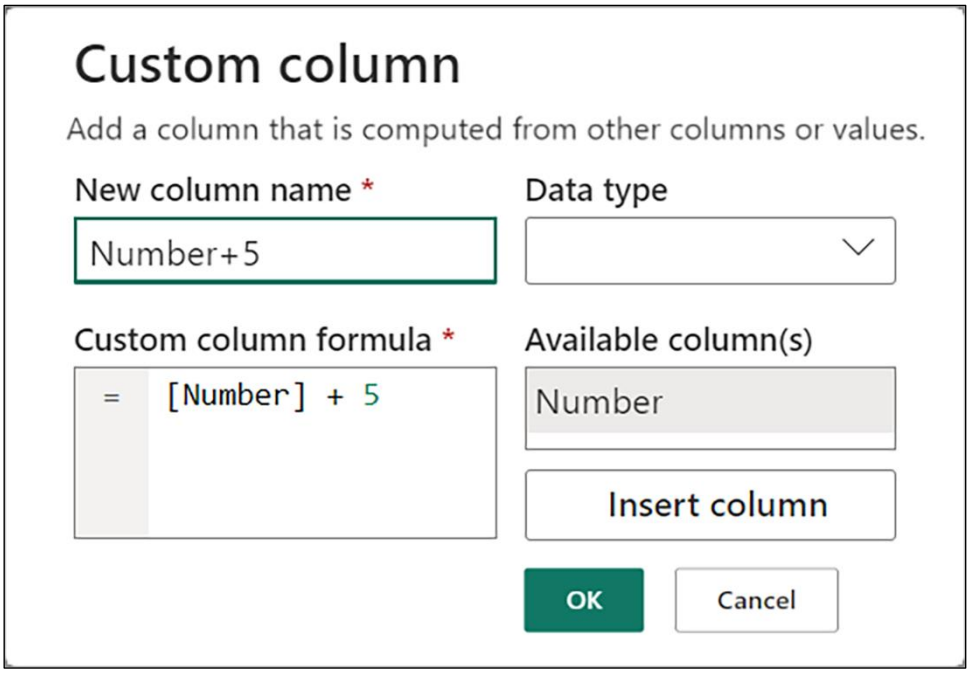

Edit Function
The definition of function 'fxCreateCalendar' is updated whenever query 'CalendarLogic' is updated. However, updates will stop if you directly modify function 'fxCreateCalendar'. Are you sure you want to continue?
OK
Cancel


Query Properties
Name
fxCreateCalendar
Description
The definition of this function updates when query 'CalendarLogic' is updated.
Stop Updates
Enable load to report
Include in report refresh
OK
Cancel

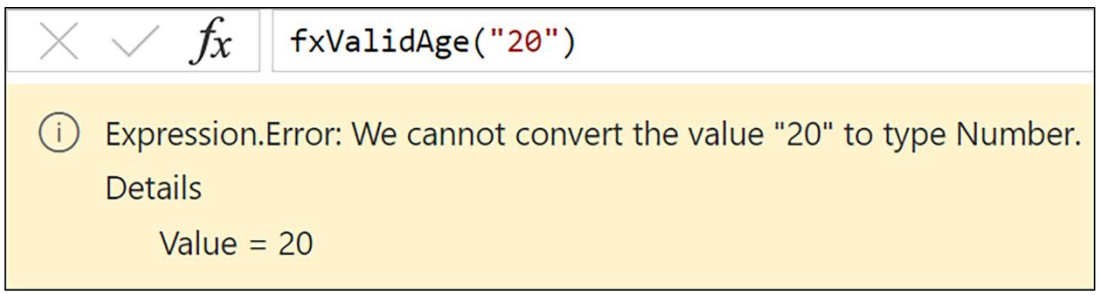

Queries [2]
MyFunction
Query
MyFunction(20)
30

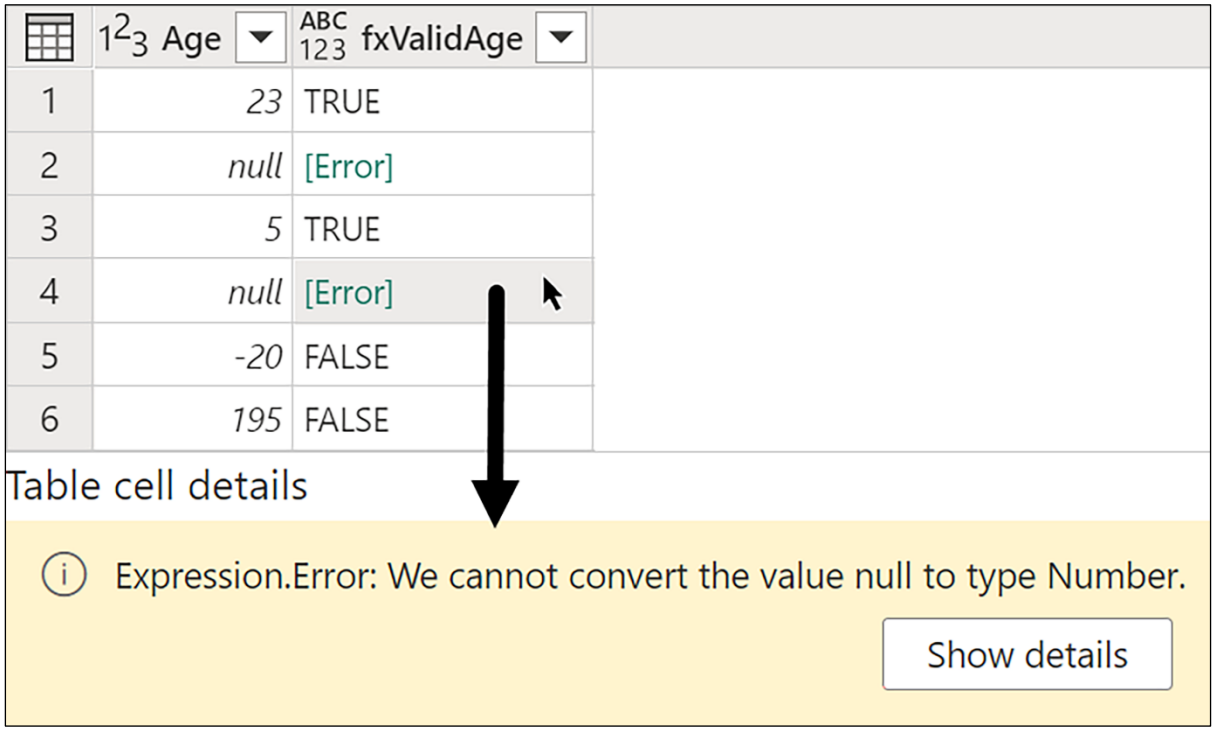

Home
Transform
Add column
View
Help
Column from examples
Custom column
Invoke custom function
Conditional column
Index column
Rank column
Duplicate column
General


Invoke custom function
Invoke a custom function for each row of this table.
Function name *
MyFunction
parameter *
Numbers
Enter a value
Use values in a column
Select a parameter
Select a query
OK
Cancel



	1²3 Age	ABC 123 fxValidAge
1	23	TRUE
2	105	TRUE
3	5	TRUE
4	71	TRUE
5	-20	FALSE
6	195	FALSE





	Age	fxValidAge
1	23	TRUE
2	null	[Error]
3	5	TRUE
4	null	[Error]
5	-20	FALSE
6	195	FALSE

Table cell details

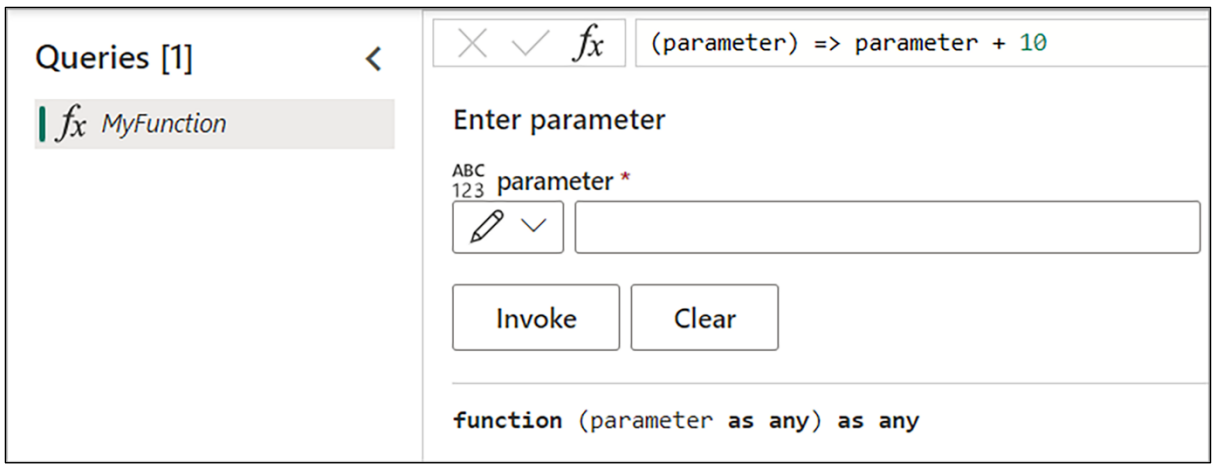
Expression.Error: We cannot convert the value null to type Number.

Show details

	Age	fxValidAge
1	23	TRUE
2	null	null
3	5	TRUE
4	null	null
5	-20	FALSE
6	195	FALSE

```
( numberList as list, minValue as number, maxValue as number ) as text =>
  let
    List        = List.Select ( numberList, each _ >= minValue and _ <= maxValue ),
    ListToText  = List.Transform ( FilteredList, each Text.From ( _ ) ),
    CombineText = Text.Combine ( ListToText, ", " )
  in
    CombineText
```

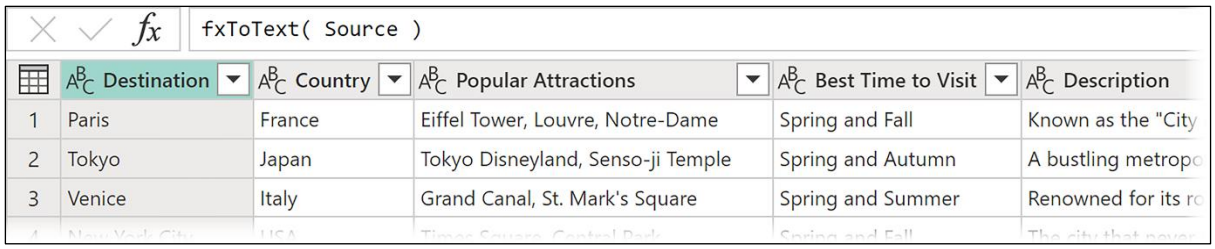
```
// ( numberList as list, minValue as number, maxValue as number ) as text =>
  let
    numberList  = { 1 .. 10 },
    minValue    = 2,
    maxValue    = 8,
    List        = List.Select ( numberList, each _ >= minValue and _ <= maxValue ),
    ListToText  = List.Transform ( List, each Text.From ( _ ) ),
    CombineText = Text.Combine ( ListToText, ", " )
  in
    CombineText
```



Destination	Country	Popular Attractions	Best Time to Visit	Description
Paris	France	Eiffel Tower, Louvre, Notre-Dame	Spring and Fall	Known as the "City of Love," fam
Tokyo	Japan	Tokyo Disneyland, Senso-ji Temple	Spring and Autumn	A bustling metropolis with a rich
Venice	Italy	Grand Canal, St. Mark's Square	Spring and Summer	Renowned for its romantic canals
New York City	USA	Times Square, Central Park	Spring and Fall	The city that never sleeps, offerin
Sydney	Australia	Sydney Opera House, Bondi Beach	Spring and Summer	Home to iconic landmarks like th
Santorini	Greece	Oia, Fira	Summer	A stunning island in the Aegean
Cairo	Egypt	Pyramids of Giza, Egyptian Museum	Fall and Winter	Explore ancient history, includin
Rio de Janeiro	Brazil	Christ the Redeemer, Copacabana Beach	Summer	Famous for its Carnival, Christ the

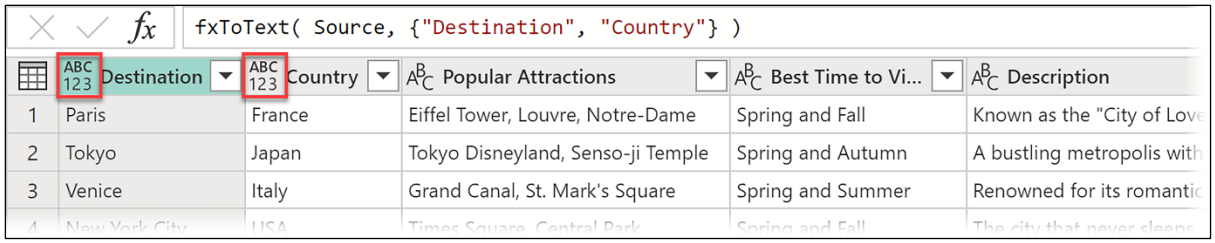
```
let
  Source = MyTable,
  Columns = Table.ColumnNames( Source ),
  TransformTypes = List.Transform( Columns, each { _, type text } ),
  TypeToText =  Table.TransformColumnTypes( Source, TransformTypes )
in
  TypeToText
```

```
( InputTable as table ) as table =>
let
  Source = InputTable,
  Columns = Table.ColumnNames( Source ),
  TransformTypes = List.Transform( Columns, each { _, type text } ),
  TypeToText =  Table.TransformColumnTypes( Source, TransformTypes )
in
  TypeToText
```



	Destination	Country	Popular Attractions	Best Time to Visit	Description
1	Paris	France	Eiffel Tower, Louvre, Notre-Dame	Spring and Fall	Known as the "City
2	Tokyo	Japan	Tokyo Disneyland, Senso-ji Temple	Spring and Autumn	A bustling metropo
3	Venice	Italy	Grand Canal, St. Mark's Square	Spring and Summer	Renowned for its ro

```
( InputTable as table, optional Exclusions as list ) as table =>
let
  Source = InputTable,
  AllColumns = Table.ColumnNames( Source ),
  Exclusions = Exclusions ?? {} ,
  RelevantColumns = List.RemoveItems( AllColumns, Exclusions ),
  TransformTypes = List.Transform( RelevantColumns, each { _, type text } ),
  TypeToText =  Table.TransformColumnTypes( Source, TransformTypes )
in
  TypeToText
```



	Destination	Country	Popular Attractions	Best Time to Vi...	Description
1	Paris	France	Eiffel Tower, Louvre, Notre-Dame	Spring and Fall	Known as the "City of Love
2	Tokyo	Japan	Tokyo Disneyland, Senso-ji Temple	Spring and Autumn	A bustling metropolis with
3	Venice	Italy	Grand Canal, St. Mark's Square	Spring and Summer	Renowned for its romantic

Table: Transactions

	Id	Date	Amount
1	SI-1401	*2/7/2023*	*237*
2	SI-1402	*11/3/2023*	*489*
3	SI-1403	*9/22/2023*	*712*
4	SI-1404	*5/9/2023*	*56*
5	SI-1405	*12/29/2023*	*901*

Table: Contract

	ContractID	Start	End
1	VI-2023-A1	*1/1/2023*	*3/31/2023*
2	VI-2023-B2	*4/1/2023*	*6/30/2023*
3	VI-2023-C3	*7/1/2023*	*8/31/2023*
4	VI-2023-D4	*9/1/2023*	*12/31/2023*

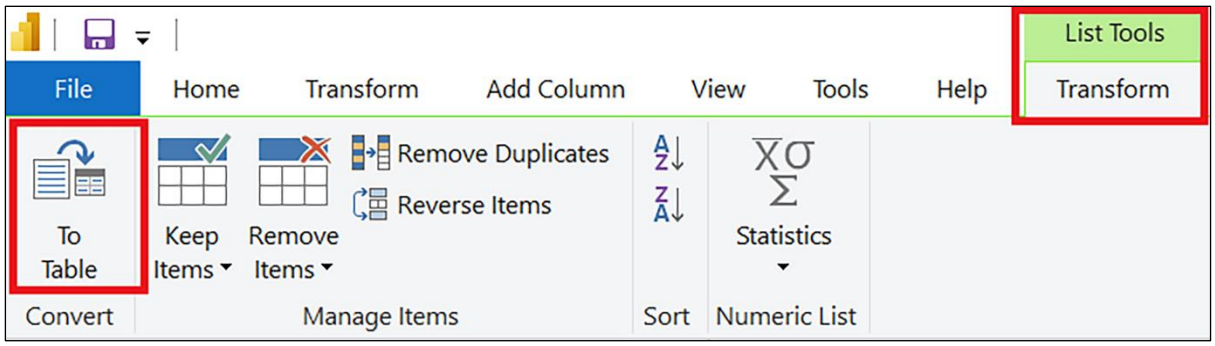
	Id	Date	Amount	Contracts
1	SI-1401	*2/7/2023*	*237*	[Table]
2	SI-1402	*11/3/2023*	*489*	[Table]
3	SI-1403	*9/22/2023*	*712*	[Table]
4	SI-1404	*5/9/2023*	*56*	[Table]
5	SI-1405	*12/29/2023*	*901*	[Table]

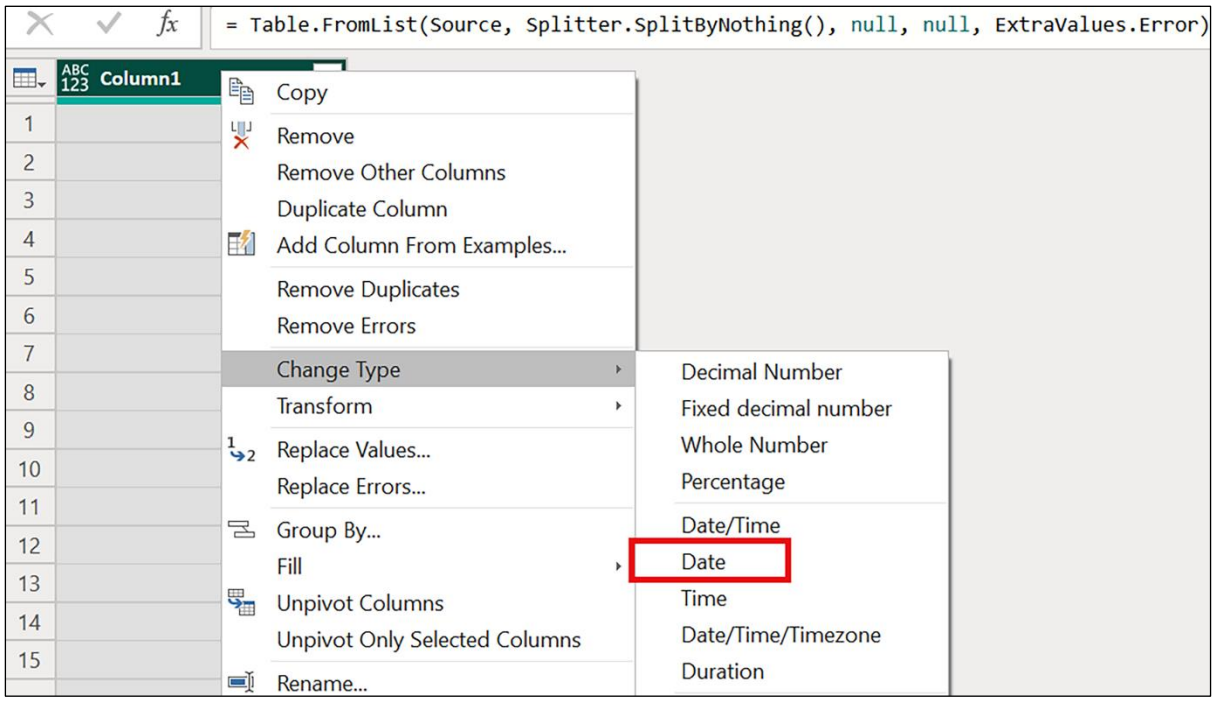
```
fxDateRangeJoin( Source, "Date", Contract, "Start", "End", "Contracts" )
```

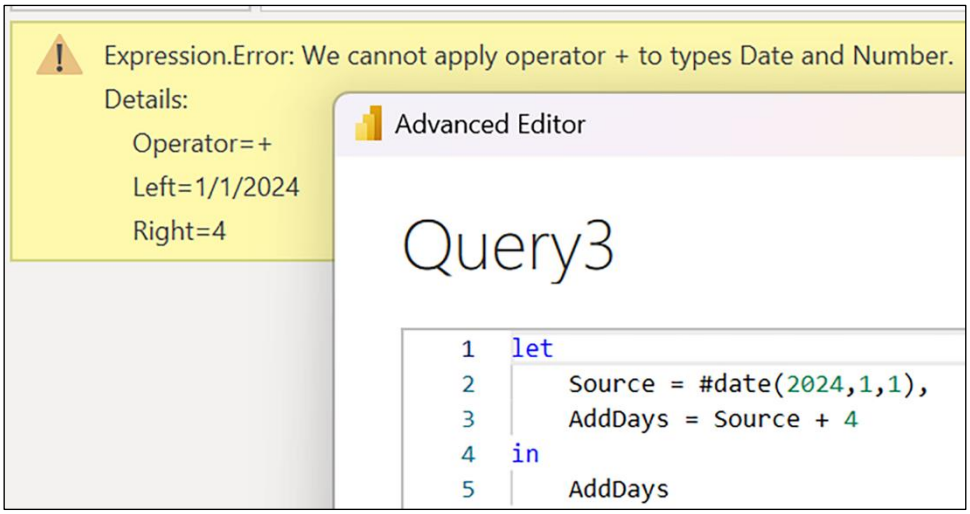
	Id	Date	Amount	Contracts
1	SI-1401	*2/7/2023*	*237*	[Table]
2	SI-1402	*11/3/2023*	*489*	[Table]
3	SI-1403	*9/22/2023*	*712*	[Table]
4	SI-1404	*5/9/2023*	*56*	[Table]
5	SI-1405	*12/29/2023*	*901*	[Table]

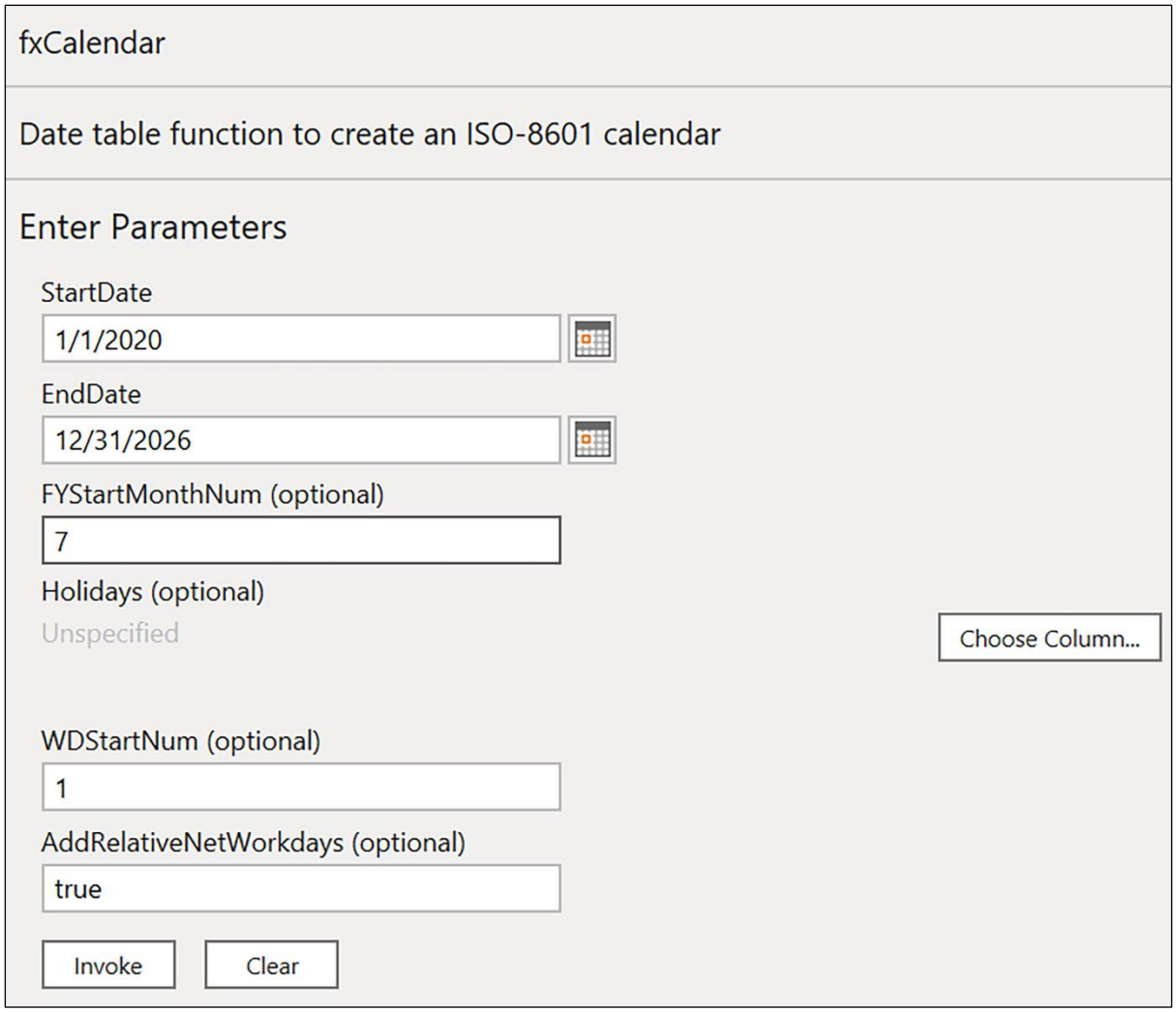
	Id	Date	Amount	ContractID	Start	End
1	SI-1401	*2/7/2023*	*237*	SI-1401	*1/1/2023*	*3/31/2023*
2	SI-1402	*11/3/2023*	*489*	SI-1404	*9/1/2023*	*12/31/2023*
3	SI-1403	*9/22/2023*	*712*	SI-1404	*9/1/2023*	*12/31/2023*
4	SI-1404	*5/9/2023*	*56*	SI-1402	*4/1/2023*	*6/30/2023*
5	SI-1405	*12/29/2023*	*901*	SI-1404	*9/1/2023*	*12/31/2023*

Chapter 10: Dealing with Dates, Times, and Durations



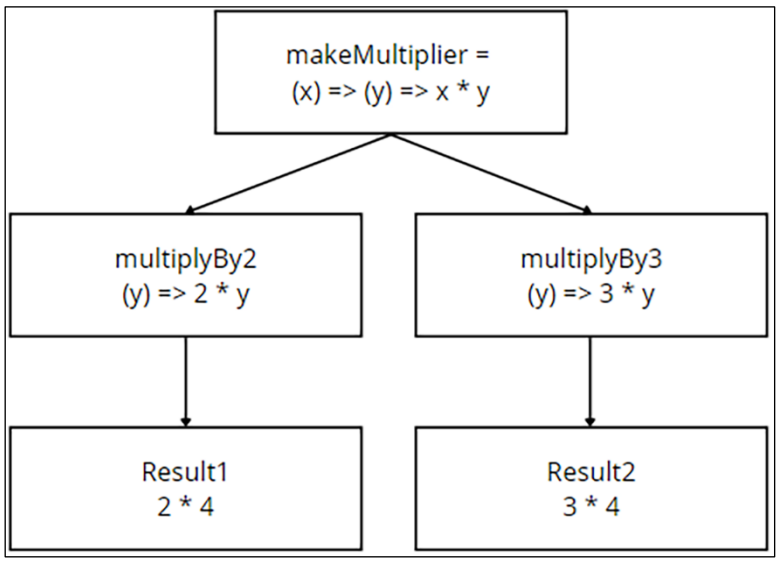





fxCalendar
Date table function to create an ISO-8601 calendar
Enter Parameters
StartDate
1/1/2020
EndDate
12/31/2026
FYStartMonthNum (optional)
7
Holidays (optional)
Unspecified
Choose Column...
WDStartNum (optional)
1
AddRelativeNetWorkdays (optional)
true
Invoke
Clear

	Column1	Multiplication	Time
1	0	0	12:00:00 AM
2	1	0.041666667	1:00:00 AM
3	2	0.083333333	2:00:00 AM
4	3	0.125	3:00:00 AM
5	4	0.166666667	4:00:00 AM
6	5	0.208333333	5:00:00 AM
7	6	0.25	6:00:00 AM
8	7	0.291666667	7:00:00 AM
9	8	0.333333333	8:00:00 AM
10	9	0.375	9:00:00 AM
11	10	0.416666667	10:00:00 AM
12	11	0.458333333	11:00:00 AM
13	12	0.5	12:00:00 PM
14	13	0.541666667	1:00:00 PM
15	14	0.583333333	2:00:00 PM
16	15	0.625	3:00:00 PM
17	16	0.666666667	4:00:00 PM
18	17	0.708333333	5:00:00 PM
19	18	0.75	6:00:00 PM
20	19	0.791666667	7:00:00 PM
21	20	0.833333333	8:00:00 PM
22	21	0.875	9:00:00 PM
23	22	0.916666667	10:00:00 PM
24	23	0.958333333	11:00:00 PM
25	24	1	Error

Chapter 11: Comparers, Replacers, Combiners, and Splitters



	Column1	Column2	equivalenceCheck
1	APPLE	apple	*FALSE*
2	apple	apple	*TRUE*
3	42	*42*	*FALSE*
4	*42*	*42*	*TRUE*

	Value1	Unicode code value
1	A	65
2	APPLE	65, 80, 80, 76, 69
3	Apple	65, 112, 112, 108, 101
4	B	66
5	a	97
6	apple	97, 112, 112, 108, 101

	Value1	Value2	equivalenceCheck
1	APPLE	apple	*-1*
2	apple	apple	*0*
3	apple	APPLE	*1*

	Value1	Value2	equivalenceCheck
1	APPLE	apple	*0*
2	apple	Apple	*0*
3	apple	APPLE	*0*

	Value1	Value2	equivalenceCheck
1	APPLE	apple	*1*
2	apple	apple	*0*
3	apple	APPLE	*-1*

DK	1
US	0

	Label	Order
1	High	*0*
2	Moderate	*1*
3	Low	*2*

	List
1	January
2	February
3	March
4	April

	Label
1	High
2	Moderate
3	Low

	Category	SubCategory
1	Category C	SubCategory 9
2	Category C	SubCategory 20
3	Category B	SubCategory 2
4	Category B	SubCategory 5
5	Category A	SubCategory 1
6	Category A	SubCategory 10
7	Category A	SubCategory 11

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

```
= fxCreateWeekdayRecord(#date(2024, 1, 1))
```

