Gpearls

EXCEL POWERBI WORKFLOW

EXPLORING WHAT POWERBI CAN DO IN EXCEL

28TH NOVEMBER, 2020 5:00PM

EXCEL POWERBI WORKFLOW

BUSINESS INTELLIGENCE WORKFLOW TOOLS ARE ALL AVAILABLE DIRECTLY IN EXCEL

COURSE OUTLINE

1 Power Excel

- *Power Query/Power Pivot workflow and key benefits vs. “traditional” Excel*

2 Power Query

- *Types of data connectors, query editing tools, loading options, etc.*

3 Data Model

- *Excel Data Model interface, normalization, table relationships, hierarchies, etc.*

4 Power Pivot

- *Power Pivots vs. “normal” pivots, calculated columns vs. measures, row & filter context, etc.*

5 Common DAX Functions

- *Basic syntax, math & stats functions, filter functions, time intelligence tools, etc.*

"THE BEST THING TO HAPPEN TO EXCEL IN 20 YEARS"

**Quote by Bill Jelen (aka "Mr. Excel")*

- **Import and analyze MILLIONS of rows of data in Excel**
  - *Access data from virtually anywhere (database tables, flat files, cloud services, folders, etc.)*
- **Quickly build models to blend and analyze data across sources**
  - *Instantly connect sources and analyze holistic performance across your entire data model*
- **Create fully automated data shaping and loading procedures**
  - *Connect to databases and watch data flow through your model with the click of a button*
- **Define calculated measures using Data Analysis Expressions (DAX)**
  - *No more redundant A1-style "grid" formulas; DAX expressions are flexible, powerful and portable*

WHEN TO USE POWER QUERY & POWER PIVOT

*Use **Power Query** and **Power Pivot** when you want to…*

1 Analyze more data than can fit into a worksheet

2 Create connections to databases or external sources

3 Blend data across multiple large tables

4 Automate the process of loading and shaping your data

5 Unleash the **full business intelligence capabilities** of Excel

EXCEL POWERBI WORKFLOW

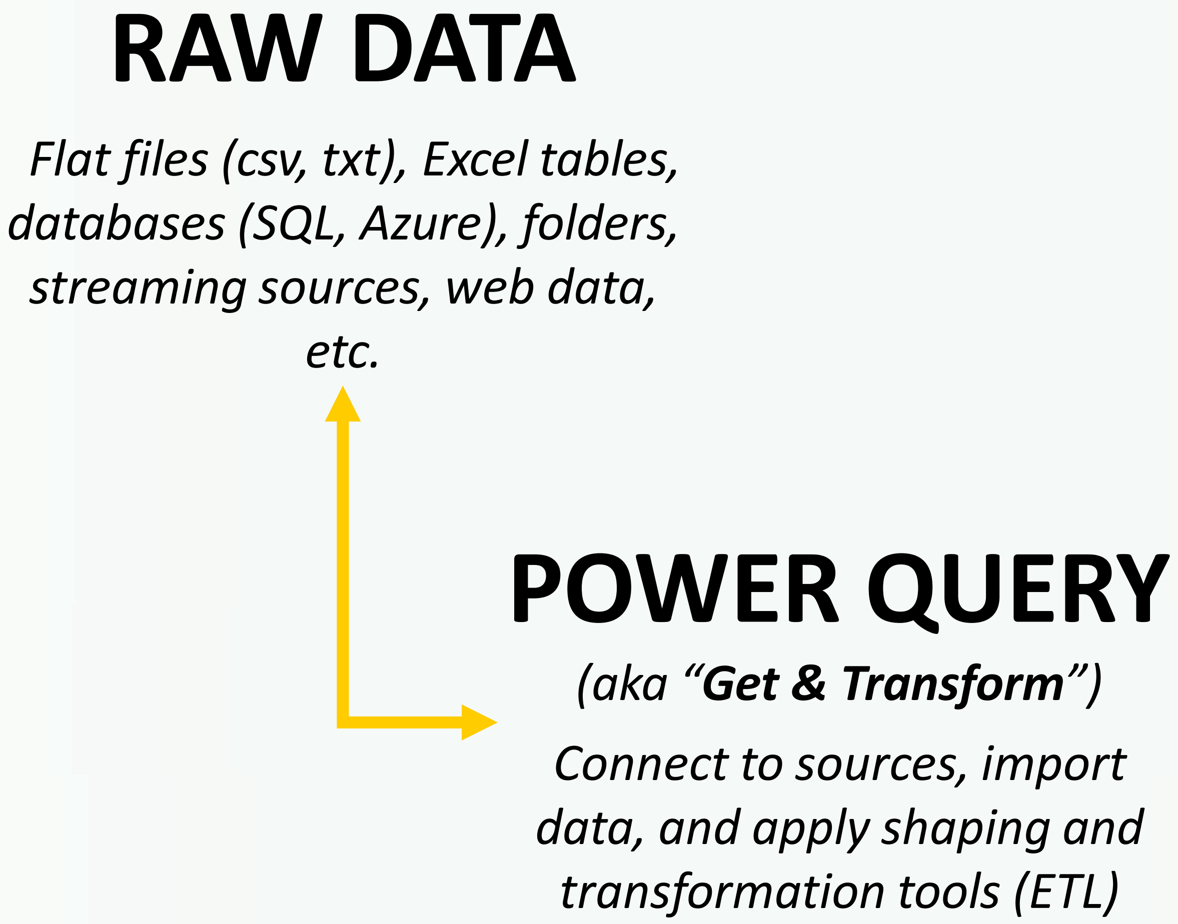
RAW DATA

Flat files (csv, txt), Excel tables, databases (SQL, Azure), folders, streaming sources, web data, etc.

Business Intelligence workflow tools are all available directly in Excel

*Copyright 2020, Gpearls Data Analytics,
https://gpearls.com.ng

EXCEL POWERBI WORKFLOW



RAW DATA

Flat files (csv, txt), Excel tables, databases (SQL, Azure), folders, streaming sources, web data, etc.

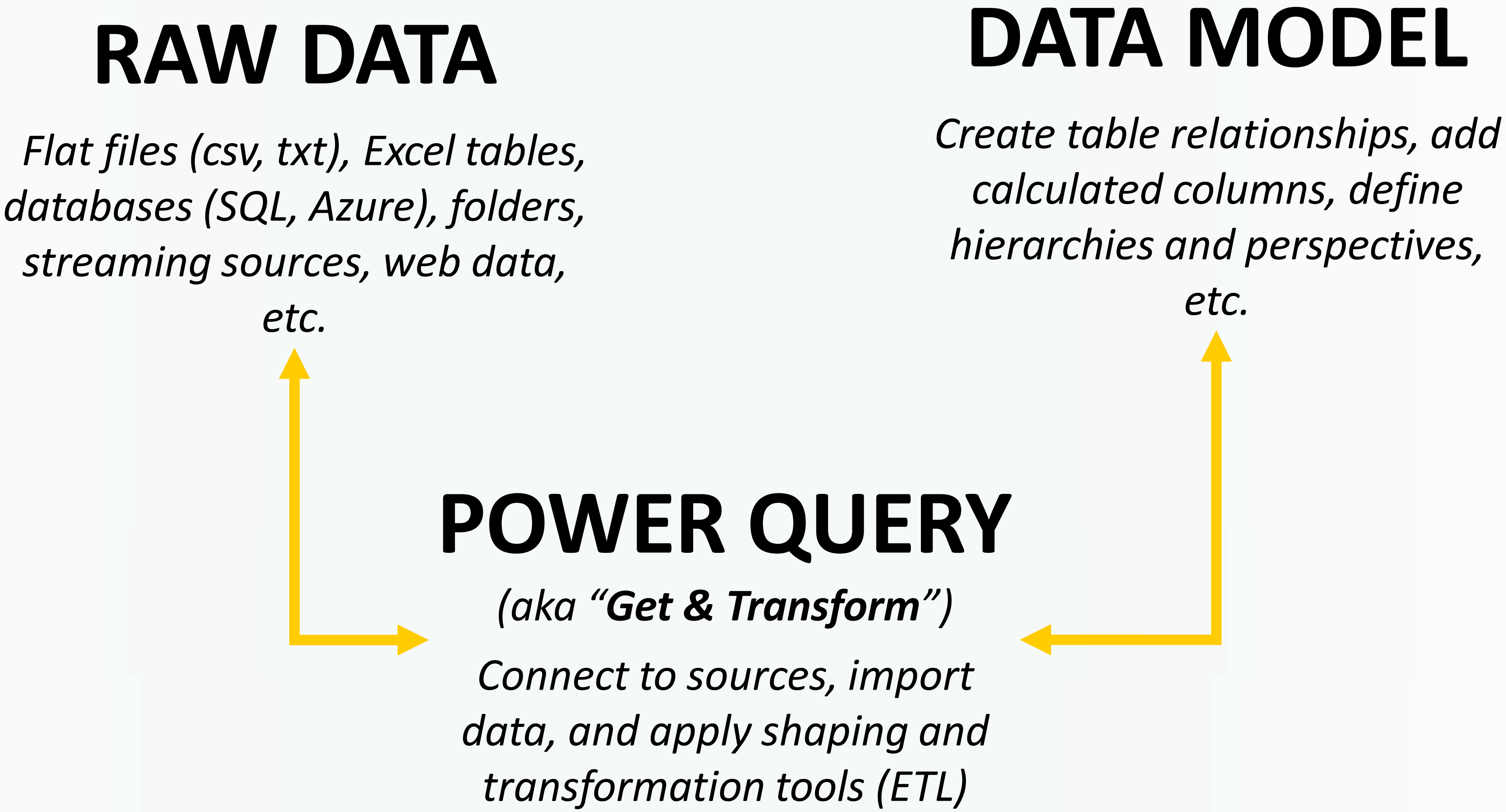
POWER QUERY

*(aka “**Get & Transform**”)*

Connect to sources, import data, and apply shaping and transformation tools (ETL)

Business Intelligence workflow tools are all available directly in Excel

EXCEL POWERBI WORKFLOW



RAW DATA

Flat files (csv, txt), Excel tables, databases (SQL, Azure), folders, streaming sources, web data, etc.

DATA MODEL

Create table relationships, add calculated columns, define hierarchies and perspectives, etc.

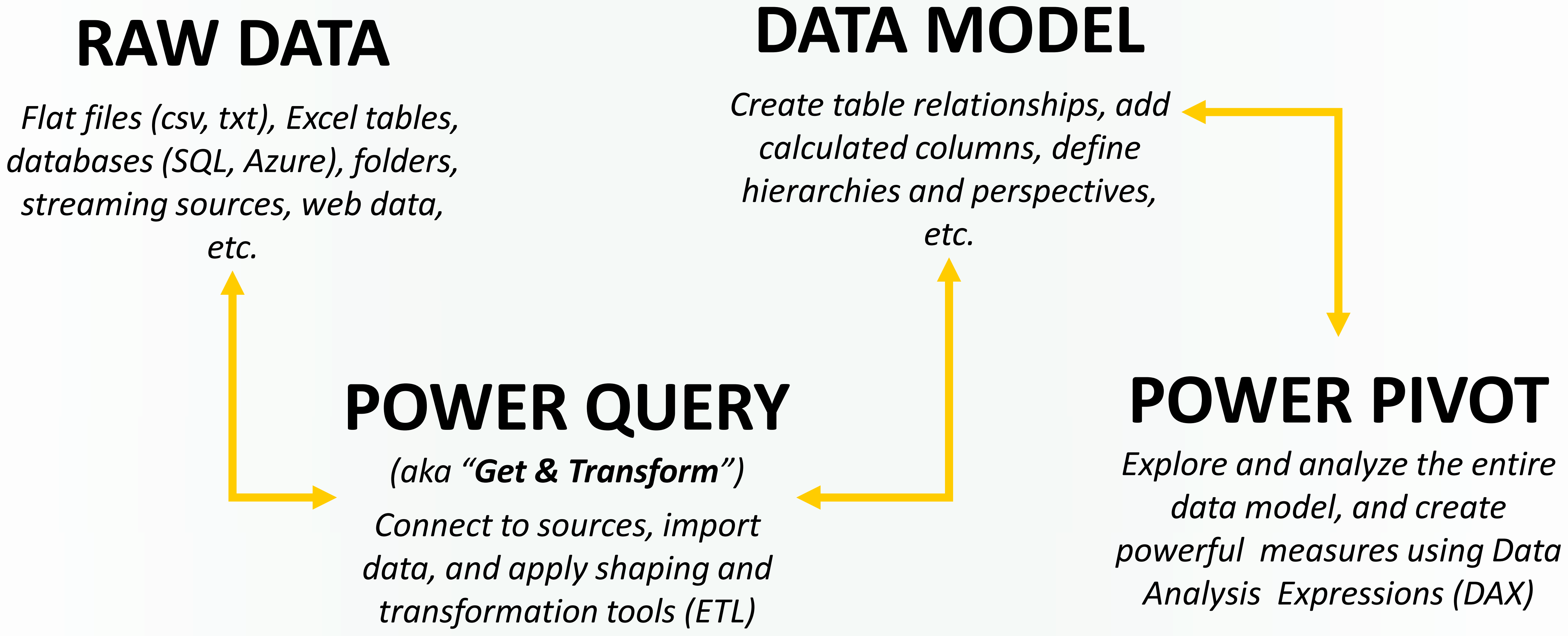
POWER QUERY

*(aka “**Get & Transform**”)*

Connect to sources, import data, and apply shaping and transformation tools (ETL)

Business Intelligence workflow tools are all available directly in Excel

EXCEL POWERBI WORKFLOW



RAW DATA

Flat files (csv, txt), Excel tables, databases (SQL, Azure), folders, streaming sources, web data, etc.

DATA MODEL

Create table relationships, add calculated columns, define hierarchies and perspectives, etc.

POWER QUERY

*(aka “**Get & Transform**”)*

Connect to sources, import data, and apply shaping and transformation tools (ETL)

POWER PIVOT

Explore and analyze the entire data model, and create powerful measures using Data Analysis Expressions (DAX)

Business Intelligence workflow tools are all available directly in Excel

EXCEL POWERBI WORKFLOW

NORMAL PIVOT

POWER PIVOT

EXCEL POWERBI WORKFLOW

NORMAL PIVOT

- Can analyze data from **one table at a time**; multiple tables must be flattened or "stitched" together with cell functions

POWER PIVOT

EXCEL POWERBI WORKFLOW

NORMAL PIVOT

- Can analyze data from **one table at a time**; multiple tables must be flattened or “stitched” together with cell functions

POWER PIVOT

- Can analyze an **entire data model**, consisting of multiple tables connected via relationships rather than cell functions

Gpearls

EXCEL POWERBI WORKFLOW

NORMAL PIVOT

- Can analyze data from **one table at a time**; multiple tables must be flattened or "stitched" together with cell functions

- Restricted to the data capacity of a **single Excel worksheet** (1,048,576 rows)

POWER PIVOT

- Can analyze an **entire data model**, consisting of multiple tables connected via relationships rather than cell functions

EXCEL POWERBI WORKFLOW

NORMAL PIVOT

- Can analyze data from **one table at a time**; multiple tables must be flattened or “stitched” together with cell functions

- Restricted to the data capacity of a **single Excel worksheet** (1,048,576 rows)

POWER PIVOT

- Can analyze an **entire data model**, consisting of multiple tables connected via relationships rather than cell functions

- Virtually **unlimited data capacity** as tables are compressed outside of normal worksheets

EXCEL POWERBI WORKFLOW

NORMAL PIVOT

- Can analyze data from **one table at a time**; multiple tables must be flattened or “stitched” together with cell functions
- Restricted to the data capacity of a **single Excel worksheet** (1,048,576 rows)
- Limited to relatively **basic calculated fields**, using a sub-set of Excel functions

POWER PIVOT

- Can analyze an **entire data model**, consisting of multiple tables connected via relationships rather than cell functions
- Virtually **unlimited data capacity** as tables are compressed outside of normal worksheets

EXCEL POWERBI WORKFLOW

NORMAL PIVOT

- Can analyze data from **one table at a time**; multiple tables must be flattened or "stitched" together with cell functions

- Restricted to the data capacity of a **single Excel worksheet** (1,048,576 rows)

- Limited to relatively **basic calculated fields**, using a sub-set of Excel functions

POWER PIVOT

- Can analyze an **entire data model**, consisting of multiple tables connected via relationships rather than cell functions

- Virtually **unlimited data capacity** as tables are compressed outside of normal worksheets

- Performs **complex calculations** using Data Analysis Expressions (DAX)

EXCEL POWERBI WORKFLOW

NORMAL PIVOT

- Can analyze data from **one table at a time**; multiple tables must be flattened or "stitched" together with cell functions
- Restricted to the data capacity of a **single Excel worksheet** (1,048,576 rows)
- Limited to relatively **basic calculated fields**, using a sub-set of Excel functions

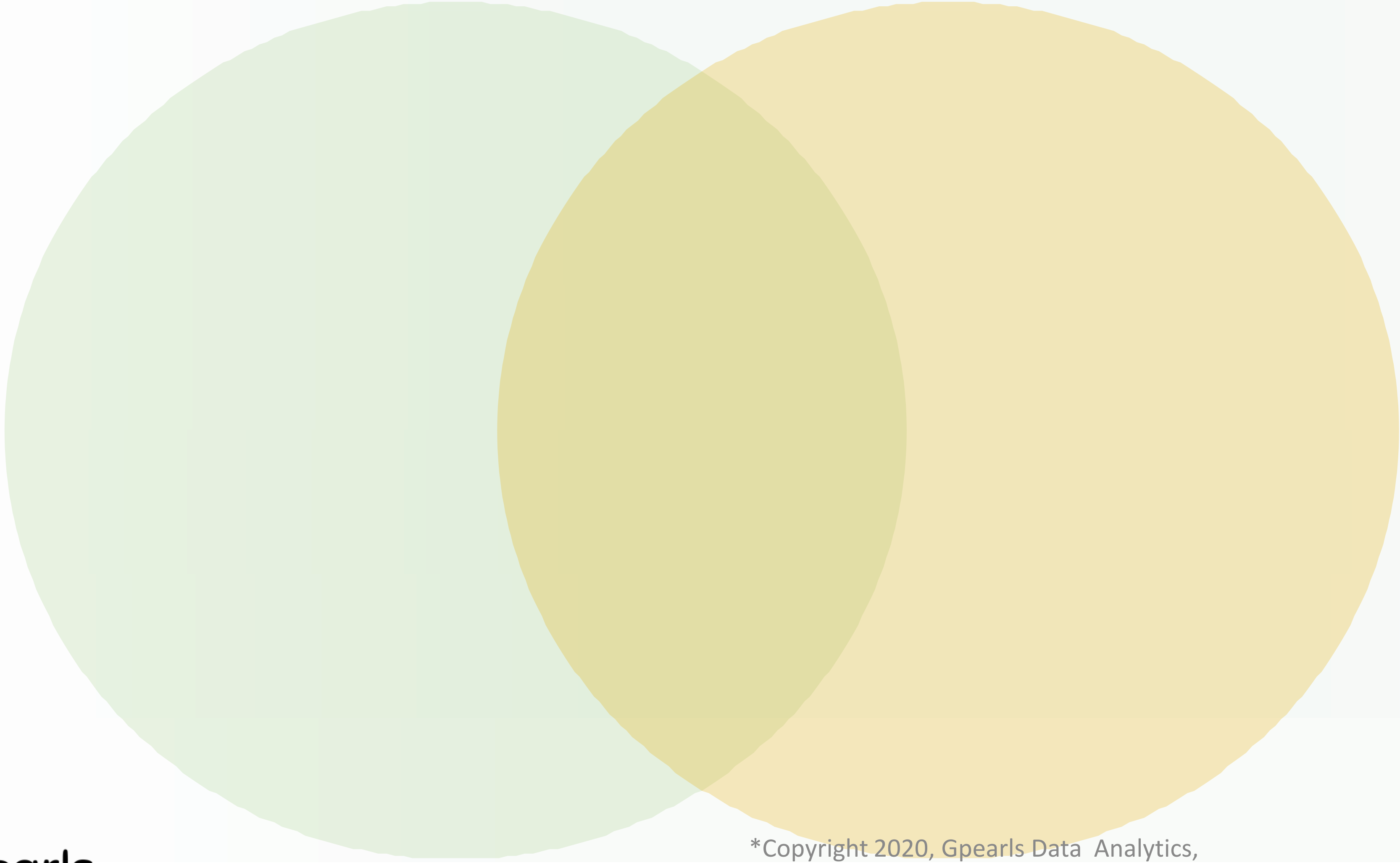
POWER PIVOT

- Can analyze an **entire data model**, consisting of multiple tables connected via relationships rather than cell functions
- Virtually **unlimited data capacity** as tables are compressed outside of normal worksheets
- Performs **complex calculations** using Data Analysis Expressions (DAX)

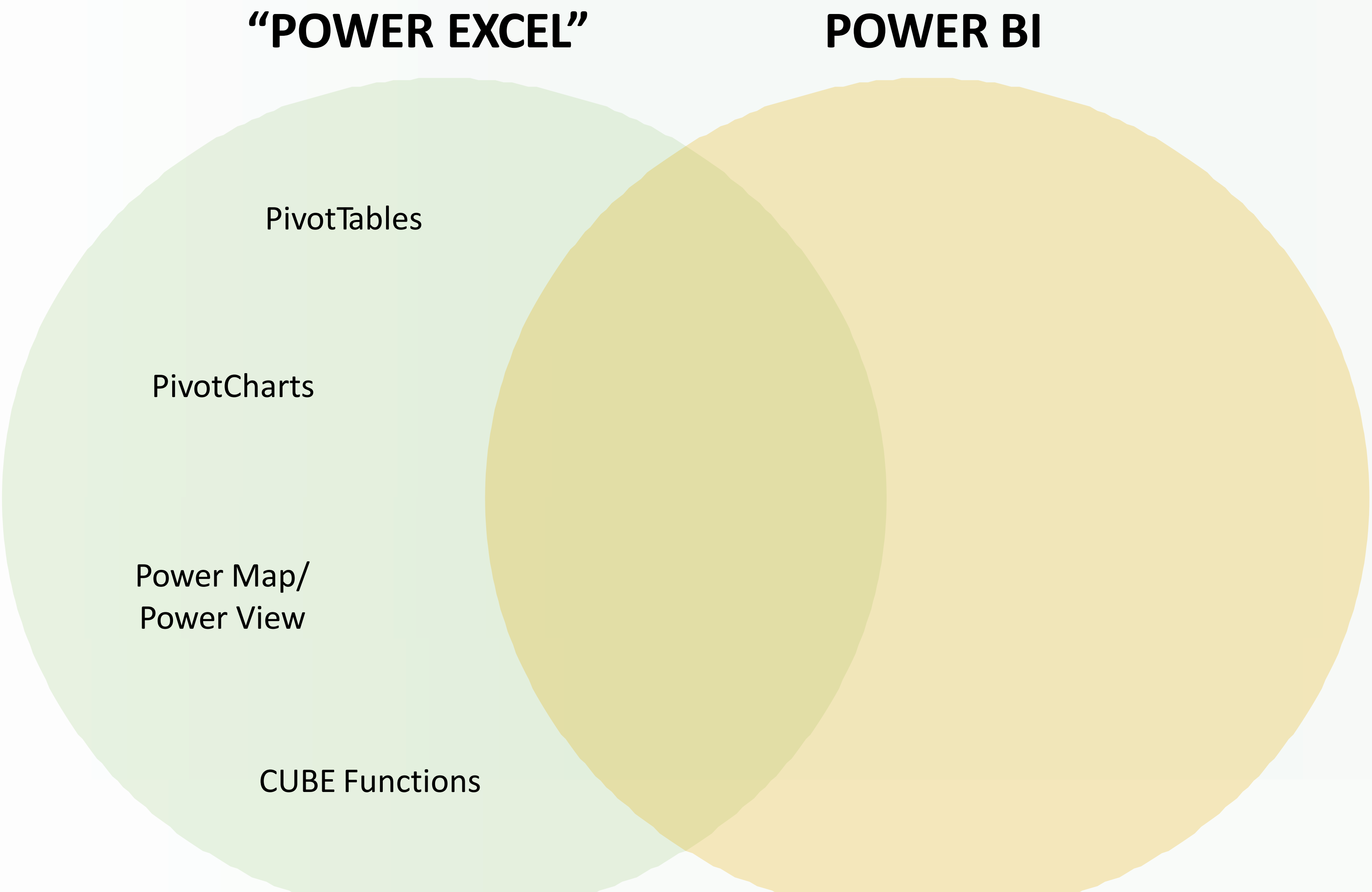
NOTE: It's not the ***PivotTable*** itself that's different; it's the ***data behind it***

POWER BI VS. “POWER EXCEL”

“POWER EXCEL” **POWER BI**

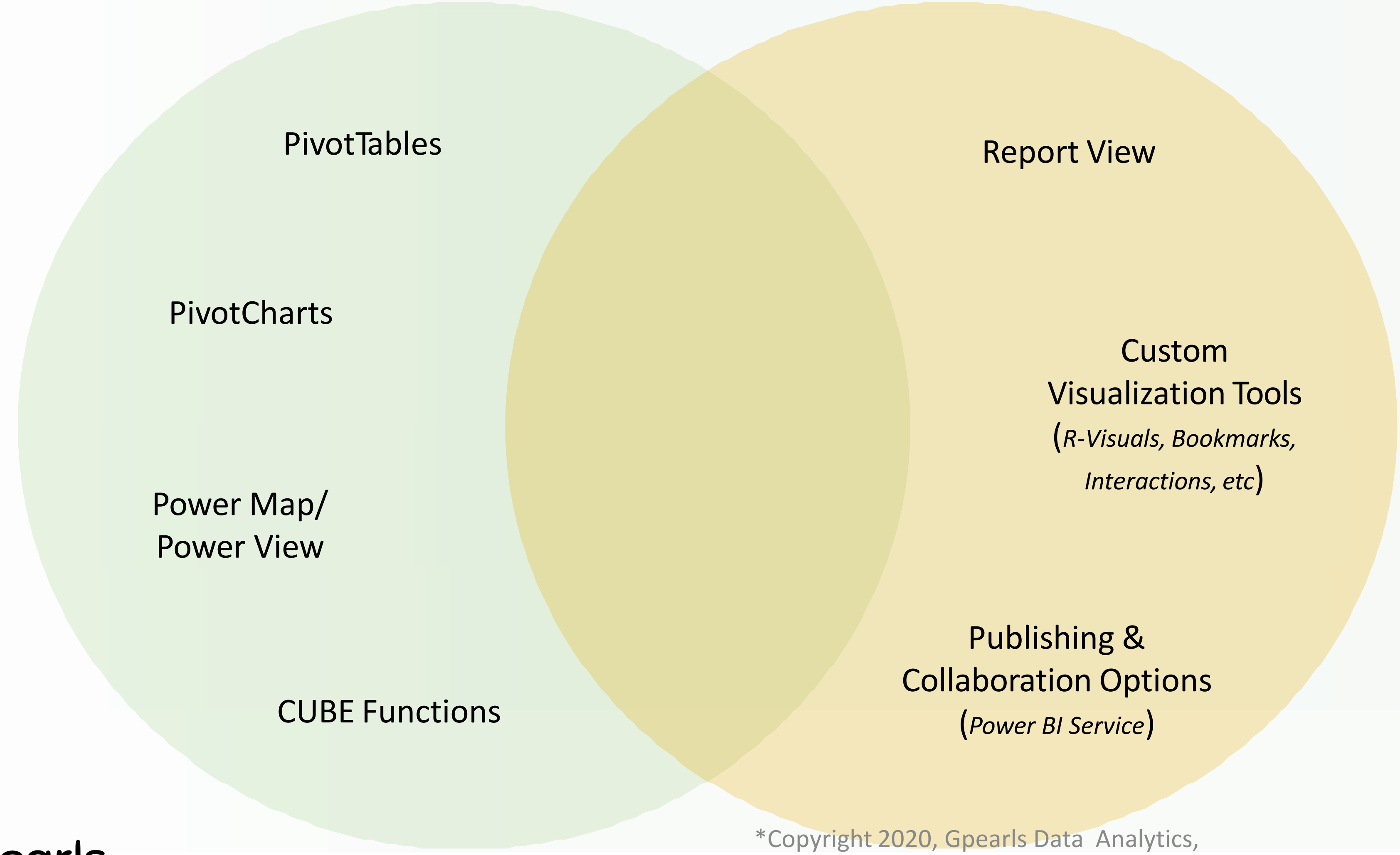


POWER BI VS. “POWER EXCEL”



POWER BI VS. “POWER EXCEL”

