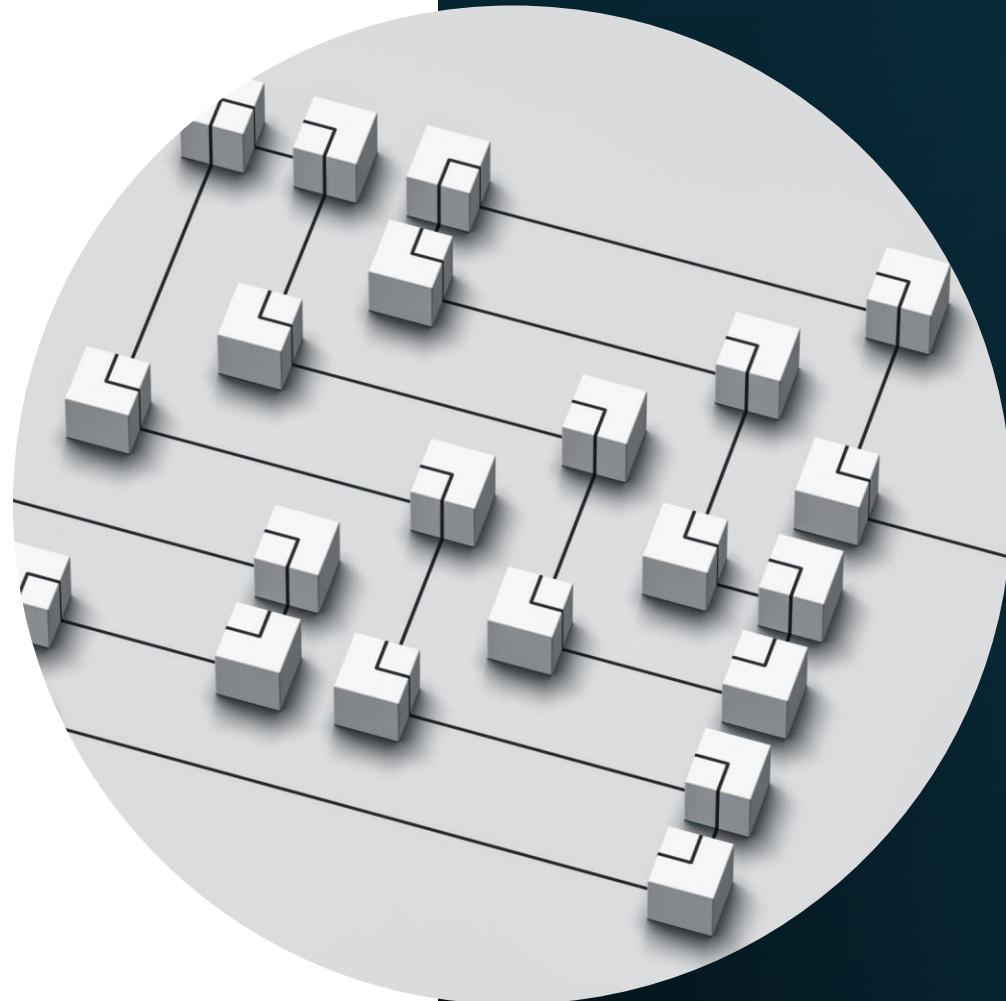


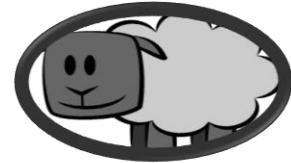
Basics of Building a Data Model in Power BI

Jason Romans



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



Favorite Data Model

Our Journey



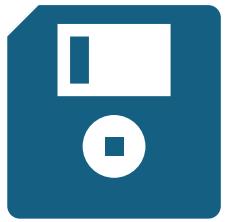
- 1. Intro**
- 2. Power Query**
- 3. Data Model**
- 4. External Tools**
- 5. Conclusion**