Monday	0
Tuesday	1
Wednesday	2
Thursday	3
Friday	4
Saturday	5
Sunday	6

	Employee name	Department
1	John	Sales
2	Lisa	Marketing
3	David	Finance
4	Sarah	HR
5	Alex	Operations

	Employee name	Department
1	John	Sales
2	Lisa	Marketing
3	David	Finance
4	Sarah	Human Resources
5	Alex	Operations

	curText	oldText	newText	ReplaceText
1	A sudden rattle noise startled birds.	t	f	A sudden raffle noise sfarfled birds.
2	A sudden rattle noise startled birds.	tt	ff	A sudden raffle noise startled birds.
3	The sun shines, shines brightly.	shines	radiates	The sun radiates, radiates brightly.
4	The sun shines, shines brightly.	shines,	radiates	The sun radiates shines brightly.

```
= Table.ReplaceValue( Example, ".", "!", Replacer.ReplaceText, {"curText", "ReplaceText"} )
```

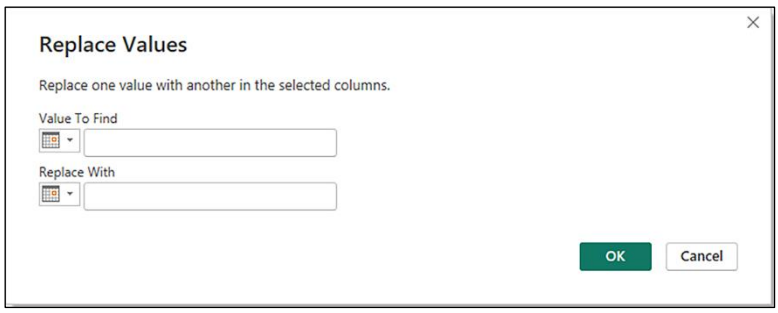
	curText	oldText	newText	ReplaceText
1	A sudden rattle noise startled birds!	t	f	A sudden raffle noise sfarfled birds!
2	A sudden rattle noise startled birds!	tt	ff	A sudden raffle noise startled birds!
3	The sun shines, shines brightly!	shines	radiates	The sun radiates, radiates brightly!
4	The sun shines, shines brightly!	shines,	radiates	The sun radiates shines brightly!

	curVal	oldVal	newVal	ReplaceValue
1	Make some noise.	noise	love	Make some noise.
2	Make some noise.	Make some noise.	You rock!	You rock!
3	*10,1*	*10*	*11*	*10,1*
4	*10*	*10*	*11*	*11*
5	List	List	*1*	List
6	List	List	*1*	*1*

	Emp	StartDate	EndDate	Dept
1	John	*1-1-2020*	*5-11-2023*	Operations
2	Lisa	*15-9-2015*	*10-12-2023*	Sales
3	David	*1-3-2019*	*null*	Finance
4	John	*6-11-2023*	*null*	Sales
5	Lisa	*11-12-2023*	*null*	Marketing

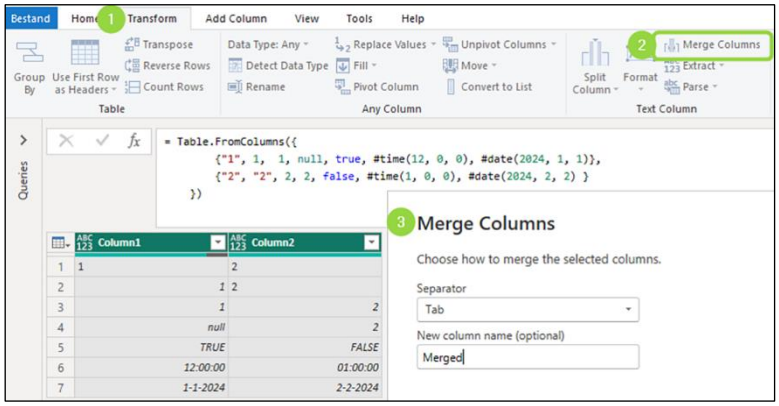
	Date	Employee	Department
1	*1-11-2023*	John	Table
2	*1-12-2023*	John	Table
3	*1-1-2024*	John	Table
4	*1-11-2023*	Lisa	Table
5	*1-12-2023*	Lisa	Table

Emp	StartDate	EndDate	Dept
John	*1-1-2020*	*5-11-2023*	Operations
John	*6-11-2023*	*null*	Sales



	Date	Employee	Department
1	*1-11-2023*	John	Operations
2	*1-12-2023*	John	Sales
3	*1-1-2024*	John	Sales
4	*1-11-2023*	Lisa	Sales
5	*1-12-2023*	Lisa	Sales
6	*1-1-2024*	Lisa	Marketing
7	*1-11-2023*	David	Finance
8	*1-12-2023*	David	Finance
9	*1-1-2024*	David	Finance

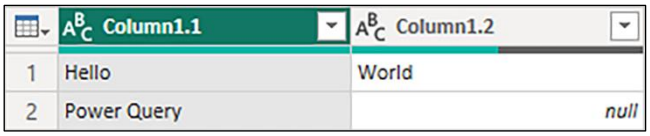
	Label	Date Column1	Date Column2	Date Column3	Date Column4
1	Entry 1	1-11-2023	9-12-2023	21-1-2024	1-1-0001
2	Entry 2	9-9-2023	1-1-1900	1-1-0001	9-3-2024
3	Entry 3	2-2-2024	11-12-2023	9-9-9999	3-9-2023
4	Entry 4	25-5-2021	1-1-0001	1-1-1900	11-1-2024
5	Entry 5	9-9-9999	5-1-2024	9-10-2023	7-3-2024



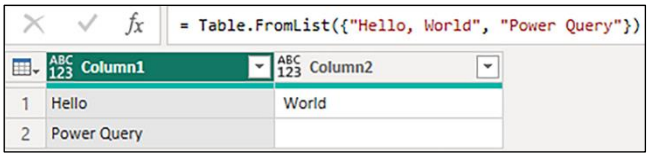
	Merged
1	1 2
2	1 2
3	1 2
4	2
5	truefalse
6	12:00 PM 1:00 AM
7	1/1/2024 2/2/2024

Texts	**98**	**DELTA**	**19990110**	**NO**	**¯_(ツ)_/¯**	Wait	What
Positions	0	4	6	10	12	12	
Chars to extact	**4**	**2**	**4**	**2**	**0**		
explanation	*=4-0*	*=6-4*	*=10-6*	*=12-10*	*=12-12*		
yield	*98***	*DE*	*1999*	*NO*		Wait	

	Column1
1	Hello, World
2	Power Query

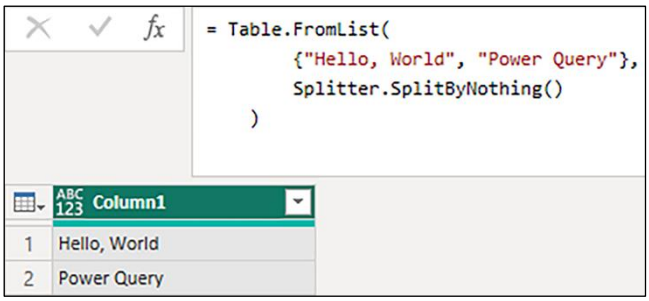


	Column1.1	Column1.2
1	Hello	World
2	Power Query	*null*



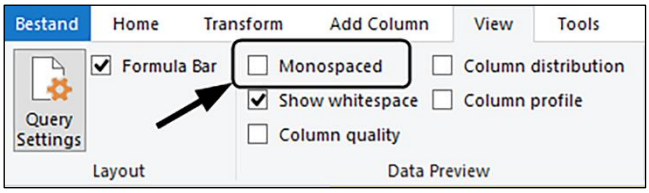
```
= Table.FromList({"Hello, World", "Power Query"})
```

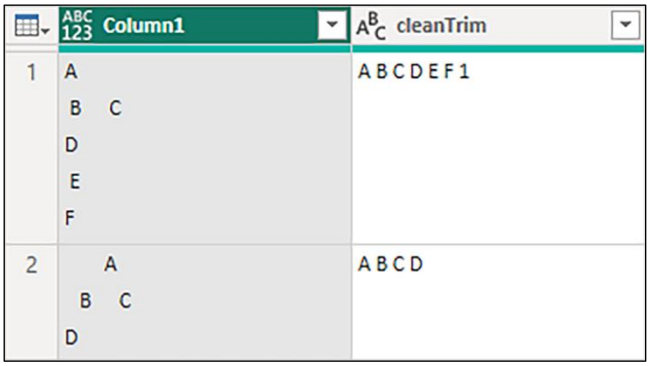
	Column1	Column2
1	Hello	World
2	Power Query	



```
= Table.FromList(
        {"Hello, World", "Power Query"},
        Splitter.SplitByNothing()
    )
```

	Column1
1	Hello, World
2	Power Query

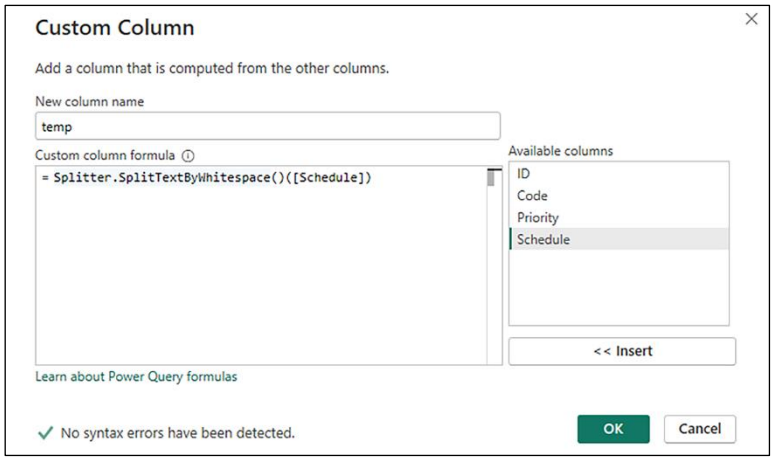


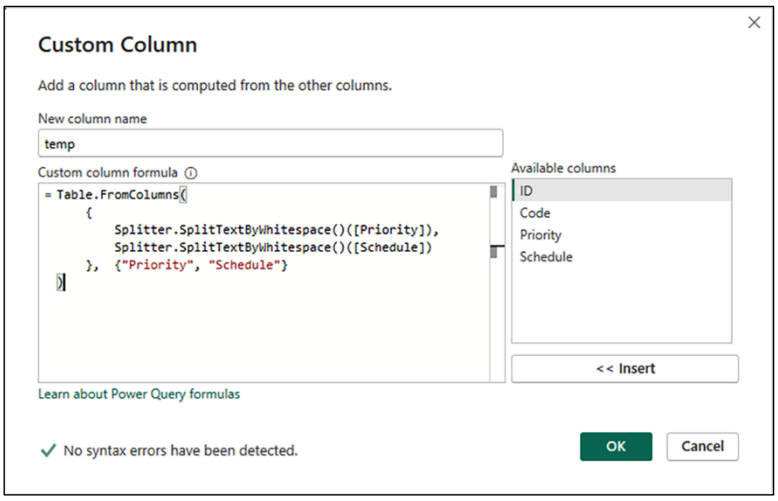


	Column1	cleanTrim
1	A B C D E F	A B C D E F 1
2	A B C D	A B C D

	Contact info	getEmail
1	Wendy Darling, UK, wendy.darling@nevergrowup.co.uk	wendy.darling@nevergrowup.co.uk
2	john.darling@nevergrowup.co.uk, UK;john.boy@abc.com	john.darling@nevergrowup.co.uk, john.boy@abc.com
3	michael; michael.darling@nevergrowup	
4	Peter Pan peter.pan@neverland.com	peter.pan@neverland.com
5	tink@pixie@fairy.org Tinker Bell	
6	Captain Hook: captain.hook@piratescove.org	captain.hook@piratescove.org

	ID	Code	Priority	Schedule
1	A	W01	1	24x7
			2	24x7
			3	24x7
			4	"8x5 (9:00 - 17:00)"
2	B	W01	1	"9x5 (9:00 - 18:00)"
			2	
			3	"9x5 (9:00 - 18:00)"
			4	
				"9x5 (9:00 - 18:00)"
3	C	W01	1	" "
			2	" "
			3	"9x5 (9:00 - 18:00)"
			4	"9x5 (9:00 - 18:00)"





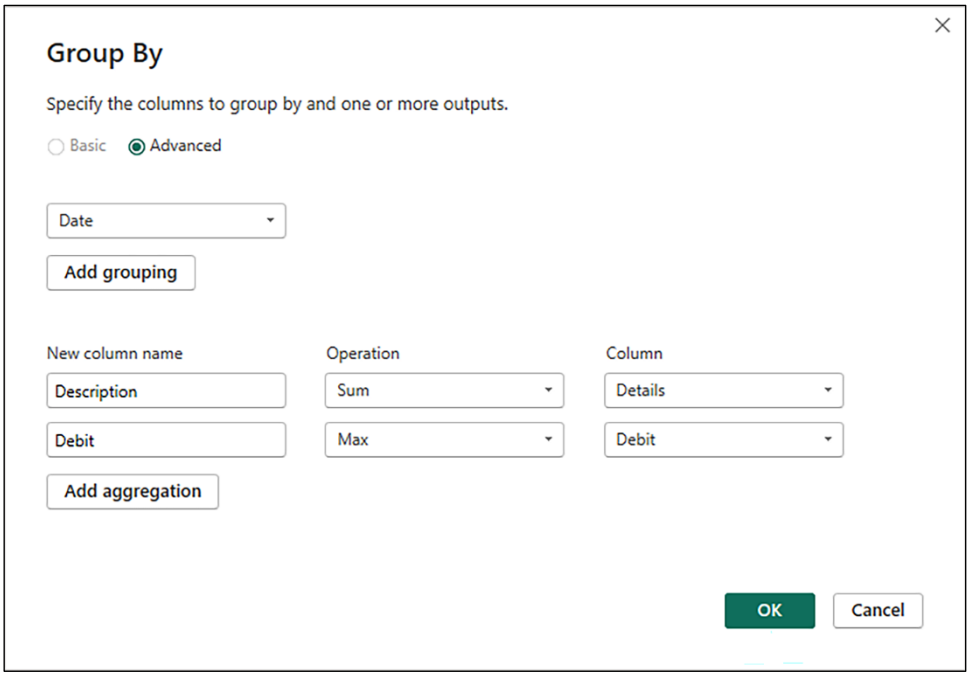
	ID	Code	Priority	Schedule
1	A	W01	1	24x7
2	A	W01	2	24x7
3	A	W01	3	24x7
4	A	W01	4	8x5 (9:00 - 17:00)
5	B	W01	1	9x5 (9:00 - 18:00)
6	B	W01	2	9x5 (9:00 - 18:00)
7	B	W01	3	9x5 (9:00 - 18:00)
8	B	W01	4	9x5 (9:00 - 18:00)
9	C	W01	1	
10	C	W01	2	
11	C	W01	3	9x5 (9:00 - 18:00)
12	C	W01	4	9x5 (9:00 - 18:00)

	ID	Description	Description2
1	1	Prdct A, 5pcs	Prdct A, 5pcs
2	2	Product B, Qty: 10	Product B, Qty: 10
3	3	Prdct C; Quantity: 2	Prdct C; Quantity: 2
4	1	Prdct A, 5pcs	Prdct A, 5pcs
5	2	Product B, Qty: 10	Product B, Qty: 10

	Old	New
1	Prdct	Product
2	pcs	pieces
3	Qty	Quantity

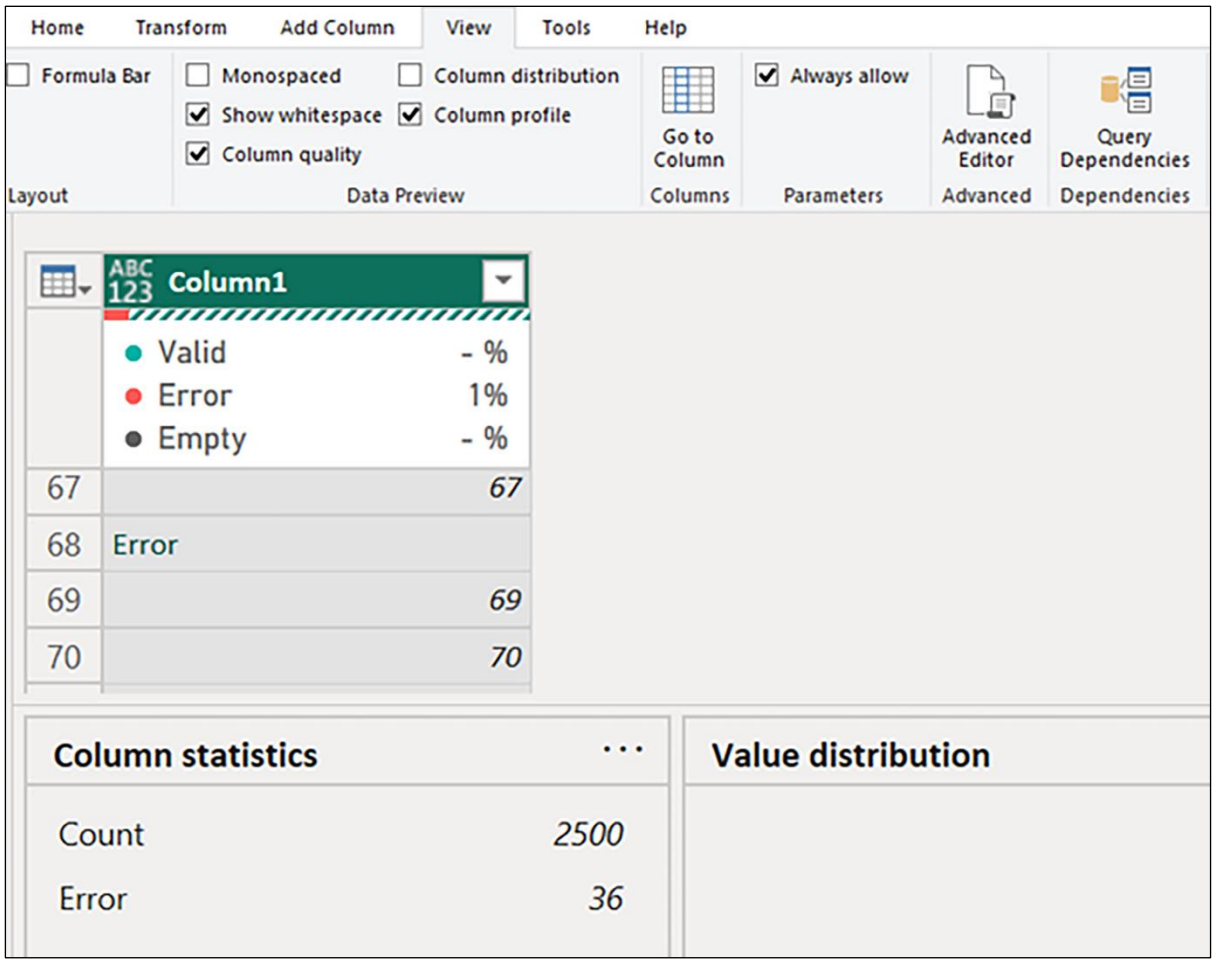
	ID	Description	Description2
1	1	Product A, 5 pieces	Product A, 5 pieces
2	2	Product B, Quantity: 10	Product B, Quantity: 10
3	3	Product C; Quantity: 2	Product C; Quantity: 2
4	1	Product A, 5 pieces	Product A, 5 pieces
5	2	Product B, Quantity: 10	Product B, Quantity: 10

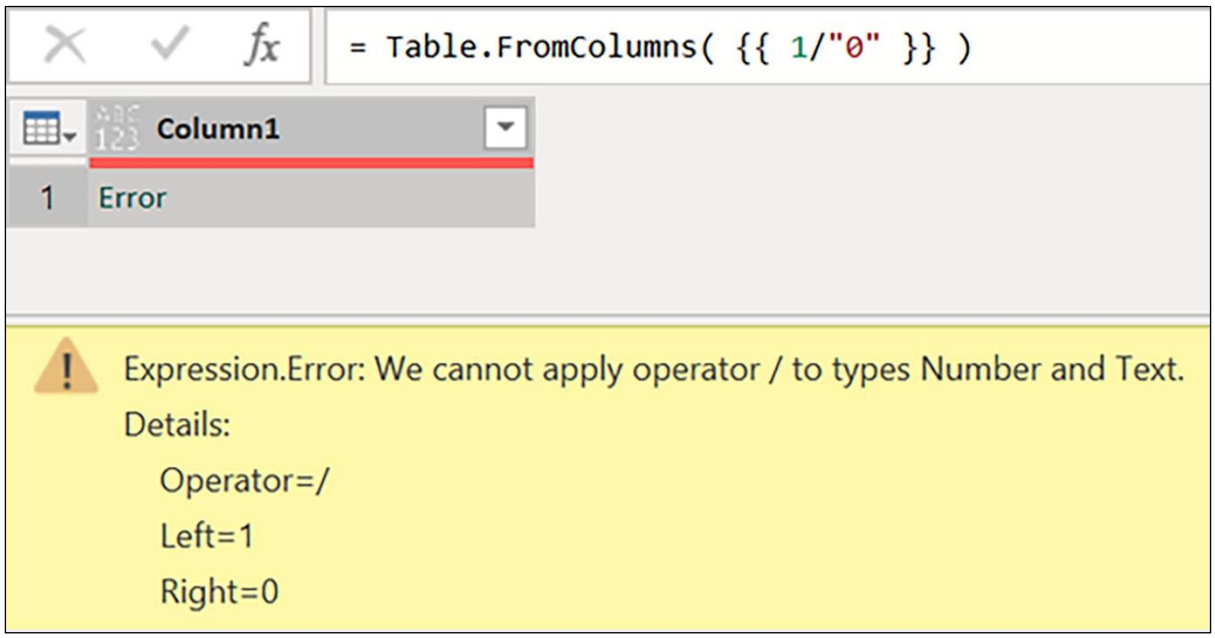
	Date	Details	Debit
1	05-02-24	CloudBliss Shopping	-30,99
2	05-02-24	Grocery Store	-15,50
3	null	Nature's Pantry	null
4	null	Card number 564	null
5	05-02-24	PowerPro Utilities	-90,00
6	null	Period: January 2024	null
7	null	Account: 123456789	null
8	null	Reference Number: PPU-7890	null
9	06-02-24	Gourmet Bistro	-75,50
10	null	Date: February 6, 2024	null
11	null	123 Main Street, Anytown	null
12	null	Card number 564	null

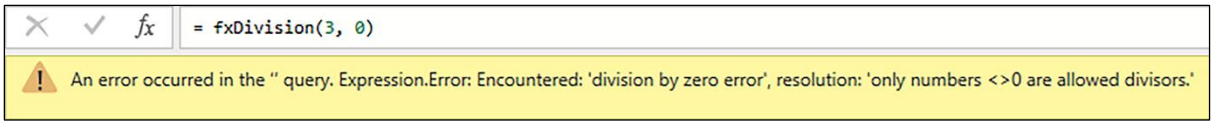


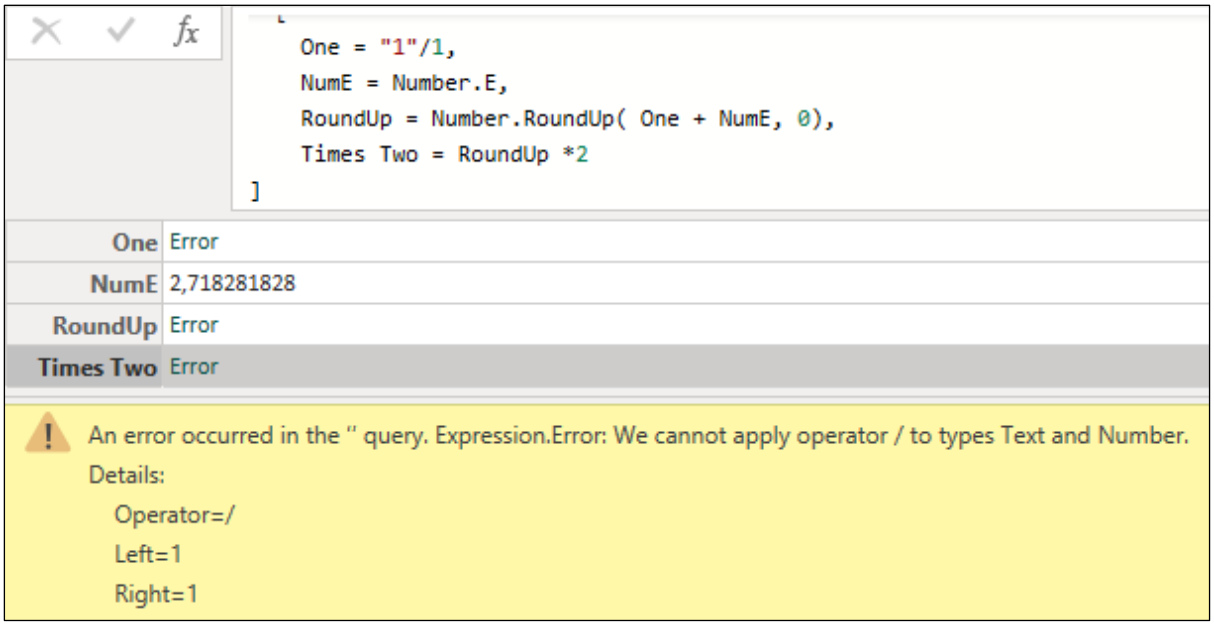
	Date	Description	1.2 Debit
1	*05-02-24*	CloudBliss Shopping	*-30,99*
2	*05-02-24*	Grocery Store Nature's Pantry Card number 564	*-15,5*
3	*05-02-24*	PowerPro Utilities Period: January 2024 Account: 123456789 Referenc...	*-90*
4	*06-02-24*	Gourmet Bistro Date: February 6, 2024 123 Main Street, Anytown Car...	*-75,5*
5	*06-02-24*	TrustWise Bank ATM cash withdrawl Location: 456 Oak Street, Anytow...	*-150*
6	*06-02-24*	HappyTimes: PQE-789012	*5*

Chapter 12: Handling Errors and Debugging



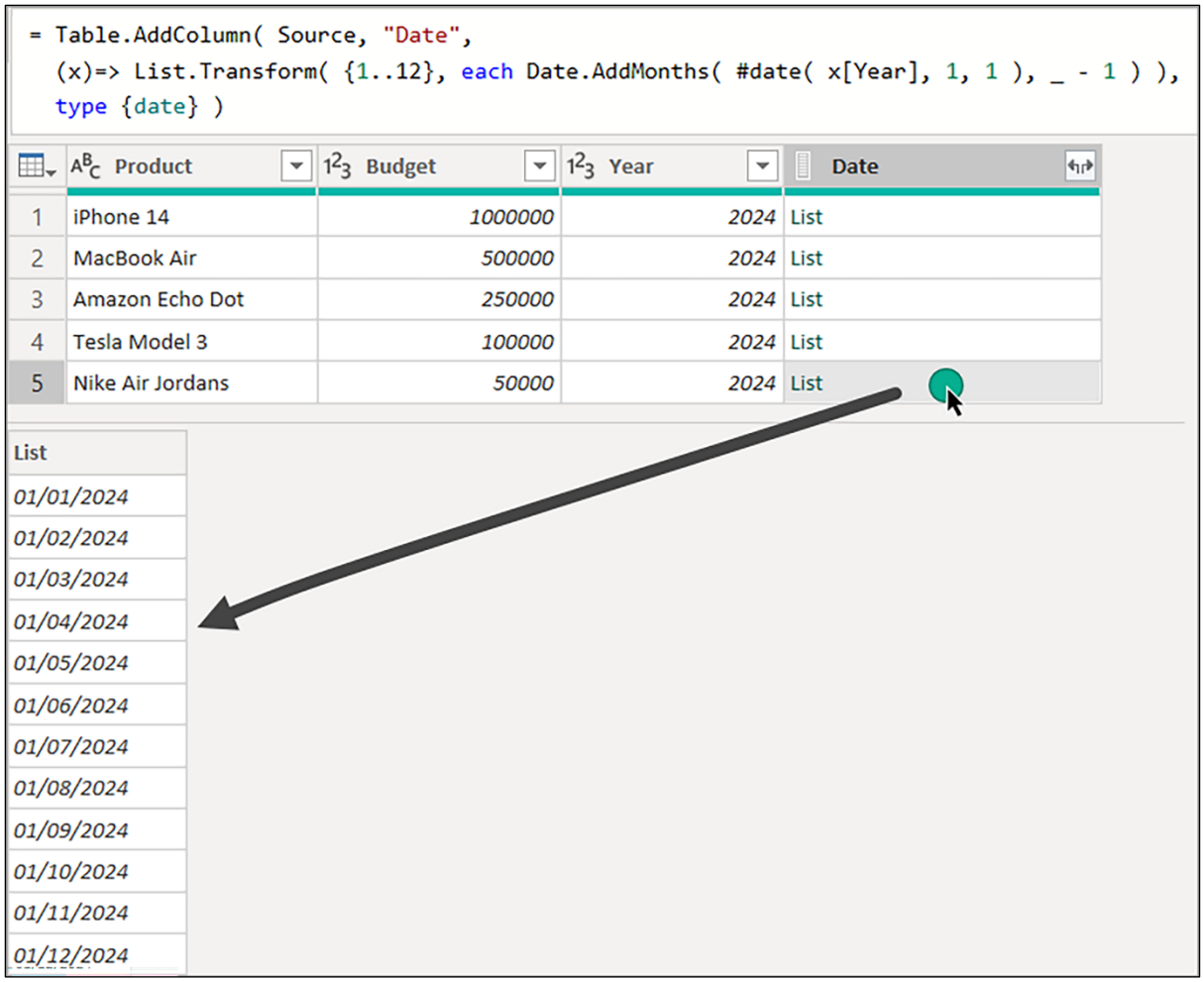



= fxDivision(3, 0)
An error occurred in the '' query. Expression.Error: Encountered: 'division by zero error', resolution: 'only numbers <>0 are allowed divisors.'


