



Business Analytics with Power BI

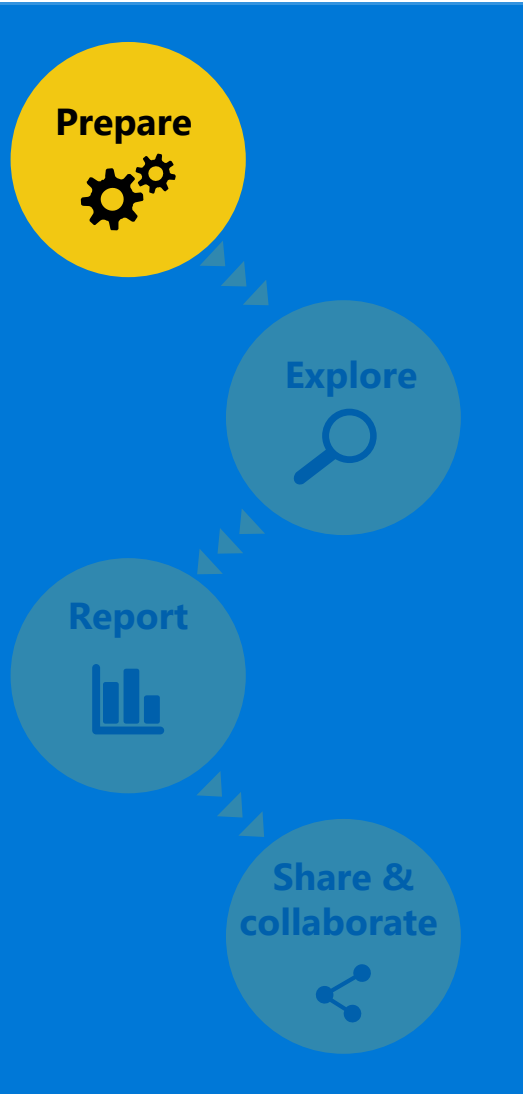
Microsoft Services



Module 1: Power BI Desktop

Lesson 3: Shaping Data

Shaping Data

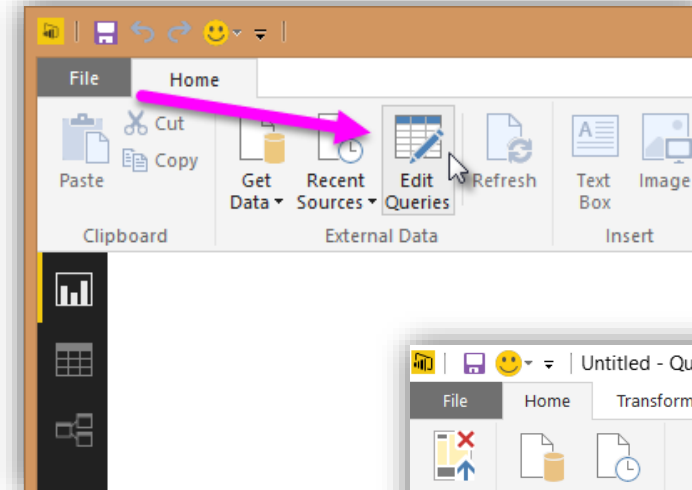


- Shaping is the act of **transforming** the data to meet our **business requirements**
- Apply **data cleaning operations** and correct/remove errors
- **Rename the data** so it is more meaningful
- **Combine data** from different data sources in a single table
- Create **personal** (reusable) **views** of the data
- A set of **out-of-the box transformations** are available to help
- A **programming language** is available for addressing more complex use-cases
- **Source data is not changed**, all the transformations are applied on the Desktop engine

Shaping Data

Query Editor

- After the data is loaded or referenced into Power BI Desktop, the “**Edit Queries**” option in the “**Home**” ribbon allows the edition of the imported entities



- Each entity that was loaded from each data source will generate a **separate query**
- Transformations** are applied **per-entity**, although it is possible to combine them.

A screenshot of the Power BI Query Editor. The title bar shows 'Untitled - Query Editor'. The ribbon has tabs for 'File', 'Home', 'Transform', 'Add Column', and 'View'. The 'Home' tab is active, showing buttons for 'Close & Apply', 'New Source', 'Recent Sources', 'Refresh Preview', 'Properties', and 'Advanced Editor'. Below the ribbon, there is a list of 8 queries: 'FactInternetSales', 'DimCurrency', 'DimCustomer', 'DimDate', 'DimProduct', 'DimPromotion', 'DimSalesTerritory', and 'FactInternetSalesReason'. The 'FactInternetSales' query is selected. To the right of the list is a preview of the data for the selected query. The preview shows a table with columns 'ProductKey' and 'OrderKey'. The data is as follows:

ProductKey	OrderKey
1	310
2	346
3	346
4	336
5	346
6	311
7	310
8	351
9	344

Shaping Data

Query Editor

Prepare



Explore



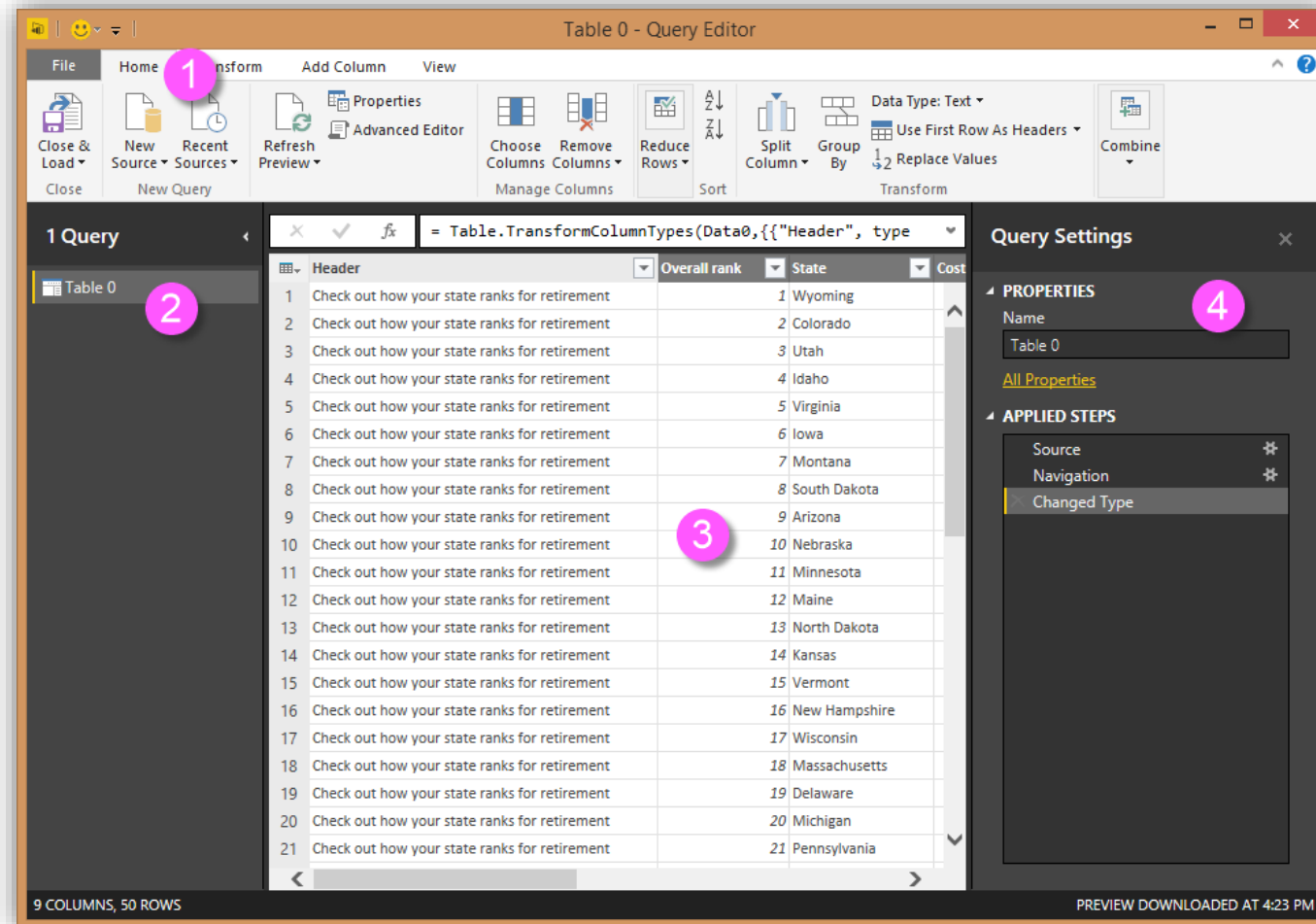
Report



Share & collaborate



- 1 Query Ribbon
- 2 The Left Pane
- 3 The Center Pane
- 4 The Query Settings Pane



Shaping Data

Query Editor – Query Ribbon

Prepare



Explore



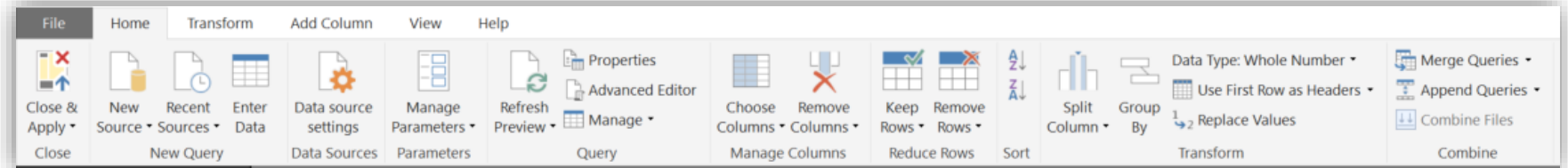
Report



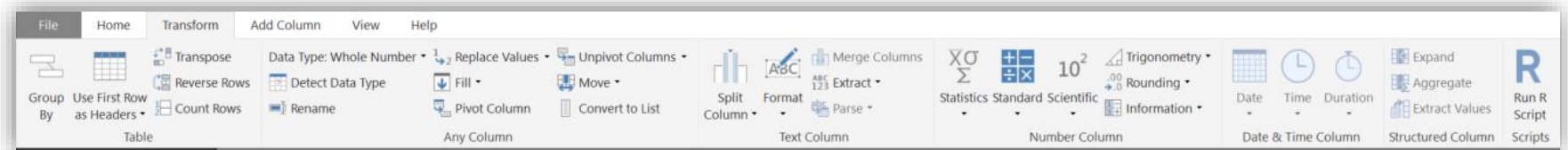
Share & collaborate



- The “**Home**” tab contains the common query tasks including the combination of queries