“POWER EXCEL” **POWER BI**



PivotTables

PivotCharts

Power Map/
Power View

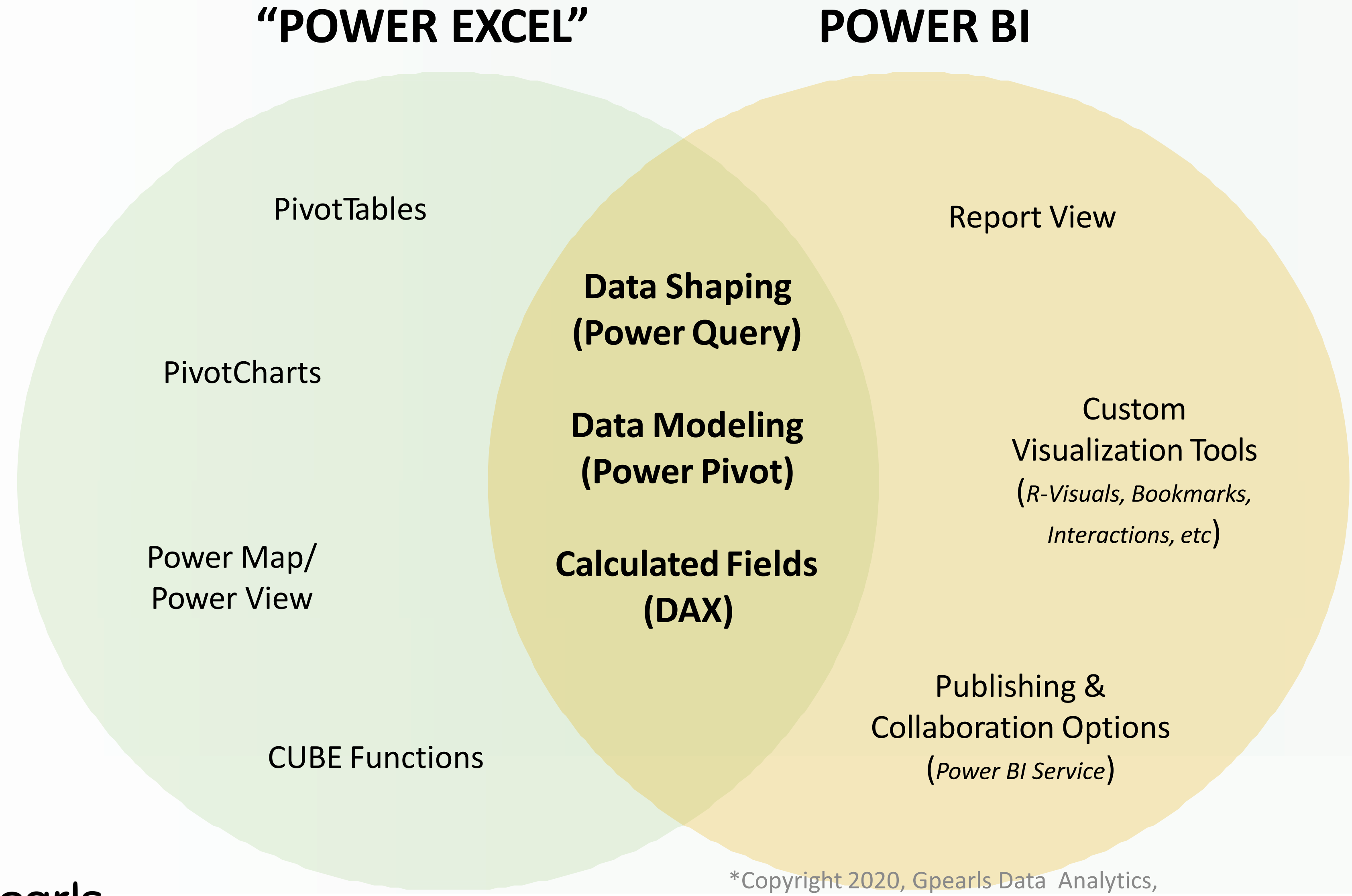
CUBE Functions

Report View

Custom
Visualization Tools
(*R-Visuals, Bookmarks, Interactions, etc*)

Publishing &
Collaboration Options
(*Power BI Service*)

POWER BI VS. “POWER EXCEL”



“POWER EXCEL”

PivotTables

PivotCharts

Power Map/
Power View

CUBE Functions

**Data Shaping
(Power Query)**

**Data Modeling
(Power Pivot)**

**Calculated Fields
(DAX)**

POWER BI

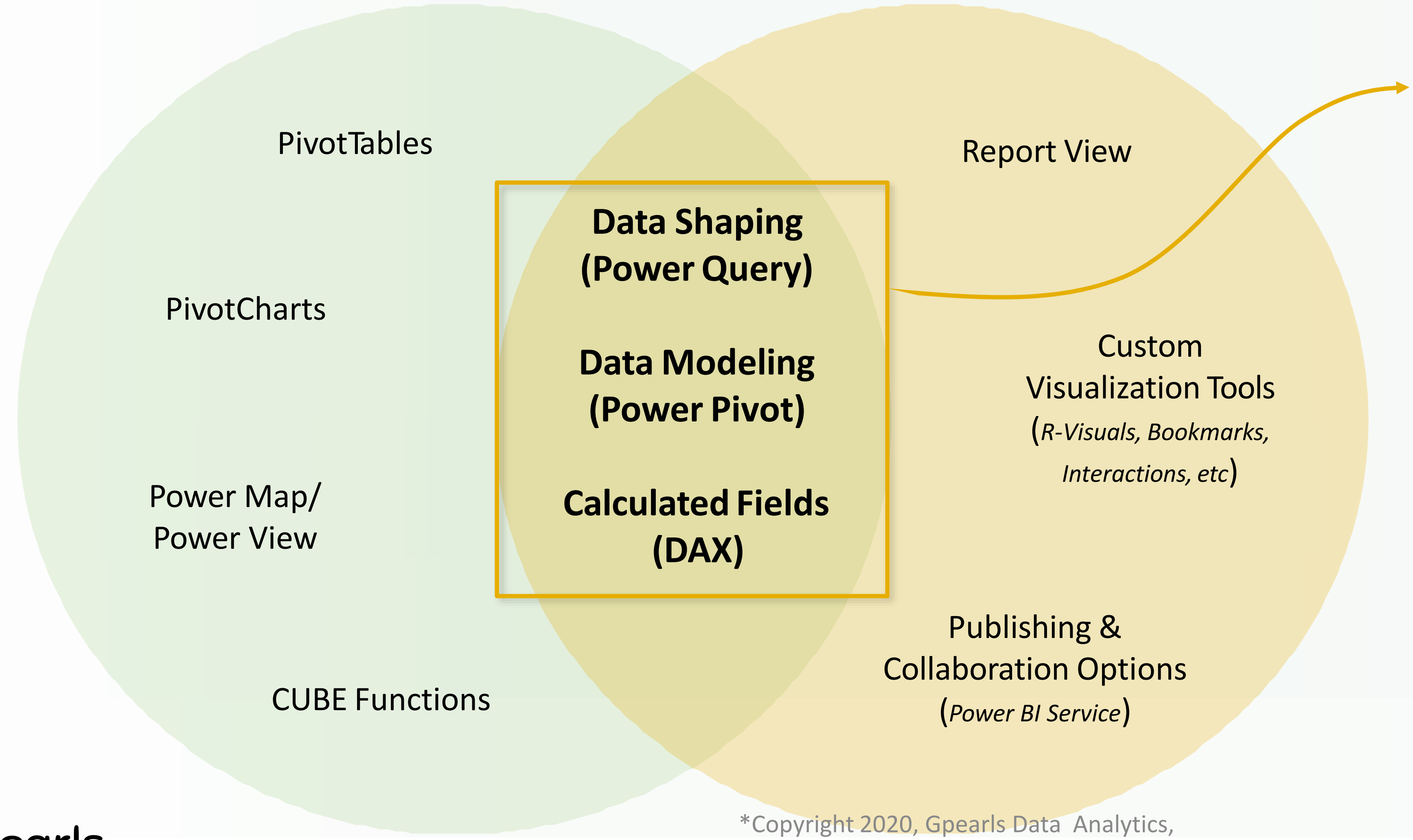
Report View

Custom
Visualization Tools
(*R-Visuals, Bookmarks,
Interactions, etc*)

Publishing &
Collaboration Options
(*Power BI Service*)

Gpearls

POWER BI VS. “POWER EXCEL”



“POWER EXCEL”

POWER BI

PivotTables

PivotCharts

Power Map/
Power View

CUBE Functions

Data Shaping (Power Query)

Data Modeling (Power Pivot)

Calculated Fields (DAX)

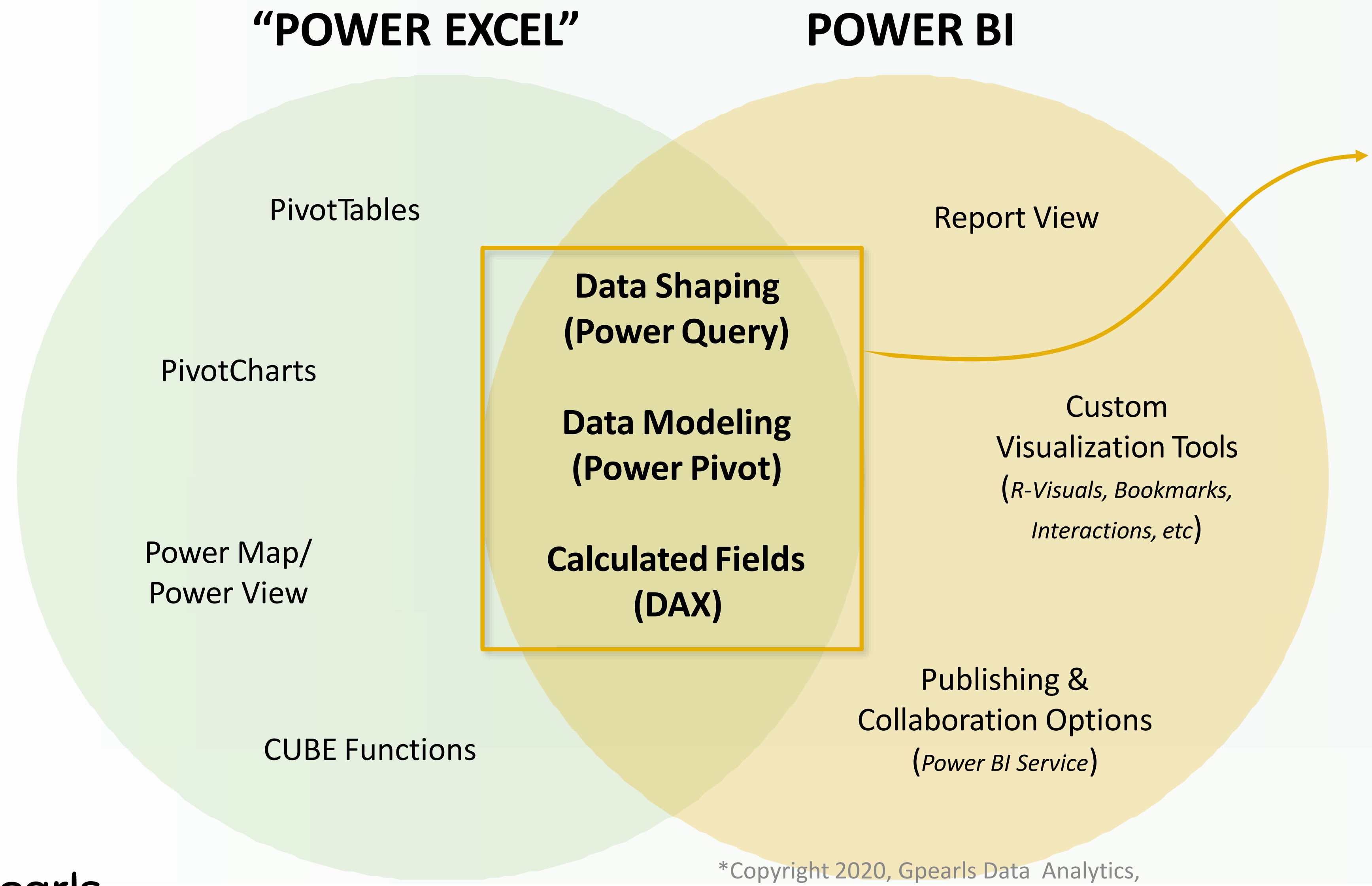
Report View

Custom Visualization Tools
(*R-Visuals, Bookmarks, Interactions, etc*)

Publishing & Collaboration Options
(*Power BI Service*)

“Power Excel” and Power BI are built on top of the ***exact same engine***!

POWER BI VS. "POWER EXCEL"



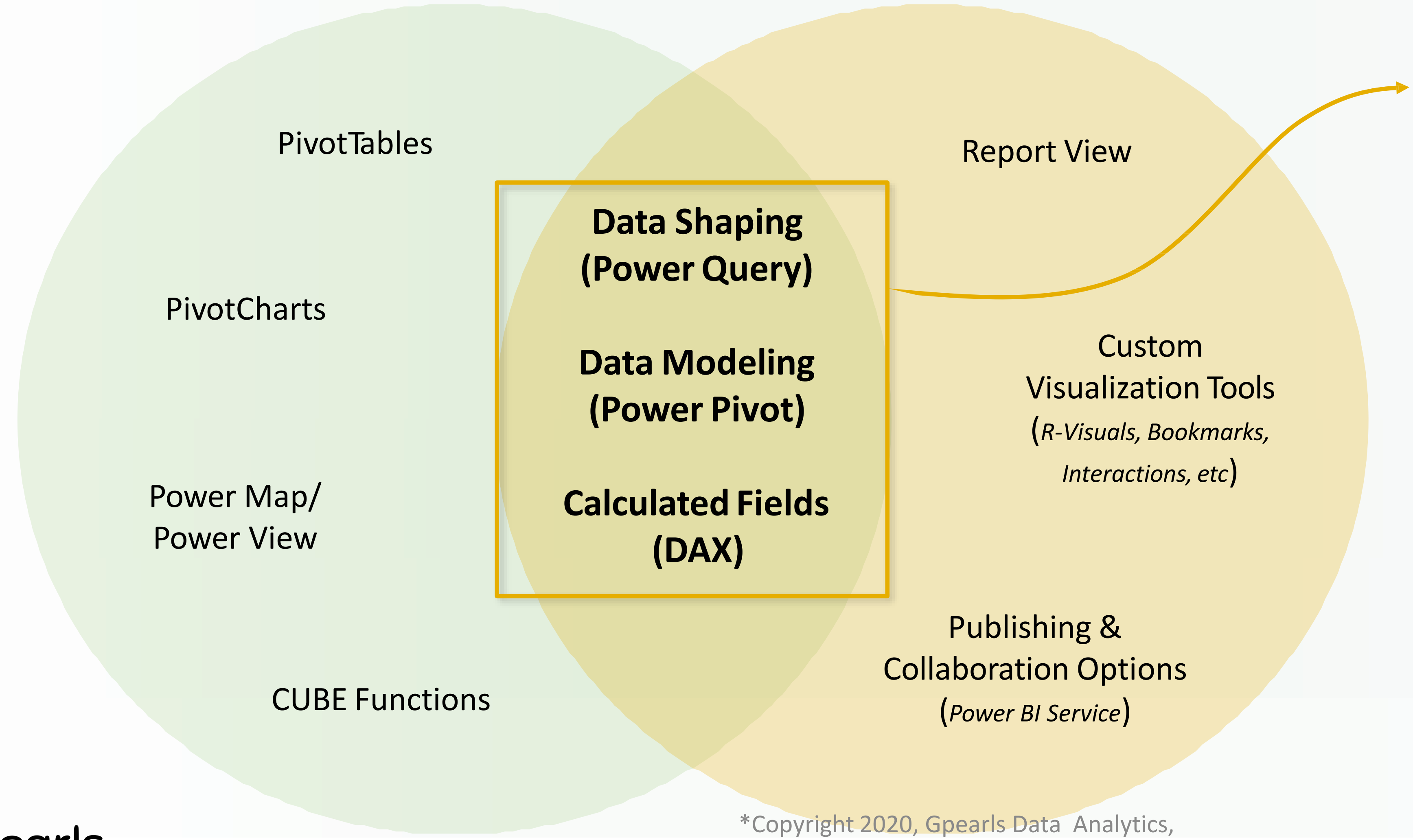
"Power Excel" and Power BI are built on top of the ***exact same engine*****!**

- Power BI takes the same data shaping, modeling and analytics capabilities and adds ***new reporting and publishing tools***

POWER BI VS. “POWER EXCEL”

“POWER EXCEL”

POWER BI



PivotTables

PivotCharts

Power Map/ Power View

CUBE Functions

Data Shaping (Power Query)

Data Modeling (Power Pivot)

Calculated Fields (DAX)

Report View

Custom Visualization Tools (*R-Visuals, Bookmarks, Interactions, etc*)

Publishing & Collaboration Options (*Power BI Service*)

“Power Excel” and Power BI are built on top of the ***exact same engine*!**

- Power BI takes the same data shaping, modeling and analytics capabilities and adds ***new reporting and publishing tools***
- Transitioning is easy; you can import an ***entire data model*** directly from Excel!

MEET POWER QUERY

MEET POWER QUERY

Power Query (aka “**Get & Transform**”) allows you to:

- Connect to data across a wide range of sources

MEET POWER QUERY

Power Query (aka “**Get & Transform**”) allows you to:

- Connect to data across a wide range of sources
- Filter, shape, append and transform raw data for further analysis and modeling

MEET POWER QUERY

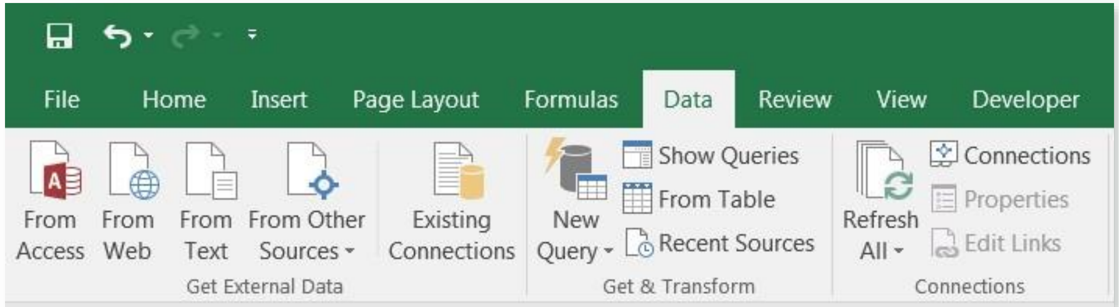
Power Query (aka "**Get & Transform**") allows you to:

- Connect to data across a wide range of sources
- Filter, shape, append and transform raw data for further analysis and modeling
- Create stored procedures to automate your data prep (*like a macro!)*

MEET POWER QUERY

Power Query (aka "**Get & Transform**") allows you to:

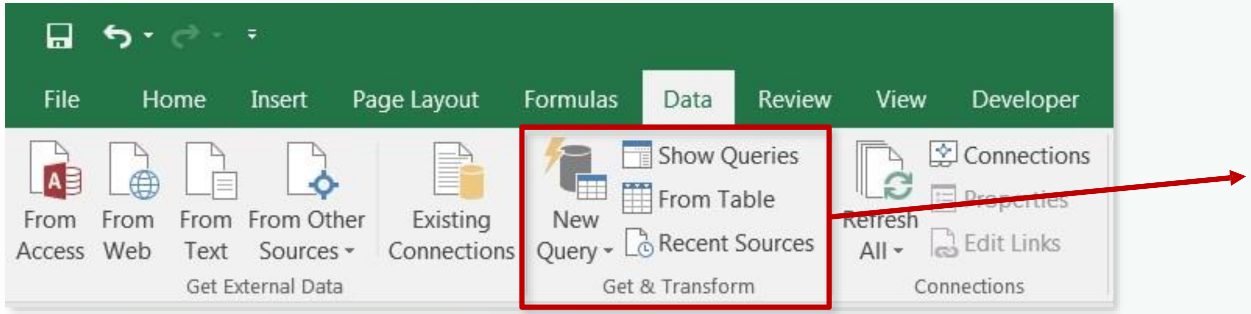
- Connect to data across a wide range of sources
- Filter, shape, append and transform raw data for further analysis and modeling
- Create stored procedures to automate your data prep (*like a macro!)*



MEET POWER QUERY

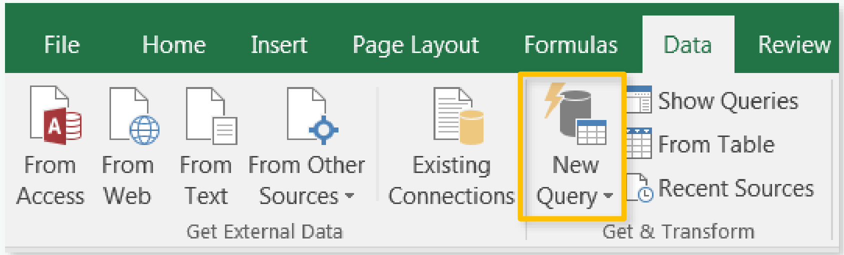
Power Query (aka "**Get & Transform**") allows you to:

- Connect to data across a wide range of sources
- Filter, shape, append and transform raw data for further analysis and modeling
- Create stored procedures to automate your data prep (*like a macro!)*

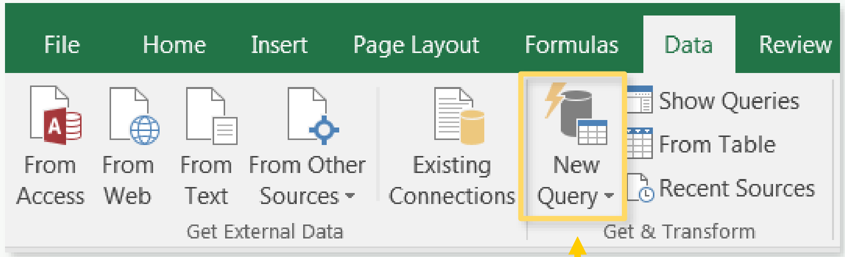


The Power Query tools live in the **Data** tab, under the "**Get & Transform**" section (Excel 2016)

TYPES OF DATA CONNECTIONS



TYPES OF DATA CONNECTIONS



From File

- From Workbook
- From CSV
- From XML
- From Text
- From JSON
- From Folder
- From SharePoint Folder

From Database

- From SQL Server Database
- From Microsoft Access Database
- From SQL Server Analysis Services Database
- From Oracle Database
- From IBM DB2 Database
- From MySQL Database
- From PostgreSQL Database
- From Sybase Database
- From Teradata Database
- From SAP HANA Database

FromAzure

- From Microsoft Azure SQL Database
- From Microsoft Azure SQL Data Warehouse
- From Microsoft Azure Marketplace
- From Microsoft Azure HDInsight
- From Microsoft Azure Blob Storage
- From Microsoft Azure Table Storage

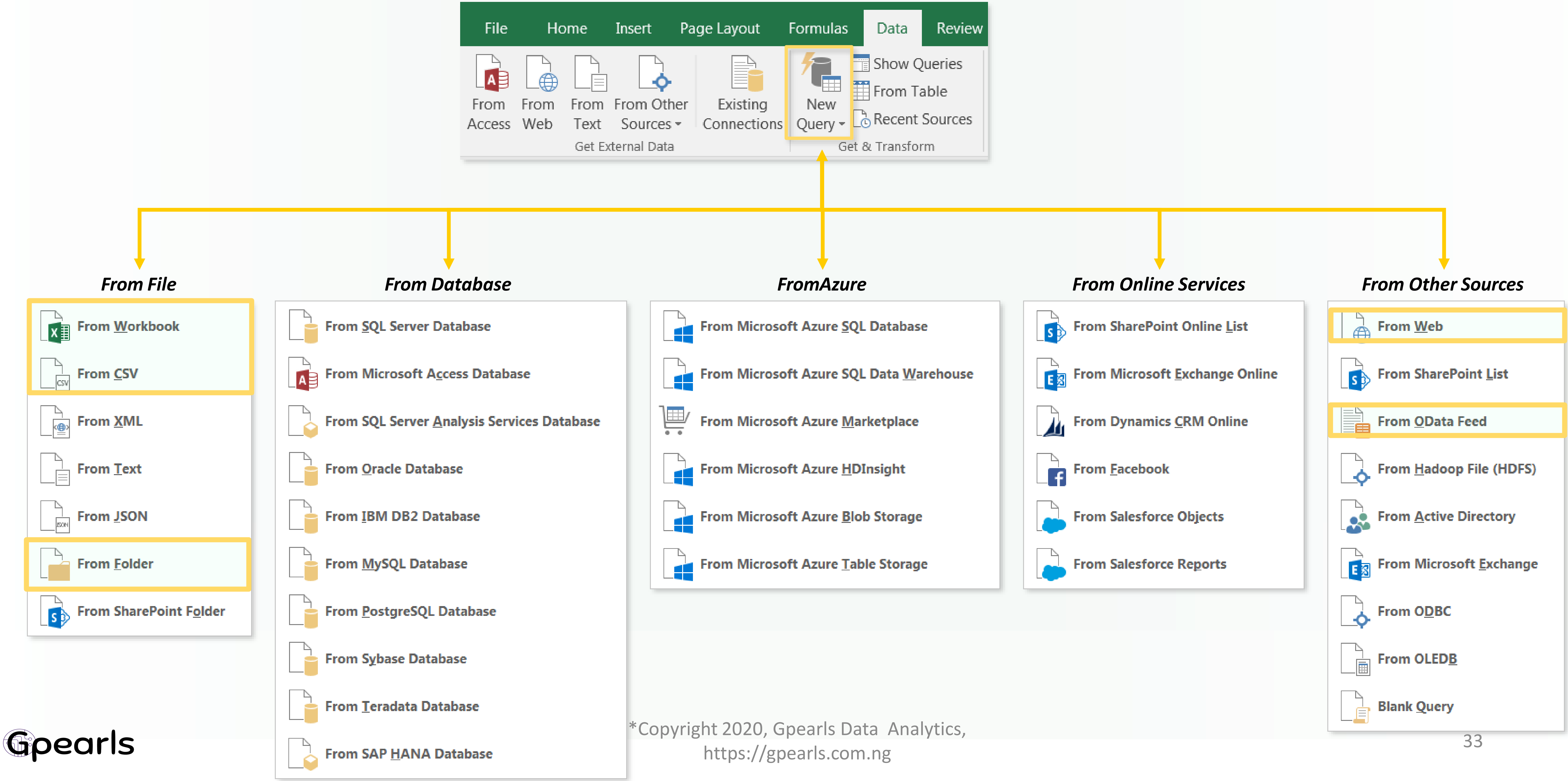
From Online Services

- From SharePoint Online List
- From Microsoft Exchange Online
- From Dynamics CRM Online
- From Facebook
- From Salesforce Objects
- From Salesforce Reports

From Other Sources

- From Web
- From SharePoint List
- From OData Feed
- From Hadoop File (HDFS)
- From Active Directory
- From Microsoft Exchange
- From ODBC
- From OLEDB
- Blank Query

TYPES OF DATA CONNECTIONS



From File

- From Workbook
- From CSV
- From XML
- From Text
- From JSON
- From Folder
- From SharePoint Folder

From Database

- From SQL Server Database
- From Microsoft Access Database
- From SQL Server Analysis Services Database
- From Oracle Database
- From IBM DB2 Database
- From MySQL Database
- From PostgreSQL Database
- From Sybase Database
- From Teradata Database
- From SAP HANA Database

FromAzure

- From Microsoft Azure SQL Database
- From Microsoft Azure SQL Data Warehouse
- From Microsoft Azure Marketplace
- From Microsoft Azure HDInsight
- From Microsoft Azure Blob Storage
- From Microsoft Azure Table Storage

From Online Services

- From SharePoint Online List
- From Microsoft Exchange Online
- From Dynamics CRM Online
- From Facebook
- From Salesforce Objects
- From Salesforce Reports

