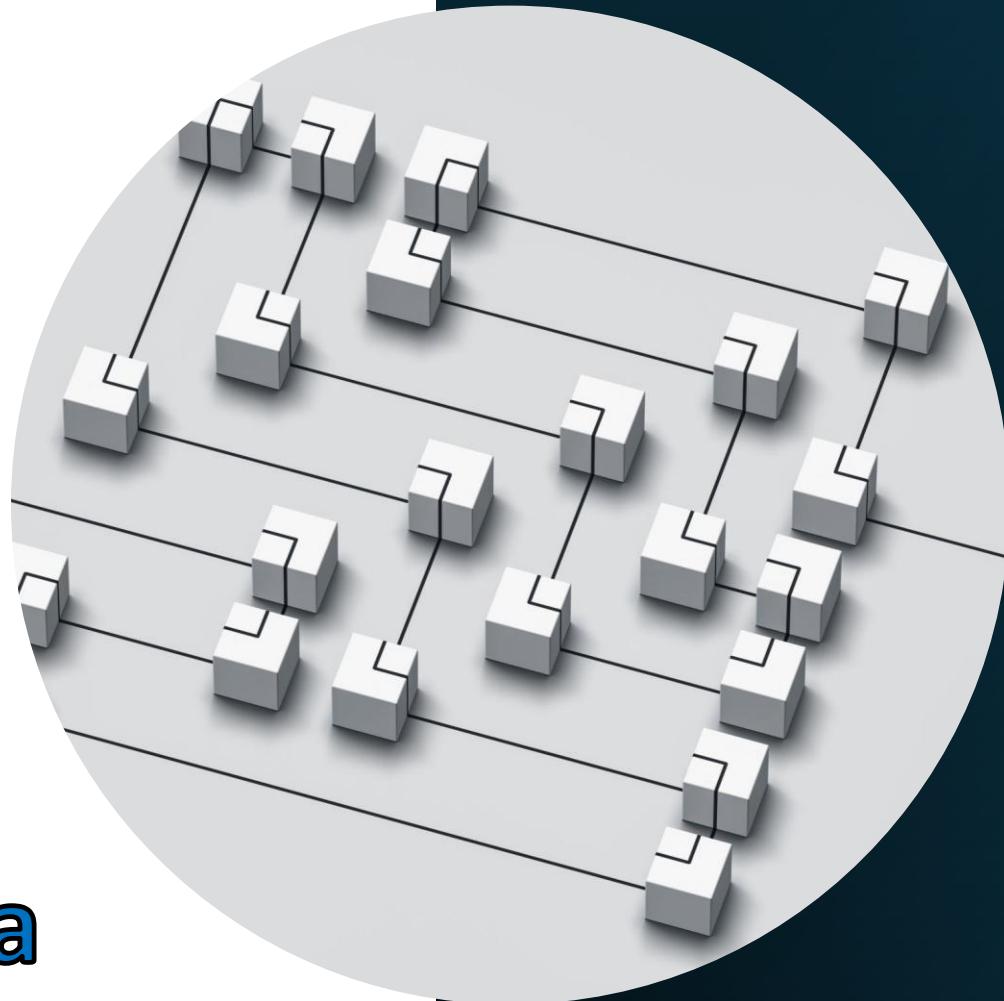


# Basics of Building a Data Model in Power BI

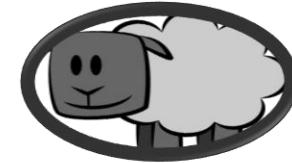
**SQL Saturday Minnesota**

September 2024



# Jason Romans

Senior BI Engineer  
Builder of Models



## The Dax Shepherd



**Lives in Nashville, Tennessee, United States**



**Started as SQL Server DBA**



**Transitioned to the Microsoft BI Stack**



**Work on everything from SQL Server Integration Services,  
SQL Server Database, Analysis Services, and Power BI**



**Simple Talk Author at Redgate**



**Favorite Data Model**

# Shoulders of Giants



This Photo by Unknown Author is licensed under [CC BY-SA NC](#)



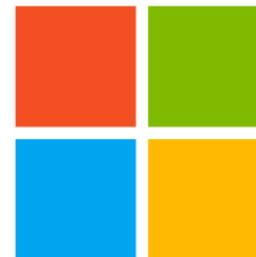
This Photo by Unknown Author is licensed under [CC BY-ND](#)

- Visit the Sponsor Booths
  - Lots of Great Raffle Prizes!
  - Enter your details on Raffle Entry Form & visit Sponsors to fill in Bingo card for Raffle prizes
- 

# Thank you Sponsors!



Gold Sponsors:



Microsoft

---

Silver Sponsors:

TRACE 3

---

Bronze Sponsors:

 redgate

---

Blog Sponsors:



# PASSMN – News/Info



Thanks to all our sponsors of 2024!

We need Speakers & Sponsors for 2025 PASSMN Meetings!

- Sign up to present at one of the monthly meetings!



Monthly Meetup: 3<sup>rd</sup> Tuesday of Each Month

Signup on Meetup:

<https://www.meetup.com/MN-SQL-Server-User-Group-PASSMN/>



Board Member Elections in Q4:

- Your chance to help the MN Data/SQL Community!
  - <https://forms.office.com/r/rWfPhV6Chd>





Connect, share, & learn with peers and thought leaders while celebrating all things data for a week of learning and networking opportunities.

November 4<sup>th</sup> Through November 8<sup>th</sup> in Seattle, WA

<https://passdatacommunitysummit.com/>



# Our Plan

---



1. Intro
2. Power Query
3. Data Model
4. External Tools
5. Conclusion

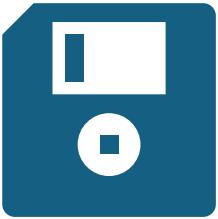
# Our Plan

---

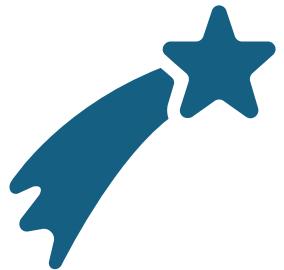


- 1. Intro**
2. Power Query
3. Data Model
4. External Tools
5. Conclusion

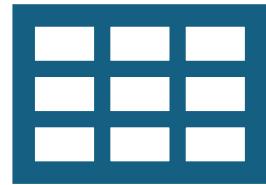
# Daunting Task



How do I bring in the data



What should the model  
look like



Visualize Data

What you  
picture

---



# Reality



Encourage not  
discourage

---



# Warner Bros. Studio Tour London

---

## MODEL MAKING

'Be it model railways or dollhouses, models have played a part in all of our lives.

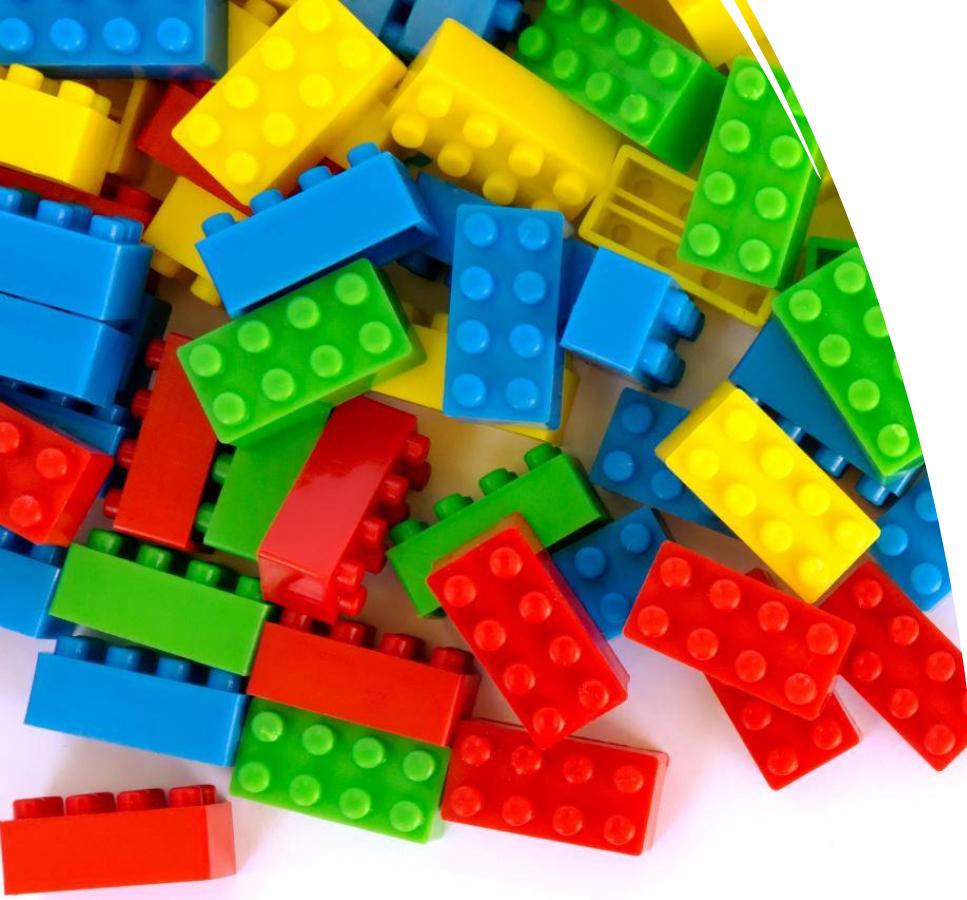
**We can all relate to miniatures.'**

– Gary Tomkins, Art Director, Models



Leavesden's Model Making Department constructed many of the wizarding world's most iconic locales as *picture models*, or perfect miniatures of the on-camera sets. Their full-colour, hand-painted models were crafted with the tiniest details, including foliage, stained glass and working lights. Along with Hogwarts castle, the talented crew also built miniatures of Hogsmeade village, the Owley and the Weasley Burrow, a model that took 14 weeks to build and just six minutes to burn for *Harry Potter and the Half-Blood Prince*.





# Instructions

---

- No step-by-step instructions
- Lego blocks
- Guidelines

# Our Plan

---



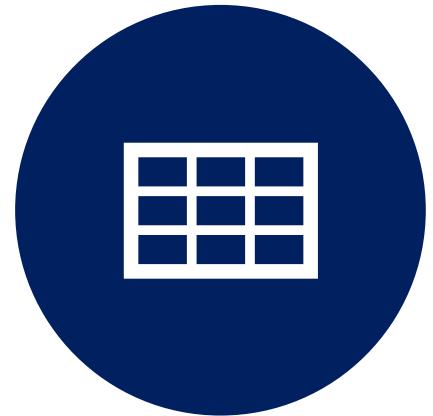
1. Intro
2. Power Query
3. Data Model
4. External Tools
5. Conclusion



Get Data

---

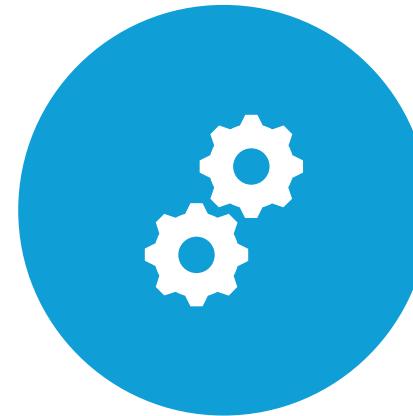
# Where is the data



EXCEL SPREADSHEET

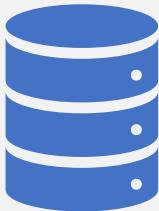


SQL SERVER

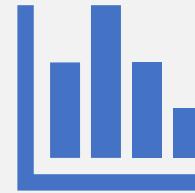


CSV

# What form is the data in



SOURCE DATA THAT IS IN  
SEPARATE TABLES



REPORT FORM WITH ALL  
THE DATA COMBINED

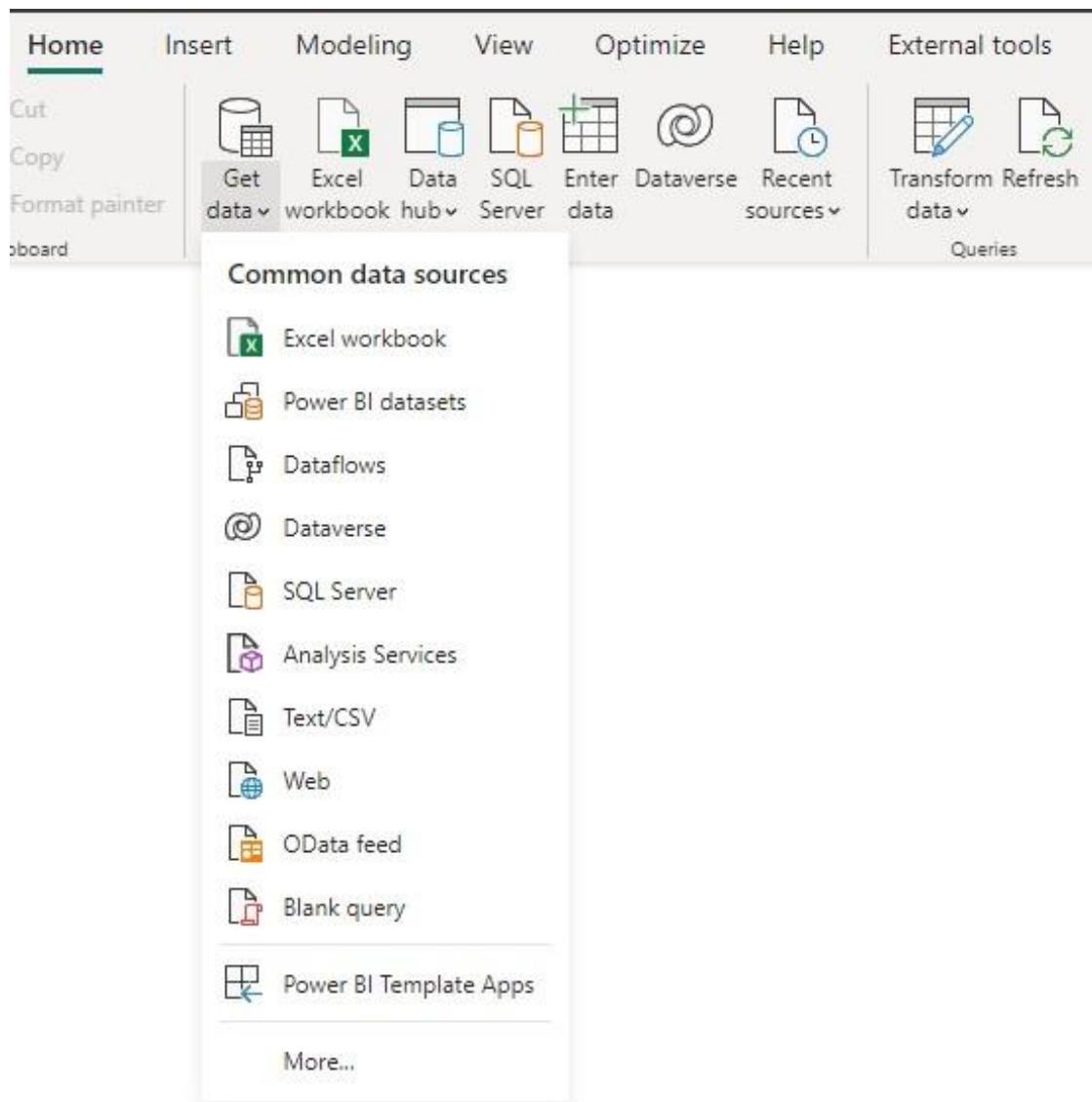
# Toy Model Builder

- Start - picture of finished product
- Proceed through instructions step by step
- Unlike toy model – changes can be made – no glue



