

Power BI

Data Modeling and Shaping

Overview

The estimated time to complete this lab is 3 hours and 45 minutes.

In this lab you will learn how to upload multiple tables from a single data source. As well as create a new dimension for the model and enhance an existing dimension. You will learn how to create a fact table for a budget. You will also learn how to create parameters and a dynamic path to your data sources.

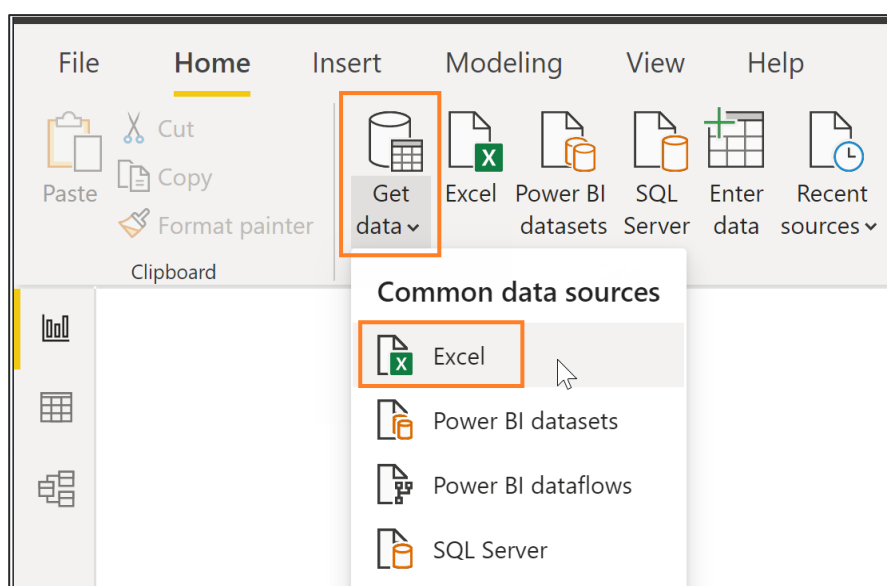
Data Shaping Labs

Lab 1a: Import multiple tables from a single source file

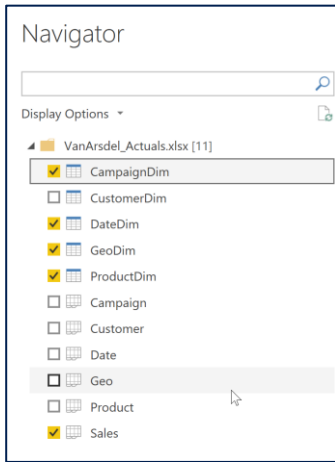
Task: Import multiple tables from a single excel source file.

The estimated time to complete this lab is 15 minutes.

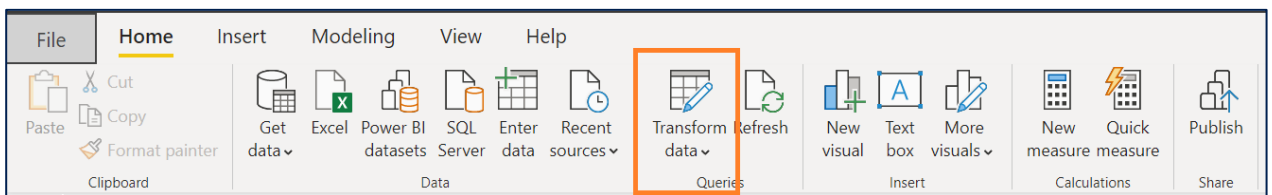
1. **GetData** > from Excel **C:\PowerBI_Adv_M\VansArsdel_Actuals.xlsx**



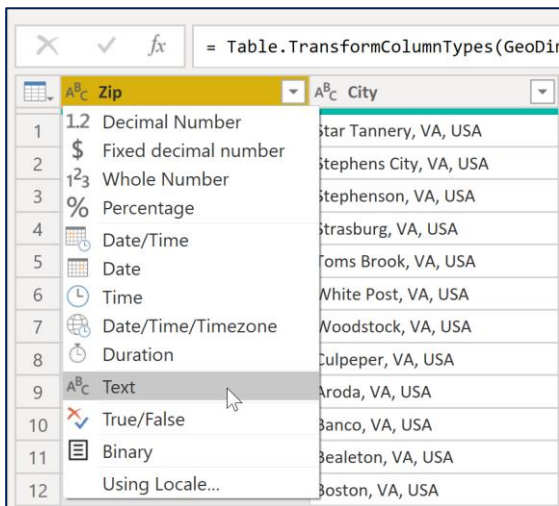
2. Select the following Tables (All but CustomerDim):
 - a. **CampaignDim**
 - b. **GeoDim**
 - c. **ProductDim**
 - d. **DateDim**
 - e. **Sales**



3. After tables are loaded click on **Transform Data**



4. In **GeoDim**, change the **Zip** data type to **Text**

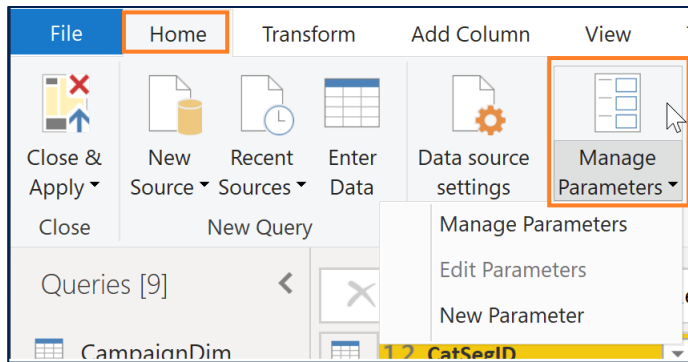


Lab 1b: Create lab parameters

Task: Create new parameters

The estimated time to complete this lab is 15 minutes.

1. Click on **Transform Data**
2. From the **Home Ribbon > Manage Parameters**



3. Create a new **Parameter**
 - a. Parameter Name: **Path**
 - b. Type: **Text**
 - c. Current Value = **C:\Power BI_Adv_M**

A screenshot of the 'Manage Parameters' dialog box in Power BI. The dialog has a title bar 'Manage Parameters'. On the left, there is a list of parameters with one entry 'Path' selected. On the right, the configuration for the 'Path' parameter is shown. The 'Name' field contains 'Path'. The 'Description' field is empty. The 'Required' checkbox is checked. The 'Type' dropdown is set to 'Text'. The 'Suggested Values' dropdown is set to 'Any value'. The 'Current Value' field contains 'C:\Power BI_Adv_M\'. At the bottom right, there are 'OK' and 'Cancel' buttons.

4. Create a new **Parameter**

- a. Parameter Name: **Actuals_File**
- b. Type: **Text**
- c. Current Value = **VanArsdel_Actuals.xlsx**

The screenshot shows the 'Manage Parameters' dialog box. On the left, there is a list of parameters with columns 'Name' and 'Path'. The parameter 'Actuals_File' is selected. On the right, the configuration for 'Actuals_File' is shown. The 'Name' field contains 'Actuals_File'. The 'Description' field is empty. The 'Required' checkbox is checked. The 'Type' dropdown is set to 'Text'. The 'Suggested Values' dropdown is set to 'Any value'. The 'Current Value' field contains 'VanArsdel_Actuals.xlsx'. At the bottom right, there are 'OK' and 'Cancel' buttons.

5. Create a new **Parameter**

- a. Parameter Name: **Budget_File**
- b. Type: **Text**
- c. Current Value = **VanArsdel_Budget.csv**

Manage Parameters

New

Path
Actuals_File
Budget_File

Name
Budget_File

Description

☒ Required

Type
Text

Suggested Values
Any value

Current Value
VanArsdel_Budget.csv

OK Cancel

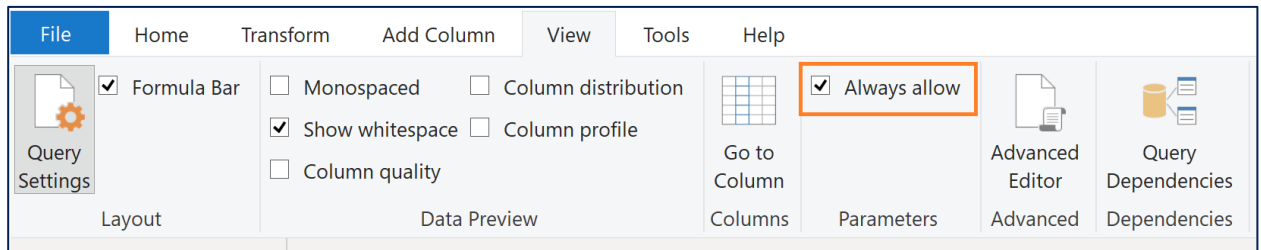
6. Update the text files to ensure the parameter names are consistent

Lab 2a: Create CatSegDim

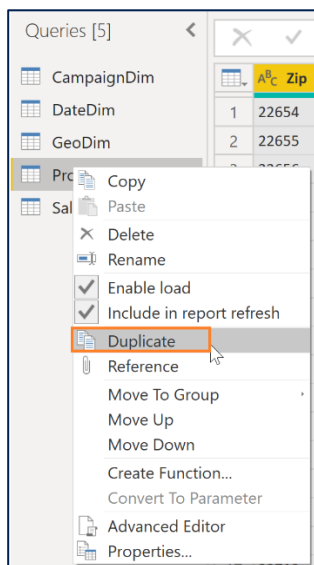
Task: Create a dimension using Category and Segment attributes from the Product dimension

The estimated time to complete this lab is 30 minutes.