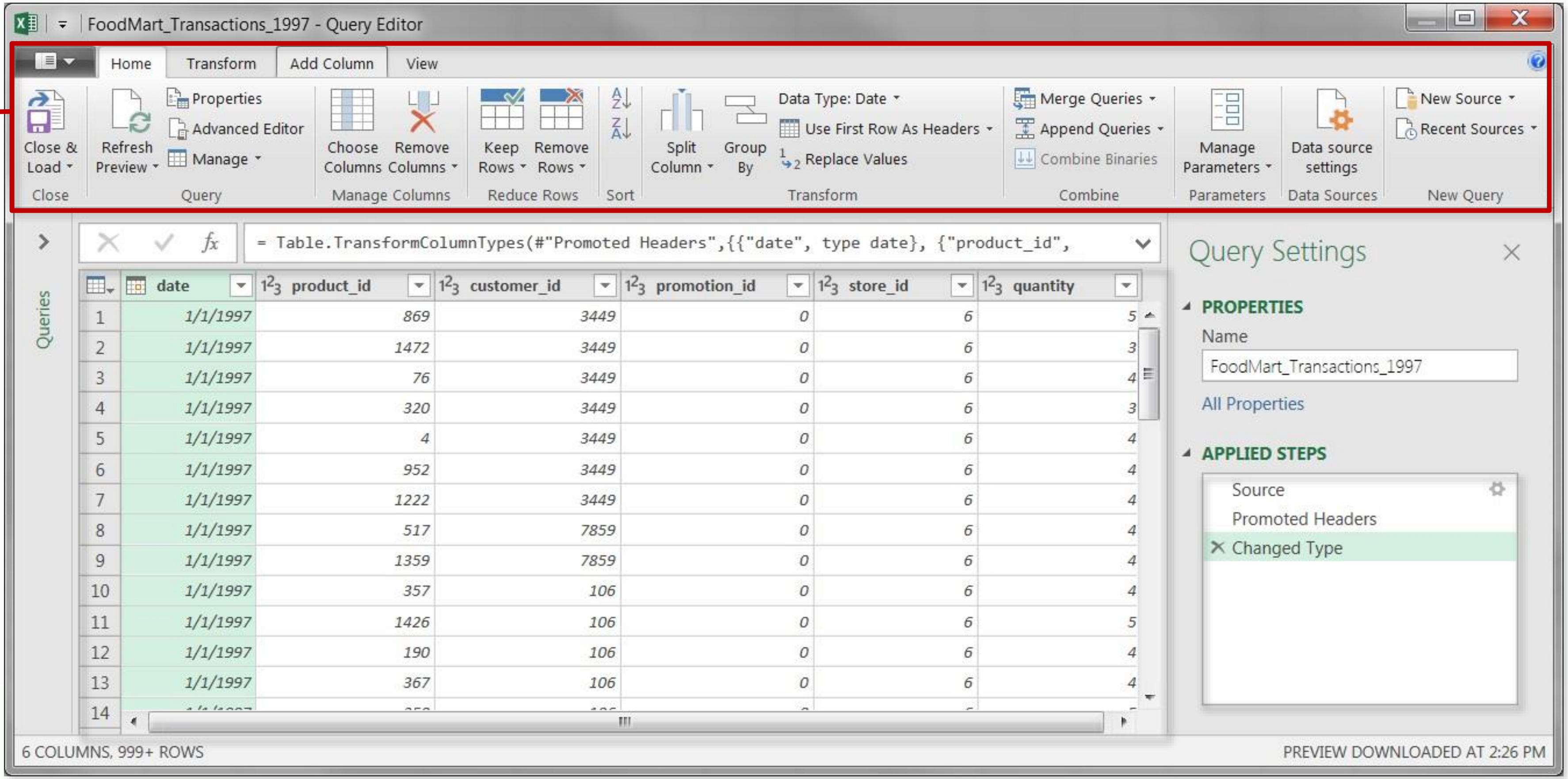
From Other Sources

- From Web
- From SharePoint List
- From OData Feed
- From Hadoop File (HDFS)
- From Active Directory
- From Microsoft Exchange
- From ODBC
- From OLEDB
- Blank Query

Gpearls

THE QUERY EDITOR

Query Editing Tools



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"date", type date}, {"product_id",
```

	date	product_id	customer_id	promotion_id	store_id	quantity
1	1/1/1997	869	3449	0	6	5
2	1/1/1997	1472	3449	0	6	3
3	1/1/1997	76	3449	0	6	4
4	1/1/1997	320	3449	0	6	3
5	1/1/1997	4	3449	0	6	4
6	1/1/1997	952	3449	0	6	4
7	1/1/1997	1222	3449	0	6	4
8	1/1/1997	517	7859	0	6	4
9	1/1/1997	1359	7859	0	6	4
10	1/1/1997	357	106	0	6	4
11	1/1/1997	1426	106	0	6	5
12	1/1/1997	190	106	0	6	4
13	1/1/1997	367	106	0	6	4
14						

Name: FoodMart_Transactions_1997

- Source
- Promoted Headers
- Changed Type

6 COLUMNS, 999+ ROWS

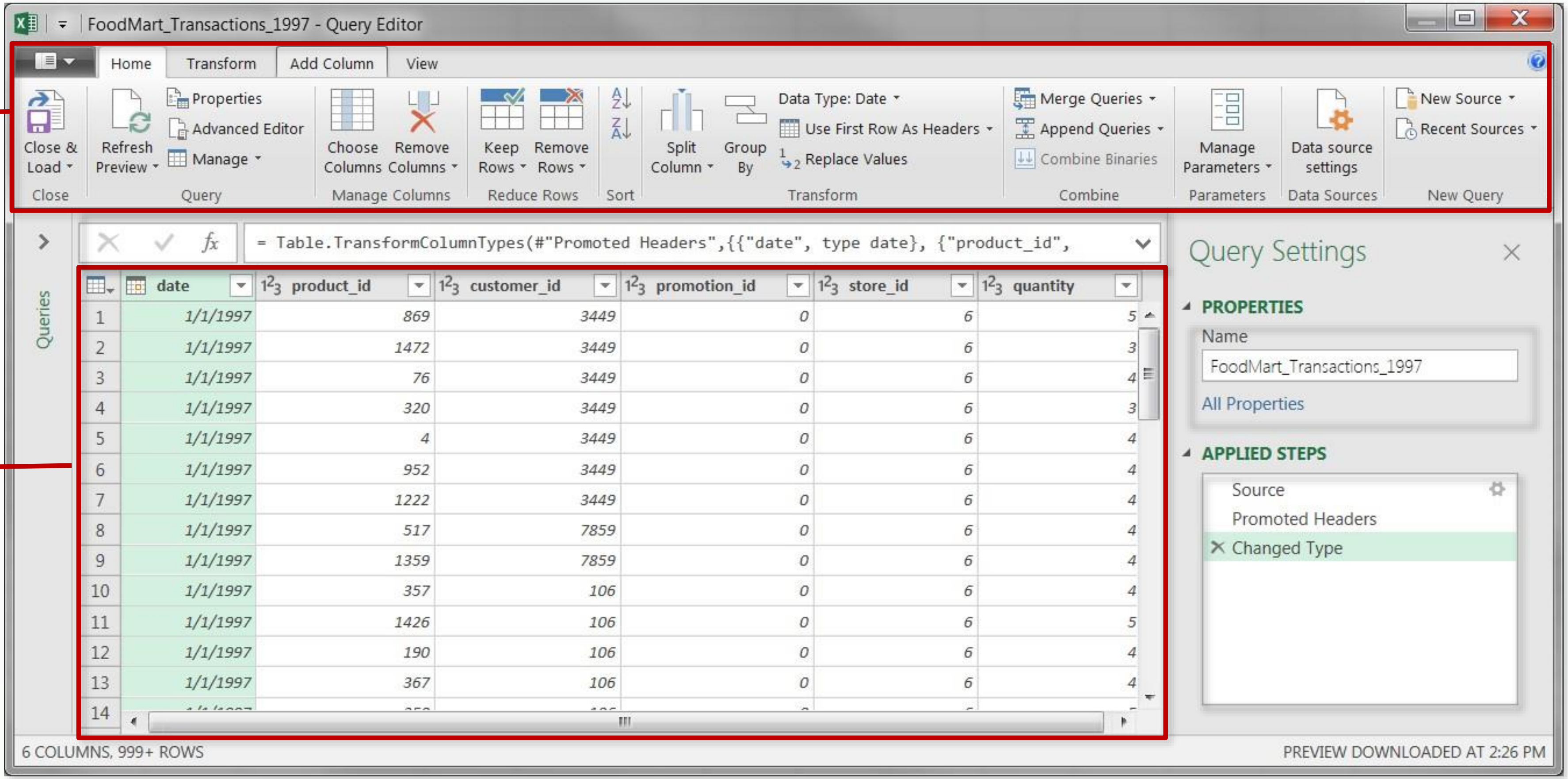
PREVIEW DOWNLOADED AT 2:26 PM

Gpearls

THE QUERY EDITOR

Query Editing Tools

Data Preview



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"date", type date}, {"product_id",
```

	date	product_id	customer_id	promotion_id	store_id	quantity
1	1/1/1997	869	3449	0	6	5
2	1/1/1997	1472	3449	0	6	3
3	1/1/1997	76	3449	0	6	4
4	1/1/1997	320	3449	0	6	3
5	1/1/1997	4	3449	0	6	4
6	1/1/1997	952	3449	0	6	4
7	1/1/1997	1222	3449	0	6	4
8	1/1/1997	517	7859	0	6	4
9	1/1/1997	1359	7859	0	6	4
10	1/1/1997	357	106	0	6	4
11	1/1/1997	1426	106	0	6	5
12	1/1/1997	190	106	0	6	4
13	1/1/1997	367	106	0	6	4
14						

PROPERTIES

Name: FoodMart_Transactions_1997

All Properties

APPLIED STEPS

- Source
- Promoted Headers
- Changed Type

6 COLUMNS, 999+ ROWS

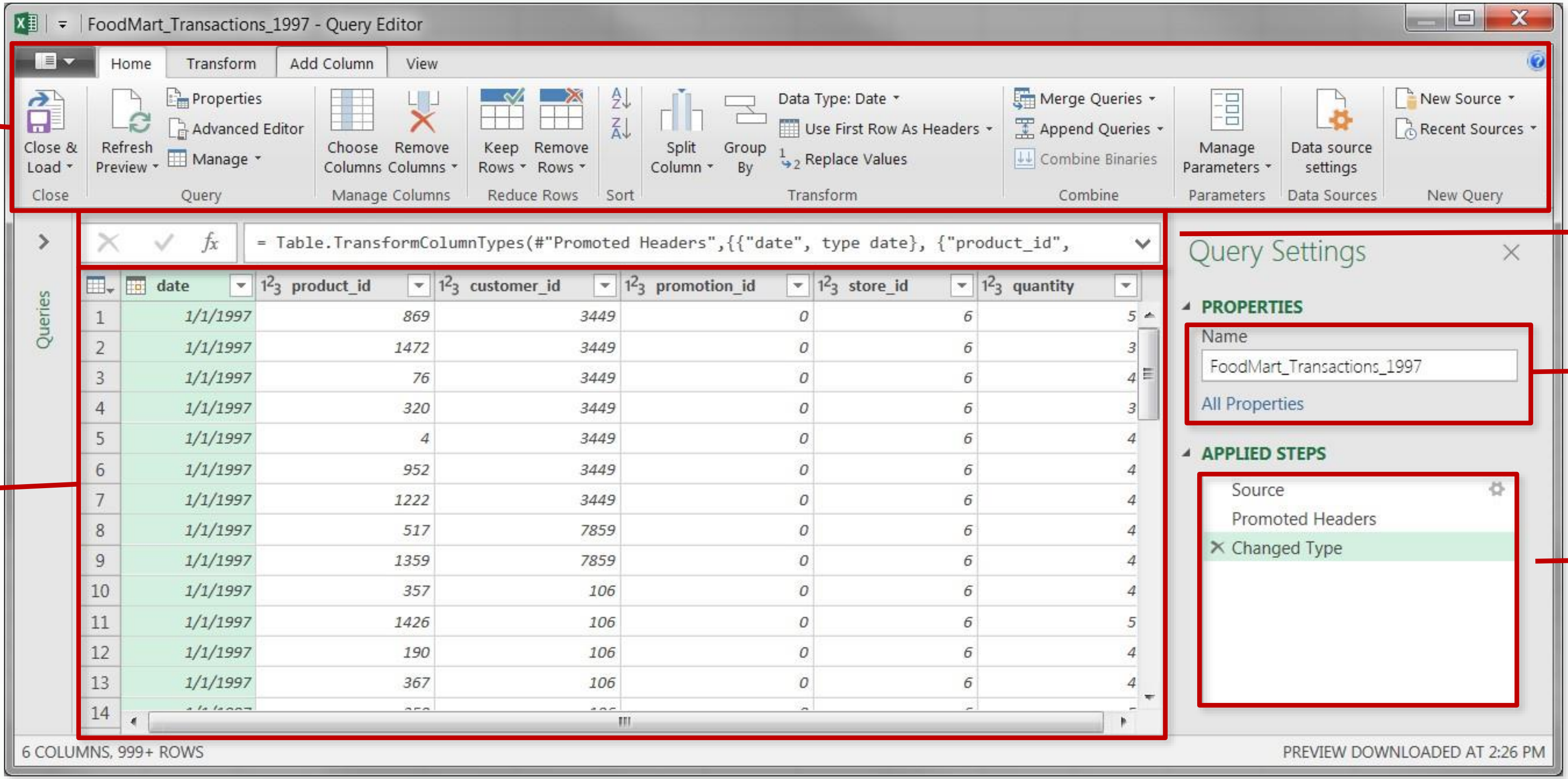
PREVIEW DOWNLOADED AT 2:26 PM

THE QUERY EDITOR

Query Editing Tools

Data Preview

Formula Bar
(this is "M" code)



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"date", type date}, {"product_id",
```

	date	product_id	customer_id	promotion_id	store_id	quantity
1	1/1/1997	869	3449	0	6	5
2	1/1/1997	1472	3449	0	6	3
3	1/1/1997	76	3449	0	6	4
4	1/1/1997	320	3449	0	6	3
5	1/1/1997	4	3449	0	6	4
6	1/1/1997	952	3449	0	6	4
7	1/1/1997	1222	3449	0	6	4
8	1/1/1997	517	7859	0	6	4
9	1/1/1997	1359	7859	0	6	4
10	1/1/1997	357	106	0	6	4
11	1/1/1997	1426	106	0	6	5
12	1/1/1997	190	106	0	6	4
13	1/1/1997	367	106	0	6	4
14						

Name: FoodMart_Transactions_1997

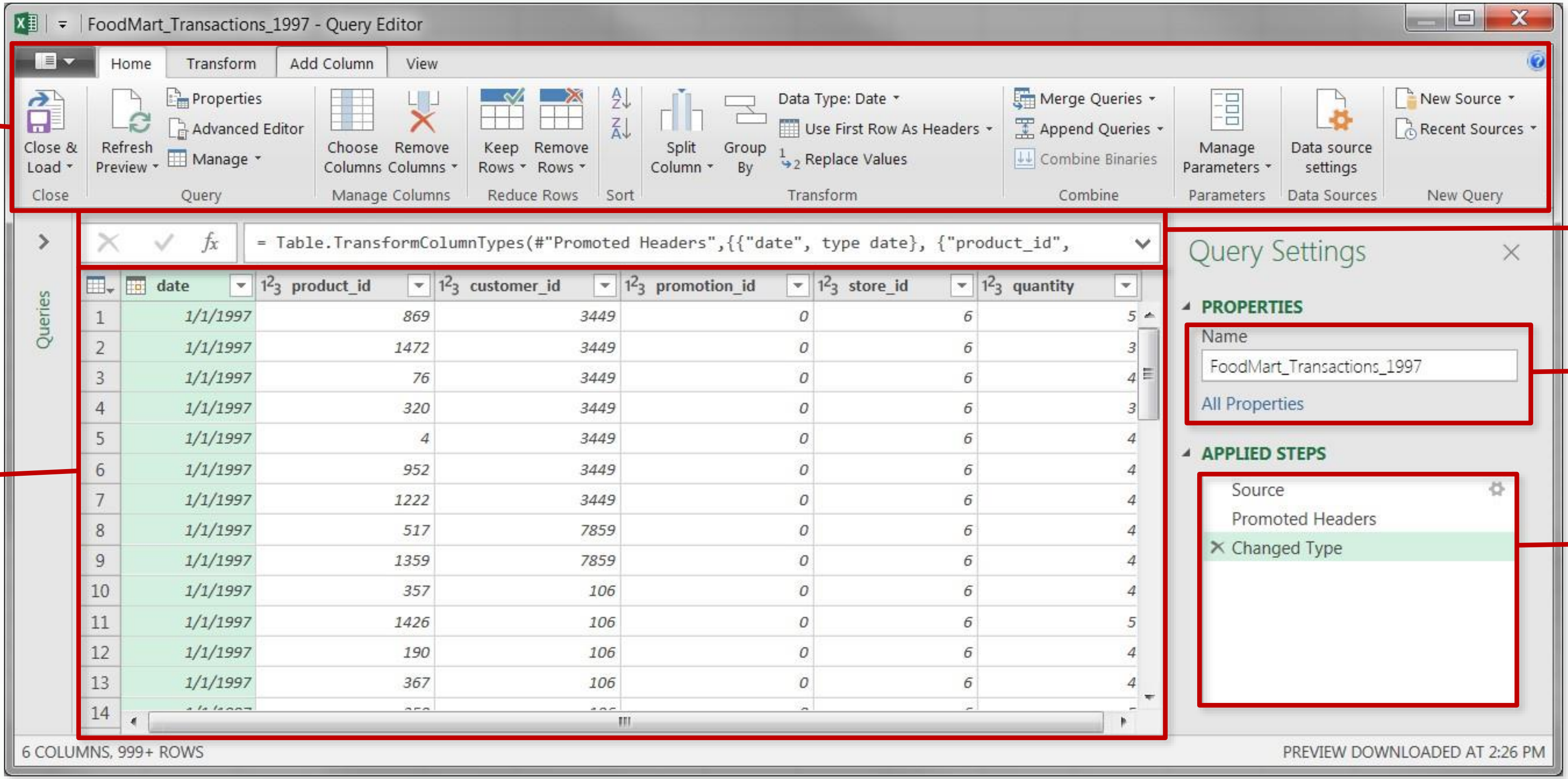
- Source
- Promoted Headers
- Changed Type

6 COLUMNS, 999+ ROWS

PREVIEW DOWNLOADED AT 2:26 PM

THE QUERY EDITOR

Query Editing Tools



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"date", type date}, {"product_id",
```

	date	product_id	customer_id	promotion_id	store_id	quantity
1	1/1/1997	869	3449	0	6	5
2	1/1/1997	1472	3449	0	6	3
3	1/1/1997	76	3449	0	6	4
4	1/1/1997	320	3449	0	6	3
5	1/1/1997	4	3449	0	6	4
6	1/1/1997	952	3449	0	6	4
7	1/1/1997	1222	3449	0	6	4
8	1/1/1997	517	7859	0	6	4
9	1/1/1997	1359	7859	0	6	4
10	1/1/1997	357	106	0	6	4
11	1/1/1997	1426	106	0	6	5
12	1/1/1997	190	106	0	6	4
13	1/1/1997	367	106	0	6	4
14						

Name: FoodMart_Transactions_1997

Applied steps:
- Source
- Promoted Headers
- Changed Type

6 COLUMNS, 999+ ROWS

PREVIEW DOWNLOADED AT 2:26 PM

Data Preview

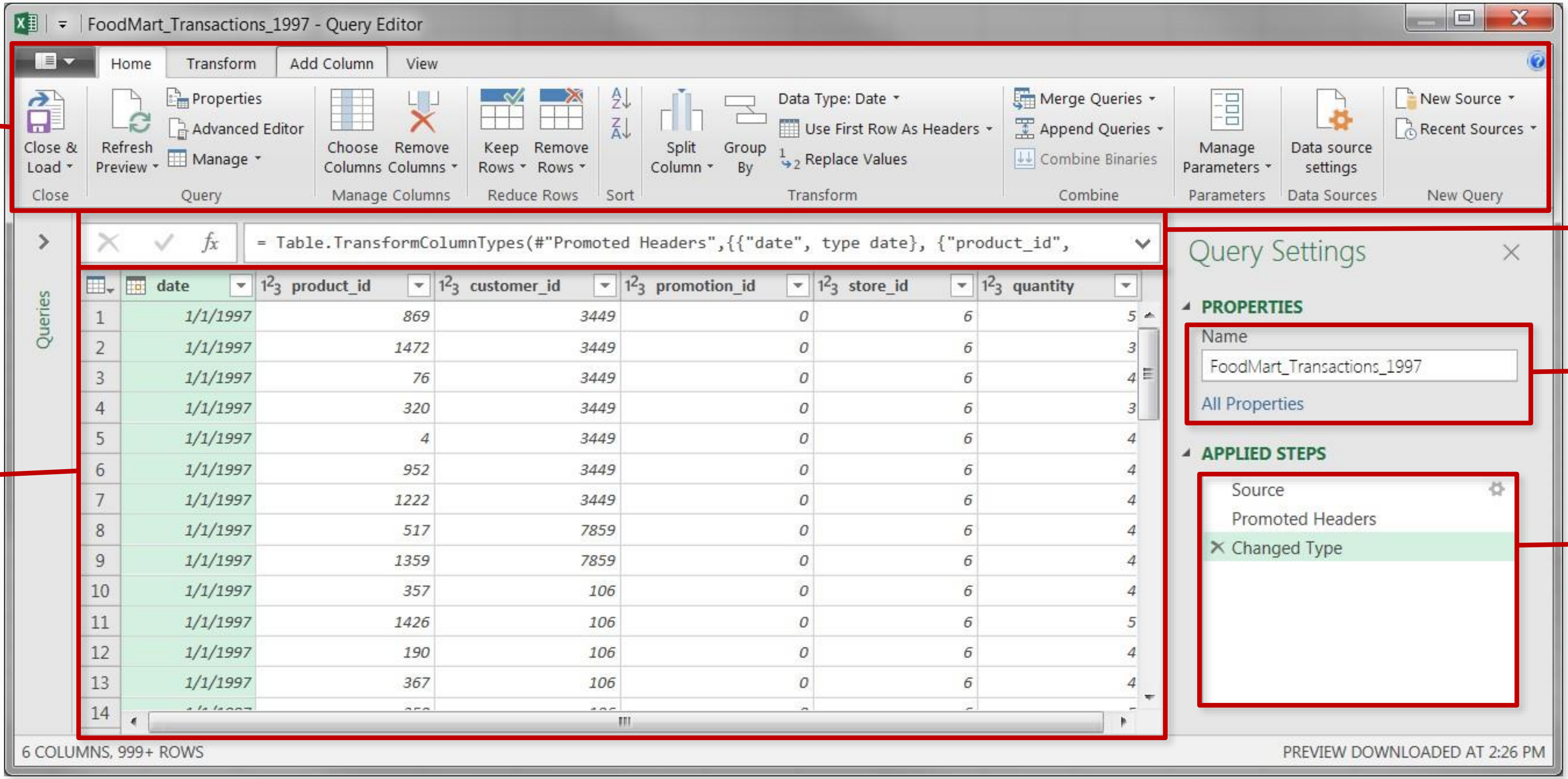
Formula Bar
(this is "M" code)

Name your table!

Gpearls

*Copyright 2020, Gpearls Data Analytics,
https://gpearls.com.ng

THE QUERY EDITOR



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"date", type date}, {"product_id",
```

	date	product_id	customer_id	promotion_id	store_id	quantity
1	1/1/1997	869	3449	0	6	5
2	1/1/1997	1472	3449	0	6	3
3	1/1/1997	76	3449	0	6	4
4	1/1/1997	320	3449	0	6	3
5	1/1/1997	4	3449	0	6	4
6	1/1/1997	952	3449	0	6	4
7	1/1/1997	1222	3449	0	6	4
8	1/1/1997	517	7859	0	6	4
9	1/1/1997	1359	7859	0	6	4
10	1/1/1997	357	106	0	6	4
11	1/1/1997	1426	106	0	6	5
12	1/1/1997	190	106	0	6	4
13	1/1/1997	367	106	0	6	4
14						

Query Settings

PROPERTIES

Name: FoodMart_Transactions_1997

All Properties

APPLIED STEPS

- Source
- Promoted Headers
- Changed Type

6 COLUMNS, 999+ ROWS

PREVIEW DOWNLOADED AT 2:26 PM

Query Editing Tools

Data Preview

Formula Bar *(this is "M" code)*

Name your table!

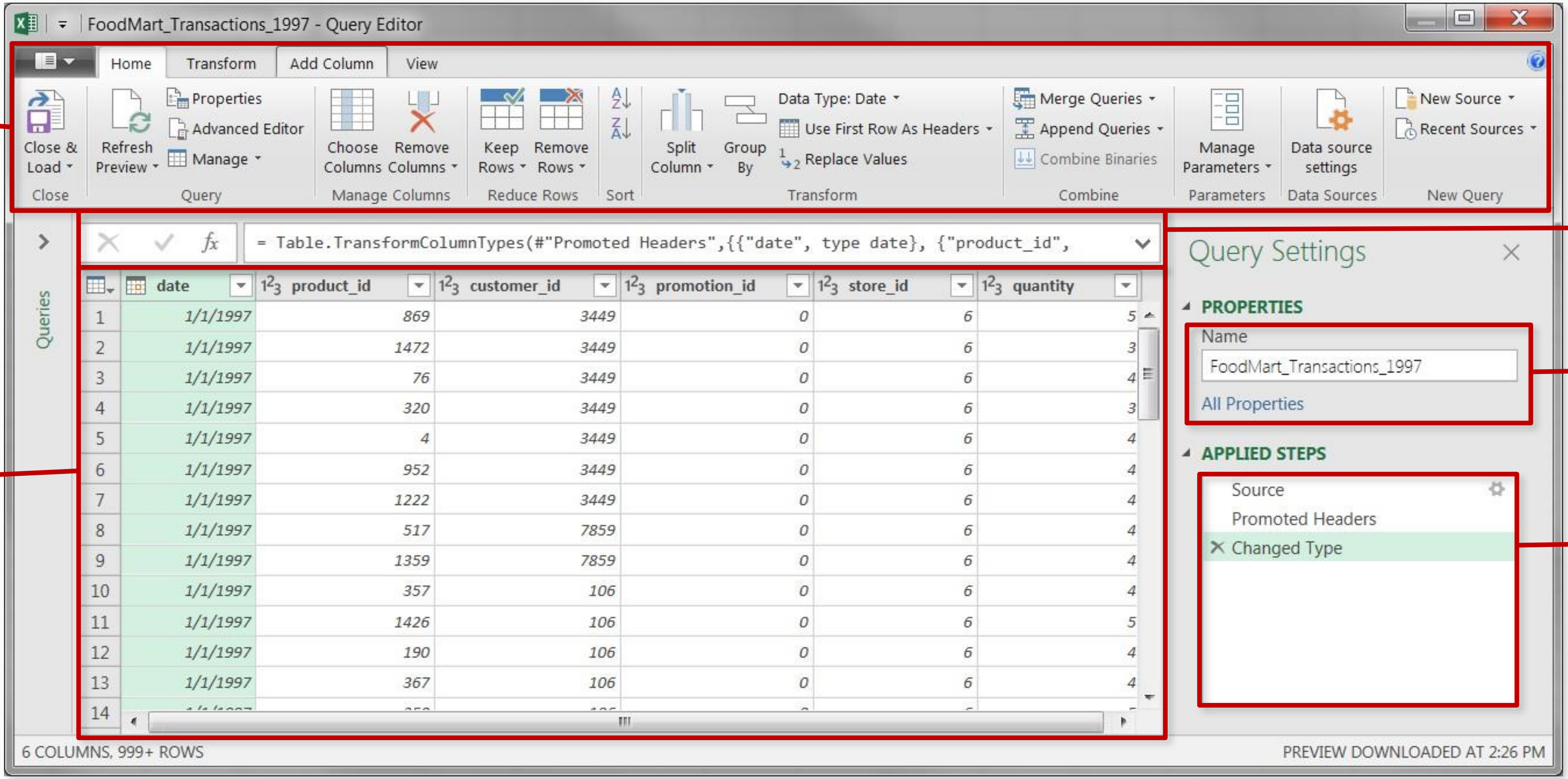
Applied Steps

Gpearls

THE QUERY EDITOR

Query Editing Tools

Data Preview



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"date", type date}, {"product_id",
```

	date	product_id	customer_id	promotion_id	store_id	quantity
1	1/1/1997	869	3449	0	6	5
2	1/1/1997	1472	3449	0	6	3
3	1/1/1997	76	3449	0	6	4
4	1/1/1997	320	3449	0	6	3
5	1/1/1997	4	3449	0	6	4
6	1/1/1997	952	3449	0	6	4
7	1/1/1997	1222	3449	0	6	4
8	1/1/1997	517	7859	0	6	4
9	1/1/1997	1359	7859	0	6	4
10	1/1/1997	357	106	0	6	4
11	1/1/1997	1426	106	0	6	5
12	1/1/1997	190	106	0	6	4
13	1/1/1997	367	106	0	6	4
14						

Name: FoodMart_Transactions_1997

- Source
- Promoted Headers
- Changed Type

6 COLUMNS, 999+ ROWS

PREVIEW DOWNLOADED AT 2:26 PM

Formula Bar
(this is "M" code)

Name your table!