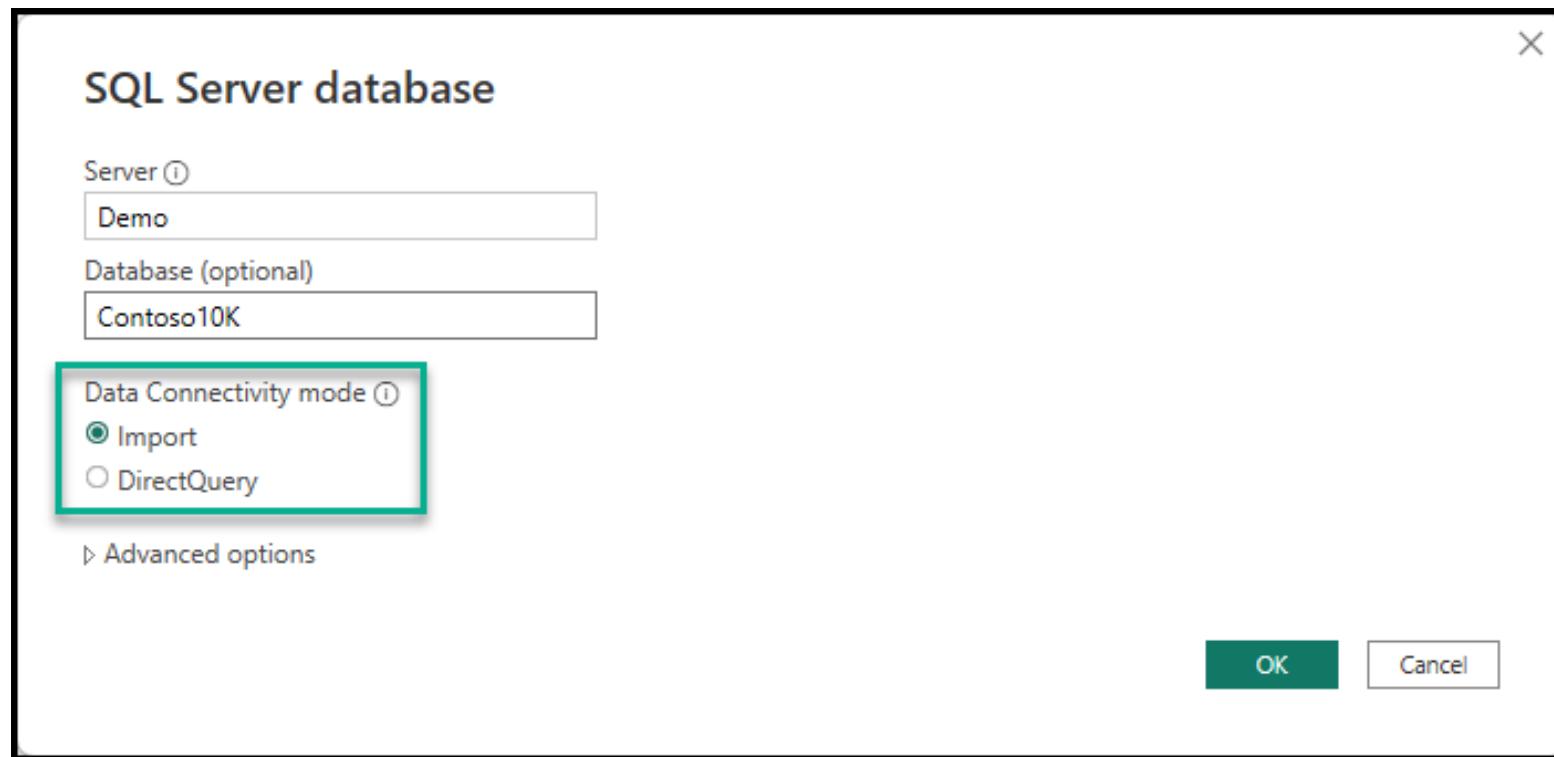
# Get Data

---



# Import or DirectQuery

---



# Select Tables or Views

Navigator

Display Options ▾

- demo: Contoso10K [20]
  - Currency Exchange
  - Customer
  - Date
  - DIM\_Customer
  - DIM\_Date
  - DIM\_Product
  - DIM\_Store
  - FACT\_Sales
  - Product
  - Sales
  - Store
  - Data.CurrencyExchange
  - Data.Customer
  - Data.Date
  - Data.GeoLocations
  - Data.OrderRows
  - Data.Orders
  - Data.Product
  - Data.Store
  - fn\_IsHolidav

Sales

Order Number	Line Number	Order Date	Delivery Date	CustomerKey	StoreKey	ProductKey
36600	0	1/1/2018	1/1/2018	1620763	540	
36700	0	1/2/2018	1/2/2018	1320351	480	
36700	1	1/2/2018	1/2/2018	1320351	480	
36700	2	1/2/2018	1/2/2018	1320351	480	
36700	3	1/2/2018	1/2/2018	1320351	480	
36700	4	1/2/2018	1/2/2018	1320351	480	
36701	0	1/2/2018	1/2/2018	2077250	585	
36800	0	1/3/2018	1/10/2018	1743422	999999	
36800	1	1/3/2018	1/10/2018	1743422	999999	
36800	2	1/3/2018	1/10/2018	1743422	999999	
36800	3	1/3/2018	1/10/2018	1743422	999999	
36801	0	1/3/2018	1/3/2018	1013948	390	
36801	1	1/3/2018	1/3/2018	1013948	390	
36801	2	1/3/2018	1/3/2018	1013948	390	
36801	3	1/3/2018	1/3/2018	1013948	390	
36900	0	1/4/2018	1/4/2018	544050	220	
36901	0	1/4/2018	1/14/2018	485447	999999	
36902	0	1/4/2018	1/17/2018	1075463	999999	
37000	0	1/5/2018	1/5/2018	1319917	550	
37000	1	1/5/2018	1/5/2018	1319917	550	
37000	2	1/5/2018	1/5/2018	1319917	550	
37000	3	1/5/2018	1/5/2018	1319917	550	
37100	0	1/6/2018	1/6/2018	1107123	380	
37100	1	1/6/2018	1/6/2018	1107123	380	

Select Related Tables Load Transform Data Cancel

# Load or Transform Data

**Navigator**

Display Options ▾

- demo: Contoso10K [20]
  - Currency Exchange
  - Customer
  - Date
  - DIM\_Customer
  - DIM\_Date
  - DIM\_Product
  - DIM\_Store
  - FACT\_Sales
  - Product
  - Sales
  - Store
  - Data.CurrencyExchange
  - Data.Customer
  - Data.Date
  - Data.GeoLocations
  - Data.OrderRows
  - Data.Orders
  - Data.Product
  - Data.Store
  - fn\_IsHoliday

Select Related Tables

**Sales**

Order Number	Line Number	Order Date	Delivery Date	CustomerKey	StoreKey	ProductKey
36600	0	1/1/2018	1/1/2018	1620763	540	
36700	0	1/2/2018	1/2/2018	1320351	480	
36700	1	1/2/2018	1/2/2018	1320351	480	
36700	2	1/2/2018	1/2/2018	1320351	480	
36700	3	1/2/2018	1/2/2018	1320351	480	
36700	4	1/2/2018	1/2/2018	1320351	480	
36701	0	1/2/2018	1/2/2018	2077250	585	
36800	0	1/3/2018	1/10/2018	1743422	999999	
36800	1	1/3/2018	1/10/2018	1743422	999999	
36800	2	1/3/2018	1/10/2018	1743422	999999	
36800	3	1/3/2018	1/10/2018	1743422	999999	
36801	0	1/3/2018	1/3/2018	1013948	390	
36801	1	1/3/2018	1/3/2018	1013948	390	
36801	2	1/3/2018	1/3/2018	1013948	390	
36801	3	1/3/2018	1/3/2018	1013948	390	
36900	0	1/4/2018	1/4/2018	544050	220	
36901	0	1/4/2018	1/14/2018	485447	999999	
36902	0	1/4/2018	1/17/2018	1075463	999999	
37000	0	1/5/2018	1/5/2018	1319917	550	
37000	1	1/5/2018	1/5/2018	1319917	550	
37000	2	1/5/2018	1/5/2018	1319917	550	
37000	3	1/5/2018	1/5/2018	1319917	550	
37100	0	1/6/2018	1/6/2018	1107123	380	
37100	1	1/6/2018	1/6/2018	1107123	380	

**Load** **Transform Data** **Cancel**

# Power Query Editor

The screenshot shows the Microsoft Power Query Editor interface. The title bar reads "Untitled - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected, displaying various icons for operations like Close & Apply, New Source, Refresh, Manage Parameters, Properties, Advanced Editor, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Sort, Split Column, Group By, and Replace Values. A dropdown menu for "Data Type" is set to "Whole Number". The "Transform" tab is also visible.

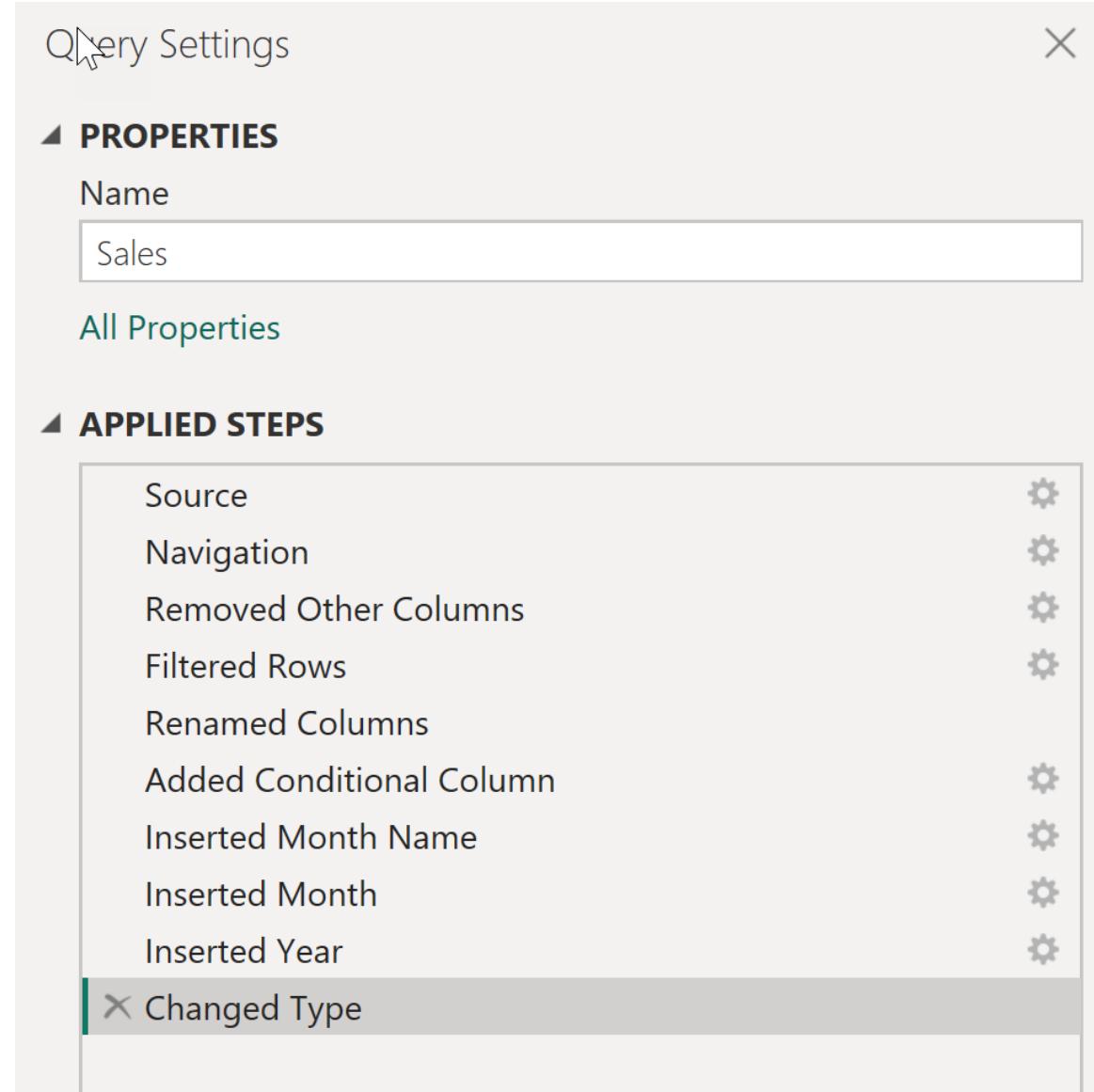
The left pane shows a list of queries: Store, Product, Sales (selected), and Customer. The main area displays a preview of the "Sales" data with columns: Order Number, Line Number, Order Date, Delivery Date, and CustomerKey. The data consists of 25 rows of sales records, starting with Order Number 36600 and ending with 37100. The preview indicates there are 13 columns and over 999 rows in total, with column profiling based on the top 1000 rows. The preview was downloaded at 2:04 PM.

The right pane contains "Query Settings" and two sections under "APPLIED STEPS": "Properties" (Name: Sales) and "Applied Steps" (Source: Navigation).

Order Number	Line Number	Order Date	Delivery Date	CustomerKey
36600	0	1/1/2018	1/1/2018	
36700	0	1/2/2018	1/2/2018	
36700	1	1/2/2018	1/2/2018	
36700	2	1/2/2018	1/2/2018	
36700	3	1/2/2018	1/2/2018	
36700	4	1/2/2018	1/2/2018	
36701	0	1/2/2018	1/2/2018	
36800	0	1/3/2018	1/10/2018	
36800	1	1/3/2018	1/10/2018	
36800	2	1/3/2018	1/10/2018	
36800	3	1/3/2018	1/10/2018	
36801	0	1/3/2018	1/3/2018	
36801	1	1/3/2018	1/3/2018	
36801	2	1/3/2018	1/3/2018	
36801	3	1/3/2018	1/3/2018	
36900	0	1/4/2018	1/4/2018	
36901	0	1/4/2018	1/14/2018	
36902	0	1/4/2018	1/17/2018	
37000	0	1/5/2018	1/5/2018	
37000	1	1/5/2018	1/5/2018	
37000	2	1/5/2018	1/5/2018	
37000	3	1/5/2018	1/5/2018	
37100	0	1/6/2018	1/6/2018	
37100	1	1/6/2018	1/6/2018	

# Power Query Steps like a Recipe

---



# Promote First Row to Headers (CSV)

The screenshot shows the Microsoft Power Query Editor interface. The ribbon at the top has 'File', 'Home', 'Transform' (which is selected), 'Add Column', 'View', 'Tools', and 'Help'. In the 'Transform' tab, there are several options: 'Group By', 'Use First Row as Headers' (which is currently selected), 'Queri...', 'Sales', 'Transpose', 'Reverse Rows', 'Count Rows', 'Data Type: Text', 'Replace Values', 'Unpivot Columns', 'Detect Data Type', 'Fill', 'Move', 'Rename', 'Pivot Column', and 'Convert to List'. A tooltip for the 'Use First Row as Headers' option is open, stating 'Promote the first row of this table into column headers.' Below the ribbon, a table preview shows two columns: 'Column1' and 'Column2'. The data in the table is as follows:

	Column1	Column2
1	Order Number	Line Number
2	36600	0
3	36700	0
4	36700	1
5	36700	2

# Promote First Row to Headers (CSV)

---

The screenshot shows the Microsoft Power Query Editor interface. The main area displays a table titled "Sales" with four columns: "Column1", "Column2", "Column3", and "Column4". The first row of the table is highlighted with a red border and contains the header names: "Order Number", "Line Number", "Order Date", and "Delivery Date". Below the table, the status bar indicates "13 COLUMNS, 999+ ROWS" and "Column profiling based on top 1000 rows". On the right side of the screen, there is a "Query Settings" pane. Under the "PROPERTIES" section, the "Name" is set to "Sales". Under the "APPLIED STEPS" section, there is a single step named "Source". The bottom right corner of the interface shows the timestamp "PREVIEW DOWNLOADED AT 1:27 PM".

# Promote First Row to Headers (CSV)

---

The screenshot shows the Microsoft Power Query Editor interface. The main area displays a table titled "Sales" with four columns: "Order Number", "Line Number", "Order Date", and "Delivery Date". The first row of the table is highlighted with a red border, indicating it has been promoted to a header. The formula bar at the top shows the formula: `= Table.PromoteHeaders(Source, [PromoteAllScalars=true])`. On the right side, the "Query Settings" pane is open, showing the "APPLIED STEPS" section which lists the "Promoted Headers" step. The status bar at the bottom indicates "13 COLUMNS, 999+ ROWS" and "Column profiling based on top 1000 rows".

# Remove Other Columns (Select Columns)

The screenshot shows the Microsoft Power Query Editor interface. The ribbon at the top has tabs for File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. On the far right of the ribbon, there is a 'Transform' section with several icons: Close & Apply, New Source, Recent Sources, Enter Data, Data source settings, Manage Parameters, Refresh Preview, Advanced Editor, Properties, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Split Column, Group By, Replace Values, and Data Type: Any. Below the ribbon is a 'Queries' list containing a single item named 'Sales'. The main area displays a preview of the 'Sales' data, which includes columns for Order Number, Line Number, Order Date, Delivery Date, CustomerKey, and others. A tooltip 'Remove Other Columns' is visible over the 'Remove Columns' icon in the ribbon. To the right of the preview, there is a 'Query Settings' pane with sections for 'PROPERTIES' (Name: Sales, All Properties) and 'APPLIED STEPS' (Source, Navigation). At the bottom of the preview area, it says '13 COLUMNS, 999+ ROWS' and 'Column profiling based on top 1000 rows'. The status bar at the bottom right says 'PREVIEW DOWNLOADED ON TUESDAY'.