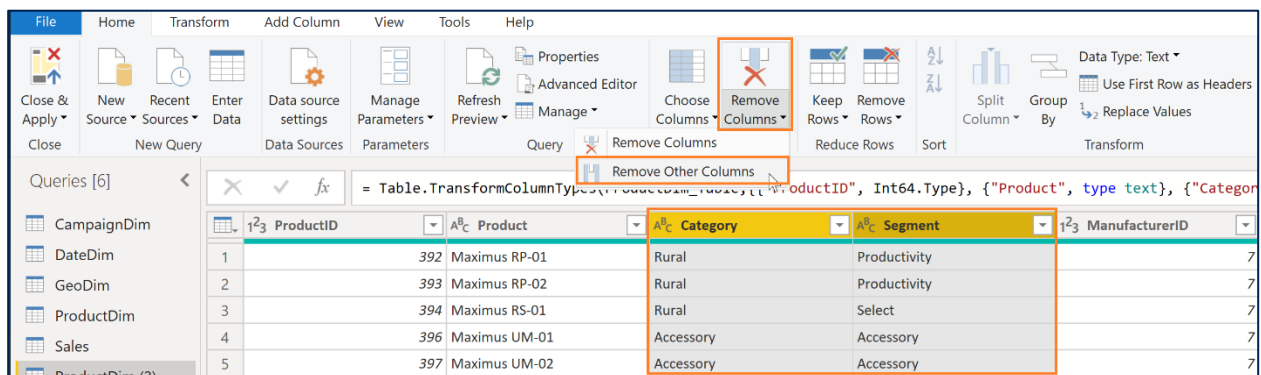
1. From the **View** ribbon check "**Always Allow**"



2. **Duplicate** the **ProductDim** query



3. Highlight **Category** and **Segment**, and **Remove other columns**



4. Highlight **Category** and **Segment**, and **Remove Duplicates**

	ABC Category	ABC Segment
1	Rural	Productivity
2	Rural	Productivity
3	Rural	Select
4	Accessory	Accessory
5	Accessory	Accessory
6	Accessory	Accessory
7	Accessory	Accessory
8	Accessory	Accessory
9	Accessory	Accessory
10	Accessory	Accessory
11	Accessory	Accessory
12	Accessory	Accessory
13	Accessory	Accessory

5. **Add Column**> Add **Index** Column starting at 1, with column name **CatSegID**

File	Home	Transform	Add Column	View	Tools	Help
Column From Examples	Custom Column	Invoke Custom Function	Conditional Column	Merge Columns	Format	Statistics
		General	Index Column	Extract	Parse	
			From 0	From Text		
			From 1			
			Custom...			

	ABC Category	ABC Segment
1	Rural	Productivity
2	Rural	Select
3	Accessory	Accessory
4	Urban	Moderation
5	Urban	Regular
6	Urban	Extreme
7	Mix	All Season
8	Mix	Productivity
9	Youth	Youth
10	Urban	Convenience

6. Reorder Columns: **CatSegID**, **Category**, **Segment**

1.2 CatSegID	A ^B _C Category	A ^B _C Segment
	1 Rural	Productivity
	2 Rural	Select
	3 Accessory	Accessory
	4 Urban	Moderation
	5 Urban	Regular
	6 Urban	Extreme
	7 Mix	All Season
	8 Mix	Productivity
	9 Youth	Youth
	10 Urban	Convenience

7. Rename the query "**CatSegDim**"

Queries [6]

fx

= Table.RenameColumns(#"Reordered Columns",{{"CatSegID", "Ca

1.2 CatSegDim

A^B_C Category

A^B_C Segment

1

1 Rural

Productivity

2

2 Rural

Select

3

3 Accessory

Accessory

4

4 Urban

Moderation

5

5 Urban

Regular

6

6 Urban

Extreme

7

7 Mix

All Season

8

8 Mix

Productivity

9

9 Youth

Youth

10

10 Urban

Convenience

Lab 2b: Update Product dimension

Task: Update the Product dimension

The estimated time to complete this lab is 15 minutes.

1. Select the **ProductDim** query
2. From **Home Ribbon > Merge Queries > Select CatSegDim**
 - a. From **ProductDim**, highlight **Category** and **Segment**
 - b. From **CatSegDim**, highlight **Category** and **Segment**
 - c. Note the Join Kinds available, and leave **Left Outer**

Merge

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

ProductDim

ProductID	Product	Category 1	Segment 2	ManufacturerID	Manufacturer	Unit Cost	Unit Price
392	Maximus RP-01	Rural	Productivity	7	VanArsdel	37.2710625	51.0562
393	Maximus RP-02	Rural	Productivity	7	VanArsdel	37.2710625	51.0562
394	Maximus RS-01	Rural	Select	7	VanArsdel	119.7617925	164.0572
396	Maximus UM-01	Accessory	Accessory	7	VanArsdel	66.2830875	90.7987

CatSegDim

CatSegDim	Category 1	Segment 2
1	Rural	Productivity
2	Rural	Select
3	Accessory	Accessory
4	Urban	Moderation
5	Urban	Regular

Join Kind

Left Outer (all from first, matching from second)

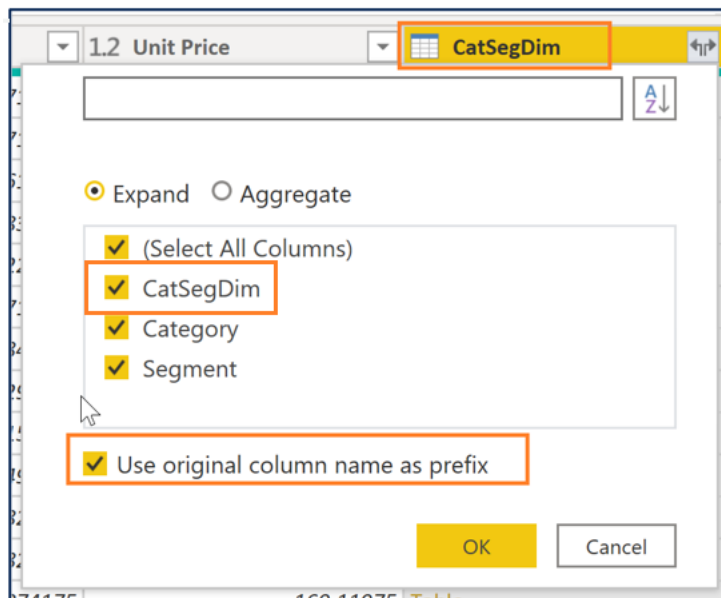
☐ Use fuzzy matching to perform the merge

> Fuzzy matching options

✓ The selection matches 212 of 212 rows from the first table.

OK Cancel

- d. Expand the **NewColumn** > Select **CatSegID** and deselect **“Use Original column name as prefix”**



3. Remove columns: **Manufacturer ID**, and **Manufacturer**.
 - a. **Hint:** There is only one manufacturer name and one manufacturer ID, so we don't need this information!
4. Reorder columns: **ProductID, Product, CatSegID, Unit Price, Unit Cost, Category, Segment**

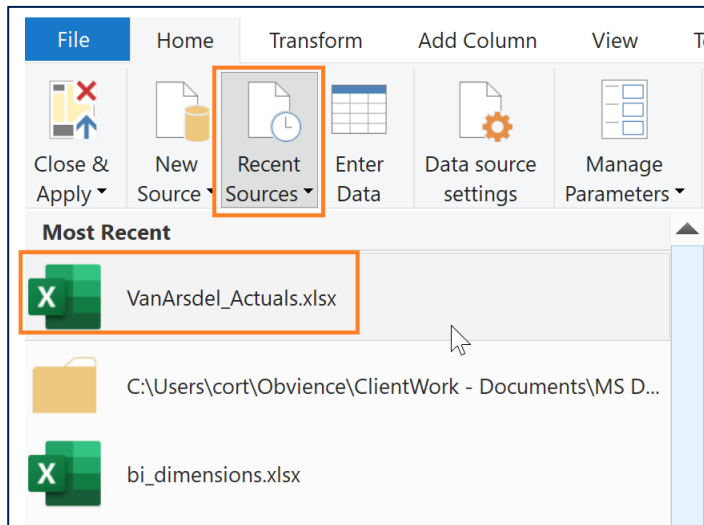
	1.2 ProductID	1.2 CatSegID	A ^B C Product	1.2 Unit Price	1.2 Unit Cost
1	392	1	Maximus RP-01	51.05625	37.2710625
2	393	1	Maximus RP-02	51.05625	37.2710625
3	394	2	Maximus RS-01	164.05725	119.7617925
4	396	3	Maximus UM-01	90.79875	66.2830875
5	397	3	Maximus UM-02	149.61975	109.2224175