Our Journey

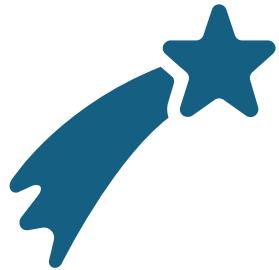


- 1. Intro**
2. Power Query
3. Data Model
4. External Tools
5. Conclusion

Daunting Task



How do I bring in the data



What should the model
look like



Visualize Data

What you
picture



Reality





Encourage not discourage

WB Studio Tour

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

- Unfortunately, no step-by-step instructions
- Lego blocks
- Guidelines

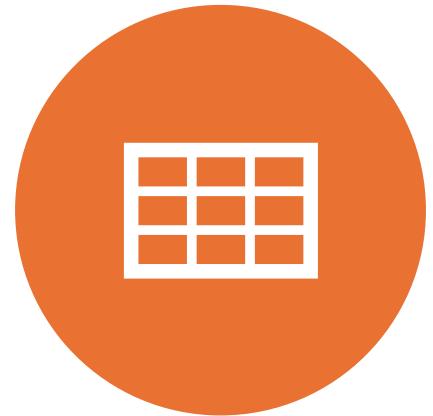
Our Journey



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

Get Data

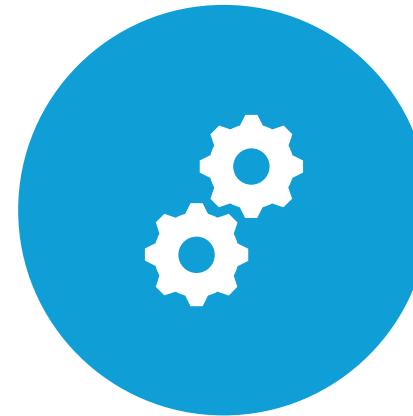
Where is the data



EXCEL SPREADSHEET

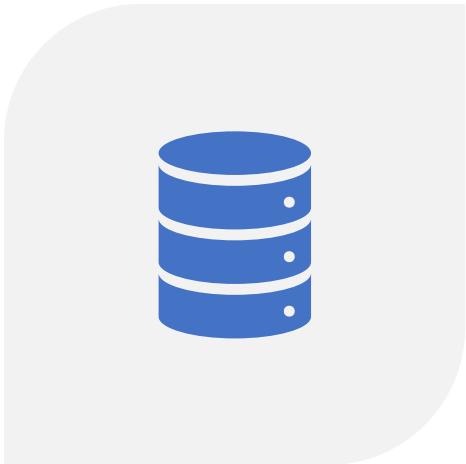


SQL SERVER

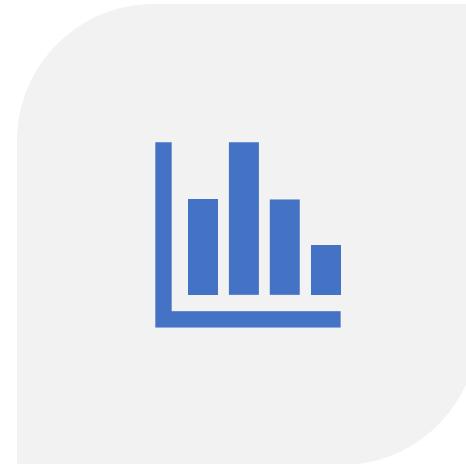


CSV

What form is the data in



SOURCE DATA THAT IS IN
SEPARATE TABLES

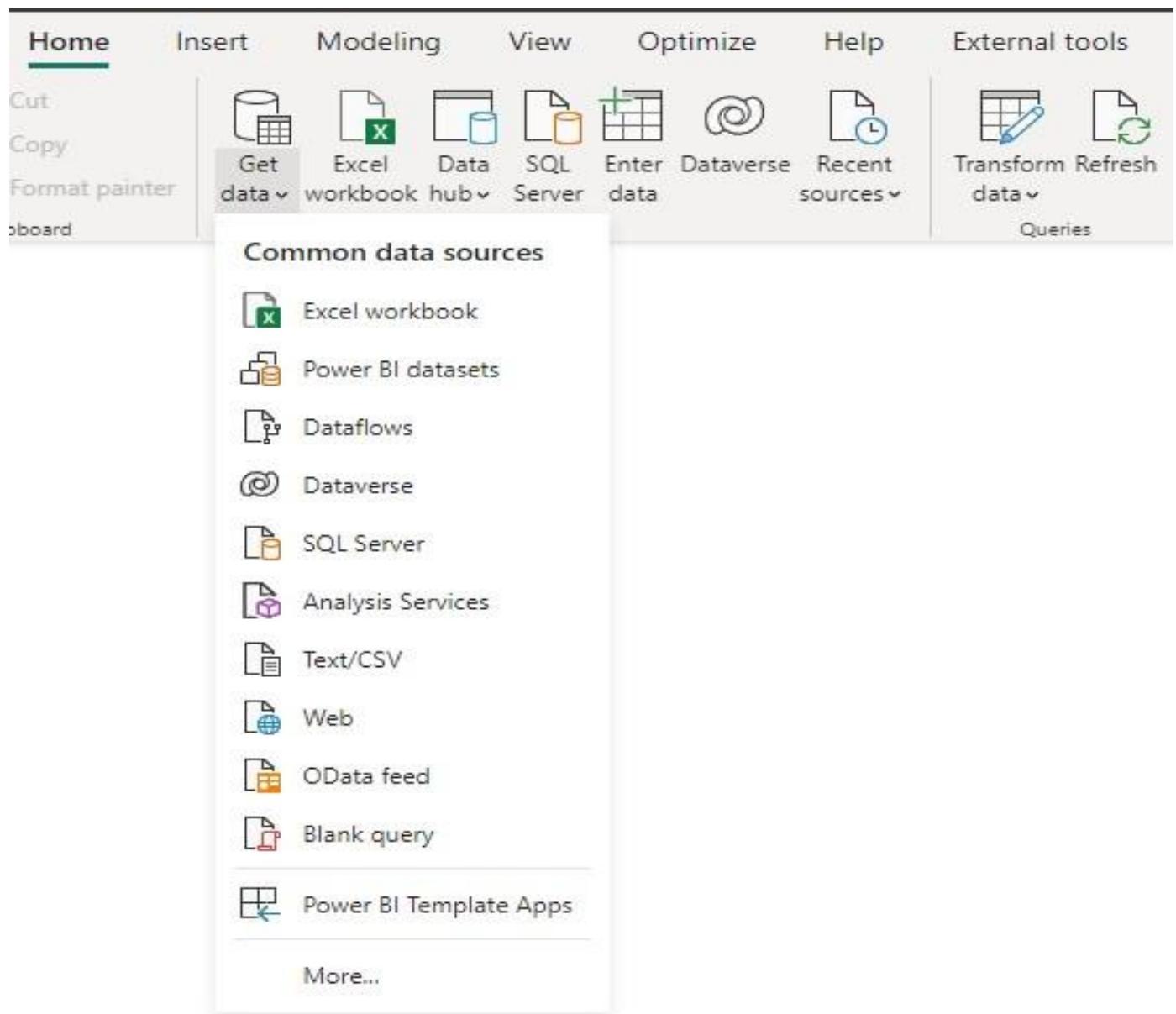


REPORT FORM WITH ALL
THE DATA COMBINED

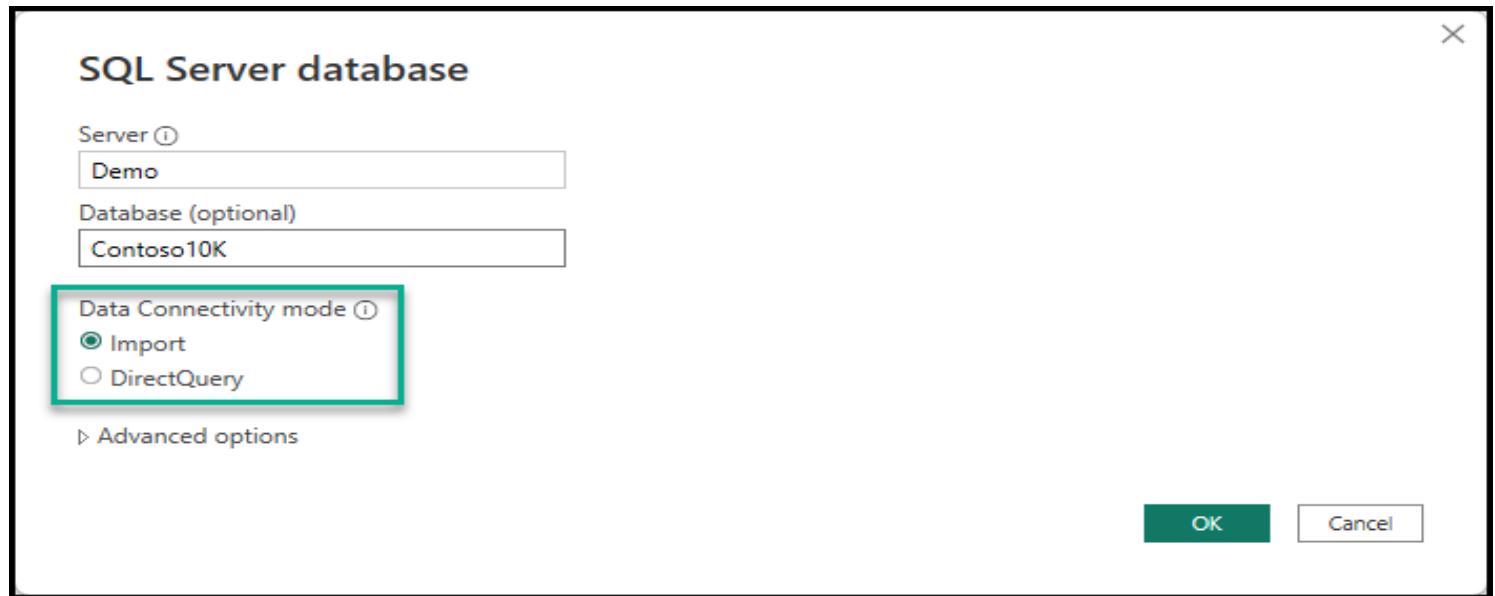
Toy Model Builder

- 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	

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_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

Power Query Editor

Untitled - Power Query Editor

File Home Transform Add Column View Tools Help

Close & Apply New Recent Enter Data source settings Manage Parameters Refresh Advanced Editor Properties Close Data Sources Parameters Query Choose Columns Remove Columns Keep Rows Remove Rows Sort Split Column Group By Data Type: Whole Number Use First Row as Headers Replace Values Transform

Queries [4]

Store Product Sales Customer

= Source{[Schema="dbo",Item="Sales"]}[Data]

	Order Number	Line Number	Order Date	Delivery Date	CustomerKey
1	36600	0	1/1/2018	1/1/2018	
2	36700	0	1/2/2018	1/2/2018	
3	36700	1	1/2/2018	1/2/2018	
4	36700	2	1/2/2018	1/2/2018	
5	36700	3	1/2/2018	1/2/2018	
6	36700	4	1/2/2018	1/2/2018	
7	36701	0	1/2/2018	1/2/2018	
8	36800	0	1/3/2018	1/10/2018	
9	36800	1	1/3/2018	1/10/2018	
10	36800	2	1/3/2018	1/10/2018	
11	36800	3	1/3/2018	1/10/2018	
12	36801	0	1/3/2018	1/3/2018	
13	36801	1	1/3/2018	1/3/2018	
14	36801	2	1/3/2018	1/3/2018	
15	36801	3	1/3/2018	1/3/2018	
16	36900	0	1/4/2018	1/4/2018	
17	36901	0	1/4/2018	1/14/2018	
18	36902	0	1/4/2018	1/17/2018	
19	37000	0	1/5/2018	1/5/2018	
20	37000	1	1/5/2018	1/5/2018	
21	37000	2	1/5/2018	1/5/2018	
22	37000	3	1/5/2018	1/5/2018	
23	37100	0	1/6/2018	1/6/2018	
24	37100	1	1/6/2018	1/6/2018	
25					

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

Query Settings

PROPERTIES

Name: Sales

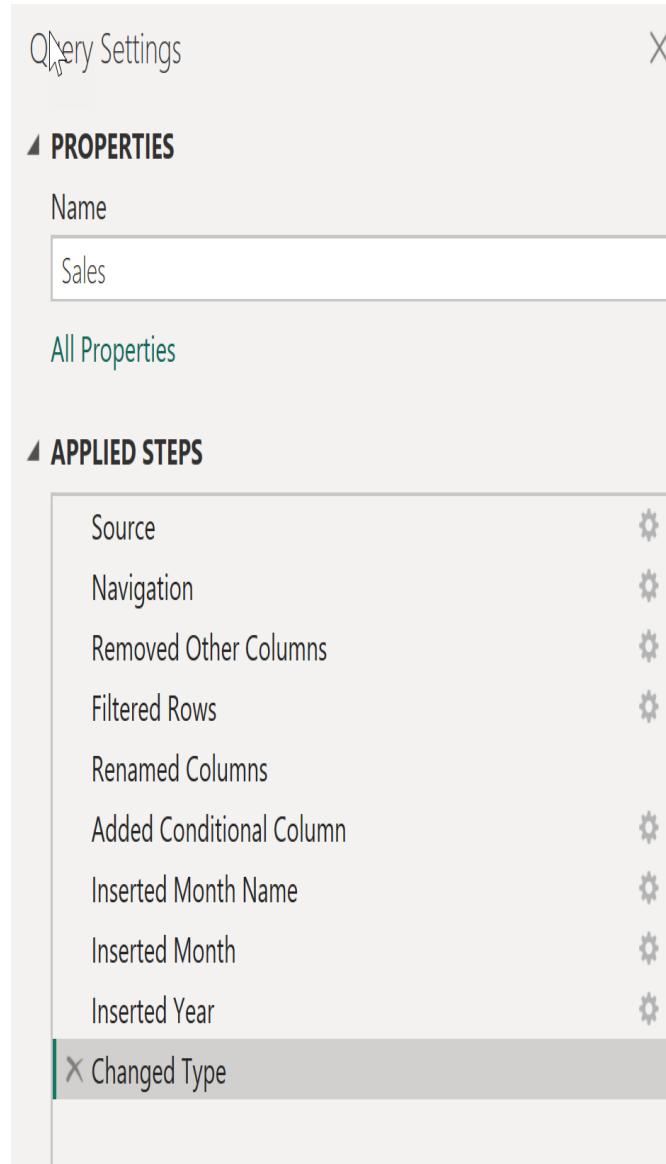
All Properties

APPLIED STEPS

Source Navigation

PREVIEW DOWNLOADED AT 2:04 PM

Power Query Steps like a Recipe



Promote First Row to Headers (CSV)

The screenshot shows the Microsoft Power Query Editor interface. The title bar reads "Untitled - Power Query Editor". The top navigation bar includes "File", "Home", "Transform" (which is selected), "Add Column", "View", "Tools", and "Help". The "Transform" tab has several options: "Group By" (with "Use First Row as Headers" dropdown), "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 states: "Promote the first row of this table into column headers." Below the toolbar is a query editor pane titled "Sales" containing 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)