# Filter Rows

The screenshot shows the Microsoft Power Query Editor interface. The main area displays a table of data with columns: Order Number, Line Number, Order Date, Delivery Date, and CustomerKey. A filter dialog box is open over the table, titled "Filter Rows". The dialog has "Basic" selected and "Keep rows where 'Order Date'" is chosen. Under "is after", the value "6/13/2018" is entered. Below it, "And" is selected, and an empty field "Enter or select a value" is shown. At the bottom right of the dialog are "OK" and "Cancel" buttons. The status bar at the bottom left indicates "13 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows". The status bar at the bottom right indicates "PREVIEW DOWNLOADED ON TUESDAY".

# **Set Data Types**

---

- Correct types matter
  - Same type for relationships
- Performance
- Calculations

# ProductKey as a String

---

Name	Cardinality	Total Size ↓	Data	Dictionary	Hier Size	Encoding	Data Type	RI Violations	User Hier Size	Rel Size	% Table
Product	2,517	15,142,472	27,520	15,043,768	71,184	Many	-	-	0	0	
Product Name	2,517	1,128,480	4,160	1,104,176	20,144	HASH	String	-	-	-	7.45%
Product Code	2,517	1,127,544	4,160	1,103,240	20,144	HASH	String	-	-	-	7.45%
ProductKey	2,517	1,127,416	4,160	1,103,112	20,144	HASH	String	-	-	-	7.45%
Unit Cost	480	1,076,408	3,008	1,069,544	3,856	HASH	String	-	-	-	7.11%
Unit Price	426	1,075,520	3,008	1,069,088	3,424	HASH	String	-	-	-	7.10%
Weight	297	1,073,176	2,736	1,068,056	2,384	HASH	String	-	-	-	7.09%
Subcategory	32	1,067,560	1,352	1,065,936	272	HASH	String	-	-	-	7.05%
Subcategory Code	32	1,067,560	1,352	1,065,936	272	HASH	String	-	-	-	7.05%
Color	16	1,067,144	1,192	1,065,808	144	HASH	String	-	-	-	7.05%
Brand	11	1,066,464	600	1,065,768	96	HASH	String	-	-	-	7.04%
Manufacturer	11	1,066,464	600	1,065,768	96	HASH	String	-	-	-	7.04%
Category	8	1,066,288	464	1,065,744	80	HASH	String	-	-	-	7.04%
Category Code	8	1,066,288	464	1,065,744	80	HASH	String	-	-	-	7.04%
Weight Unit Measure	4	1,065,896	136	1,065,712	48	HASH	String	-	-	-	7.04%

# ProductKey as a Whole Number

Name	Cardinality	Total Size ↓	Data	Dictionary	Hier Size	Encoding	Data Type	RI Violations	User Hier Size	Rel Size	% Table
Product	2,517	14,115,268	27,520	14,016,564	71,184	Many	-	-	0	0	
Product Name	2,517	1,128,480	4,160	1,104,176	20,144	HASH	String	-	-	-	7.99%
Product Code	2,517	1,127,544	4,160	1,103,240	20,144	HASH	String	-	-	-	7.99%
Unit Cost	480	1,076,408	3,008	1,069,544	3,856	HASH	String	-	-	-	7.63%
Unit Price	426	1,075,520	3,008	1,069,088	3,424	HASH	String	-	-	-	7.62%
Weight	297	1,073,176	2,736	1,068,056	2,384	HASH	String	-	-	-	7.60%
Subcategory	32	1,067,560	1,352	1,065,936	272	HASH	String	-	-	-	7.56%
Subcategory Code	32	1,067,560	1,352	1,065,936	272	HASH	String	-	-	-	7.56%
Color	16	1,067,144	1,192	1,065,808	144	HASH	String	-	-	-	7.56%
Brand	11	1,066,464	600	1,065,768	96	HASH	String	-	-	-	7.56%
Manufacturer	11	1,066,464	600	1,065,768	96	HASH	String	-	-	-	7.56%
Category	8	1,066,288	464	1,065,744	80	HASH	String	-	-	-	7.55%
Category Code	8	1,066,288	464	1,065,744	80	HASH	String	-	-	-	7.55%
Weight Unit Measure	4	1,065,896	136	1,065,712	48	HASH	String	-	-	-	7.55%
ProductKey	2,517	100,212	4,160	75,908	20,144	HASH	Int64	-	-	-	0.71%

# Cardinality

---

Name	Cardinality	Total Size ↓	Data	Dictionary	Hier Size	Encoding	Data Type	RI Violations	User Hier Size	Rel Size	% Table
↳ Product	2,517	15,142,472	27,520	15,043,768	71,184	Many	-	-	0	0	
Product Name	2,517	1,128,480	4,160	1,104,176	20,144	HASH	String	-	-	-	7.45%
Product Code	2,517	1,127,544	4,160	1,103,240	20,144	HASH	String	-	-	-	7.45%
ProductKey	2,517	1,127,416	4,160	1,103,112	20,144	HASH	String	-	-	-	7.45%
Unit Cost	480	1,076,408	3,008	1,069,544	3,856	HASH	String	-	-	-	7.11%
Unit Price	426	1,075,520	3,008	1,069,088	3,424	HASH	String	-	-	-	7.10%
Weight	297	1,073,176	2,736	1,068,056	2,384	HASH	String	-	-	-	7.09%
Subcategory	32	1,067,560	1,352	1,065,936	272	HASH	String	-	-	-	7.05%
Subcategory Code	32	1,067,560	1,352	1,065,936	272	HASH	String	-	-	-	7.05%
Color	16	1,067,144	1,192	1,065,808	144	HASH	String	-	-	-	7.05%
Brand	11	1,066,464	600	1,065,768	96	HASH	String	-	-	-	7.04%
Manufacturer	11	1,066,464	600	1,065,768	96	HASH	String	-	-	-	7.04%
Category	8	1,066,288	464	1,065,744	80	HASH	String	-	-	-	7.04%
Category Code	8	1,066,288	464	1,065,744	80	HASH	String	-	-	-	7.04%
Weight Unit Measure	4	1,065,896	136	1,065,712	48	HASH	String	-	-	-	7.04%

# Cardinality – Unique Values

---

- The less unique values the better
- Do you really need Date and Time
- Does the time need to include seconds
  - What about morning, afternoon, night



# Custom Columns

---

**Add Conditional Column**

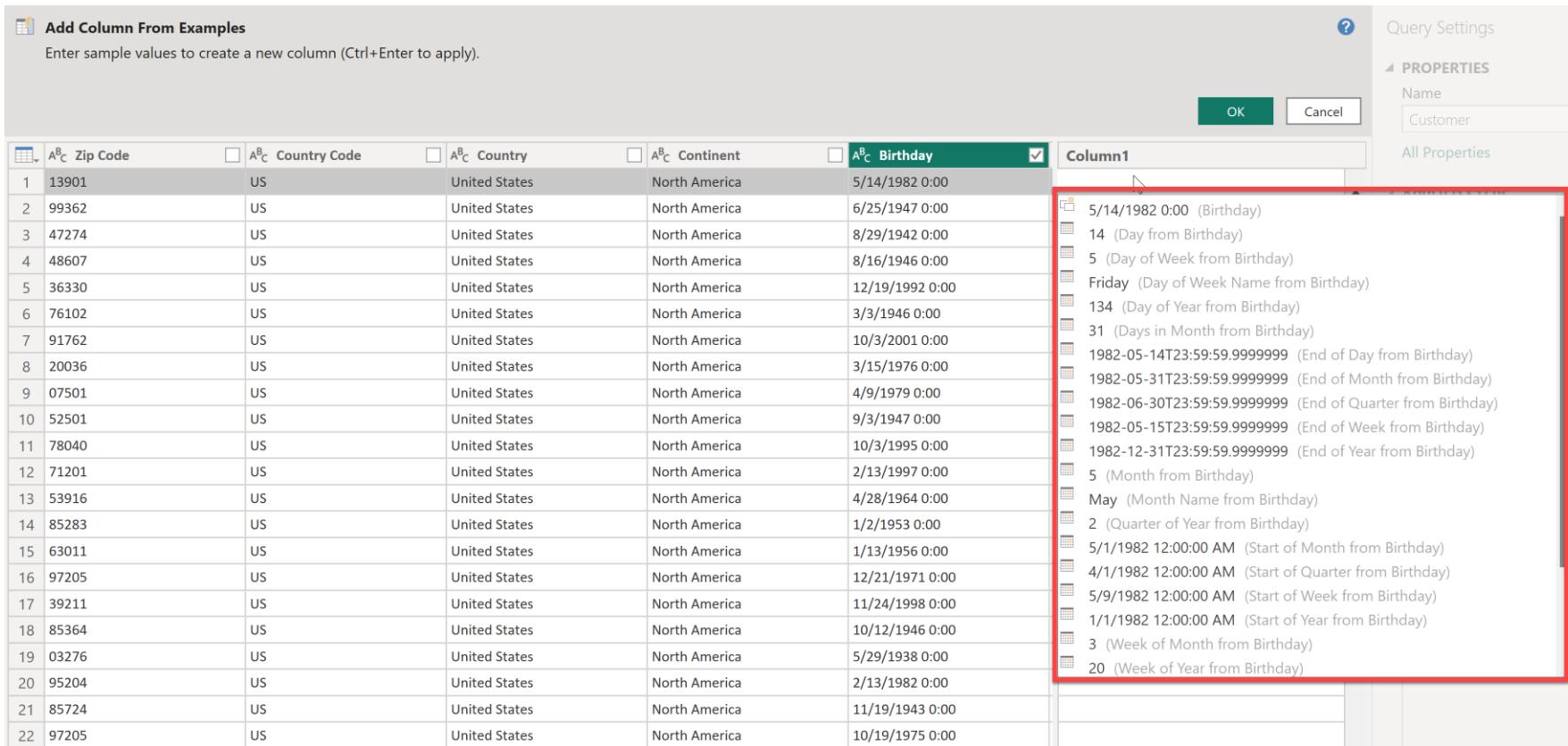
Add a conditional column that is computed from the other columns or values.

New column name

Column Name	Operator	Value ⓘ	Output ⓘ					
If	Square Meters	equals	ABC 123	null	Then	ABC 123	N/A	...
Else If	Square Meters	is less than or equ...	ABC 123	500	Then	ABC 123	Small	...
Else If	Square Meters	is less than or equ...	ABC 123	1000	Then	ABC 123	Medium	...

Else ⓘ  
 Large

# Add Column from Example

Add Column From Examples  
Enter sample values to create a new column (Ctrl+Enter to apply).

	A <sup>B</sup> <sub>C</sub> Zip Code	A <sup>B</sup> <sub>C</sub> Country Code	A <sup>B</sup> <sub>C</sub> Country	A <sup>B</sup> <sub>C</sub> Continent	A <sup>B</sup> <sub>C</sub> Birthday	<input checked="" type="checkbox"/>
1	13901	US	United States	North America	5/14/1982 0:00	
2	99362	US	United States	North America	6/25/1947 0:00	
3	47274	US	United States	North America	8/29/1942 0:00	
4	48607	US	United States	North America	8/16/1946 0:00	
5	36330	US	United States	North America	12/19/1992 0:00	
6	76102	US	United States	North America	3/3/1946 0:00	
7	91762	US	United States	North America	10/3/2001 0:00	
8	20036	US	United States	North America	3/15/1976 0:00	
9	07501	US	United States	North America	4/9/1979 0:00	
10	52501	US	United States	North America	9/3/1947 0:00	
11	78040	US	United States	North America	10/3/1995 0:00	
12	71201	US	United States	North America	2/13/1997 0:00	
13	53916	US	United States	North America	4/28/1964 0:00	
14	85283	US	United States	North America	1/2/1953 0:00	
15	63011	US	United States	North America	1/13/1956 0:00	
16	97205	US	United States	North America	12/21/1971 0:00	
17	39211	US	United States	North America	11/24/1998 0:00	
18	85364	US	United States	North America	10/12/1946 0:00	
19	03276	US	United States	North America	5/29/1938 0:00	
20	95204	US	United States	North America	2/13/1982 0:00	
21	85724	US	United States	North America	11/19/1943 0:00	
22	97205	US	United States	North America	10/19/1975 0:00	

Query Settings  
Properties  
Name: Customer  
All Properties

Column1

- 5/14/1982 0:00 (Birthday)
- 14 (Day from Birthday)
- 5 (Day of Week from Birthday)
- Friday (Day of Week Name from Birthday)
- 134 (Day of Year from Birthday)
- 31 (Days in Month from Birthday)
- 1982-05-14T23:59:59.9999999 (End of Day from Birthday)
- 1982-05-31T23:59:59.9999999 (End of Month from Birthday)
- 1982-06-30T23:59:59.9999999 (End of Quarter from Birthday)
- 1982-05-15T23:59:59.9999999 (End of Week from Birthday)
- 1982-12-31T23:59:59.9999999 (End of Year from Birthday)
- 5 (Month from Birthday)
- May (Month Name from Birthday)
- 2 (Quarter of Year from Birthday)
- 5/1/1982 12:00:00 AM (Start of Month from Birthday)
- 4/1/1982 12:00:00 AM (Start of Quarter from Birthday)
- 5/9/1982 12:00:00 AM (Start of Week from Birthday)
- 1/1/1982 12:00:00 AM (Start of Year from Birthday)
- 3 (Week of Month from Birthday)
- 20 (Week of Year from Birthday)

# Add Column from Example

---

**Add Column From Examples**

Enter sample values to create a new column (Ctrl+Enter to apply).  
Transform: Date.Year(DateTime.From({Birthday}))

OK Cancel

	A <sup>B</sup> <sub>C</sub> Zip Code	A <sup>B</sup> <sub>C</sub> Country Code	A <sup>B</sup> <sub>C</sub> Country	A <sup>B</sup> <sub>C</sub> Continent	A <sup>B</sup> <sub>C</sub> Birthday	Year
1	13901	US	United States	North America	5/14/1982 0:00	1982
2	99362	US	United States	North America	6/25/1947 0:00	1947
3	47274	US	United States	North America	8/29/1942 0:00	1942
4	48607	US	United States	North America	8/16/1946 0:00	1946
5	36330	US	United States	North America	12/19/1992 0:00	1992
6	76102	US	United States	North America	3/3/1946 0:00	1946
7	91762	US	United States	North America	10/3/2001 0:00	2001
8	20036	US	United States	North America	3/15/1976 0:00	1976
9	07501	US	United States	North America	4/9/1979 0:00	1979
10	52501	US	United States	North America	9/3/1947 0:00	1947
11	78040	US	United States	North America	10/3/1995 0:00	1995
12	71201	US	United States	North America	2/13/1997 0:00	1997
13	53916	US	United States	North America	4/28/1964 0:00	1964
14	85283	US	United States	North America	1/2/1953 0:00	1953
15	63011	US	United States	North America	1/13/1956 0:00	1956
16	97205	US	United States	North America	12/21/1971 0:00	1971
17	39211	US	United States	North America	11/24/1998 0:00	1998
18	85364	US	United States	North America	10/12/1946 0:00	1946
19	03276	US	United States	North America	5/29/1938 0:00	1938
20	95204	US	United States	North America	2/13/1982 0:00	1982
21	85724	US	United States	North America	11/19/1943 0:00	1943
22	97205	US	United States	North America	10/19/1975 0:00	1975
23	48066	US	United States	North America	8/13/1995 0:00	1995
24	38049	US	United States	North America	11/13/1939 0:00	1939
25	13202	US	United States	North America	11/24/1961 0:00	1961
26	49546	US	United States	North America	8/24/1940 0:00	1940
27	60014	US	United States	North America	6/17/1985 0:00	1985
28	32501	US	United States	North America	4/16/1951 0:00	1951
29	00234	US	United States	North America	8/23/1989 0:00	1989



# Roche's Maxim

***Data should be transformed as far upstream as possible, and as far downstream as necessary.***

--***Matthew Roche***

**<https://ssbipolar.com/2021/05/31/roches-maxim/>**

# Query Folding

The screenshot shows the Microsoft Power Query Editor interface. The main area displays a table titled "Sales" with columns: Order Number, Line Number, Order Date, Delivery Date, and CustomerKey. The table contains 25 rows of data. Above the table, a formula bar shows: `= Table.SelectRows(#"Removed Other Columns", each [Order Date] > #date(2018, 6, 13))`. The ribbon menu is visible at the top, and the "Transform" tab is selected. On the right side, there is a "Query Settings" pane and a "PROPERTIES" pane where the name is set to "Sales". A context menu is open over the "Removed Other Columns" step in the "APPLIED STEPS" list, with the "View Native Query" option highlighted by a red box.