One = "1"/1,
NumE = Number.E,
RoundUp = Number.RoundUp(One + NumE, 0),
Times Two = RoundUp *2
]
One Error
NumE 2,718281828
RoundUp Error
Times Two Error
An error occurred in the '' query. Expression.Error: We cannot apply operator / to types Text and Number.
Details:
Operator=/
Left=1
Right=1

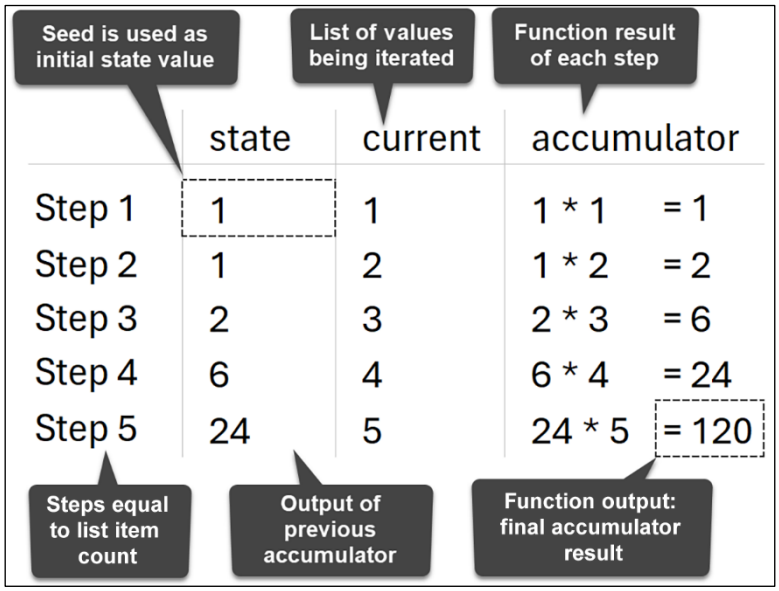
Chapter 13: Iteration and Recursion

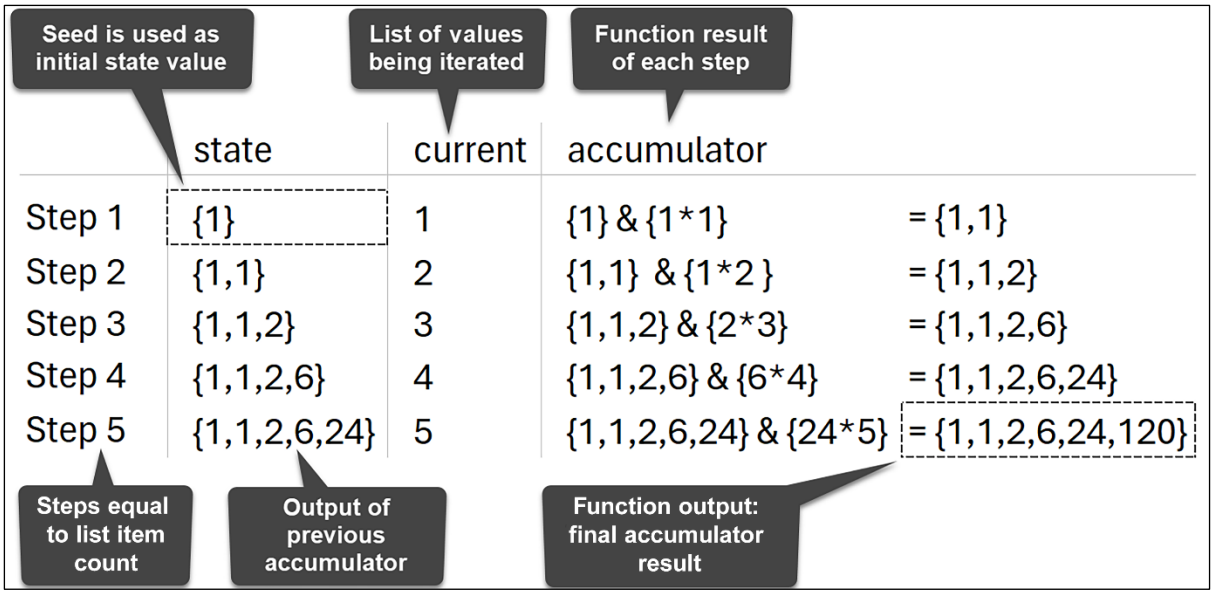
	Product	Budget	Year
1	iPhone 14	1000000	2024
2	MacBook Air	500000	2024
3	Amazon Echo Dot	250000	2024
4	Tesla Model 3	100000	2024
5	Nike Air Jordans	50000	2024



	Product	Budget	Year	Date
1	iPhone 14	1000000	2024	01/01/2024
2	iPhone 14	1000000	2024	01/02/2024
3	iPhone 14	1000000	2024	01/03/2024
4	iPhone 14	1000000	2024	01/04/2024
5	iPhone 14	1000000	2024	01/05/2024
6	iPhone 14	1000000	2024	01/06/2024
7	iPhone 14	1000000	2024	01/07/2024
8	iPhone 14	1000000	2024	01/08/2024
9	iPhone 14	1000000	2024	01/09/2024
10	iPhone 14	1000000	2024	01/10/2024
11	iPhone 14	1000000	2024	01/11/2024
12	iPhone 14	1000000	2024	01/12/2024

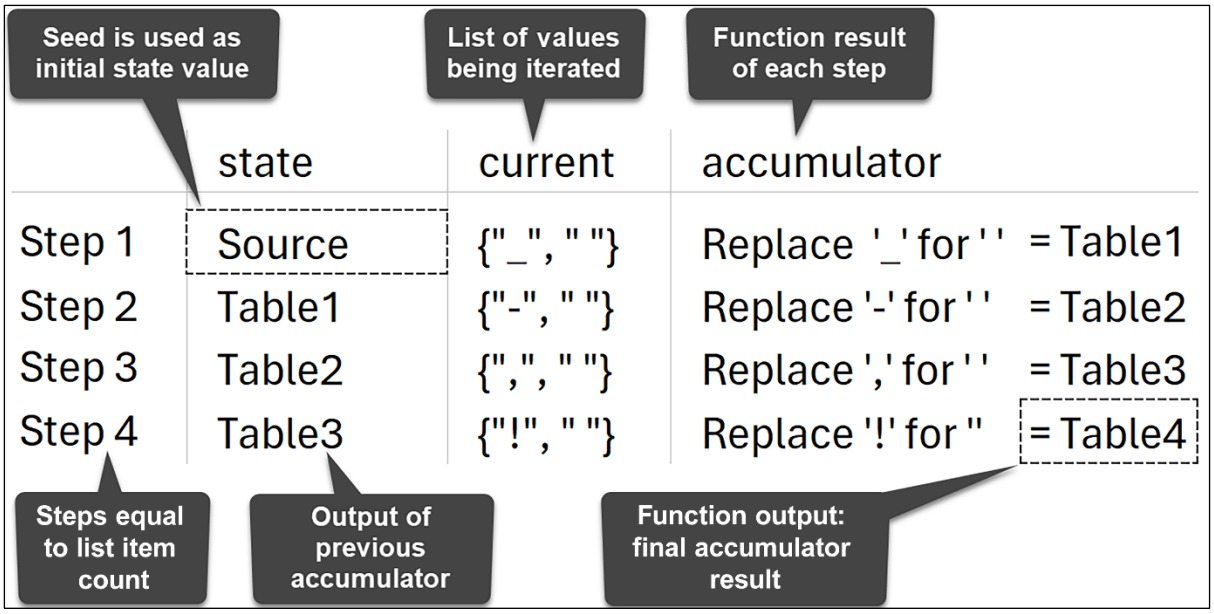
	Product	1.2 Budget	Year	Date
1	iPhone 14	83333.33333	2024	01/01/2024
2	iPhone 14	83333.33333	2024	01/02/2024
3	iPhone 14	83333.33333	2024	01/03/2024
4	iPhone 14	83333.33333	2024	01/04/2024
5	iPhone 14	83333.33333	2024	01/05/2024
6	iPhone 14	83333.33333	2024	01/06/2024
7	iPhone 14	83333.33333	2024	01/07/2024
8	iPhone 14	83333.33333	2024	01/08/2024
9	iPhone 14	83333.33333	2024	01/09/2024
10	iPhone 14	83333.33333	2024	01/10/2024
11	iPhone 14	83333.33333	2024	01/11/2024
12	iPhone 14	83333.33333	2024	01/12/2024



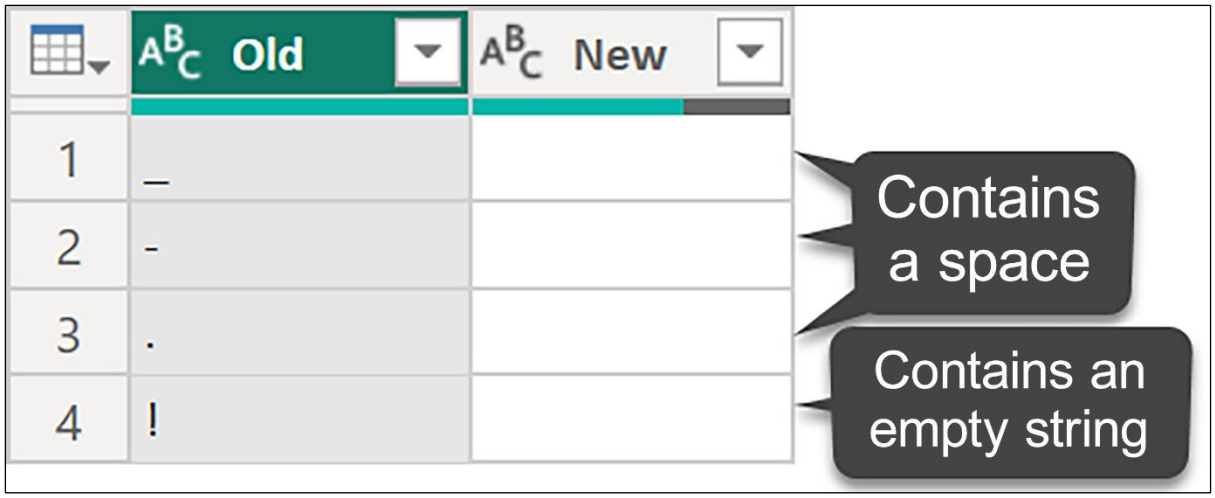

Seed is used as initial state value
List of values being iterated
Function result of each step
state
current
accumulator
Step 1
{1}
1
{1} & {1*1}
= {1,1}
Step 2
{1,1}
2
{1,1} & {1*2 }
= {1,1,2}
Step 3
{1,1,2}
3
{1,1,2} & {2*3}
= {1,1,2,6}
Step 4
{1,1,2,6}
4
{1,1,2,6} & {6*4}
= {1,1,2,6,24}
Step 5
{1,1,2,6,24}
5
{1,1,2,6,24} & {24*5}
= {1,1,2,6,24,120}
Steps equal to list item count
Output of previous accumulator
Function output: final accumulator result

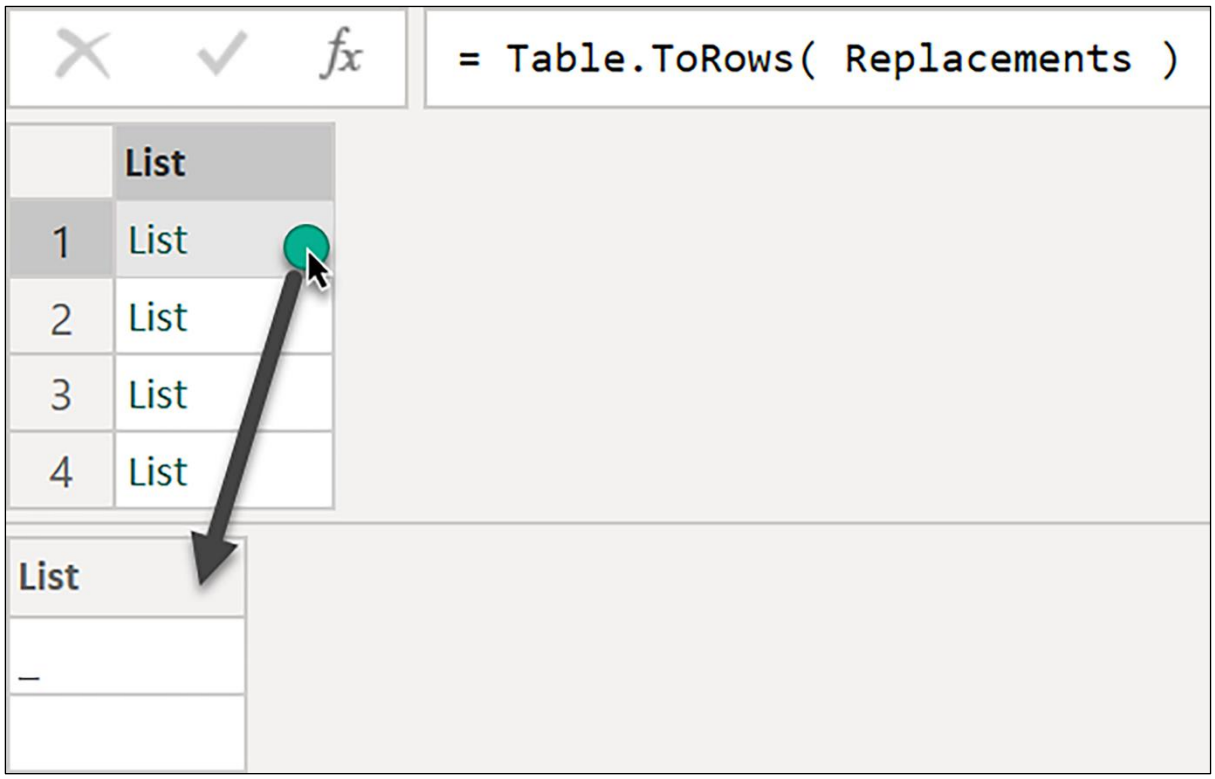
	Names
1	Rick_de_Groot!
2	Greg-Deckler!
3	Melissa.de_Korte
4	Brian.Julius!
5	John.Doe!
6	Jane_Smith!
7	Sally.Jones!
8	Tommy_Shelby!

```
List.Accumulate(
    { {"_"," "}, {"-", " "}, {".", " "}, {"!",""} }, // Sets of replacements
    Source, // Initial table value (seed)
    ( state, current ) =>
      Table.ReplaceValue(
        state,      // Table being updated with each replacement
        current{0}, // Character to be replaced - first item from the current pair
        current{1}, // Replacement character - second item from the current pair
        Replacer.ReplaceText,{"Names"} // Column name where replacements are made
      )
)
```


Seed is used as initial state value
List of values being iterated
Function result of each step
state
current
accumulator
Step 1
Source
{"_", " "}
Replace '_' for ' ' = Table1
Step 2
Table1
{"-", " "}
Replace '-' for ' ' = Table2
Step 3
Table2
{",", " "}
Replace ',' for ' ' = Table3
Step 4
Table3
{"!", " "}
Replace '!' for '' = Table4
Steps equal to list item count
Output of previous accumulator
Function output: final accumulator result

	Names
1	Rick de Groot
2	Greg Deckler
3	Melissa de Korte
4	Brian Julius
5	John Doe
6	Jane Smith
7	Sally Jones
8	Tommy Shelby


Old
New
1
_
2
-
3
.
4
!
Contains a space
Contains an empty string


= Table.ToRows(Replacements)
List
1
List
2
List
3
List
4
List
List
_

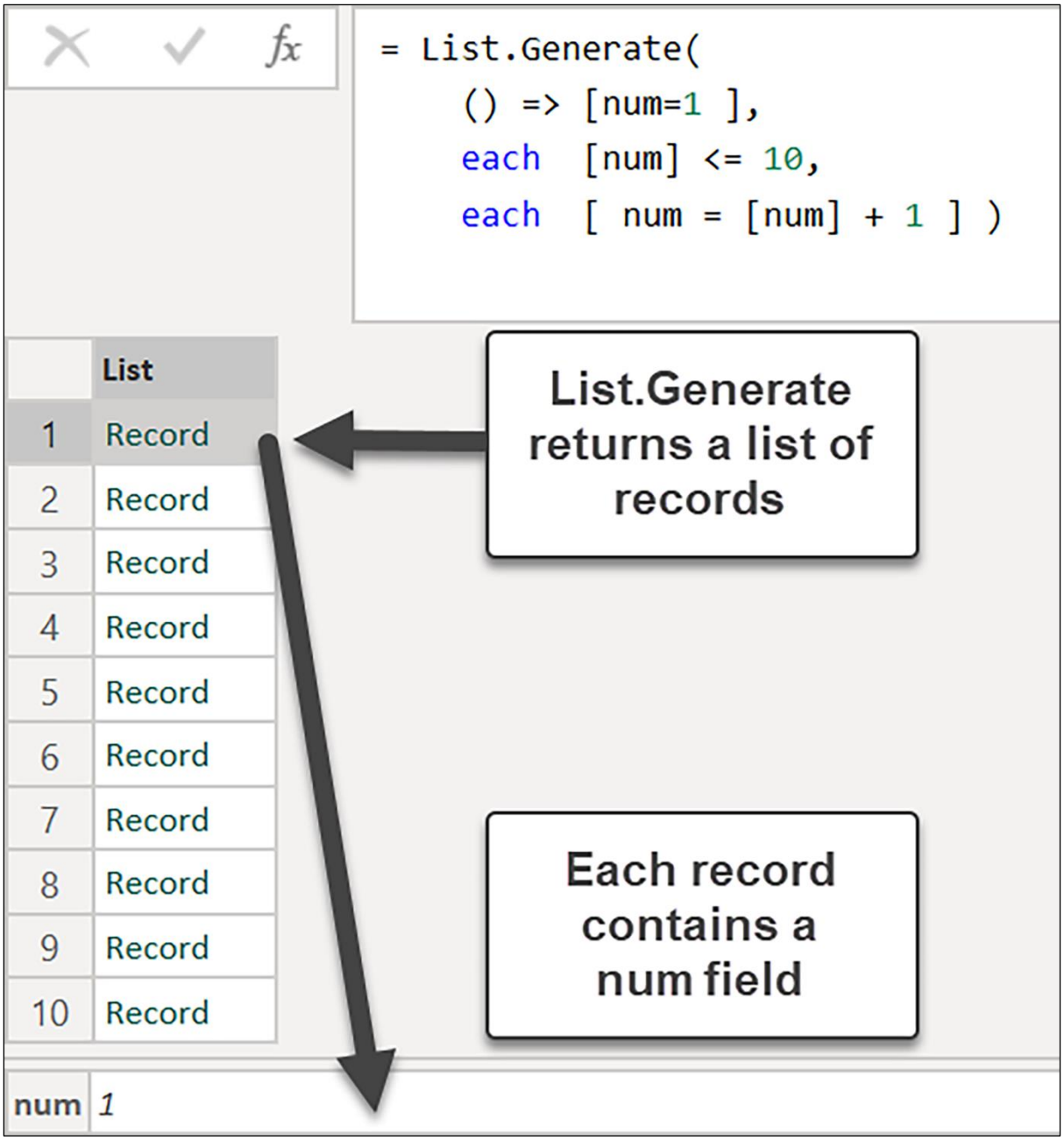
	List
1	01/01/2024
2	01/02/2024
3	01/03/2024
4	01/04/2024
5	01/05/2024
6	01/06/2024
7	01/07/2024
8	01/08/2024
9	01/09/2024
10	01/10/2024
11	01/11/2024
12	01/12/2024

```
= List.Generate(
      () => #date( 2024,1,1 ),     // starting value
      each _ < #date( 2025,1,1 ), // only for 2024
      each Date.AddMonths( _, 1)  // increment by month
)
```

Explanation:

- Generates a sequence of dates
- Beginning on January 1, 2024
- Increasing by one month at a time
- Up to December 1, 2024

	List
1	January 1, 2024 is on a Monday
2	February 1, 2024 is on a Thursday
3	March 1, 2024 is on a Friday
4	April 1, 2024 is on a Monday
5	May 1, 2024 is on a Wednesday
6	June 1, 2024 is on a Saturday
7	July 1, 2024 is on a Monday
8	August 1, 2024 is on a Thursday
9	September 1, 2024 is on a Sunday
10	October 1, 2024 is on a Tuesday
11	November 1, 2024 is on a Friday
12	December 1, 2024 is on a Sunday


= List.Generate(
() => [num=1],
each [num] <= 10,
each [num = [num] + 1])
List
1 Record
2 Record
3 Record
4 Record
5 Record
6 Record
7 Record
8 Record
9 Record
10 Record
List.Generate returns a list of records
Each record contains a num field
num 1

```
= Table.FromRecords(
    List.Generate(
      () => [num=1 ],
      each  [num] <= 10,
      each  [ num = [num] + 1 ] ) )
```

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

```
= Table.FromRecords(
    List.Generate(
      () => [num=1,             IsEven = Number.IsEven( 1 ) ],
      each  [num] <= 10,
      each  [ num = [num] + 1, IsEven = Number.IsEven( [num] + 1 ) ] ) )
```

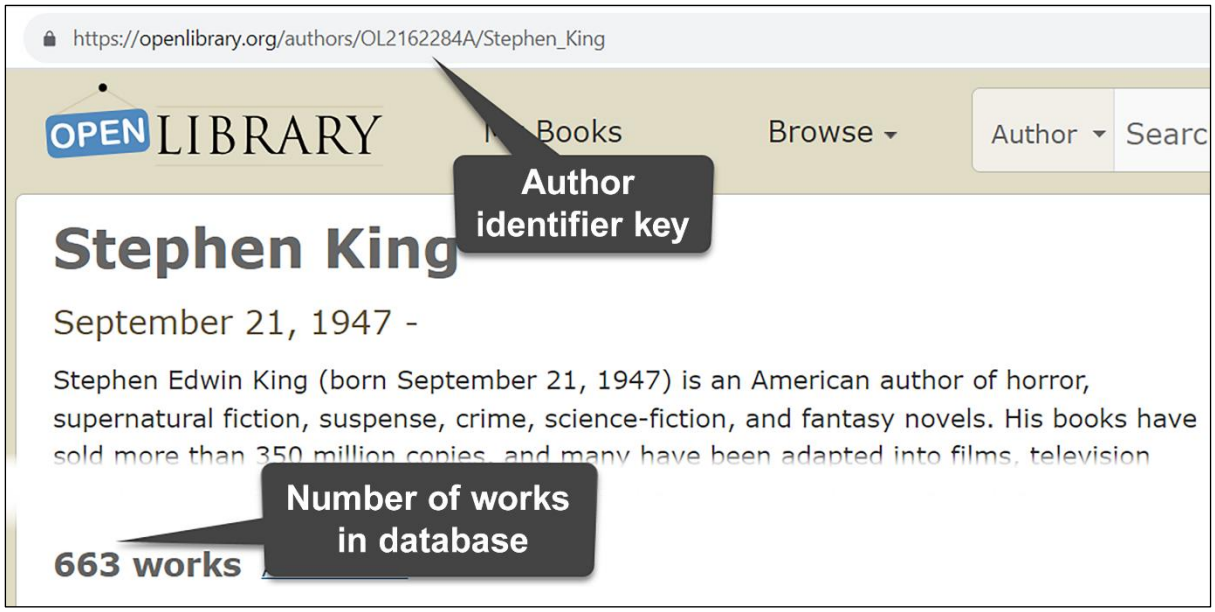
	num	IsEven
1	1	FALSE
2	2	TRUE
3	3	FALSE
4	4	TRUE
5	5	FALSE
6	6	TRUE
7	7	FALSE
8	8	TRUE
9	9	FALSE
10	10	TRUE

Works by an Author

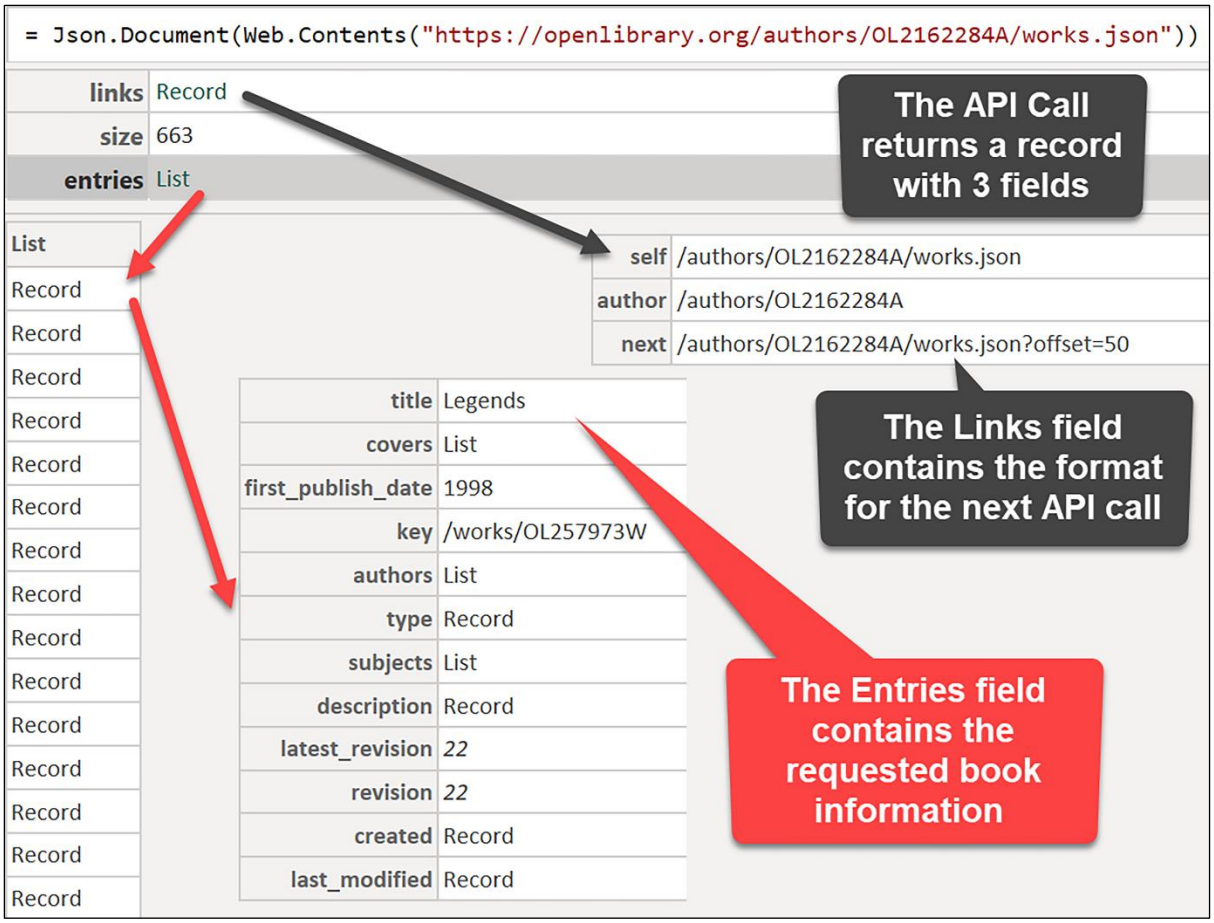
https://openlibrary.org/authors/OL23919A/works.json

The above URL will return 50 works by an author.