Lab 2c: Create Customer dimension

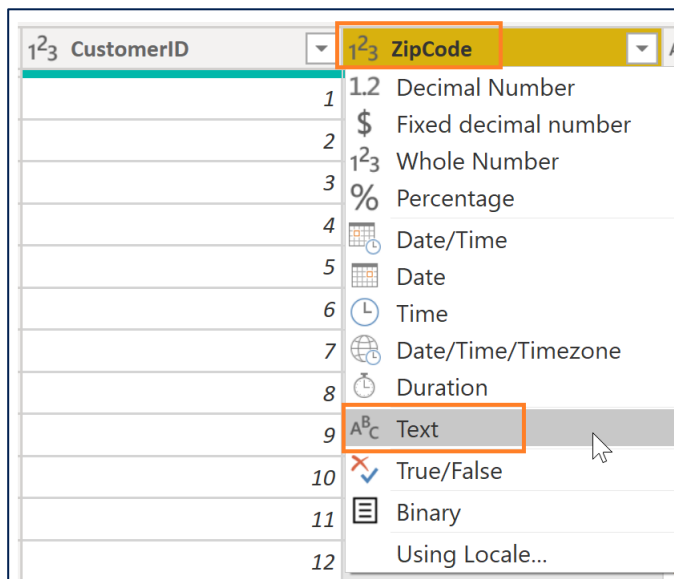
Task: Create a Customer dimension

The estimated time to complete this lab is 30 minutes.

1. Use **Recent Sources** to get **CustomerDim** from Excel



2. Change the **ZipCode** column data type to **Text**



3. Split **Email Name** by Delimiter Custom": " (colon space)

Split Column by Delimiter

Specify the delimiter used to split the text column.

Select or enter delimiter

--Custom--

:

Split at

☐ Left-most delimiter

☐ Right-most delimiter

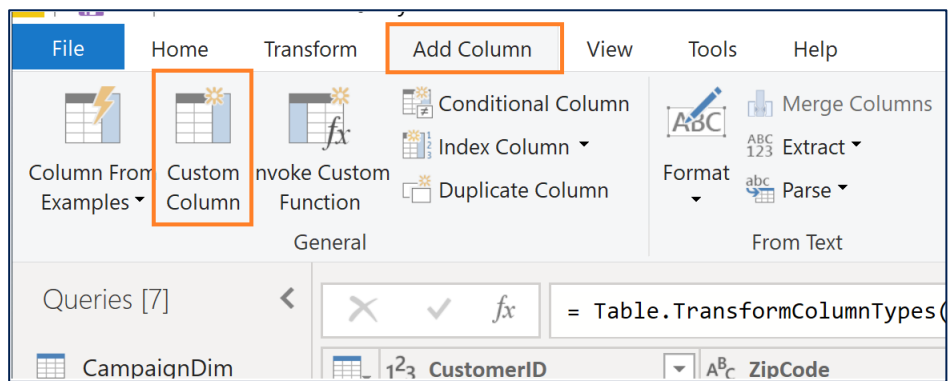
☒ Each occurrence of the delimiter

> Advanced options

OK Cancel

4. **Add Column** to find Text.PositionOf() the comma

a. **Add Column > Custom Column**



b. Name = **"Separator"**

c. Formula = **Text.PositionOf([Email Name.2],",")**

Custom Column

Add a column that is computed from the other columns.

New column name

Separator

Custom column formula

= Text.PositionOf([Email Name.2],",")

Available columns

CustomerID

ZipCode

Email Name.1

Email Name.2

<< Insert

[Learn about Power BI Desktop formulas](#)

✓ No syntax errors have been detected.

OK Cancel

5. Use position of comma to split **Last Name** and **First Name**

6. **Add Columns** for **First Name, Last Name and Full Name**

- a. Last Name = ***Text.Start([Email Name.2], [Separator])***
- b. First Name = ***Text.Range([Email Name.2],[Separator]+2)***
- c. Full Name = ***[First Name] & " " & [Last Name]***

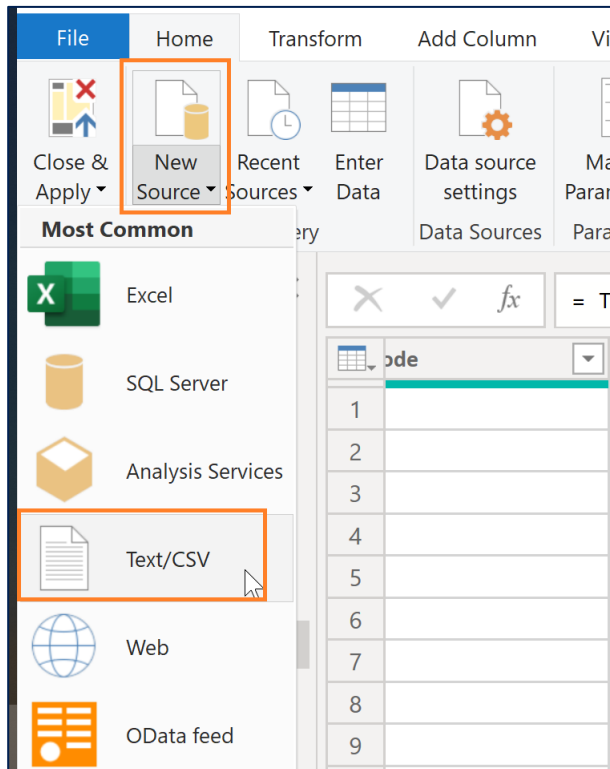
7. Remove **Separator** column

Lab 2d: Create Budget Fact table

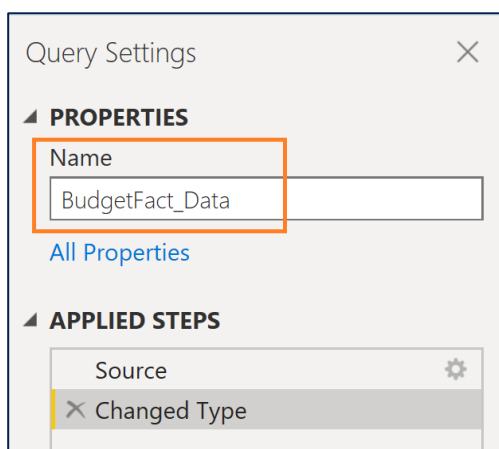
Task: Create Budget fact table

The estimated time to complete this lab is 45 minutes.

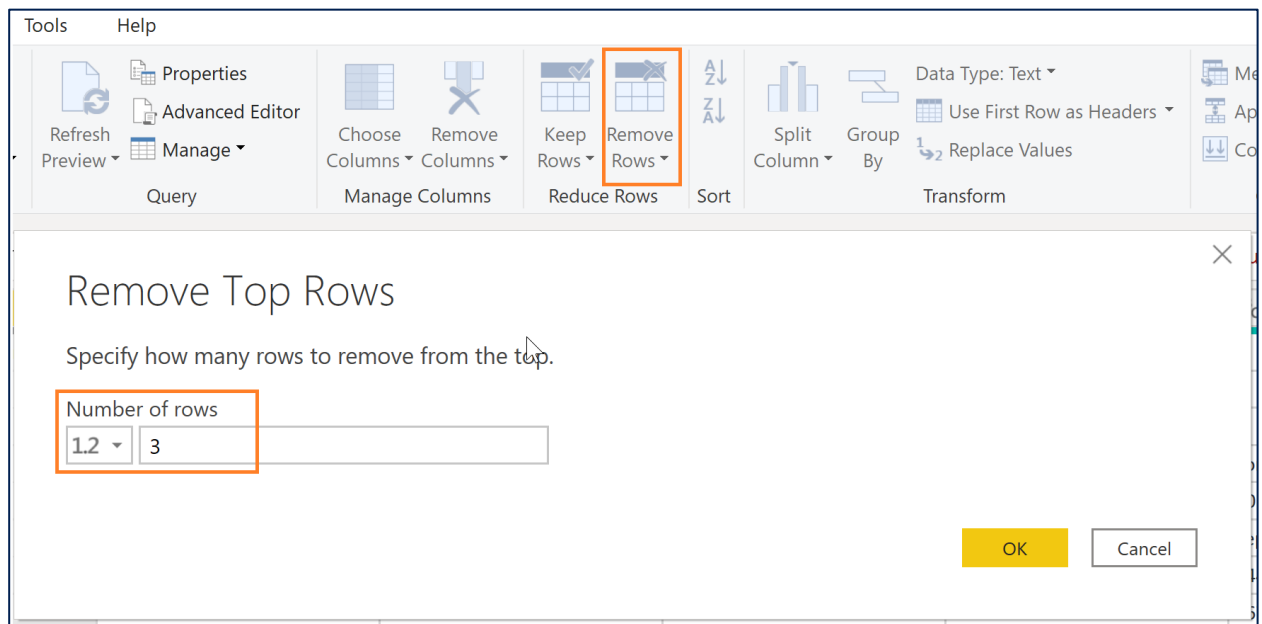
1. Import CSV document “**C:/Power BI Adv_M/VanArsdel_Budget.csv**”