Untitled - Power Query Editor

File Home Transform Add Column View Tools Help

New Recent Enter Data source settings Manage Parameters Refresh Preview Manage

Close & Apply Close New Query Data Source... Parameters Query Sort Transform

Manage Columns Reduce Rows Sort

Data Type: Text ▾ Use First Row as Headers ▾ Split Column By Replace Values

Text Analytics Vision Azure Machine Learning AI Insights

Queries [1] Sales

= Csv.Document(File.Contents("C:\ContosoDataFiles\Sales.csv"),[Delimiter=",",

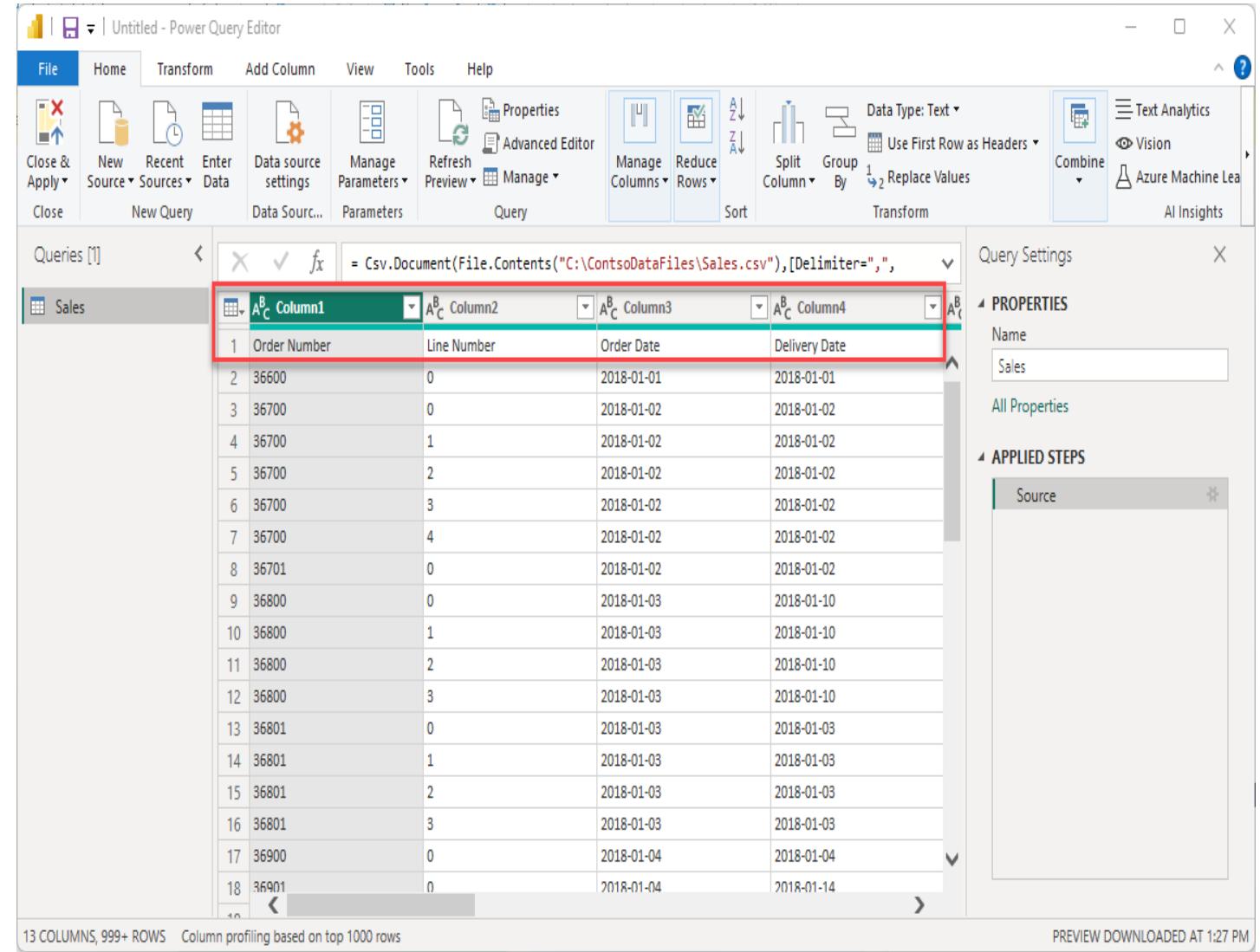
	Column1	Column2	Column3	Column4
1	Order Number	Line Number	Order Date	Delivery Date
2	36600	0	2018-01-01	2018-01-01
3	36700	0	2018-01-02	2018-01-02
4	36700	1	2018-01-02	2018-01-02
5	36700	2	2018-01-02	2018-01-02
6	36700	3	2018-01-02	2018-01-02
7	36700	4	2018-01-02	2018-01-02
8	36701	0	2018-01-02	2018-01-02
9	36800	0	2018-01-03	2018-01-10
10	36800	1	2018-01-03	2018-01-10
11	36800	2	2018-01-03	2018-01-10
12	36800	3	2018-01-03	2018-01-10
13	36801	0	2018-01-03	2018-01-03
14	36801	1	2018-01-03	2018-01-03
15	36801	2	2018-01-03	2018-01-03
16	36801	3	2018-01-03	2018-01-03
17	36900	0	2018-01-04	2018-01-04
18	36901	0	2018-01-04	2018-01-14

13 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 1:27 PM

Query Settings

PROPERTIES Name Sales All Properties

APPLIED STEPS Source



Promote First Row to Headers (CSV)

Untitled - Power Query Editor

File Home Transform Add Column View Tools Help

New Recent Enter Data source settings Manage Parameters Refresh Preview Manage

Close & Apply Close New Query Data Source... Parameters Query Sort Transform

Manage Columns Reduce Rows Sort

Data Type: Text Use First Row as Headers

Split Group By Replace Values

Text Analytics Vision Azure Machine Lea

Combine AI Insights

Queries [1] Sales = Table.PromoteHeaders(Source, [PromoteAllScalars=true])

Order Number Line Number Order Date Delivery Date

1	36600	0	2018-01-01	2018-01-01
2	36700	0	2018-01-02	2018-01-02
3	36700	1	2018-01-02	2018-01-02
4	36700	2	2018-01-02	2018-01-02
5	36700	3	2018-01-02	2018-01-02
6	36700	4	2018-01-02	2018-01-02
7	36701	0	2018-01-02	2018-01-02
8	36800	0	2018-01-03	2018-01-10
9	36800	1	2018-01-03	2018-01-10
10	36800	2	2018-01-03	2018-01-10
11	36800	3	2018-01-03	2018-01-10
12	36801	0	2018-01-03	2018-01-03
13	36801	1	2018-01-03	2018-01-03
14	36801	2	2018-01-03	2018-01-03
15	36801	3	2018-01-03	2018-01-03
16	36900	0	2018-01-04	2018-01-04
17	36901	0	2018-01-04	2018-01-14
18	36902	0	2018-01-04	2018-01-17

13 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 1:28 PM

Query Settings

PROPERTIES Name Sales All Properties

APPLIED STEPS Source Promoted Headers

Remove Other Columns

Screenshot of the Power Query Editor showing the "Remove Other Columns" feature.

The ribbon shows the "Home" tab selected. In the "Transform" section, the "Remove Other Columns" button is highlighted.

The query editor displays a table titled "Sales" with columns: Order Number, Line Number, Order Date, Delivery Date, CustomerKey, and Sales. The "Remove Other Columns" dropdown menu is open, showing options like "Remove Columns", "Keep Rows", and "Remove Rows".

The "APPLIED STEPS" pane shows a single step named "Navigation".

Properties pane shows the query name is "Sales".

Query Settings pane shows the "Source" step is applied.

Bottom status bar: 13 COLUMNS, 999+ ROWS | Column profiling based on top 1000 rows | PREVIEW DOWNLOADED ON TUESDAY

Order Number	Line Number	Order Date	Delivery Date	CustomerKey	Sales
1	36600	0	1/1/2018	1/1/2018	1620763
2	36700	0	1/2/2018	1/2/2018	1320351
3	36700	1	1/2/2018	1/2/2018	1320351
4	36700	2	1/2/2018	1/2/2018	1320351
5	36700	3	1/2/2018	1/2/2018	1320351
6	36700	4	1/2/2018	1/2/2018	1320351
7	36701	0	1/2/2018	1/2/2018	2077250
8	36800	0	1/3/2018	1/10/2018	1743422
9	36800	1	1/3/2018	1/10/2018	1743422
10	36800	2	1/3/2018	1/10/2018	1743422
11	36800	3	1/3/2018	1/10/2018	1743422
12	36801	0	1/3/2018	1/3/2018	1013948
13	36801	1	1/3/2018	1/3/2018	1013948
14	36801	2	1/3/2018	1/3/2018	1013948
15	36801	3	1/3/2018	1/3/2018	1013948
16	36900	0	1/4/2018	1/4/2018	544050
17	36901	0	1/4/2018	1/14/2018	485447
18	36902	0	1/4/2018	1/17/2018	1075463
19	37000	0	1/5/2018	1/5/2018	1319917
20	37000	1	1/5/2018	1/5/2018	1319917
21	37000	2	1/5/2018	1/5/2018	1319917
22	37000	3	1/5/2018	1/5/2018	1319917
23	37100	0	1/6/2018	1/6/2018	1107123
24	37100	1	1/6/2018	1/6/2018	1107123
25					

Filter Rows

Untitled - Power Query Editor

File Home Transform Add Column View Tools Help

New Recent Enter Data source settings Manage Parameters Refresh Advanced Editor Properties Preview Manage Choose Columns Remove Keep Rows Remove Sort Split Column Group By Replace Values Data Type: Date Use First Row as Headers Combine Text Analytics Vision Azure Machine Learning AI Insights

Queries [1] Sales = Contoso10K[[Schema="dbo",Item="Sales"]][Data]

	Order Number	Line Number	Order Date	Delivery Date	CustomerKey
1	36600	0	1/1/2018	1/1/2018	1620763
2	36700	0	1/2/2018	1/2/2018	1320351
3	36700	1	1/2/2018	1/2/2018	1320351
4	36700	2	1/2/2018	1/2/2018	1320351
5					1320351
6					1320351
7					2077250
8					1743422
9					1743422
10					1743422
11					1743422
12					1013948
13					1013948
14					1013948
15					1013948
16					544050
17					485447
18					1075463
19					1319917
20					1319917
21					1319917
22					1319917
23					1107123
24					1107123
25					

Filter Rows

Apply one or more filter conditions to the rows in this table.