Untitled - Power Query Editor

File Home Transform Add Column View Tools Help

Close & Apply Close New Source Sources Data source settings Manage Parameters Refresh Preview Advanced Editor Properties Choose Columns Remove Columns Keep Rows Remove Rows Reduce Rows Sort Split Column Group By Use First Row as Headers Data Type: Any Transform

Queries []

Sales

Order Number Line Number Order Date Delivery Date CustomerKey

1	53000	0	6/14/2018	6/14/2018
2	53000	1	6/14/2018	6/14/2018
3	53200	0	6/16/2018	6/16/2018
4	53200	1	6/16/2018	6/16/2018
5	53200	2	6/16/2018	6/16/2018
6	53200	3	6/16/2018	6/16/2018
7	53600	0	6/20/2018	7/2/2018
8	53700	0	6/21/2018	6/21/2018
9	53700	1	6/21/2018	6/21/2018
10	53700	2	6/21/2018	6/21/2018
11	53900	0	6/23/2018	6/23/2018
12	53900	1	6/23/2018	6/23/2018
13	53900	2	6/23/2018	6/23/2018
14	54300	0	6/27/2018	6/27/2018
15	54400	0	6/28/2018	6/28/2018
16	54400	1	6/28/2018	6/28/2018
17	54600	0	6/30/2018	6/30/2018
18	54600	1	6/30/2018	6/30/2018
19	55000	0	7/4/2018	7/4/2018
20	55000	1	7/4/2018	7/4/2018
21	55000	2	7/4/2018	7/4/2018
22	55000	3	7/4/2018	7/4/2018
23	55000	4	7/4/2018	7/4/2018
24	55000	5	7/4/2018	7/4/2018
25	55000	6	7/4/2018	7/4/2018

5 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED AT 2:06 PM

Query Settings

PROPERTIES

APPLIED STEPS

Source Navigation Removed Other Columns

View Native Query

# Native Query

---

## Native Query

```
select [__].[Order Number],  
    [__].[Line Number],  
    [__].[Order Date],  
    [__].[Delivery Date],  
    [__].[CustomerKey]  
from  
(  
    select [Order Number],  
        [Line Number],  
        [Order Date],  
        [Delivery Date],  
        [CustomerKey]  
    from [dbo].[Sales] as [$Table]  
) as [__]  
where [__].[Order Date] > convert(date, '2018-06-13')
```

OK

# Breaking (Bad) Query Folding

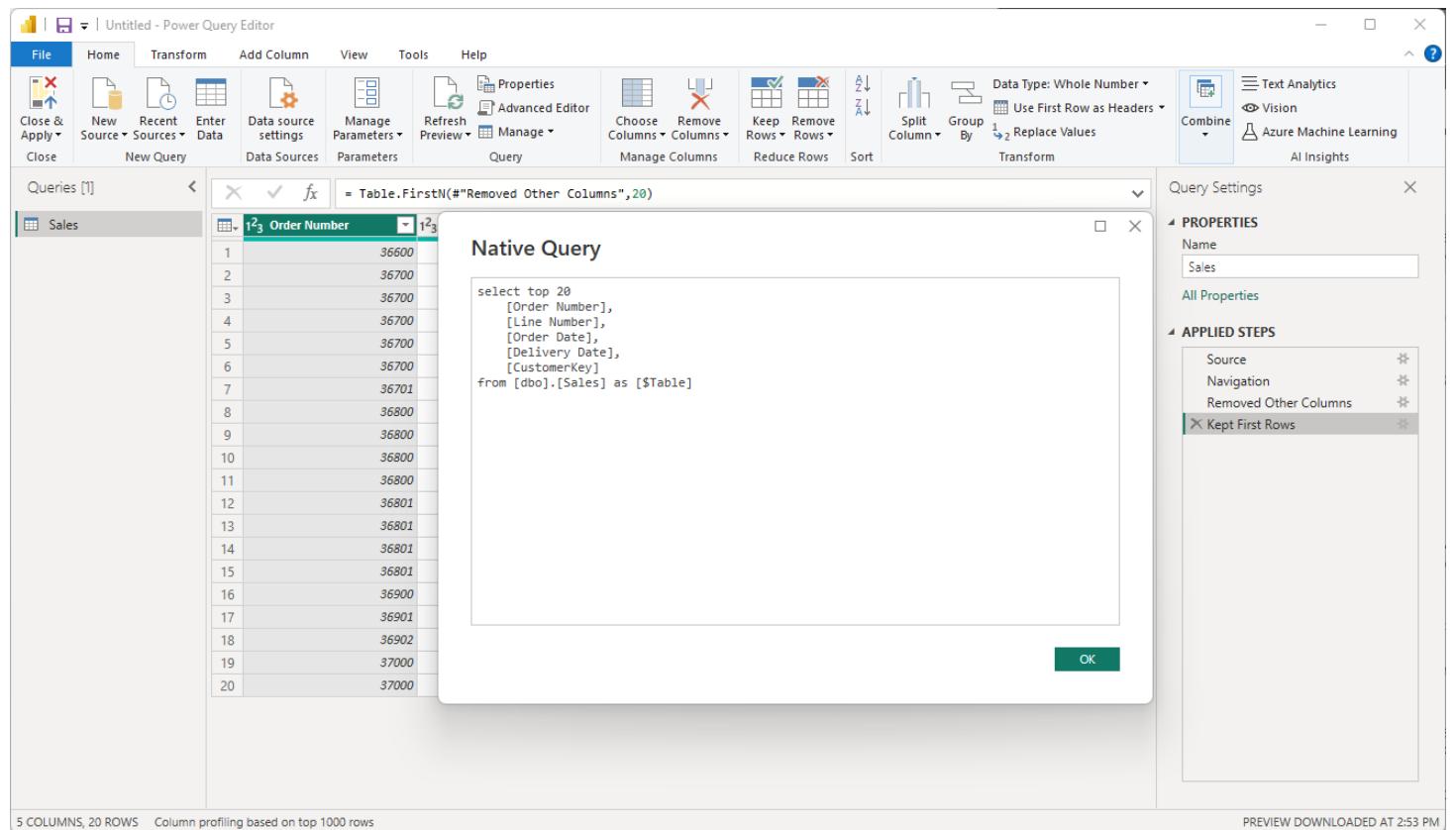
- Keep bottom 20 rows  
breaks query folding
- SQL Server does not  
have a bottom clause

The screenshot shows the Power Query Editor interface. The 'Queries [1]' list contains a single item named 'Sales'. The preview area displays a table with columns: Order Number, Line Number, Order Date, Delivery Date, and CustomerKey. The formula bar at the top has the expression: = Table.LastN(#"Removed Other Columns", 20). The 'APPLIED STEPS' pane on the right lists a step named 'Removed Other Columns'. A context menu is open over this step, with the 'View Native Query' option highlighted.

Order Number	Line Number	Order Date	Delivery Date	CustomerKey
115302	1	2/27/2020	2/27/2020	924770
115400	0	2/28/2020	2/28/2020	353571
115400	1	2/28/2020	2/28/2020	353571
115401	0	2/28/2020	3/8/2020	1358107
115500	0	2/29/2020	2/29/2020	345884
115501	0	2/29/2020	3/2/2020	1485625
115501	1	2/29/2020	3/2/2020	1485625
115501	2	2/29/2020	3/2/2020	1485625
115501	3	2/29/2020	3/2/2020	1485625
115501	4	2/29/2020	3/2/2020	1485625
115501	5	2/29/2020	3/2/2020	1485625
115501	6	2/29/2020	3/2/2020	1485625
115502	0	2/29/2020	2/29/2020	982512
115502	1	2/29/2020	2/29/2020	982512
115502	2	2/29/2020	2/29/2020	982512
115502	3	2/29/2020	2/29/2020	982512
115502	4	2/29/2020	2/29/2020	982512
115502	5	2/29/2020	2/29/2020	982512
115502	6	2/29/2020	2/29/2020	982512
115800	0	3/3/2020	3/3/2020	237000

# Query Folding

- Keep Top 20 Rows does not break query folding
- SQL Server has a top N clause



# Snowflake Data Source

---

- View Native Query is grey for Snowflake data source
- Even online experience is not very reliable
- All else fails – check query being sent at source



# Query Folding

---

- This is the query sent when the model is refreshed
- This is not the query sent to the tabular engine when you interact with the visuals
- Put off breaking the query folding till the later steps

# Restore the fold

---

```
let  
    myQuery="SELECT * FROM Sales",  
    Source = Sql.Database("demo",  
"Contoso10K"),  
    Folding = Value.NativeQuery(Source,  
myQuery, null, [EnableFolding=true]),  
    #"Removed Other Columns" =  
Table.SelectColumns(Folding, {"Order  
Number", "Line Number", "Order Date",  
"Delivery Date"})  
in  
    #"Removed Other Columns"
```



# Power Query in Power BI Desktop

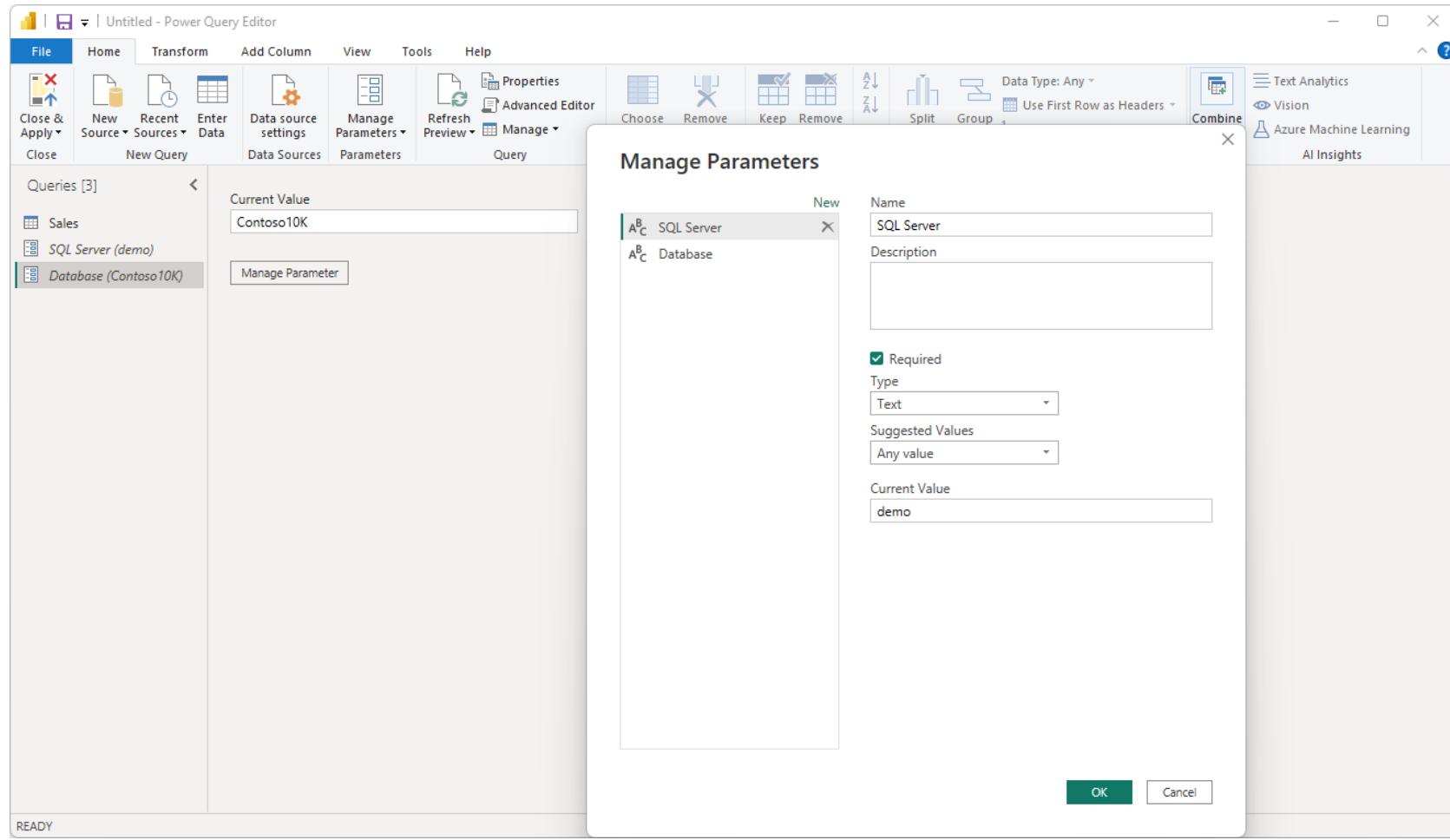
---

- New Experience with Icons like Power Query Online
- Projected to be released this year

# Parameters

---

## Creating parameters



# Parameter: One Use

Using a parameter  
for the Server and  
Database

The screenshot shows the Microsoft Power Query Editor interface. In the center, there's a preview grid for a query named "Sales" which connects to "SQL Server (demo)" and the "Database (Contoso10K)". The preview shows 20 rows of data from various tables like Data.CurrencyExchange, Data.Customer, etc. On the right side, there's a "Query Settings" pane open. A modal dialog box is displayed over the main area, titled "SQL Server database". It contains fields for "Server" (set to "SQL Server") and "Database (optional)" (set to "Database"). Below these are "Advanced options" and "OK" and "Cancel" buttons. The "APPLIED STEPS" pane on the far right lists actions taken on the source data.

Untitled - Power Query Editor

File Home Transform Add Column View Tools Help

New Source Sources Enter Data Data source settings Manage Parameters Refresh Advanced Editor Properties Preview Manage Choose Columns Remove Columns Keep Rows Remove Rows Sort Split Column Group By Replace Values Data Type: Text

Queries [3] Sales = Sql.Database("demo", Database)

SQL Server (demo) Database (Contoso10K)

ABC Name ABC 123 Data ABC Schema ABC Item ABC Kind

	ABC Name	ABC 123 Data	ABC Schema	ABC Item	ABC Kind
1	Data.CurrencyExchange	Table	Data	CurrencyExchange	Table
2	Data.Customer	Table	Data	Customer	Table
3	Data.Date	Table	Data	Date	Table
4	Data.GeoLocations	Table	Data	GeoLocations	Table
5	Data.OrderRows	Table	Data	OrderRows	Table
6	Data.Orders				
7	Data.Product				
8	Data.Store				
9	Currency Exchange				
10	Customer				
11	Date				
12	DIM_Customer				
13	DIM_Date				
14	DIM_Product				
15	DIM_Store				
16	FACT_Sales				
17	fn_IsHoliday				
18	Product	Table	dbo	Product	View
19	Sales	Table	dbo	Sales	View
20	Store	Table	dbo	Store	View

5 COLUMNS, 20 ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED AT 4:23 PM

Query Settings

Properties

Applied Steps

Source

Kept Last Rows

Removed Other Columns

Changed Type

Kept First Rows

# Parameters

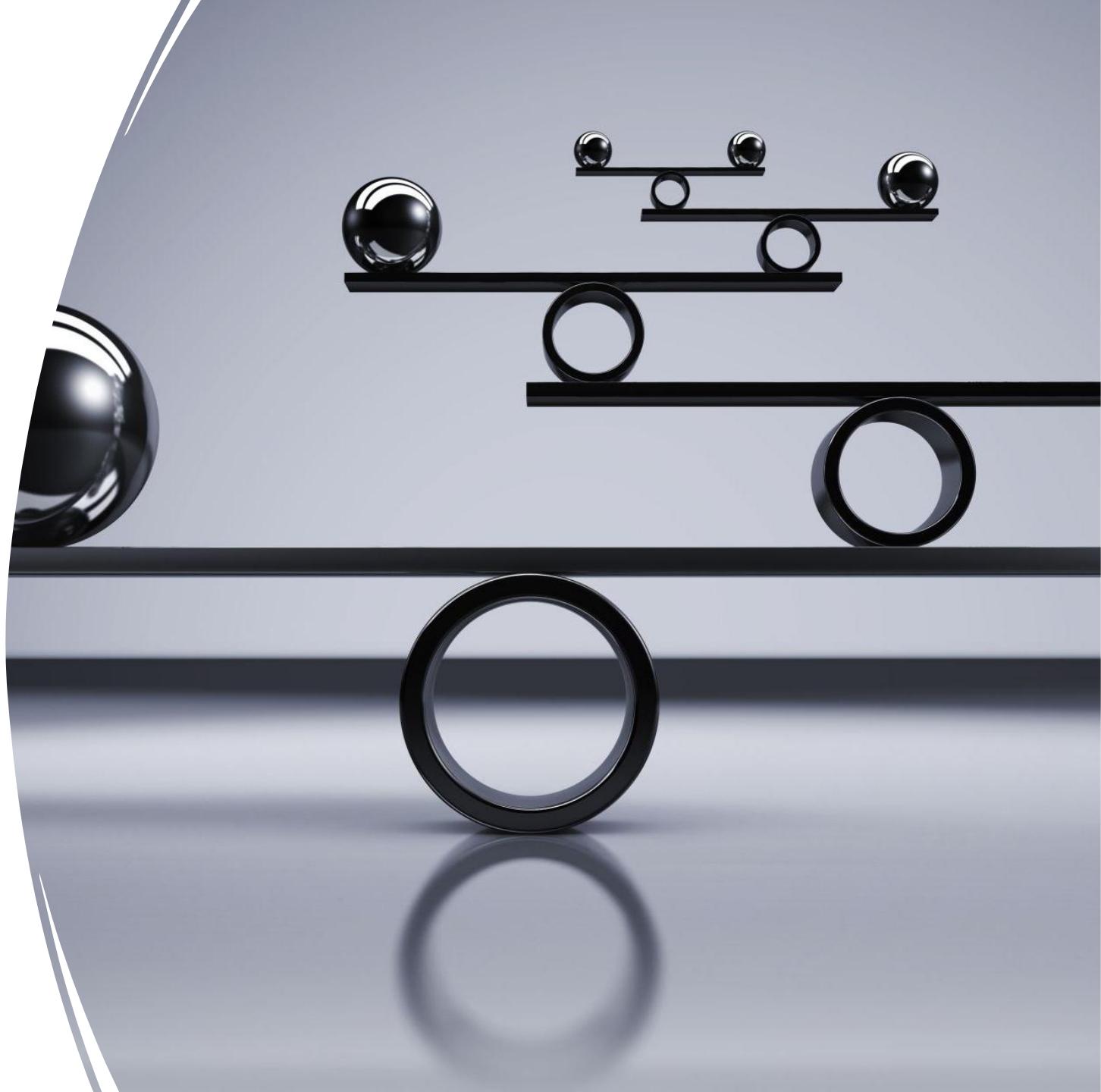
---

- Incremental Refresh
- Small dataset for developing on Power BI Desktop
  - Fully hydrated model in Service