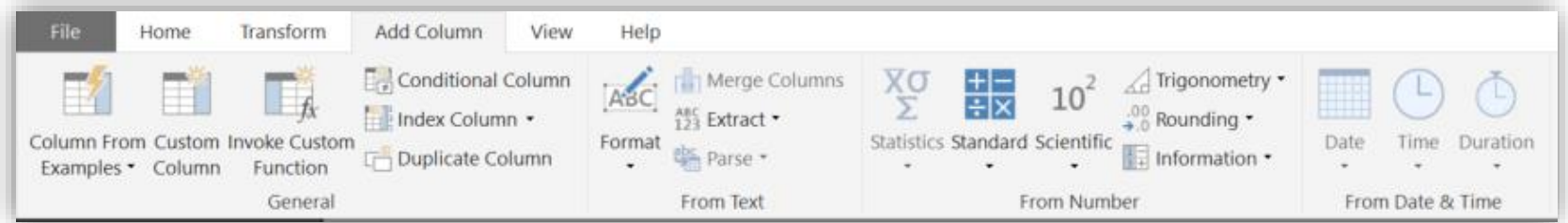
- The “**Transform**” tab provides access to **common data transformation tasks**, such as adding or removing columns, changing data types, splitting columns, and other data-driven tasks



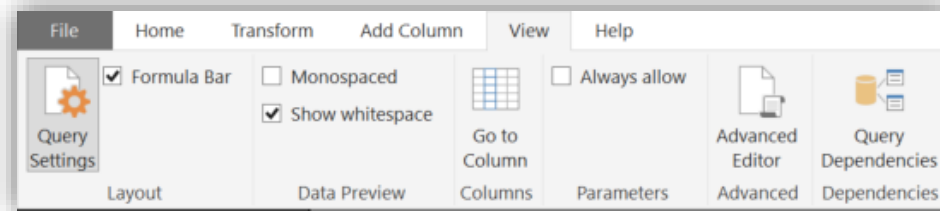
Shaping Data

Query Editor – Query Ribbon

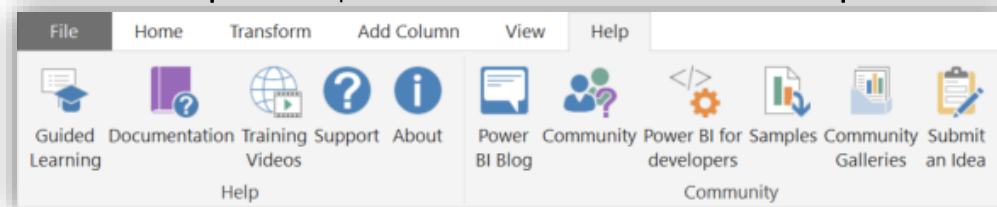
- The **"Add Column"** tab provides additional tasks associated with columns such as formatting column data, adding custom columns or invoking functions



- The **"View"** tab provides access to **query settings** and the **Advanced Editor** where we can develop our own data transformation scripts

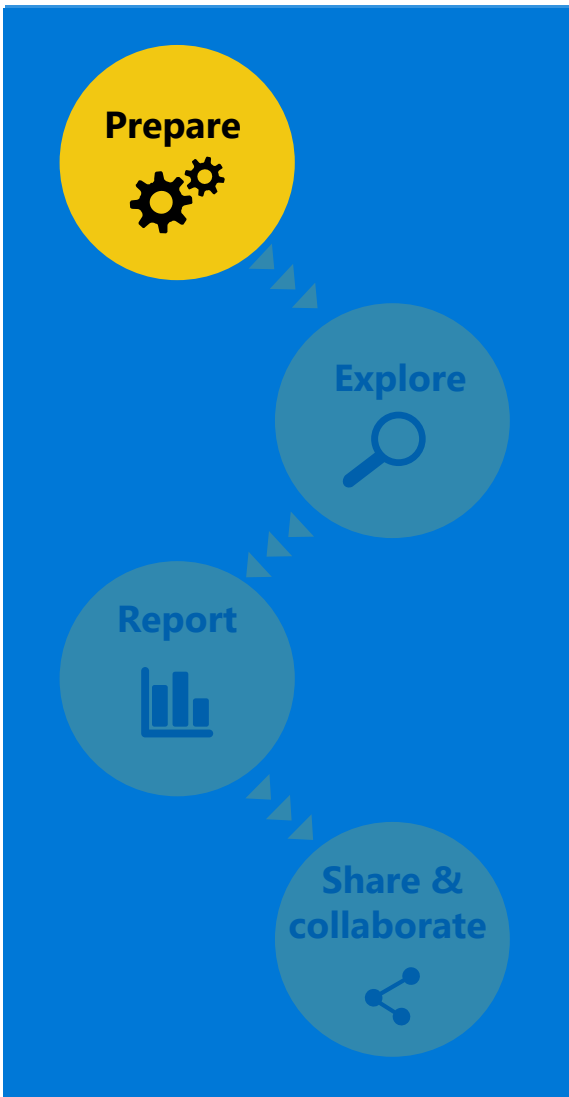


- The **"Help"** tab provides access to some **public resources** and **community**

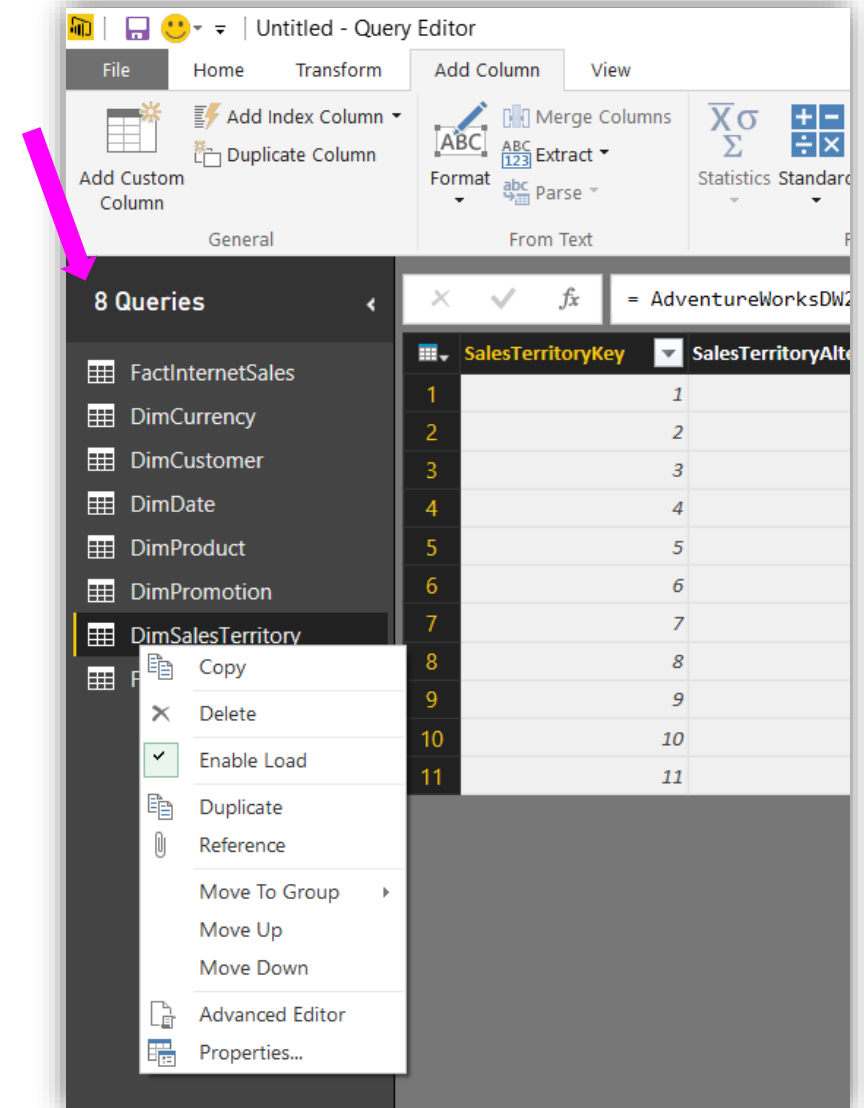


Shaping Data

Query Editor – The Left Pane

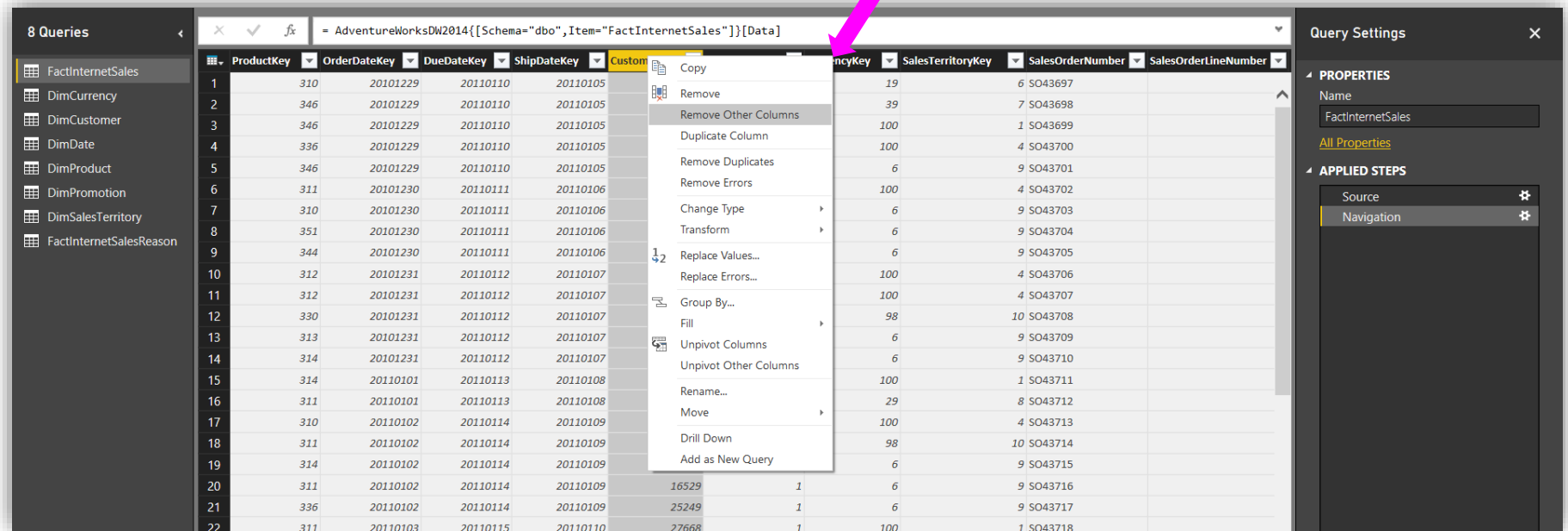


- Displays the **number** of active queries
- The **name** of each query
- Clicking on a query displays the query preview on the Center Pane
- Also allows to **disable** the query load
- **Organizing** the queries into groups
- And also **duplicating** or creating a reference