Basic Advanced

Keep rows where 'Order Date'

And Or

OK Cancel

13 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows 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
Store Size

	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	

Add Clause

Else ⓘ ABC 123 Large

OK Cancel

Add Column from Example

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

	A ^B _C Zip Code	A ^B _C Country Code	A ^B _C Country	A ^B _C Continent	A ^B _C Birthday	Column1
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	

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]))

	A ^B _C Zip Code	A ^B _C Country Code	A ^B _C Country	A ^B _C Continent	A ^B _C 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	00336	US	United States	North America	8/23/1990 0:00	1990

OK Cancel

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/>

Demo

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 data consists of 25 rows of order information. The "Applied Steps" pane on the right lists a step named "Removed Other Columns". A context menu is open over this step, with the option "View Native Query" highlighted.

	Order Number	Line Number	Order Date	Delivery Date	CustomerKey
1	53000	0	6/14/2018	6/14/2018	1365902
2	53000	1	6/14/2018	6/14/2018	1365902
3	53200	0	6/16/2018	6/16/2018	1656264
4	53200	1	6/16/2018	6/16/2018	1656264
5	53200	2	6/16/2018	6/16/2018	1656264
6	53200	3	6/16/2018	6/16/2018	1656264
7	53600	0	6/20/2018	7/2/2018	1799884
8	53700	0	6/21/2018	6/21/2018	1373034
9	53700	1	6/21/2018	6/21/2018	1373034
10	53700	2	6/21/2018	6/21/2018	1373034
11	53900	0	6/23/2018	6/23/2018	326607
12	53900	1	6/23/2018	6/23/2018	326607
13	53900	2	6/23/2018	6/23/2018	326607
14	54300	0	6/27/2018	6/27/2018	687715
15	54400	0	6/28/2018	6/28/2018	1734949
16	54400	1	6/28/2018	6/28/2018	1734949
17	54600	0	6/30/2018	6/30/2018	1043780
18	54600	1	6/30/2018	6/30/2018	1043780
19	55000	0	7/4/2018	7/4/2018	98175
20	55000	1	7/4/2018	7/4/2018	98175
21	55000	2	7/4/2018	7/4/2018	98175
22	55000	3	7/4/2018	7/4/2018	98175
23	55000	4	7/4/2018	7/4/2018	98175
24	55000	5	7/4/2018	7/4/2018	98175
25	55000	6	7/4/2018	7/4/2018	98175

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

PREVIEW DOWNLOADED AT 2:06 PM

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

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 formula bar at the top contains the expression: `= Table.LastN(#"Removed Other Columns", 20)`. On the right side, there are sections for 'Query Settings' and 'APPLIED STEPS'. Under 'APPLIED STEPS', the step 'Removed Other Columns' is selected, and a context menu is open with options like 'Edit Settings', 'Rename', 'Delete', etc. The bottom status bar indicates '5 COLUMNS, 20 ROWS' and 'Column profiling based on top 1000 rows'.

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

Query Folding

The screenshot shows the Microsoft Power Query Editor interface. On the left, there's a preview pane displaying a table titled "Sales" with columns "Order Number" and "Line Number". The table contains 20 rows of data, all starting with the value 36. To the right of the preview is a "Native Query" dialog box containing the following SQL code:

```
select top 20
    [Order Number],
    [Line Number],
    [Order Date],
    [Delivery Date],
    [CustomerKey]
from [dbo].[Sales] as [$Table]
```

Below the preview and the query editor are two status bars: "5 COLUMNS, 20 ROWS" and "Column profiling based on top 1000 rows". In the bottom right corner of the main window, it says "PREVIEW DOWNLOADED AT 2:53 PM".

Keep Top 20 Rows does not
SQL Server has a top N clause

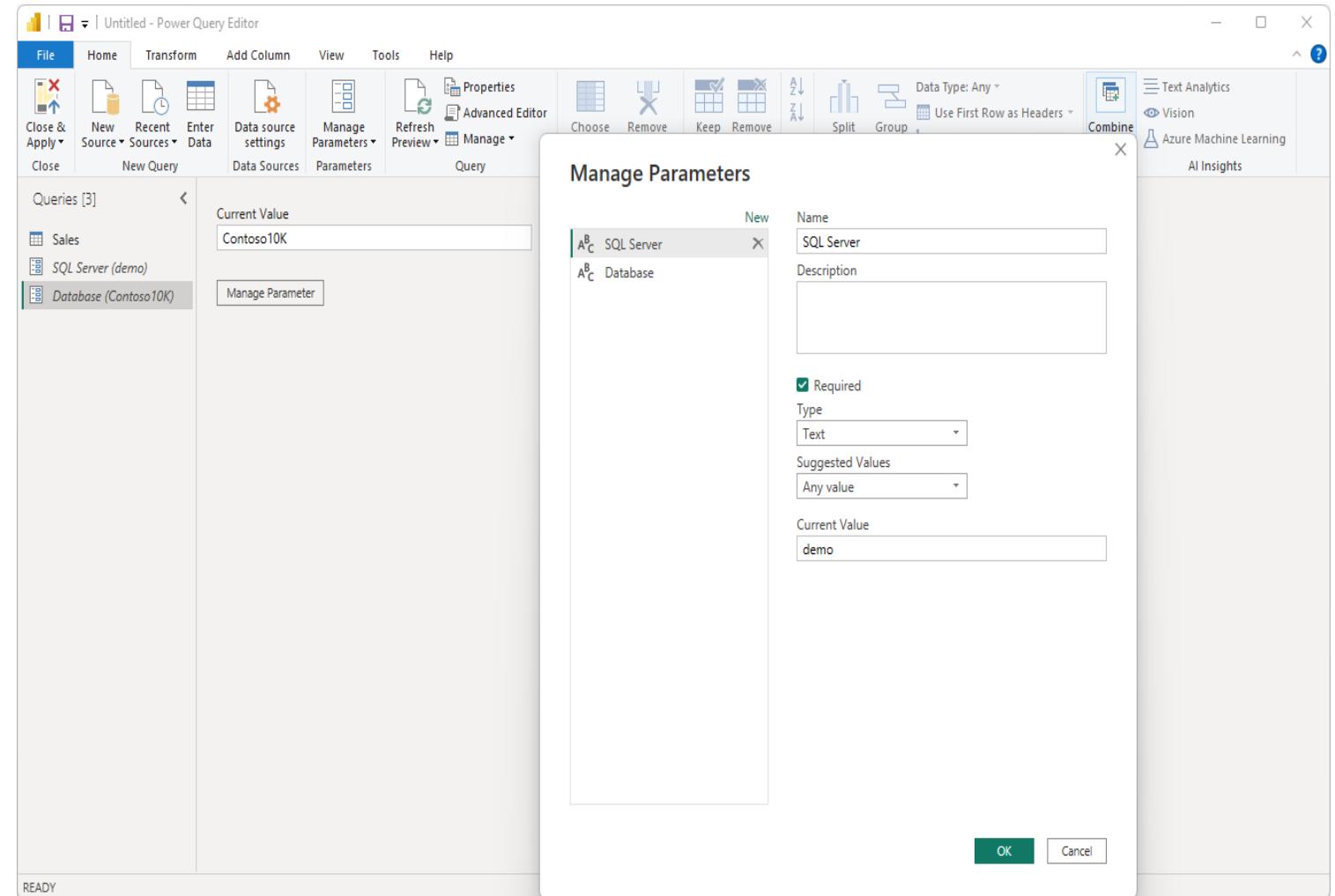
Query Folding

- 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

Power Query in Power BI Desktop

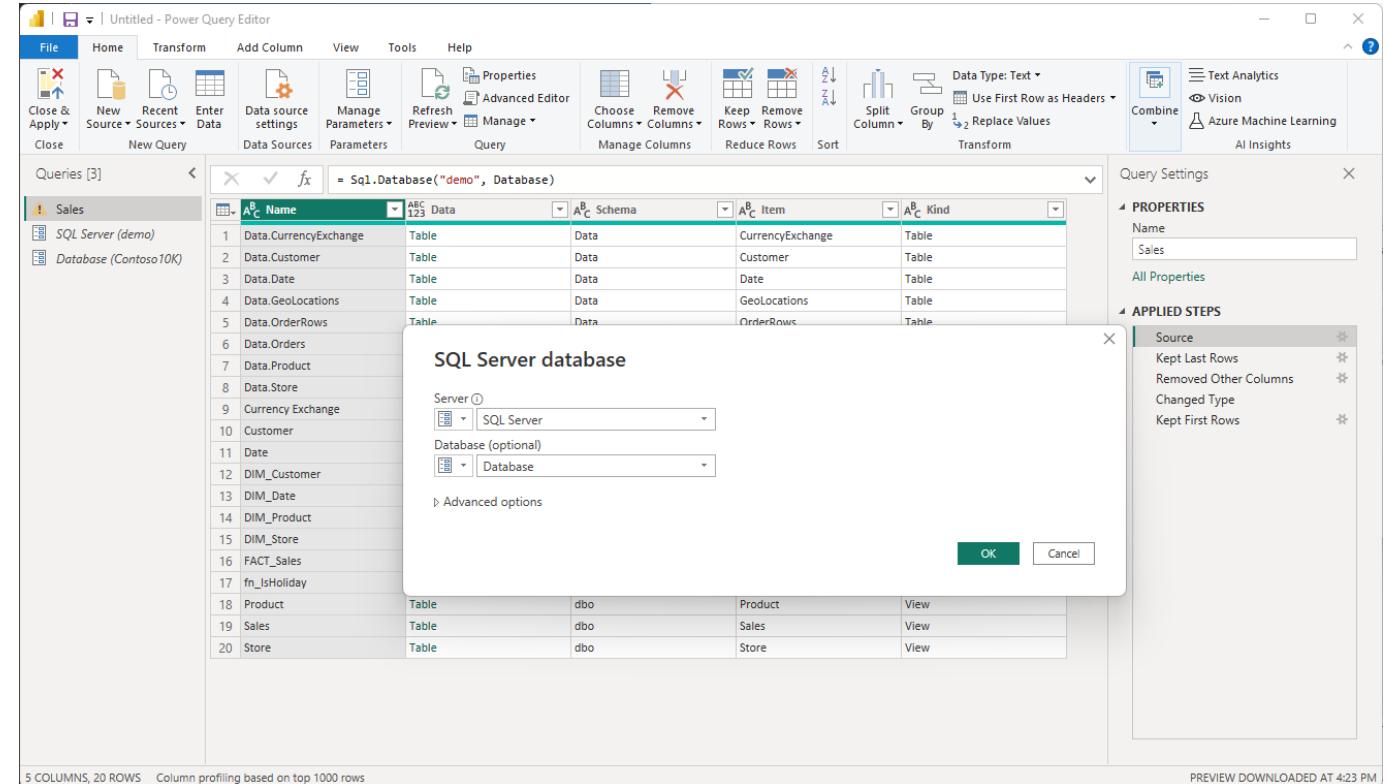
- New Experience with Icons like Power Query Online
- Projected to be released this year