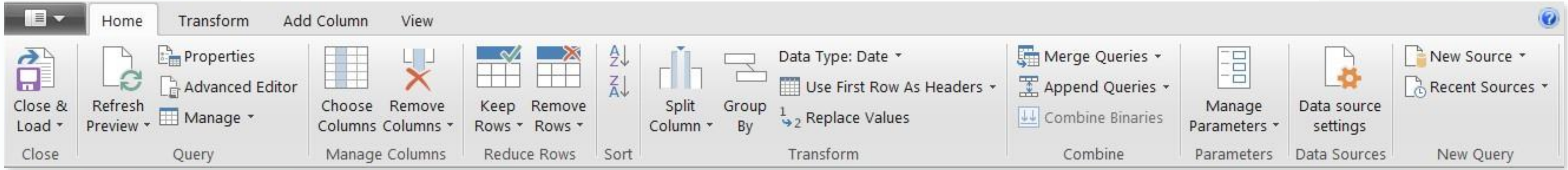
Applied Steps

Access the **Query Editor** by creating a new query and choosing the "*Edit*" option, or by launching the Workbook Queries pane (***Data*** > ***Show Queries***) and right-clicking an existing query to edit

QUERY EDITOR TOOLS

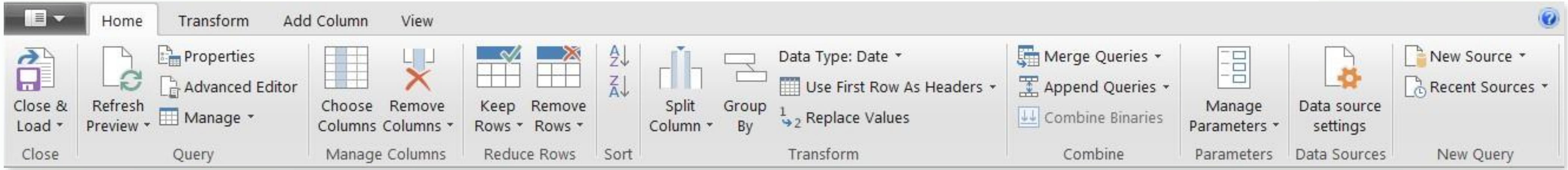
QUERY EDITOR TOOLS

The ***HOME*** *tab includes* ***general settings*** *and* ***common table transformation tools***

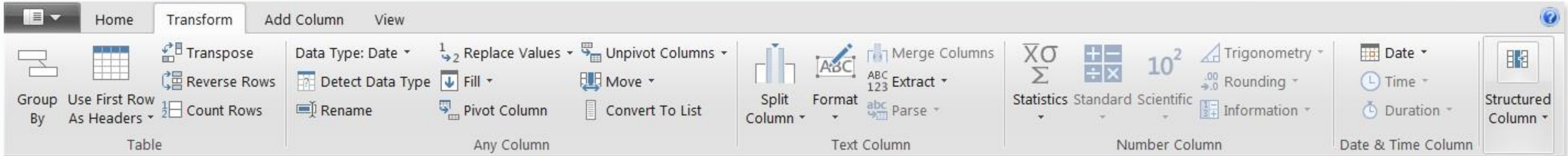


QUERY EDITOR TOOLS

*The **HOME** tab includes **general settings** and **common table transformation tools***

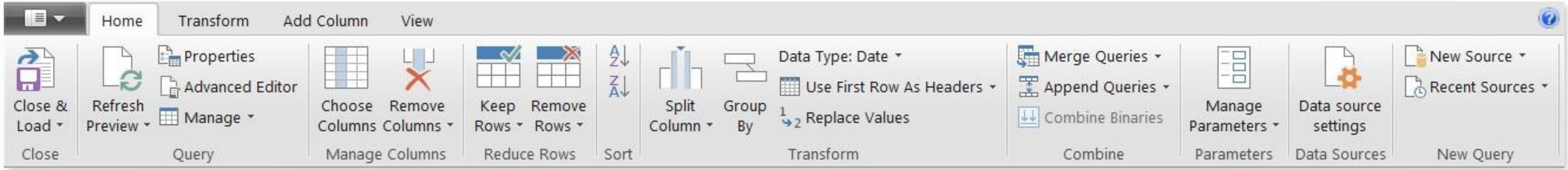


*The **TRANSFORM** tab includes tools to **modify existing columns** (splitting/grouping, transposing, extracting text, etc.*

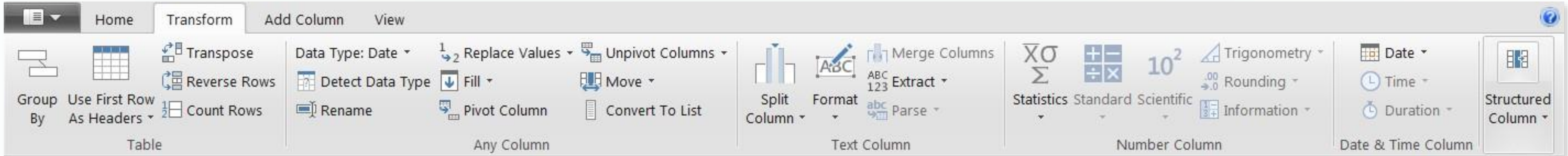


QUERY EDITOR TOOLS

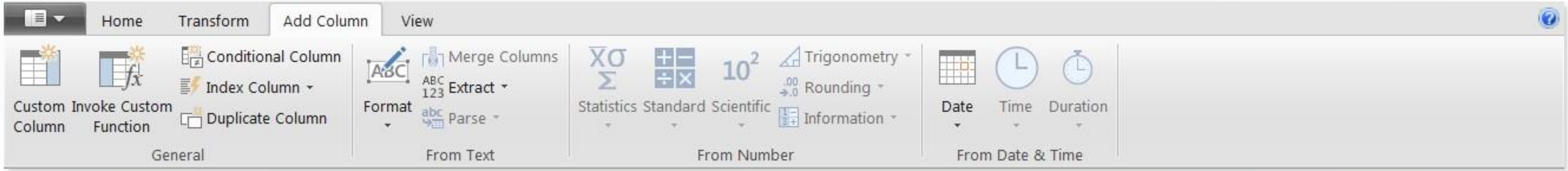
*The **HOME** tab includes **general settings** and **common table transformation tools***



*The **TRANSFORM** tab includes tools to **modify existing columns** (splitting/grouping, transposing, extracting text, etc.*

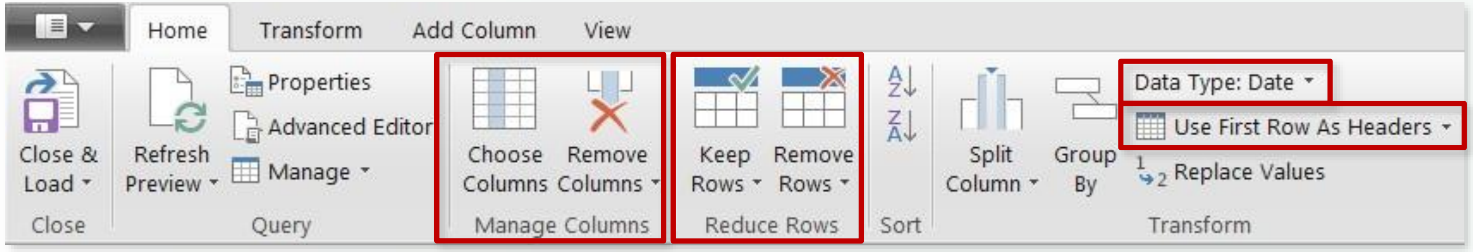


*The **ADD COLUMN** tools **create new columns** based on conditional rules, text operations, calculations, dates, etc.*

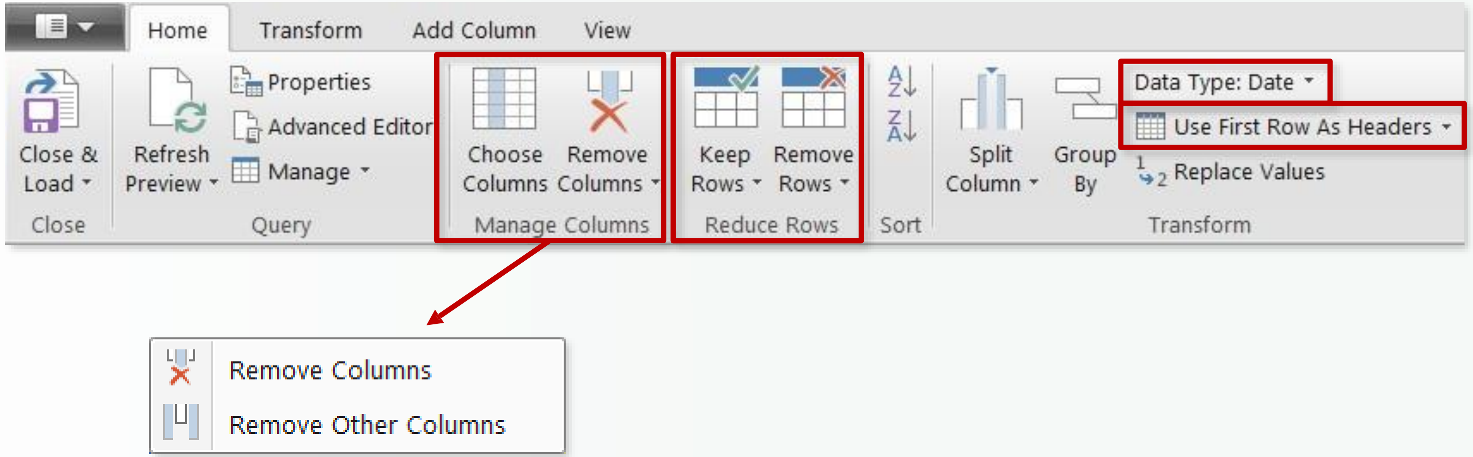


Gpearls

BASIC TABLE TRANSFORMATIONS



BASIC TABLE TRANSFORMATIONS

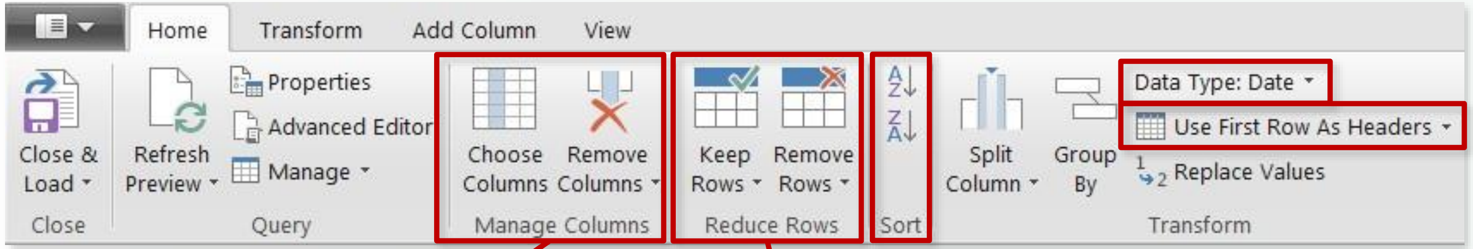


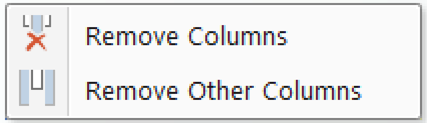
Keep or remove columns

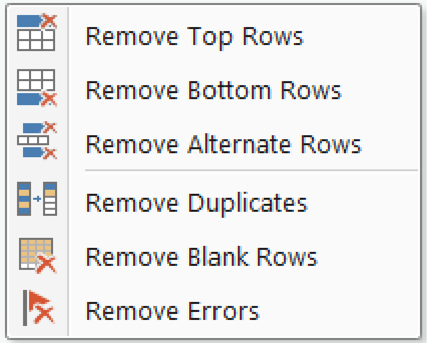
Tip: *use the "Remove Other Columns" option if you always want a specific set*

Gpearls

BASIC TABLE TRANSFORMATIONS







Keep or remove columns

Tip: *use the "Remove Other Columns" option if you always want a specific set*

Keep or remove rows

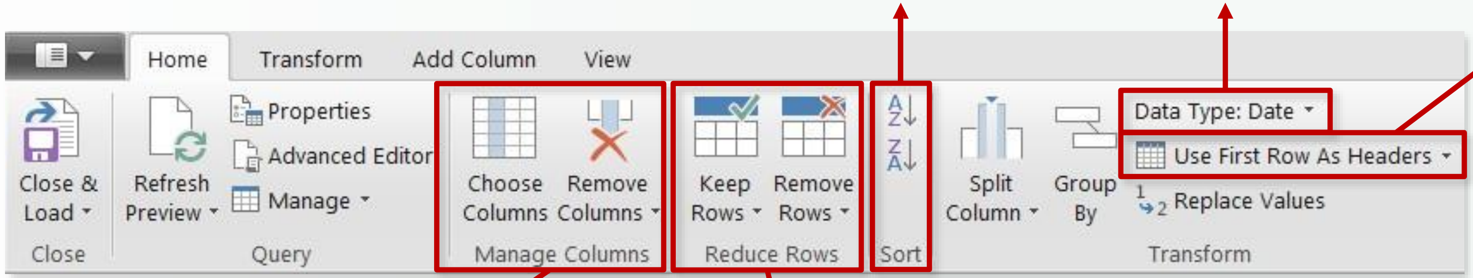
Tip: *use the "Remove Duplicates" option to create a new lookup table from scratch*

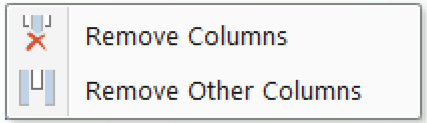
BASIC TABLE TRANSFORMATIONS

Sort values
(A-Z, Low-High, etc.)

Change data types
(date, $, %, text, etc.)

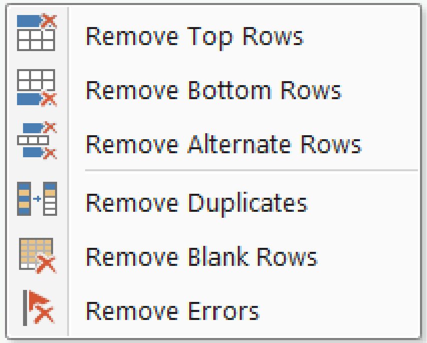
Promote header row





Keep or remove columns

Tip: *use the "Remove Other Columns" option if you always want a specific set*



Keep or remove rows

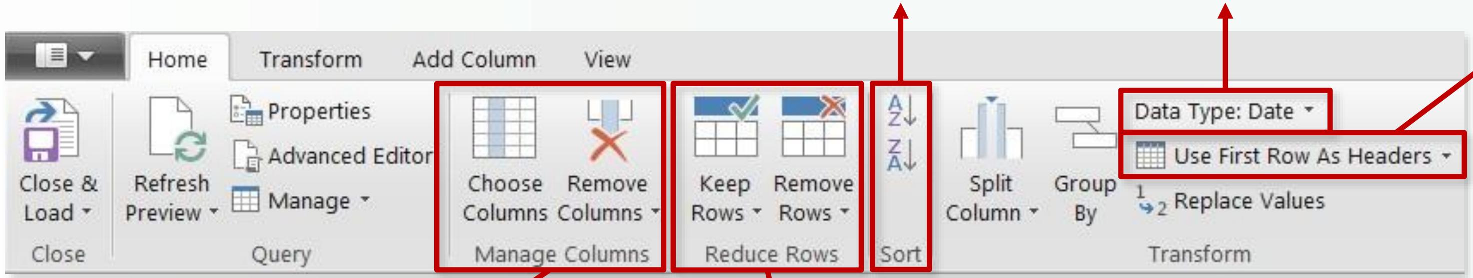
Tip: *use the "Remove Duplicates" option to create a new lookup table from scratch*

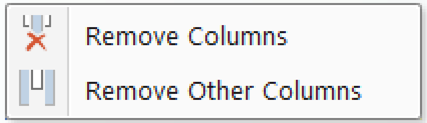
BASIC TABLE TRANSFORMATIONS

Sort values
(A-Z, Low-High, etc.)

Change data types
(date, $, %, text, etc.)

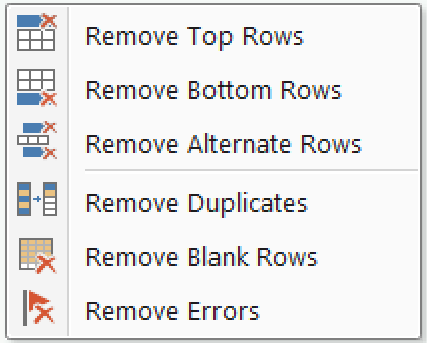
Promote header row





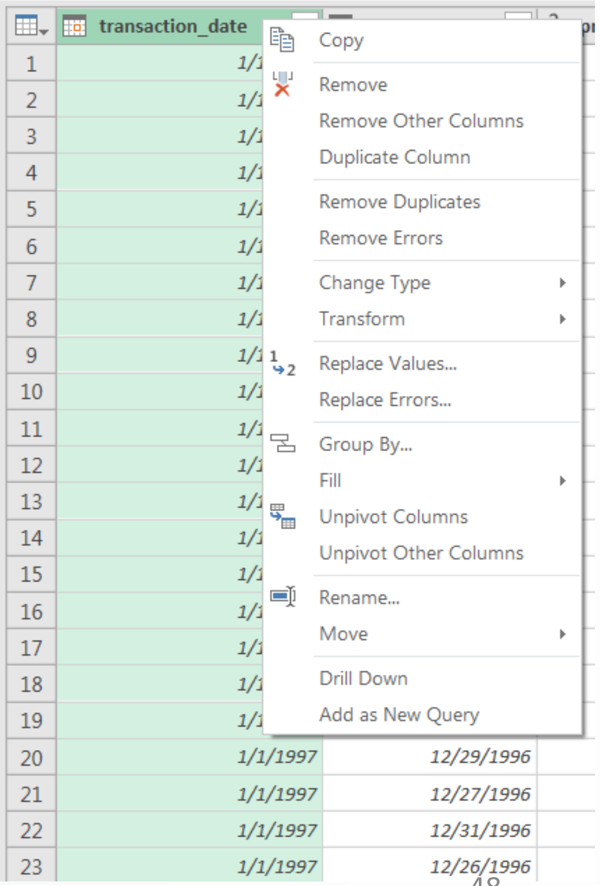
Keep or remove columns

***Tip:** use the "Remove Other Columns" option if you always want a specific set*



Keep or remove rows

***Tip:** use the "Remove Duplicates" option to create a new lookup table from scratch*

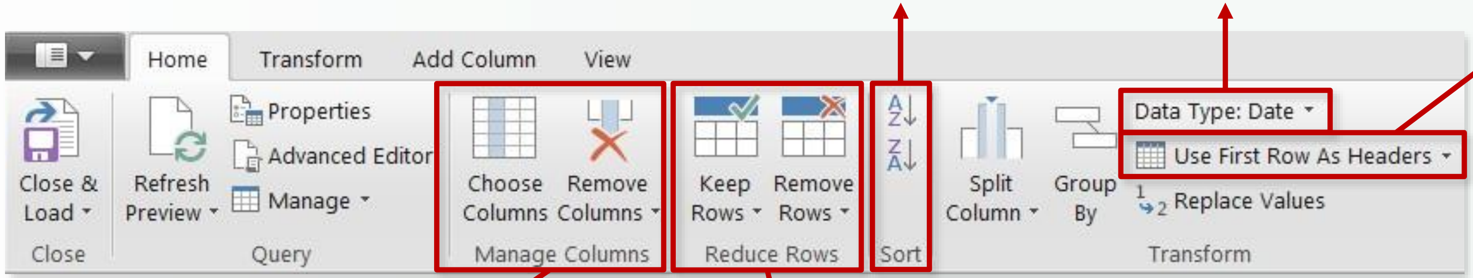


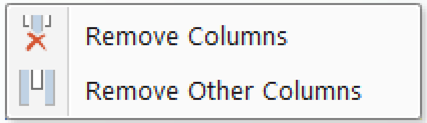
BASIC TABLE TRANSFORMATIONS

Sort values
(A-Z, Low-High, etc.)

Change data types
(date, $, %, text, etc.)

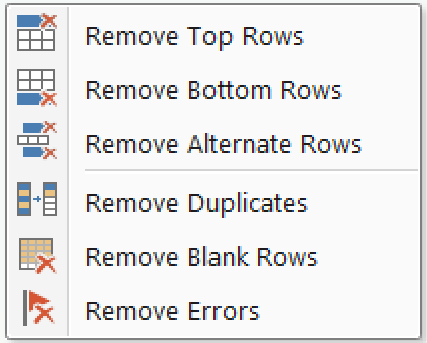
Promote header row





Keep or remove columns

Tip: *use the "Remove Other Columns" option if you always want a specific set*

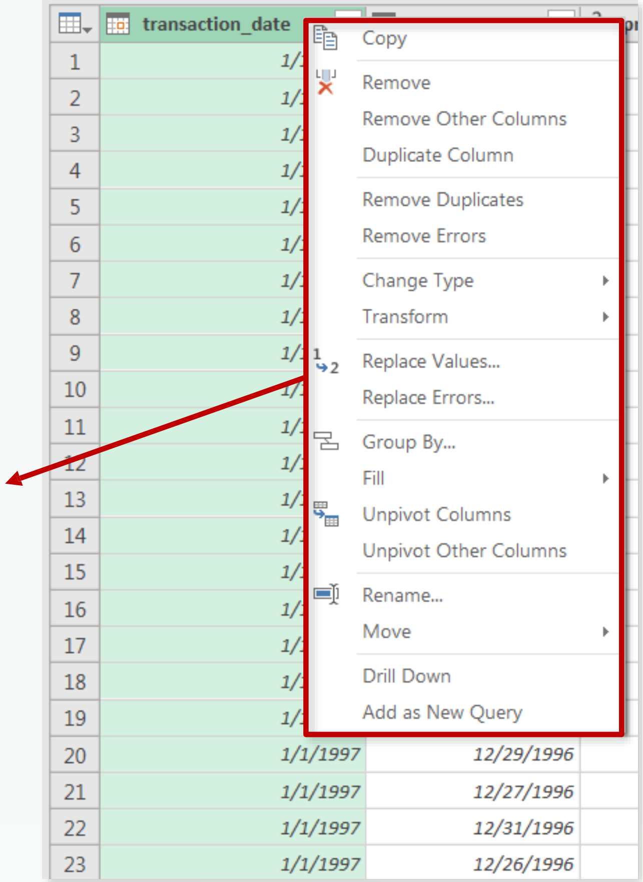


Keep or remove rows

Tip: *use the "Remove Duplicates" option to create a new lookup table from scratch*

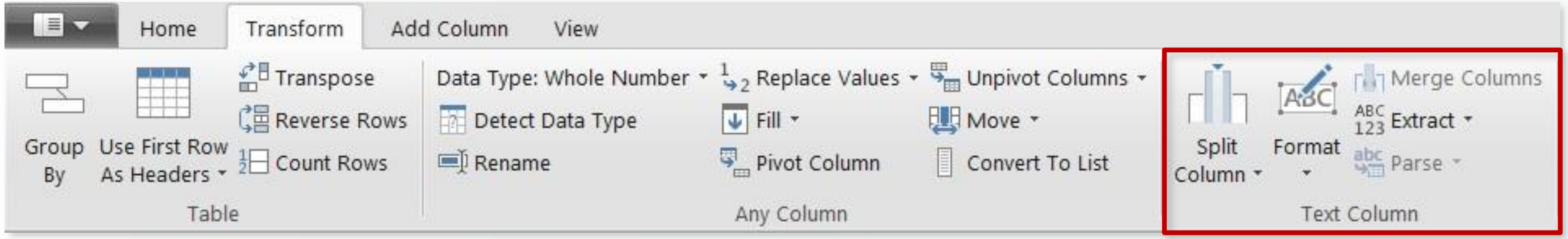
Duplicate, move & rename columns

Tip: *Right-click the column header to access common tools*

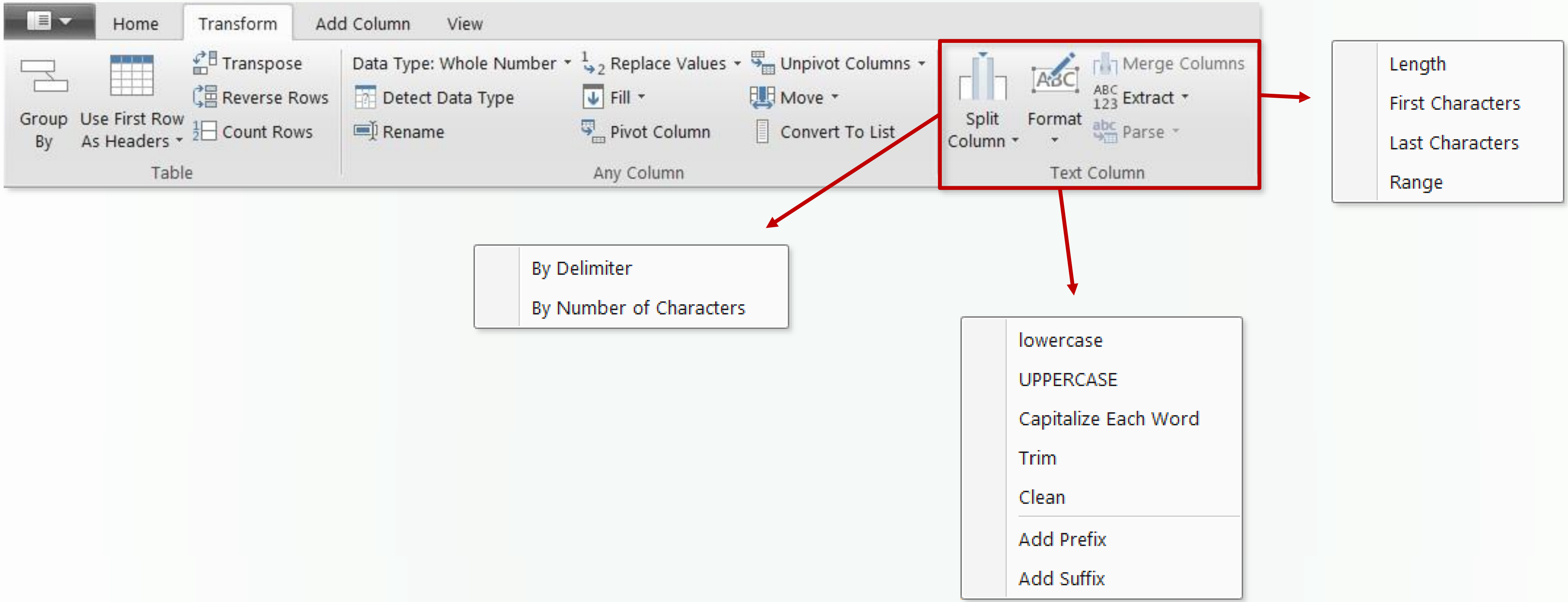


*Copyright 2020, Gpearls Data Analytics,
https://gpearls.com.ng

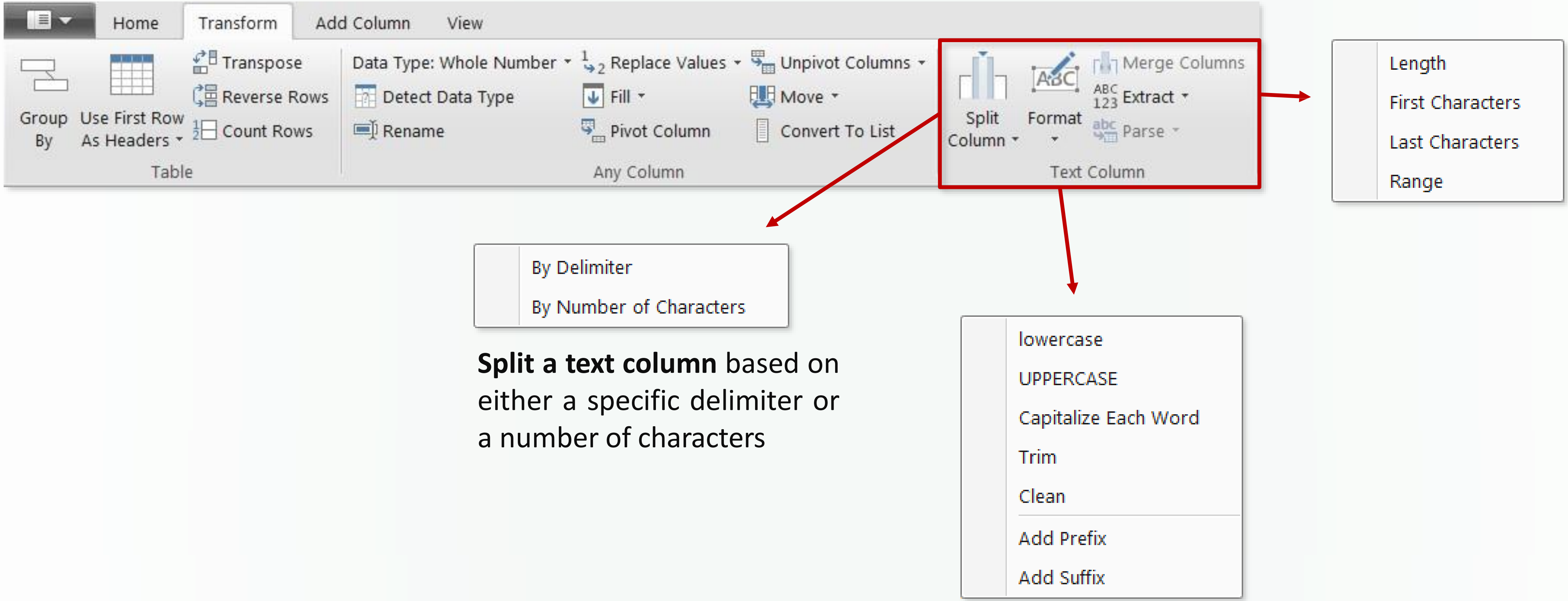
TEXT-SPECIFIC TOOLS



TEXT-SPECIFIC TOOLS

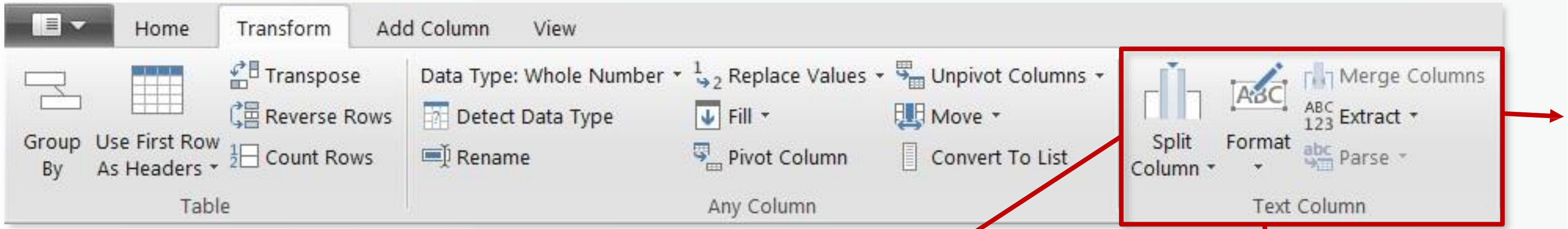


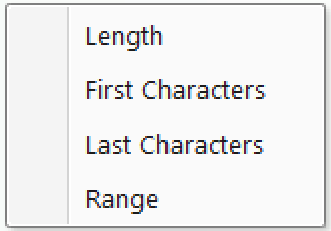
TEXT-SPECIFIC TOOLS

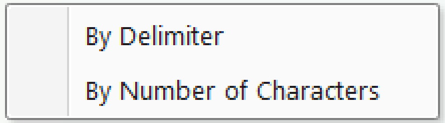


Split a text column based on either a specific delimiter or a number of characters