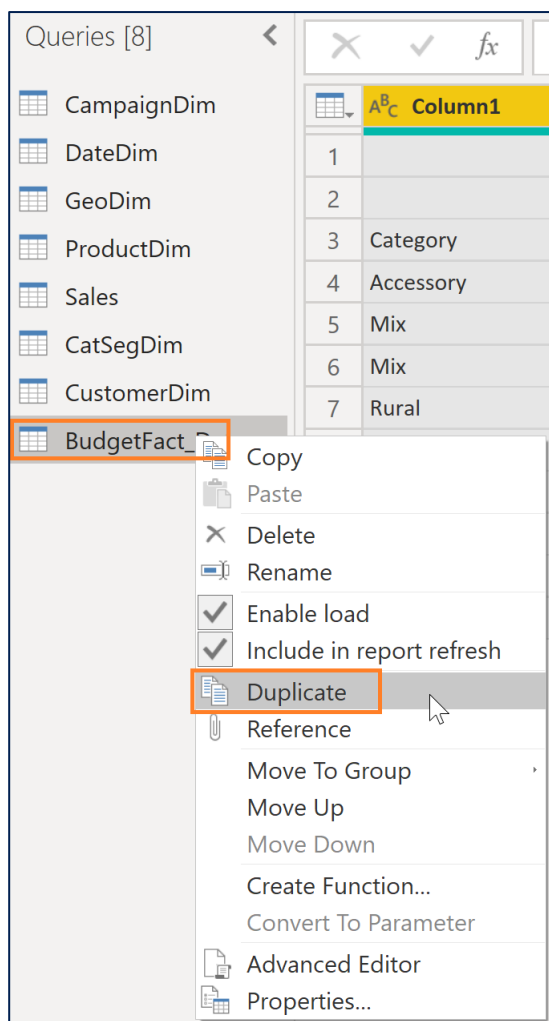
2. Rename query from **VanArsdel_Budget** to **BudgetFact_Data**



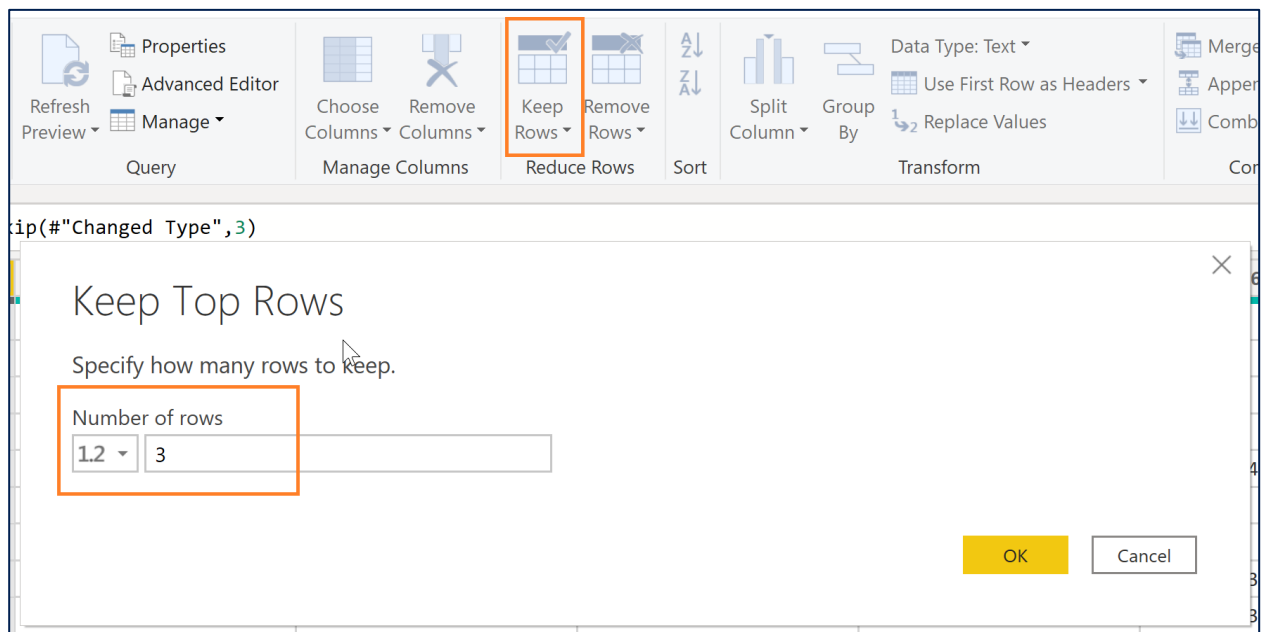
3. **Remove Rows** > Remove Top Rows, enter 3 (to remove the first 3 rows)



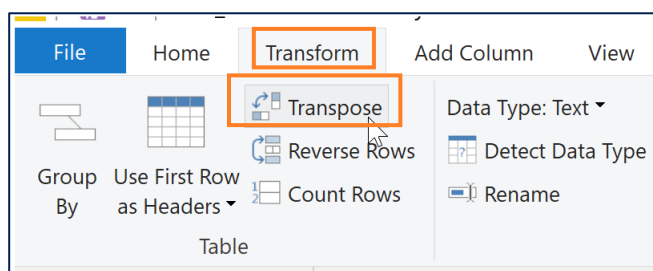
4. **Duplicate** query **BudgetFact_Data** rename to "**BudgetFact**"



5. **Keep Rows** > Keep Top Rows, enter 3 (to keep the first three rows)



6. **Transform** > **Transpose**



7. **Add Column** to combine month and year into a date a
- a. Add Column > **Custom Column**
 - b. Name = "**Budget Month**"
 - c. Formula = ***Date.From ([Column3] & [Column2])***

Custom Column

Add a column that is computed from the other columns.

New column name

Custom column formula ⓘ

Available columns
 Column1
 Column2
 Column3

< < Insert

[Learn about Power BI Desktop formulas](#)

✓ No syntax errors have been detected.

OK Cancel

8. **Add Column** to combine Month and Scenario

- d. Add Column > **Custom Column**
- e. Name = **"FullyCombinedHeader"**
- f. Formula = ***if Text.Length([Column3]) > 3 then [Column3] else [Column1] & "~" & Date.ToText([Budget Month], "M/D/YYYY")***
- g. **Hint:** Day did not come through correctly, as it is case sensitive. Update to "MM/dd/yy"

Custom Column

Add a column that is computed from the other columns.

New column name

Custom column formula ⓘ

Available columns
 Column1
 Column2
 Column3
 Budget Month

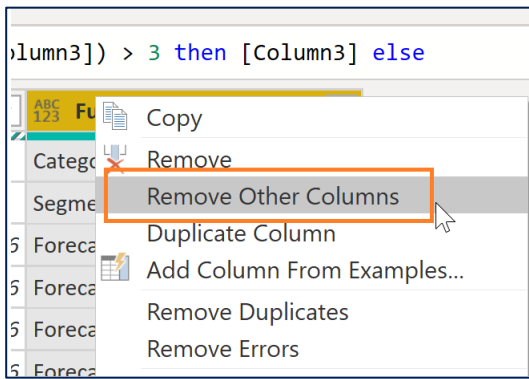
< < Insert

[Learn about Power BI Desktop formulas](#)

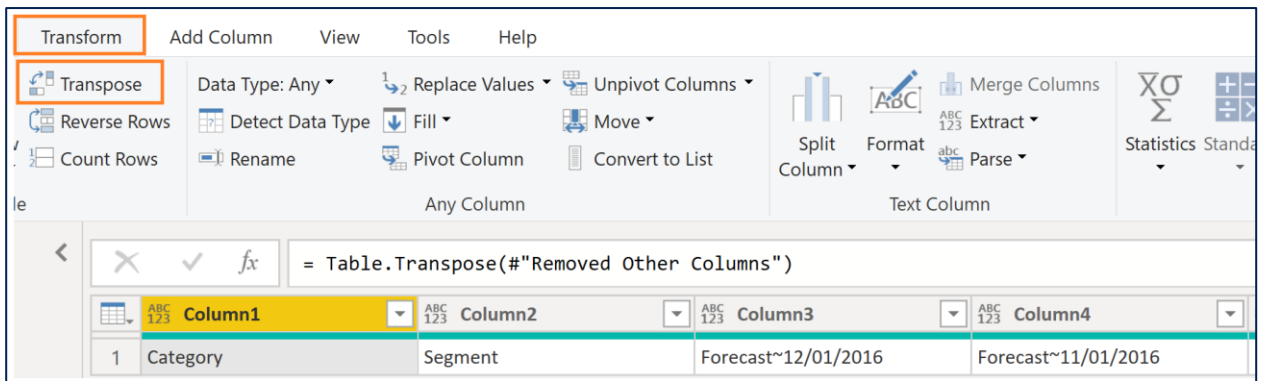
✓ No syntax errors have been detected.