Parameters



Creating parameters

Using Parameters



Using a parameter for the
Server and Database

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 Journey



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

Data Model

Date Tables

Turn off Auto Date Tables

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

Background Data

- Always allow data previews to download in the background
- Allow data previews to download in the background according to each file's setting
- Never allow data previews to download in the background

Parallel loading of tables

When you load data into Power BI (via import or DirectQuery), each data table is backed by a Power Query query. These queries are evaluated simultaneously instead of one-by-one, which can speed up the process. In certain situations, you might want to adjust the default number of simultaneous query evaluations and memory used. [Learn more](#)

Maximum number of simultaneous evaluations:

Maximum memory used per simultaneous evaluation (MB):

Time intelligence

Auto date/time for new files [Learn more](#)

Data Cache Management Options ⓘ

Currently used: 655 KB

Clear Cache

Maximum allowed (MB):

Restore Defaults

Q&A Cache Options ⓘ

Currently used: 1.43 MB

Clear Cache

OK **Cancel**

Hidden Date Tables

-
- >  Customer
 - >  Date
 - >  DateTableTemplate_a987aaf9-4324-4fff-9e6b-91eb917a6145
 -  Date
 - >  Date Hierarchy
 - 1²3 Day
 - A^BC Month
 - 1²3 MonthNo
 - A^BC Quarter
 - 1²3 QuarterNo
 - 1²3 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	

28	29	30	31			
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

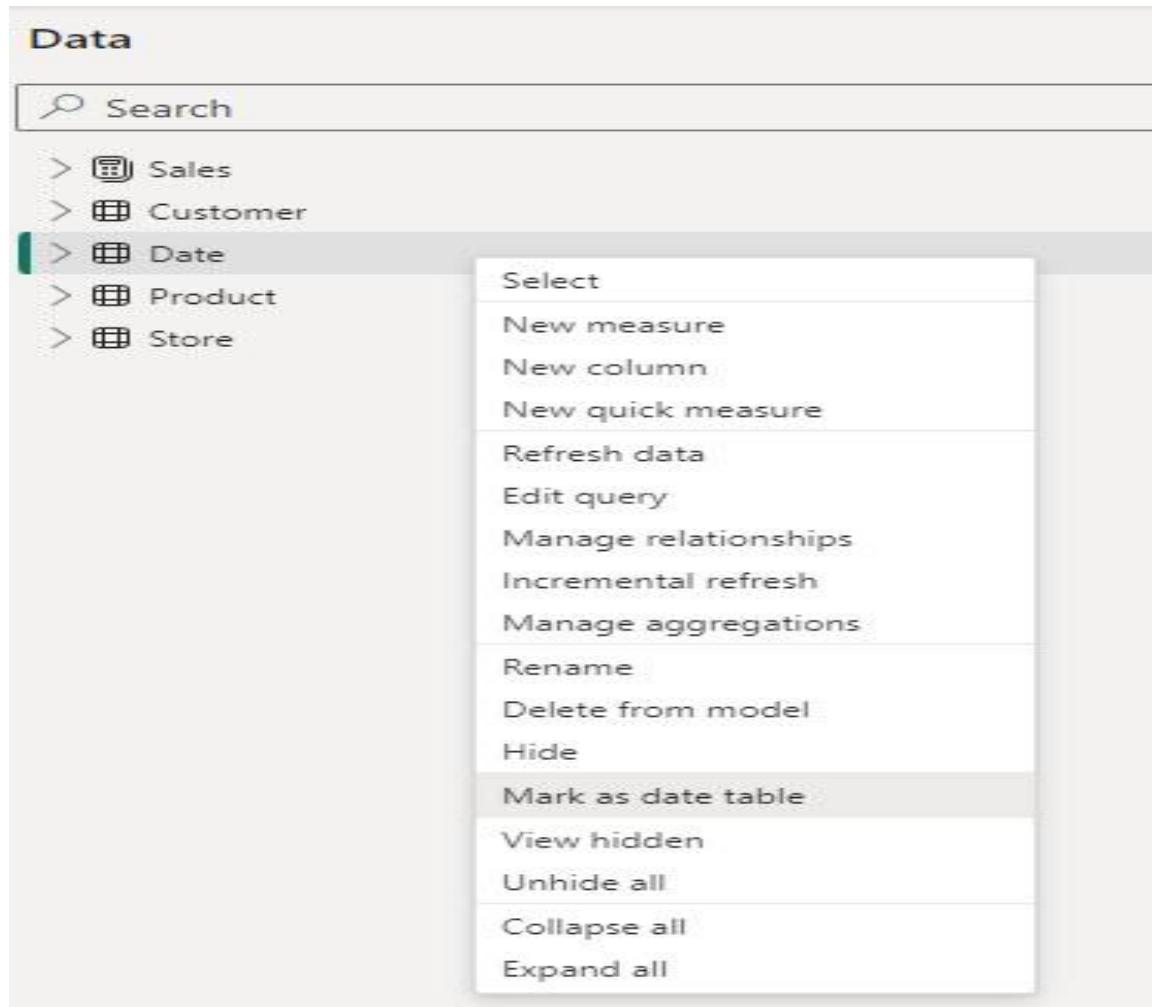
CalendarDate	DateKey	Year	Month	Month Name	MMM	Month Year	Day	Day Name	Day
Friday, December 3, 2021	44533	2021	12	December	Dec	Dec 2021	3	Friday	
Friday, December 10, 2021	44540	2021	12	December	Dec	Dec 2021	10	Friday	
Friday, December 17, 2021	44547	2021	12	December	Dec	Dec 2021	17	Friday	
Friday, December 24, 2021	44554	2021	12	December	Dec	Dec 2021	24	Friday	
Friday, December 31, 2021	44561	2021	12	December	Dec	Dec 2021	31	Friday	
Friday, April 22, 2022	44673	2022	4	April	Apr	Apr 2022	22	Friday	
Friday, April 29, 2022	44680	2022	4	April	Apr	Apr 2022	29	Friday	
Friday, May 6, 2022	44687	2022	5	May	May	May 2022	6	Friday	
Friday, May 13, 2022	44694	2022	5	May	May	May 2022	13	Friday	
Friday, May 20, 2022	44701	2022	5	May	May	May 2022	20	Friday	
Friday, May 27, 2022	44708	2022	5	May	May	May 2022	27	Friday	
Friday, June 3, 2022	44715	2022	6	June	Jun	Jun 2022	3	Friday	
Friday, June 10, 2022	44722	2022	6	June	Jun	Jun 2022	10	Friday	
Friday, June 17, 2022	44729	2022	6	June	Jun	Jun 2022	17	Friday	
Friday, June 24, 2022	44736	2022	6	June	Jun	Jun 2022	24	Friday	
Friday, July 1, 2022	44743	2022	7	July	Jul	Jul 2022	1	Friday	
Friday, July 8, 2022	44750	2022	7	July	Jul	Jul 2022	8	Friday	
Friday, July 15, 2022	44757	2022	7	July	Jul	Jul 2022	15	Friday	
Friday, July 22, 2022	44764	2022	7	July	Jul	Jul 2022	22	Friday	
Friday, July 29, 2022	44771	2022	7	July	Jul	Jul 2022	29	Friday	
Friday, August 5, 2022	44778	2022	8	August	Aug	Aug 2022	5	Friday	
Friday, August 12, 2022	44785	2022	8	August	Aug	Aug 2022	12	Friday	
Friday, August 19, 2022	44792	2022	8	August	Aug	Aug 2022	19	Friday	
Friday, August 26, 2022	44799	2022	8	August	Aug	Aug 2022	26	Friday	
Friday, September 2, 2022	44806	2022	9	September	Sep	Sep 2022	2	Friday	
Friday, September 9, 2022	44813	2022	9	September	Sep	Sep 2022	9	Friday	
Friday, September 16, 2022	44820	2022	9	September	Sep	Sep 2022	16	Friday	
Friday, September 23, 2022	44827	2022	9	September	Sep	Sep 2022	23	Friday	
Friday, September 30, 2022	44834	2022	9	September	Sep	Sep 2022	30	Friday	
Friday, October 7, 2022	44841	2022	10	October	Oct	Oct 2022	7	Friday	
Friday, October 14, 2022	44848	2022	10	October	Oct	Oct 2022	14	Friday	
Friday, October 21, 2022	44855	2022	10	October	Oct	Oct 2022	21	Friday	
Friday, October 28, 2022	44862	2022	10	October	Oct	Oct 2022	28	Friday	
Friday, November 4, 2022	44869	2022	11	November	Nov	Nov 2022	4	Friday	
Friday, November 11, 2022	44876	2022	11	November	Nov	Nov 2022	11	Friday	
Friday, November 18, 2022	44883	2022	11	November	Nov	Nov 2022	18	Friday	
Friday, November 25, 2022	44890	2022	11	November	Nov	Nov 2022	25	Friday	
Friday, December 2, 2022	44897	2022	12	December	Dec	Dec 2022	2	Friday	
Friday, December 9, 2022	44904	2022	12	December	Dec	Dec 2022	9	Friday	
Friday, December 16, 2022	44911	2022	12	December	Dec	Dec 2022	16	Friday	
Friday, December 23, 2022	44918	2022	12	December	Dec	Dec 2022	23	Friday	
Friday, December 30, 2022	44925	2022	12	December	Dec	Dec 2022	30	Friday	
Friday, April 21, 2023	45037	2023	4	April	Apr	Apr 2023	21	Friday	
Friday, April 28, 2023	45044	2023	4	April	Apr	Apr 2023	28	Friday	
Friday, May 5, 2023	45051	2023	5	May	May	May 2023	5	Friday	
Friday, May 12, 2023	45058	2023	5	May	May	May 2023	12	Friday	
Friday, May 19, 2023	45065	2023	5	May	May	May 2023	19	Friday	
Friday, May 26, 2023	45072	2023	5	May	May	May 2023	26	Friday	
Friday, June 2, 2023	45079	2023	6	June	Jun	Jun 2023	2	Friday	
Friday, June 9, 2023	45086	2023	6	June	Jun	Jun 2023	9	Friday	
Friday, June 16, 2023	45093	2023	6	June	Jun	Jun 2023	16	Friday	
Friday, June 23, 2023	45100	2023	6	June	Jun	Jun 2023	23	Friday	
Friday, June 30, 2023	45107	2023	6	June	Jun	Jun 2023	30	Friday	
Friday, July 7, 2023	45114	2023	7	July	Jul	Jul 2023	7	Friday	