Shaping Data

Query Editor – The Center Pane



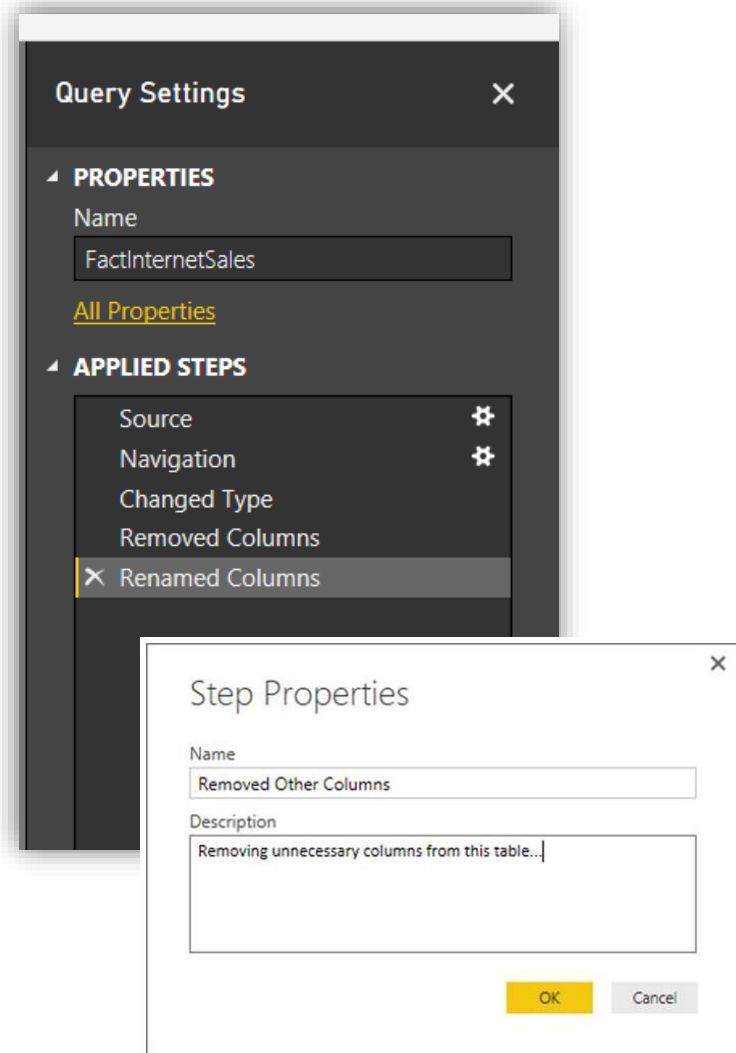
The screenshot displays the Microsoft Query Editor interface. On the left, a sidebar lists 8 queries: FactInternetSales, DimCurrency, DimCustomer, DimDate, DimProduct, DimPromotion, DimSalesTerritory, and FactInternetSalesReason. The main center pane shows a data table with columns: ProductKey, OrderDateKey, DueDateKey, ShipDateKey, CurrencyKey, SalesTerritoryKey, SalesOrderNumber, and SalesOrderLineNumber. A context menu is open over the 'CurrencyKey' column, listing various transformation options such as Copy, Remove, Remove Other Columns, Duplicate Column, Remove Duplicates, Remove Errors, Change Type, Transform, Replace Values..., Replace Errors..., Group By..., Fill, Unpivot Columns, Unpivot Other Columns, Rename..., Move, Drill Down, and Add as New Query. A pink arrow points to the 'CurrencyKey' column header. On the right, the 'Query Settings' pane is visible, showing the 'Name' property set to 'FactInternetSales' and the 'APPLIED STEPS' section with 'Source' and 'Navigation' steps.

	ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CurrencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber
1	310	20101229	20110110	20110105	19	6	SO43697	
2	346	20101229	20110110	20110105	39	7	SO43698	
3	346	20101229	20110110	20110105	100	1	SO43699	
4	336	20101229	20110110	20110105	100	4	SO43700	
5	346	20101229	20110110	20110105	6	9	SO43701	
6	311	20101230	20110111	20110106	100	4	SO43702	
7	310	20101230	20110111	20110106	6	9	SO43703	
8	351	20101230	20110111	20110106	6	9	SO43704	
9	344	20101230	20110111	20110106	6	9	SO43705	
10	312	20101231	20110112	20110107	100	4	SO43706	
11	312	20101231	20110112	20110107	100	4	SO43707	
12	330	20101231	20110112	20110107	98	10	SO43708	
13	313	20101231	20110112	20110107	6	9	SO43709	
14	314	20101231	20110112	20110107	6	9	SO43710	
15	314	20110101	20110113	20110108	100	1	SO43711	
16	311	20110101	20110113	20110108	29	8	SO43712	
17	310	20110102	20110114	20110109	100	4	SO43713	
18	311	20110102	20110114	20110109	98	10	SO43714	
19	314	20110102	20110114	20110109	6	9	SO43715	
20	311	20110102	20110114	20110109	16529	1	9	SO43716
21	336	20110102	20110114	20110109	25249	1	9	SO43717
22	311	20110103	20110115	20110110	27668	1	100	SO43718

- Displays the data for the selected query and the current transformation step
- Many of the transformations can be directly applied here, namely, column related
- It displays a preview of the data and might not display the entire dataset

Shaping Data

Query Editor – The Query Settings Pane



- Each transformation that is applied to a query generates a step
- Those steps are applied in order to generate the desired result
- Order can be changed as long as it respects the needs of previous steps
- It is also possible to edit each step's properties via this pane
- Possible to add Description to Query steps

Demonstration

Building a Query

Shaping Data

Available Data Transformations:

Prepare



Explore



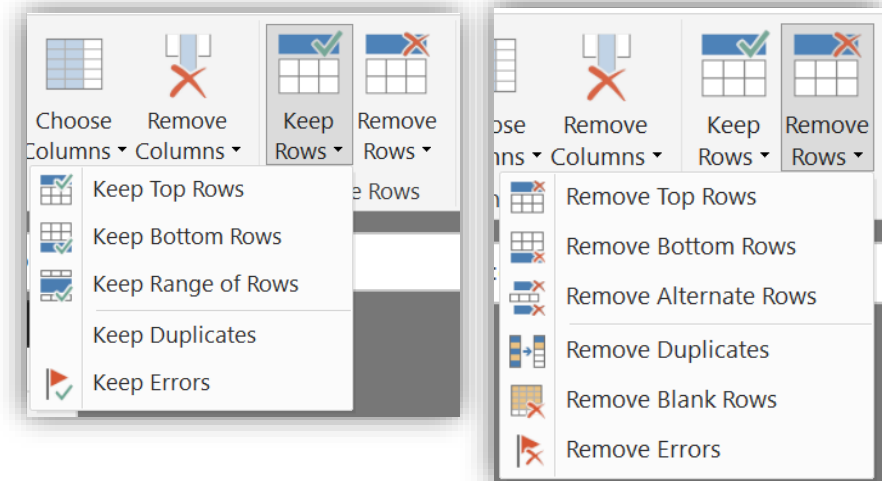
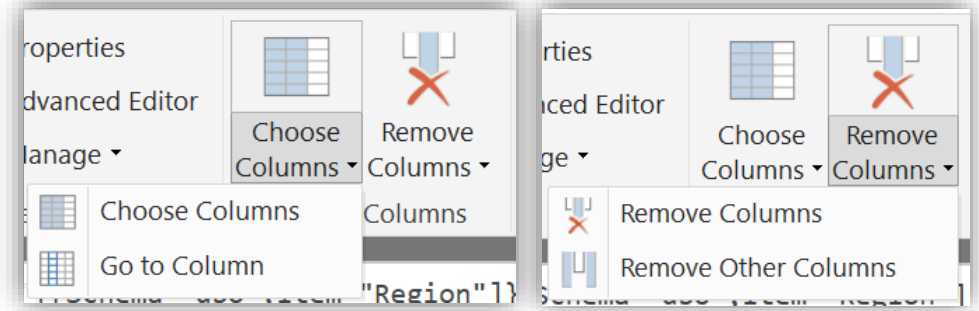
Report



Share & collaborate



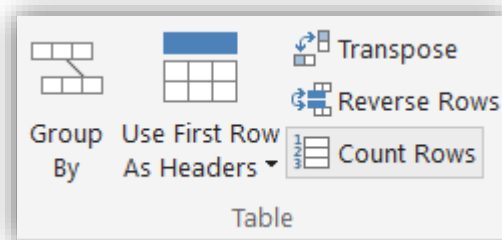
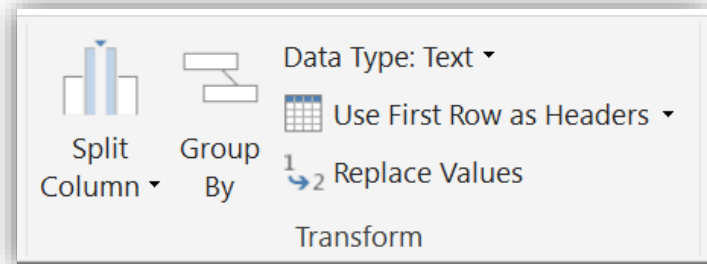
- Manage Columns
 - Choose Columns
 - Remove Columns
 - Remove Other Columns
- Reduce Rows
 - Keep Top/Bottom Rows
 - Keep Range of Rows
 - Remove Top/Bottom Rows
 - Remove Alternate Rows
 - Remove Blank Rows
 - Keep/Remove Duplicates
 - Keep/Remove Errors