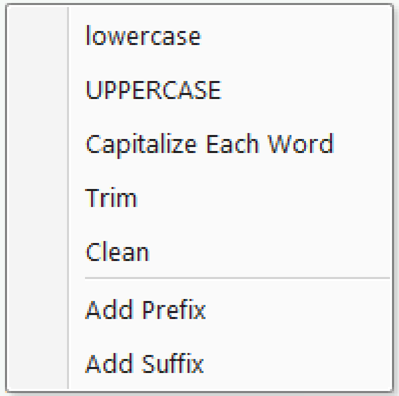
TEXT-SPECIFIC TOOLS







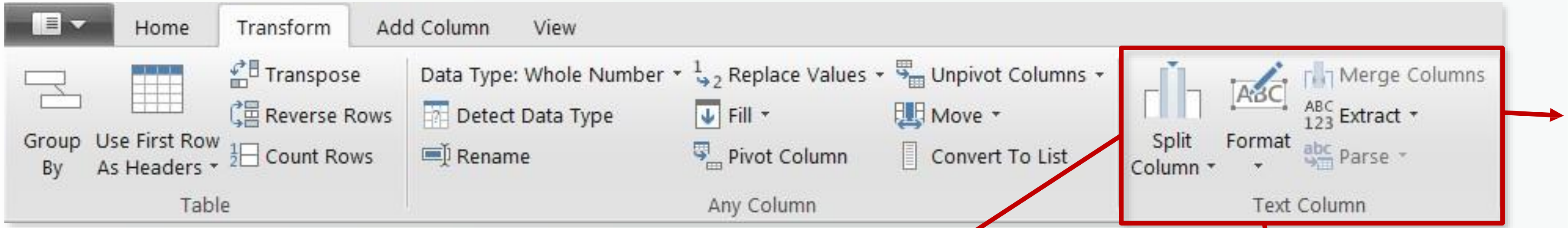
Split a text column based on either a specific delimiter or a number of characters

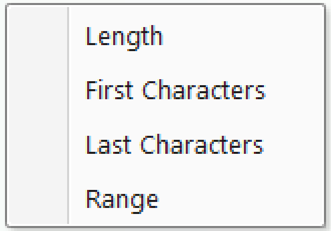


Format a text column to upper, lower or proper case, or add a prefix or suffix

***Tip:** Use "Trim" to eliminate leading & trailing spaces, or "Clean" to remove non-printable characters*

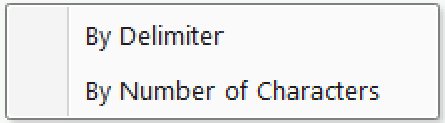
TEXT-SPECIFIC TOOLS



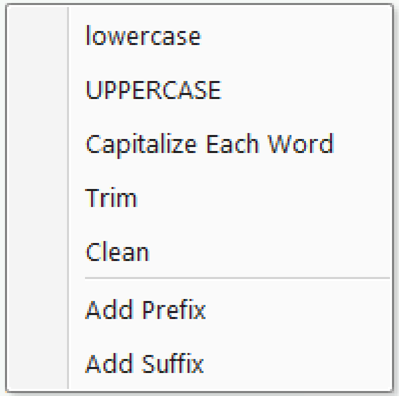


Extract characters from a text column using a fixed length, first or last, or a defined range

Tip*: Select two or more columns to* ***merge*** *or* ***concatenate*** *fields*



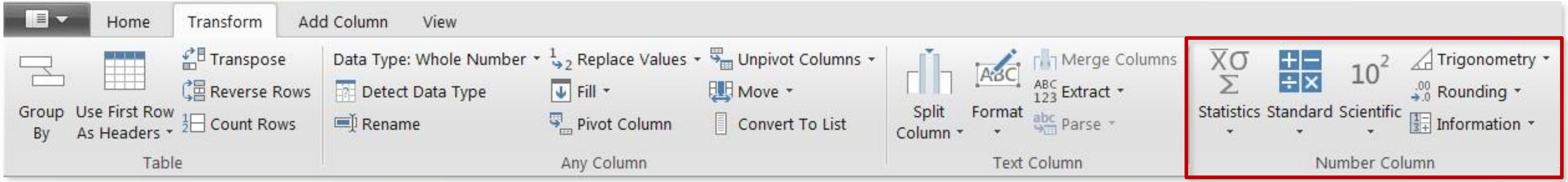
Split a text column based on either a specific delimiter or a number of characters



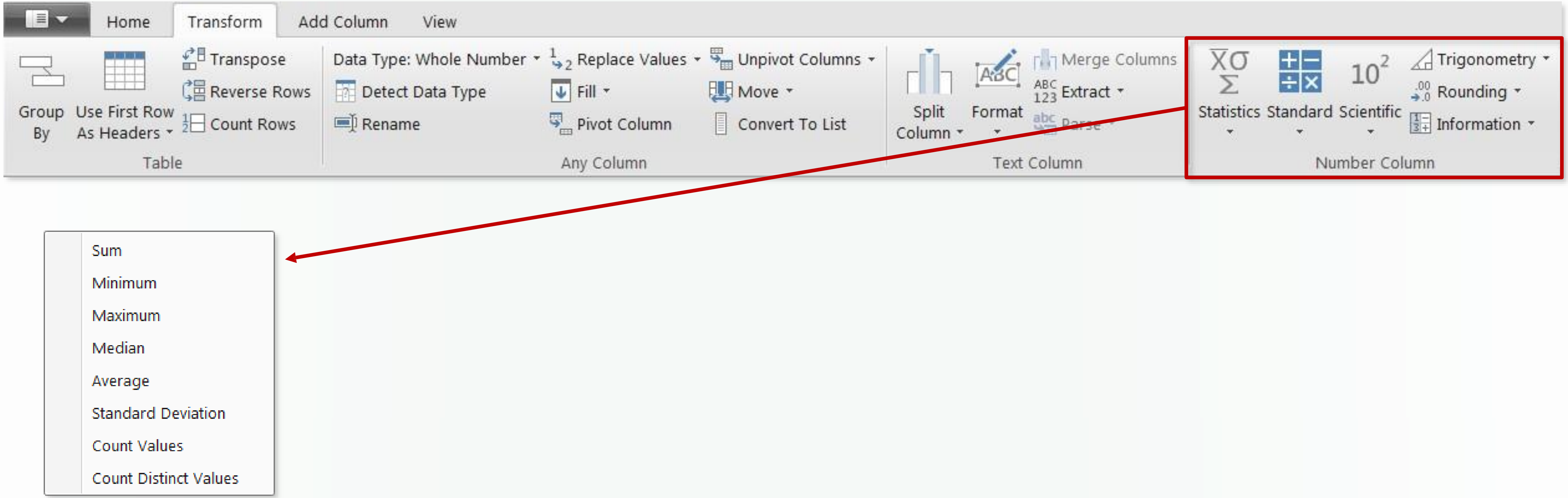
Format a text column to upper, lower or proper case, or add a prefix or suffix

Tip: *Use "Trim" to eliminate leading & trailing spaces, or "Clean" to remove non-printable characters*

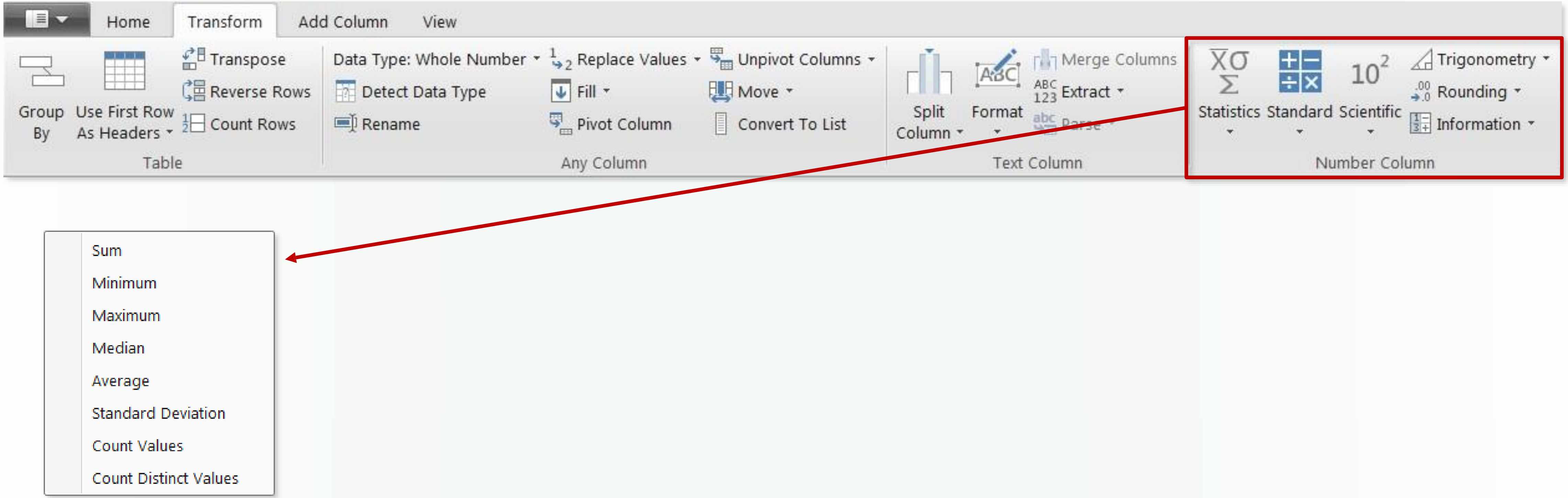
NUMBER-SPECIFIC TOOLS



NUMBER-SPECIFIC TOOLS



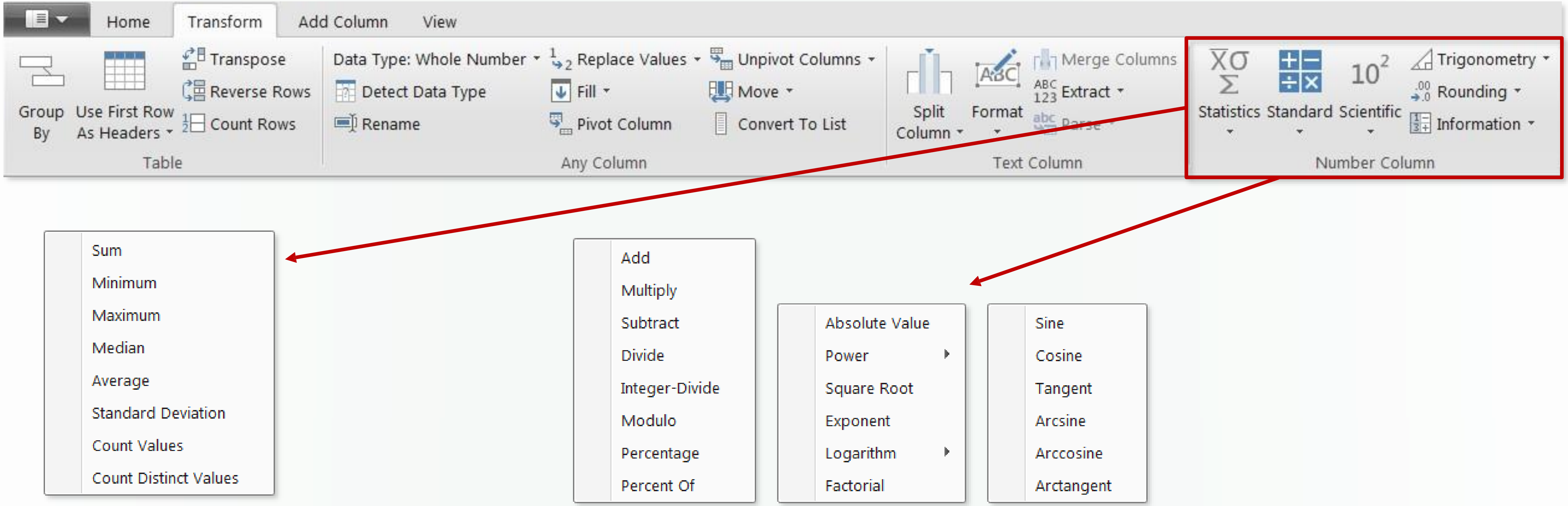
NUMBER-SPECIFIC TOOLS



Statistics functions allow you to evaluate basic stats for the selected column (sum, min/max, average, count, countdistinct, etc)

Note: *These tools return a SINGLE value, and are commonly used to explore a table rather than prepare it for loading*

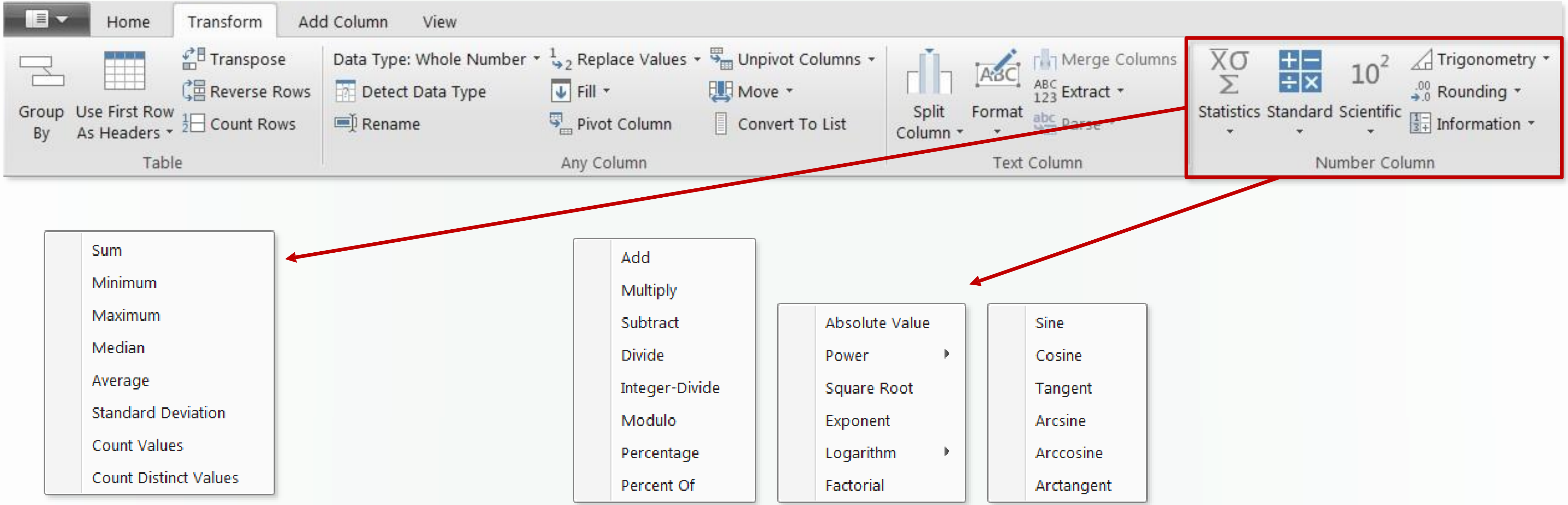
NUMBER-SPECIFIC TOOLS



Statistics functions allow you to evaluate basic stats for the selected column (sum, min/max, average, count, countdistinct, etc)

Note: *These tools return a SINGLE value, and are commonly used to explore a table rather than prepare it for loading*

NUMBER-SPECIFIC TOOLS



Standard *Scientific* *Trigonometry*

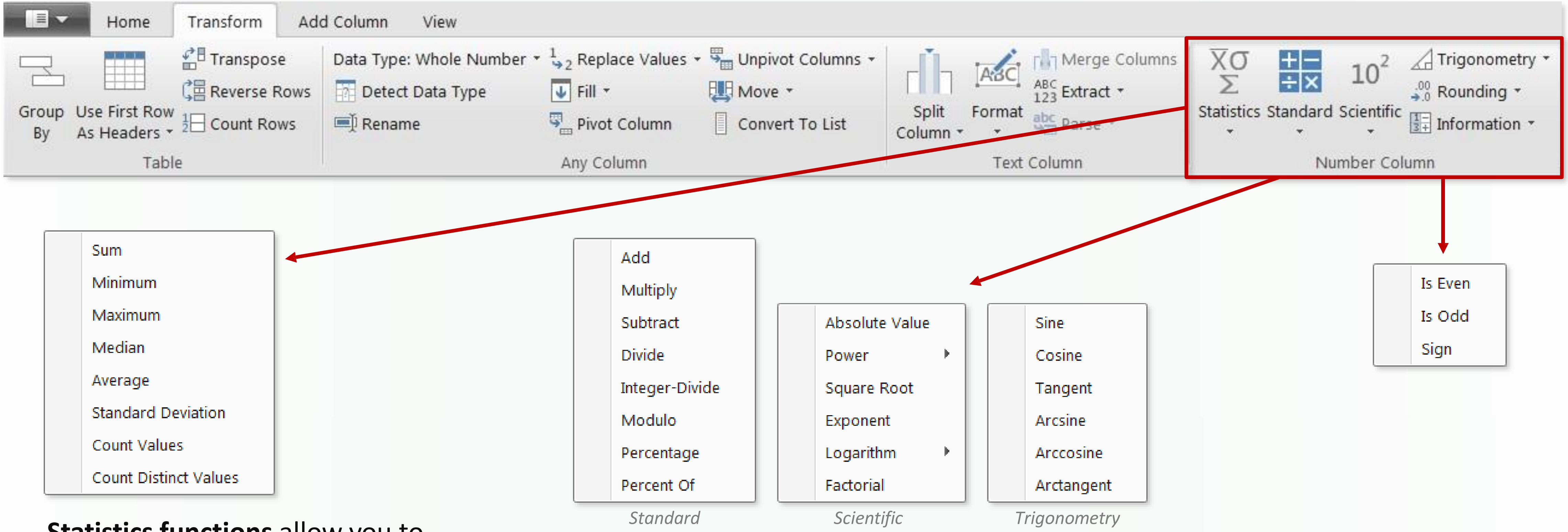
Statistics functions allow you to evaluate basic stats for the selected column (sum, min/max, average, count, countdistinct, etc)

Note: *These tools return a SINGLE value, and are commonly used to explore a table rather than prepare it for loading*

Standard, Scientific and **Trigonometry** tools allow you to apply standard operations (addition, multiplication, division, etc.) or more advanced calculations (power, logarithm, sine, tangent, etc) to each value in a column

Note: *Unlike the Statistics options, these tools are applied to each individual row in the table*

NUMBER-SPECIFIC TOOLS



Standard *Scientific* *Trigonometry*

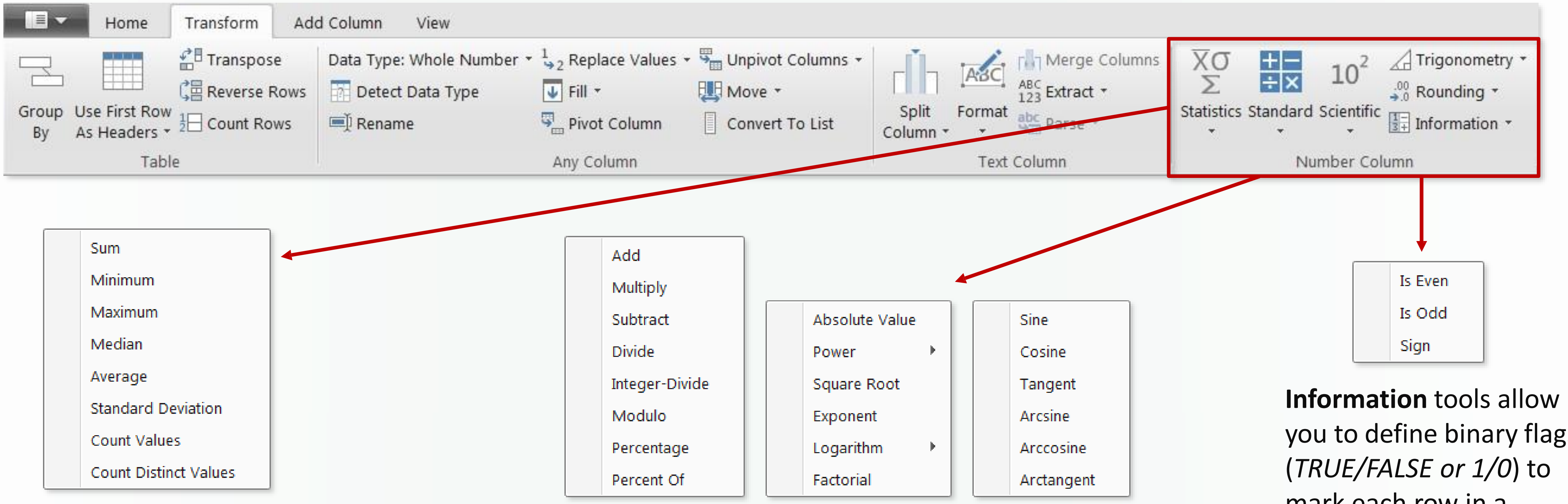
Statistics functions allow you to evaluate basic stats for the selected column (sum, min/max, average, count, countdistinct, etc)

Note: *These tools return a SINGLE value, and are commonly used to explore a table rather than prepare it for loading*

Standard, Scientific and **Trigonometry** tools allow you to apply standard operations (addition, multiplication, division, etc.) or more advanced calculations (power, logarithm, sine, tangent, etc) to each value in a column

Note: *Unlike the Statistics options, these tools are applied to each individual row in the table*

NUMBER-SPECIFIC TOOLS



Standard *Scientific* *Trigonometry*

Statistics functions allow you to evaluate basic stats for the selected column (sum, min/max, average, count, countdistinct, etc)

Note: *These tools return a SINGLE value, and are commonly used to explore a table rather than prepare it for loading*

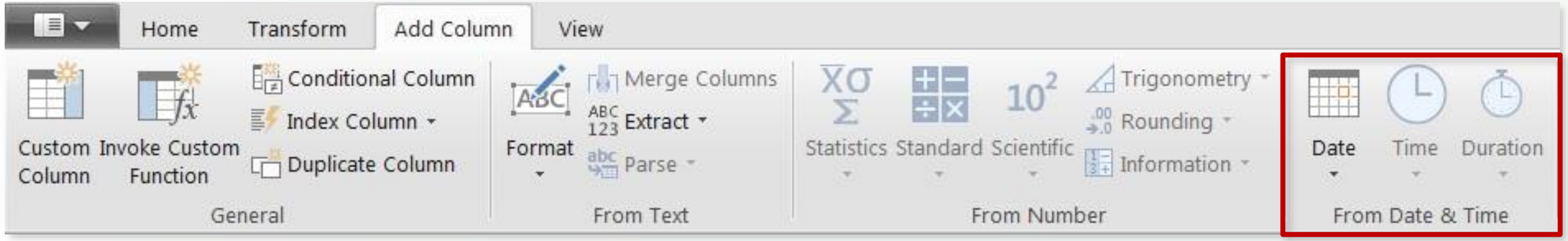
Standard, Scientific and **Trigonometry** tools allow you to apply standard operations (addition, multiplication, division, etc.) or more advanced calculations (power, logarithm, sine, tangent, etc) to each value in a column

Note: *Unlike the Statistics options, these tools are applied to each individual row in the table*

Information tools allow you to define binary flags (*TRUE/FALSE or 1/0*) to mark each row in a column as even, odd, positive or negative