Week Number	IsYTD	IsMTD	IsWTD	IsCurrentYTD	IsCurrentMTD	IsCurrent
49	0	1	0	0	0	0
50	0	1	0	0	0	0
51	0	1	0	0	0	0
52	0	0	0	0	0	0
1	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	1	0	0	0	0
20	0	1	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	1	0	0	0	0
24	0	1	0	0	0	0
25	0	1	0	0	0	0
26	0	0	0	0	0	0
27	0	1	0	0	0	0
28	0	1	0	0	0	0
29	0	1	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0
32	0	1	0	0	0	0
33	0	1	0	0	0	0
34	0	0	0	0	0	0
35	0	0	0	0	0	0
36	0	1	0	0	0	0
37	0	1	0	0	0	0
38	0	1	0	0	0	0
39	0	0	0	0	0	0
40	0	0	0	0	0	0
41	0	1	0	0	0	0
42	0	1	0	0	0	0
43	0	0	0	0	0	0
44	0	0	0	0	0	0
45	0	1	0	0	0	0
46	0	1	0	0	0	0
47	0	0	0	0	0	0
48	0	0	0	0	0	0
49	0	1	0	0	0	0
50	0	1	0	0	0	0
51	0	1	0	0	0	0
52	0	0	0	0	0	0
0	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	1	0	0	0	0
19	0	1	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	1	0	0	0	0
23	0	1	0	0	0	0
24	0	1	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	1	0	0	0	0

External Tools: Date Table

Mark as Date Table

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 ⓰

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

Default

One (disable parallel loading)

Custom

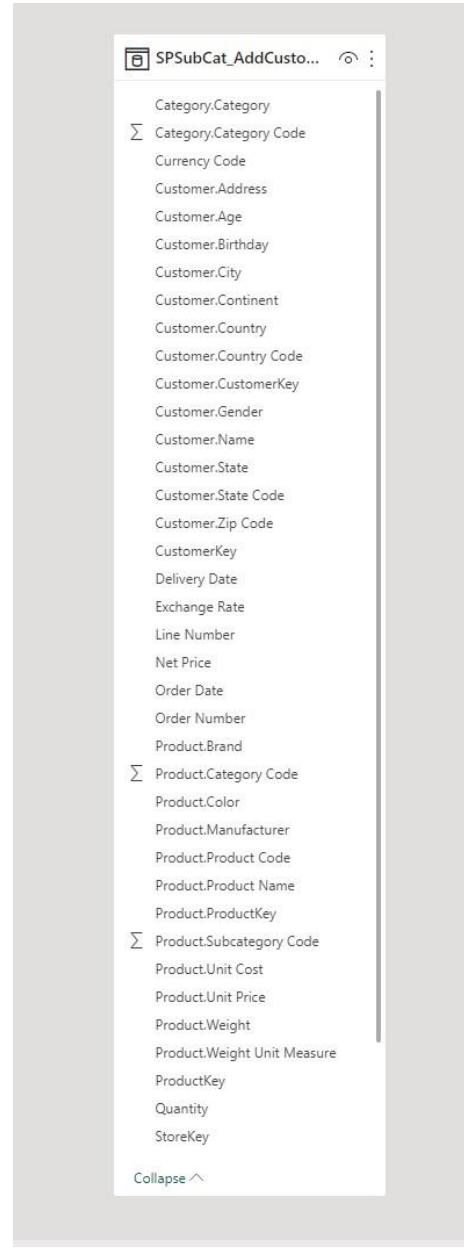
Q&A

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

Share your synonyms with everyone in your org

OK **Cancel**

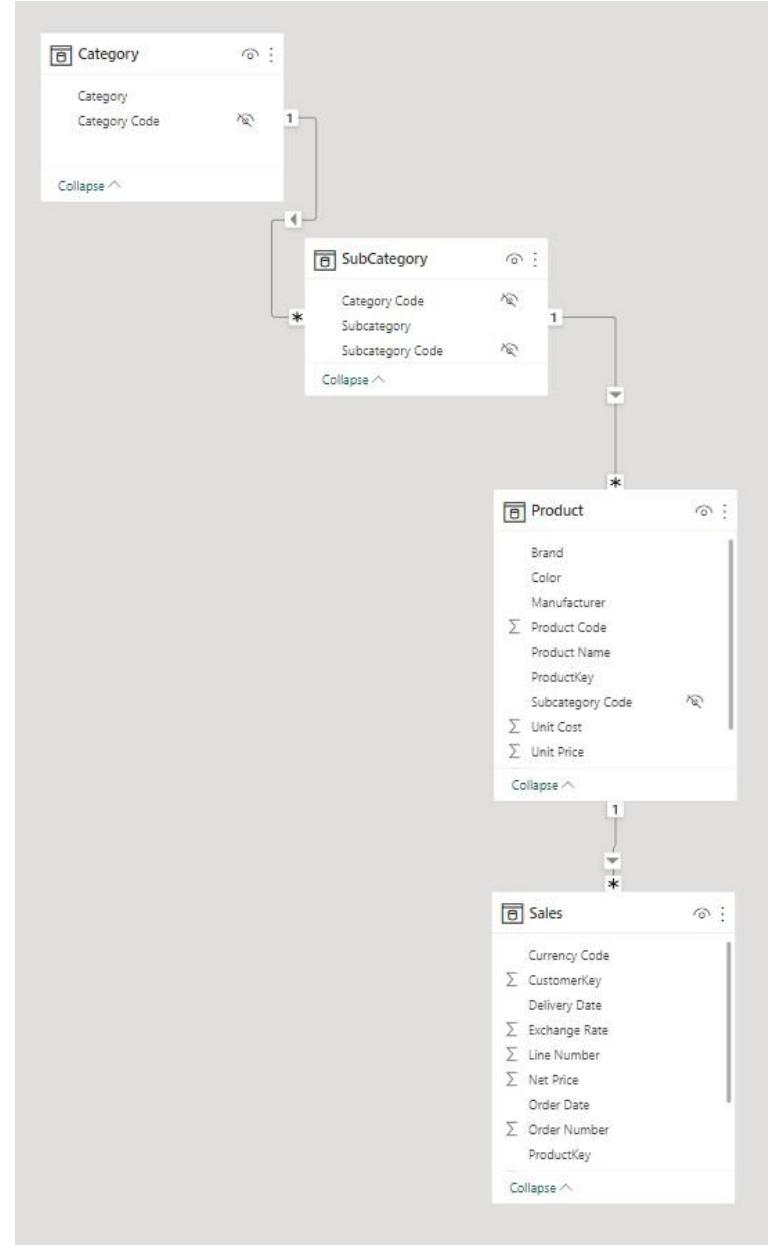
One big wide table (or tall)