If you want to paginate, you can set offset like so:
https://openlibrary.org/authors/OL1394244A/works.json?offset=50

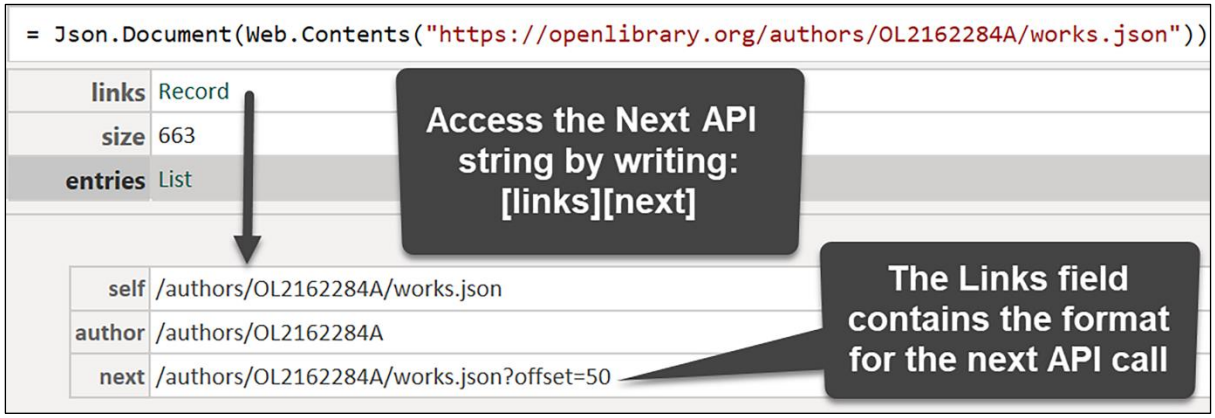

https://openlibrary.org/authors/OL2162284A/Stephen_King
OPEN LIBRARY
Books
Browse
Author
Searc
Author identifier key
Stephen King
September 21, 1947 -
Stephen Edwin King (born September 21, 1947) is an American author of horror, supernatural fiction, suspense, crime, science-fiction, and fantasy novels. His books have sold more than 350 million copies, and many have been adapted into films, television
Number of works in database
663 works


From Web
Basic
Advanced
URL
https://openlibrary.org/authors/OL2162284A/works.json
OK
Cancel

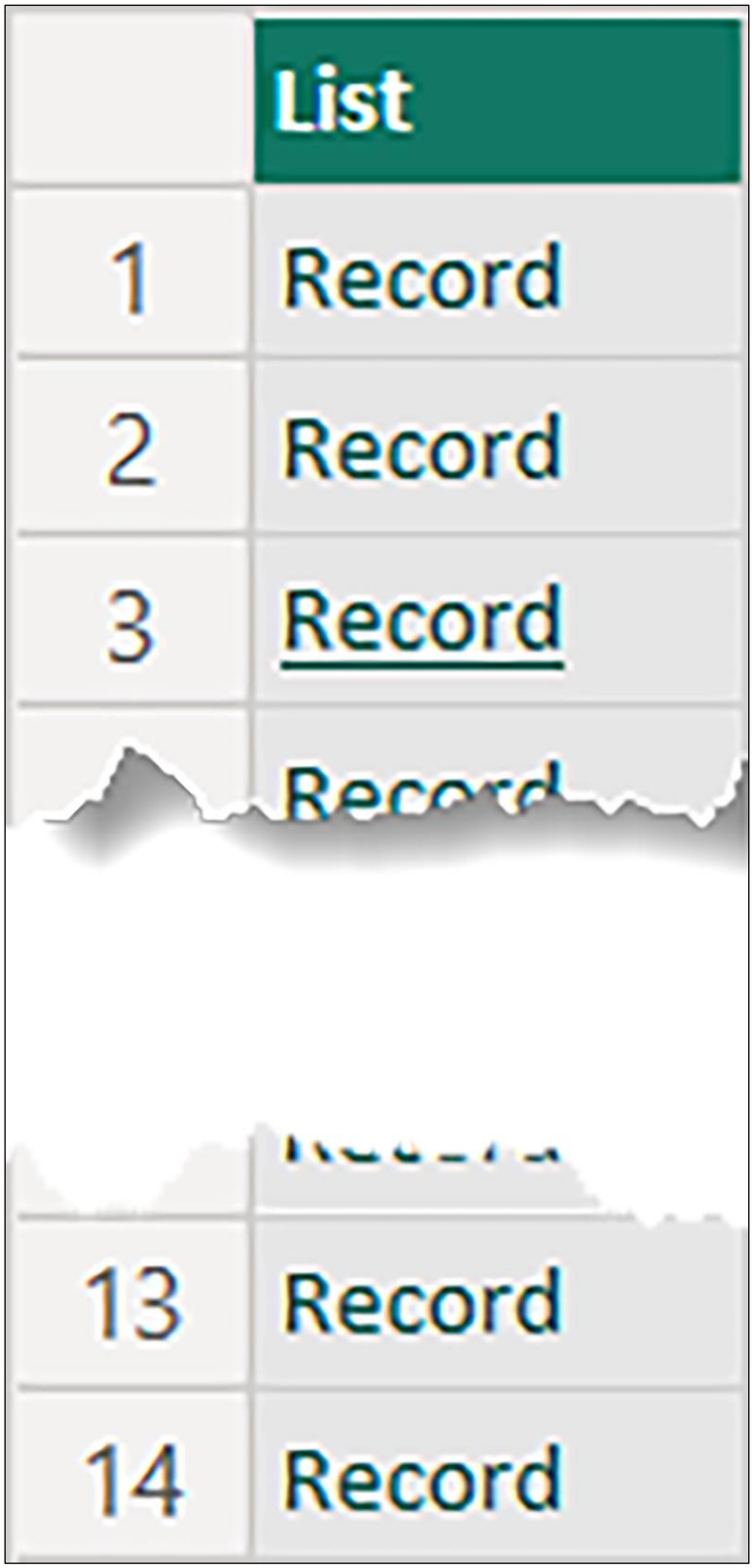

= Json.Document(Web.Contents("https://openlibrary.org/authors/OL2162284A/works.json"))
links Record
size 663
entries List
The API Call returns a record with 3 fields
List
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
self /authors/OL2162284A/works.json
author /authors/OL2162284A
next /authors/OL2162284A/works.json?offset=50
The Links field contains the format for the next API call
title Legends
covers List
first_publish_date 1998
key /works/OL257973W
authors List
type Record
subjects List
description Record
latest_revision 22
revision 22
created Record
last_modified Record
The Entries field contains the requested book information

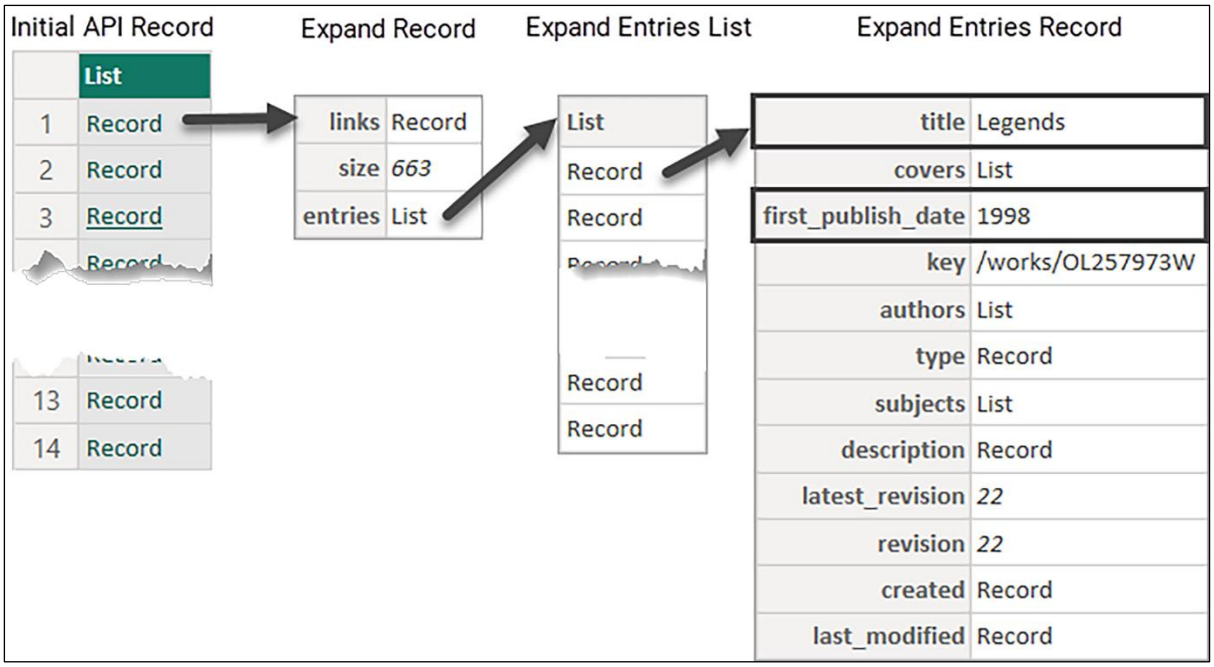

= Json.Document(Web.Contents("https://openlibrary.org/authors/OL2162284A/works.json?offset=999"))
links Record
size 663
entries List
self /authors/OL2162284A/works.json?offset=999
author /authors/OL2162284A
prev /authors/OL2162284A/works.json?offset=0
Skip 999 records
There is no next URL
List
Returns an empty list

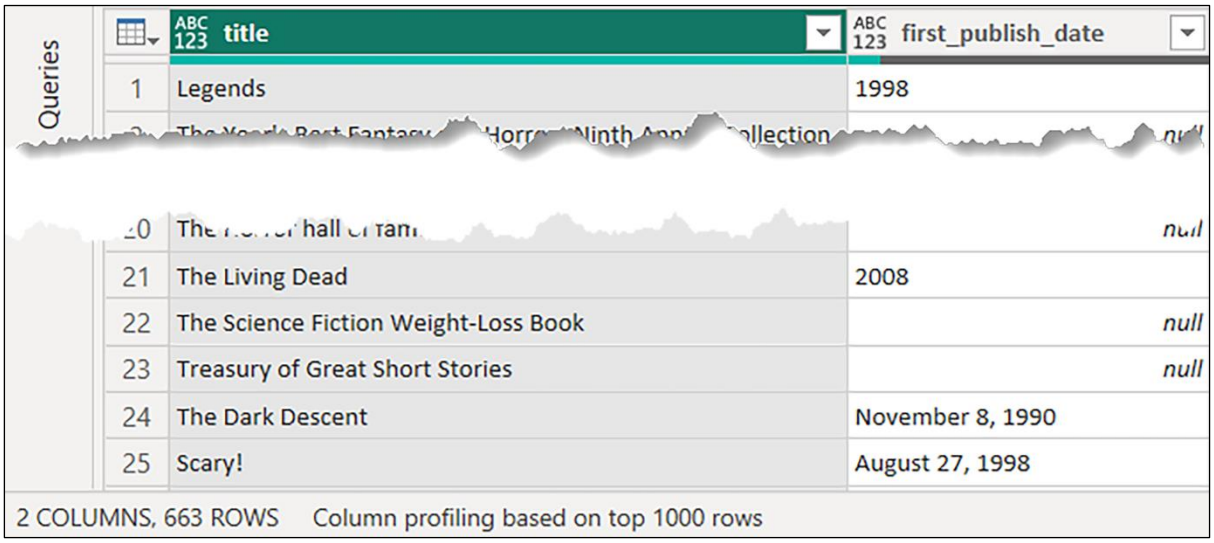
```
List.Generate(
  () =>
    [
     Request = fxGetData( BaseURL & OffsetSuffix ),          Initial
     HasNext = true
    ],
  each [HasNext],                                          Condition
  each
   [
    Request = fxGetData( BaseURL & [Request][links][next] ),  Next
    HasNext = Record.HasFields( [Request][links], "next")
   ],
  each [Request]                                           Selector
)
```



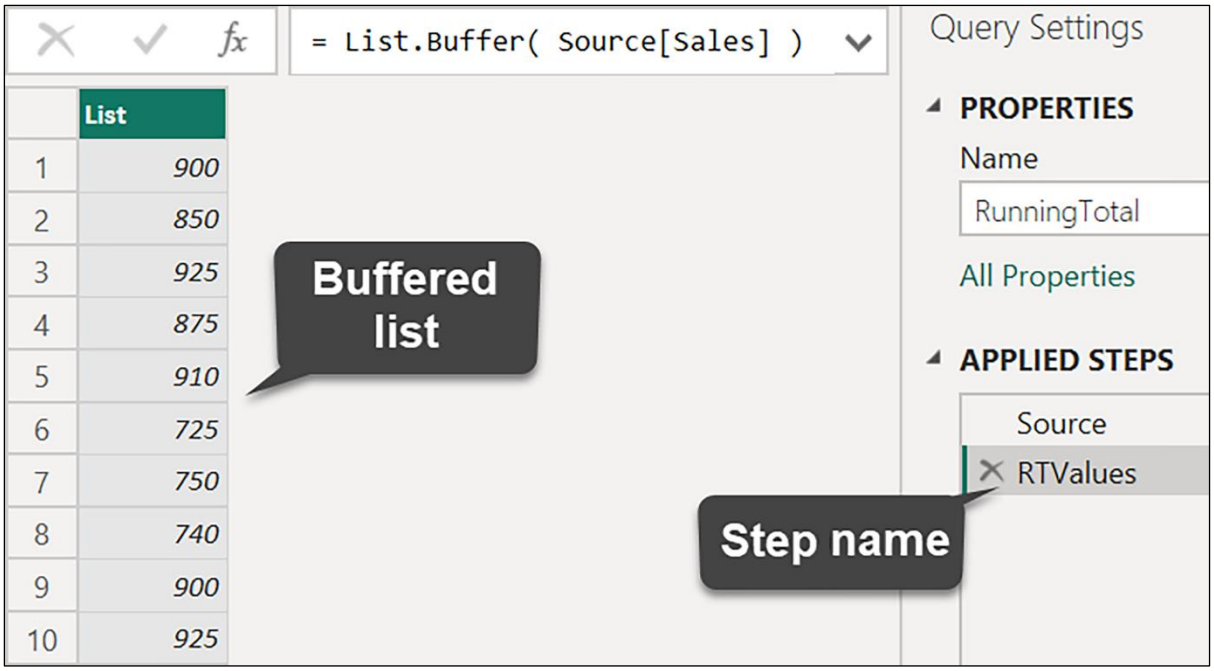

```
List.Generate(
  () =>
    [
      Counter = 0,                                  Initial
      Request = fxGetData( 0 )
    ],
  each not List.IsEmpty( [Request][entries] ),      Condition
  each
    [
      Counter = [Counter] + 50,                     Next
      Request = fxGetData( [Counter] + 50 ) ],
  each [Request]                                    Selector
)
```

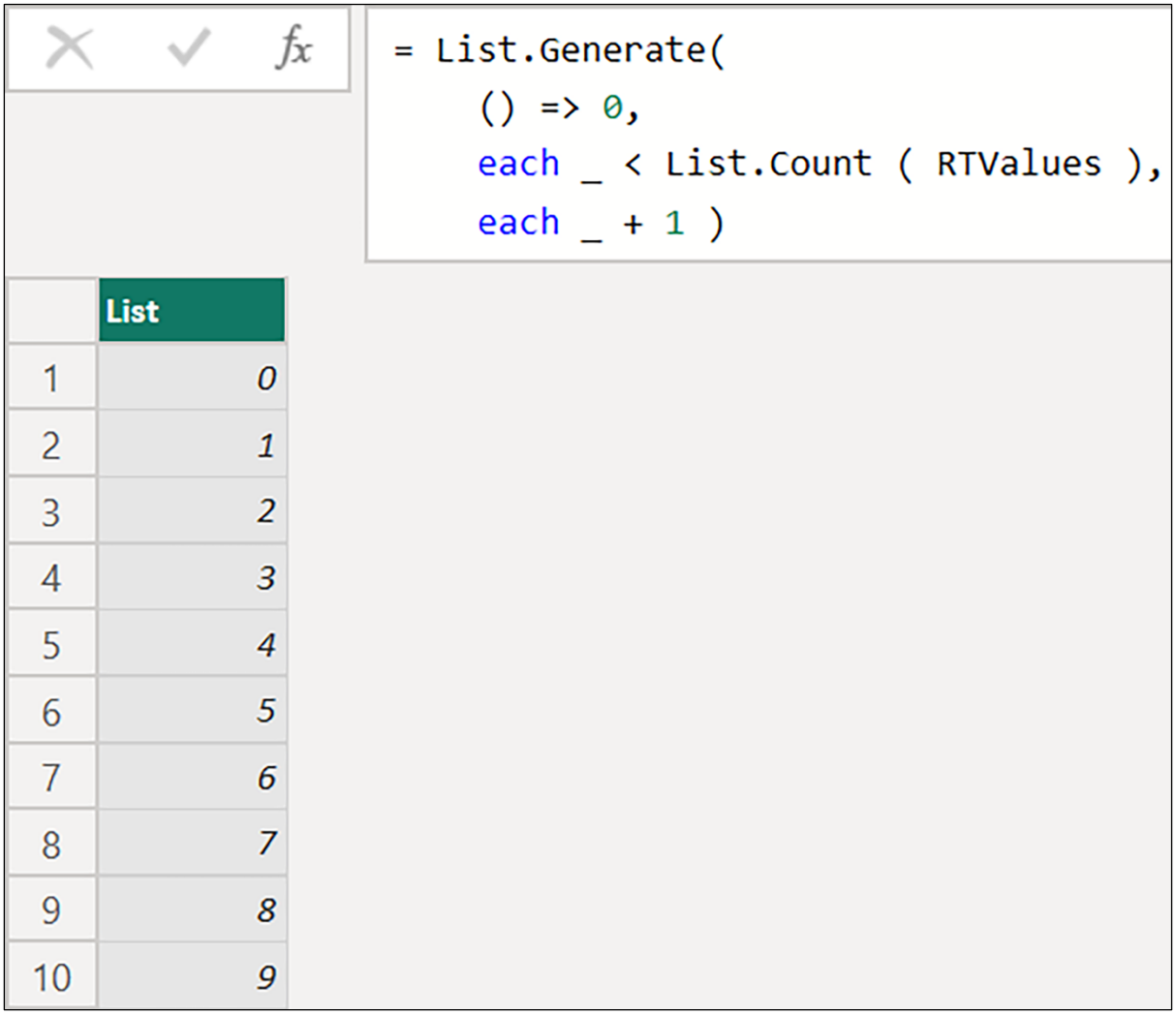

List
1
Record
2
Record
3
Record
13
Record
14
Record

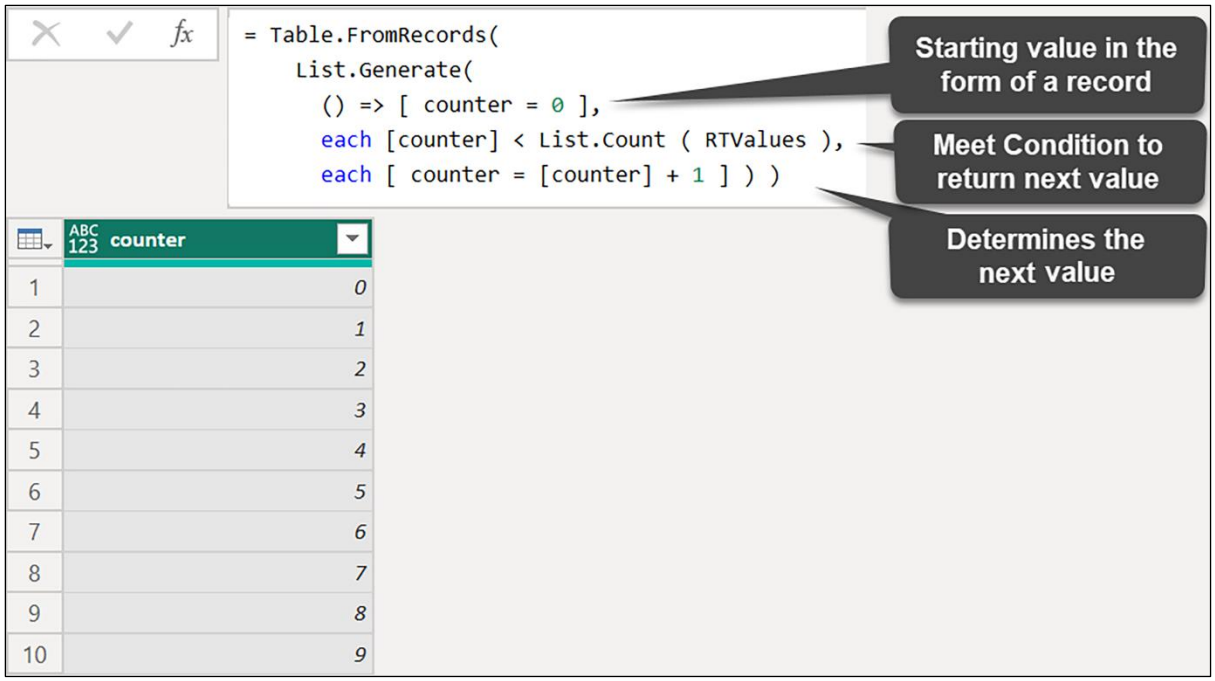

Initial API Record
Expand Record
Expand Entries List
Expand Entries Record
List
1 Record
2 Record
3 Record
13 Record
14 Record
links Record
size 663
entries List
List
Record
Record
Record
Record
title Legends
covers List
first_publish_date 1998
key /works/OL257973W
authors List
type Record
subjects List
description Record
latest_revision 22
revision 22
created Record
last_modified Record


Queries
ABC 123 title
ABC 123 first_publish_date
1 Legends 1998
21 The Living Dead 2008
22 The Science Fiction Weight-Loss Book null
23 Treasury of Great Short Stories null
24 The Dark Descent November 8, 1990
25 Scary! August 27, 1998
2 COLUMNS, 663 ROWS
Column profiling based on top 1000 rows

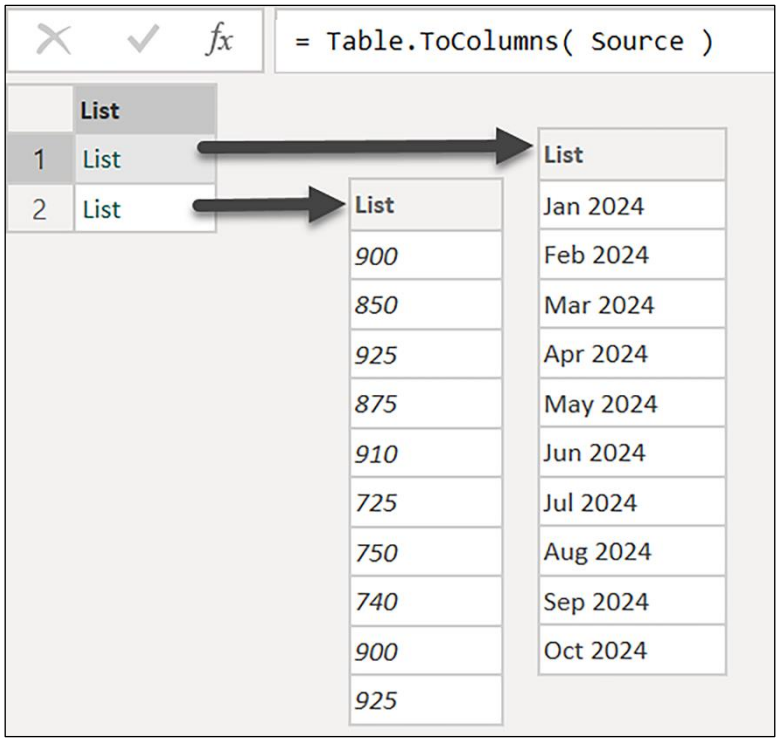
	Period	Sales
1	Jan 2024	900
2	Feb 2024	850
3	Mar 2024	925
4	Apr 2024	875
5	May 2024	910
6	Jun 2024	725
7	Jul 2024	750
8	Aug 2024	740
9	Sep 2024	900
10	Oct 2024	925


= List.Buffer(Source[Sales])
List
1 900
2 850
3 925
4 875
5 910
6 725
7 750
8 740
9 900
10 925
Buffered list
Query Settings
PROPERTIES
Name
RunningTotal
All Properties
APPLIED STEPS
Source
RTValues
Step name


= List.Generate(
() => 0,
each _ < List.Count (RTValues),
each _ + 1)
List
1 0
2 1
3 2
4 3
5 4
6 5
7 6
8 7
9 8
10 9



	ABC 123 counter	ABC 123 RT
1	0	900
2	1	1750
3	2	2675
4	3	3550
5	4	4460
6	5	5185
7	6	5935
8	7	6675
9	8	7575
10	9	8500



	List
1	List
2	List
3	List

	Column1	Column2	Column3
1	Jan 2024	*900*	*900*
2	Feb 2024	*850*	*1750*
3	Mar 2024	*925*	*2675*
4	Apr 2024	*875*	*3550*
5	May 2024	*910*	*4460*
6	Jun 2024	*725*	*5185*
7	Jul 2024	*750*	*5935*
8	Aug 2024	*740*	*6675*
9	Sep 2024	*900*	*7575*
10	Oct 2024	*925*	*8500*

	Period	Sales	Running Total
1	Jan 2024	900	900
2	Feb 2024	850	1750
3	Mar 2024	925	2675
4	Apr 2024	875	3550
5	May 2024	910	4460
6	Jun 2024	725	5185
7	Jul 2024	750	5935
8	Aug 2024	740	6675
9	Sep 2024	900	7575
10	Oct 2024	925	8500

Chapter 14: Troublesome Data Patterns

	String	Code
1	The code DGPQ33446 is important for this task.	DGPQ33446
2	Unique document identifier: DGPQ13295.	DGPQ13295
3	Please use the code DGPQ36006 to access the system.	DGPQ36006
4	For verification, enter DGPQ30881 on the website.	DGPQ30881
5	The transaction ID DGPQ78388 must be noted.	DGPQ78388
6	DGPQ10273 is the code for your appointment.	DGPQ10273
7	Reference code DGPQ36144 is included in the report.	DGPQ36144
8	To complete registration, use DGPQ90158 as your code.	DGPQ90158
9	Package with tracking number DGPQ52287 has been shipped.	DGPQ52287

	String	Code
1	The code CHLO33446 is important for this task.	CHLO33446
2	Unique document identifier: JXKE13295.	JXKE13295
3	Please use the code SBT36006 to access the system.	*null*
4	For verification, enter ERKZ30881 on the website.	ERKZ30881
5	The transaction ID YNZ78388 must be noted.	*null*
6	IHCY10273 is the code for your appointment.	IHCY10273
7	Reference code SSK36144 is included in the report.	*null*
8	To complete registration, use PRLL90158 as your code.	PRLL90158
9	Package with tracking number MGAK52287 has been shipped.	MGAK52287

	String	Code
1	The code CHLO33446 is IMPORTANT.	CHLO33446
2	Unique document identifier: JXKE13295.	JXKE13295
3	Please use the code SBT36006 to access the system.	
4	For verification, enter ERKZ30881 and MdKZ85426.	ERKZ30881, MdKZ85426
5	The transaction ID YNZ78388 must be noted.	
6	IHCY10273 is the code for your appointment.	IHCY10273
7	Reference code SSK36144 is included in the report.	
8	To complete registration, use PRLL90158 as your code.	PRLL90158
9	Package with tracking number MGA5K2287 has been shipped.	

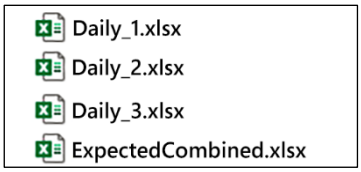
	Column1	Code
1	Hardware ZEQNNZE Nails - 2 i…	ZEQNNZE, 4SPYBBU8, WRPBOSGEHTFD
2	Software CGUL2L Antivirus S…	CGUL2L, ZH1R987S2, SADBT0
3	Hardware OY6IL4VFH21 Hammer - 5k…	OY6IL4VFH21, O08AUF8JG, 80LC6OMO
4	Software DLTQ80V7X Operating S…	DLTQ80V7X, NZ8DD797, AB9MTEI09L
5	FurnitureWCPKVSZJZ Office Chai…	WCPKVSZJX, FTHJK, QQMR1QZNR71
6	FurnitureZRE4CCR1 Office Desk…	ZRE4CCR1, OYOWTO7IOQMF, SNGRQ5
7	Hardware WJAA7DHJ Screwdriver…	WJAA7DHJ, EO65VPCJ, IGC2FHI8G
8	Software JX6URB9HI Database So…	JX6URB9HI, 143OD1ADFC4, 3YD7QU07T8OF
9	Hardware 1ADXHN Electric Dr…	1ADXHN, AK3M3MJMY6M, XUBQZP7R
10	FurnitureD20R63MNM6 Bookshelf …	D20R63MNM6, 4G6GD3G, 373NWC1I9JT5

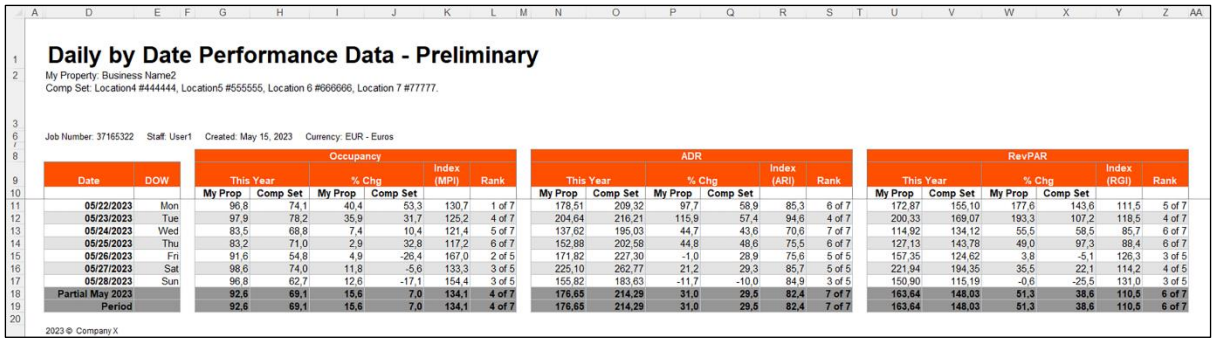
	String	Code
1	The code CHLO-33-446 is IMPORTANT.	CHLO-33-446
2	Unique document identifier: JXKE-13-295.	JXKE-13-295
3	Please use the code SBT3-6006 to access the system.	
4	For verification, enter ERKZ-30-881 and MdKZ-85.426.	ERKZ-30-881, MdKZ-85.426
5	The transaction ID YNZ-78-388 must be noted.	
6	IHC-Y10-273 is the code for your appointment.	
7	Reference code SSK-36-144 is included in the report.	
8	To complete registration, use PRLL-90-158 as your code.	PRLL-90-158
9	Package with tracking number MGA5-K22.87 has been shipped.	

```
let
    fxRegex = (input as text) as text =>
        Web.Page(
            "<script>
                var a = '" & input & "';
                var b = a.match(/\d{5}/g);
                document.write(b);
            </script>"
        ){0}[Data]{0}[Children]{1}[Children]{0}[Text],
    Source = Table.FromColumns({
        {
```

	Address	Postal Code
1	Boulevard des Écoles 73, 31000 Lyon	31000
2	6 Boulevard du Château, 69001 La Ville Rose Toulouse	69001
3	Rue Saint-Martin 65, 31000 Lyon	31000
4	Chemin Victor Hugo 143, 69001 Bordeaux	69001
5	Avenue des Vignes 7, 67000 Toulouse	67000
6	74 Quai de la République 69001 Latin Quarter Paris	69001
7	55 Boulevard de la Liberté, 67000 La Petite France Strasbourg	67000
8	82 Chemin des Jardins, 59000 Paris	59000

{"31000","69001","31000","69001","67000","69001","67000","59000"}





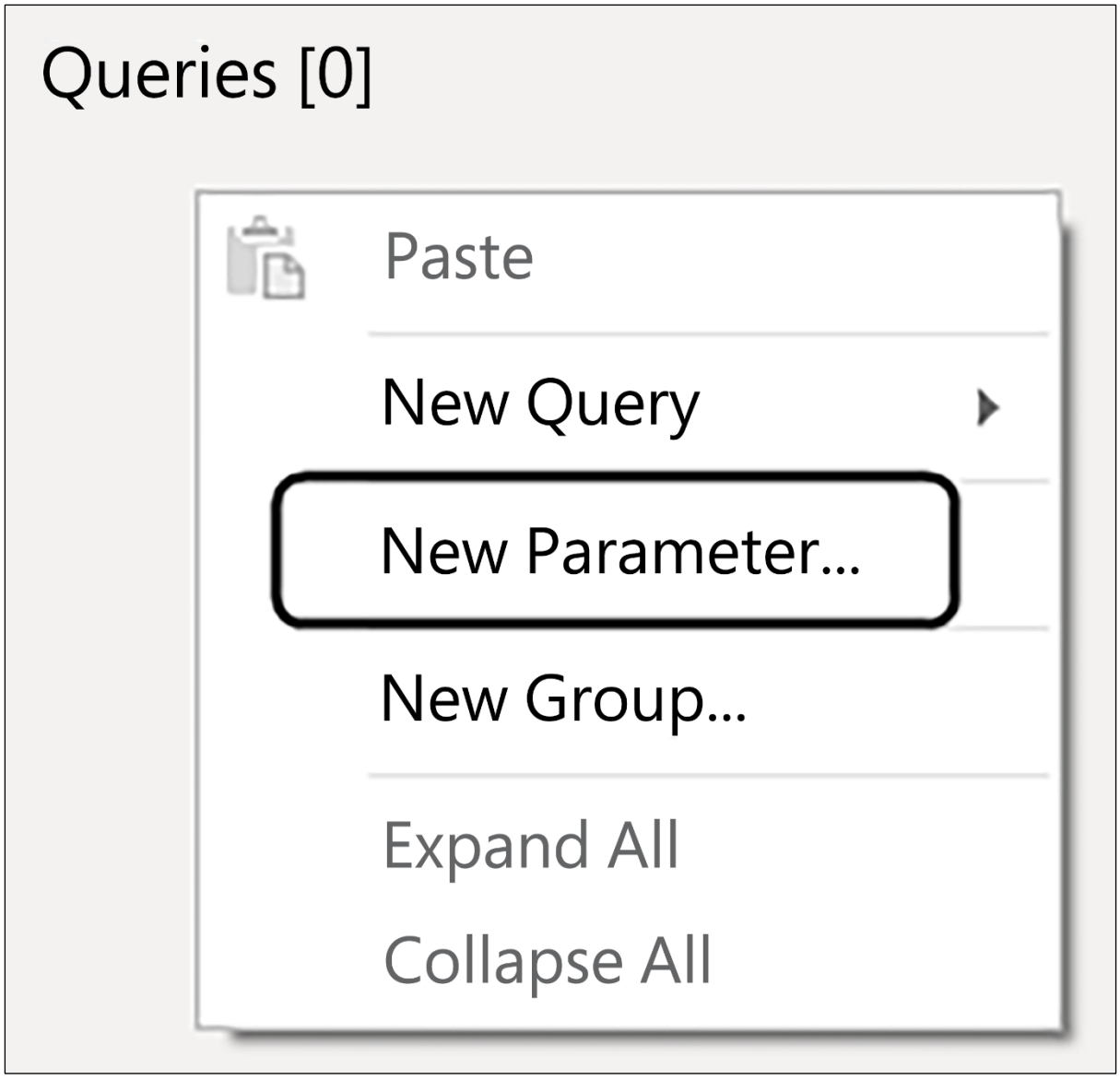
Daily by Date Performance Data - Preliminary

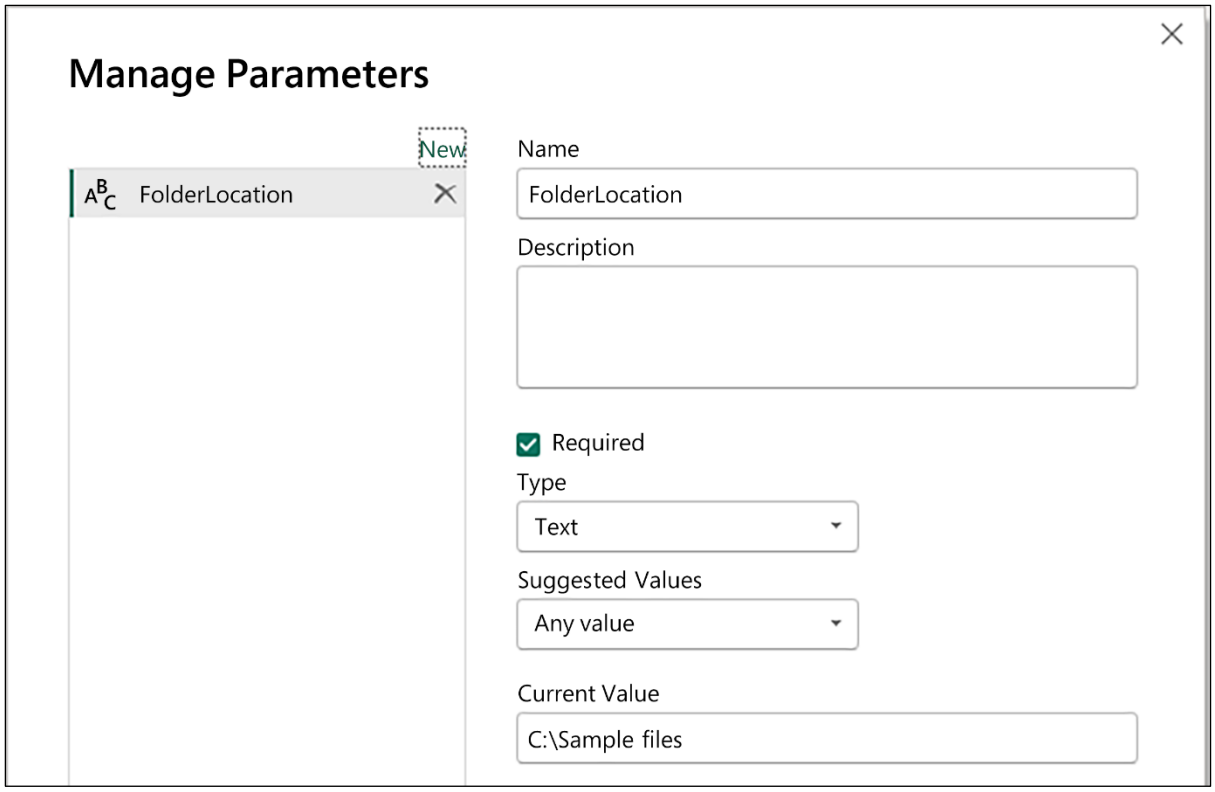
My Property: Business Name2
Comp Set: Location4 #444444, Location5 #555555, Location 6 #666666, Location 7 #77777.