OK Cancel

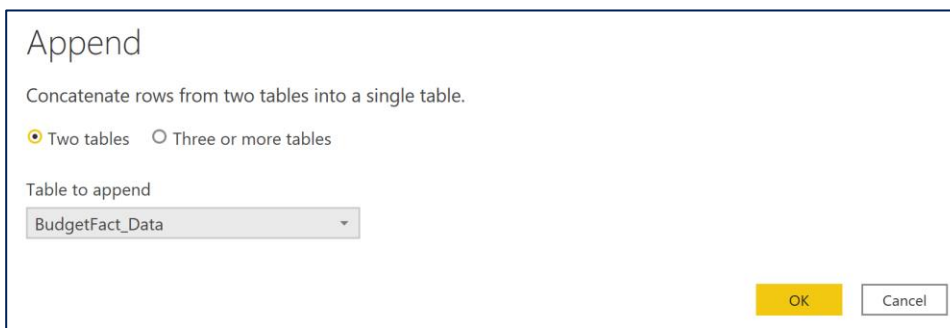
9. **Remove all** columns except for **FullyCombinedHeader**



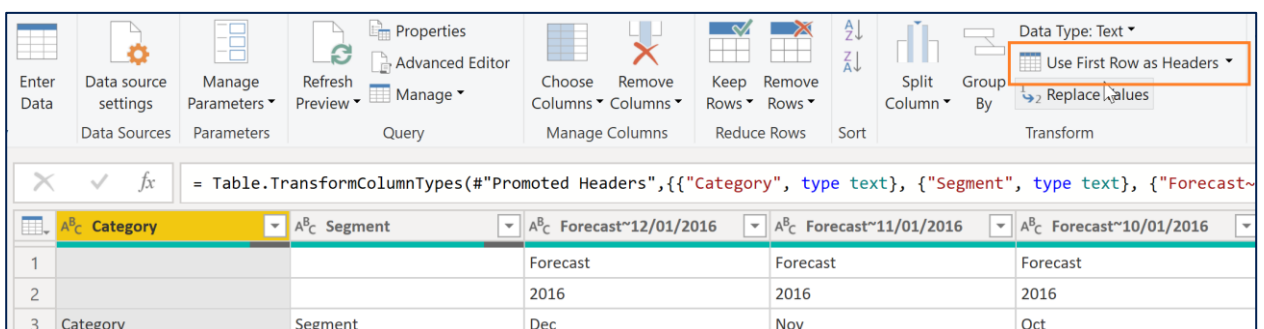
10. **Transform**> **Transpose** to transpose back to wide



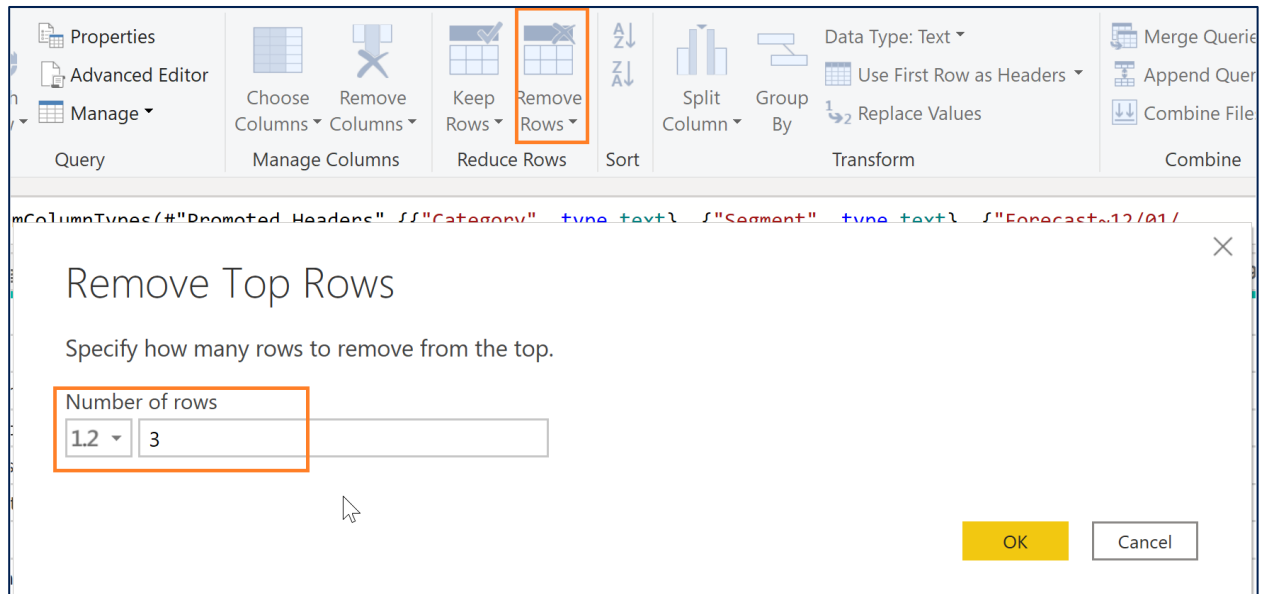
11. **Append** query **BudgetFact_Data**



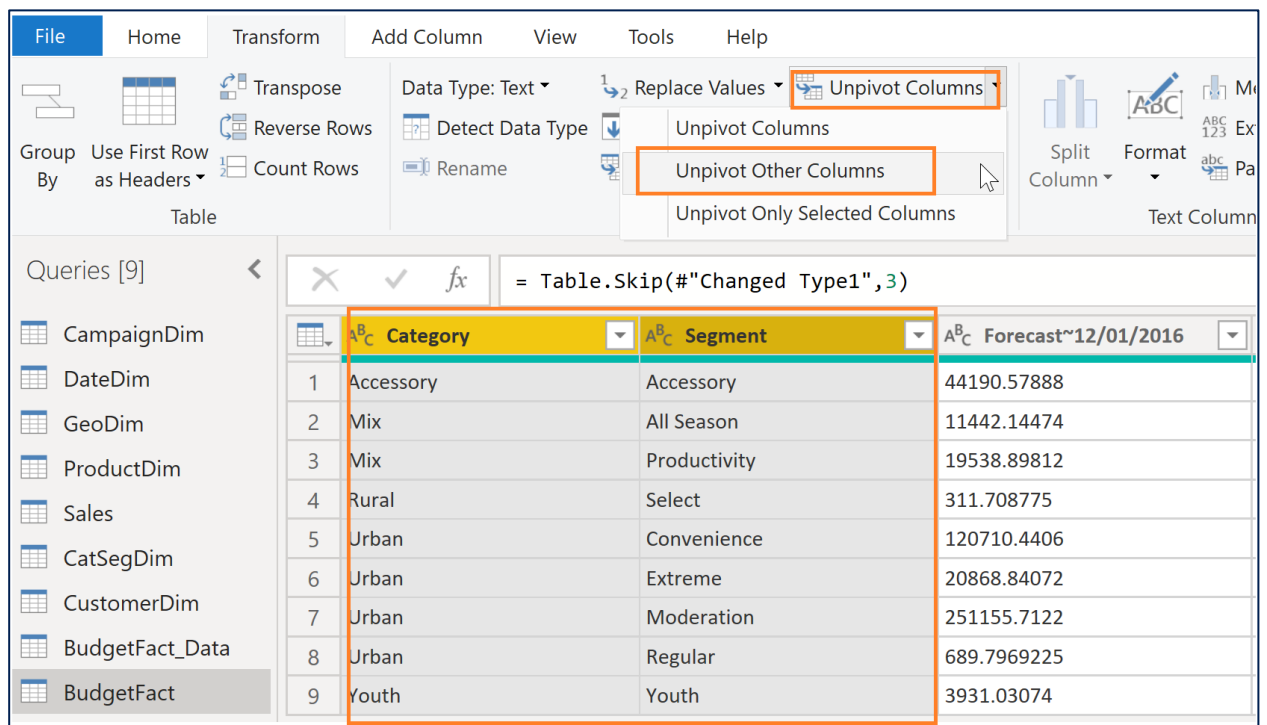
12. **Use First Row as Header** to promote the newly fixed header row