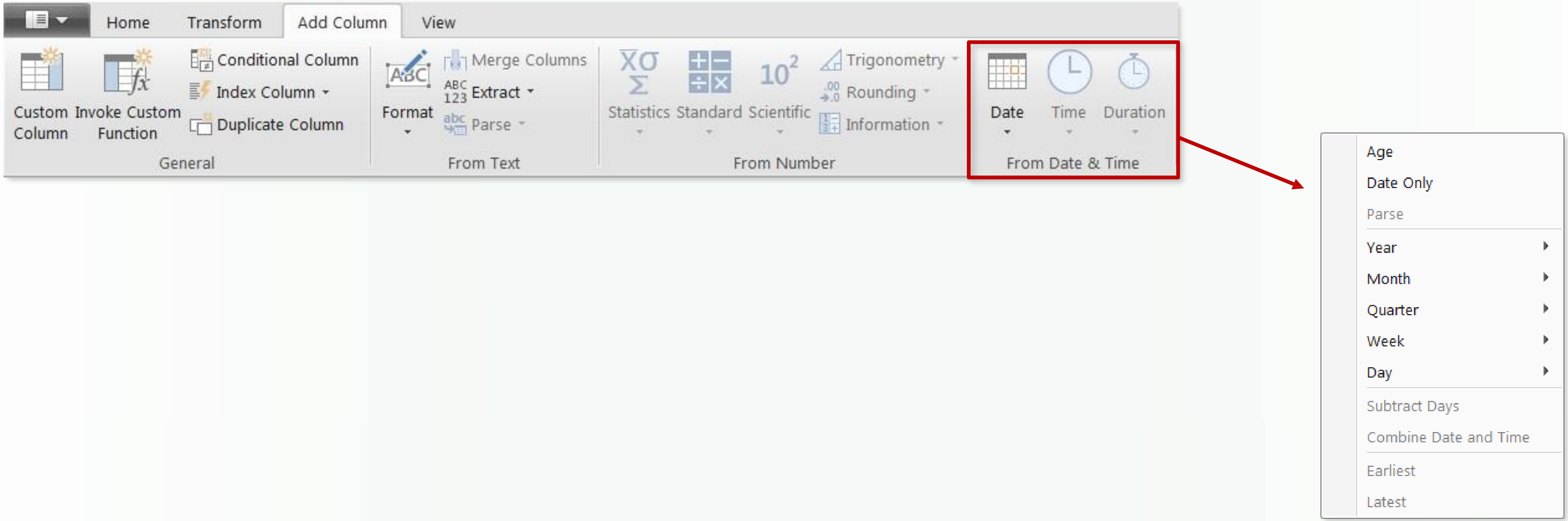
Gpearls

DATE-SPECIFIC TOOLS

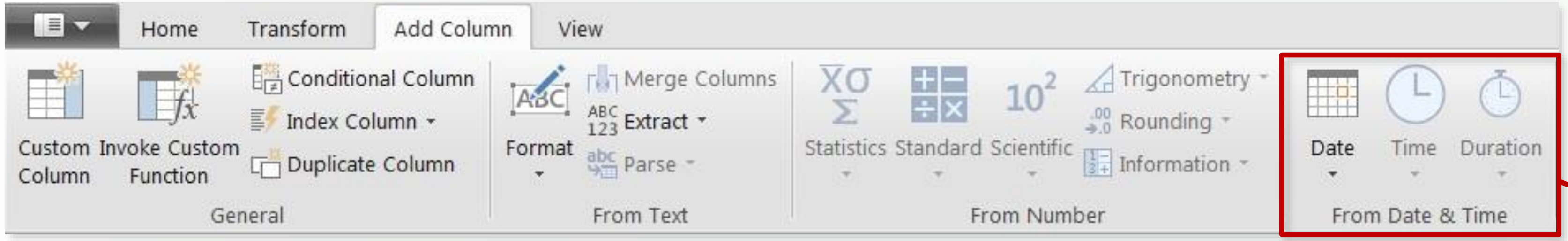


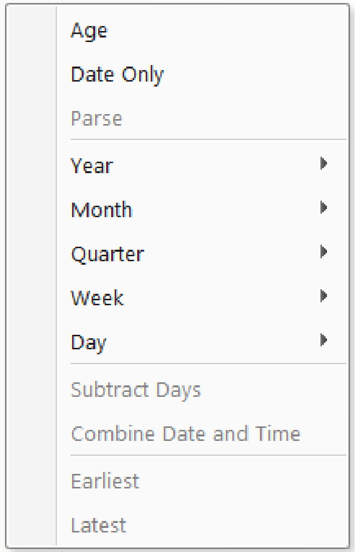
*Copyright 2020, Gpearls Data Analytics, https://gpearls.com.ng

DATE-SPECIFIC TOOLS



DATE-SPECIFIC TOOLS





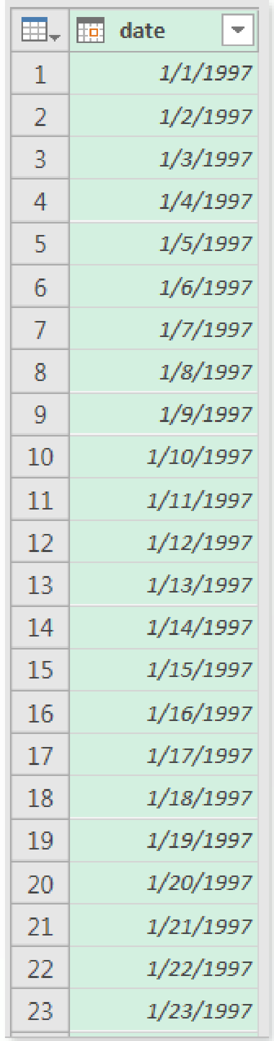
Date & Time tools are relatively straight-forward, and include the following options:

- **Age**: Difference between the current time and the date in each row
- **Date Only:** Removes the time component of a date/time field
- **Year/Month/Quarter/Week/Day**: Extracts individual components from a date field (Time-specific options include Hour, Minute, Second, etc.)
- **Earliest/Latest:** Evaluates the earliest or latest date from a column as a single value (can only be accessed from the "Transform" menu)

Note: *You will almost always want to perform these operations from the "Add Column" menu to build out new fields, rather than transforming an individual date/time column*

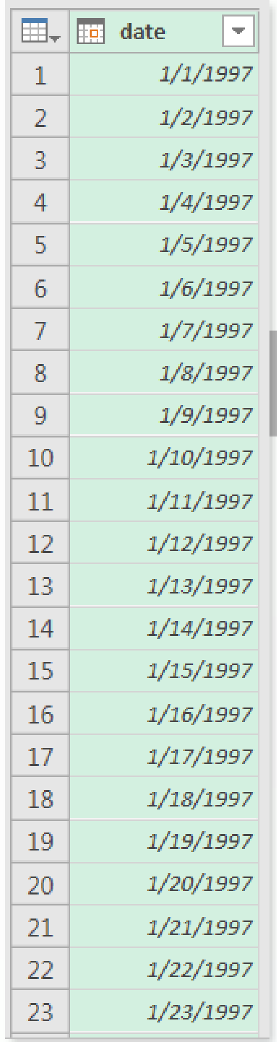
CREATING A BASIC CALENDAR TABLE

CREATING A BASIC CALENDAR TABLE

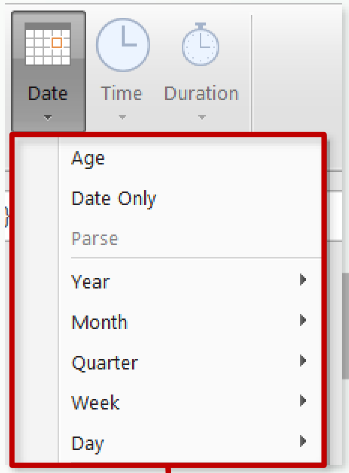


	date
1	1/1/1997
2	1/2/1997
3	1/3/1997
4	1/4/1997
5	1/5/1997
6	1/6/1997
7	1/7/1997
8	1/8/1997
9	1/9/1997
10	1/10/1997
11	1/11/1997
12	1/12/1997
13	1/13/1997
14	1/14/1997
15	1/15/1997
16	1/16/1997
17	1/17/1997
18	1/18/1997
19	1/19/1997
20	1/20/1997
21	1/21/1997
22	1/22/1997
23	1/23/1997

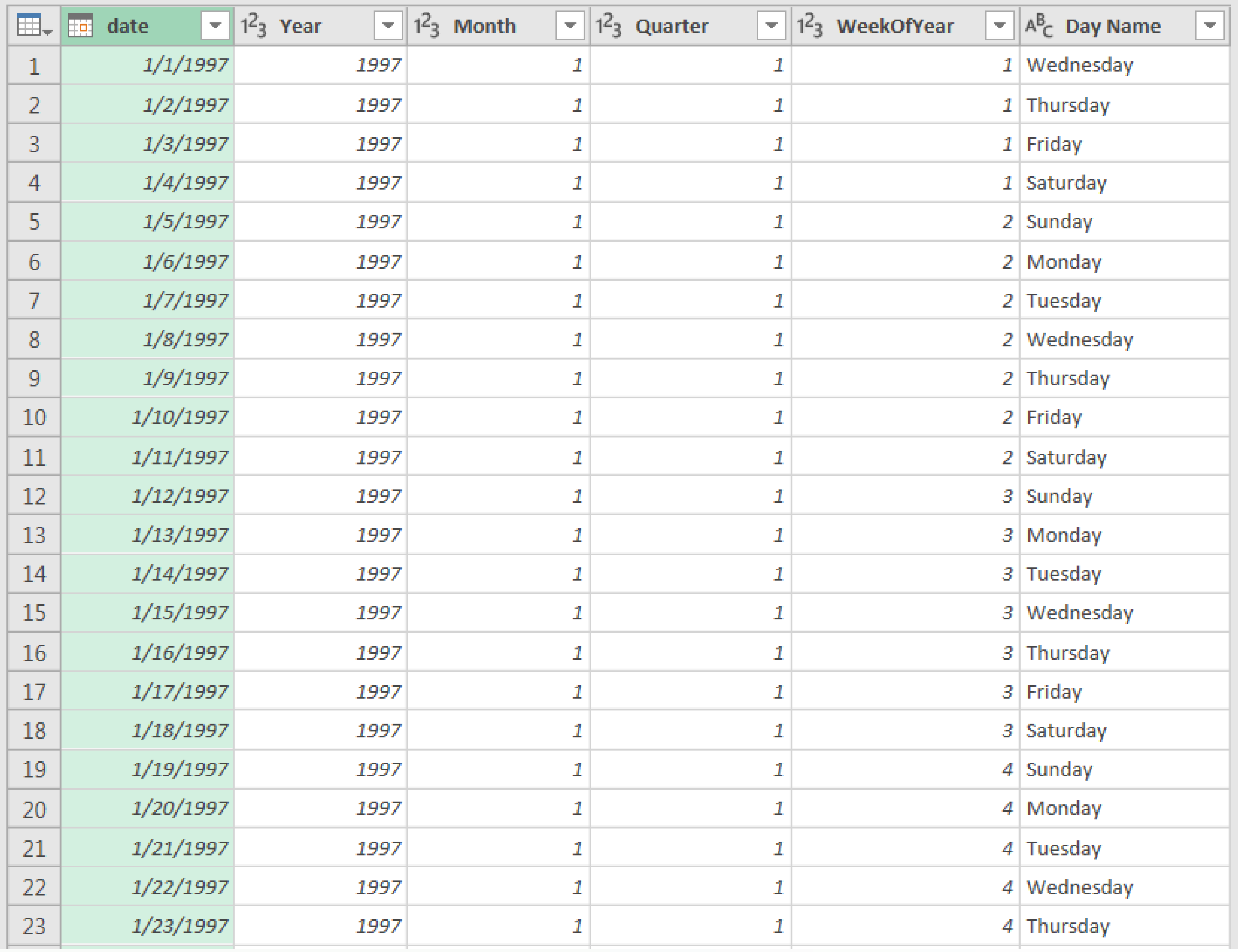
CREATING A BASIC CALENDAR TABLE



	date
1	1/1/1997
2	1/2/1997
3	1/3/1997
4	1/4/1997
5	1/5/1997
6	1/6/1997
7	1/7/1997
8	1/8/1997
9	1/9/1997
10	1/10/1997
11	1/11/1997
12	1/12/1997
13	1/13/1997
14	1/14/1997
15	1/15/1997
16	1/16/1997
17	1/17/1997
18	1/18/1997
19	1/19/1997
20	1/20/1997
21	1/21/1997
22	1/22/1997
23	1/23/1997

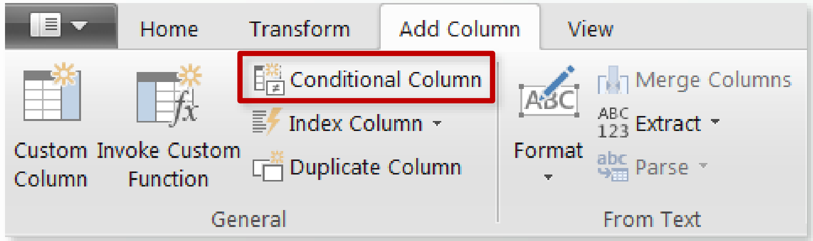


Use pre-defined **Date** options in the “**Add Column**” menu to quickly build out a calendar table from a list of dates



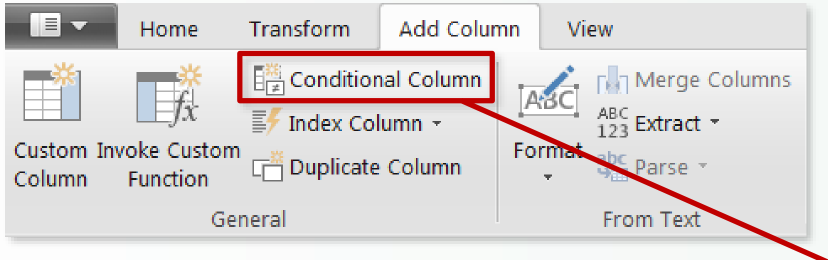
	date	Year	Month	Quarter	WeekOfYear	Day Name
1	1/1/1997	1997	1	1	1	Wednesday
2	1/2/1997	1997	1	1	1	Thursday
3	1/3/1997	1997	1	1	1	Friday
4	1/4/1997	1997	1	1	1	Saturday
5	1/5/1997	1997	1	1	2	Sunday
6	1/6/1997	1997	1	1	2	Monday
7	1/7/1997	1997	1	1	2	Tuesday
8	1/8/1997	1997	1	1	2	Wednesday
9	1/9/1997	1997	1	1	2	Thursday
10	1/10/1997	1997	1	1	2	Friday
11	1/11/1997	1997	1	1	2	Saturday
12	1/12/1997	1997	1	1	3	Sunday
13	1/13/1997	1997	1	1	3	Monday
14	1/14/1997	1997	1	1	3	Tuesday
15	1/15/1997	1997	1	1	3	Wednesday
16	1/16/1997	1997	1	1	3	Thursday
17	1/17/1997	1997	1	1	3	Friday
18	1/18/1997	1997	1	1	3	Saturday
19	1/19/1997	1997	1	1	4	Sunday
20	1/20/1997	1997	1	1	4	Monday
21	1/21/1997	1997	1	1	4	Tuesday
22	1/22/1997	1997	1	1	4	Wednesday
23	1/23/1997	1997	1	1	4	Thursday

ADDING A CONDITIONAL COLUMN

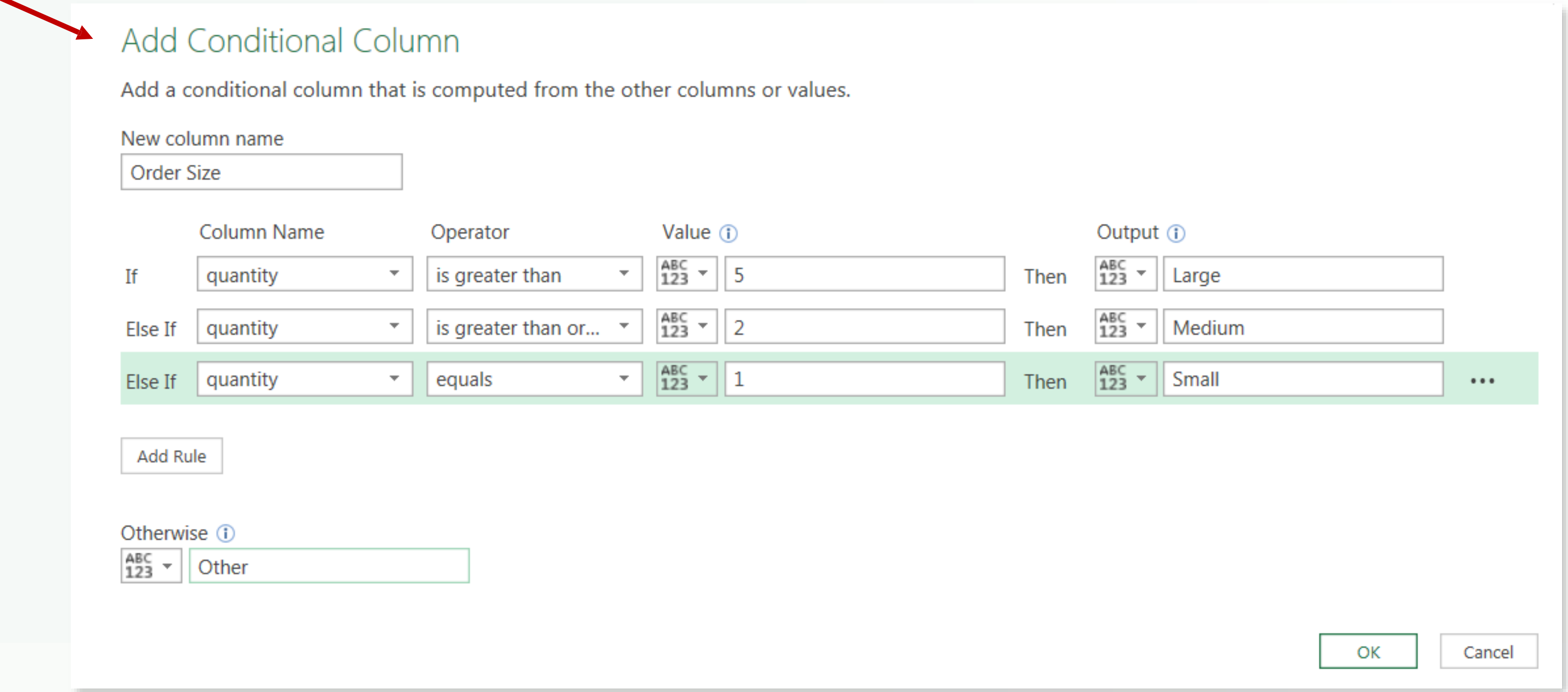


Gpearls

ADDING A CONDITIONAL COLUMN



Conditional Columns allow you to define new fields based on logical rules and conditions (*IF/THEN statements*)



Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name: Order Size

	Column Name	Operator	Value		Output
If	quantity	is greater than	5	Then	Large
Else If	quantity	is greater than or...	2	Then	Medium
Else If	quantity	equals	1	Then	Small

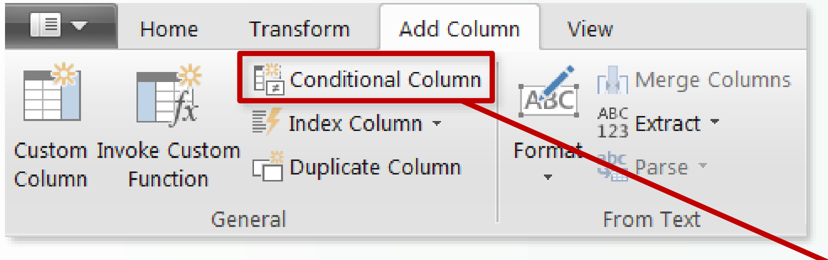
Add Rule

Otherwise: Other

OK Cancel

*Copyright 2020, Gpearls Data Analytics,
https://gpearls.com.ng

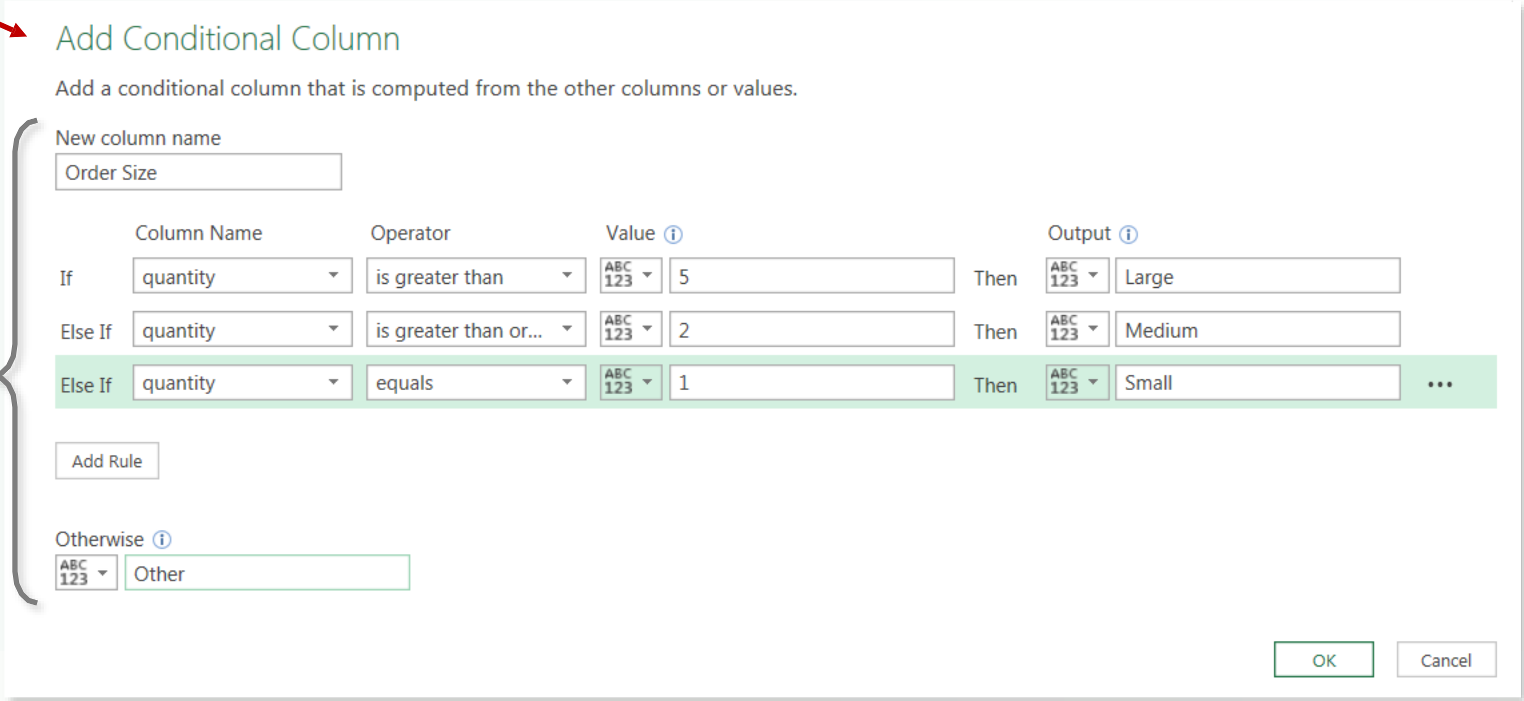
ADDING A CONDITIONAL COLUMN



Conditional Columns allow you to define new fields based on logical rules and conditions (*IF/THEN statements*)

*In this case we're creating a new conditional column called "**Order Size**", which depends on the values in the "**quantity**" column, as follows:*

- *If quantity **>5**, Order Size = "**Large**"*
- *If quantity is **from 2-5**, Order Size = "**Medium**"*
- *If quantity **=1**, Order Size = "**Small**"*
- *Otherwise Order Size = "**Other**"*



Add Conditional Column

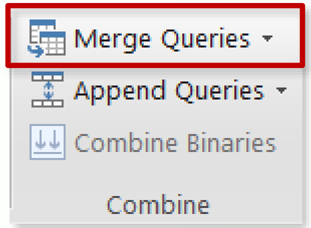
Add a conditional column that is computed from the other columns or values.

New column name: Order Size

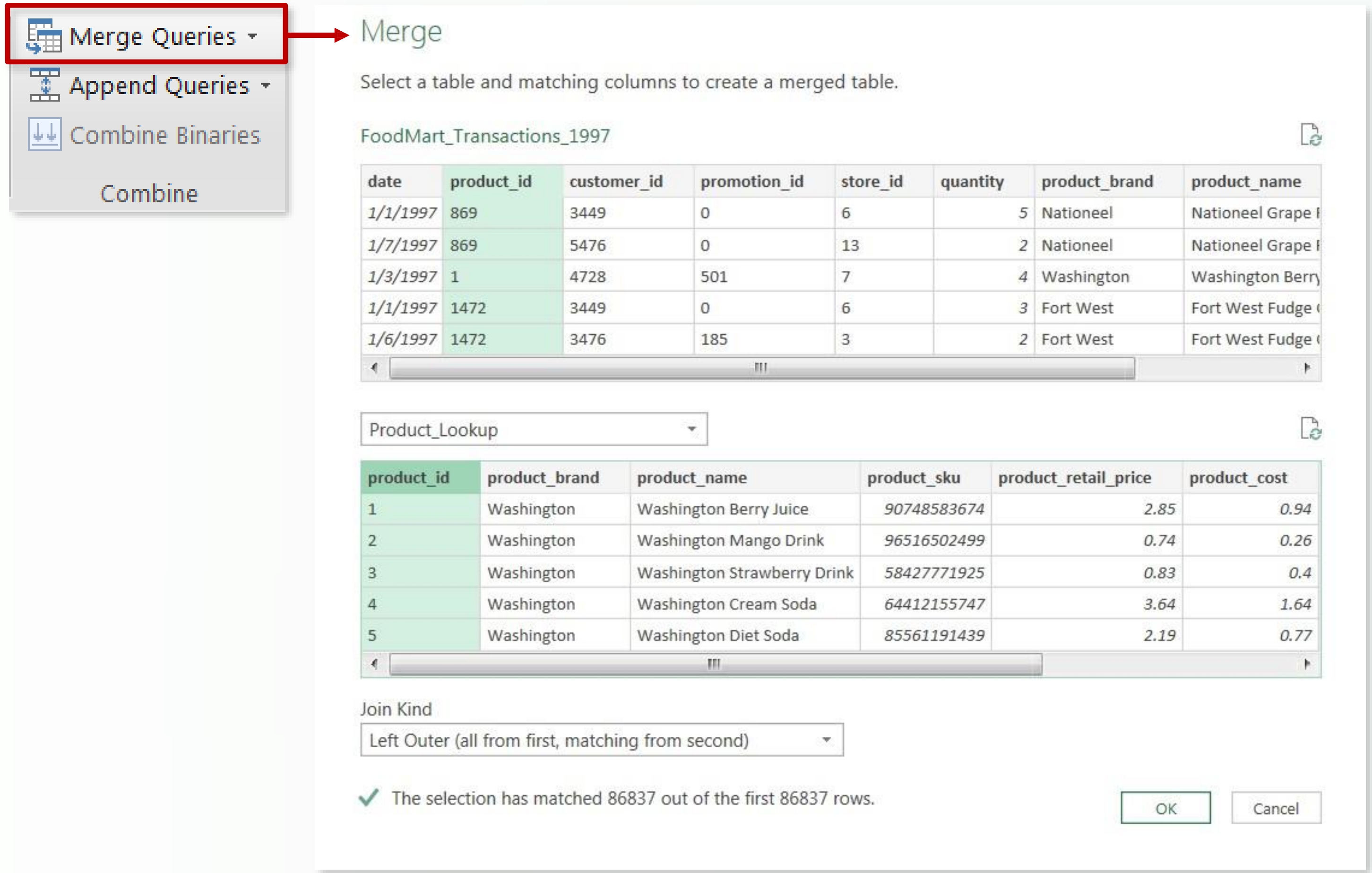
	Column Name	Operator	Value		Output
If	quantity	is greater than	5	Then	Large
Else If	quantity	is greater than or...	2	Then	Medium
Else If	quantity	equals	1	Then	Small

Otherwise: Other

MERGING QUERIES



MERGING QUERIES



Merge

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

FoodMart_Transactions_1997

date	product_id	customer_id	promotion_id	store_id	quantity	product_brand	product_name
1/1/1997	869	3449	0	6	5	Nationeel	Nationeel Grape
1/7/1997	869	5476	0	13	2	Nationeel	Nationeel Grape
1/3/1997	1	4728	501	7	4	Washington	Washington Berry
1/1/1997	1472	3449	0	6	3	Fort West	Fort West Fudge
1/6/1997	1472	3476	185	3	2	Fort West	Fort West Fudge

Product_Lookup

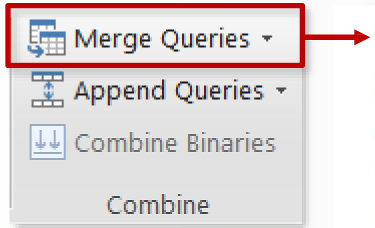
product_id	product_brand	product_name	product_sku	product_retail_price	product_cost
1	Washington	Washington Berry Juice	90748583674	2.85	0.94
2	Washington	Washington Mango Drink	96516502499	0.74	0.26
3	Washington	Washington Strawberry Drink	58427771925	0.83	0.4
4	Washington	Washington Cream Soda	64412155747	3.64	1.64
5	Washington	Washington Diet Soda	85561191439	2.19	0.77

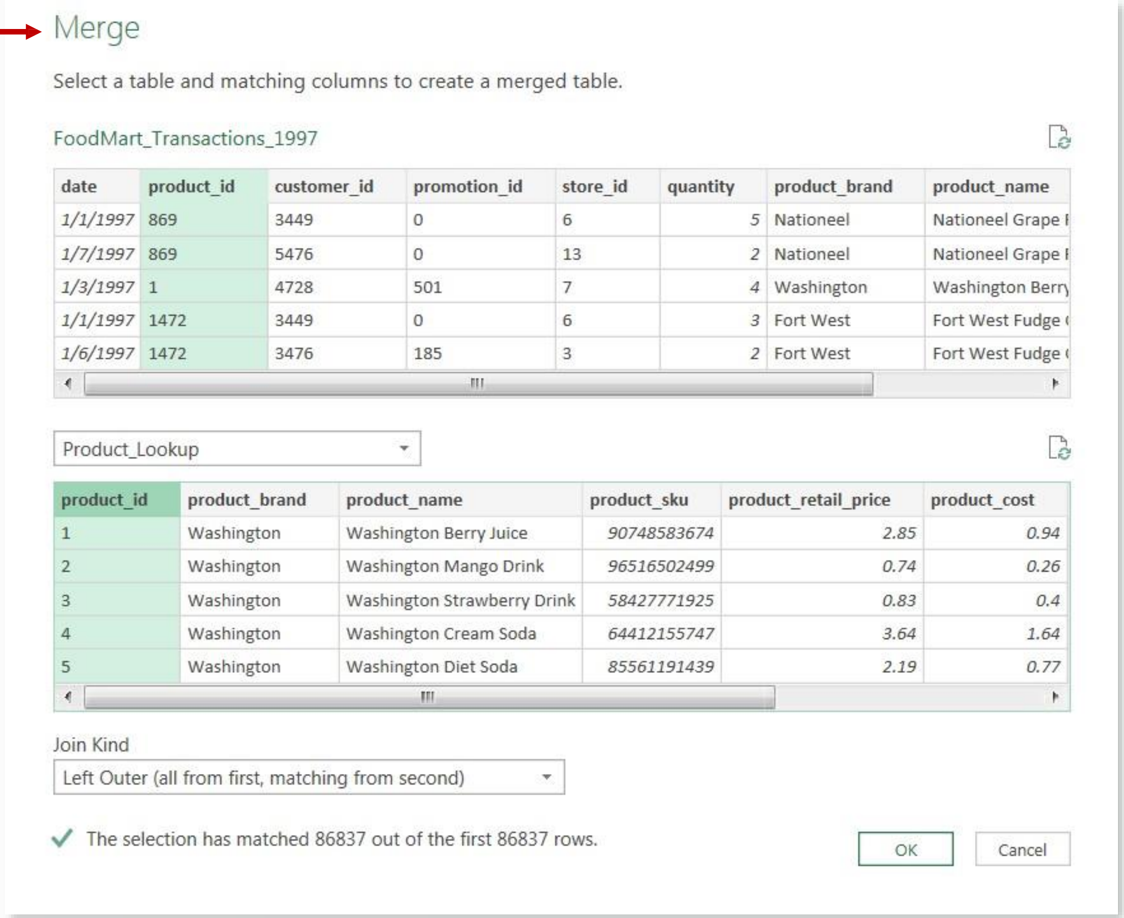
Join Kind

Left Outer (all from first, matching from second)

✓ The selection has matched 86837 out of the first 86837 rows.