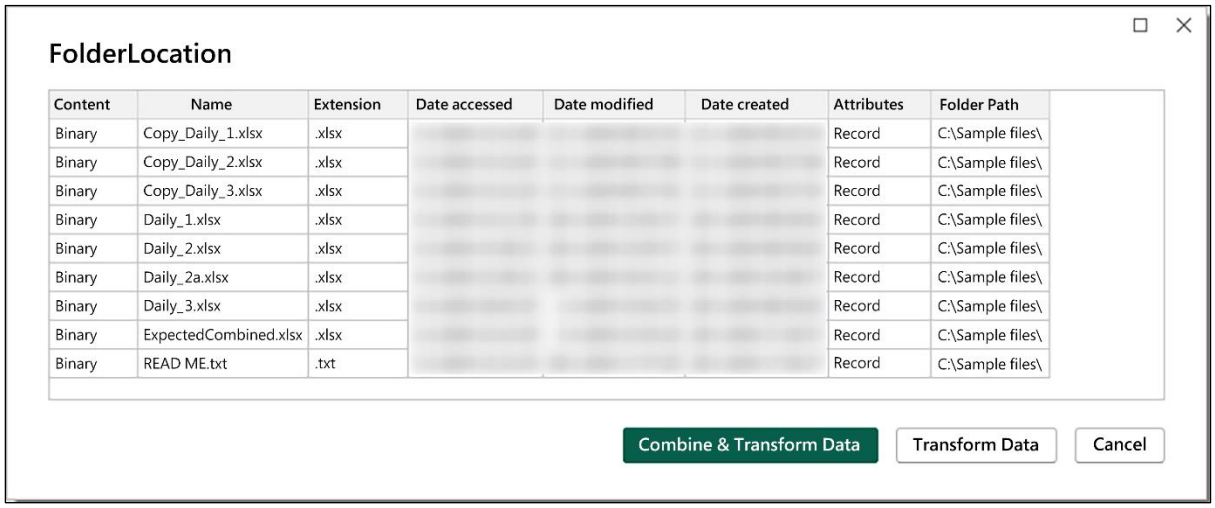
Job Number: 37165322 Staff: User1 Created: May 15, 2023 Currency: EUR - Euros

Date	DOW	Occupancy						ADR						RevPAR					
		This Year		% Chg		Index (MPI)	Rank	This Year		% Chg		Index (ARI)	Rank	This Year		% Chg		Index (RGI)	Rank
		My Prop	Comp Set	My Prop	Comp Set			My Prop	Comp Set	My Prop	Comp Set			My Prop	Comp Set	My Prop	Comp Set		
05/22/2023	Mon	96,8	74,1	40,4	53,3	130,7	1 of 7	178,51	209,32	97,7	58,9	85,3	6 of 7	172,87	155,10	177,6	143,6	111,5	5 of 7
05/23/2023	Tue	97,9	78,2	35,9	31,7	125,2	4 of 7	204,64	216,21	115,9	57,4	94,6	4 of 7	200,33	169,07	193,3	107,2	118,5	4 of 7
05/24/2023	Wed	83,5	68,8	7,4	10,4	121,4	5 of 7	137,62	195,03	44,7	43,6	70,6	7 of 7	114,92	134,12	55,5	58,5	85,7	6 of 7
05/25/2023	Thu	83,2	71,0	2,9	32,8	117,2	6 of 7	152,88	202,58	44,8	48,6	75,5	6 of 7	127,13	143,78	49,0	97,3	88,4	6 of 7
05/26/2023	Fri	91,6	64,8	4,9	-26,4	167,0	2 of 5	171,82	227,30	-1,0	28,9	75,6	5 of 5	157,35	124,62	3,8	-5,1	126,3	3 of 5
05/27/2023	Sat	98,6	74,0	11,8	-5,6	133,3	3 of 5	225,10	262,77	21,2	29,3	85,7	5 of 5	221,94	194,35	35,5	22,1	114,2	4 of 5
05/28/2023	Sun	96,8	62,7	12,6	-17,1	154,4	3 of 5	155,82	183,63	-11,7	-10,0	84,9	3 of 5	150,90	115,19	-0,6	-25,5	131,0	3 of 5
Partial May 2023		**92,6**	**69,1**	**15,6**	**7,0**	**134,1**	**4 of 7**	**176,65**	**214,29**	**31,0**	**29,5**	**82,4**	**7 of 7**	**163,64**	**148,03**	**51,3**	**38,6**	**110,5**	**6 of 7**
Period		**92,6**	**69,1**	**15,6**	**7,0**	**134,1**	**4 of 7**	**176,65**	**214,29**	**31,0**	**29,5**	**82,4**	**7 of 7**	**163,64**	**148,03**	**51,3**	**38,6**	**110,5**	**6 of 7**

	A	B	C	D	E	F	G	H	I
1	Name	Date	DOW	Occupancy | This Year | My Prop	Occupancy | This Year | Comp Set	Occupancy | % Chg | My Prop	Occupancy | % Chg | Comp Set	Occupancy | Index (MPI)	Occupancy | Rank
2	Daily_1.xlsx	5/22/2023	Mon	89.393939393939391	93.126385809312623	69.540229885057471	73.553719008264466	95.992063492063494	5 of 6
3	Daily_1.xlsx	5/23/2023	Tue	100	98.59571322985957	87.5	77.158034528552463	101.42428785607196	1 of 6
4	Daily_1.xlsx	5/24/2023	Wed	70.707070707070713	84.405025868440504	28.205128205128204	48.119325551232166	83.771161704611785	6 of 6
5	Daily_1.xlsx	5/25/2023	Thu	61.616161616161612	77.605321507760536	9.3189964157706093	12.419700214132762	79.396825396825392	5 of 6
6	Daily_1.xlsx	5/26/2023	Fri	71.717171717171713	77.646892830745	10.59190031152648	-17.[illegible]79.7743	[illegible]	5 of 6
19	Daily_3.xlsx	5/26/2023	Thu	72.631589473684425	79.171741778519117	-21.[illegible]	-3.[illegible]422804749558	9[illegible]	5 of 5
20	Daily_3.xlsx	5/26/2023	Fri	73.684210526315795	77.611940298507463	-19.185059422750424	-18.16082983773941	94.939271255060731	4 of 5
21	Daily_3.xlsx	5/27/2023	Sat	88.421052631578945	96.865671641791039	-6.052631578947369	3.723418306696606	91.282134457870399	5 of 5
22	Daily_3.xlsx	5/28/2023	Sun	84.210526315789465	93.283582089552226	-10.526315789473683	-0.22264368763915232	90.273684210526312	5 of 5


Queries [0]
Paste
New Query
New Parameter...
New Group...
Expand All
Collapse All





	Content	Name	Extension
1	Binary	Daily_1.xlsx	.xlsx
2	Binary	Daily_2.xlsx	.xlsx
3	Binary	Daily_2a.xlsx	.xlsx
4	Binary	Daily_3.xlsx	.xlsx

	Name	Data	Item	Kind
1	Daily	Table	Daily	Sheet

Column3	Column4	Column5	Column6
null	Daily by Date Performance Data - Preliminary	null	null	null
null	My Property: Business Name2	null	null	null
null	Comp Set: Location4 #444444, Location5 #555555, Location 6 #66666...	null	null	null
null	null	null	null	null
null	null	null	null	null
null	Job Number: 37165322 Staff: User1 Created: May 15, 2023 Curr...	null	null	null
null	null	null	null	null
null	null	null	null	Occupancy
null	Date	DOW	null	This Year
null	null	null	null	My Prop
null	05/22/2023	Mon	null	96,84210526
null	05/23/2023	Tue	null	97,89473684

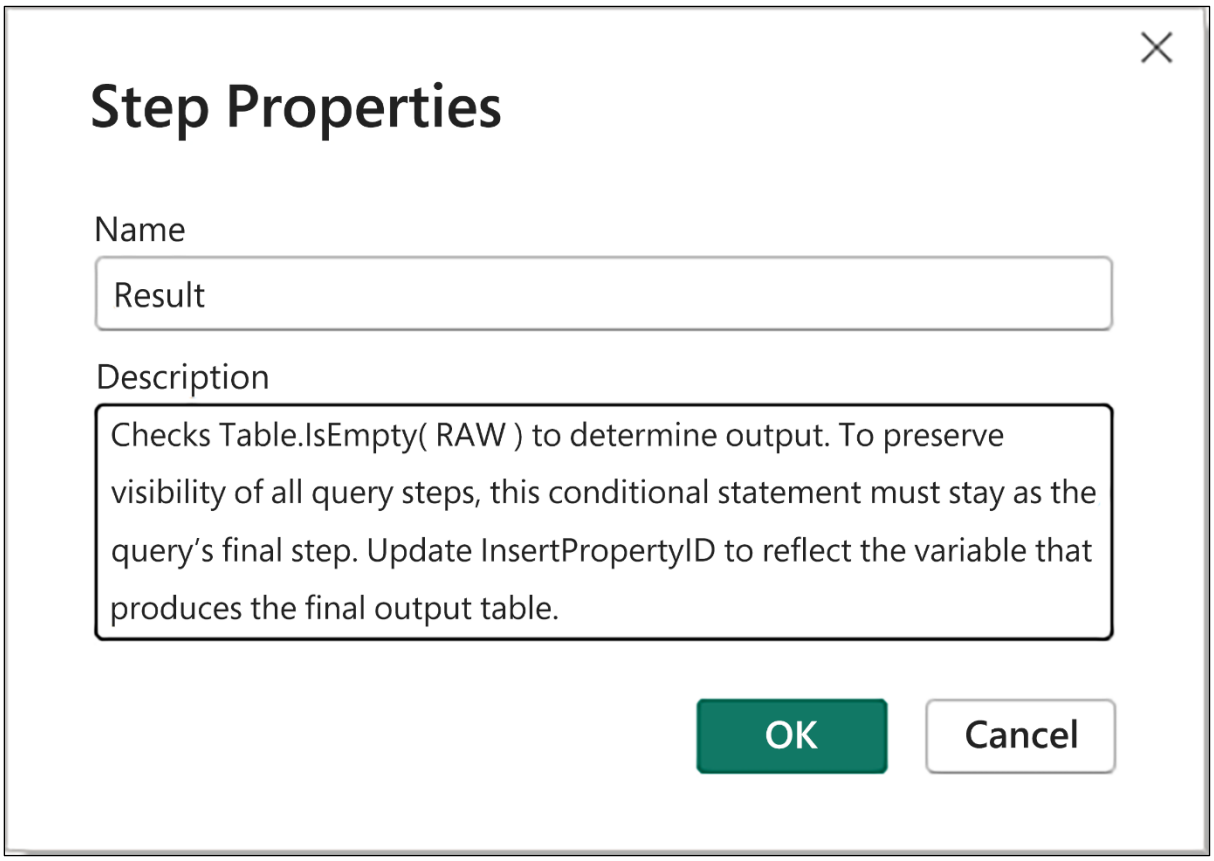
	Column3	Column4
1	Daily by Date Performance Data - Preliminary	null
2	My Property: Business Name2	null
3	Comp Set: Location4 #444444, Location5 #555555, Location 6 #66666...	null
4	Job Number: 37165322 Staff: User1 Created: May 15, 2023 Curr...	null

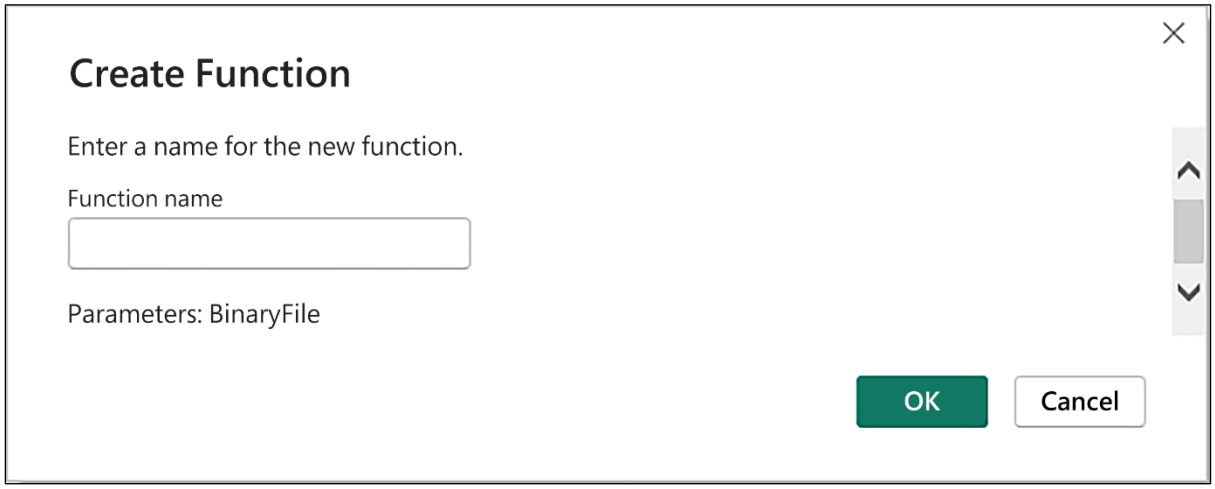
	Column3	Column4	Column6
1	null	null	Occupancy
2	Date	DOW	This Year
3	null	null	My Prop
4	05/22/2023	Mon	96,84210526
5	05/22/2023	Tue	97,89473684

	List
1	List
2	List
3	List

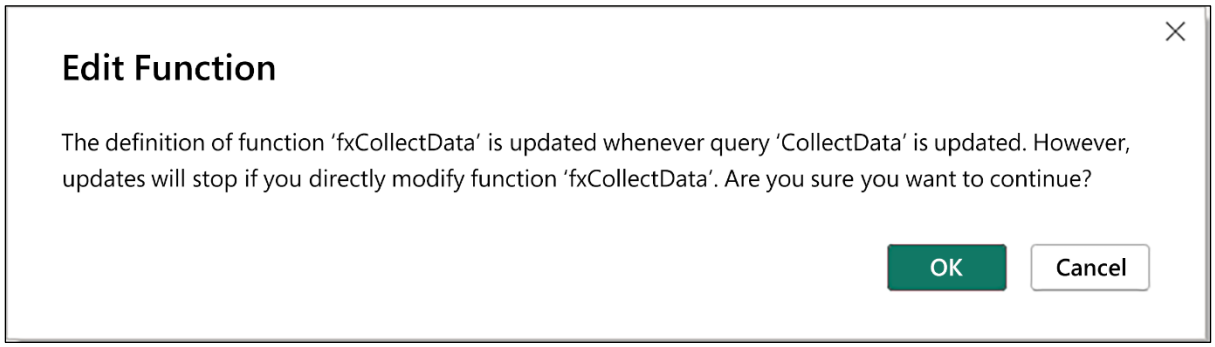
	List
1	Date
2	DOW
3	Occupancy | This Year | My Prop
4	Occupancy | This Year | Comp Set
5	Occupancy | % Chg | My Prop

```
let
    GetHeaderRows = Table.TransformRows( Table.FirstN(GetDataRange,3), each
        Table.FillDown( Record.ToTable(_), {"Value"})[Value]
    ),
    Headers = List.Transform( List.Zip( GetHeaderRows ), each Text.Combine( _, " | ")),
    GetPropertyID = [
            PropertyString = List.First( Record.ToList(NoEmptyCols{1})),
            PropertyID = List.Last( Text.Split( PropertyString, " "))
        ][PropertyID],
    fxNoEmptyCols = (tbl as table) as table => Table.SelectColumns( tbl,
        List.Select( Table.ColumnNames(tbl),
            each not List.IsEmpty ( List.RemoveMatchingItems( Table.Column( tbl, _), {null, ""})) )
        ),
    Source = Excel.Workbook( BinaryFile ),
```

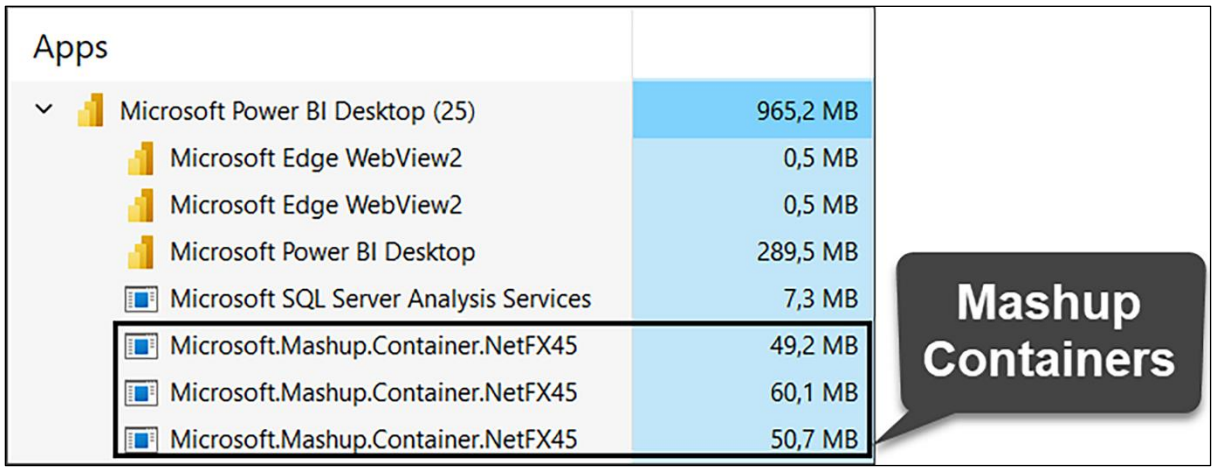


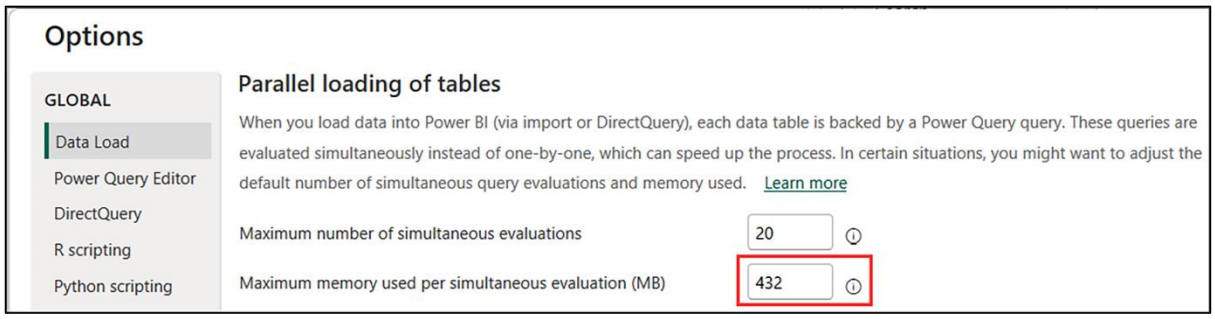


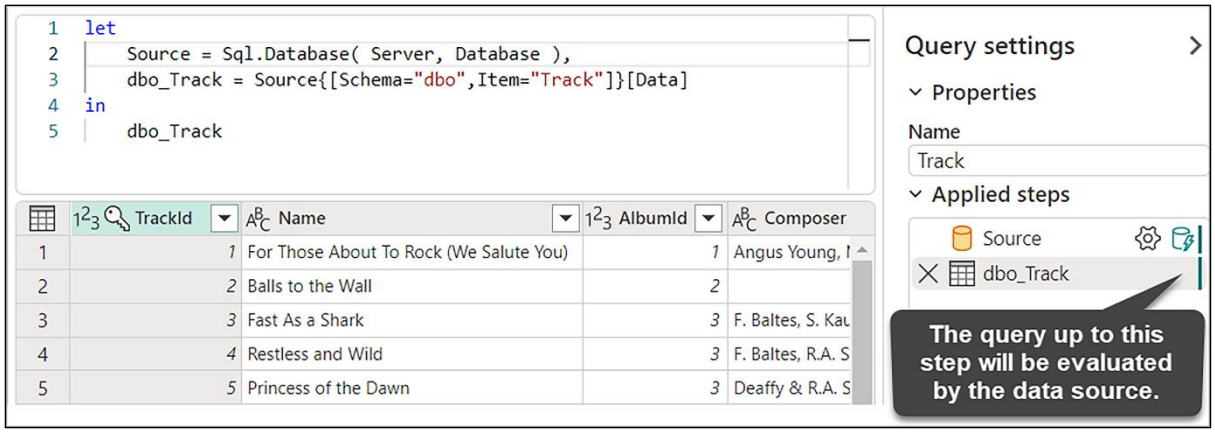
	Name	Content
1	Daily_1.xlsx	Table
2	Daily_2.xlsx	Table
3	Daily_2a.xlsx	File is empty.
4	Daily_3.xlsx	Table

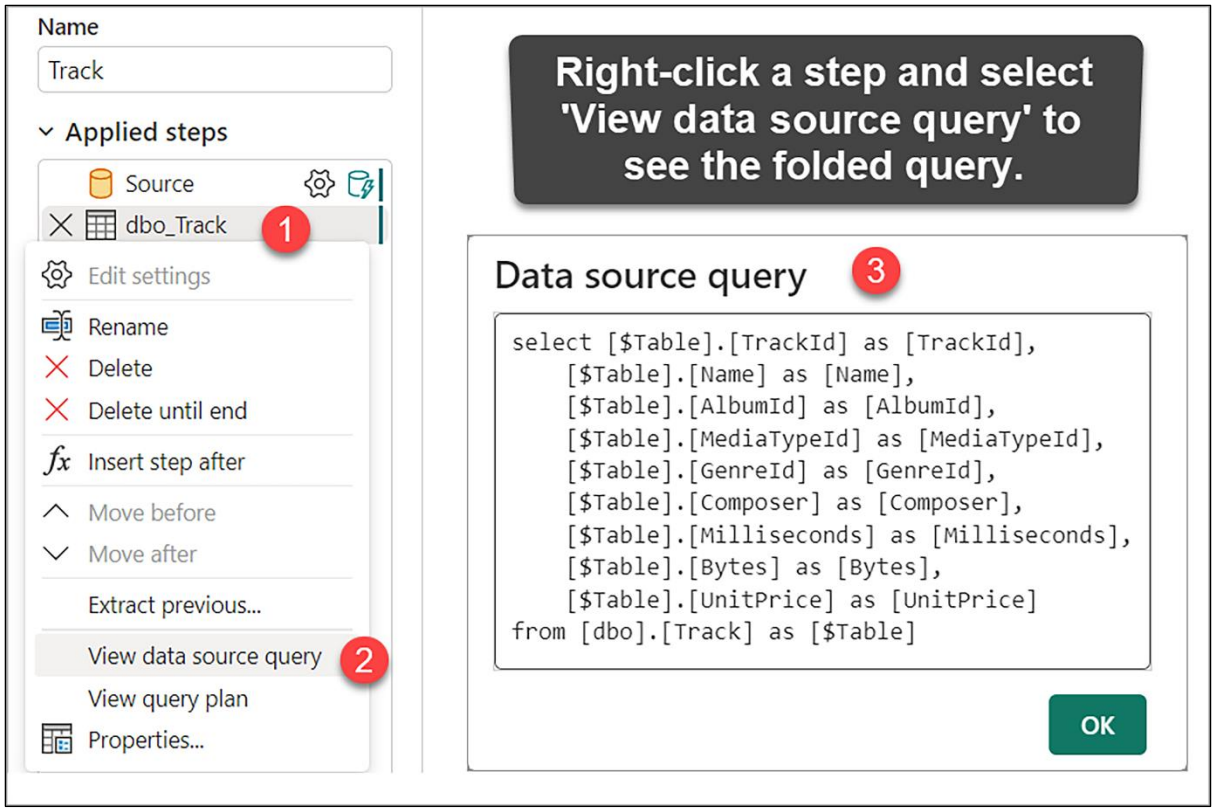


Chapter 15: Optimizing Performance








Name
Track
Applied steps
Source
dbo_Track
1
Edit settings
Rename
Delete
Delete until end
Insert step after
Move before
Move after
Extract previous...
View data source query
2
View query plan
Properties...
Right-click a step and select 'View data source query' to see the folded query.
Data source query
3
select [$Table].[TrackId] as [TrackId],
[$Table].[Name] as [Name],
[$Table].[AlbumId] as [AlbumId],
[$Table].[MediaTypeId] as [MediaTypeId],
[$Table].[GenreId] as [GenreId],
[$Table].[Composer] as [Composer],
[$Table].[Milliseconds] as [Milliseconds],
[$Table].[Bytes] as [Bytes],
[$Table].[UnitPrice] as [UnitPrice]
from [dbo].[Track] as [$Table]
OK