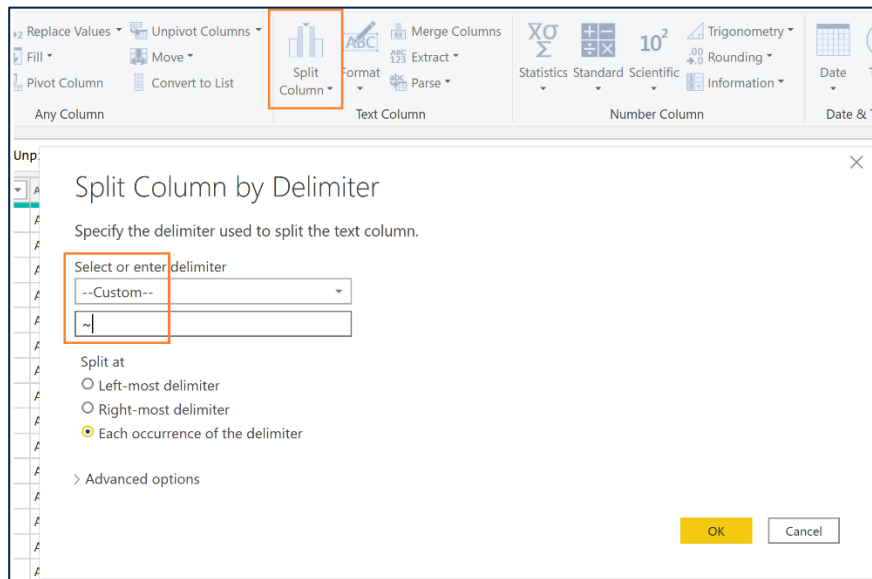
13. **Remove Rows** > Remove Top Rows, enter 3 (to remove the first 3 rows – the old header rows)



14. Highlight **Category** and **Segment** and **Transform** > **Unpivot Other Columns**



15. Highlight **Attribute** and navigate to **Home** > **Split Column** > By Delimiter > "~"



16. Rename: Attribute.1 = **"Scenario"**, Attribute.2 = **"Date"**, Value = **"Budget Amount"**

Scenario	Date	Budget Amount
Forecast	12/1/2016	44190.57888
Forecast	11/1/2016	50598.81566
Forecast	10/1/2016	54740.5709

17. Change the Data Types: **Budget Amount** = **Fixed Decimal**, **Date** = **Date**

18. **Home** > **Merge Queries** > Select **CatSegDim**

- h. a) From the CatSegDim highlight both **Category** and **Segment**
- i. b) Go back up to BudgetFact, highlight both **Category** and **Segment**
- j. c) Show the Join Kinds available, and leave **"Left Outer"**
- k. d) Expand NewColumn > Select **"CatSegID"** and deselect **"Use Original column name as prefix"**

Merge

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

BudgetFact

Category	1	Segment	2	Scenario	Date	Budget Amount
Accessory	Accessory	Forecast		12/1/2016		44,190.58
Accessory	Accessory	Forecast		11/1/2016		50,598.82
Accessory	Accessory	Forecast		10/1/2016		54,740.57
Accessory	Accessory	Forecast		9/1/2016		64,442.91
Accessory	Accessory	Forecast		8/1/2016		98,285.91

CatSegDim

CatSegDim	Category	1	Segment	2
1	Rural		Productivity	
2	Rural		Select	
3	Accessory		Accessory	
4	Urban		Moderation	
5	Urban		Regular	

Join Kind
Left Outer (all from first, matching from second)

☐ Use fuzzy matching to perform the merge

> Fuzzy matching options

✓ The selection matches 324 of 324 rows from the first table.

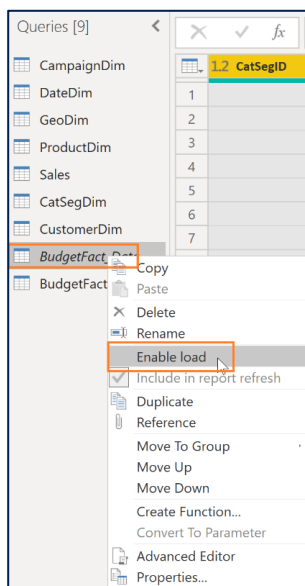
OK Cancel

19. Remove: **Category, Segment**

20. Reorder: **CatSegID, Scenario, Date, Budget Amount**

	1.2 CatSegID	Scenario	Date	\$ Budget Amount
1	3	Forecast	12/1/2016	44,190.58
2	3	Forecast	11/1/2016	50,598.82
3	3	Forecast	10/1/2016	54,740.57

21. **Disable** the load of **BudgetFact_Data**

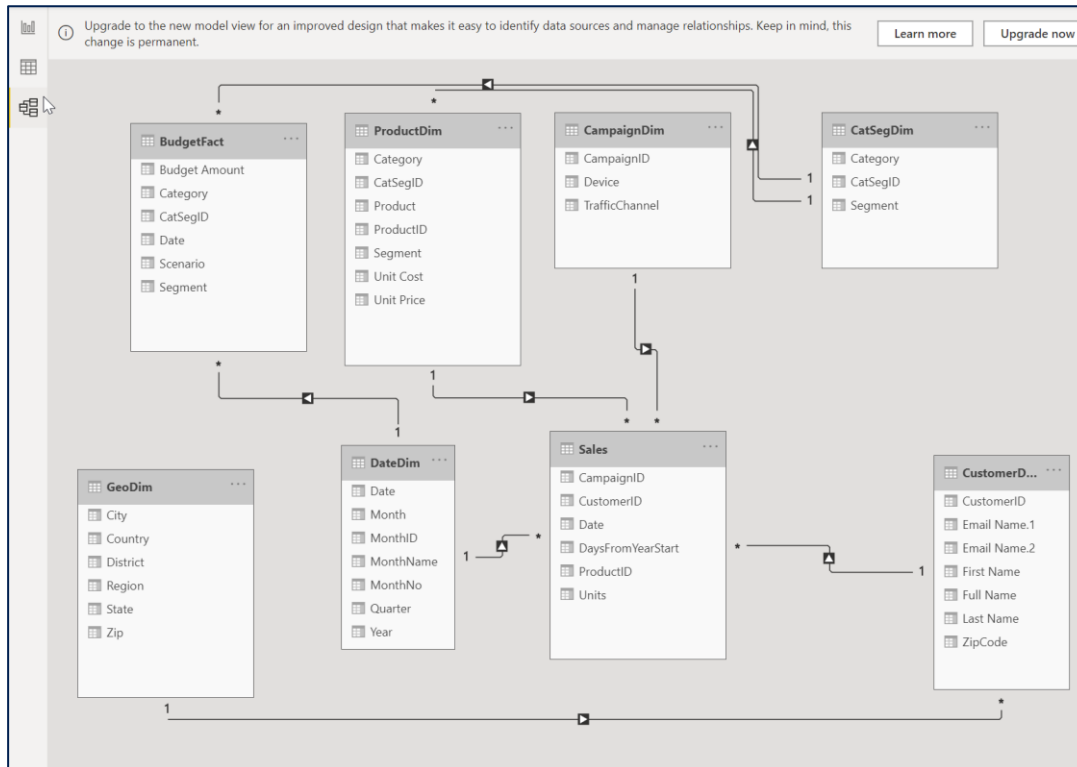


Lab 3: Create relationships between tables

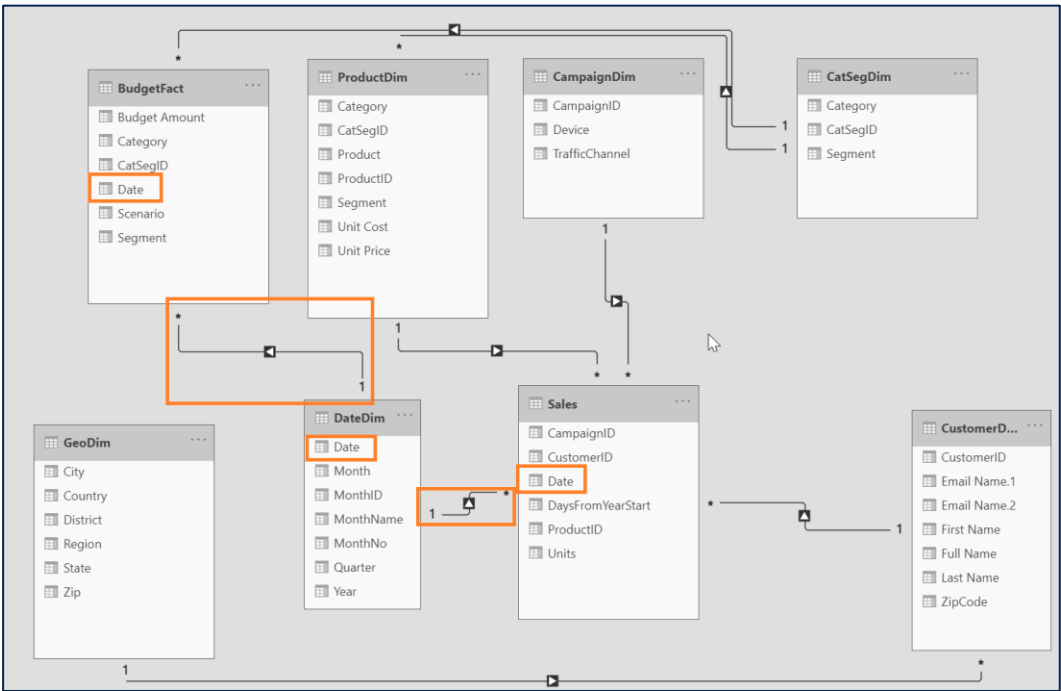
Task: Create relationships between multiple tables.