# Goals of Power Query Steps

---

- Get smallest dataset possible early in query steps
  - Remove Columns
  - Filter Rows
- Unique Values
- Preserve Query Folding as long as possible



# Our Plan

---



1. Intro
2. Power Query
- 3. Data Model**
4. External Tools
5. Conclusion

# 2 Goals

---



# 1. Creating the Power BI Developer UI

---



# Model Name

---



# Model Names

SnowflakeTest\_Silver\_\_DirectQuery

PayrollSystem\_with\_HR\_excel

Thing\_to\_Export\_To\_Excel

One\_Model\_To\_Rule\_Them\_All

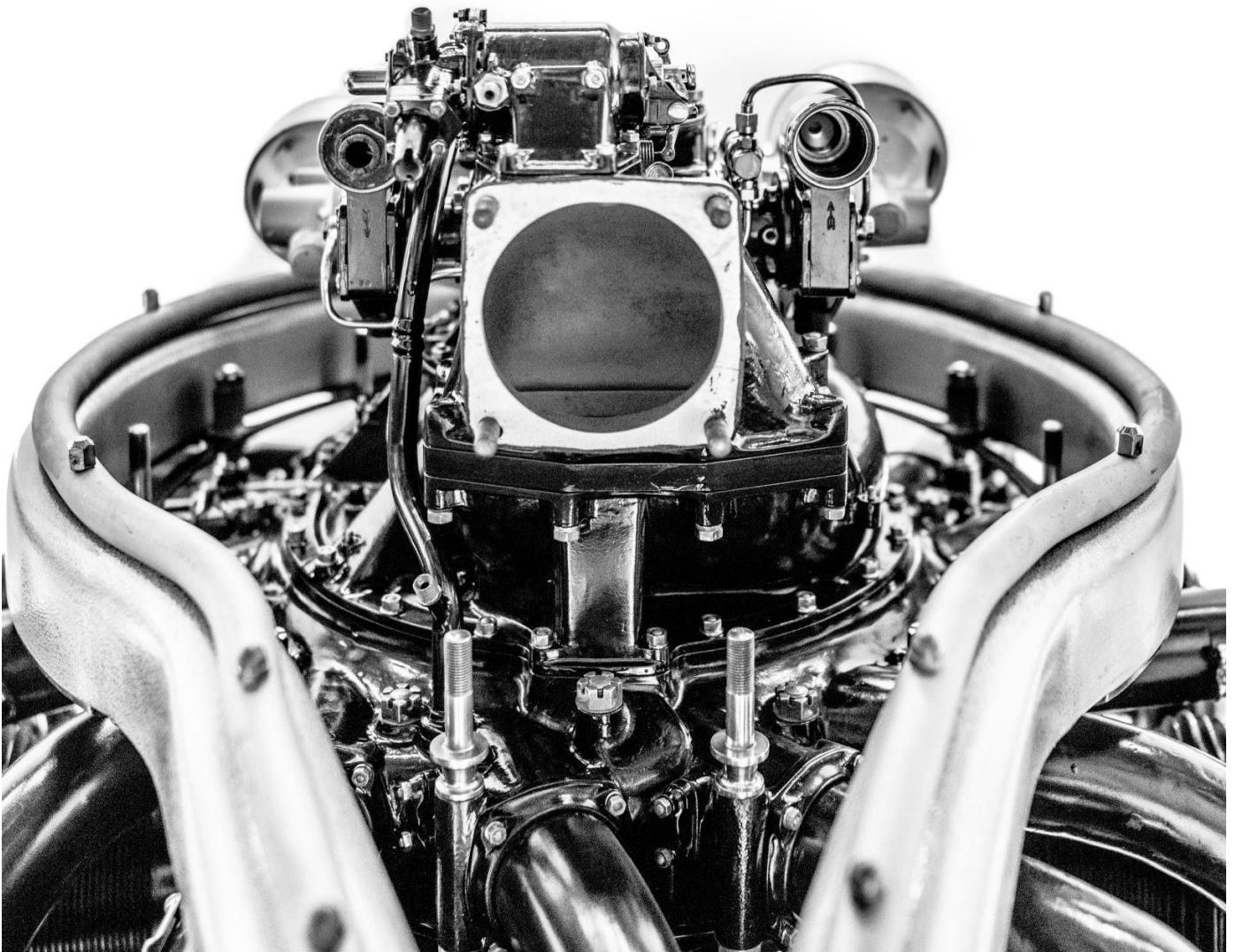
```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True
```

```
selection at the end -add
    ob.select= 1
    ob.select=1
    context.scene.objects.active
    ("Selected" + str(modifier))
    mirror_mod.select = 0
    bpy.context.selected_objects
    data.objects[one.name].sele
    int("please select exactly one ob")
    - OPERATOR CLASSES -----
```

```
types.Operator):
    # X mirror to the selected
    # object.mirror_mirror_x"
    "mirror X"
context):
    context.active_object is not
```

## 2. Building a Model that Performs and Scales

---

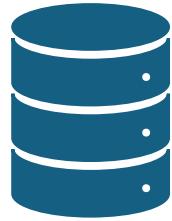


# Power BI: Under the Hood

---



# 2 Engines



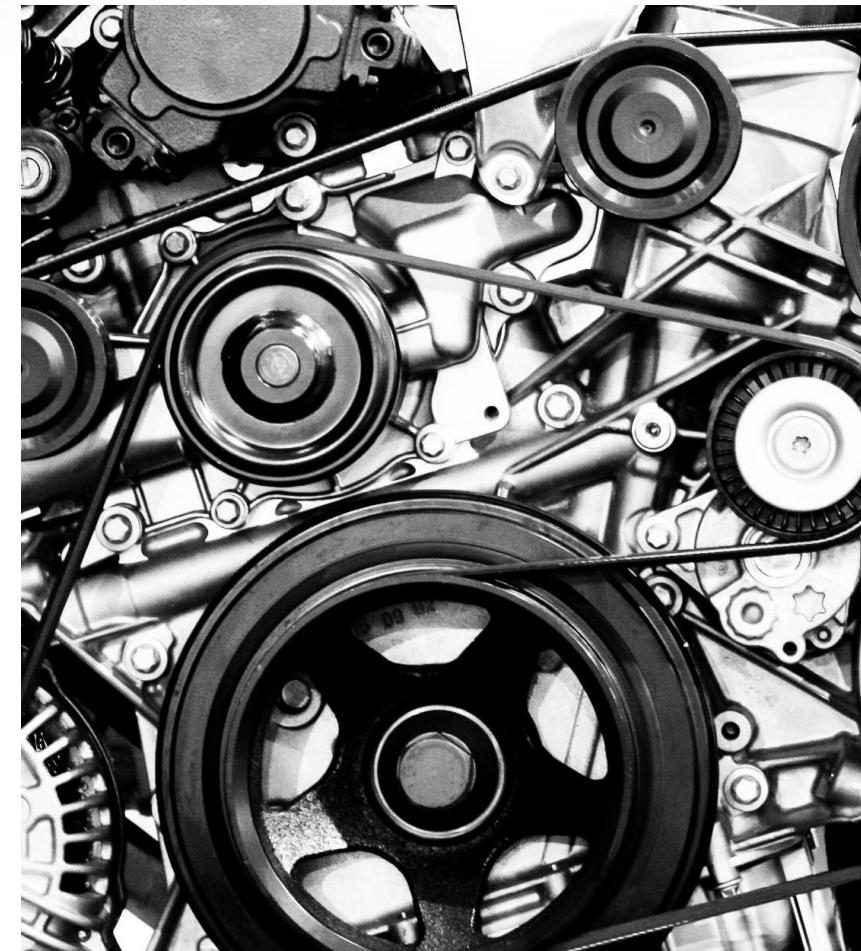
## Storage Engine

Gets the data



## Formula Engine

Takes data from the storage engine and performs calculations



# **Vertipaq (Import)**

Bottom line - data is optimized for quick retrieval

Encoded – Value and Hash

Run Length Encoding – Repeat values good

Goal is smallest amount of memory

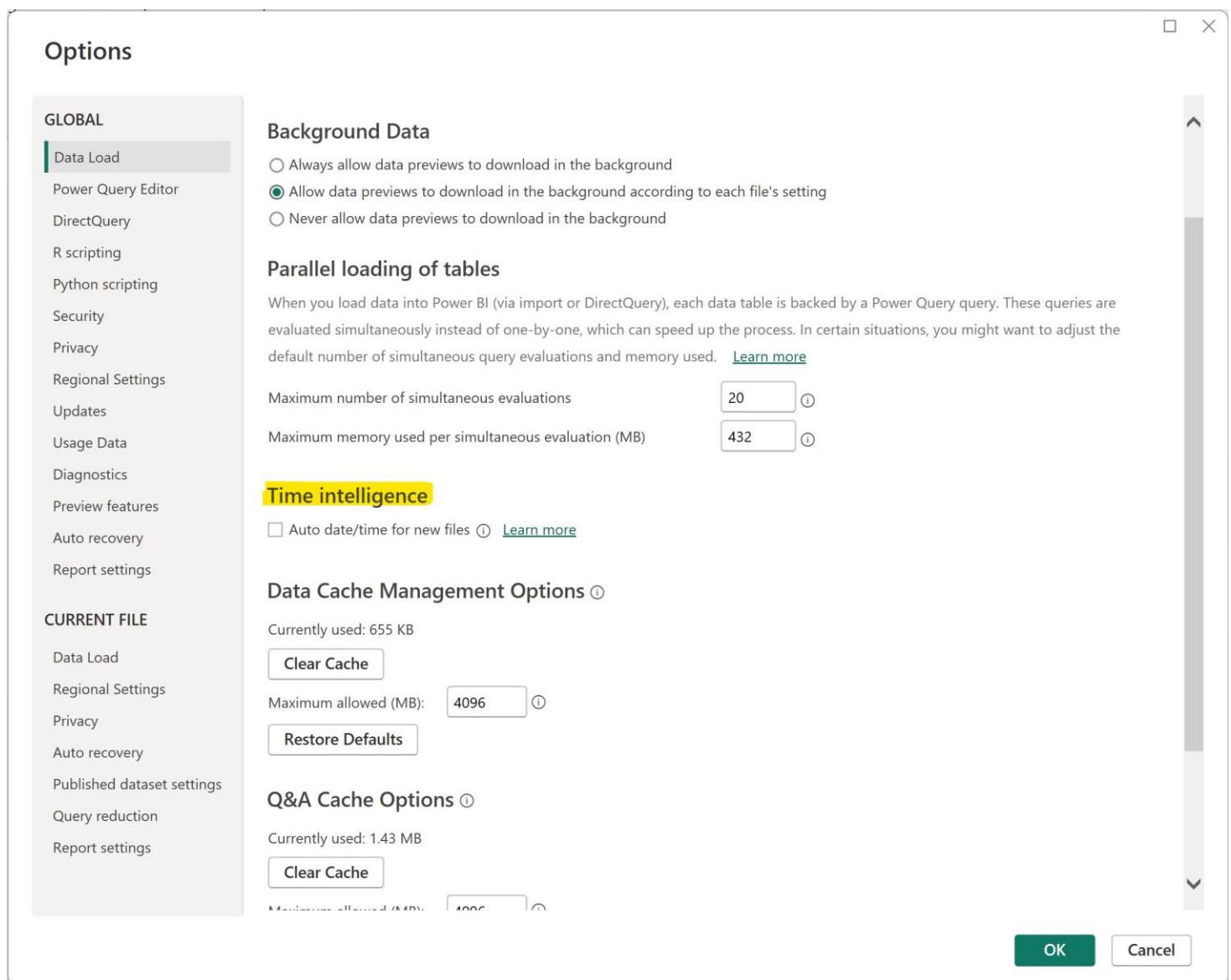
# Date Tables

---



# Turn off Auto Date Tables

---



# Hidden Date Tables

---

- >  Customer
- >  Date
- ✓  DateTableTemplate\_a987aa9-4324-4fff-9e6b-91eb917a6145
  - >  Date
  - >  Date Hierarchy
    - 1<sup>2</sup>3 Day
    - A<sup>B</sup>C Month
    - 1<sup>2</sup>3 MonthNo
    - A<sup>B</sup>C Quarter
    - 1<sup>2</sup>3 QuarterNo
    - 1<sup>2</sup>3 Year
- >  LocalDateTable\_329979ec-f948-448d-8d40-da32309c689b
- >  LocalDateTable\_565d20a7-37da-41eb-acea-3d1681afac62
- >  LocalDateTable\_6082b466-aa3a-416d-92f5-dee233105c41
- >  LocalDateTable\_729c8b24-1385-4545-897f-e90fb9602a1d
- >  LocalDateTable\_86220e95-4654-4888-819a-8af2ddbe1da
- >  LocalDateTable\_8ce283be-c847-412c-907b-87781a0d19e1
- >  Product
- >  Sales
- >  Store

# Date Table Alternatives

---

SUN	MON	TUE	WED	THU	FRI	SAT
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

30	31	32	33	34	35	36
10	11	12	13	14	15	16
17	18	19	20	21	22	23

# Power Query (M): Date Table

---



# Kristyna Ferris

---

- Date Table Built in Power Query
- [https://github.com/Anytsirk12/  
DataOnWheels/tree/main/SQL  
Bits%202024](https://github.com/Anytsirk12/DataOnWheels/tree/main/SQL%202024)