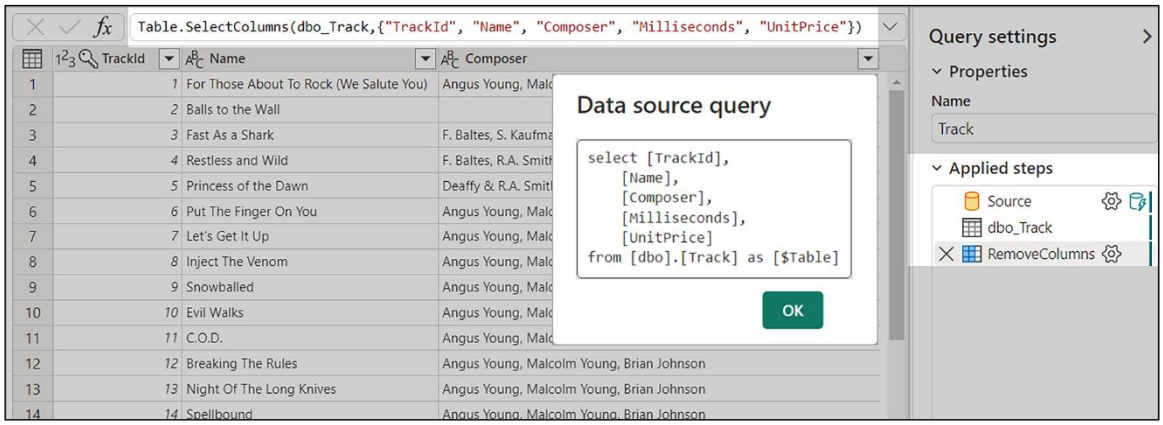
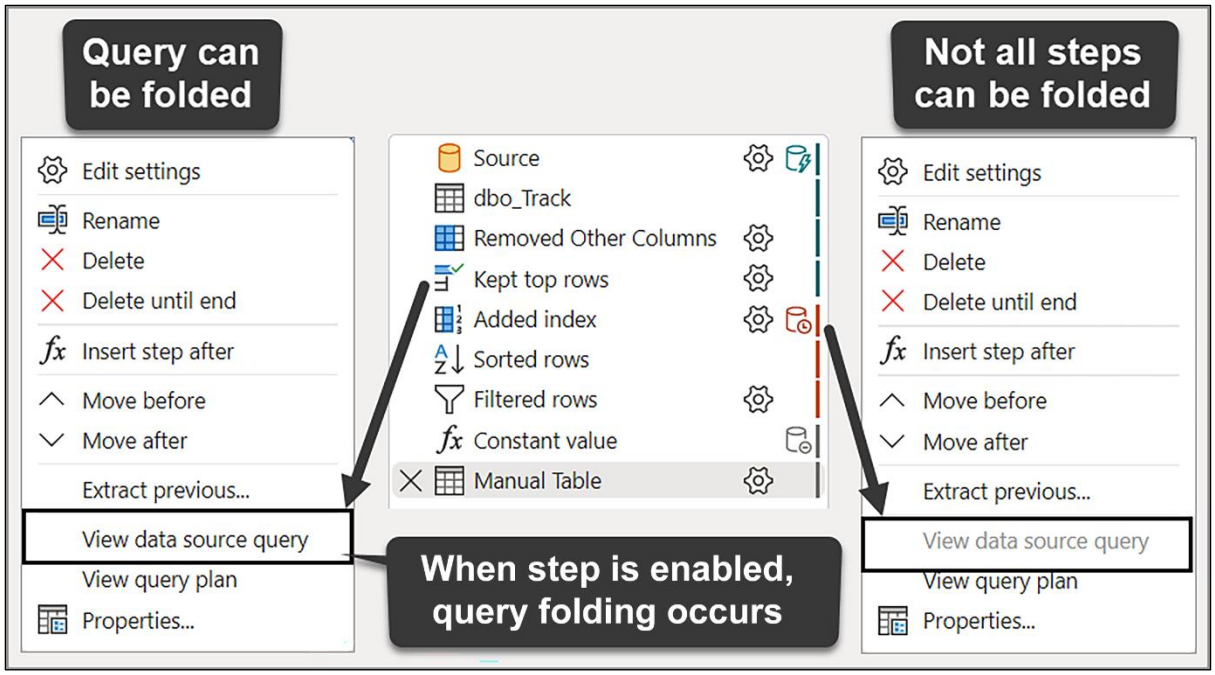
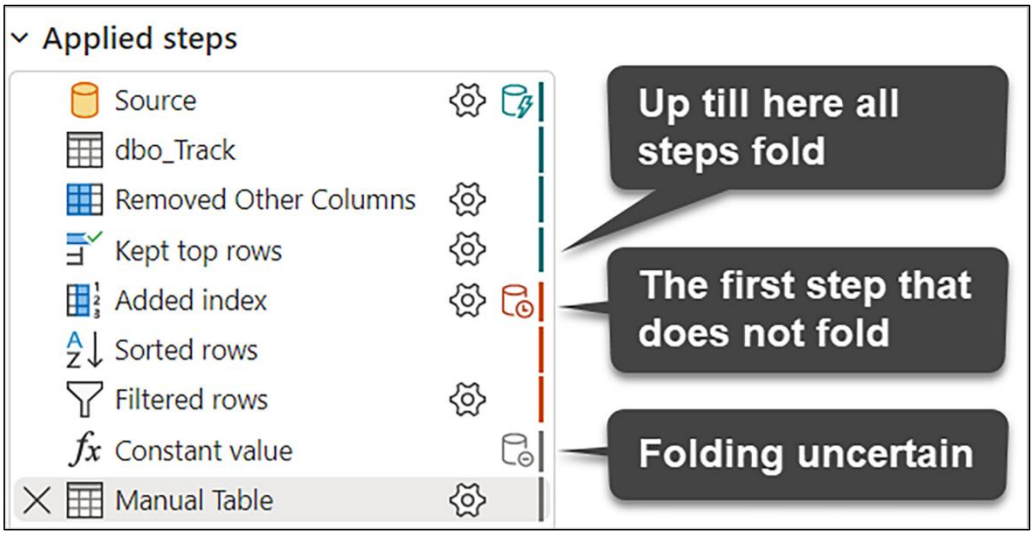


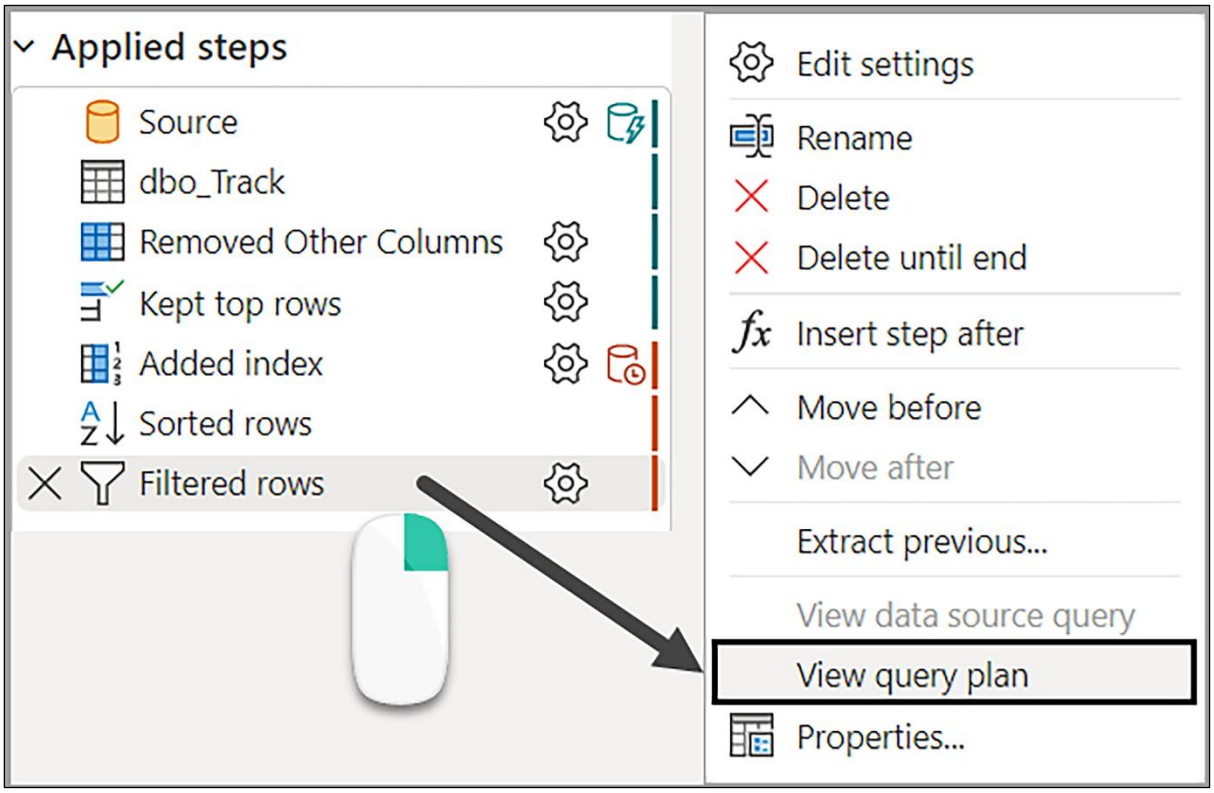
Table.SelectColumns(dbo_Track,{"TrackId", "Name", "Composer", "Milliseconds", "UnitPrice"})
TrackId
Name
Composer
1 1 For Those About To Rock (We Salute You) Angus Young, Malc
2 2 Balls to the Wall
3 3 Fast As a Shark F. Baltes, S. Kaufma
4 4 Restless and Wild F. Baltes, R.A. Smith
5 5 Princess of the Dawn Deaffy & R.A. Smith
6 6 Put The Finger On You Angus Young, Malc
7 7 Let's Get It Up Angus Young, Malc
8 8 Inject The Venom Angus Young, Malc
9 9 Snowballed Angus Young, Malc
10 10 Evil Walks Angus Young, Malc
11 11 C.O.D. Angus Young, Malc
12 12 Breaking The Rules Angus Young, Malcolm Young, Brian Johnson
13 13 Night Of The Long Knives Angus Young, Malcolm Young, Brian Johnson
14 14 Spellbound Angus Young, Malcolm Young, Brian Johnson
Data source query
select [TrackId],
[Name],
[Composer],
[Milliseconds],
[UnitPrice]
from [dbo].[Track] as [$Table]
OK
Query settings
Properties
Name
Track
Applied steps
Source
dbo_Track
RemoveColumns

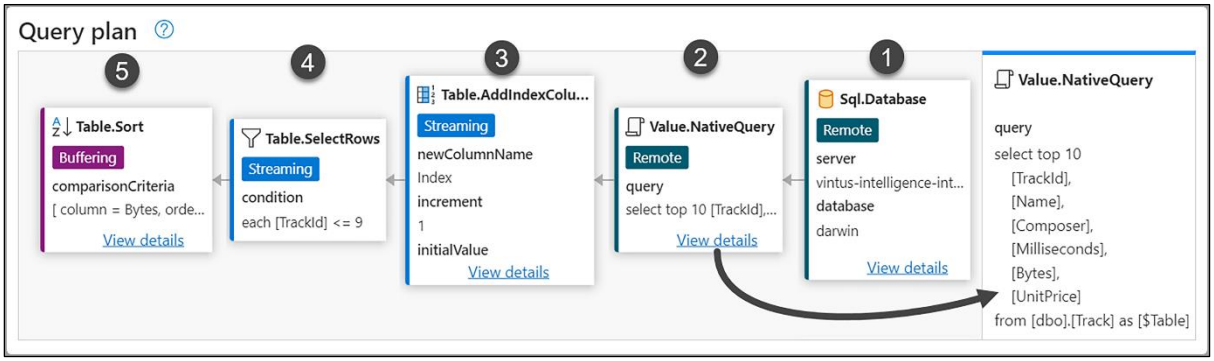


Indicator	Icon	Description
Folding		Indicates that this part of the query will be processed by the data source.
Not Folding		Indicates that this step will be processed outside the data source.
Might Fold		Whether or not a query step will be processed by the data source is uncertain and will be determined during query execution. Likely happens for ODBC or Odata connections.
Uncertain		Indicates an uncertain query plan, often due to providing a manual table or using transformations/connectors unsupported by the query plan tool and indicators.
Unknown		Indicates that there is no query plan available, which could be due to an error or because the query involves data formats other than tables.

** When an applied step in a query displays a specific folding indicator, any subsequent steps that have a vertical line with the same color as this indicator share the same query folding status.*


Applied steps
Source
dbo_Track
Removed Other Columns
Kept top rows
Added index
Sorted rows
Filtered rows
Constant value
Manual Table
Up till here all steps fold
The first step that does not fold
Folding uncertain


Applied steps
Source
dbo_Track
Removed Other Columns
Kept top rows
Added index
Sorted rows
Filtered rows
Edit settings
Rename
Delete
Delete until end
Insert step after
Move before
Move after
Extract previous...
View data source query
View query plan
Properties...


Query plan
5
Table.Sort
Buffering
comparisonCriteria
[column = Bytes, orde...
View details
4
Table.SelectRows
Streaming
condition
each [TrackId] <= 9
3
Table.AddIndexColu...
Streaming
newColumnName
Index
increment
1
initialValue
View details
2
Value.NativeQuery
Remote
query
select top 10 [TrackId],...
View details
1
Sql.Database
Remote
server
vintus-intelligence-int...
database
darwin
View details
Value.NativeQuery
query
select top 10
[TrackId],
[Name],
[Composer],
[Milliseconds],
[Bytes],
[UnitPrice]
from [dbo].[Track] as [$Table]

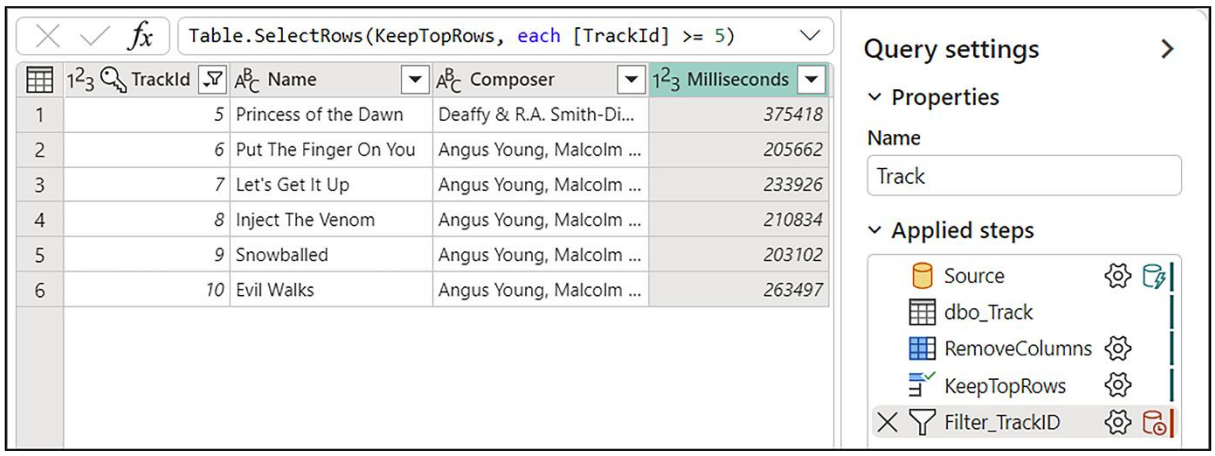


Table.SelectRows(KeepTopRows, each [TrackId] >= 5)
TrackId
Name
Composer
Milliseconds
1 5 Princess of the Dawn Deaffy & R.A. Smith-Di... 375418
2 6 Put The Finger On You Angus Young, Malcolm ... 205662
3 7 Let's Get It Up Angus Young, Malcolm ... 233926
4 8 Inject The Venom Angus Young, Malcolm ... 210834
5 9 Snowballed Angus Young, Malcolm ... 203102
6 10 Evil Walks Angus Young, Malcolm ... 263497
Query settings
Properties
Name
Track
Applied steps
Source
dbo_Track
RemoveColumns
KeepTopRows
Filter_TrackID

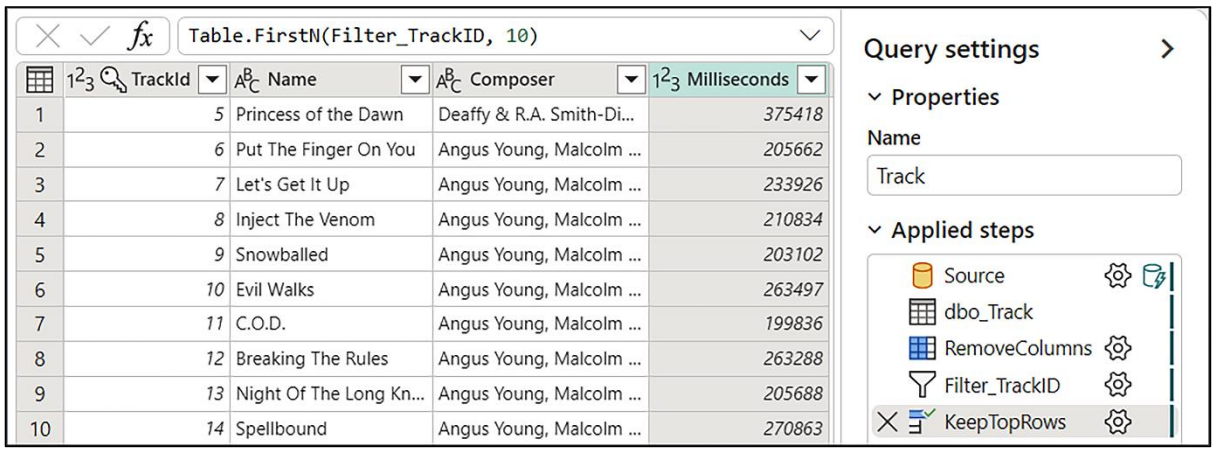
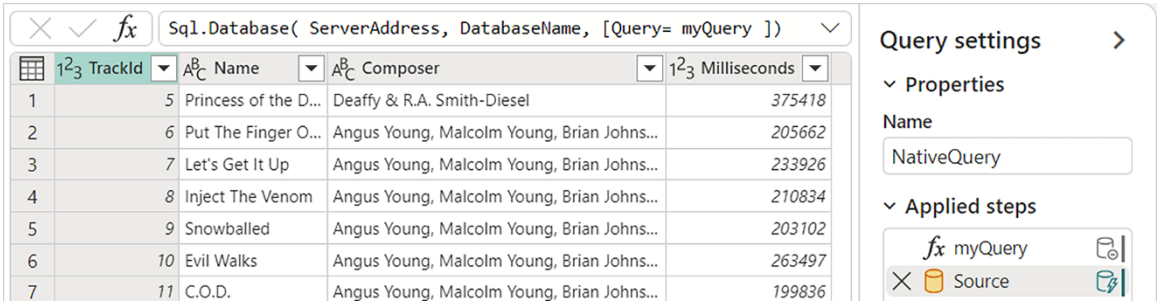
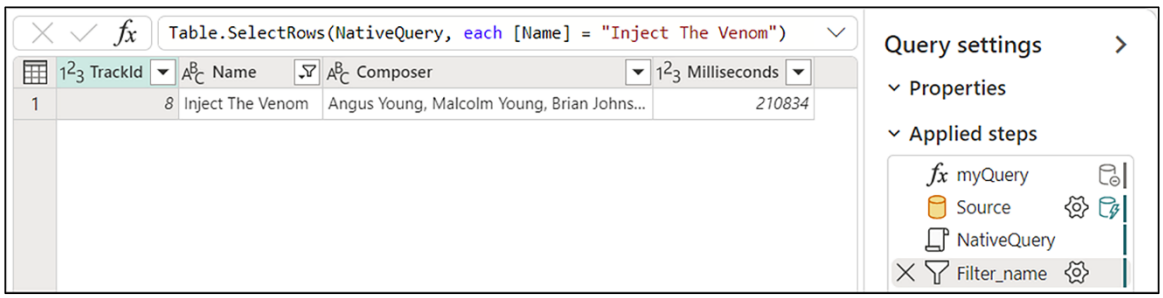


Table.FirstN(Filter_TrackID, 10)
TrackId
Name
Composer
Milliseconds
1 5 Princess of the Dawn Deaffy & R.A. Smith-Di... 375418
2 6 Put The Finger On You Angus Young, Malcolm ... 205662
3 7 Let's Get It Up Angus Young, Malcolm ... 233926
4 8 Inject The Venom Angus Young, Malcolm ... 210834
5 9 Snowballed Angus Young, Malcolm ... 203102
6 10 Evil Walks Angus Young, Malcolm ... 263497
7 11 C.O.D. Angus Young, Malcolm ... 199836
8 12 Breaking The Rules Angus Young, Malcolm ... 263288
9 13 Night Of The Long Kn... Angus Young, Malcolm ... 205688
10 14 Spellbound Angus Young, Malcolm ... 270863
Query settings
Properties
Name
Track
Applied steps
Source
dbo_Track
RemoveColumns
Filter_TrackID
KeepTopRows

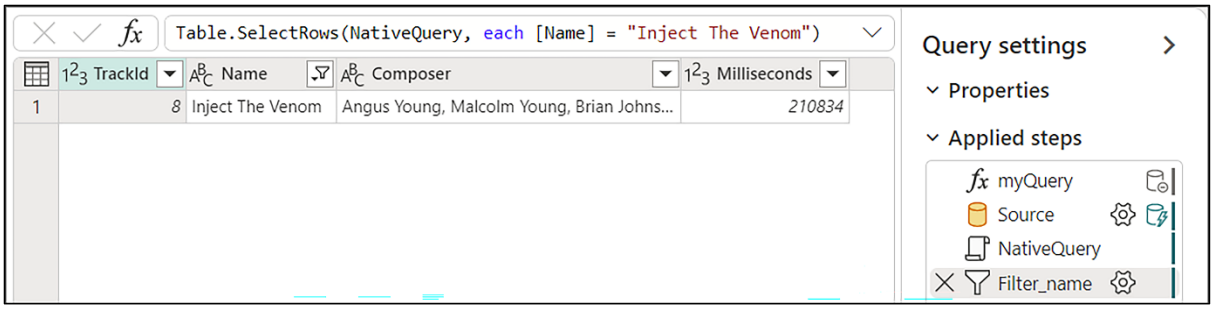
Data source query

```
select top 10
    [_].[TrackId],
    [_].[Name],
    [_].[Composer],
    [_].[Milliseconds]
from
(
    select [TrackId],
        [Name],
        [Composer],
        [Milliseconds]
    from [dbo].[Track] as [$Table]
) as [_]
where [_].[TrackId] >= 5
```





```
let
    myQuery =
    "SELECT TOP (10)
       [TrackId], [Name], [Composer] ,[Milliseconds]
     FROM [dbo].[Track]
     WHERE [TrackId] >= 5",
    Source = Sql.Database( ServerAddress, DatabaseName ),
    NativeQuery = Value.NativeQuery( Source, myQuery, null, [EnableFolding = true] )
in
    NativeQuery
```

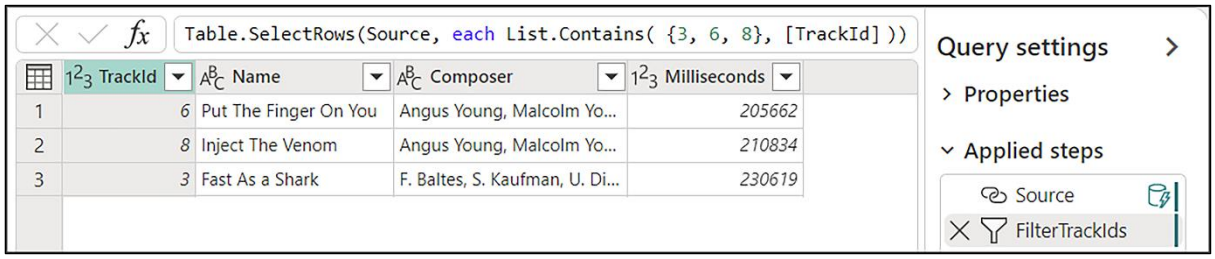


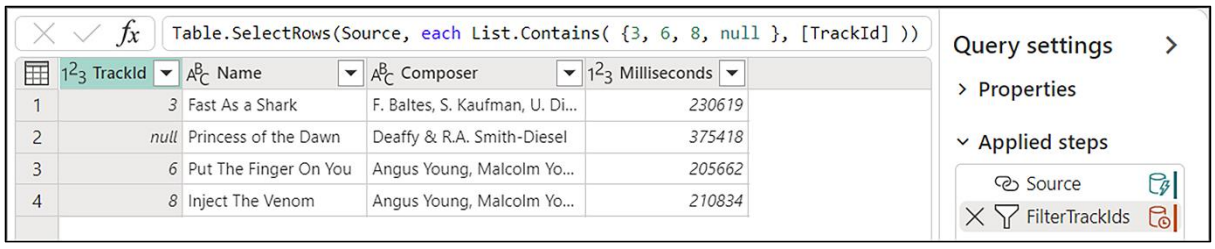
Data source query

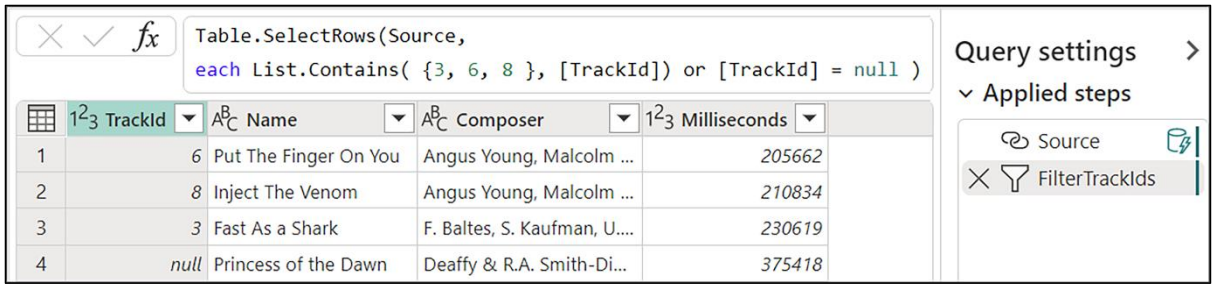
```
select [_].[TrackId],
    [_].[Name],
    [_].[Composer],
    [_].[Milliseconds]
from
(
    SELECT TOP (10)
       [TrackId], [Name], [Composer] ,[Milliseconds]
     FROM [dbo].[Track]
     WHERE [TrackId] >= 5
) as [_]
where [_].[Name] = 'Inject The Venom'
```

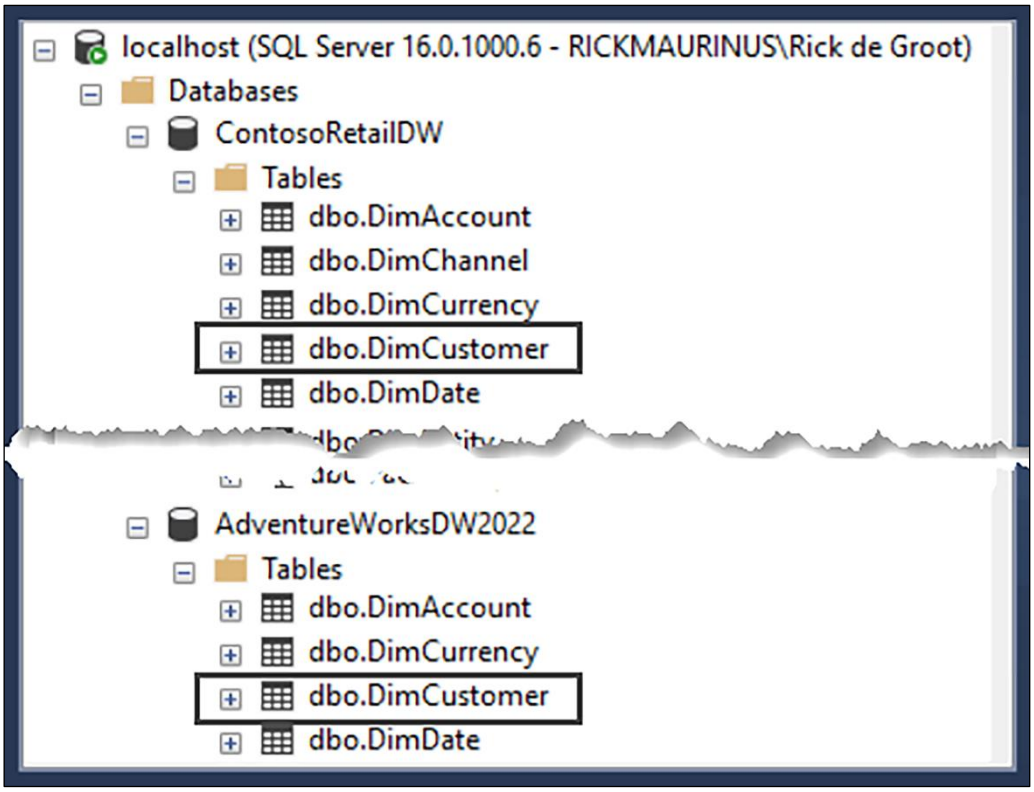
Custom SQL query

	TrackId	Name	Composer	Milliseconds
1	1	For Those About ...	Angus Young, Malc...	343719
2	2	Balls to the Wall	null	342562
3	3	Fast As a Shark	F. Baltes, S. Kaufma...	230619
4	4	Restless and Wild	F. Baltes, R.A. Smit...	252051
5	null	Princess of the D...	Deaffy & R.A. Smit...	375418
6	6	Put The Finger O...	Angus Young, Malc...	205662
7	7	Let's Get It Up	Angus Young, Malc...	233926
8	8	Inject The Venom	Angus Young, Malc...	210834







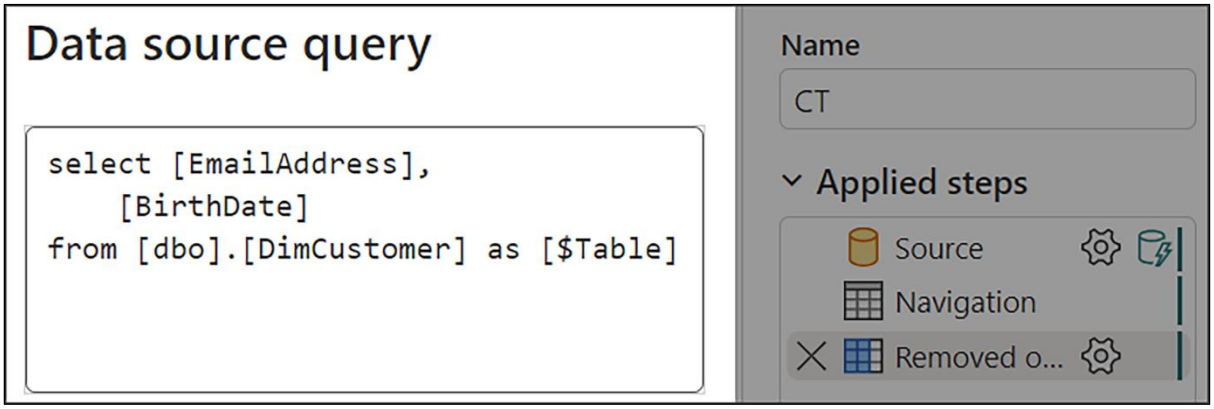


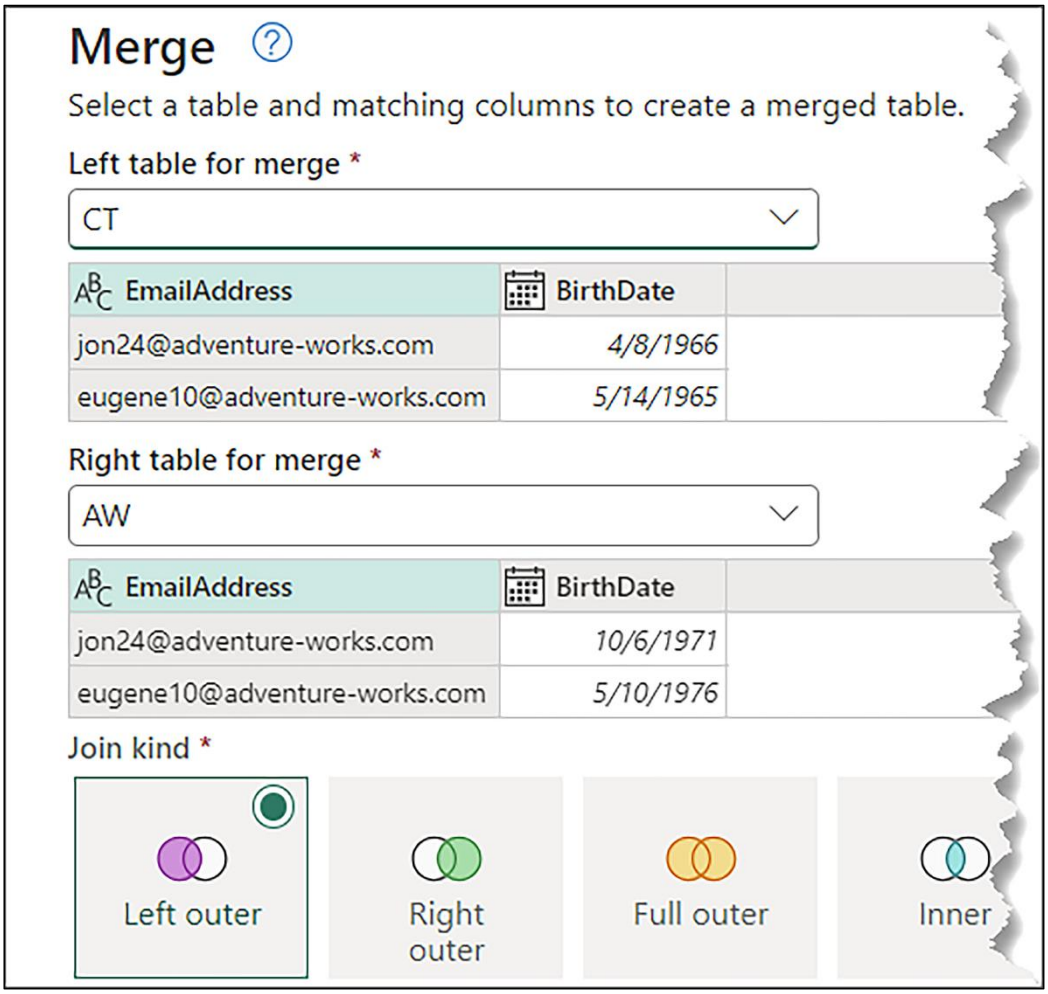
AW

```
let
  Source = Sql.Database("localhost", "ContosoRetailDW"),
  Navigation = Source{[Schema = "dbo", Item = "DimCustomer"]}[Data],
  #"Removed other columns" = Table.SelectColumns(Navigation, {"EmailAddress", "BirthDate"})
in
  #"Removed other columns"
```