Shaping Data

Available Data Transformations

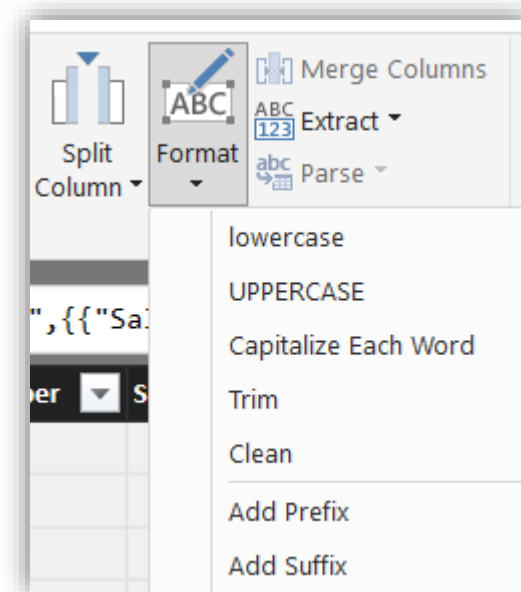
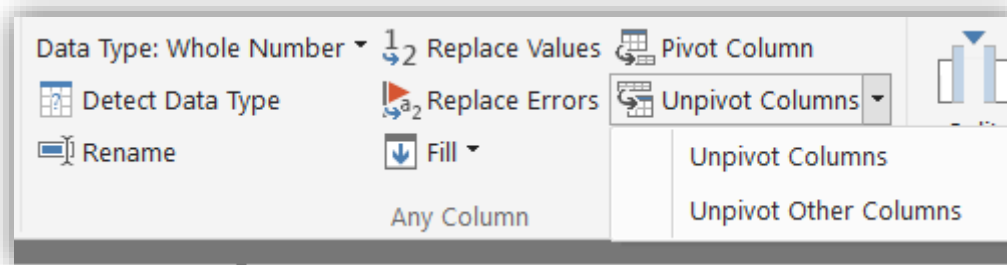
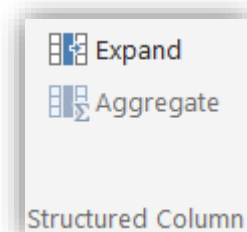
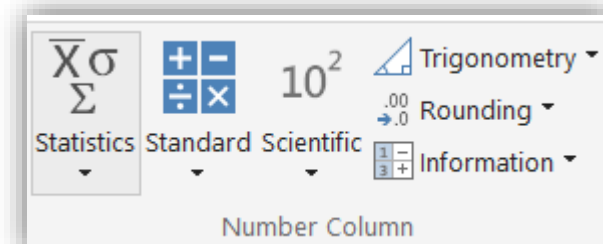
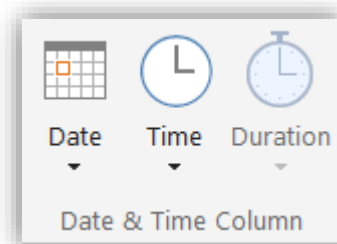
- Sort Columns
- Transform
 - Merge/Split columns
 - Group rows in a table
 - Aggregate data from a table
 - Use first Row as headers
 - Replace Values
 - Transpose
 - Reverse Rows
 - Count Rows
 - Conditional Columns



Shaping Data

Available Data Transformations

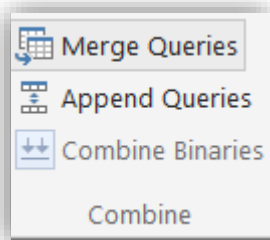
- Transform
 - Filter Table
 - Pivot/Unpivot
 - Change Data Types
 - Text/Number/Date Formatting
 - Extract parts of strings
 - Fill Rows
 - Expand Rows from a related table



Shaping Data

Available Data Transformations

- Combine
 - Merge Queries
 - Append Queries
 - Combine Binaries



Merge

Select a table and matching columns to create a merged table.

FactInternetSales

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	SubcategoryKey
310	20101229	20110110	20110105	21768	1	19	
346	20101229	20110110	20110105	28389	1	39	
346	20101229	20110110	20110105	25863	1	100	
336	20101229	20110110	20110105	14501	1	100	
346	20101229	20110110	20110105	11003	1	6	

DimProduct

ProductKey	ProductAlternateKey	ProductSubcategoryKey	WeightUnitMeasureCode	SizeUnitMeasureCode
1	AR-5381	null	null	null
2	BA-8327	null	null	null
3	BE-2349	null	null	null
4	BE-2908	null	null	null
5	BL-2036	null	null	null

Join Kind

Left Outer (all from first, matching from second)

Left Outer (all from first, matching from second)

Right Outer (all from second, matching from first)

Full Outer (all rows from both)

Inner (only matching rows)

Left Anti (rows only in first)

Right Anti (rows only in second)

OK Cancel

Shaping Data

Available Data Transformations - Merge

Prepare



Explore



Report



Share & collaborate



Merge

Select a table and matching columns to create a merged table.

FactInternetSales

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	...
310	20101229	20110110	20110105	21768	1	19	
346	20101229	20110110	20110105	28389	1	39	
346	20101229	20110110	20110105	25863	1	100	
336	20101229	20110110	20110105	14501	1	100	
346	20101229	20110110	20110105	11003	1	6	

DimProduct

ProductKey	ProductAlternateKey	ProductSubcategoryKey	WeightUnitMeasureCode	SizeUnitMeasureCode
1	AR-5381	null	null	null
2	BA-8327	null	null	null
3	BE-2349	null	null	null
4	BE-2908	null	null	null
5	BL-2036	null	null	null

Join Kind

Left Outer (all from first, matching from second)

✓ The selection has matched 60398 out of the first 60398 rows.

OK Cancel



Expand

Search Columns to Expand

☒ Expand ☐ Aggregate

- ☒ (Select All Columns)
- ☒ ProductKey
- ☒ ProductAlternateKey
- ☒ ProductSubcategoryKey
- ☒ WeightUnitMeasureCode
- ☒ SizeUnitMeasureCode
- ☒ EnglishProductName
- ☒ SpanishProductName
- ☒ FrenchProductName
- ☒ StandardCost
- ☒ FinishedGoodsFlag
- ☒ Color
- ☒ SafetyStockLevel
- ☒ ReorderPoint
- ☒ ListPrice
- ☒ Size
- ☒ SizeRange
- ☒ Weight

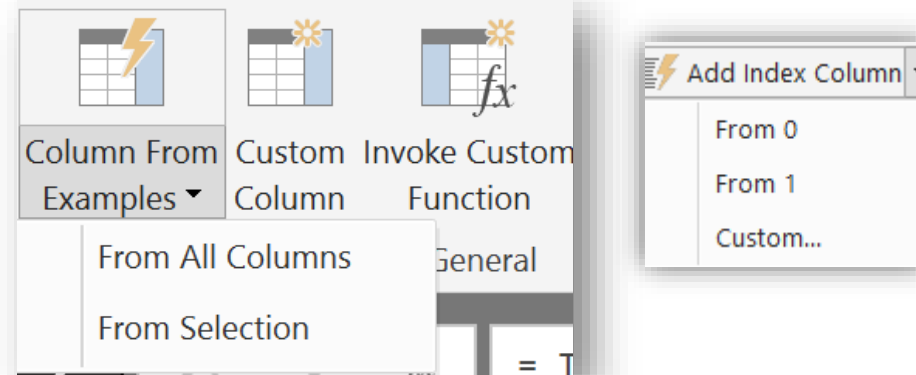
☒ Use original column name as prefix

OK Cancel

Shaping Data

Available Data Transformations

- **Add Column**
 - Insert Custom Column (also used to invoke functions)
 - Insert Index Column
 - Duplicate Column
 - Column From Examples



Shaping Data

Available Data Transformations

Prepare



Explore



Report



Share & collaborate



- Column From Examples
 - Useful when the result is known, but not the transformations to apply or where they are
 - Generates an expression automatically (similar to Excel's Flash Fill)

Add Column From Examples

Enter sample values to create a new column (Ctrl+Enter to apply).

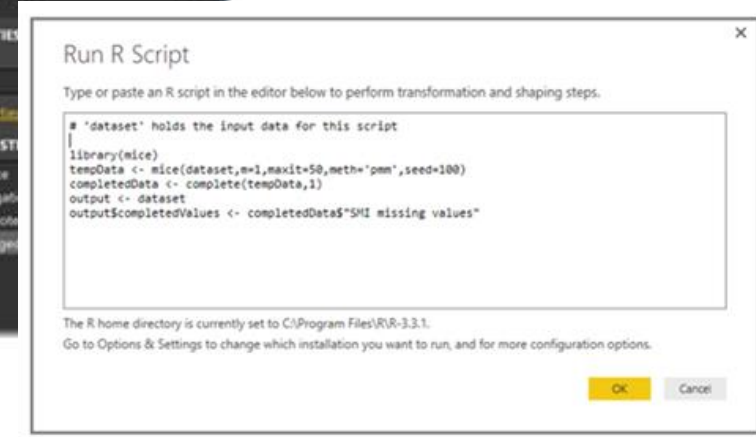
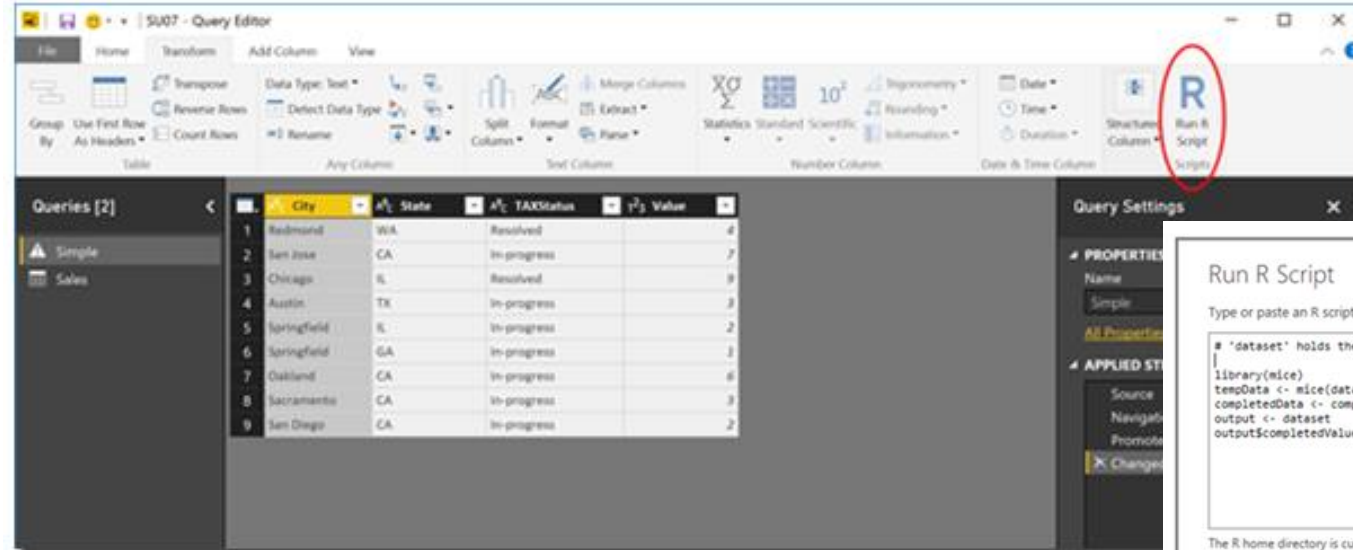
Transform: `Text.Combine([{"State & postal abbreviation"}, " ", "was founded in ", Text.From([Year from ParseDate]), " and has ", [{"Replaced Population (2016 est)[18]}], " people")`