MERGING QUERIES





Merge

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

FoodMart_Transactions_1997

date	product_id	customer_id	promotion_id	store_id	quantity	product_brand	product_name
1/1/1997	869	3449	0	6	5	Nationeel	Nationeel Grape
1/7/1997	869	5476	0	13	2	Nationeel	Nationeel Grape
1/3/1997	1	4728	501	7	4	Washington	Washington Berry
1/1/1997	1472	3449	0	6	3	Fort West	Fort West Fudge
1/6/1997	1472	3476	185	3	2	Fort West	Fort West Fudge

Product_Lookup

product_id	product_brand	product_name	product_sku	product_retail_price	product_cost
1	Washington	Washington Berry Juice	90748583674	2.85	0.94
2	Washington	Washington Mango Drink	96516502499	0.74	0.26
3	Washington	Washington Strawberry Drink	58427771925	0.83	0.4
4	Washington	Washington Cream Soda	64412155747	3.64	1.64
5	Washington	Washington Diet Soda	85561191439	2.19	0.77

Join Kind

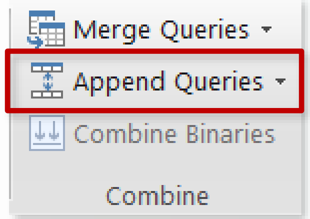
Left Outer (all from first, matching from second)

✓ The selection has matched 86837 out of the first 86837 rows.

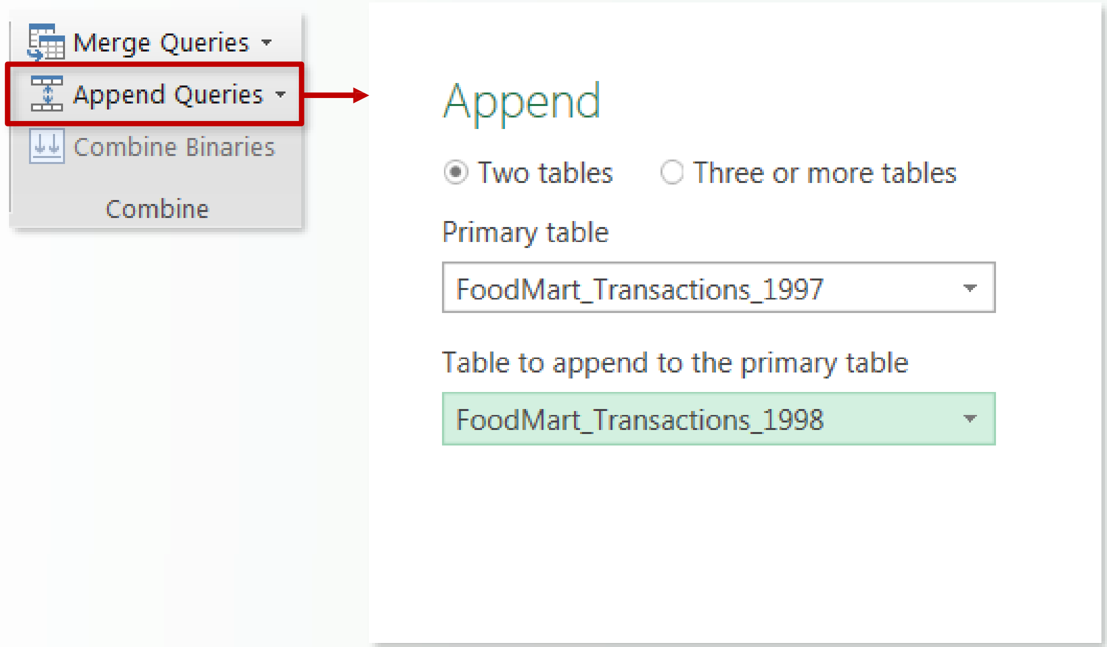
- Merging queries allows you to **join tables** based on a common column (like VLOOKUP)
- In this case we're merging the **FoodMart_Transactions_1997** table with the **Product_Lookup** table, which share a "*product_id*" column

TIP: Merging **adds columns** to an existing table

APPENDING QUERIES



APPENDING QUERIES



Append

● Two tables ○ Three or more tables

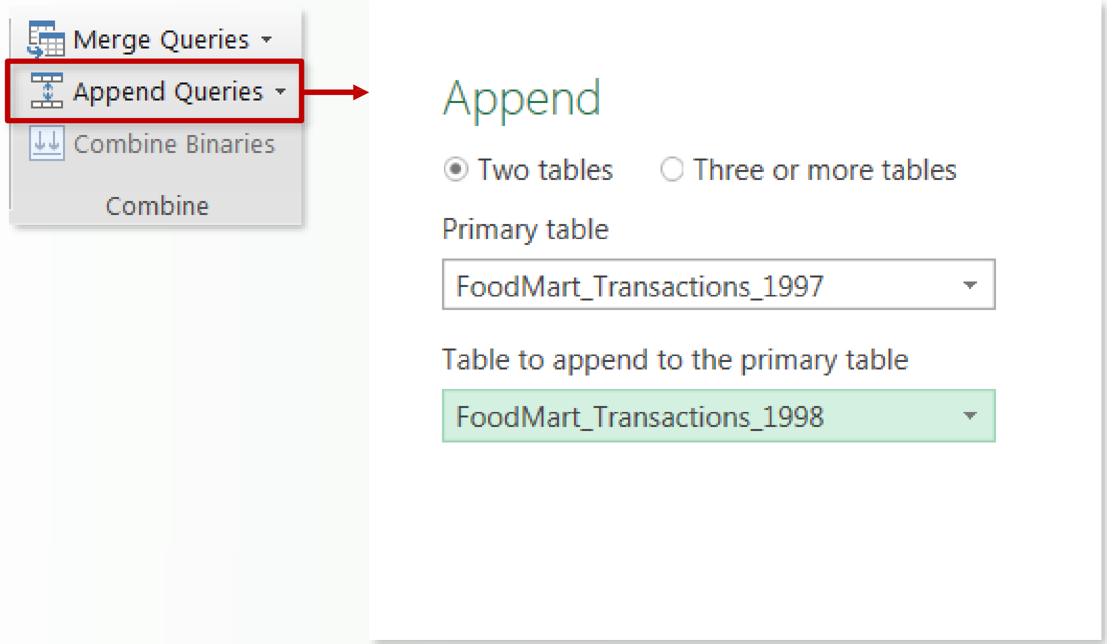
Primary table

FoodMart_Transactions_1997

Table to append to the primary table

FoodMart_Transactions_1998

APPENDING QUERIES



Append

● Two tables ○ Three or more tables

Primary table

FoodMart_Transactions_1997

Table to append to the primary table

FoodMart_Transactions_1998

- Appending queries allows you to **combine** (or ***stack***) tables that share a common structure and set of columns

- In this case we're appending the **FoodMart_Transactions_1998** table to the **FoodMart_Transactions_1997** table, since they contain the same set of columns and data types

TIP: Appending **adds rows** to an existing table

DEMO

GET DATA FROM:

- **CSV:** FROM DESKTOP
- **FOLDER:** FROM DESKTOP
- **WEB:**
  - **HTTPS://COVID19.NCDC.GOV.NG/**
  - **HTTPS://EN.WIKIPEDIA.ORG/WIKI/TEMPLATE:COVID-19_PANDEMIC_DATA**
- **GITHUB:** **HTTPS://GITHUB.COM/GPEARLS/DATA**
- **ODATA FEED:** HTTPS://SERVICES.ODATA.ORG/NORTHWIND/NORTHWIND.SVC/

PERFORMANCE TRANSFORMATION IN QUERY EDITOR:

- **NUMBER**
- **TEXT**
- **DATE**
- **APPEND**
- **MERGE**
- **FOLDER REFERENCE FOR MERGE ACTION**

MEET POWER PIVOT

MEET POWER PIVOT

A **“Power” Pivot** is just like a normal PivotTable, except it sits on top of an *entire data model* rather than a single table or range. This allows you to:

MEET POWER PIVOT

A **“Power” Pivot** is just like a normal PivotTable, except it sits on top of an *entire data model* rather than a single table or range. This allows you to:

- Explore massive datasets consisting of multiple sources and tables, using familiar, user-friendly PivotTable tools and options

MEET POWER PIVOT

A **“Power” Pivot** is just like a normal PivotTable, except it sits on top of an *entire data model* rather than a single table or range. This allows you to:

- Explore massive datasets consisting of multiple sources and tables, using familiar, user-friendly PivotTable tools and options
- Create powerful and flexible calculations using Data Analysis Expressions (DAX)

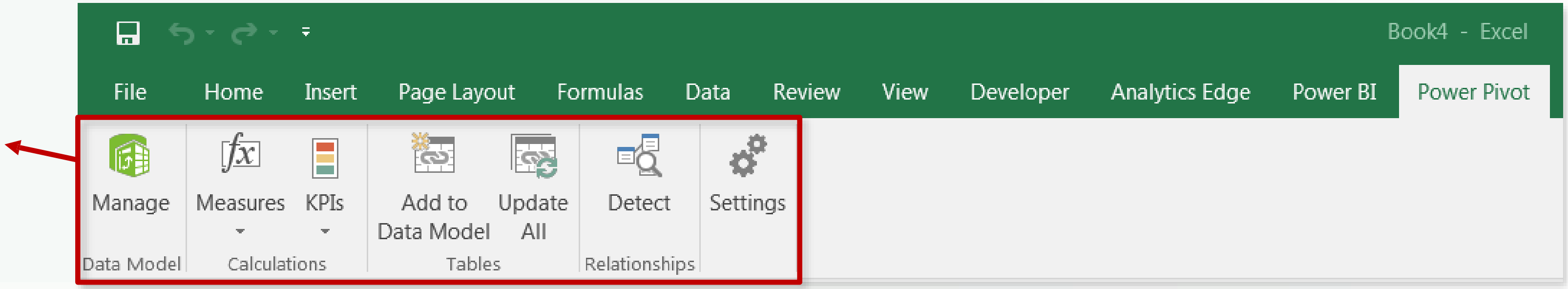
MEET POWER PIVOT

A **“Power” Pivot** is just like a normal PivotTable, except it sits on top of an *entire data model* rather than a single table or range. This allows you to:

- Explore massive datasets consisting of multiple sources and tables, using familiar, user-friendly PivotTable tools and options
- Create powerful and flexible calculations using Data Analysis Expressions (DAX)

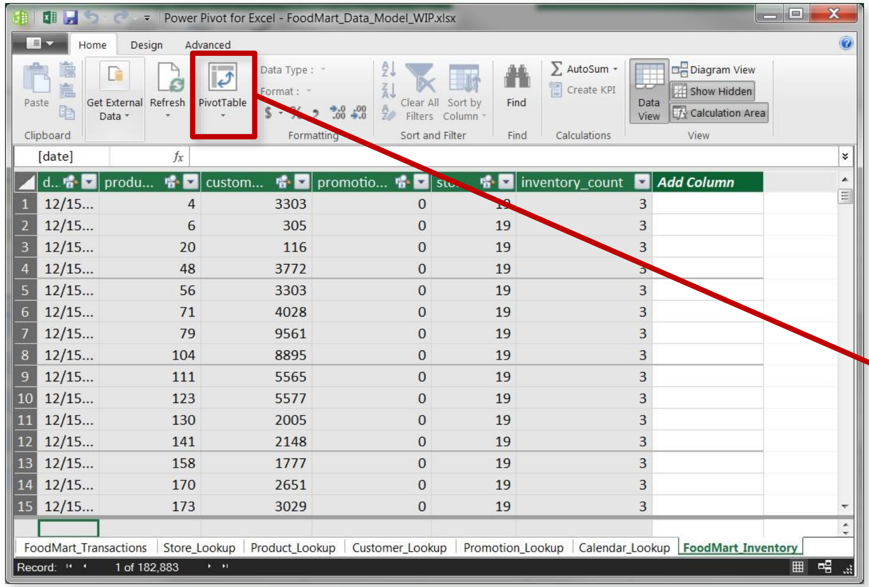
The **Power Pivot** tab includes tools to manage the data model and define new measures

(***Note:*** *you may need to enable this tab by selecting* ***File > Options > Add-Ins > Manage COM Add-Ins****)*



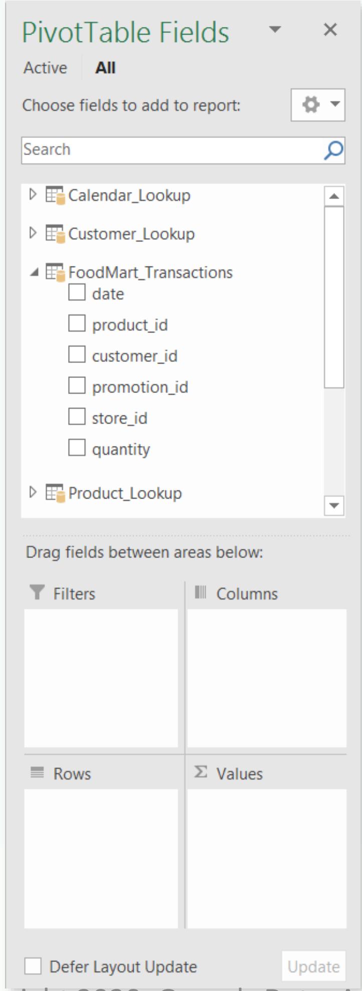
CREATING A “POWER” PIVOT TABLE

Option #1: From the Data Model



	d...	produ...	custom...	promotio...	sto...	inventory_count	Add Column
1	12/15...	4	3303	0	19	3	
2	12/15...	6	305	0	19	3	
3	12/15...	20	116	0	19	3	
4	12/15...	48	3772	0	19	3	
5	12/15...	56	3303	0	19	3	
6	12/15...	71	4028	0	19	3	
7	12/15...	79	9561	0	19	3	
8	12/15...	104	8895	0	19	3	
9	12/15...	111	5565	0	19	3	
10	12/15...	123	5577	0	19	3	
11	12/15...	130	2005	0	19	3	
12	12/15...	141	2148	0	19	3	
13	12/15...	158	1777	0	19	3	
14	12/15...	170	2651	0	19	3	
15	12/15...	173	3029	0	19	3	

FoodMart_Transactions | Store_Lookup | Product_Lookup | Customer_Lookup | Promotion_Lookup | Calendar_Lookup | FoodMart_Inventory



PivotTable Fields

Active   All

Choose fields to add to report:

Search

- Calendar_Lookup
- Customer_Lookup
- FoodMart_Transactions
  - □ date
  - □ product_id
  - □ customer_id
  - □ promotion_id
  - □ store_id
  - □ quantity
- Product_Lookup

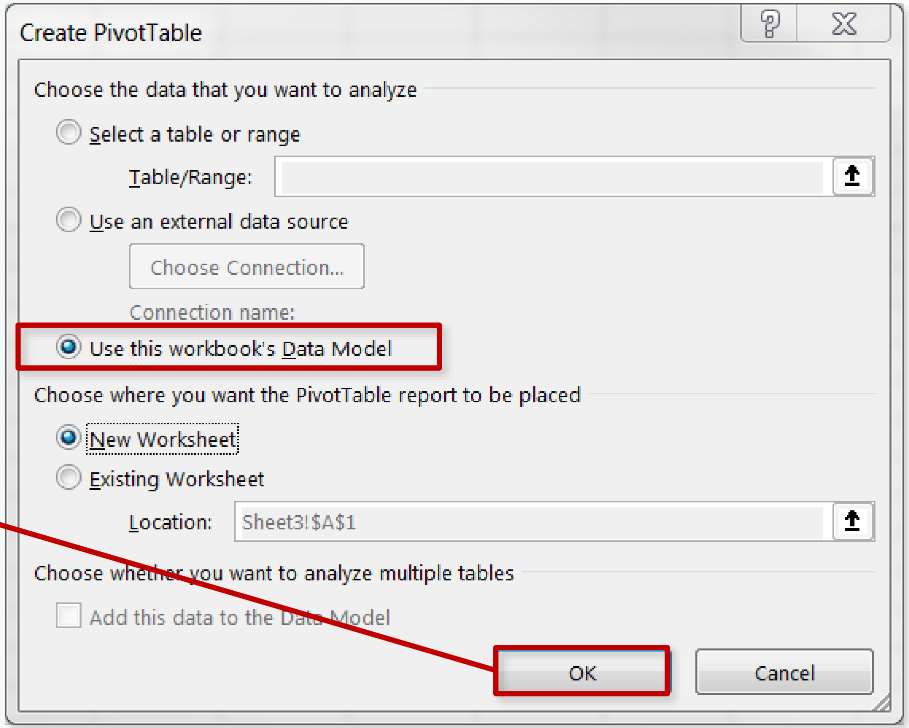
Drag fields between areas below:

Filters | Columns

Rows | Values

□ Defer Layout Update   Update

Option #2: From the *Insert* > *PivotTable* dialog box



Create PivotTable

Choose the data that you want to analyze

○ Select a table or range

Table/Range:

○ Use an external data source

Choose Connection...

Connection name:

● Use this workbook's Data Model

Choose where you want the PivotTable report to be placed

● New Worksheet

○ Existing Worksheet

Location: Sheet3!A1

Choose whether you want to analyze multiple tables

□ Add this data to the Data Model

OK   Cancel

Gpearls

*Copyright 2020, Gpearls Data Analytics, https://gpearls.com.ng

MEET DATA MODEL

MEET EXCEL’S DATA MODEL

The **Data Model** provides simple and intuitive tools for building relational databases directly in Excel. With the data model you can:

MEET DATA MODEL

The **Data Model** provides simple and intuitive tools for building relational databases directly in Excel. With the data model you can:

- Manage massive datasets that can’t fit into worksheets

MEET DATA MODEL

The **Data Model** provides simple and intuitive tools for building relational databases directly in Excel. With the data model you can:

- Manage massive datasets that can’t fit into worksheets
- Create table relationships to blend data across multiple sources

MEET DATA MODEL

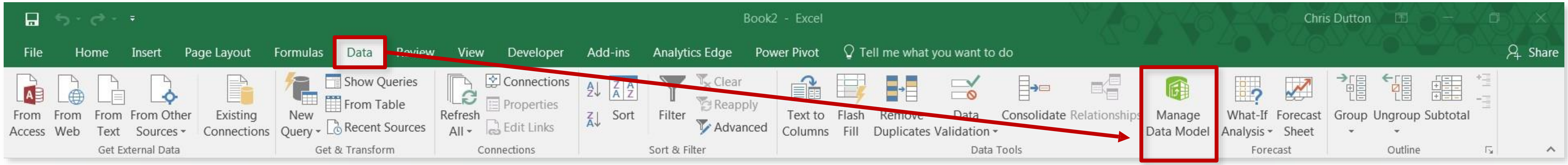
The **Data Model** provides simple and intuitive tools for building relational databases directly in Excel. With the data model you can:

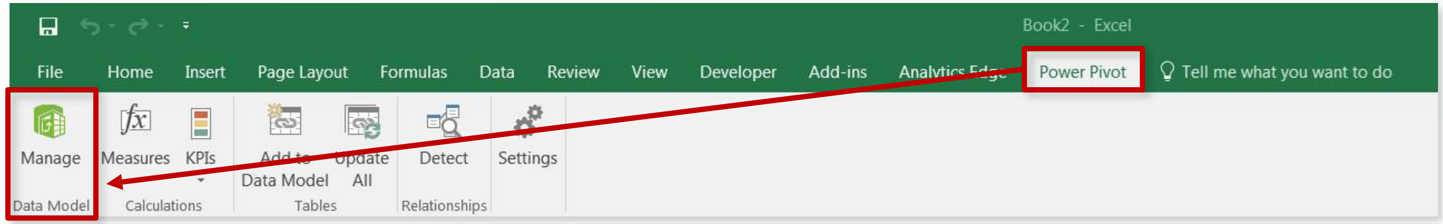
- Manage massive datasets that can’t fit into worksheets
- Create table relationships to blend data across multiple sources
- Define custom hierarchies and perspectives

MEET DATA MODEL

The **Data Model** provides simple and intuitive tools for building relational databases directly in Excel. With the data model you can:

- Manage massive datasets that can't fit into worksheets
- Create table relationships to blend data across multiple sources
- Define custom hierarchies and perspectives

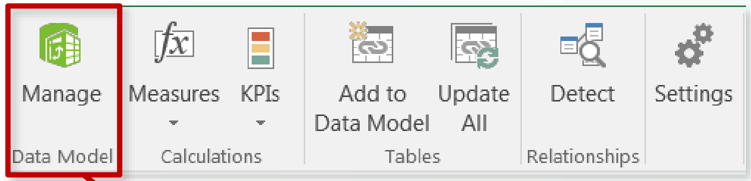




Access the **Data Model** through the **Power Pivot** tab or the **Data** tab

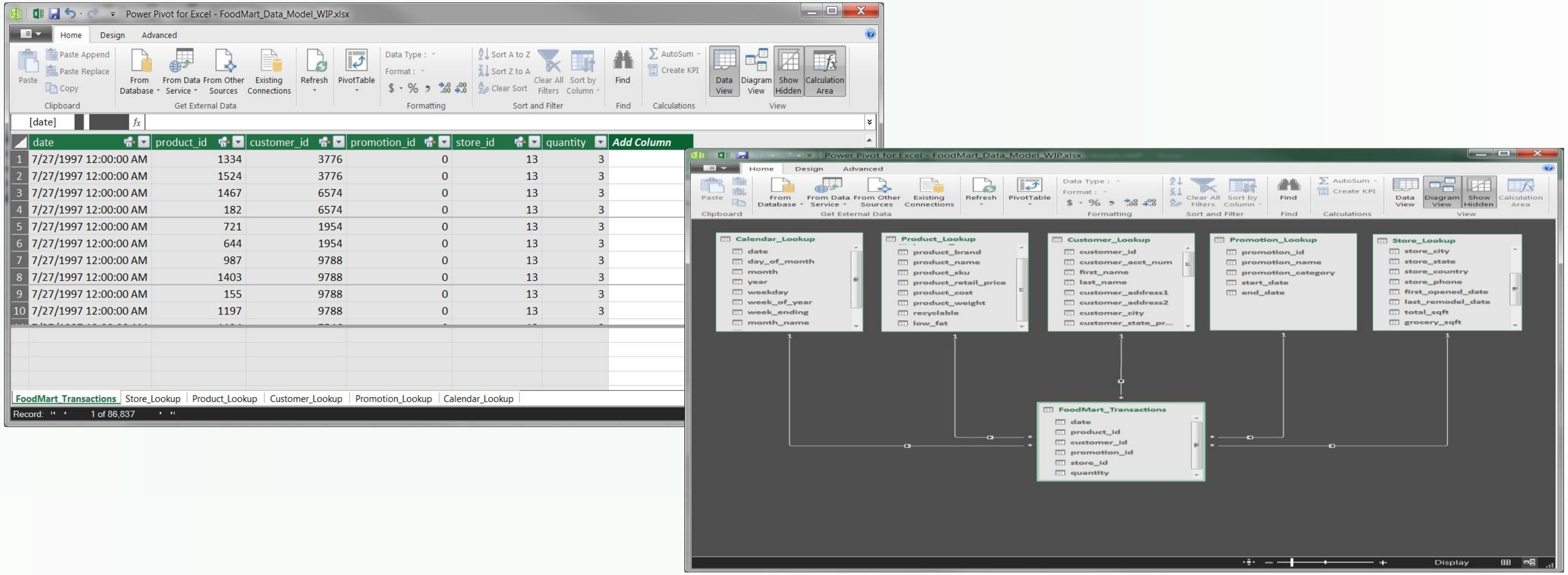
*(**Note:** you may need to enable the Power Pivot tab via **File > Options > Add-Ins > Manage COM Add-Ins**)*

THE DATA MODEL WINDOW



The **Data Model** opens in a separate Excel window, where you can view your data tables, calculate new measures, and define table relationships

Note: *Closing the Data Model window does NOT close your Excel workbook*



	date	product_id	customer_id	promotion_id	store_id	quantity	Add Column
1	7/27/1997 12:00:00 AM	1334	3776	0	13	3	
2	7/27/1997 12:00:00 AM	1524	3776	0	13	3	
3	7/27/1997 12:00:00 AM	1467	6574	0	13	3	
4	7/27/1997 12:00:00 AM	182	6574	0	13	3	
5	7/27/1997 12:00:00 AM	721	1954	0	13	3	
6	7/27/1997 12:00:00 AM	644	1954	0	13	3	
7	7/27/1997 12:00:00 AM	987	9788	0	13	3	
8	7/27/1997 12:00:00 AM	1403	9788	0	13	3	
9	7/27/1997 12:00:00 AM	155	9788	0	13	3	
10	7/27/1997 12:00:00 AM	1197	9788	0	13	3	

DEMO

LET’S EXPLORE THE FLOW FROM POWER QUERY TO POWER PIVOT

COMMON FUNCTION CATEGORIES

COMMON FUNCTION CATEGORIES

MATH & STATS
Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Gpearls

COMMON FUNCTION CATEGORIES

MATH & STATS
Functions

*Basic **aggregation** functions as well as “**iterators**” evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

COMMON FUNCTION CATEGORIES

MATH & STATS
Functions

*Basic **aggregation** functions as well as “**iterators**” evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Basic **aggregation** functions as well as “**iterators**” evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions for returning information about values in a given **conditional expression***

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions for returning information about values in a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

Gpearls

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions for returning information about values in a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

*Functions to manipulate **text strings** or **control formats** for dates, times or numbers*

Gpearls

COMMON FUNCTION CATEGORIES

MATH & STATS
Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL
Functions

*Functions for returning information about values in a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT
Functions

*Functions to manipulate **text strings** or **control formats** for dates, times or numbers*

Common Examples:

- CONCATENATE
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- PROPER
- LEN
- SEARCH/FIND
- REPLACE
- REPT
- SUBSTITUTE
- TRIM
- UNICHAR

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Basic **aggregation** functions as well as “**iterators**” evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions for returning information about values in a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

*Functions to manipulate **text strings** or **control formats** for dates, times or numbers*

Common Examples:

- CONCATENATE
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- PROPER
- LEN
- SEARCH/FIND
- REPLACE
- REPT
- SUBSTITUTE
- TRIM
- UNICHAR

FILTER Functions

***Lookup** functions based on related tables and **filtering** functions for dynamic calculations*

Gpearls

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions for returning information about values in a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

*Functions to manipulate **text strings** or **control formats** for dates, times or numbers*

Common Examples:

- CONCATENATE
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- PROPER
- LEN
- SEARCH/FIND
- REPLACE
- REPT
- SUBSTITUTE
- TRIM
- UNICHAR

FILTER Functions

***Lookup** functions based on related tables and **filtering** functions for dynamic calculations*

Common Examples:

- CALCULATE
- FILTER
- ALL
- ALLEXCEPT
- RELATED
- RELATEDTABLE
- DISTINCT
- VALUES
- EARLIER/EARLIEST
- HASONEVALUE
- HASONEFILTER
- ISFILTERED
- USERELATIONSHIP

Gpearls

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions for returning information about values in a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

*Functions to manipulate **text strings** or **control formats** for dates, times or numbers*

Common Examples:

- CONCATENATE
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- PROPER
- LEN
- SEARCH/FIND
- REPLACE
- REPT
- SUBSTITUTE
- TRIM
- UNICHAR

FILTER Functions

***Lookup** functions based on related tables and **filtering** functions for dynamic calculations*

Common Examples:

- CALCULATE
- FILTER
- ALL
- ALLEXCEPT
- RELATED
- RELATEDTABLE
- DISTINCT
- VALUES
- EARLIER/EARLIEST
- HASONEVALUE
- HASONEFILTER
- ISFILTERED
- USERELATIONSHIP

DATE & TIME Functions

*Basic **date and time** functions as well as advanced **time intelligence** operations*

Common Examples:

- DATEDIFF
- YEARFRAC
- YEAR/MONTH/DAY
- HOUR/MINUTE/SECOND
- TODAY/NOW
- WEEKDAY/WEEKNUM

Gpearls

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions for returning information about values in a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

*Functions to manipulate **text strings** or **control formats** for dates, times or numbers*

Common Examples:

- CONCATENATE
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- PROPER
- LEN
- SEARCH/FIND
- REPLACE
- REPT
- SUBSTITUTE
- TRIM
- UNICHAR

FILTER Functions

***Lookup** functions based on related tables and **filtering** functions for dynamic calculations*

Common Examples:

- CALCULATE
- FILTER
- ALL
- ALLEXCEPT
- RELATED
- RELATEDTABLE
- DISTINCT
- VALUES
- EARLIER/EARLIEST
- HASONEVALUE
- HASONEFILTER
- ISFILTERED
- USERELATIONSHIP

DATE & TIME Functions

*Basic **date and time** functions as well as advanced **time intelligence** operations*

Common Examples:

- DATEDIFF
- YEARFRAC
- YEAR/MONTH/DAY
- HOUR/MINUTE/SECOND
- TODAY/NOW
- WEEKDAY/WEEKNUM

Time Intelligence Functions:

- DATESYTD
- DATESQTD
- DATESMTD
- DATEADD
- DATESINPERIOD

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions for returning information about values in a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

*Functions to manipulate **text strings** or **control formats** for dates, times or numbers*

Common Examples:

- CONCATENATE
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- PROPER
- LEN
- SEARCH/FIND
- REPLACE
- REPT
- SUBSTITUTE
- TRIM
- UNICHAR

FILTER Functions

***Lookup** functions based on related tables and **filtering** functions for dynamic calculations*

Common Examples:

- CALCULATE
- FILTER
- ALL
- ALLEXCEPT
- RELATED
- RELATEDTABLE
- DISTINCT
- VALUES
- EARLIER/EARLIEST
- HASONEVALUE
- HASONEFILTER
- ISFILTERED
- USERELATIONSHIP

DATE & TIME Functions

*Basic **date and time** functions as well as advanced **time intelligence** operations*

Common Examples:

- DATEDIFF
- YEARFRAC
- YEAR/MONTH/DAY
- HOUR/MINUTE/SECOND
- TODAY/NOW
- WEEKDAY/WEEKNUM

Time Intelligence Functions:

- DATESYTD
- DATESQTD
- DATESMTD
- DATEADD
- DATESINPERIOD

Gpearls

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions for returning information about values in a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

*Functions to manipulate **text strings** or **control formats** for dates, times or numbers*

Common Examples:

- CONCATENATE
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- PROPER
- LEN
- SEARCH/FIND
- REPLACE
- REPT
- SUBSTITUTE
- TRIM
- UNICHAR

FILTER Functions

***Lookup** functions based on related tables and **filtering** functions for dynamic calculations*

Common Examples:

- CALCULATE
- FILTER
- ALL
- ALLEXCEPT
- RELATED
- RELATEDTABLE
- DISTINCT
- VALUES
- EARLIER/EARLIEST
- HASONEVALUE
- HASONEFILTER
- ISFILTERED
- USERELATIONSHIP

DATE & TIME Functions

*Basic **date and time** functions as well as advanced **time intelligence** operations*

Common Examples:

- DATEDIFF
- YEARFRAC
- YEAR/MONTH/DAY
- HOUR/MINUTE/SECOND
- TODAY/NOW
- WEEKDAY/WEEKNUM

Time Intelligence Functions:

- DATESYTD
- DATESQTD
- DATESMTD
- DATEADD
- DATESINPERIOD

***Note:** *This is NOT a comprehensive list (does not include trigonometry functions, parent/child functions, information functions, or other less common functions)*

DAX SYNTAX

Total Quantity: =SUM(Transactions[quantity])

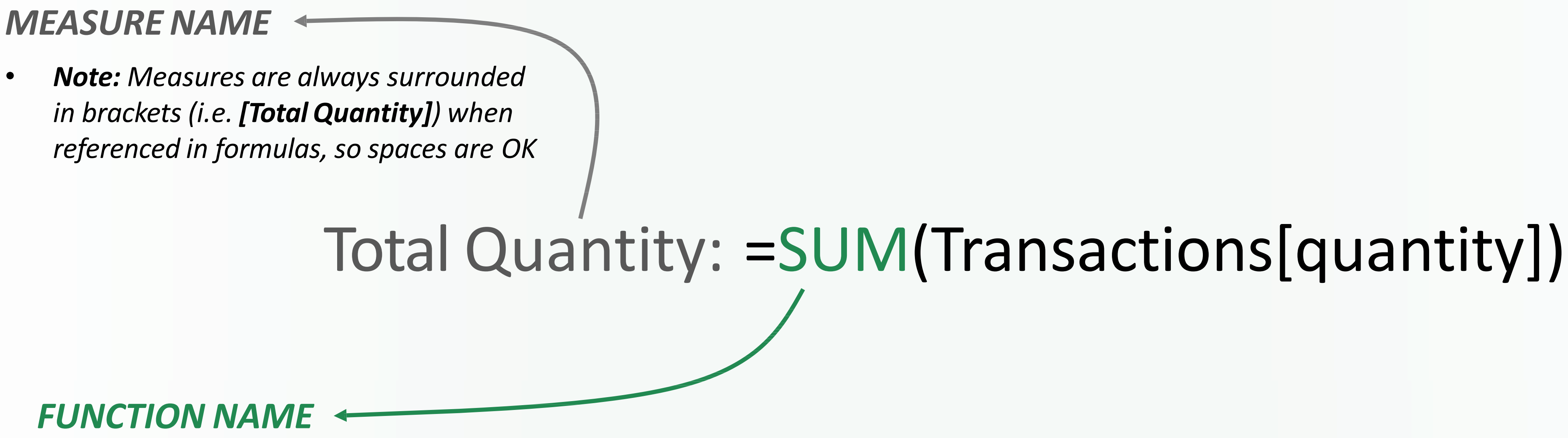
DAX SYNTAX

MEASURE NAME

- ***Note:*** *Measures are always surrounded in brackets (i.e.* ***[Total Quantity]****) when referenced in formulas, so spaces are OK*

Total Quantity: =SUM(Transactions[quantity])

DAX SYNTAX



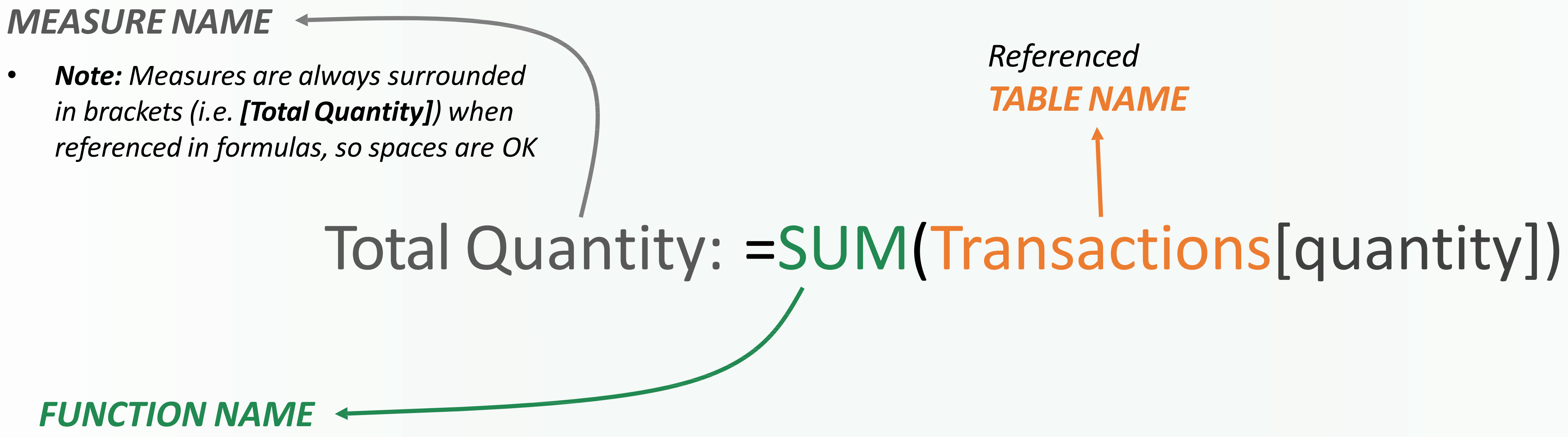
MEASURE NAME

- ***Note:*** *Measures are always surrounded in brackets (i.e.* ***[Total Quantity]****) when referenced in formulas, so spaces are OK*

Total Quantity: =SUM(Transactions[quantity])

FUNCTION NAME

DAX SYNTAX



MEASURE NAME

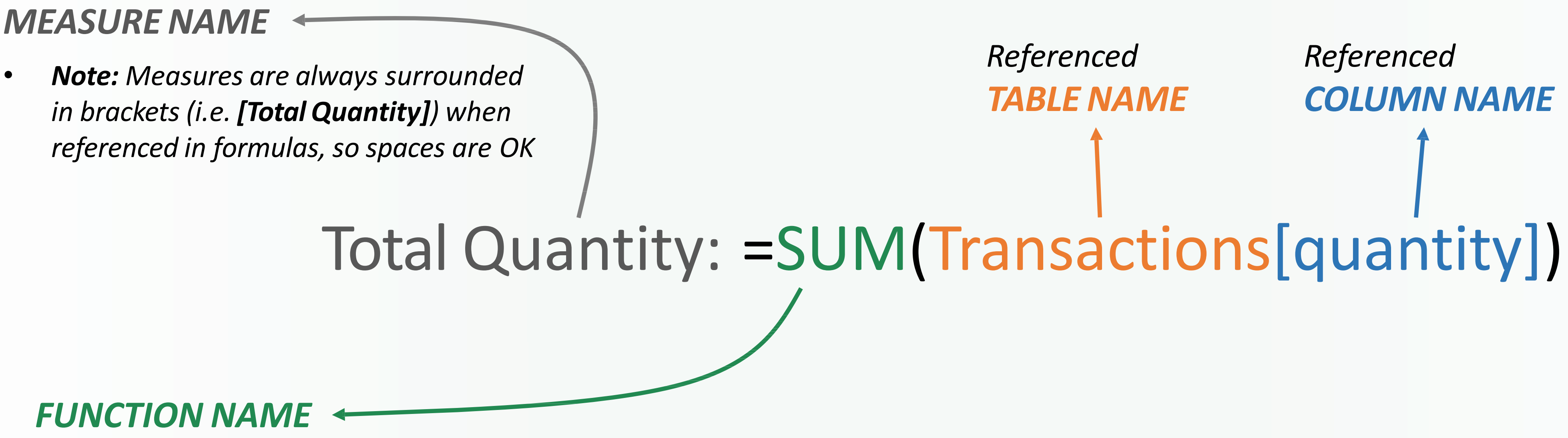
- ***Note:*** *Measures are always surrounded in brackets (i.e.* ***[Total Quantity]****) when referenced in formulas, so spaces are OK*

Referenced
TABLE NAME

Total Quantity: =SUM(Transactions[quantity])

FUNCTION NAME

DAX SYNTAX



MEASURE NAME

- ***Note:*** *Measures are always surrounded in brackets (i.e.* ***[Total Quantity]****) when referenced in formulas, so spaces are OK*

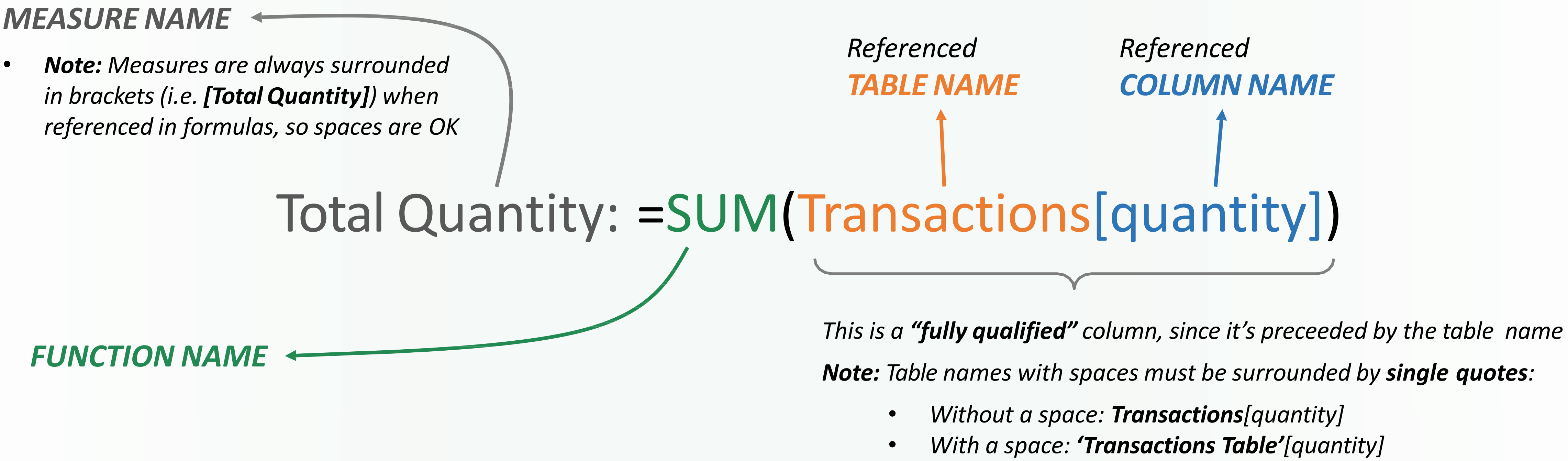
Referenced
TABLE NAME

Referenced
COLUMN NAME

Total Quantity: =SUM(Transactions[quantity])

FUNCTION NAME

DAX SYNTAX



MEASURE NAME

- ***Note:*** *Measures are always surrounded in brackets (i.e.* ***[Total Quantity]****) when referenced in formulas, so spaces are OK*

Referenced ***TABLE NAME***

Referenced ***COLUMN NAME***

Total Quantity: =SUM(Transactions[quantity])

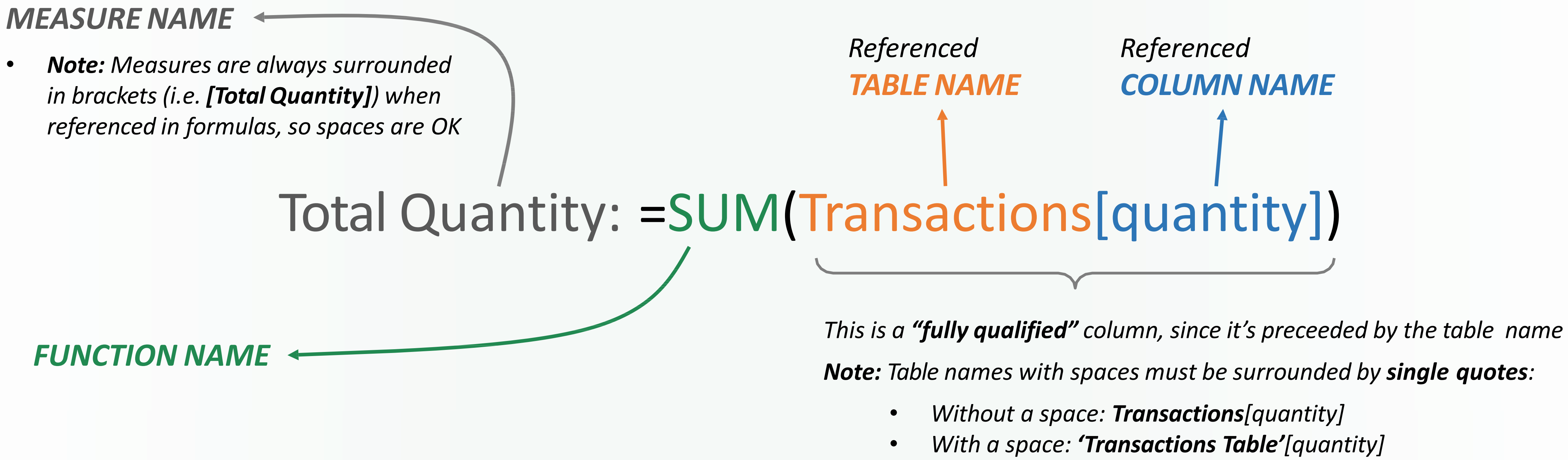
FUNCTION NAME

This is a ***"fully qualified"*** *column, since it's preceeded by the table name*

Note: *Table names with spaces must be surrounded by* ***single quotes****:*

- *Without a space:* ***Transactions****[quantity]*
- *With a space:* ***'Transactions Table'****[quantity]*

DAX SYNTAX



MEASURE NAME

- ***Note:*** *Measures are always surrounded in brackets (i.e.* ***[Total Quantity]****) when referenced in formulas, so spaces are OK*

Referenced ***TABLE NAME***

Referenced ***COLUMN NAME***

Total Quantity: =SUM(Transactions[quantity])

FUNCTION NAME

This is a ***"fully qualified"*** *column, since it's preceeded by the table name*

Note: *Table names with spaces must be surrounded by* ***single quotes****:*

- *Without a space:* ***Transactions****[quantity]*
- *With a space:* ***'Transactions Table'****[quantity]*

For ***column*** *references, use the fully qualified name (i.e.* ***Table[Column]****)*
For ***measure*** *references, just use the measure name (i.e.* ***[Measure]****)*

BASIC MATH & STATS FUNCTIONS

SUM()

Evaluates the sum of a column

=**SUM**(<column>)

BASIC MATH & STATS FUNCTIONS

SUM()	*Evaluates the sum of a column*	=**SUM**(<column>)
AVERAGE()	*Returns the average (arithmetic mean) of all the numbers in a column*	=**AVERAGE**(<column>)

BASIC MATH & STATS FUNCTIONS

SUM()	*Evaluates the sum of a column*	=**SUM**(<column>)
AVERAGE()	*Returns the average (arithmetic mean) of all the numbers in a column*	=**AVERAGE**(<column>)
MAX()	*Returns the largest value in a column or between two scalar expressions*	=**MAX**(<column>) *or* =**MAX**(<exp1>, <exp2>)

BASIC MATH & STATS FUNCTIONS

SUM()	*Evaluates the sum of a column*	=**SUM**(<column>)
AVERAGE()	*Returns the average (arithmetic mean) of all the numbers in a column*	=**AVERAGE**(<column>)
MAX()	*Returns the largest value in a column or between two scalar expressions*	=**MAX**(<column>) *or* =**MAX**(<exp1>, <exp2>)
MIN()	*Returns the smallest value in a column or between two scalar expressions*	=**MIN**(<column>) *or* =**MIN**(<exp1>, <exp2>)

BASIC MATH & STATS FUNCTIONS

SUM()	*Evaluates the sum of a column*	=**SUM**(<column>)
AVERAGE()	*Returns the average (arithmetic mean) of all the numbers in a column*	=**AVERAGE**(<column>)
MAX()	*Returns the largest value in a column or between two scalar expressions*	=**MAX**(<column>) *or* =**MAX**(<exp1>, <exp2>)
MIN()	*Returns the smallest value in a column or between two scalar expressions*	=**MIN**(<column>) *or* =**MIN**(<exp1>, <exp2>)
DIVIDE()	*Performs division and returns the alternate result (or blank) if div/0*	=**DIVIDE**(<numerator>, <denominator>, *<other>*)

COUNT, COUNTA, DISTINCTCOUNT & COUNTROWS

COUNTROWS()

Counts the number of rows in the specified table, or a table defined by an expression

=**COUNTROWS**(<table>)

Gpearls

*Copyright 2020, Gpearls Data Analytics,
https://gpearls.com.ng

COUNT, COUNTA, DISTINCTCOUNT & COUNTROWS

COUNTROWS()	*Counts the number of rows in the specified table, or a table defined by an expression*	=**COUNTROWS**(<table>)
COUNT()	*Counts the number of cells in a column that contain numbers*	=**COUNT**(<column>)

COUNT, COUNTA, DISTINCTCOUNT & COUNTROWS

COUNTROWS()

Counts the number of rows in the specified table, or a table defined by an expression

=**COUNTROWS**(<table>)

COUNT()

Counts the number of cells in a column that contain numbers

=**COUNT**(<column>)

COUNTA()

Counts the number of non-empty cells in a column (numerical and non-numerical)

=**COUNTA**(<column>)

COUNT, COUNTA, DISTINCTCOUNT & COUNTROWS

COUNTROWS()	*Counts the number of rows in the specified table, or a table defined by an expression*	=**COUNTROWS**(<table>)
COUNT()	*Counts the number of cells in a column that contain numbers*	=**COUNT**(<column>)
COUNTA()	*Counts the number of non-empty cells in a column (numerical and non-numerical)*	=**COUNTA**(<column>)
DISTINCTCOUNT()	*Counts the number of different cells in a column of numbers*	=**DISTINCTCOUNT**(<column>)

BASIC LOGICAL FUNCTIONS (IF/AND/OR)

IF()

Checks if a given condition is met, and returns one value if the condition is TRUE, and another if the condition is FALSE

=**IF**(<logical test>, <value_if_true>, *<value_if_false>*)

BASIC LOGICAL FUNCTIONS (IF/AND/OR)

IF()

Checks if a given condition is met, and returns one value if the condition is TRUE, and another if the condition is FALSE

=**IF**(<logical test>, <value_if_true>, *<value_if_false>*)

IFERROR()

Evaluates an expression and returns a specified value if the expression returns an error, otherwise returns the expression itself

=**IFERROR**(value, value_if_error)

BASIC LOGICAL FUNCTIONS (IF/AND/OR)

IF()

Checks if a given condition is met, and returns one value if the condition is TRUE, and another if the condition is FALSE

=**IF**(<logical test>, <value_if_true>, *<value_if_false>*)

IFERROR()

Evaluates an expression and returns a specified value if the expression returns an error, otherwise returns the expression itself

=**IFERROR**(value, value_if_error)

AND()

Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE, otherwise returns FALSE

=**AND**(<logical1>, <logical2>)

BASIC LOGICAL FUNCTIONS (IF/AND/OR)

IF()	*Checks if a given condition is met, and returns one value if the condition is TRUE, and another if the condition is FALSE*	=**IF**(<logical test>, <value_if_true>, *<value_if_false>*)
IFERROR()	*Evaluates an expression and returns a specified value if the expression returns an error, otherwise returns the expression itself*	=**IFERROR**(value, value_if_error)
AND()	*Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE, otherwise returns FALSE*	=**AND**(<logical1>, <logical2>)
OR()	*Checks whether one of the arguments is TRUE to return TRUE, and returns FALSE if both arguments are FALSE*	=**OR** (<logical1>, <logical2>)

BASIC LOGICAL FUNCTIONS (IF/AND/OR)

IF()	*Checks if a given condition is met, and returns one value if the condition is TRUE, and another if the condition is FALSE*	=**IF**(<logical test>, <value_if_true>, *<value_if_false>*)
IFERROR()	*Evaluates an expression and returns a specified value if the expression returns an error, otherwise returns the expression itself*	=**IFERROR**(value, value_if_error)
AND()	*Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE, otherwise returns FALSE*	=**AND**(<logical1>, <logical2>)
OR()	*Checks whether one of the arguments is TRUE to return TRUE, and returns FALSE if both arguments are FALSE*	=**OR** (<logical1>, <logical2>)

Note: *Use the* ***&&*** *and* ***||*** *operators if you want to include more than two conditions!*

Gpearls

SWITCH & SWITCH(TRUE)

SWITCH()

Evaluates an expression against a list of values and returns one of multiple possible result expressions

=**SWITCH**(<expression>, <value1>, <result1>, <value2>, <result2>, … *<else>*)

Any DAX expression that returns a single scalar value, evaluated multiple times (for each row/constant)

Examples:

- *Calendar_Lookup[month_num]*
- *Product_Lookup[product_brand]*

*List of **values** produced by the expression, each paired with a **result** to return for rows/cases that match*

Examples:

```
=SWITCH(Calendar_Lookup[month_num],
        1, "January",
        2, "February",
        etc…
```

```
=SWITCH(TRUE(),
        [retail_price]<5, "Low Price",
        AND([retail_price>=5, [retail_price]<20), "Med Price",
        AND([retail_price>=20, [retail_price]<50), "High Price"
        "Premium Price")
```

Value returned if the expression doesn't match any value argument

RELATED

RELATED()

Returns related values in each row of a table using relationships with other tables

=**RELATED**(<column>)

↓

The column that contains the values you want to retrieve

Examples:

- *Product_Lookup[product_brand]*
- *Store_Lookup[store_country]*

Gpearls

CALCULATE

CALCULATE()

Evaluates a given expression or formula under a set of defined filters

CALCULATE

CALCULATE()

Evaluates a given expression or formula under a set of defined filters

=**CALCULATE**(<expression>, <filter1>, <filter2>,…)

CALCULATE

CALCULATE()

Evaluates a given expression or formula under a set of defined filters

=**CALCULATE**(<expression>, <filter1>, <filter2>,…)

Name of an existing measure or a formula for a valid measure

Examples:

- *[Total Transactions]*
- *SUM(Transactions[quantity])*

CALCULATE

CALCULATE()

Evaluates a given expression or formula under a set of defined filters

=**CALCULATE**(<expression>, <filter1>, <filter2>,…)

Name of an existing measure or a formula for a valid measure

Examples:

- *[Total Transactions]*
- *SUM(Transactions[quantity])*

*List of simple Boolean (True/False) filter expressions (**note:** these require simple, fixed values; you cannot create filters based on measures)*

CALCULATE

CALCULATE()

Evaluates a given expression or formula under a set of defined filters

=**CALCULATE**(<expression>, <filter1>, <filter2>,…)

Name of an existing measure or a formula for a valid measure

Examples:

- *[Total Transactions]*
- *SUM(Transactions[quantity])*

*List of simple Boolean (True/False) filter expressions (**note:** these require simple, fixed values; you cannot create filters based on measures)*

Examples:

- *Store_Lookup[store_country]**="USA"***
- *Calendar[Year]**=1998***
- *Transactions[quantity]**>=5***

CREATE CALCULATED COLUMN (DEMO)

Table: Calendar Lookup

Weekend

Quarter_Name

Year_Month

End_Of_Month

Table: Store_Lookup

supermarket_size

store_street_num

years_since_remodel

Table: Transactions

Retail_price

Table: Product Lookup

product_price_cat

Table: Customer_Lookup

Age

education_level

customer_priority

New_Country

house_num

membership_level

CREATE MEASURES:(DEMO)

Table: Product Lookup

Avg Retail Price: $2.12

Unique Products: 1,560

Recyclable Products: 873

Max Retail Price: $3.98

Min Retail Price: $0.50

Table: Customer_Lookup

Average Age: 75.1

Total Customers: 10,281

Unique Cities: 108

Table: Product Lookup

Quantity Returned: 8,289

Return Rate: 1.0%

Table: Transactions

Total Quantity: 833,489

Weekend Transactions: 76,608

Weekday Transactions: 193,112

Profit: $1,052,819

Total Revenue (Measure): $1,764,546

Total Cost: $711,728

Total Transactions: 269,720

USA Transactions: 180,823

USA Transactions FILTER: 180,823

USA Transaction %: 67.04%

Low-Fat Quantity Sold: 294,748

Low Fat Sales %: 35.36%

SOURCES