CT

```
let
  Source = Sql.Database("localhost", "AdventureWorksDW2022"),
  Navigation = Source{[Item = "DimCustomer", Schema = "dbo"]}[Data],
  #"Removed other columns" = Table.SelectColumns(Navigation, {"EmailAddress", "BirthDate"})
in
  #"Removed other columns"
```




Merge
Select a table and matching columns to create a merged table.
Left table for merge *
CT
EmailAddress
BirthDate
jon24@adventure-works.com
4/8/1966
eugene10@adventure-works.com
5/14/1965
Right table for merge *
AW
EmailAddress
BirthDate
jon24@adventure-works.com
10/6/1971
eugene10@adventure-works.com
5/10/1976
Join kind *
Left outer
Right outer
Full outer
Inner

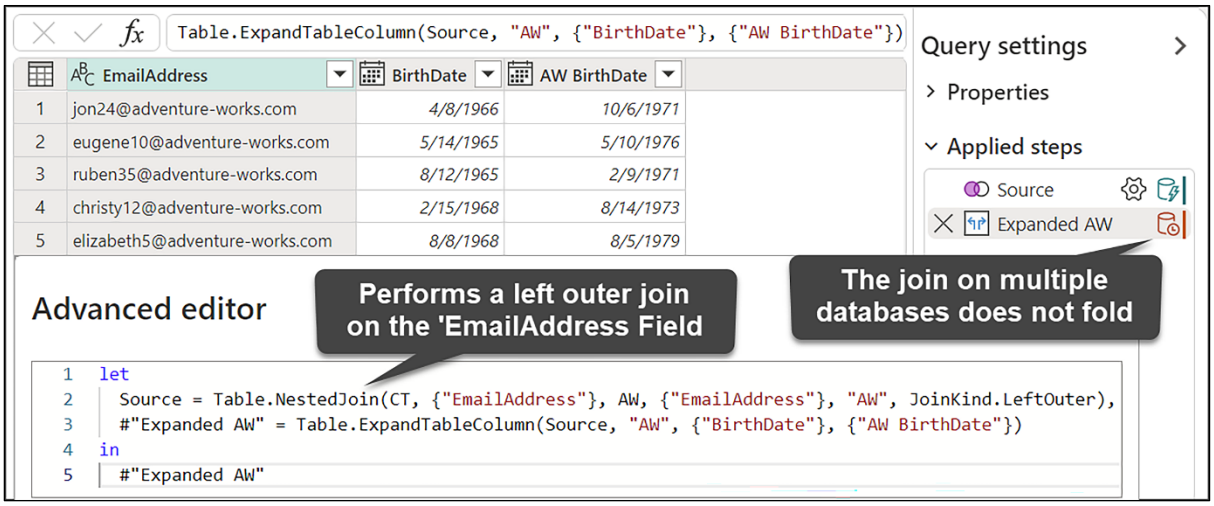


Table.ExpandTableColumn(Source, "AW", {"BirthDate"}, {"AW BirthDate"})
EmailAddress
BirthDate
AW BirthDate
1 jon24@adventure-works.com 4/8/1966 10/6/1971
2 eugene10@adventure-works.com 5/14/1965 5/10/1976
3 ruben35@adventure-works.com 8/12/1965 2/9/1971
4 christy12@adventure-works.com 2/15/1968 8/14/1973
5 elizabeth5@adventure-works.com 8/8/1968 8/5/1979
Query settings
Properties
Applied steps
Source
Expanded AW
Advanced editor
Performs a left outer join on the 'EmailAddress Field
The join on multiple databases does not fold
let
Source = Table.NestedJoin(CT, {"EmailAddress"}, AW, {"EmailAddress"}, "AW", JoinKind.LeftOuter),
#"Expanded AW" = Table.ExpandTableColumn(Source, "AW", {"BirthDate"}, {"AW BirthDate"})
in
#"Expanded AW"

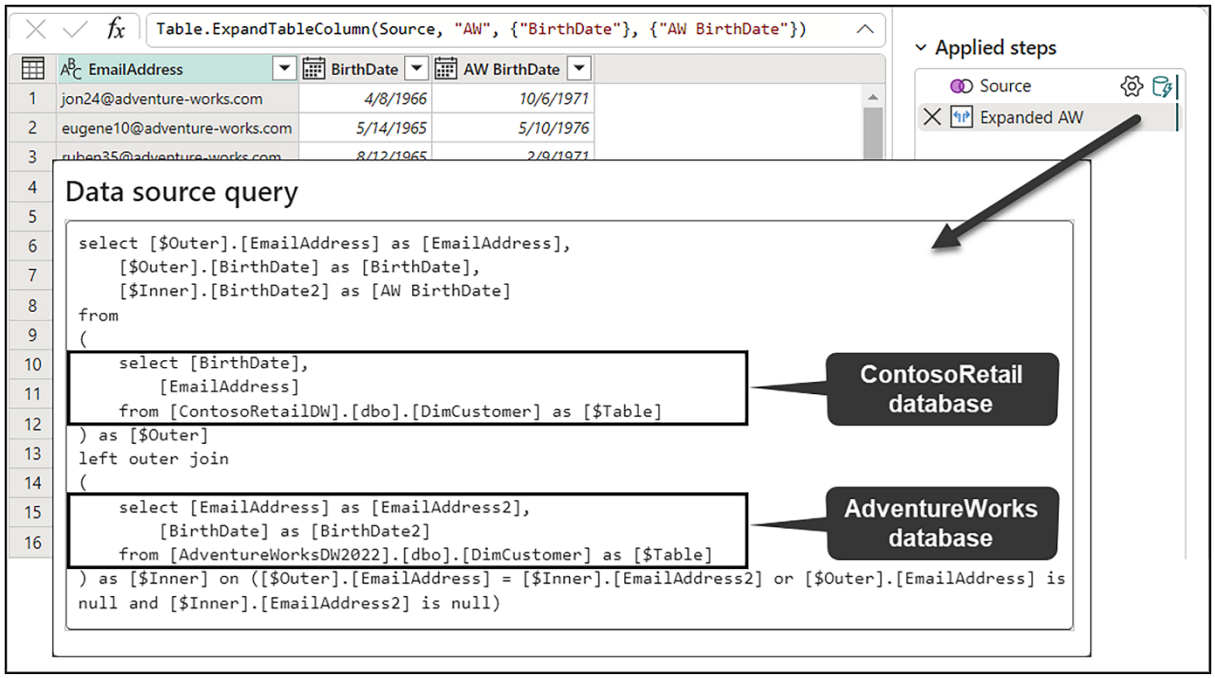
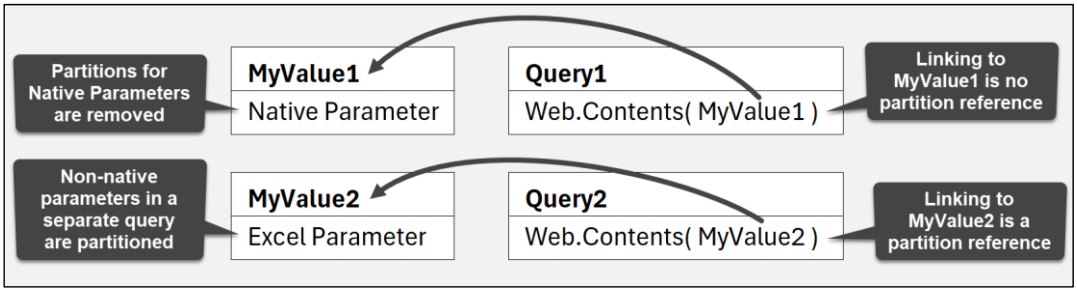
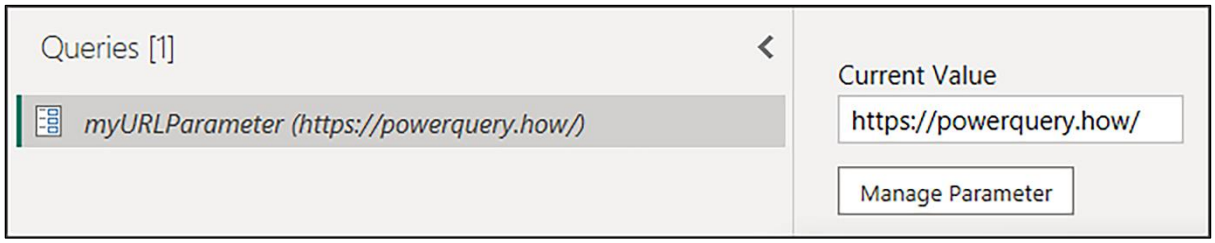
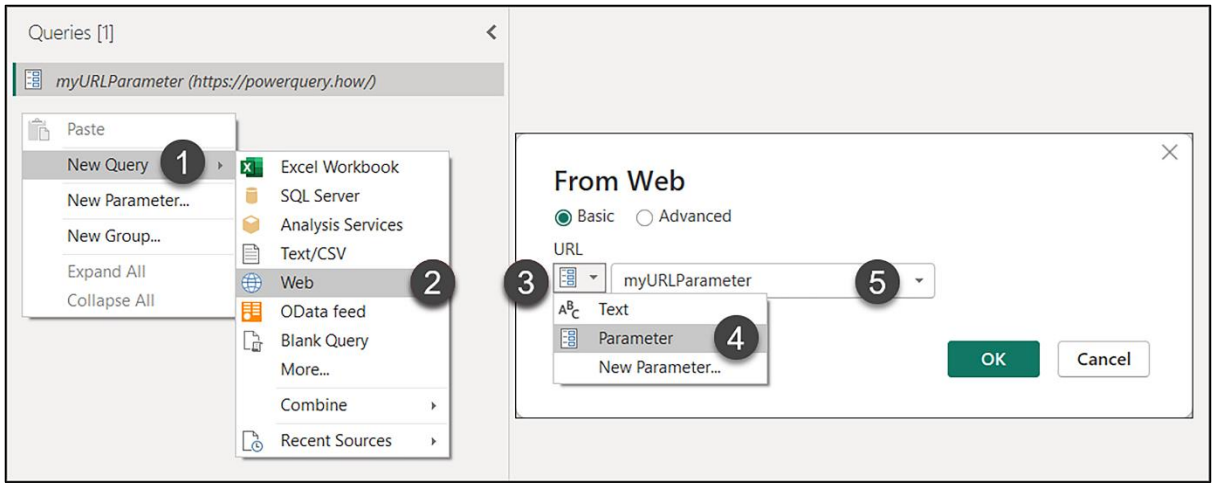
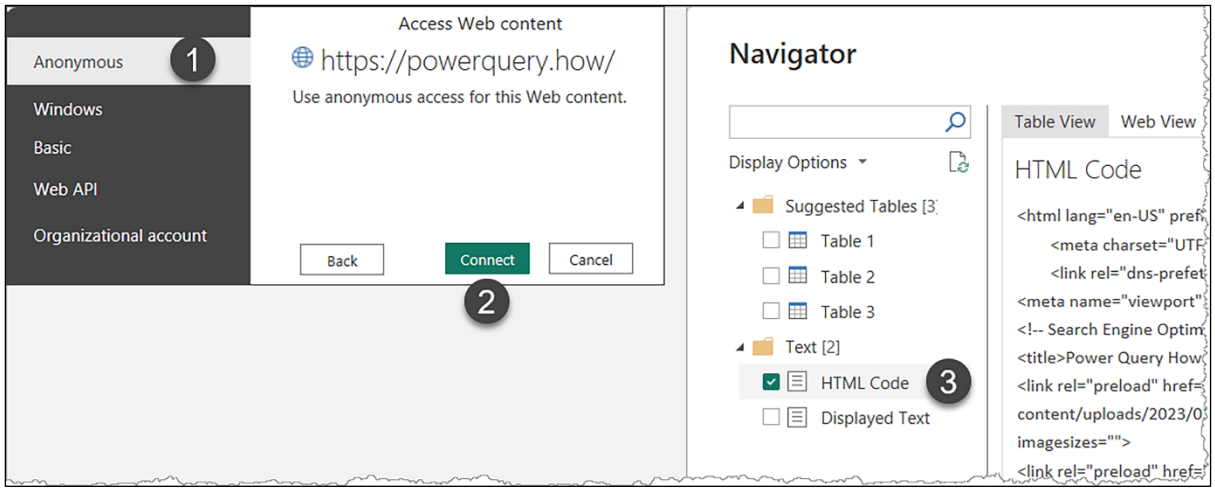


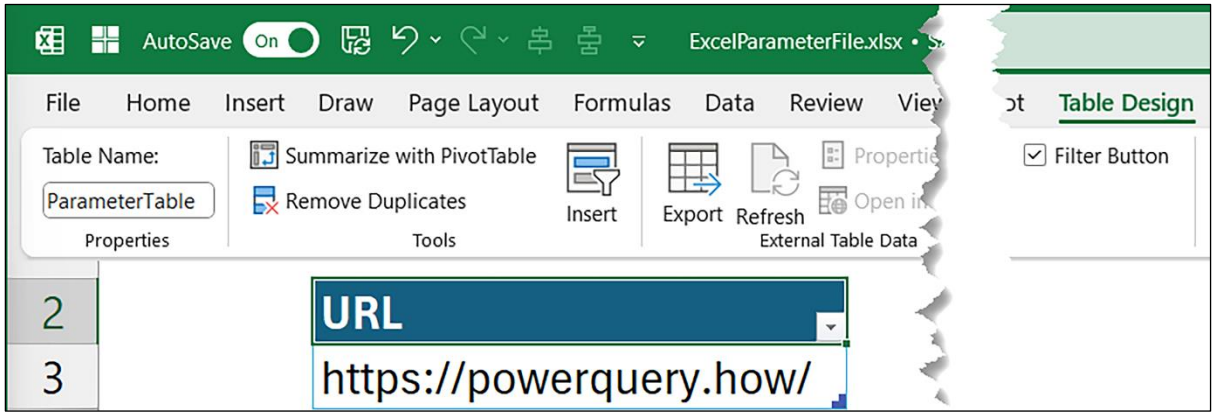
Table.ExpandTableColumn(Source, "AW", {"BirthDate"}, {"AW BirthDate"})
EmailAddress
BirthDate
AW BirthDate
1 jon24@adventure-works.com 4/8/1966 10/6/1971
2 eugene10@adventure-works.com 5/14/1965 5/10/1976
Applied steps
Source
Expanded AW
Data source query
select [$Outer].[EmailAddress] as [EmailAddress],
[$Outer].[BirthDate] as [BirthDate],
[$Inner].[BirthDate2] as [AW BirthDate]
from
(
select [BirthDate],
[EmailAddress]
from [ContosoRetailDW].[dbo].[DimCustomer] as [$Table]
) as [$Outer]
left outer join
(
select [EmailAddress] as [EmailAddress2],
[BirthDate] as [BirthDate2]
from [AdventureWorksDW2022].[dbo].[DimCustomer] as [$Table]
) as [$Inner] on ([$Outer].[EmailAddress] = [$Inner].[EmailAddress2] or [$Outer].[EmailAddress] is null and [$Inner].[EmailAddress2] is null)
ContosoRetail database
AdventureWorks database

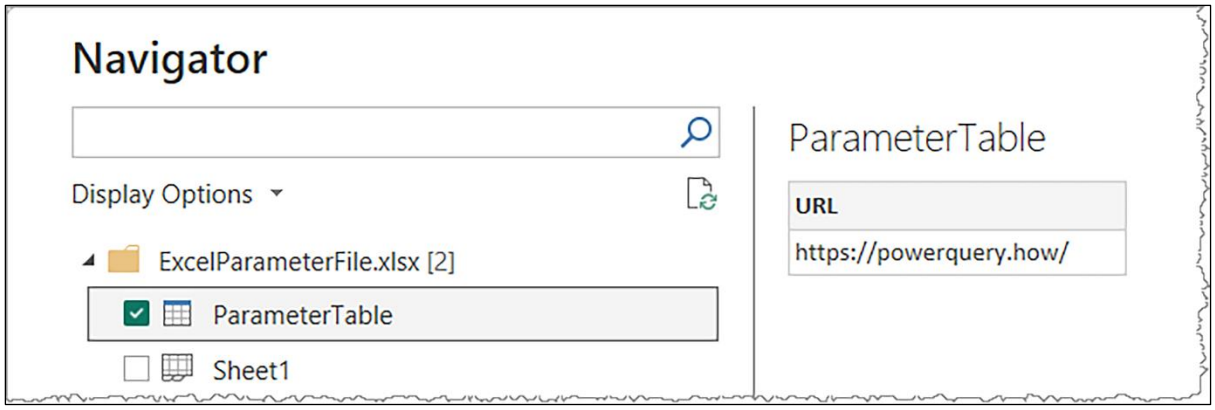

Partitions for Native Parameters are removed
MyValue1
Native Parameter
Query1
Web.Contents(MyValue1)
Linking to MyValue1 is no partition reference
Non-native parameters in a separate query are partitioned
MyValue2
Excel Parameter
Query2
Web.Contents(MyValue2)
Linking to MyValue2 is a partition reference

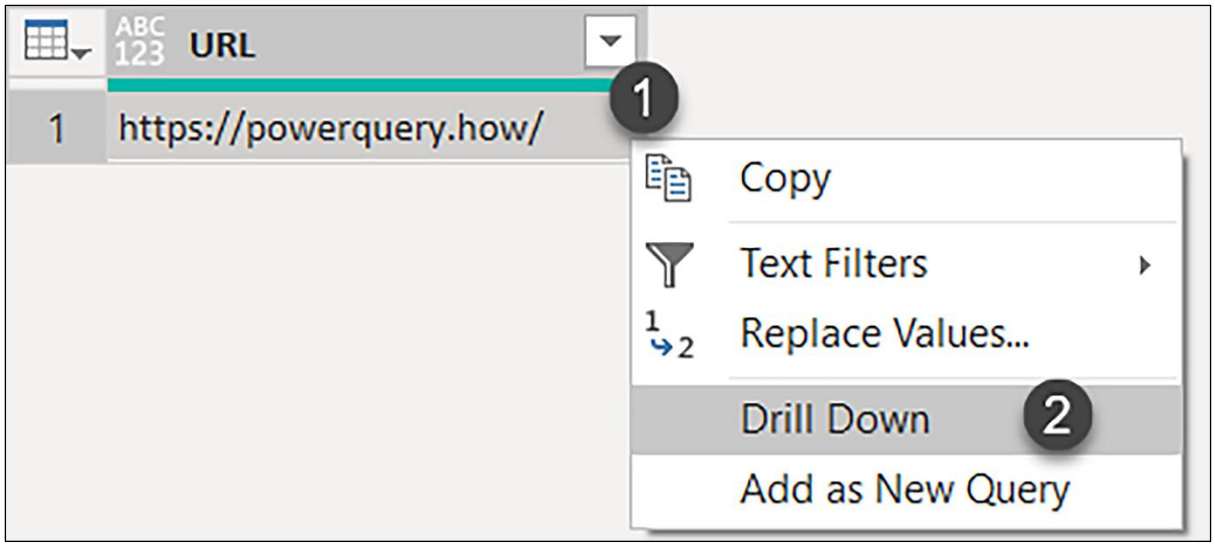

Queries [1]
myURLParameter (https://powerquery.how/)
Current Value
https://powerquery.how/
Manage Parameter

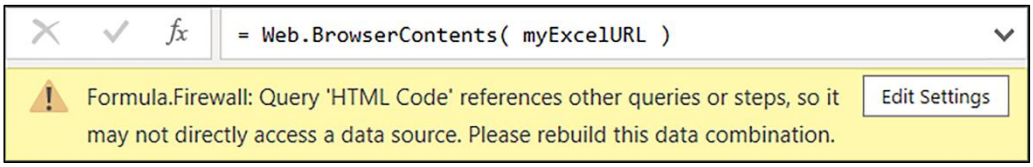

Queries [1]
myURLParameter (https://powerquery.how/)
Paste
New Query
New Parameter...
New Group...
Expand All
Collapse All
Excel Workbook
SQL Server
Analysis Services
Text/CSV
Web
OData feed
Blank Query
More...
Combine
Recent Sources
1
2
From Web
Basic
Advanced
URL
myURLParameter
Text
Parameter
New Parameter...
3
4
5
OK
Cancel

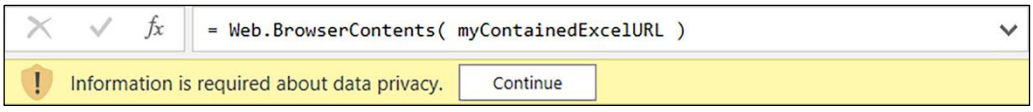

Anonymous
Windows
Basic
Web API
Organizational account
1
Access Web content
https://powerquery.how/
Use anonymous access for this Web content.
Back
Connect
Cancel
2
Navigator
Display Options
Suggested Tables [3
Table 1
Table 2
Table 3
Text [2]
HTML Code
Displayed Text
3
Table View
Web View
HTML Code
<html lang="en-US" pref
<meta charset="UTF
<link rel="dns-prefet
<meta name="viewport"
<!-- Search Engine Optim
<title>Power Query How
<link rel="preload" href=
content/uploads/2023/0
imagesizes="">
<link rel="preload" href=

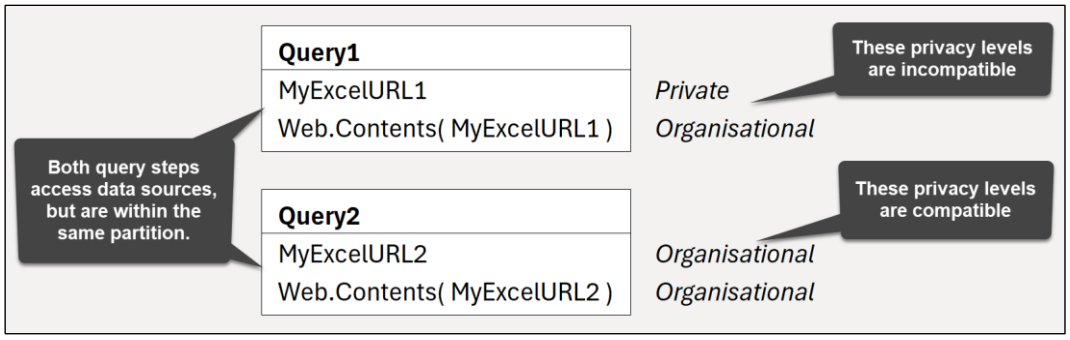

AutoSave
On
ExcelParameterFile.xlsx
File
Home
Insert
Draw
Page Layout
Formulas
Data
Review
Table Design
Table Name:
ParameterTable
Properties
Summarize with PivotTable
Remove Duplicates
Tools
Insert
Export
Refresh
External Table Data
Filter Button
2
3
URL
https://powerquery.how/

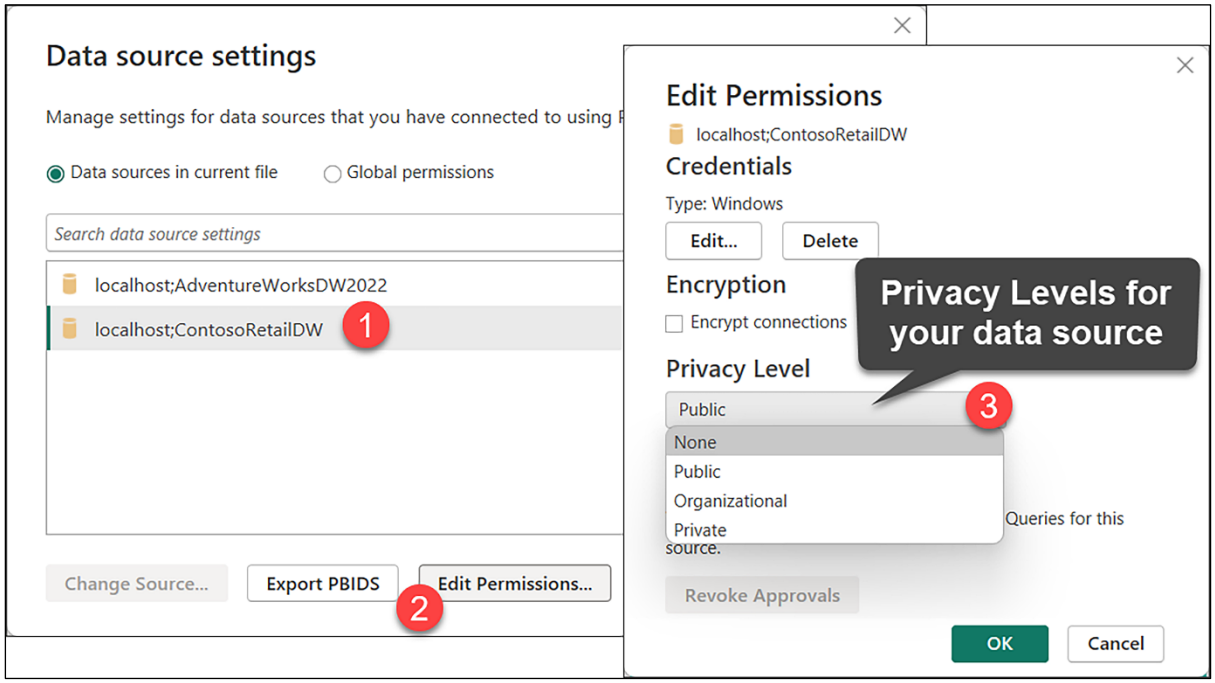

Navigator
Display Options
ExcelParameterFile.xlsx [2]
ParameterTable
Sheet1
ParameterTable
URL
https://powerquery.how/

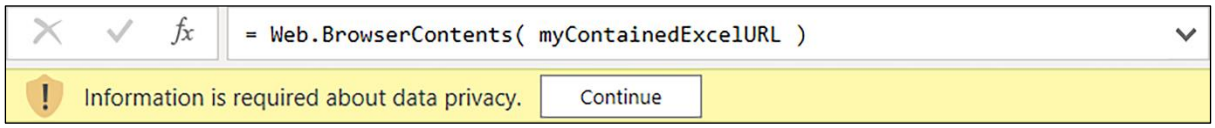

URL
1
https://powerquery.how/
Copy
Text Filters
Replace Values...
Drill Down
2
Add as New Query

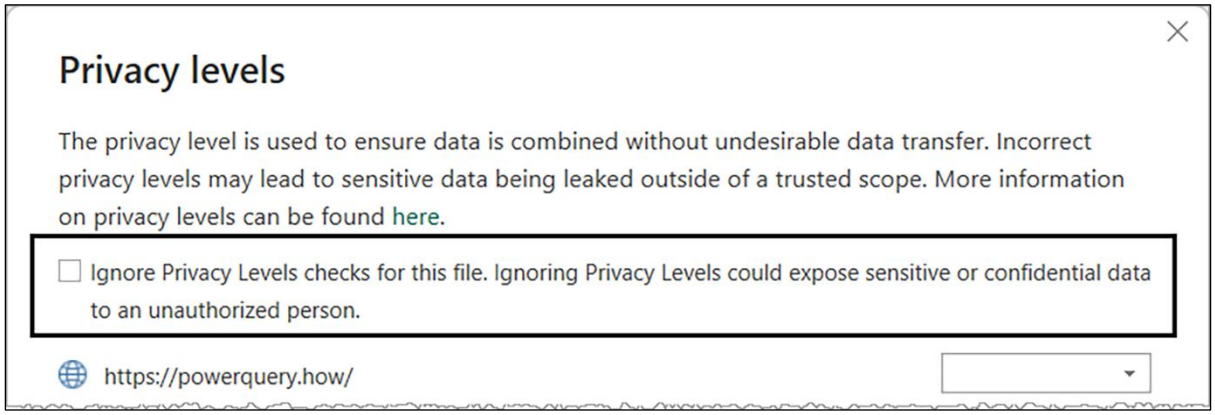

= Web.BrowserContents(myExcelURL)
Formula.Firewall: Query 'HTML Code' references other queries or steps, so it may not directly access a data source. Please rebuild this data combination.
Edit Settings