	Population (2016 est)[18]	Corrected State	Replaced Population (2016 est)[18]	ParseDate	Corrected St	Combined
1	4,863,300	Alabama	4863300	14/12/1819	ALABAMA	Alabama, was founded in 1819 and has 4863300 people
2	741,894	Alaska	741894	03/01/1959	ALASKA	Alaska, was founded in 1959 and has 741894 people
3	6,931,071	Arizona	6931071	14/02/1912	ARIZONA	Arizona, was founded in 1912 and has 6931071 people
4	2,988,248	Arkansas	2988248	15/06/1836	ARKANSAS	Arkansas, was founded in 1836 and has 2988248 people
5	39,250,017	California	39250017	09/09/1850	CALIFORNIA	California, was founded in 1850 and has 39250017 people
6	5,540,545	Colorado	5540545	01/08/1876	COLORADO	Colorado, was founded in 1876 and has 5540545 people
7	3,576,452	Connecticut	3576452	09/01/1788	CONNECTICUT	Connecticut, was founded in 1788 and has 3576452 people
8	952,065	Delaware	952065	07/12/1787	DELAWARE	Delaware, was founded in 1787 and has 952065 people
9	20,612,439	Florida	20612439	03/03/1845	FLORIDA	Florida, was founded in 1845 and has 20612439 people
10	10,310,371	Georgia	10310371	02/01/1788	GEORGIA	Georgia, was founded in 1788 and has 10310371 people
11	1,428,557	Hawaii	1428557	21/08/1959	HAWAII	Hawaii, was founded in 1959 and has 1428557 people
12	1,683,140	Idaho	1683140	03/07/1890	IDAHO	Idaho, was founded in 1890 and has 1683140 people
13	12,801,539	Illinois	12801539	03/12/1818	ILLINOIS	Illinois, was founded in 1818 and has 12801539 people
14	6,633,053	Indiana	6633053	11/12/1816	INDIANA	Indiana, was founded in 1816 and has 6633053 people
15	3,134,693	Iowa	3134693	28/12/1846	IOWA	Iowa, was founded in 1846 and has 3134693 people
16	2,907,289	Kansas	2907289	29/01/1861	KANSAS	Kansas, was founded in 1861 and has 2907289 people
17	4,436,974	Kentucky	4436974	01/06/1792	KENTUCKY	Kentucky, was founded in 1792 and has 4436974 people
18	4,581,555	Louisiana	4581555	26/01/1813	LOUISIANA	Louisiana, was founded in 1813 and has 4581555 people

Shaping Data

Available Data Transformations – R Scripts

- Run R Script from the Query Editor
- Perform **data cleansing, advanced data shaping, and analytics** including completion of missing data, predictions, and clustering etc.
- R should be installed locally



Shaping Data

Available Data Transformations – Advanced Editor

- **Advanced Editor**
 - Every transformation generates code automatically
 - For more advanced scenarios
 - OOB transformations will cover more than 90% of the cases



Shaping Data

An example (how to go from a matrix to a table?)

	Column1	Column2	Column3	Column4	Column5
1	null	Northland Region	null	Auckland Region	null
2	null	All Ages	Under 20	All Ages	Under 20
3	1991	2475	263	18347	1346
4	1992	2438	240	18183	1323



	Region	Age Group	Attribute	Value
1	Northland Region	All Ages	1991	2475
2	Northland Region	All Ages	1992	2438
3	Northland Region	Under 20	1991	263
4	Northland Region	Under 20	1992	240
5	Auckland Region	All Ages	1991	18347
6	Auckland Region	All Ages	1992	18183
7	Auckland Region	Under 20	1991	1346
8	Auckland Region	Under 20	1992	1323

Prepare



Explore



Report



Share & collaborate



Shaping Data

An example (how to go from a matrix to a table?)

	Column1	Column2	Column3	Column4	Column5
1	null	Northland Region	null	Auckland Region	null
2	null	All Ages	Under 20	All Ages	Under 20
3	1991	2475	263	18347	1346
4	1992	2438	240	18183	1323



Transpose

	Column1	Column2	Column3	Column4
1	null	null	1991	1992
2	Northland Reg	All Ages	2475	2438
3	null	Under 20	263	240
4	Auckland Regi	All Ages	18347	18183
5	null	Under 20	1346	1323

Shaping Data

An example (how to go from a matrix to a table?)

	Column1	Column2	Column3	Column4
1	null	null	1991	1992
2	Northland Reg	All Ages	2475	2438
3	null	Under 20	263	240
4	Auckland Regi	All Ages	18347	18183
5	null	Under 20	1346	1323



Use First Row as Header

	Column1	Column2	1991	1992
1	Northland Reg	All Ages	2475	2438
2	null	Under 20	263	240
3	Auckland Regi	All Ages	18347	18183
4	null	Under 20	1346	1323

Shaping Data

An example (how to go from a matrix to a table?)

	Column1	Column2	1991	1992
1	Northland Reg	All Ages	2475	2438
2	<i>null</i>	Under 20	263	240
3	Auckland Regi	All Ages	18347	18183
4	<i>null</i>	Under 20	1346	1323

Rename Columns

	Region	Age Group	1991	1992
1	Northland Region	All Ages	2475	2438
2	<i>null</i>	Under 20	263	240
3	Auckland Region	All Ages	18347	18183
4	<i>null</i>	Under 20	1346	1323

Shaping Data

An example (how to go from a matrix to a table?)

	Region	Age Group	1991	1992
1	Northland Region	All Ages	2475	2438
2		Under 20	263	240
3	Auckland Region	All Ages	18347	18183
4		Under 20	1346	1323



Unpivot "1991" and "1992" and Fill Down

	Region	Age Group	Attribute	Value
1	Northland Region	All Ages	1991	2475
2	Northland Region	All Ages	1992	2438
3	Northland Region	Under 20	1991	263
4	Northland Region	Under 20	1992	240
5	Auckland Region	All Ages	1991	18347
6	Auckland Region	All Ages	1992	18183
7	Auckland Region	Under 20	1991	1346
8	Auckland Region	Under 20	1992	1323

Shaping Data

An example (how to go from a matrix to a table?)

	Region	Age Group	Attribute	Value
1	Northland Region	All Ages	1991	2475
2	Northland Region	All Ages	1992	2438
3	Northland Region	Under 20	1991	263
4	Northland Region	Under 20	1992	240
5	Auckland Region	All Ages	1991	18347
6	Auckland Region	All Ages	1992	18183
7	Auckland Region	Under 20	1991	1346
8	Auckland Region	Under 20	1992	1323

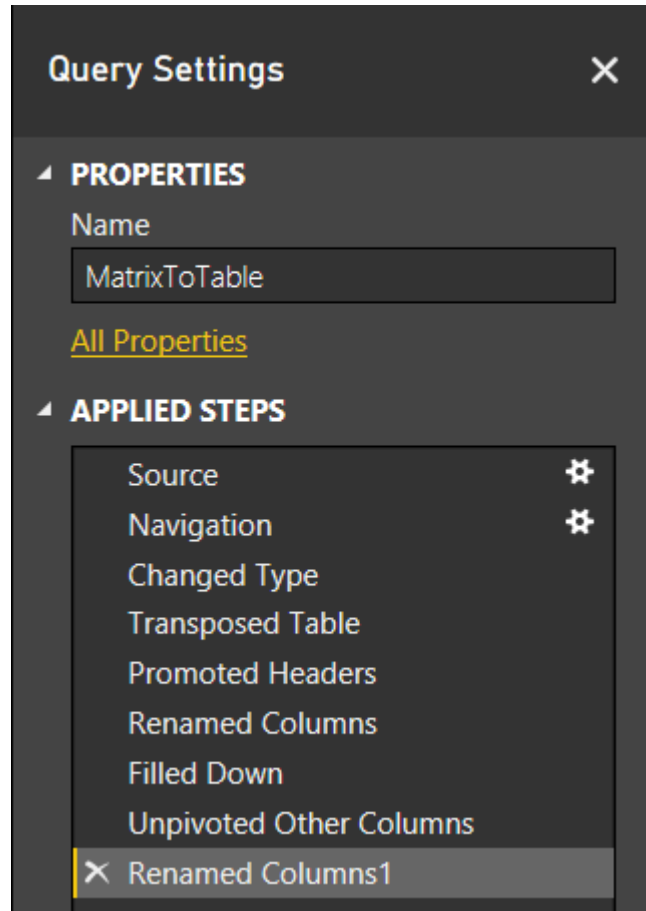
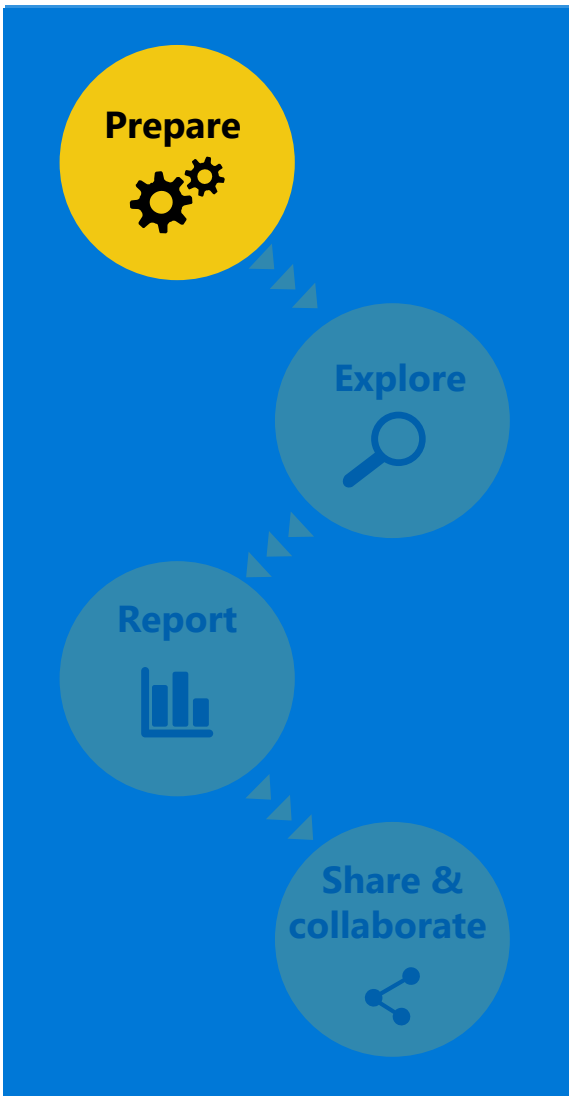