# M Code

```
----- M Pre-flight Checklist -----
//Must start with a "let" statement and end with an "in" statement
//Every step must end with a comma except the last step
//When using Table. functions, be sure to reference the previous step name as the table.
//Not whitespace sensitive, but it is case sensitive.
//Always name your steps, makes troubleshooting and updating a million times easier!
//To avoid extra steps, assign data types when creating a column.
//Follow the DRY philosophy - Don't Repeat Yourself. Use variables and already created columns to avoid having to redo logic
//If you want to remove columns, do so within power query since it will automatically fix the references in future steps

let
    //Use variables in M to set the start date
    StartDate = #date(2021, 12, 1),
    //EndDate is dynamically the start of next year. You can hard code this like StartDate, but that's not recommended.
    EndDate = #date(Date.Year(DateTime.LocalNow())+1,1,1),
    Length = Duration.Days(EndDate - StartDate),
    Today = DateTime.LocalNow(),

    //Create list of dates using our variables. #duration increments by (days,hours,minutes,seconds)
    Source = List.Dates(StartDate, Length, #duration(1, 0, 0, 0)),

    //Convert this list to a table in order to add additional columns
    #"Converted to Table" = Table.FromList(Source, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
    #"Renamed Date Column" = Table.RenameColumns(#"Converted to Table",{{"Column1", "Date"}}),
    #"Changed Type to Date" = Table.TransformColumnTypes(#"Renamed Date Column",{{"Date", type date}}),

    //Create a datekey for easy joins to fact tables
    #"Inserted DateKey" = Table.AddColumn(#"Changed Type to Date", "DateKey", each DateTime.Date([Date]), type date),
    #"Changed DateKey to Int" = Table.TransformColumnTypes(#"Inserted DateKey",{{"DateKey", Int64.Type}}),

    //Create columns for year & month
    #"Inserted Year" = Table.AddColumn(#"Changed DateKey to Int", "Year", each Date.Year([Date]), Int64.Type),
    #"Inserted Month" = Table.AddColumn(#"Inserted Year", "Month", each Date.Month([Date]), Int64.Type),
    #"Inserted Month Name" = Table.AddColumn(#"Inserted Month", "Month Name", each Date.MonthName([Date]), type text),
    //MMM is very useful for visuals since it eliminates a lot of characters and saves space
    #"Inserted MMM" = Table.AddColumn(#"Inserted Month Name", "MMM", each Text.Start([Month Name], 3), type text),
    #"Added Month Year" = Table.AddColumn(#"Inserted MMM", "Month Year", each Text.Combine({[MMM], Text.From([Year], "en-US")}), " "), type text),

    //Create columns for day & day of week
    #"Inserted Day" = Table.AddColumn(#"Added Month Year", "Day", each Date.Day([Date]), Int64.Type),
    #"Inserted Day Name" = Table.AddColumn(#"Inserted Day", "Day Name", each Date.DayOfWeekName([Date]), type text),
    //Sets Sunday as 1
    #"Inserted Day of Week" = Table.AddColumn(#"Inserted Day Name", "Day of Week", each Date.DayOfWeek([Date]), Int64.Type),
    #"Inserted DDD" = Table.AddColumn(#"Inserted Day of Week", "DDD", each Text.Start([Day Name], 3), type text),

    //Create MonthDay to use in visuals
    #"Inserted MonthDay" = Table.AddColumn(#"Inserted DDD", "Month Day", each Text.Combine({[MMM],
        if Text.Length(Text.From([Day])) = 1 then Text.Combine({"0",Text.From([Day])}, "") else Text.From([Day]), " ", type text}),
    //Need MonthDaySort to avoid visuals sorting alphabetically
    #"Inserted MonthDaySort" = Table.AddColumn(#"Inserted MonthDay", "Month Day Sort", each Text.Combine({
        if Text.Length(Text.From([Month]))=1 then Text.Combine({"0",Text.From([Month])}, "") else Text.From([Month]),
        if Text.Length(Text.From([Day])) = 1 then Text.Combine({"0",Text.From([Day])}, "") else Text.From([Day])
    }, "", type text)),
```

# Date Table View

CalendarDate	DateKey	Year	Month	Month Name	MMM	Month Year	Day	Day Name	Day	Week Number	IsYTD	IsMTD	IsWTD	IsCurrentYTD	IsCurrentMTD	IsCurrentWTD
Friday, December 3, 2021	44533	2021	12	December	Dec	Dec 2021	3	Friday		49	0	1	0	0	0	0
Friday, December 10, 2021	44540	2021	12	December	Dec	Dec 2021	10	Friday		50	0	1	0	0	0	0
Friday, December 17, 2021	44547	2021	12	December	Dec	Dec 2021	17	Friday		51	0	1	0	0	0	0
Friday, December 24, 2021	44554	2021	12	December	Dec	Dec 2021	24	Friday		52	0	0	0	0	0	0
Friday, December 31, 2021	44561	2021	12	December	Dec	Dec 2021	31	Friday		1	0	0	0	0	0	0
Friday, April 22, 2022	44673	2022	4	April	Apr	Apr 2022	22	Friday		17	0	0	0	0	0	0
Friday, April 29, 2022	44680	2022	4	April	Apr	Apr 2022	29	Friday		18	0	0	0	0	0	0
Friday, May 6, 2022	44687	2022	5	May	May	May 2022	6	Friday		19	0	1	0	0	0	0
Friday, May 13, 2022	44694	2022	5	May	May	May 2022	13	Friday		20	0	1	0	0	0	0
Friday, May 20, 2022	44701	2022	5	May	May	May 2022	20	Friday		21	0	0	0	0	0	0
Friday, May 27, 2022	44708	2022	5	May	May	May 2022	27	Friday		22	0	0	0	0	0	0
Friday, June 3, 2022	44715	2022	6	June	Jun	Jun 2022	3	Friday		23	0	1	0	0	0	0
Friday, June 10, 2022	44722	2022	6	June	Jun	Jun 2022	10	Friday		24	0	1	0	0	0	0
Friday, June 17, 2022	44729	2022	6	June	Jun	Jun 2022	17	Friday		25	0	1	0	0	0	0
Friday, June 24, 2022	44736	2022	6	June	Jun	Jun 2022	24	Friday		26	0	0	0	0	0	0
Friday, July 1, 2022	44743	2022	7	July	Jul	Jul 2022	1	Friday		27	0	1	0	0	0	0
Friday, July 8, 2022	44750	2022	7	July	Jul	Jul 2022	8	Friday		28	0	1	0	0	0	0
Friday, July 15, 2022	44757	2022	7	July	Jul	Jul 2022	15	Friday		29	0	1	0	0	0	0
Friday, July 22, 2022	44764	2022	7	July	Jul	Jul 2022	22	Friday		30	0	0	0	0	0	0
Friday, July 29, 2022	44771	2022	7	July	Jul	Jul 2022	29	Friday		31	0	0	0	0	0	0
Friday, August 5, 2022	44778	2022	8	August	Aug	Aug 2022	5	Friday		32	0	1	0	0	0	0
Friday, August 12, 2022	44785	2022	8	August	Aug	Aug 2022	12	Friday		33	0	1	0	0	0	0
Friday, August 19, 2022	44792	2022	8	August	Aug	Aug 2022	19	Friday		34	0	0	0	0	0	0
Friday, August 26, 2022	44799	2022	8	August	Aug	Aug 2022	26	Friday		35	0	0	0	0	0	0
Friday, September 2, 2022	44806	2022	9	September	Sep	Sep 2022	2	Friday		36	0	1	0	0	0	0
Friday, September 9, 2022	44813	2022	9	September	Sep	Sep 2022	9	Friday		37	0	1	0	0	0	0
Friday, September 16, 2022	44820	2022	9	September	Sep	Sep 2022	16	Friday		38	0	1	0	0	0	0
Friday, September 23, 2022	44827	2022	9	September	Sep	Sep 2022	23	Friday		39	0	0	0	0	0	0
Friday, September 30, 2022	44834	2022	9	September	Sep	Sep 2022	30	Friday		40	0	0	0	0	0	0
Friday, October 7, 2022	44841	2022	10	October	Oct	Oct 2022	7	Friday		41	0	1	0	0	0	0
Friday, October 14, 2022	44848	2022	10	October	Oct	Oct 2022	14	Friday		42	0	1	0	0	0	0
Friday, October 21, 2022	44855	2022	10	October	Oct	Oct 2022	21	Friday		43	0	0	0	0	0	0
Friday, October 28, 2022	44862	2022	10	October	Oct	Oct 2022	28	Friday		44	0	0	0	0	0	0
Friday, November 4, 2022	44869	2022	11	November	Nov	Nov 2022	4	Friday		45	0	1	0	0	0	0
Friday, November 11, 2022	44876	2022	11	November	Nov	Nov 2022	11	Friday		46	0	1	0	0	0	0
Friday, November 18, 2022	44883	2022	11	November	Nov	Nov 2022	18	Friday		47	0	0	0	0	0	0
Friday, November 25, 2022	44890	2022	11	November	Nov	Nov 2022	25	Friday		48	0	0	0	0	0	0
Friday, December 2, 2022	44897	2022	12	December	Dec	Dec 2022	2	Friday		49	0	1	0	0	0	0
Friday, December 9, 2022	44904	2022	12	December	Dec	Dec 2022	9	Friday		50	0	1	0	0	0	0
Friday, December 16, 2022	44911	2022	12	December	Dec	Dec 2022	16	Friday		51	0	1	0	0	0	0
Friday, December 23, 2022	44918	2022	12	December	Dec	Dec 2022	23	Friday		52	0	0	0	0	0	0
Friday, December 30, 2022	44925	2022	12	December	Dec	Dec 2022	30	Friday		0	0	0	0	0	0	0
Friday, April 21, 2023	45037	2023	4	April	Apr	Apr 2023	21	Friday		16	0	0	0	0	0	0
Friday, April 28, 2023	45044	2023	4	April	Apr	Apr 2023	28	Friday		17	0	0	0	0	0	0
Friday, May 5, 2023	45051	2023	5	May	May	May 2023	5	Friday		18	0	1	0	0	0	0
Friday, May 12, 2023	45058	2023	5	May	May	May 2023	12	Friday		19	0	1	0	0	0	0
Friday, May 19, 2023	45065	2023	5	May	May	May 2023	19	Friday		20	0	0	0	0	0	0
Friday, May 26, 2023	45072	2023	5	May	May	May 2023	26	Friday		21	0	0	0	0	0	0
Friday, June 2, 2023	45079	2023	6	June	Jun	Jun 2023	2	Friday		22	0	1	0	0	0	0
Friday, June 9, 2023	45086	2023	6	June	Jun	Jun 2023	9	Friday		23	0	1	0	0	0	0
Friday, June 16, 2023	45093	2023	6	June	Jun	Jun 2023	16	Friday		24	0	1	0	0	0	0
Friday, June 23, 2023	45100	2023	6	June	Jun	Jun 2023	23	Friday		25	0	0	0	0	0	0
Friday, June 30, 2023	45107	2023	6	June	Jun	Jun 2023	30	Friday		26	0	0	0	0	0	0
Friday, July 7, 2023	45114	2023	7	July	Jul	Jul 2023	7	Friday		27	0	1	0	0	0	0

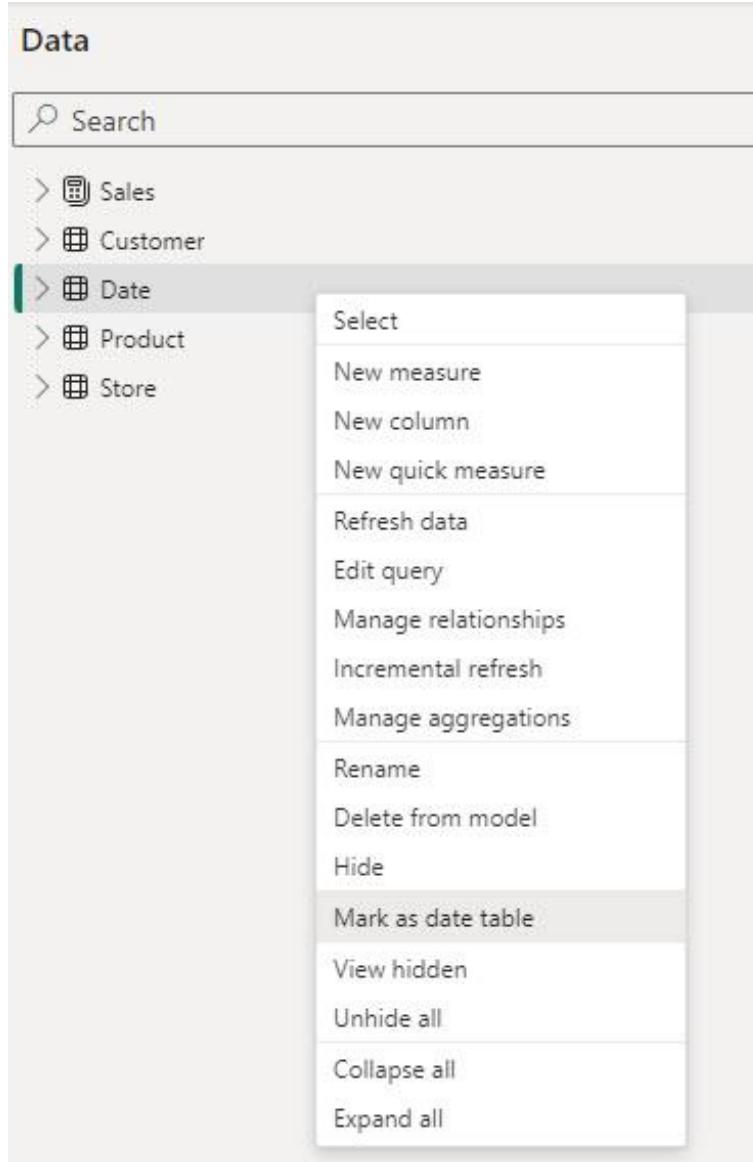
# Another Option External Tools: Bravo

---



# Mark as Date Table

---



# Relationships

---

## Options

### GLOBAL

- Data Load
- Power Query Editor
- DirectQuery
- R scripting
- Python scripting
- Security
- Privacy
- Regional Settings
- Updates
- Usage Data
- Diagnostics
- Preview features
- Auto recovery
- Report settings

### CURRENT FILE

- Data Load
- Regional Settings
- Privacy
- Auto recovery
- Published dataset settings
- Query reduction
- Report settings

### Type Detection

Detect column types and headers for unstructured sources

### Relationships

Import relationships from data sources on first load ⓘ

Update or delete relationships when refreshing data ⓘ

Autodetect new relationships after data is loaded ⓘ

[Learn more](#)

### Time intelligence

Auto date/time ⓘ [Learn more](#)

### Background Data

Allow data previews to download in the background

### Parallel loading of tables ⓘ

Maximum number of concurrent jobs [Learn more](#)

Default

One (disable parallel loading)

Custom

### Q&A

Turn on Q&A to ask natural language questions about your data ⓘ [Learn more](#)

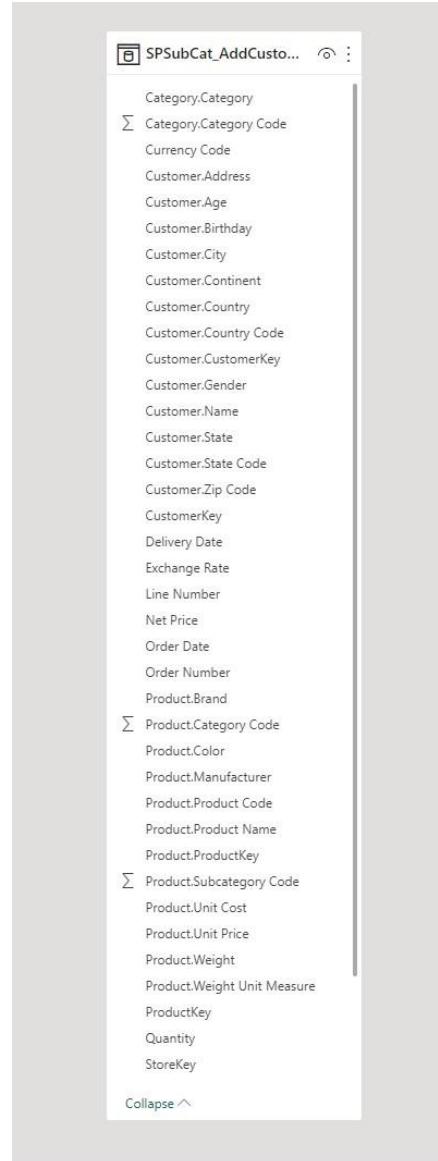
Share your synonyms with everyone in your org

OK

Cancel

# One big wide table (or tall)

---

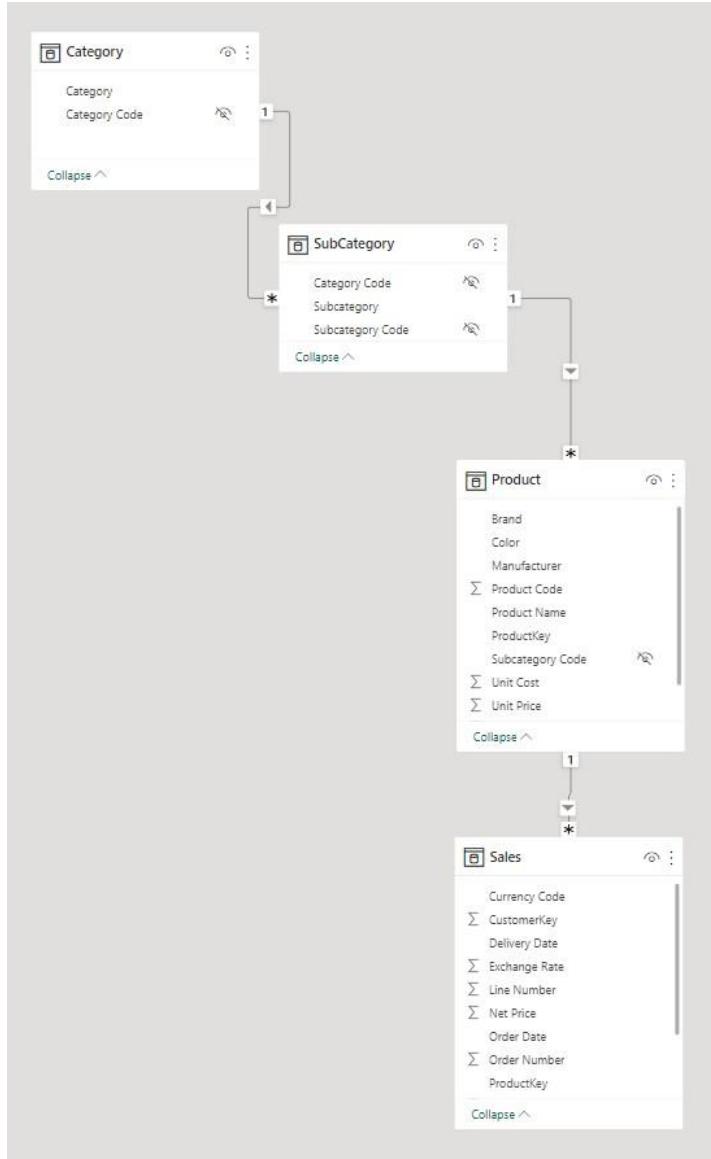


## Wide Table

- Difficult to find the column you are looking for
- Refresh can take longer
- Performance is better with Star Schema
- DAX is more complicated