The estimated time to complete this lab is 20 minutes.

1. Navigate to **model** view.



2. Drag a **relationship** line between **Date** field from **Sales** table to **Date** field from **DateDim** table, create the same relationship between the **Date** table and the **BudgetFact** table

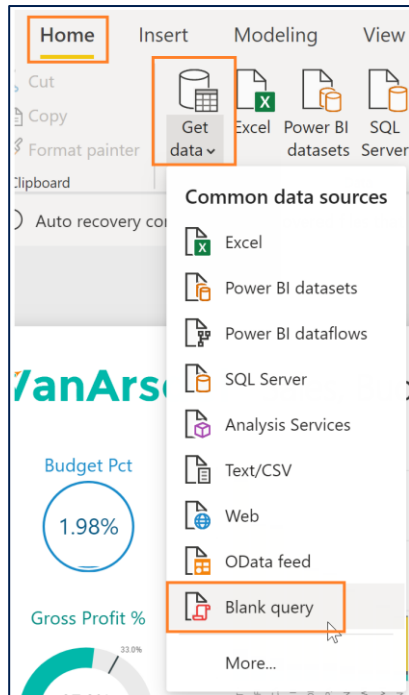


Lab 4a: Create dynamic path to excel source file

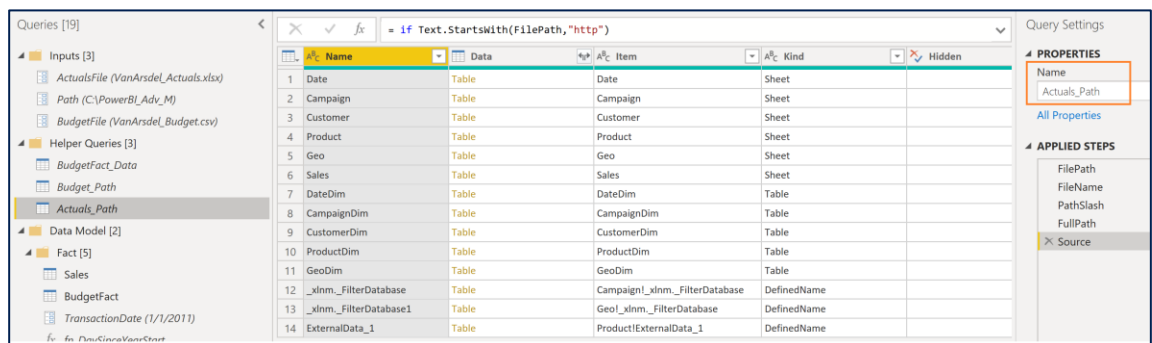
Task: Create a dynamic path to the excel source file

The estimated time to complete this lab is 15 minutes.

1. Create a new blank query
 - a) Query Name: "Actuals_Path"



- b) Copy in text from the file Actuals_Path.txt



2. Update Source Applied Step to use Resolved Path = "Actuals_Path" to the following Queries: • CampaignDim • CustomerDim • ProductDim • CatSegDim • Date • GeoDim • Sales

Queries [19]

- Inputs [3]
 - ActualsFile (VanArsdel_Actuals.xlsx)
 - Path (C:\PowerBI_Adv_M)
 - BudgetFile (VanArsdel_Budget.csv)
- Helper Queries [3]
 - BudgetFact_Data
 - Budget_Path
 - Actuals_Path
- Data Model [2]
 - Fact [5]
 - Sales
 - BudgetFact
 - TransactionDate (1/1/2011)
 - fn_DaySinceYearStart
 - Query2
 - Dimensions [6]
 - CampaignDim
 - DateDim
 - GeoDim
 - ProductDim
 - CatSegDim

Query: Actuals_Path

Name	Data	Item	Kind	Hidden
1 Date	Table	Date	Sheet	
2 Campaign	Table	Campaign	Sheet	
3 Customer	Table	Customer	Sheet	
4 Product	Table	Product	Sheet	
5 Geo	Table	Geo	Sheet	
6 Sales	Table	Sales	Sheet	
7 DateDim	Table	DateDim	Table	
8 CampaignDim	Table	CampaignDim	Table	
9 CustomerDim	Table	CustomerDim	Table	
10 ProductDim	Table	ProductDim	Table	
11 GeoDim	Table	GeoDim	Table	
12 _xlnm_FilterDatabase	Table	Campaign!_xlnm_FilterDatabase	DefinedName	
13 _xlnm_FilterDatabase1	Table	Geo!_xlnm_FilterDatabase	DefinedName	
14 ExternalData_1	Table	Product!ExternalData_1	DefinedName	

Query Settings

PROPERTIES

Name: CampaignDim

APPLIED STEPS

Source

Navigation

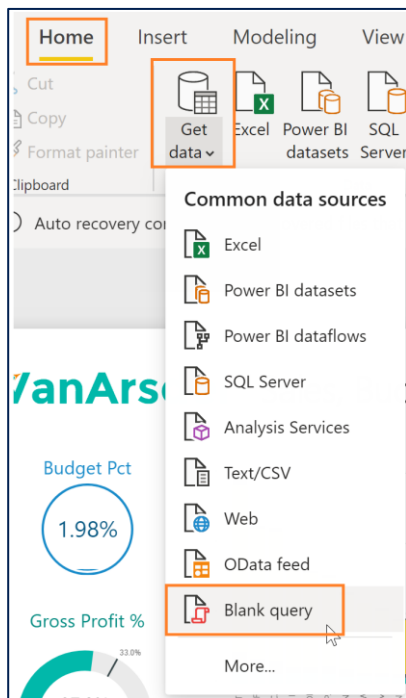
Changed Type

Lab 4b: Create dynamic path to csv source file

Task: Create a dynamic path to csv source file

The estimated time to complete this lab is 15 minutes.

1. Create a new blank query a) Query Name: "Budget_Path"



2. Copy in text from Budget_Path.txt

Query Settings					
= if Text.StartsWith(FilePath,"http")					
	A ^B C Column1	A ^B C Column2	A ^B C Column3	A ^B C Column4	A ^B C Column5
1	Budget Spreadsheet for VanArsdel				
2					
3					
4			Forecast	Forecast	Forecast
5			2016	2016	2016
6	Category	Segment	Dec	Nov	Oct
7	Accessory	Accessory	44190.57888	50598.81566	54740.5709
8	Mix	All Season	11442.14474	14120.78693	18109.64804
9	Mix	Productivity	19538.89812	17597.55926	22835.18396
10	Rural	Select	311.708775	172.2601125	662.79129
11	Urban	Convenience	120710.4406	129923.2814	169468.7696
12	Urban	Extreme	20868.84072	46971.33037	70793.02886
13	Urban	Moderation	251155.7122	322984.2215	362385.6466
14	Urban	Regular	689.7969225	427.4372025	2989.28376
15	Youth	Youth	3931.03074	2891.005425	5397.748965

PROPERTIES

Name
Budget_Path

[All Properties](#)

APPLIED STEPS

FilePath
BudgetFilename
PathSlash
FullPath
Source

- UpdateSource Applied Step to use ResolvedBudgetPath = "Budget_Path" to the following Queries: • BudgetFact • BudgetFact_Data

Queries [19]

- Inputs [3]
 - ActualsFile (VanArsdel_Actuals.xlsx)
 - Path (C:\PowerBI_Adv_M)
 - BudgetFile (VanArsdel_Budget.csv)
- Helper Queries [3]
 - BudgetFact_Data
 - Budget_Path
 - Actuals_Path
- Data Model [2]
 - Sales
 - BudgetFact**

TransactionDate (1/1/2011)
fn_DaySinceYearStart

fx = Budget_Path

	A ⁰ C Column1	A ⁰ C Column2	A ⁰ C Column3	A ⁰ C Column4	A ⁰ C Column5
1	Budget Spreadsheet for VanArsdel				
2					
3					
4			Forecast	Forecast	Forecast
5			2016	2016	2016
6	Category	Segment	Dec	Nov	Oct
7	Accessory	Accessory	44190.57888	50598.81566	54740.5709
8	Mix	All Season	11442.14474	14120.78693	18109.64804
9	Mix	Productivity	19538.89812	17597.55926	22835.18396
10	Rural	Select	311.708775	172.2601125	662.79129
11	Urban	Convenience	120710.4406	129923.2814	169468.7696
12	Urban	Extreme	20868.84072	46971.33037	70793.02886
13	Urban	Moderation	251155.7122	322984.2215	362385.6466
14	Urban	Regular	689.7969225	427.4372025	2989.28376
15	Youth	Youth	3931.03074	2891.005425	5397.748965