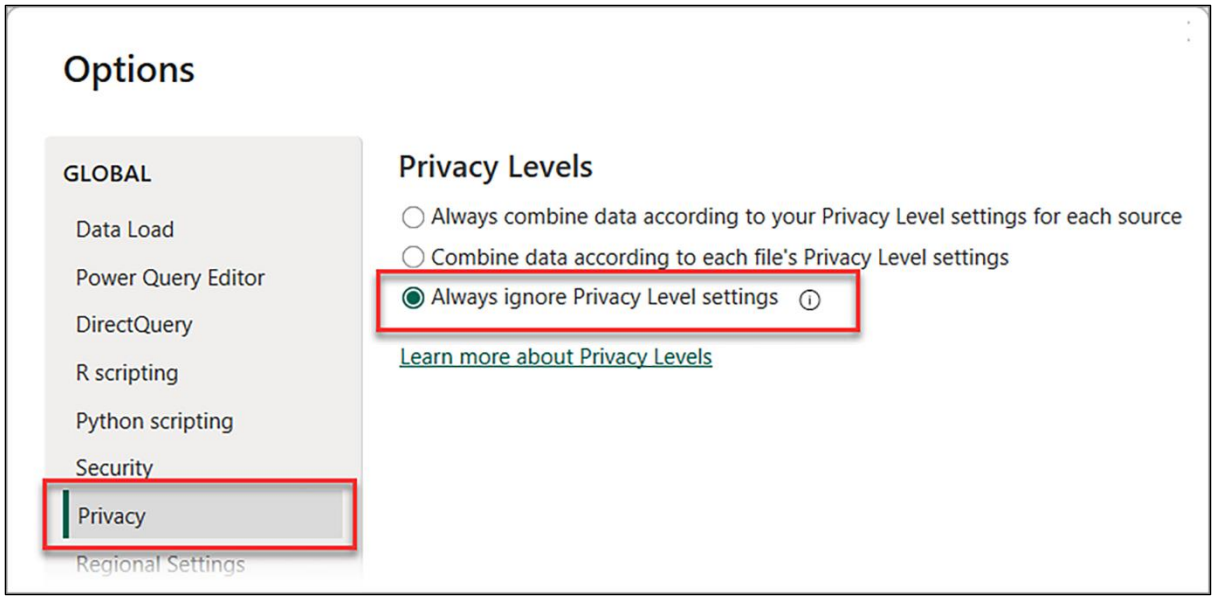

= Web.BrowserContents(myContainedExcelURL)
Information is required about data privacy.
Continue


Query1
MyExcelURL1
Web.Contents(MyExcelURL1)
Private
Organisational
These privacy levels are incompatible
Both query steps access data sources, but are within the same partition.
Query2
MyExcelURL2
Web.Contents(MyExcelURL2)
Organisational
Organisational
These privacy levels are compatible

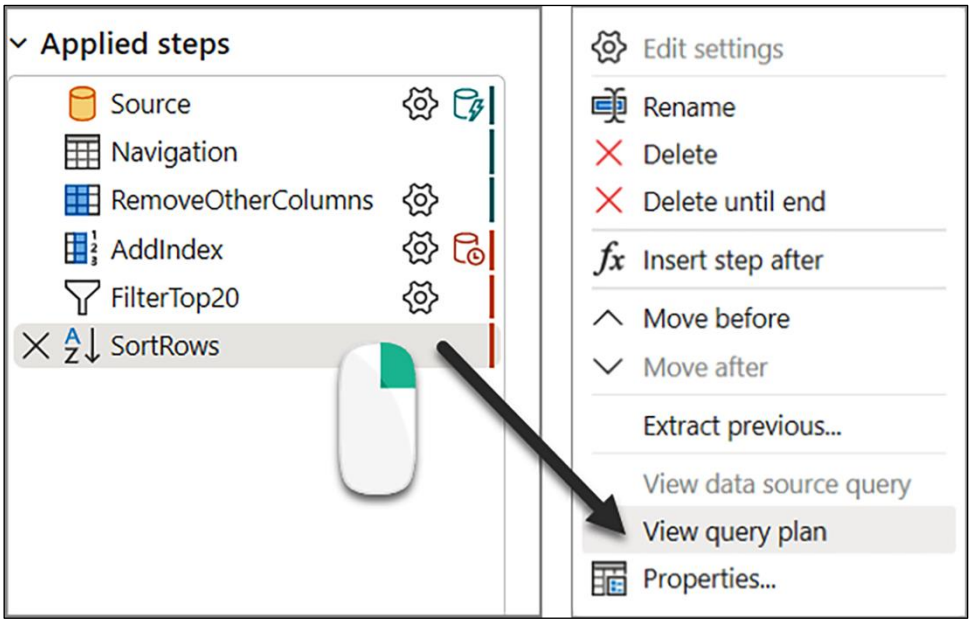

Data source settings
Manage settings for data sources that you have connected to using
Data sources in current file
Global permissions
Search data source settings
localhost;AdventureWorksDW2022
localhost;ContosoRetailDW
1
Change Source...
Export PBIDS
Edit Permissions...
2
Edit Permissions
localhost;ContosoRetailDW
Credentials
Type: Windows
Edit...
Delete
Encryption
Encrypt connections
Privacy Levels for your data source
Privacy Level
Public
3
None
Public
Organizational
Private
Queries for this source.
Revoke Approvals
OK
Cancel

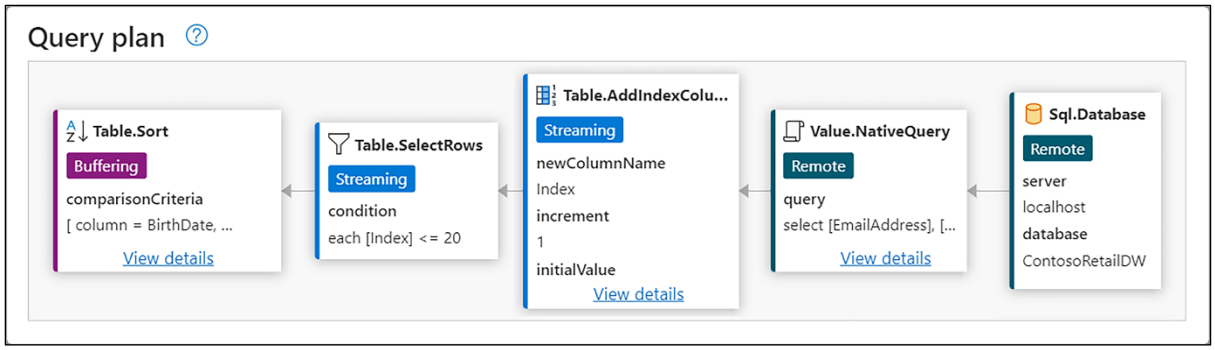

= Web.BrowserContents(myContainedExcelURL)
Information is required about data privacy.
Continue


Privacy levels
The privacy level is used to ensure data is combined without undesirable data transfer. Incorrect privacy levels may lead to sensitive data being leaked outside of a trusted scope. More information on privacy levels can be found here.
Ignore Privacy Levels checks for this file. Ignoring Privacy Levels could expose sensitive or confidential data to an unauthorized person.
https://powerquery.how/




```
let
  Source = Sql.Database("localhost", "ContosoRetailDW"),
  Navigation = Source{[Schema = "dbo", Item = "DimCustomer"]}[Data],
  RemoveOtherColumns = Table.SelectColumns(Navigation, {"EmailAddress", "BirthDate"}),
  AddIndex = Table.AddIndexColumn(RemoveOtherColumns, "Index", 0, 1, Int64.Type),
  FilterTop20 = Table.SelectRows(AddIndex, each [Index] <= 20),
  SortRows = Table.Sort(FilterTop20, {{"BirthDate", Order.Ascending}})
in
  SortRows
```



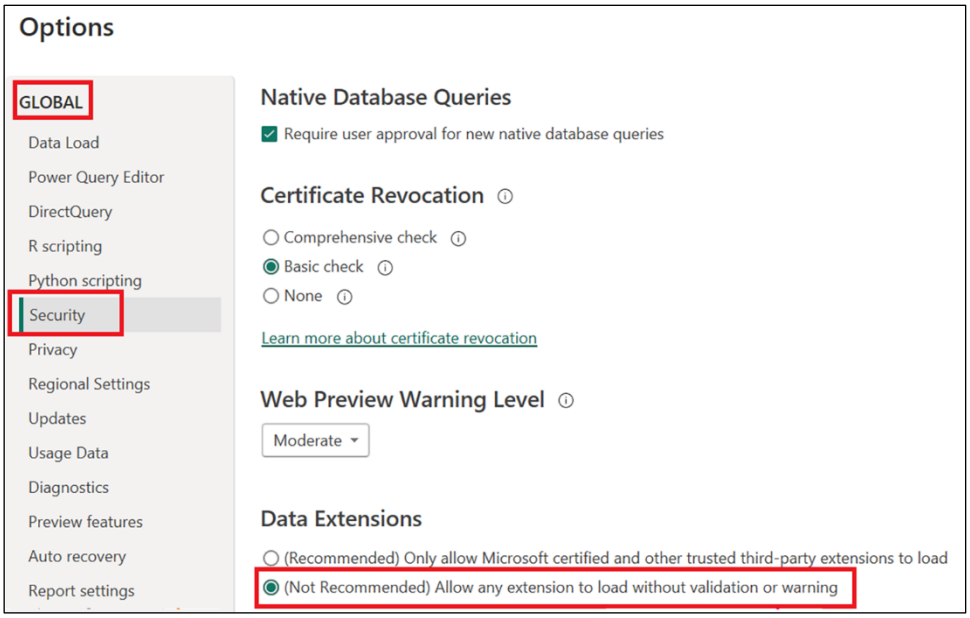


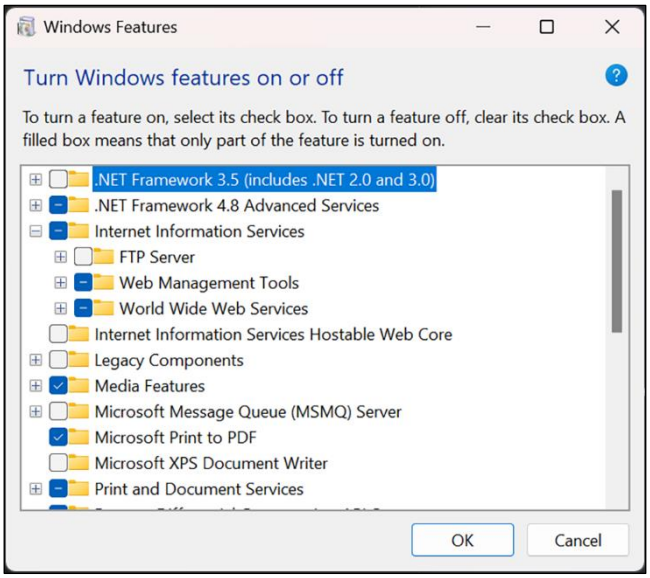
	Transaction ID	Meter Code	Transaction DateTime	Amount Paid
1	1250162207	12028002	11/7/2023 7:30:00 AM	26
2	1250162278	19232002	11/7/2023 7:31:00 AM	152
3	1250131414	19127010	11/7/2023 4:20:00 AM	967
4	1250134465	5073002	11/7/2023 4:48:00 AM	4
5	1250134860	19161010	11/7/2023 4:52:00 AM	673

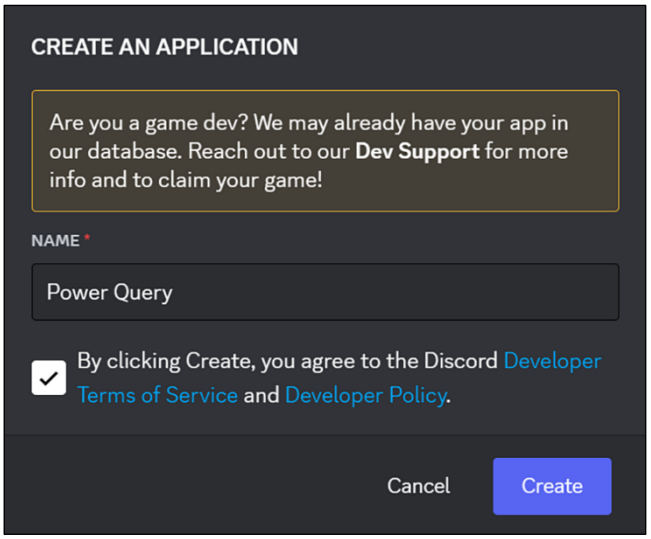
```
= Table.AddColumn(AddedIndex, "Running Total", each
List.Sum( List.FirstN( AddedIndex[Amount Paid], [Index] ) ) )
```

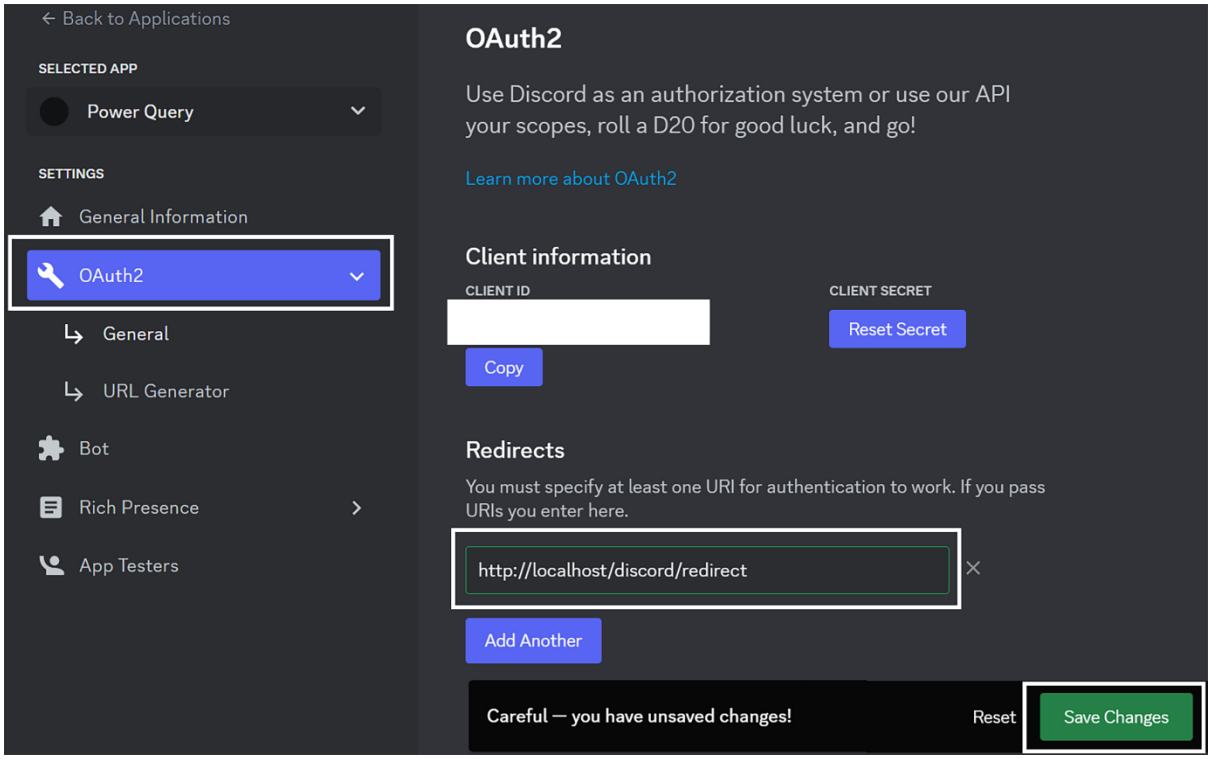
		Amount Paid	Index	Running Total
1	:00 AM	26	1	26
2	:00 AM	152	2	178
3	:00 AM	967	3	1145
4	:00 AM	4	4	1149

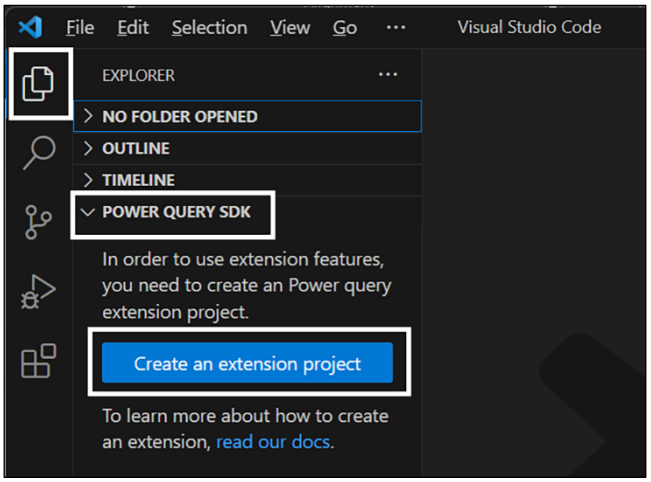
Chapter 16: Enabling Extensions






CREATE AN APPLICATION
Are you a game dev? We may already have your app in our database. Reach out to our Dev Support for more info and to claim your game!
NAME *
Power Query
By clicking Create, you agree to the Discord Developer Terms of Service and Developer Policy.
Cancel
Create


← Back to Applications
SELECTED APP
Power Query
SETTINGS
General Information
OAuth2
General
URL Generator
Bot
Rich Presence
App Testers
OAuth2
Use Discord as an authorization system or use our API your scopes, roll a D20 for good luck, and go!
Learn more about OAuth2
Client information
CLIENT ID
CLIENT SECRET
Reset Secret
Copy
Redirects
You must specify at least one URI for authentication to work. If you pass URIs you enter here.
http://localhost/discord/redirect
Add Another
Careful — you have unsaved changes!
Reset
Save Changes



https://discord.com/api/oauth2/token

POST https://discord.com/api/oauth2/token

Params Authorization Headers (13) Body Pre-request Script Tests Settings

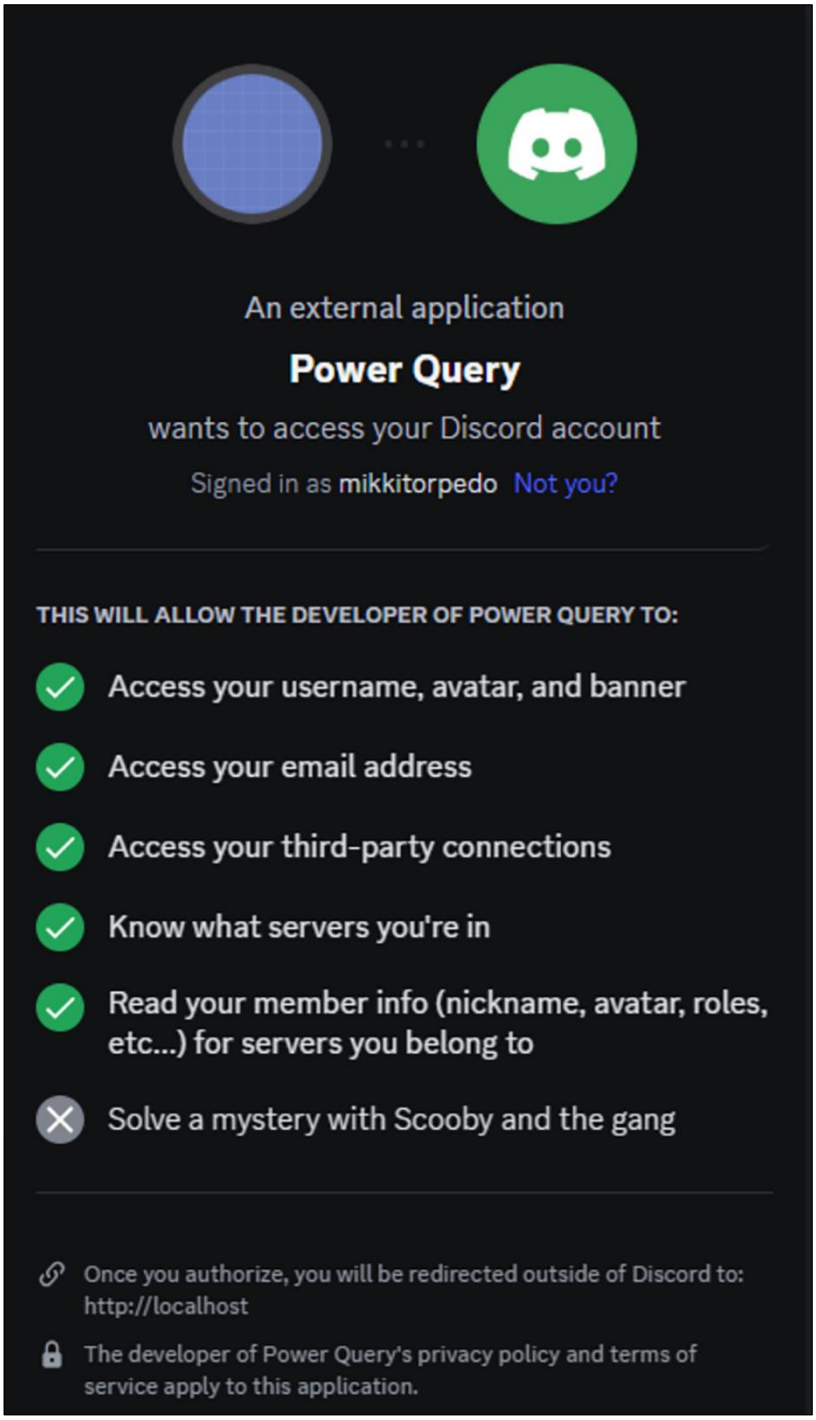
none form-data x-www-form-urlencoded raw binary GraphQL

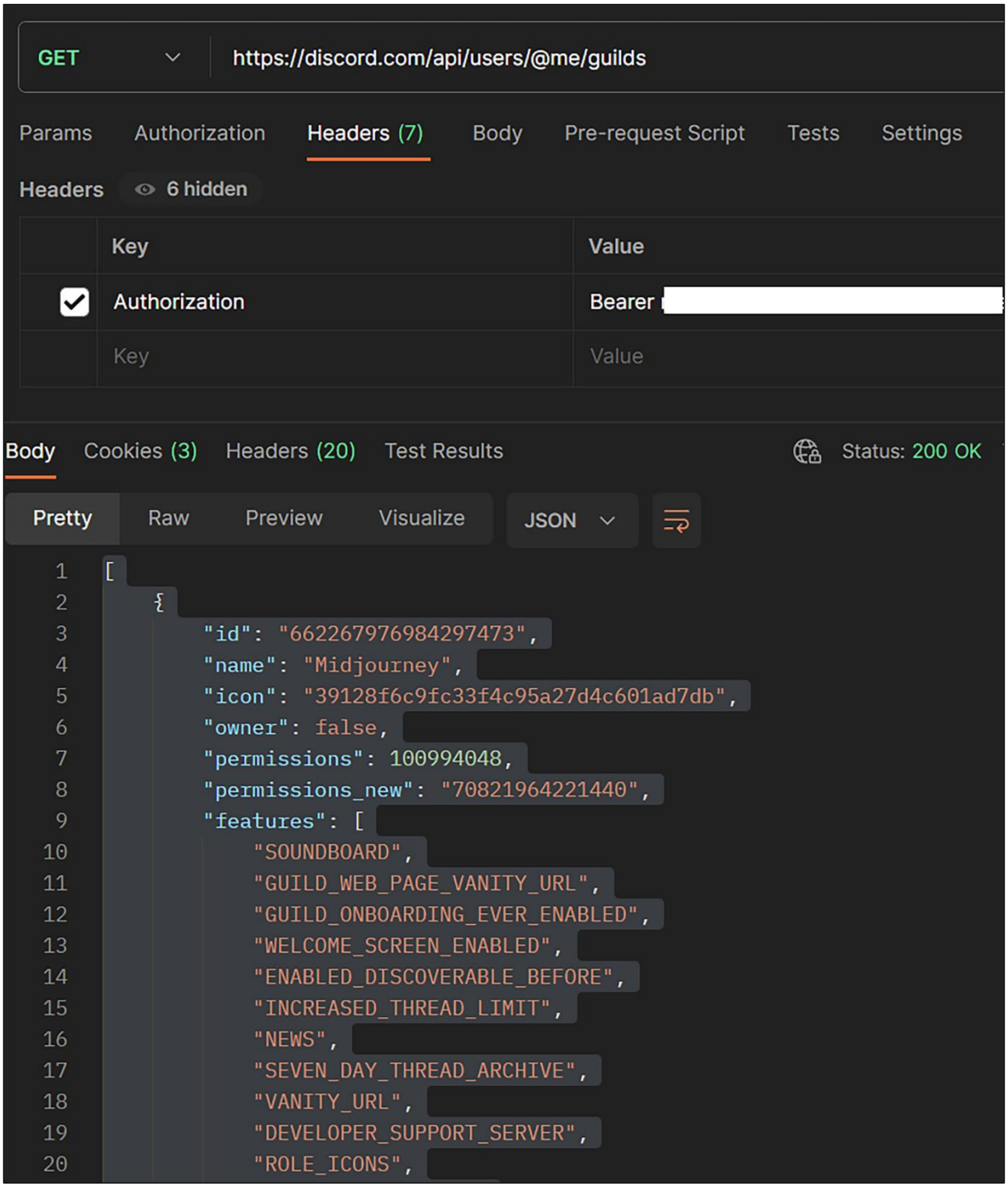
Key	Value
client_id	1128020929558098050
client_secret	[redacted]
grant_type	authorization_code
redirect_uri	http://localhost/discord/redirect
code	[redacted]

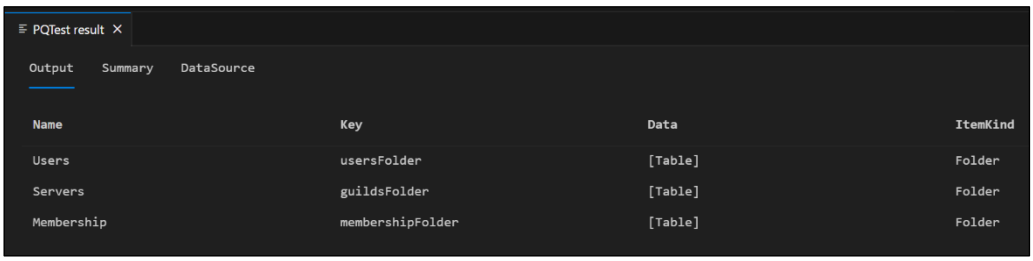
ody Cookies (3) Headers (17) Test Results Status: 200 OK

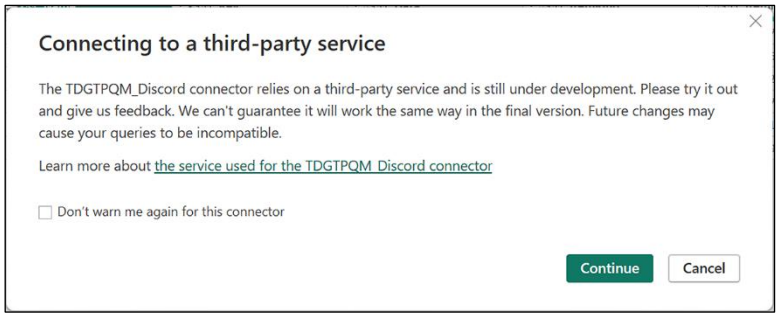
Pretty Raw Preview Visualize JSON

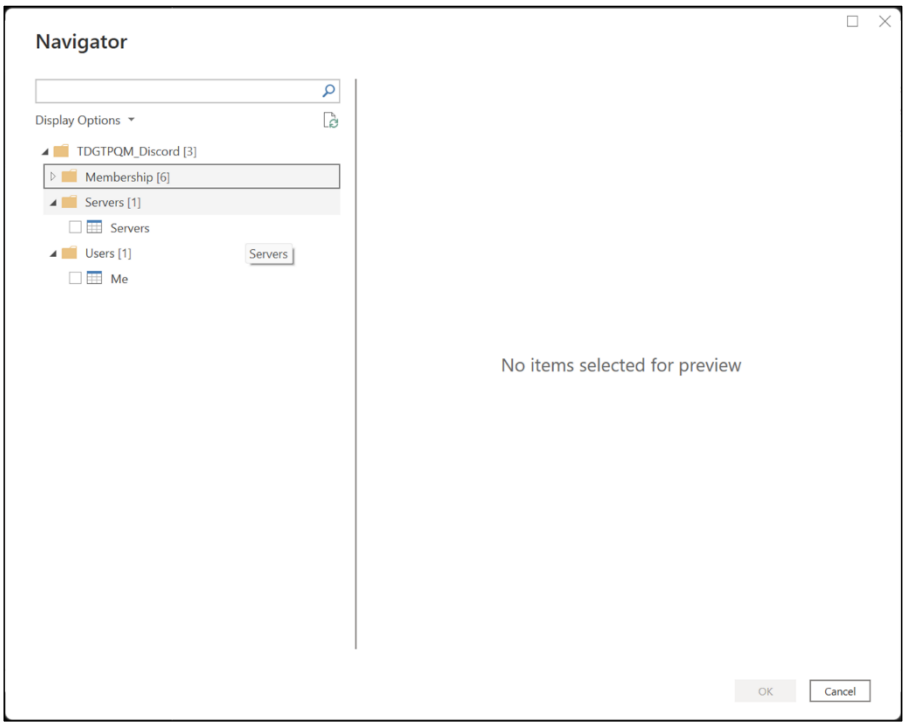

```
{
    "token_type": "Bearer",
    "access_token": "[redacted]",
    "expires_in": 604800,
    "refresh_token": "[redacted]",
    "scope": "identify connections guilds messages.read"
}
```


An external application
Power Query
wants to access your Discord account
Signed in as mikkitorpedo Not you?
THIS WILL ALLOW THE DEVELOPER OF POWER QUERY TO:
Access your username, avatar, and banner
Access your email address
Access your third-party connections
Know what servers you're in
Read your member info (nickname, avatar, roles, etc...) for servers you belong to
Solve a mystery with Scooby and the gang
Once you authorize, you will be redirected outside of Discord to: http://localhost
The developer of Power Query's privacy policy and terms of service apply to this application.


GET
https://discord.com/api/users/@me/guilds
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Headers
6 hidden
Key
Value
Authorization
Bearer
Key
Value
Body
Cookies (3)
Headers (20)
Test Results
Status: 200 OK
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": "662267976984297473",
"name": "Midjourney",
"icon": "39128f6c9fc33f4c95a27d4c601ad7db",
"owner": false,
"permissions": 100994048,
"permissions_new": "70821964221440",
"features": [
"SOUNDBOARD",
"GUILD_WEB_PAGE_VANITY_URL",
"GUILD_ONBOARDING_EVER_ENABLED",
"WELCOME_SCREEN_ENABLED",
"ENABLED_DISCOVERABLE_BEFORE",
"INCREASED_THREAD_LIMIT",
"NEWS",
"SEVEN_DAY_THREAD_ARCHIVE",
"VANITY_URL",
"DEVELOPER_SUPPORT_SERVER",
"ROLE_ICONS",


PQTest result
Output
Summary
DataSource
Name
Key
Data
ItemKind
Users
usersFolder
[Table]
Folder
Servers
guildsFolder
[Table]
Folder
Membership
membershipFolder
[Table]
Folder


Connecting to a third-party service
The TDGTPQM_Discord connector relies on a third-party service and is still under development. Please try it out and give us feedback. We can't guarantee it will work the same way in the final version. Future changes may cause your queries to be incompatible.
Learn more about the service used for the TDGTPQM_Discord connector
Don't warn me again for this connector
Continue
Cancel


Navigator
Display Options
TDGTPQM_Discord [3]
Membership [6]
Servers [1]
Servers
Users [1]
Me
Servers
No items selected for preview
OK
Cancel