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

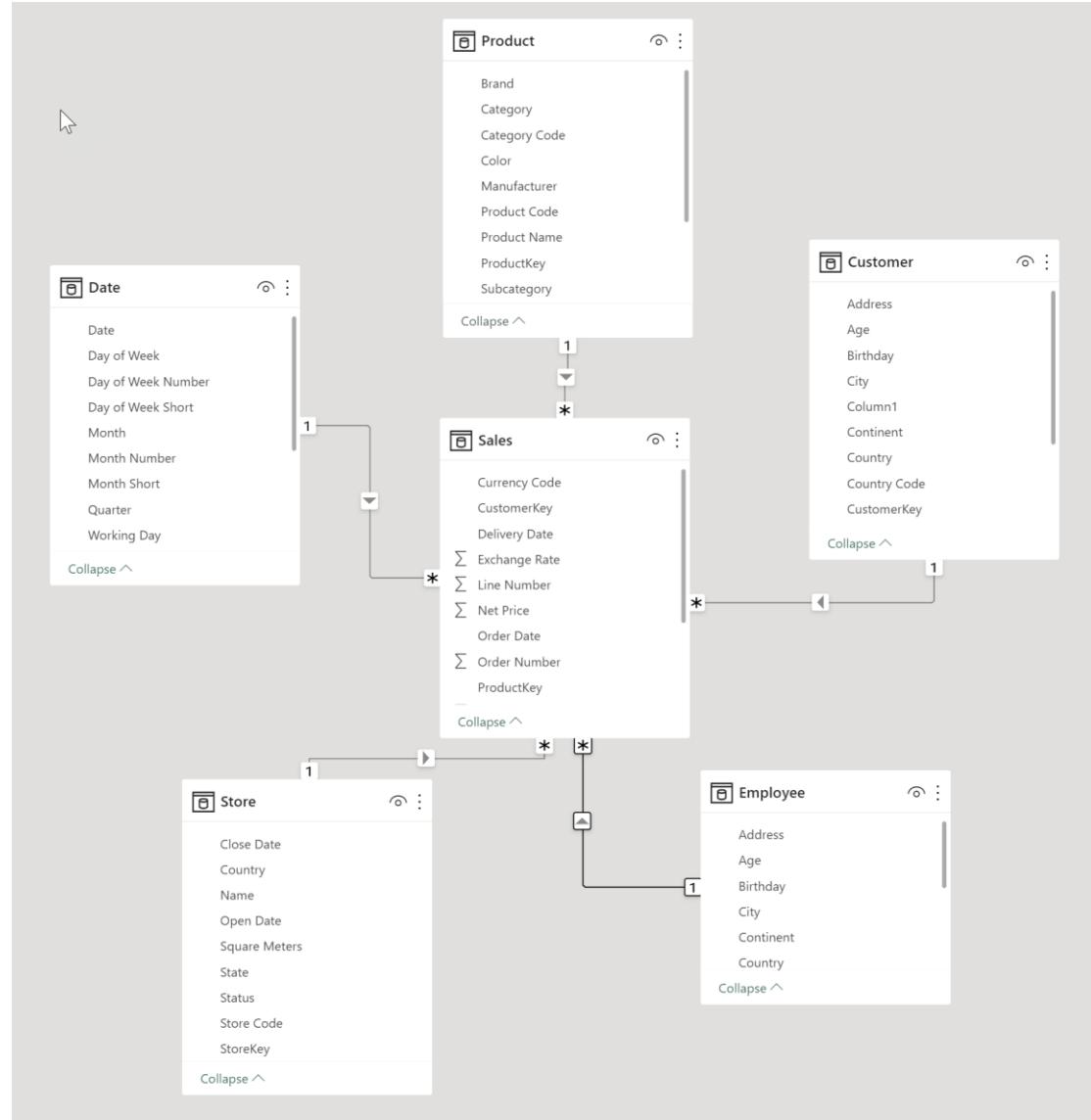
What is Snowflake Schema



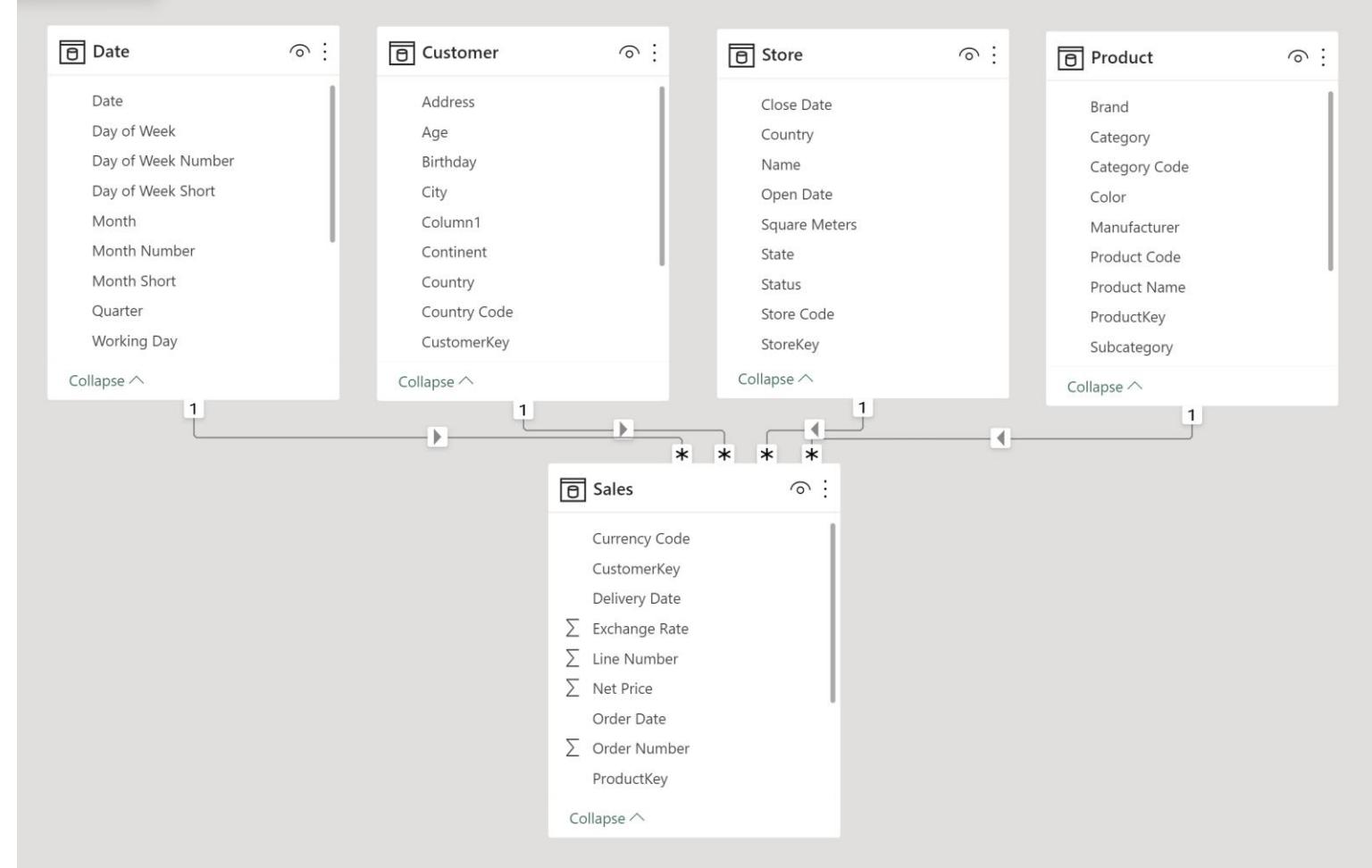
Other examples of Snowflake Schema

- 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

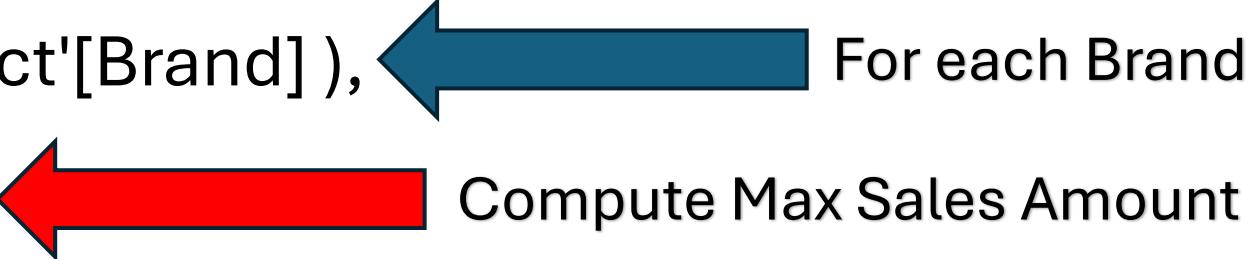
Expanded Tables

- When you apply a table filter it really is the expanded table.

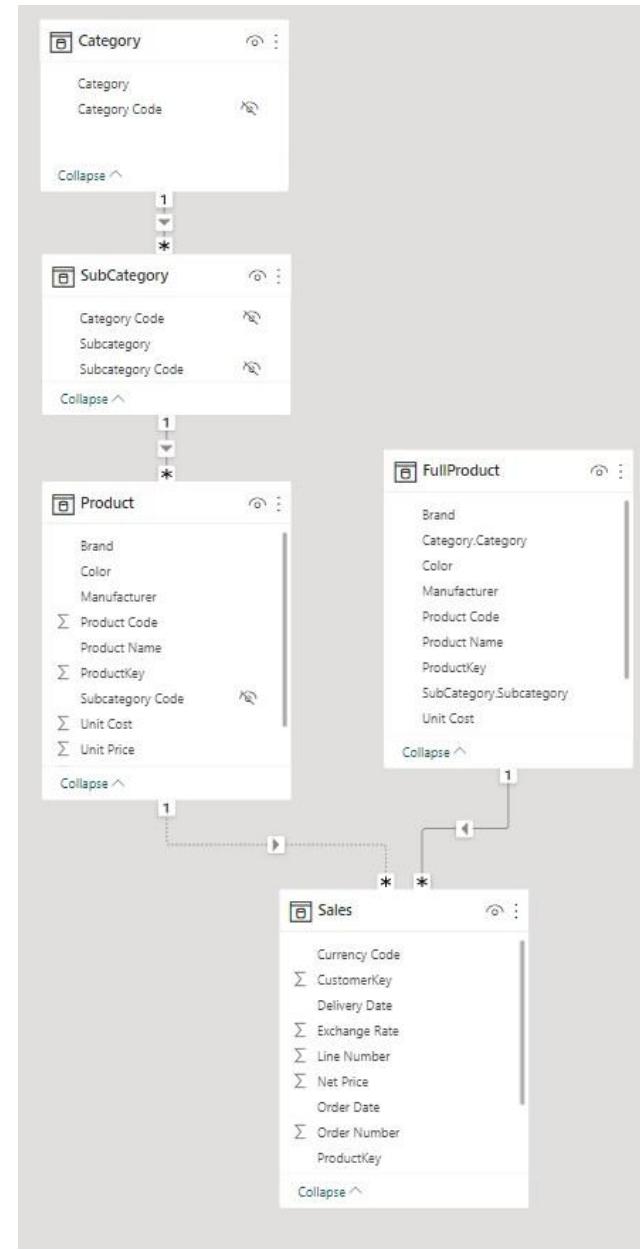
```
CALCULATE (
    COUNTROWS (Subcategory),
    Product
)
```

- It is the expanded table for Product. So only those subcategories that have a product
 - Can't reference base table

DAX can be easier

- MAXX (
VALUES ('Product'[Brand]),
[Sales Amount]
)
- For each Brand
- Compute Max Sales Amount

Snowflake to (Super) Star Schema



Hiding Columns

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

Implicit Measures

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

Calculated Columns

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

Our Journey



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	Action Buttons
[Error Prevention] Data columns must have a source c...	Data Columns	New rule... Clone rule Edit rule... Delete rule
[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	

Move to... OK Cancel

Bravo: Date Table

Bravo does
not support
AutoDate



Unsupported

Models with auto date/time option enabled are not supported.