# What is Snowflake Schema

---

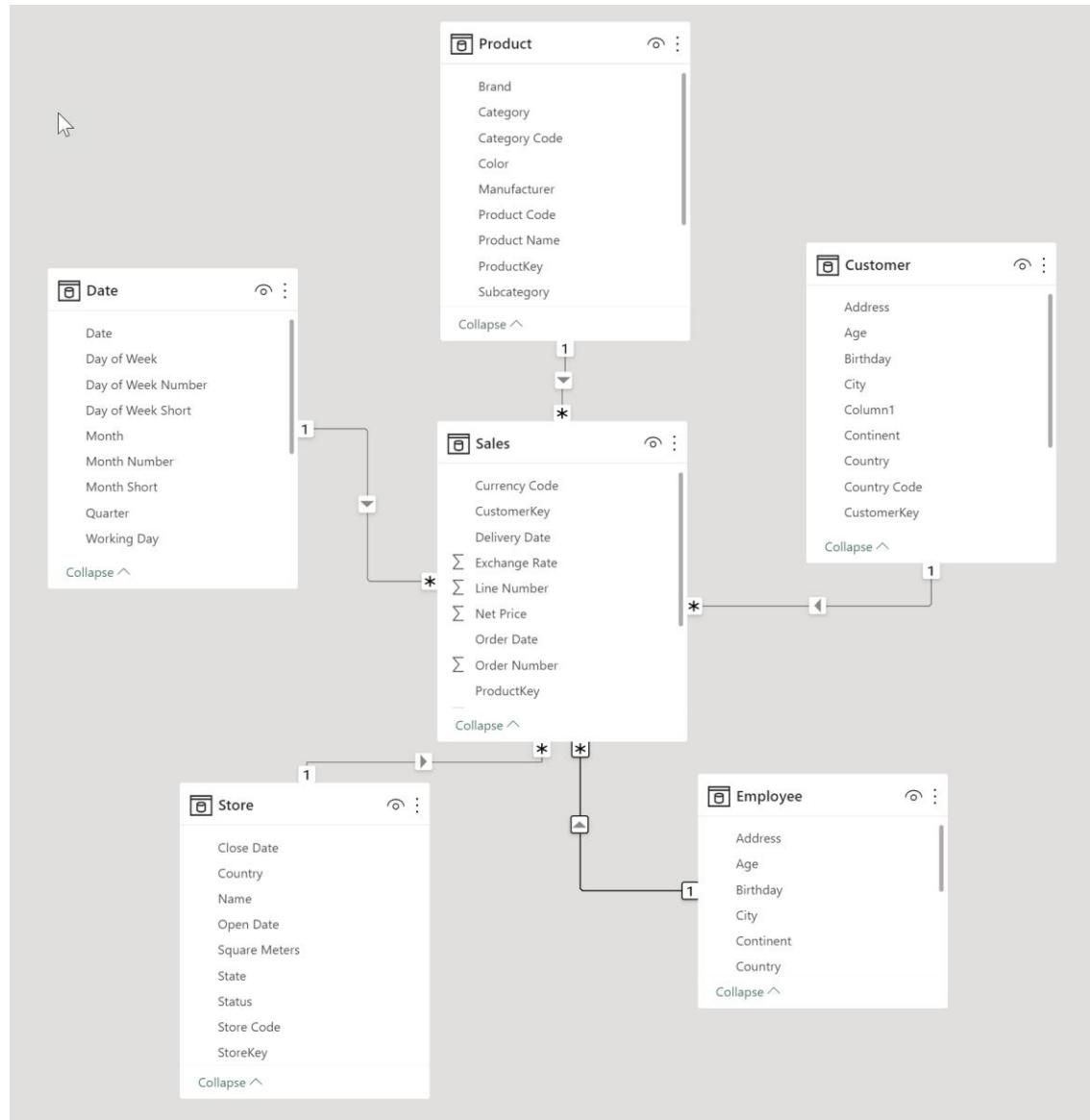


# Other examples of Snowflake Schema

- General Ledger
  - Header and Detail
  - Combine into one table
- Employee
  - Employee SubCategory
  - Employee Category

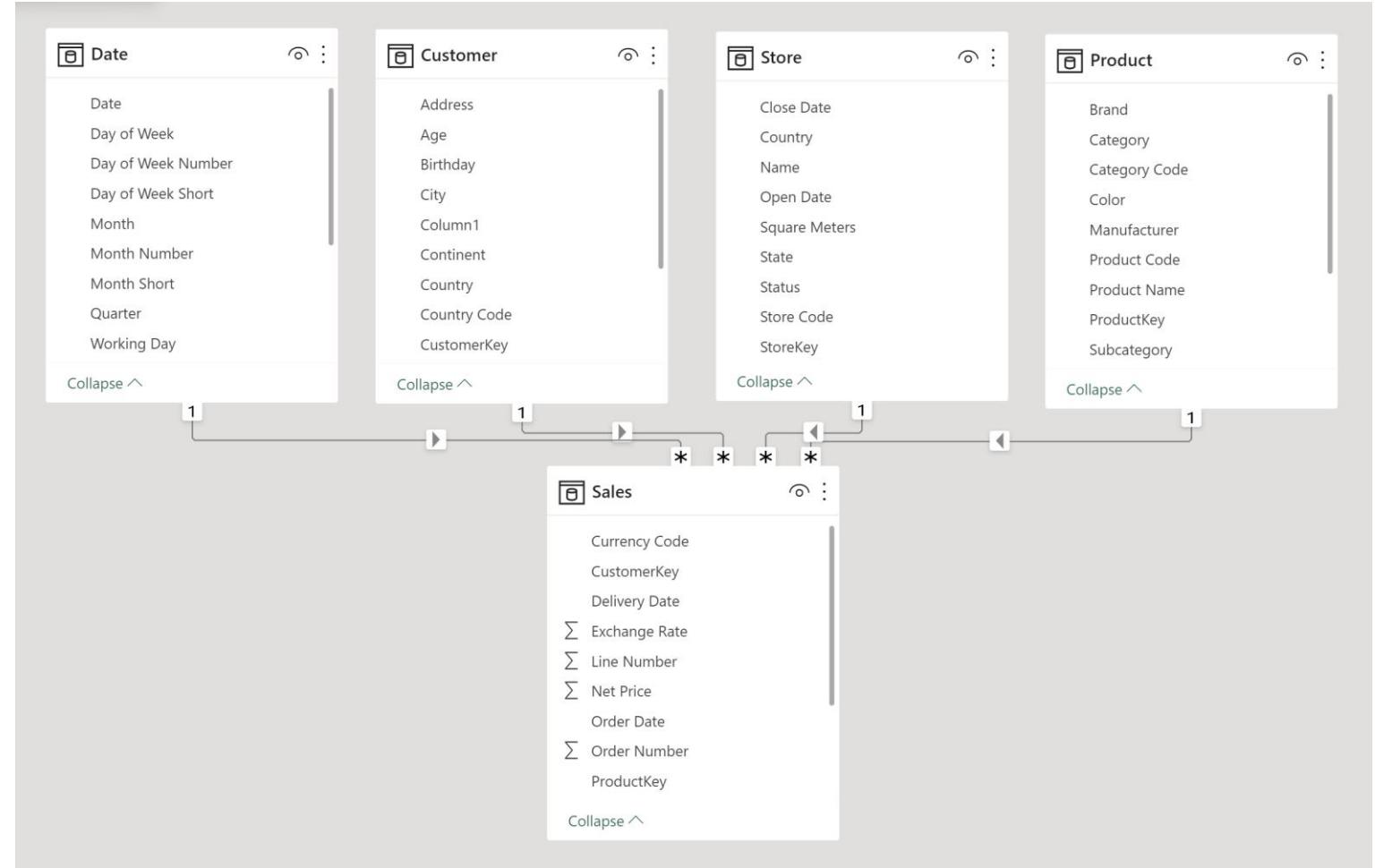
# What is a Star Schema

---



# Collie Method

---



# **Snowflake versus Star Schema**

## Snowflake

- Harder to understand as a developer
- Expanded table issue with DAX

## Star Schema

- Simpler
- Overall better performance

# Expanded Tables

- When you apply a table filter it really is the expanded table.
  - CALCULATE ( COUNTROWS (Subcategory), Product )
- It is the expanded table for Product. So only those subcategories that have a product
  - Can't reference base table

# Simplifies DAX

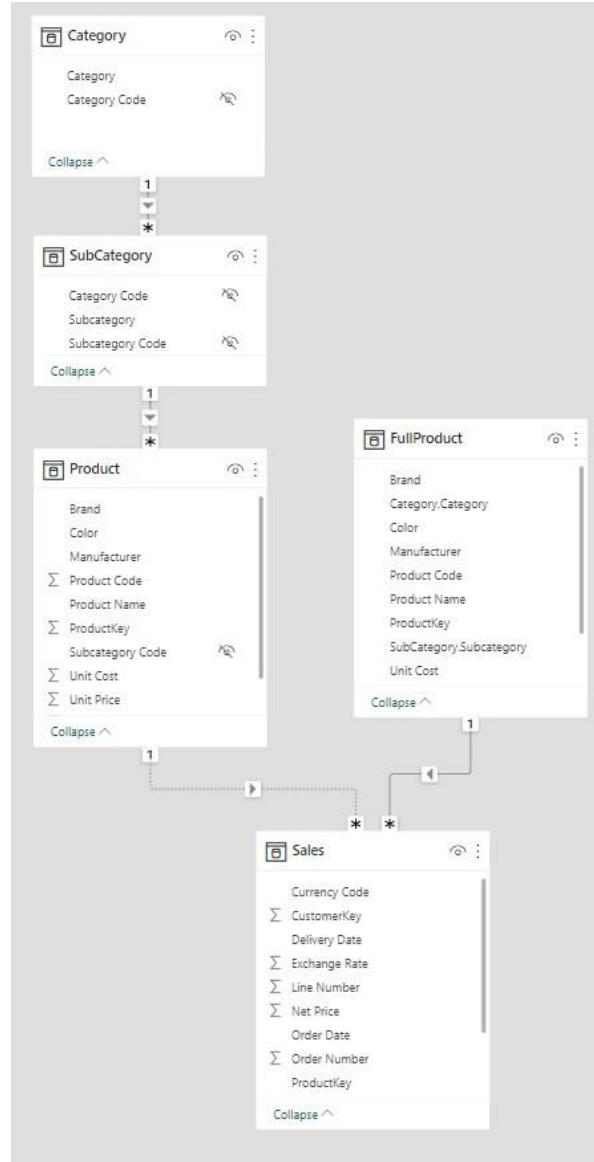
```
MAXX (  
    VALUES ( 'Product'[Brand] ),  
    [Sales Amount]  
)
```

For each Brand

Compute Max Sales Amount

# Snowflake to (Super) Star Schema

---



# Naming – Tables and Columns

---

- Names should be for humans
- Whether you put Fact or Dim is debatable
- Name with “Gold”, “Silver”, “Test”, “Prod”

# Hiding Columns

- Hide Key Columns
- Build Measures on top of value columns and hide the base column
- Implicit Measures versus Explicit Measures



# Implicit Measures

---

- Currently if you create Calculation Groups it disables implicit measures
- Discourage Implicit Measures
- Build explicit Measures

# Folders

---

*Organize your  
measures*



# Calculated Columns

---

- Create all calculated columns in source if possible
- Optimal sorting algorithm only on imported columns



# Relationships

---

- One-To-Many
- Single Direction
- Bi-Directional
  - Harder to understand
  - Not as performant

# Data Types

---

- Same type for relationships
- Smaller precision – less storage
  - Currency – Fixed Decimal
    - Versus Double or Decimal

# Our Plan

---

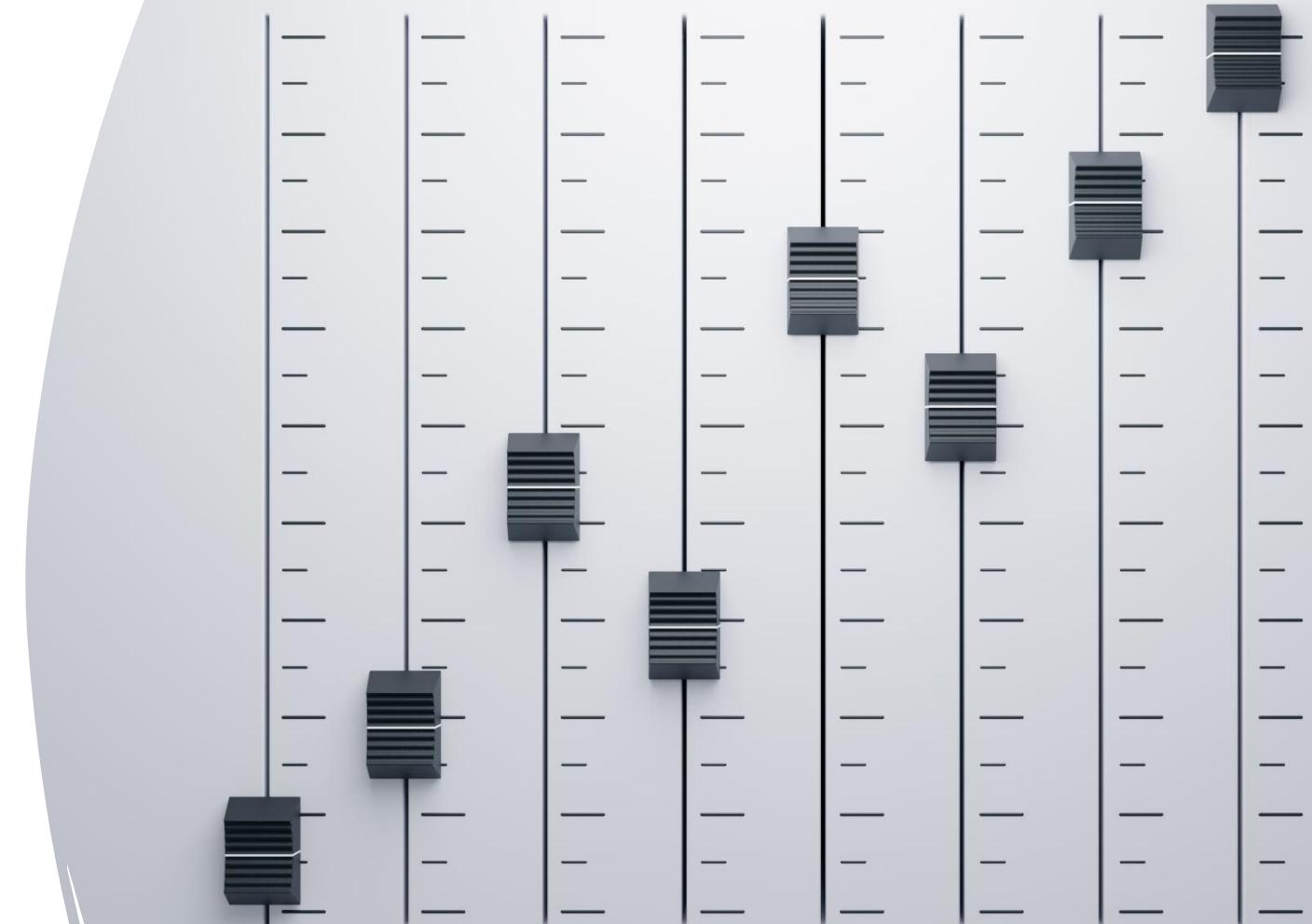


1. Intro
2. Power Query
3. Data Model
- 4. External Tools**
5. Conclusion

# Tabular Editor

---

- With C# scripts you can automate creating or changing aspects of the model
  - For example, create a measure for every column that has the word amount