Rename Columns

	Region	Age Group	Year	Value
1	Northland Region	All Ages	1991	2475
2	Northland Region	All Ages	1992	2438
3	Northland Region	Under 20	1991	263
4	Northland Region	Under 20	1992	240
5	Auckland Region	All Ages	1991	18347
6	Auckland Region	All Ages	1992	18183
7	Auckland Region	Under 20	1991	1346
8	Auckland Region	Under 20	1992	1323

Shaping Data

An example (how to go from a matrix to a table?)



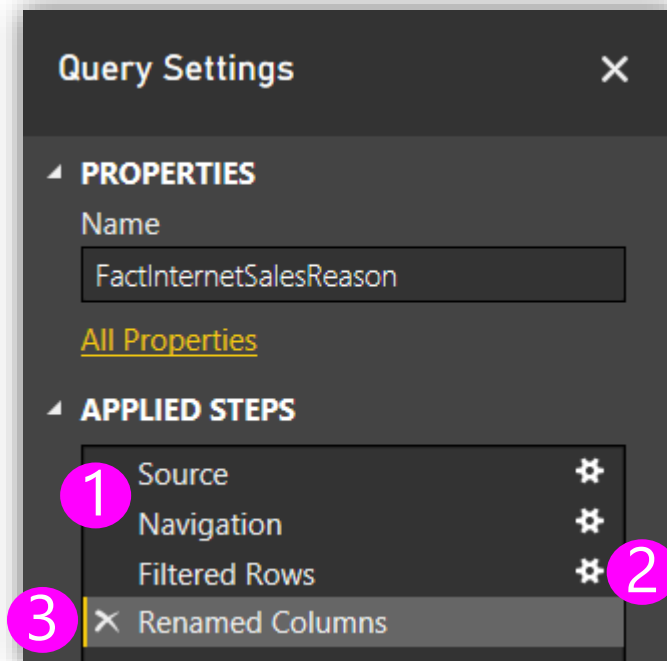
Code
Equivalent

```
let
    Source =
        Excel.Workbook(File.Contents("C:\Users\femar\Desktop\
        Business Analytics with Power BI\Empowering the Data
        Analyst\My Samples\TransposeExample.xlsx"), null, true),
    Table6_Table =
        Source[Item="Table6",Kind="Table"][Data],
    #"Changed Type" =
        Table.TransformColumnTypes(Table6_Table,{{"Column1",
        type any}, {"Column2", type any}, {"Column3", type any},
        {"Column4", type any}, {"Column5", type any}}),
    #"Transposed Table" = Table.Transpose(#"Changed
    Type"),
    #"Promoted Headers" =
        Table.PromoteHeaders(#"Transposed Table"),
    #"Renamed Columns" =
        Table.RenameColumns(#"Promoted
        Headers",{{"Column1", "Region"}, {"Column2", "Age
        Group"}}),
    #"Filled Down" = Table.FillDown(#"Renamed
        Columns",{"Region"}),
    #"Unpivoted Other Columns" =
        Table.UnpivotOtherColumns(#"Filled Down", {"Region",
        "Age Group"}, "Attribute", "Value"),
    #"Renamed Columns1" =
        Table.RenameColumns(#"Unpivoted Other
        Columns",{{"Attribute", "Year"}})
in
    #"Renamed Columns1"
```

Shaping Data

Query Folding

- In some cases, Power BI Desktop will be able to push the transformation work to the data source
- This is important from an optimization perspective.
- If query folding doesn't happen, all of the data has to be loaded into the tool



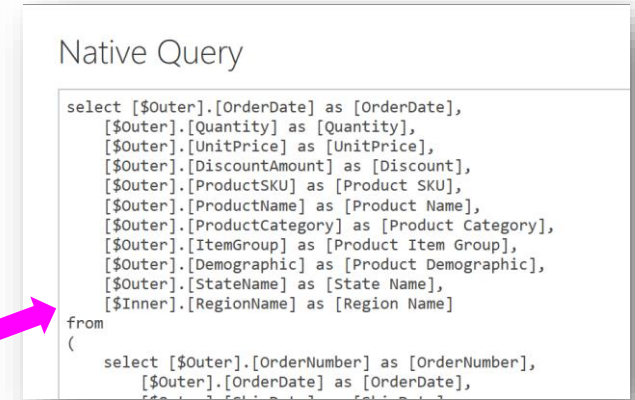
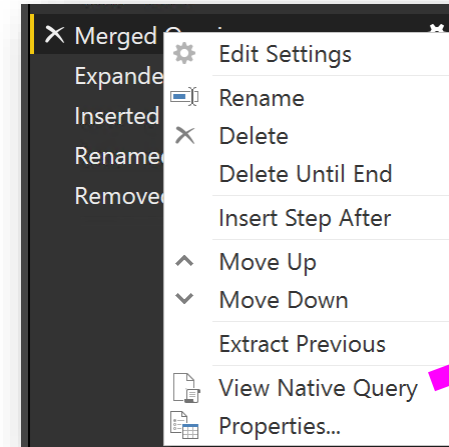
```
SELECT
[salesordernumber] as [Sales Order Number],
[salesorderlinenumber] AS
[SalesOrderLineNumber],
[salesreasonkey] AS [SalesReasonKey]
FROM
[dbo].[factinternetsalesreason] 1
WHERE
[salesorderlinenumber] = 2 2
```

3

Shaping Data

Query Folding

- Supported data sources:
 - Relational Databases
 - Odata
 - Exchange
 - Active Directory

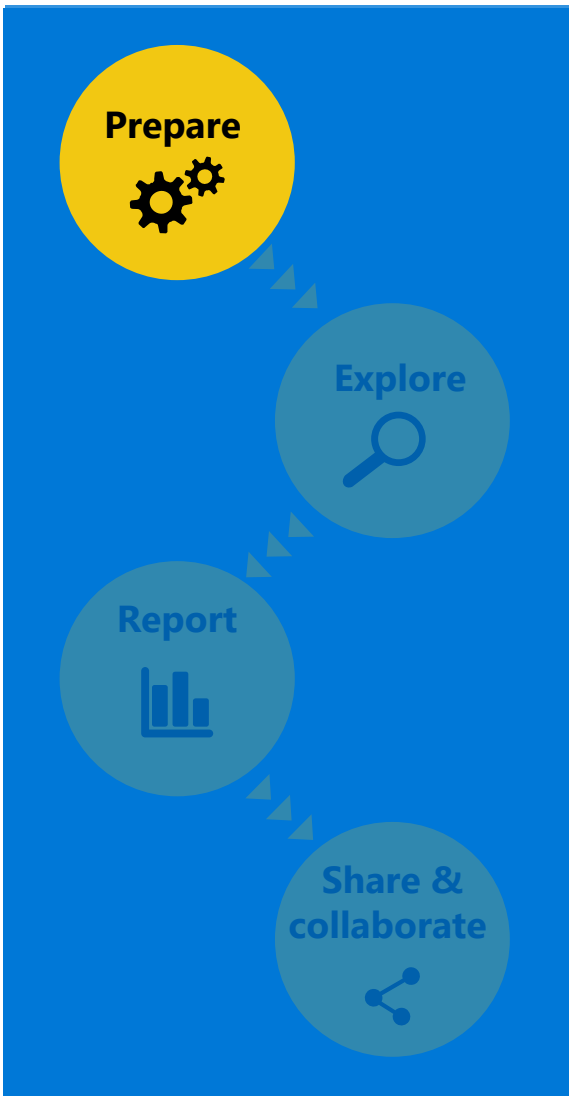


- Not all transformations are supported (for instance, custom SQL prevents folding)
- In order to see what query is being generated use the "View Native Query" option, for a given step.
- As the tool evolves, more transformations and sources will support folding

Shaping Data

Query Formulas

- The language was previously known as **M** or **Power Query Formula Language**
- Has a standard library with built-in functions
 - <https://msdn.microsoft.com/library/Mt253322?ui=en-US&rs=en-IN&ad=IN>
- Is **case-sensitive** and **strongly typed**
- Every transformation generates code automatically. Each step maps to an instruction.
- Use the **Advanced Editor** to edit the query or **Blank Query** data source to start from an empty query



Shaping Data

Query Formulas – An example

Prepare



Explore



Report



Share & collaborate



```
let
    Source = Table.FromColumns({Lines.FromBinary(File.Contents("C:\Users\femar\...\InternetSales2005.txt"),null,null,1252)}),
    #"Split Column by Delimiter" = Table.SplitColumn(Source,"Column1",Splitter.SplitTextByDelimiter(", ", QuoteStyle.Csv),{"Column1.1", "Column1.2"}),
    #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Column1.1", type text}, {"Column1.2", type text}}),
    #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type"),
    #"Removed Columns" = Table.RemoveColumns(#"Promoted Headers",{"Column27", "Column28", "Column29", "Column30", "Column31", "Column32"}),
in
    #"Removed Columns"
```

Query Settings

PROPERTIES

Name

InternetSales2005

[All Properties](#)