[Disabling auto date-time in Power BI \(video\)](#)

Time Intelligence

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

Background Data

- Always allow data previews to download in the background
- Allow data previews to download in the background according to each file's setting
- Never allow data previews to download in the background

Parallel loading of tables

When you load data into Power BI (via import or DirectQuery), each data table is backed by a Power Query query. These queries are evaluated simultaneously instead of one-by-one, which can speed up the process. In certain situations, you might want to adjust the default number of simultaneous query evaluations and memory used. [Learn more](#)

Maximum number of simultaneous evaluations

Maximum memory used per simultaneous evaluation (MB)

Time intelligence

Auto date/time for new files [Learn more](#)

Data Cache Management Options ⓘ

Currently used: 655 KB

Clear Cache

Maximum allowed (MB):

Restore Defaults

Q&A Cache Options ⓘ

Currently used: 1.43 MB

Clear Cache

OK **Cancel**

Type of Calendar

Contoso10K > Manage Dates

Model Check



This model is compatible with the Manage Dates feature.

You can create new date tables without worrying about breaking measures or reports.

Calendar Interval Dates Holidays Time Intelligence

Choose a calendar template to apply to this model. Bravo will create the required tables or update them while keeping the existing relationships intact.

Template

Select **Monthly** for calendar based on different number of months. Set **Weekly** for 445-454-544-ISO calendars. Use **Custom** for flexible calendars of variable length. [Manage Templates](#)

Monthly - Fiscal

First Month of the Year

For Weekly ISO, set **January**.

March

Interval

Calendar Interval Dates Holidays Time Intelligence

Select a date interval for your model.

Date Interval



Choose how to determine the date interval of the model. Leave *First Year* and/or *Last Year* empty to use the automatic scan.

First Year

2018

Last Year

2024

Select Measures for Time Intelligence

StarSchema_Date > Manage Dates

Model Check

This model is compatible with the Manage Dates feature.

You can create new date tables without worrying about breaking measures or reports.

Calendar Interval Dates Holidays Time Intelligence

Create the most common Time Intelligence DAX functions available in your model.

Time Intelligence Functions

Target Measures Choose the measure used to generate the Time Intelligence functions.

Choose Measures... ▾

Search

TABLE \ COLUMN

Sales

The screenshot shows the 'Manage Dates' page in Power BI. On the left, a 'Model Check' box indicates compatibility with the feature, mentioning that new date tables can be created without breaking existing measures or reports. The main area is titled 'Time Intelligence Functions' and includes a 'Target Measures' section where the 'Sales' measure is selected. A red box highlights this selection. At the top, a navigation bar includes tabs for Calendar, Interval, Dates, Holidays, and Time Intelligence, with the latter being the active tab. Below the navigation, a general instruction is given to create common Time Intelligence DAX functions.

Time Intelligence Measures

←  StarSchema_Date > Manage Dates > Preview

Dates Time Intelligence



↳ To-date growth

- ↳ Sales Amount
 - MOMTD % Sales Amount
 - MOMTD Sales Amount
 - PMTD Sales Amount
 - PQTD Sales Amount
 - PYTD Sales Amount
 - QOQTD % Sales Amount
 - QOQTD Sales Amount
 - YOYTD % Sales Amount
 - YOYTD Sales Amount

↳ To-date total

- ↳ Sales Amount
 - MTD Sales Amount
 - QTD Sales Amount
 - YTD Sales Amount

Bravo: Optimize Model

Find unused columns, size of columns, and unique elements

Bravo Optimize Model



Your dataset is **1.48 MB** large and contains **66** columns, **56** of which are not referenced within the model.

Search Column

VPAX

COLUMN	TABLE	CARDINALITY	SIZE	WEIG...
Smaller columns...		10.44 K	1.08 MB	73%
Product Name	Product	2.52 K	145.70 KB	10%
ProductKey	Product	2.52 K	97.86 KB	6%
Product Code	Product	2.52 K	84.35 KB	6%
Date	Date	1.10 K	41.05 KB	3%
Birthday	Customer	511	40.34 KB	3%
		19.60 K	1.48 MB	100%

Unreferenced columns can generally be removed from the model to optimize performance. Before removing them, make sure you are not using these columns in any reports, which Bravo cannot determine.

The diagram illustrates the relationships between the columns identified in the table. It shows five boxes: 'Product Name' (yellow), 'ProductKey' (grey), 'Product Code' (yellow), 'Date' (grey), and 'Birthday' (yellow). 'Product Name' and 'ProductKey' are connected to 'Product'. 'Product Code' is connected to 'Product'. 'Date' and 'Birthday' are connected to 'Customer'.

Bravo Save VPAX

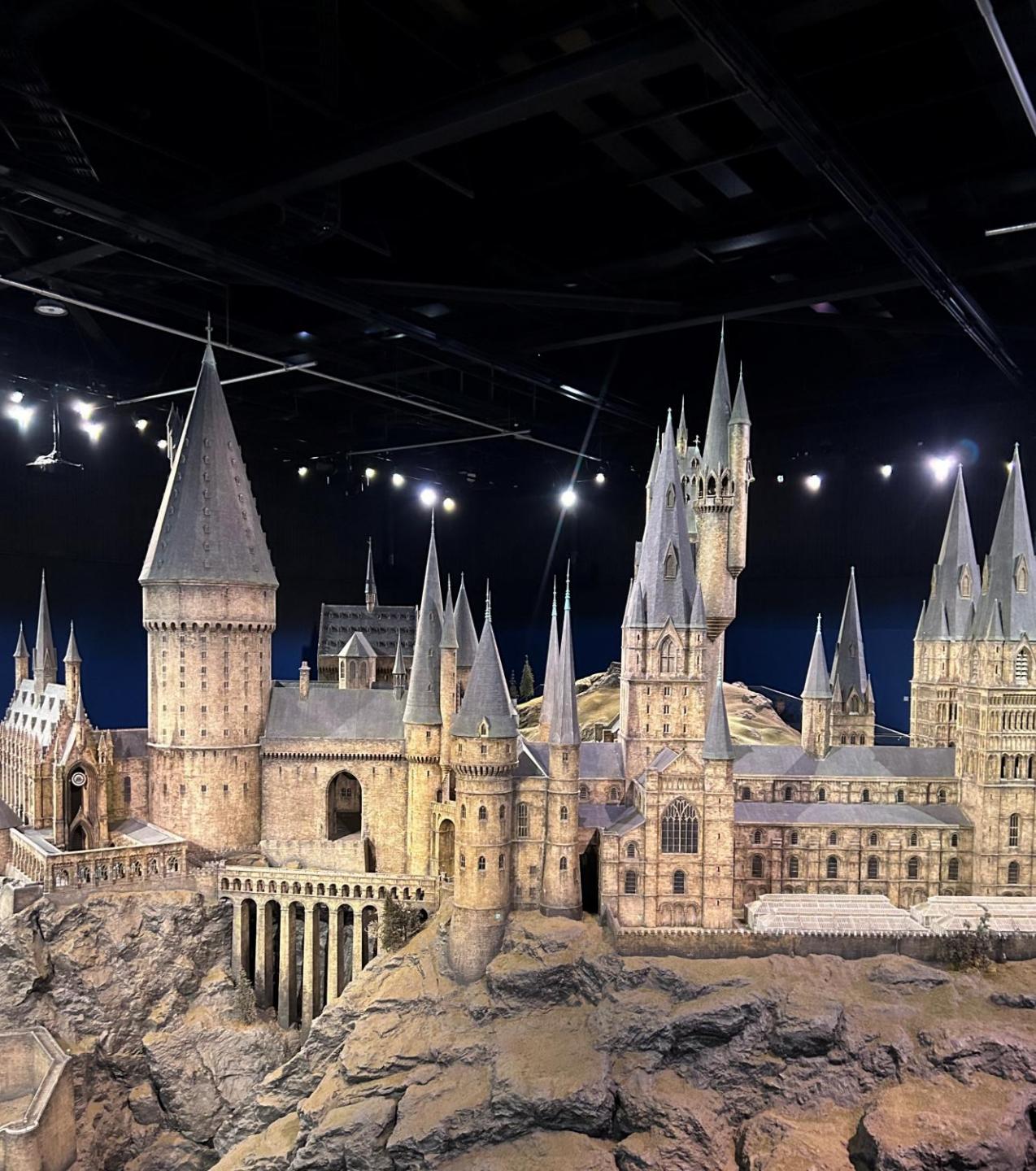


Your dataset is **1.48 MB** large and contains **66** columns, **56** of which are not referenced within the model.

The screenshot shows the StarSchema interface with a search bar, filter icons, and a red box highlighting the 'VPAX' button. Below is a table of dataset statistics:

TABLE \ COLUMN	CARDINALITY	SIZE	WEIG...
> Product	8.88 K	561.84 KB	37%
> Customer	3.47 K	346.31 KB	23%

Visualize Data



Model Goals

Our Journey



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

Conclusion

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

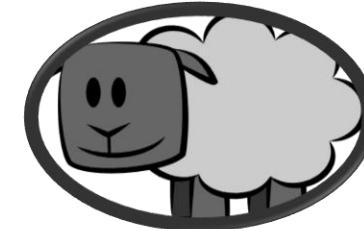
Resources

- 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/>

Thank you



The Dax Shepherd

Jason Romans
thedaxshepherd@gmail.com
www.thedaxshepherd.com