# Tabular Editor – Best Practice Analyzer

Manage Best Practice Rules

Current model

Rule collections:

- (Effective rules)
- Rules within the current model
- Rules for the local user
- Rules on the local machine

Add... Remove ▲ ▼

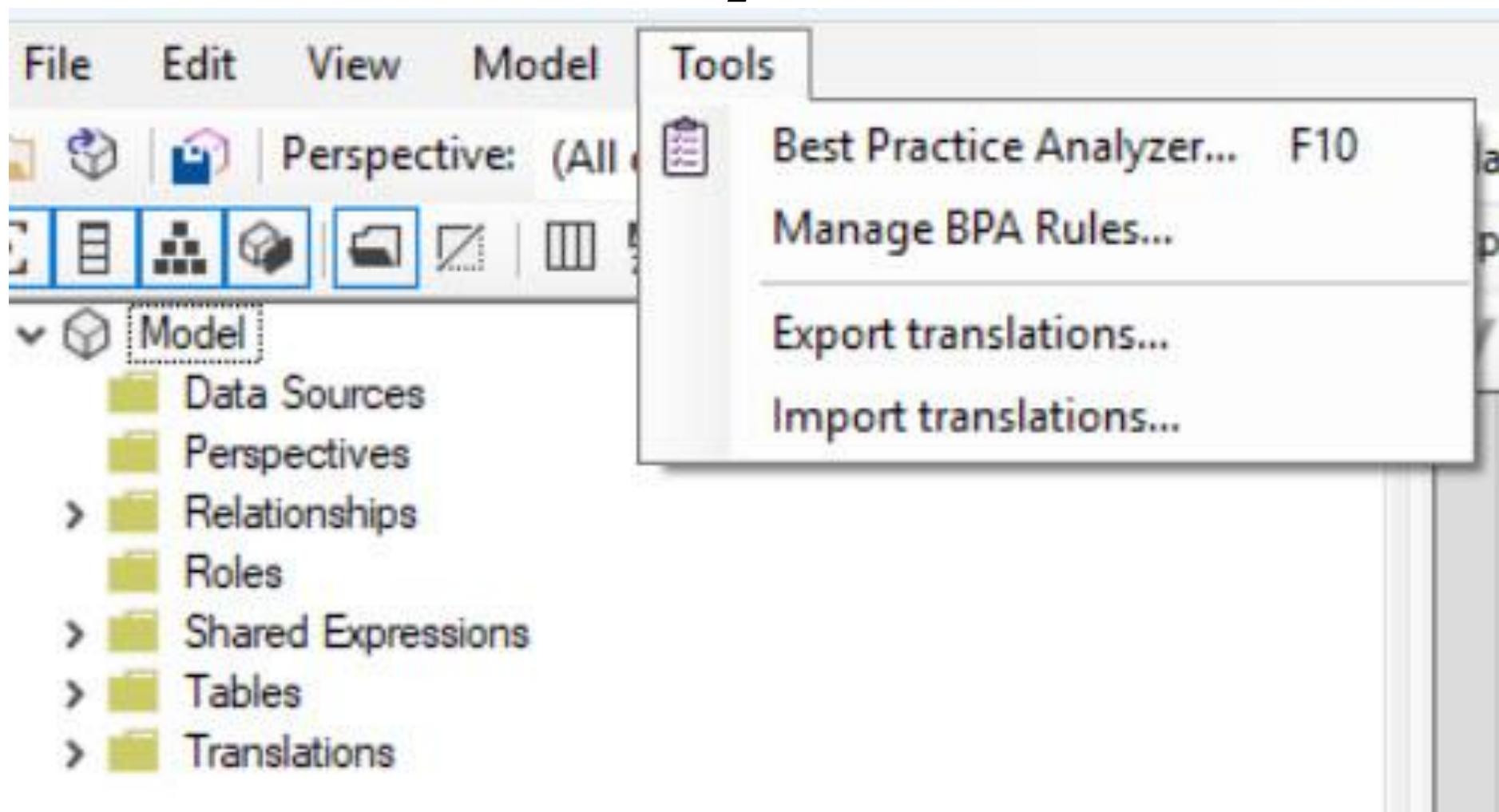
Rules in collection:

Rule name	Scope
[Error Prevention] Data columns must have a source c...	Data Columns
[Error Prevention] Expression-reliant objects must have...	Measures,Calculated Columns,C...
[Error Prevention] Relationship columns should be of t...	Relationships
Formatting	
[Formatting] Add data category for columns	Columns
[Formatting] Do not summarize numeric columns	Columns
[Formatting] First letter of objects must be capitalized	Tables,Measures,Hierarchies,Cal...
[Formatting] Format flag columns as Yes/No value strin...	Columns
[Formatting] Hide fact table columns	Columns
[Formatting] Hide foreign keys	Columns
[Formatting] Mark primary keys	Columns
[Formatting] Month (as a string) must be sorted	Columns
[Formatting] Objects should not start or end with a space	Model,Tables,Measures,Hierarch...
[Formatting] Percentages should be formatted with tho...	Measures
[Formatting] Provide format string for "Date" columns	Columns
[Formatting] Provide format string for "Month" columns	Columns
[Formatting] Provide format string for measures	Measures
[Formatting] Relationship columns should be of integer ...	Columns
[Formatting] Whole numbers should be formatted with t...	Measures
Maintenance	
[Maintenance] Calculation groups with no calculation it...	Calculation Groups
[Maintenance] Ensure tables have relationships	Tables Calculated Tables

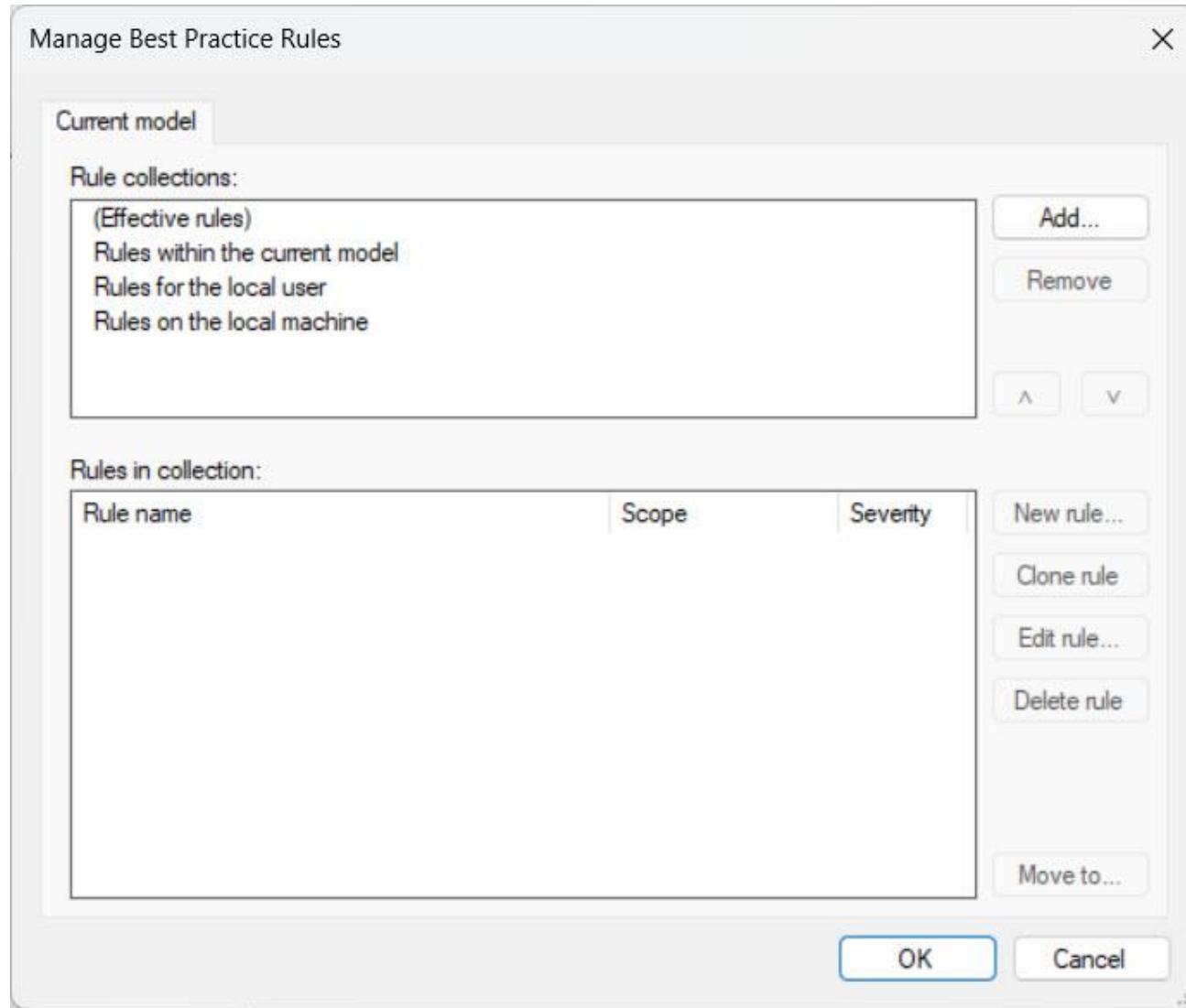
New rule... Clone rule Edit rule... Delete rule

Move to... OK Cancel

# Best Practice Analyzer Location



# Add and Manage Rules



# Best Practice Analyzer – C# Script Download

```
System.Net.WebClient w = new System.Net.WebClient();

string path = System.Environment.GetFolderPath(System.Environment.SpecialFolder.LocalApplicationData);
string url = "https://raw.githubusercontent.com/microsoft/Analysis-Services/master/BestPracticeRules/BPARules.json";
string version = System.Windows.Forms.Application.ProductVersion.Substring(0,1);
string downloadLoc = path + @"\TabularEditor\BPARules.json";

if (version == "3")
{
    downloadLoc = path + @"\TabularEditor3\BPARules.json";
}

w.DownloadFile(url, downloadLoc);

/*
// Italian
string url = "https://raw.githubusercontent.com/microsoft/Analysis-Services/master/BestPracticeRules/Italian/BPARules.json";

// Japanese
string url = "https://raw.githubusercontent.com/microsoft/Analysis-Services/master/BestPracticeRules/Japanese/BPARules.json";

// Spanish
string url = "https://raw.githubusercontent.com/microsoft/Analysis-Services/master/BestPracticeRules/Spanish/BPARules.json";
*/
```

# Nope – Not Running a Script

- Alternate Method
  - Microsoft Analysis-Services – Best Practice Rules
    - <https://github.com/microsoft/Analysis-Services/tree/master/BestPracticeRules>
    - Other languages besides English

# Identifies

---

- Data Types
- Bad DAX patterns
- Formatting

# Visualization Layer



**Beyond  
the  
Scope**

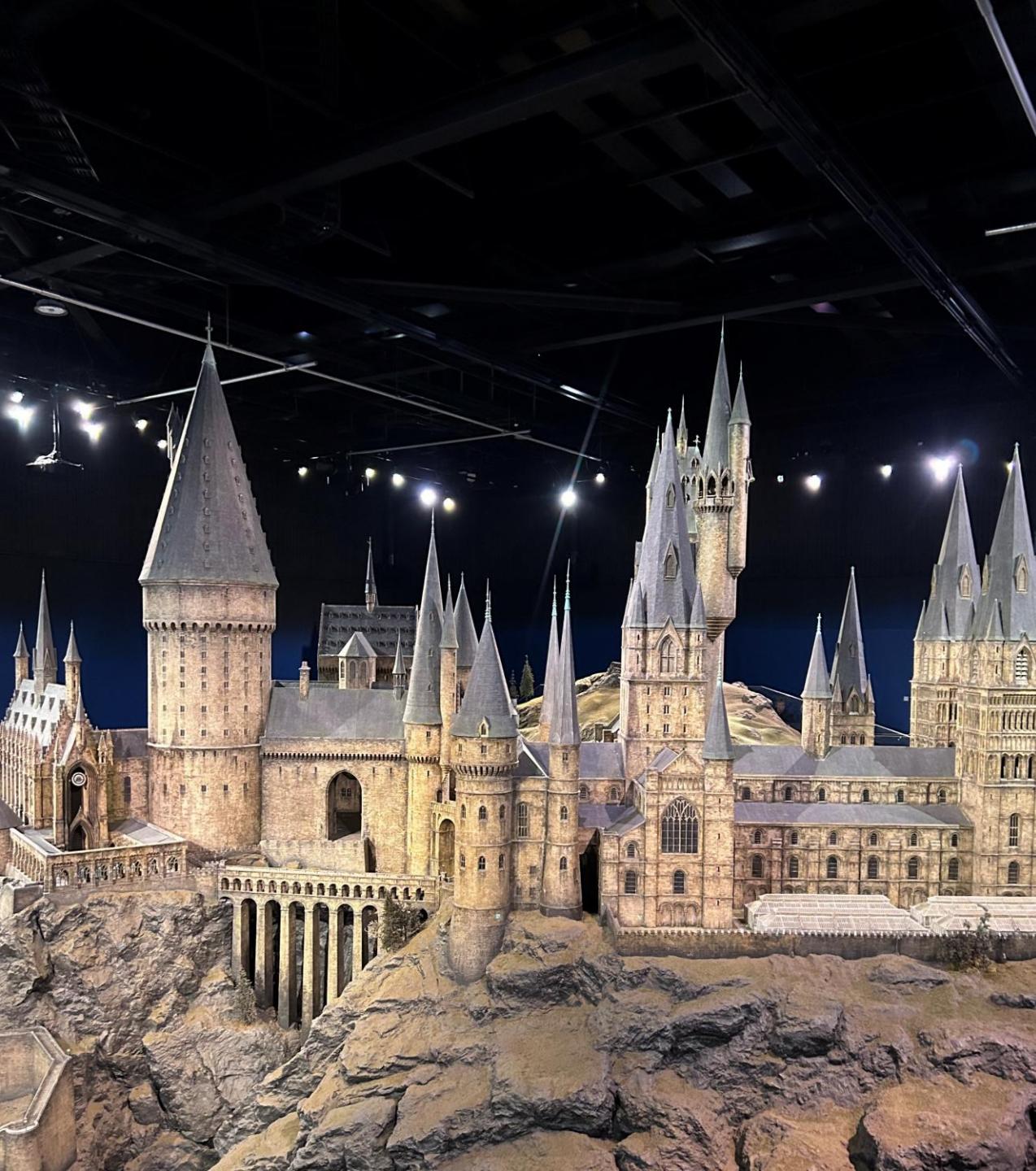


# Our Plan

---



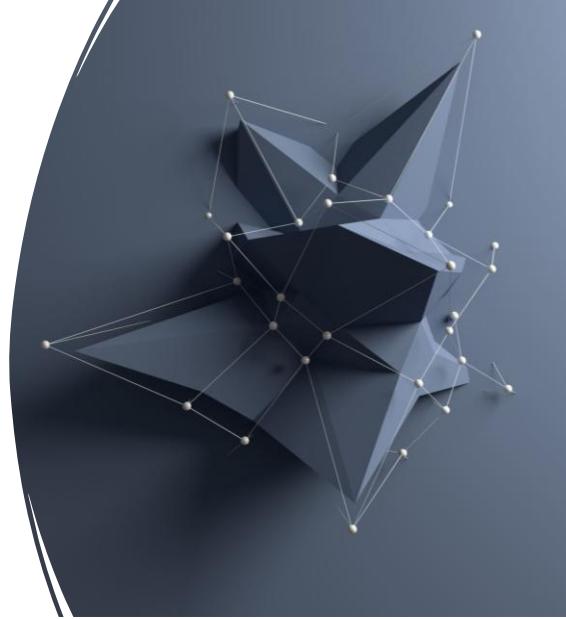
1. Intro
2. Power Query
3. Data Model
4. External Tools
- 5. Conclusion**



# Model Goals

# Conclusion

---



- Shape data as close to the source as possible
- Import it into a Star Schema
- Use external tools to help optimize the model

# Resources

---

- Guide to Query Folding
- Improve Model Performance
- Bravo for Power BI
- Tabular Editor 2



# Resources

---

- Kristyna's Date Table M
  - <https://github.com/Anytsirk12/DataOnWheels/tree/main/SQLBits%202024>
- Blog: Data on Wheels
  - <https://dataonwheels.wordpress.com/>
- DAX Studio
  - <https://daxstudio.org/>

# Evaluate My Session

---



# Thank you

**Jason Romans**

**thedaxshepherd@gmail.com**

**www.thedaxshepherd.com**