APPLIED STEPS

Source



Split Column by Delimiter



Changed Type

Promoted Headers

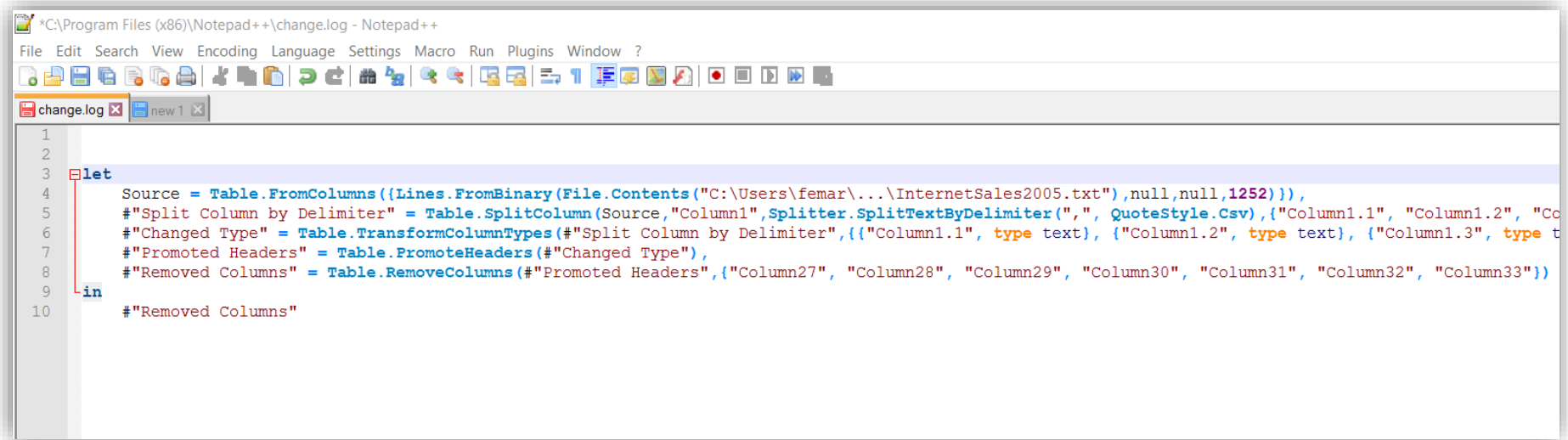
Removed Columns

- Every query starts with a **let** instruction and ends with an **in**
- Each step name is prefixed with a **#** if it has spaces and its **name matches** exactly the one in the **Query Settings Pane** and the **order** is also the **same**
- Several built-in functions implement the interface functionality with the **Object.Function(...)** syntax like **Table.RemoveColumns**

Shaping Data

Query Formulas – Syntax Highlighting

- Power BI Desktop doesn't have syntax highlighting, but there are community alternatives:
- <http://www.mattmasson.com/2014/11/notepad-language-file-for-the-power-query-formula-language-m/>



The screenshot shows a Notepad++ window with a file named 'change.log'. The text inside is a Power Query M formula, and the words are color-coded to match the Power Query syntax: 'let' is blue, 'Source' is blue, 'Table.FromColumns' is blue, 'Lines.FromBinary' is blue, 'File.Contents' is blue, 'null' is blue, '1252' is blue, 'Split Column by Delimiter' is blue, 'Table.SplitColumn' is blue, 'Splitter.SplitTextByDelimiter' is blue, 'Column1.1' is blue, 'Column1.2' is blue, 'Column1.3' is blue, 'type text' is orange, 'Promoted Headers' is blue, 'Table.PromoteHeaders' is blue, 'Removed Columns' is blue, 'Table.RemoveColumns' is blue, and 'Column27' through 'Column33' are blue. The formula is as follows:

```
1
2
3 let
4     Source = Table.FromColumns({Lines.FromBinary(File.Contents("C:\Users\femar\...\InternetSales2005.txt"),null,null,1252)}),
5     #"Split Column by Delimiter" = Table.SplitColumn(Source,"Column1",Splitter.SplitTextByDelimiter(",", QuoteStyle.Csv),{"Column1.1", "Column1.2", "Co
6     #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Column1.1", type text}, {"Column1.2", type text}, {"Column1.3", type t
7     #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type"),
8     #"Removed Columns" = Table.RemoveColumns(#"Promoted Headers",{"Column27", "Column28", "Column29", "Column30", "Column31", "Column32", "Column33"})
9 in
10     #"Removed Columns"
```

Shaping Data

Query Formulas – User Functions

- Functions allow for **code reuse**. They require a particular syntax:

```
let
ParseInternetSalesFiles=(folder as text,file as text)=>
let
    Source = Table.FromColumns({Lines.FromBinary(File.Contents(folder & file),null,null,1252)}),
    #"Split Column by Delimiter" = Table.SplitColumn(Source,"Column1",Splitter.SplitTextByDelimiter(",", QuoteStyle.Csv),{"Column1.1","Column1.2"},
    #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Column1.1", type text}, {"Column1.2", type text}},
    #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type"),
    #"Removed Columns" = Table.RemoveColumns(#"Promoted Headers",{"Column27", "Column28", "Column29", "Column30", "Column31"}),
in
    #"Removed Columns"
in
    ParseInternetSalesFiles
```

- The procedure has to be enclosed in a new **let** and **in** block
- Parameters** are declared in the function definition and then they can be used to make the procedure generic:

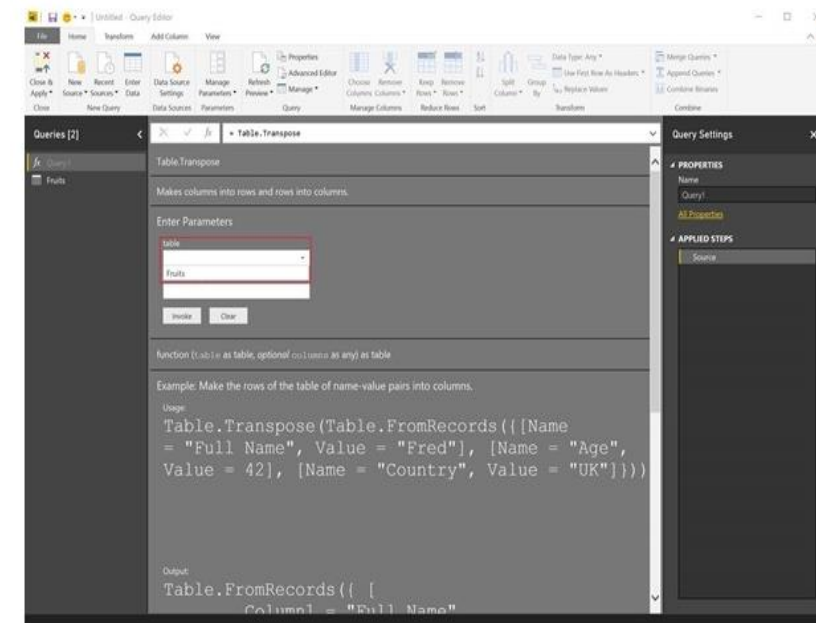
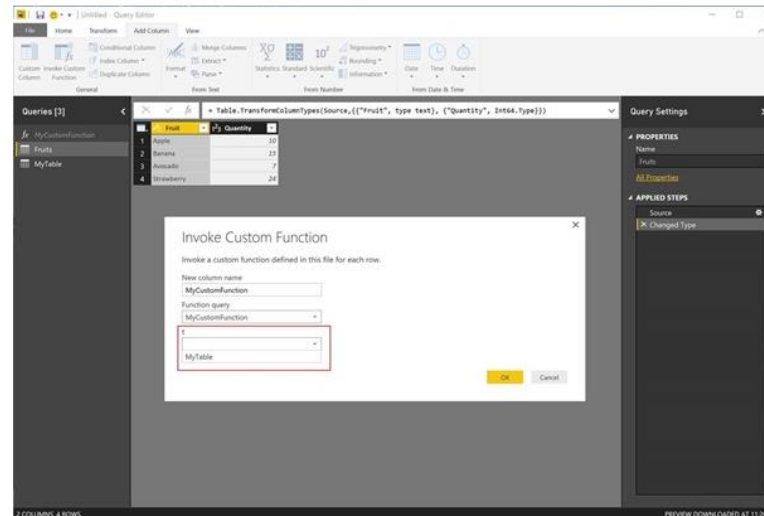
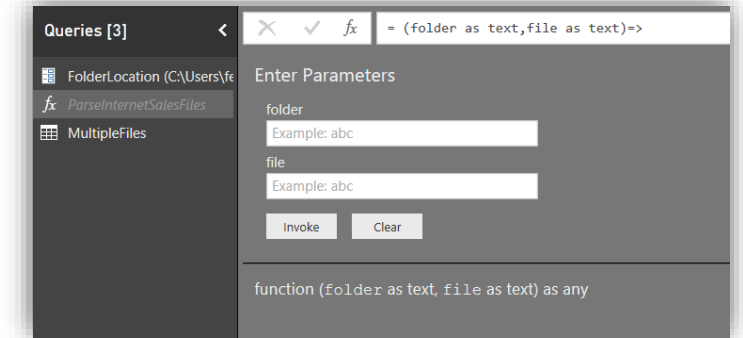
```
let
ParseInternetSalesFiles=(folder as text file as text)=>
```

```
ry(File.Contents(folder & file),null,
Column(Source,"Column1",Splitter.S
```

Shaping Data

Query Formulas – User Functions

- You can **invoke** them via the Query Editor
- Notice the **fx** icon in the Left Pane that indicates a user function
- Supports parameters of type “Table”



Shaping Data

Query Formulas – User Functions

- You can also invoke them with the **Add Custom Column** task or with **Invoke Custom Function**

The screenshot illustrates the process of adding a custom column in Power BI Desktop. It shows the 'Add Custom Column' task pane with the formula `= ParseInternetSalesFiles([Folder Path],[Name])` and the 'Invoke Custom Function' task pane with the formula `= Table.AddColumn(#"Removed Columns", "Custom", each ...)`. The 'Add Custom Column' task pane also shows the 'New column name' field set to 'ParsedFile' and the 'Available columns' list containing 'Name' and 'Folder Path'. The 'Invoke Custom Function' task pane shows the 'General' tab with the 'Custom Column' and 'Invoke Custom Function' buttons highlighted. The 'Add Custom Column' task pane also shows the 'Learn about Power BI Desktop formulas' link and the 'No syntax errors have been detected' message.

1

2

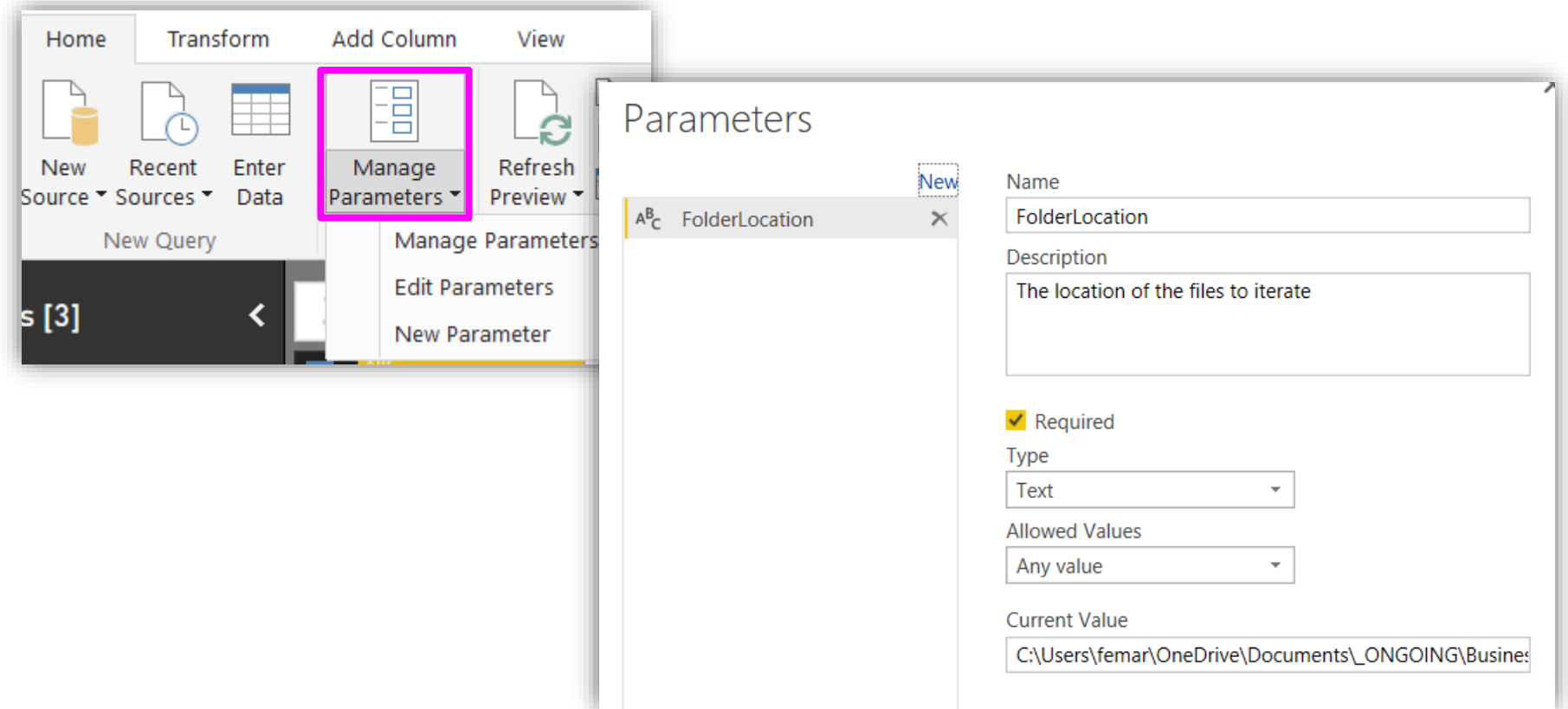
3

Name	Folder Path	ParsedFile
InternetSales2005.txt	C:\Users\femar\SkyDrive\Documents_ONGOING\Business Analytics with	Table
InternetSales2006.txt	C:\Users\femar\SkyDrive\Documents_ONGOING\Business Analytics with	Table
InternetSales2007.abc	C:\Users\femar\SkyDrive\Documents_ONGOING\Business Analytics with	Table
InternetSales2007.txt	C:\Users\femar\SkyDrive\Documents_ONGOING\Business Analytics with	Table

Shaping Data

Query Formulas – Parameters

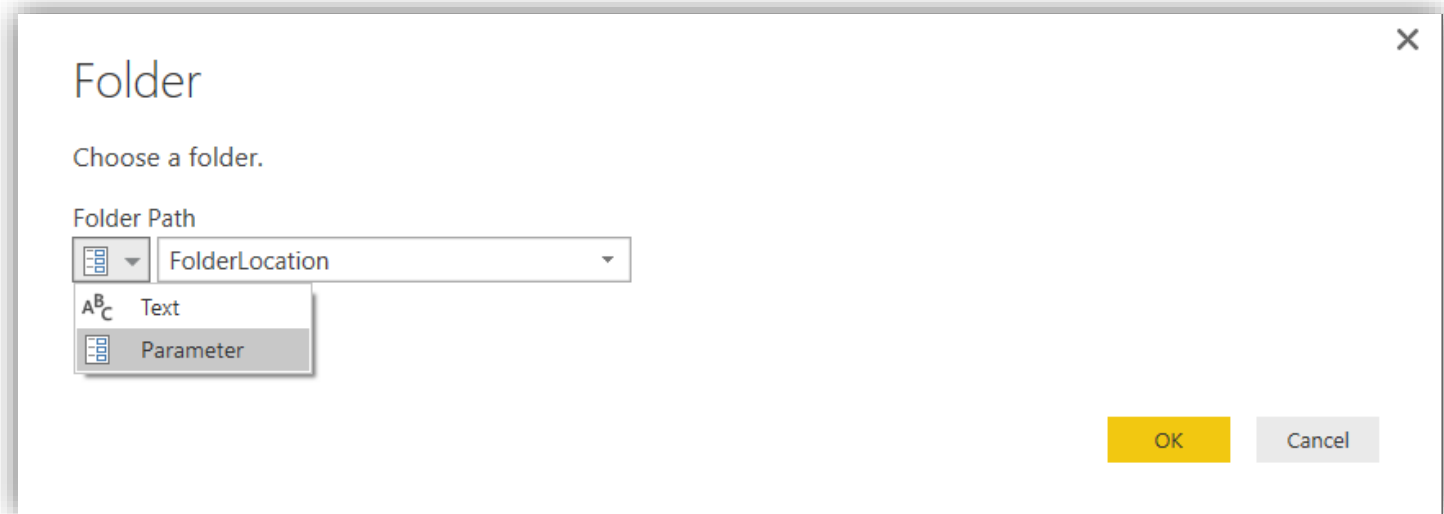
- Queries and other objects can use parameters to get a **dynamic behavior**



Shaping Data

Query Formulas – Parameters

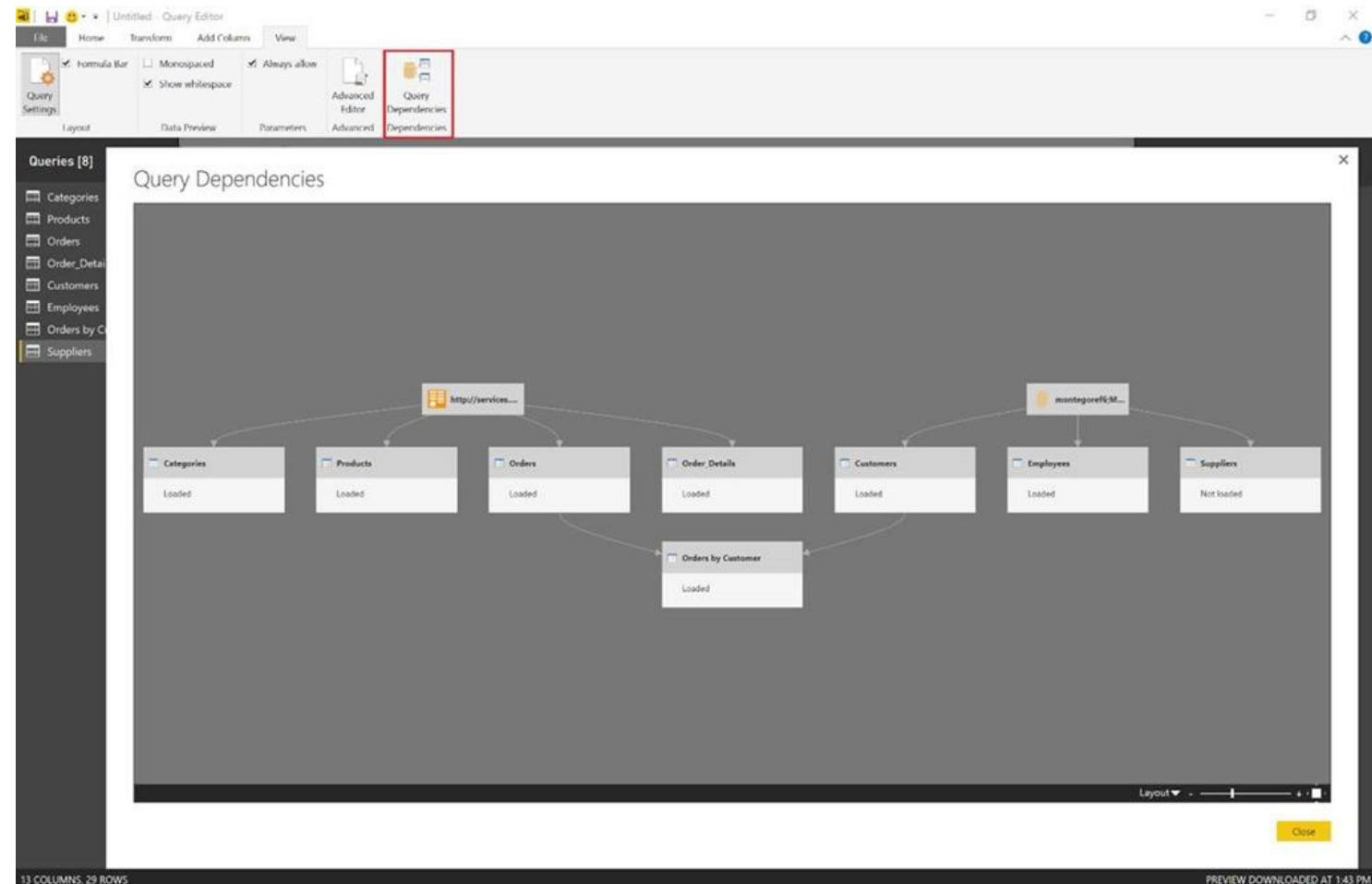
- Each parameter generates a query
- Can be referenced from other queries and loaded to the data model for use in calculations
- Support for parameters is available throughout the tool



Shaping Data

Query Dependencies view

- View dependencies across all queries and data sources



Demonstration: Functions and Iterating Files

Lab 1 Exercise 1: Shaping Data