Query Settings

PROPERTIES

Name
BudgetFact

All Properties

APPLIED STEPS

Source

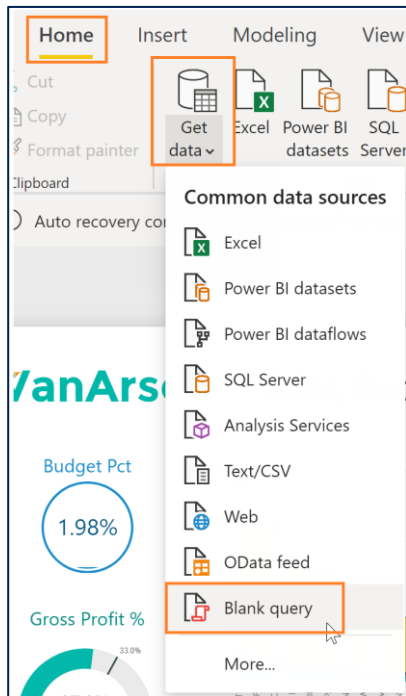
Changed Type
Removed Top Rows
Added Index
Filtered Rows
Removed Columns
Transposed Table
Added Custom
Added Custom1
Removed Other Columns
Transposed Table1

Lab 4c: Create a custom function

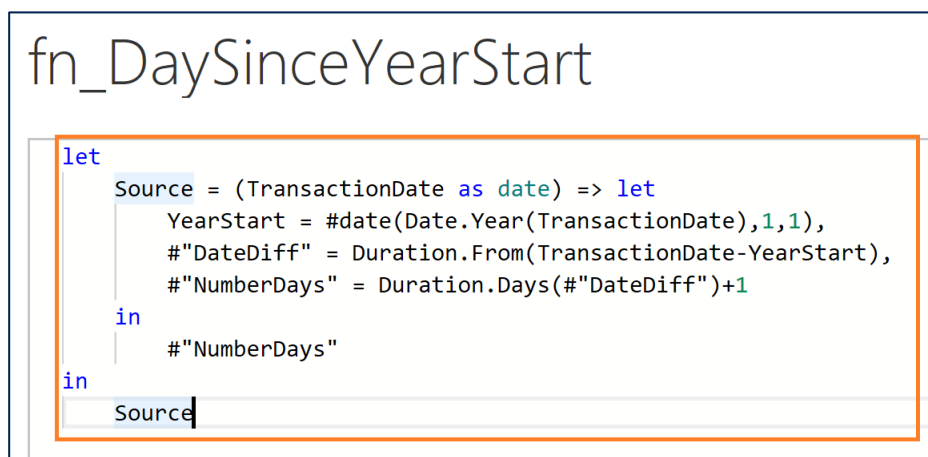
Task: Create a custom function

The estimated time to complete this lab is 15 minutes.

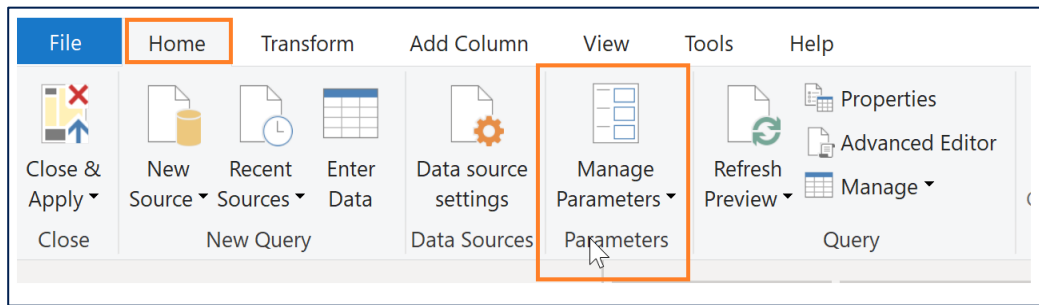
1. Create a new blank query



- a. Query Name: "**fn_DaySinceYearStart**"
- b. In Advanced Editor copy in text from **Number_Days.txt**



2. Create a new Parameter



- a. Parameter Name: **TransactionDate**
- b. Type: **Date**
- c. Current Value = **1/1/2011**

Manage Parameters

New

TransactionDate

ActualsFile

Path

BudgetFile

Name

TransactionDate

Description

☒ Required

Type

Date

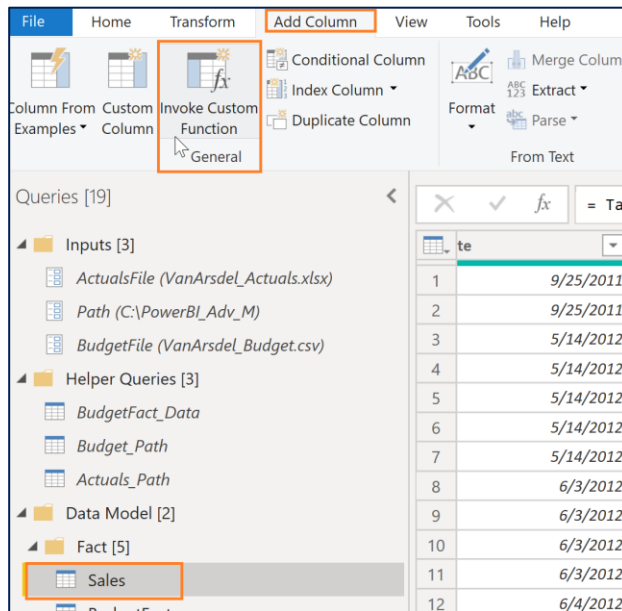
Suggested Values

Any value

Current Value

1/1/2011

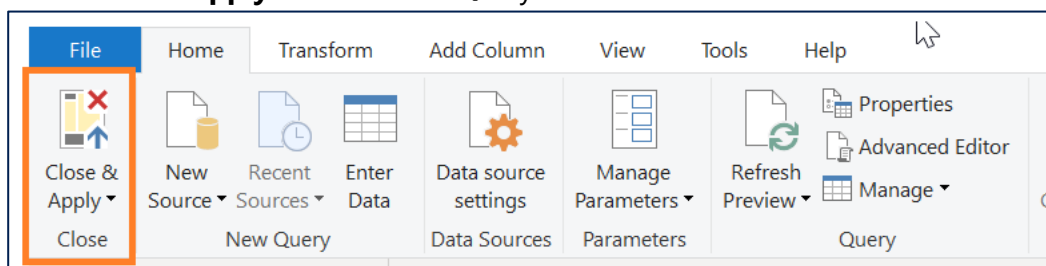
3. Update Sales query
 - a. **Add Column -> Invoke Custom Function**



- b. New Column Name: **DaysFromYearStart**
- c. Function query: **fn_DaySinceYearStart**
- d. Transaction Date = **Select current date from calendar**

= Table.AddColumn(#"Changed Type", "DaysFromYearStart", each fn_DaySinceYearStart([Date]))					
te	CustomerID	CampaignID	Units		DaysFromYearStart
1	9/25/2011	70283	22	1	268
2	9/25/2011	195385	22	1	268
3	5/14/2012	212645	22	1	135
4	5/14/2012	70666	22	1	135
5	5/14/2012	114459	22	1	135
6	5/14/2012	221670	22	1	135
7	5/14/2012	26974	22	1	135
8	6/3/2012	268392	22	1	155

4. Click **Close & Apply** to exit Power Query



5. Close and Save the pbi file.

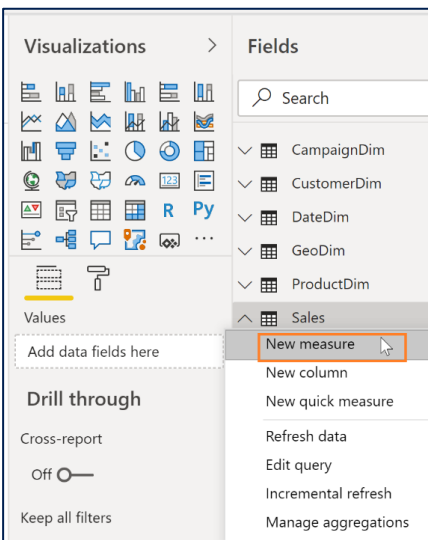
Lab 5: Create new measures and columns

Tasks: You will create a new measure for Total Units Sold, a new calculated column that combines Product Category and Campaign Traffic together, and create visualizations to test the new measure and column

Task 1. Create **Total Units Sold** measure

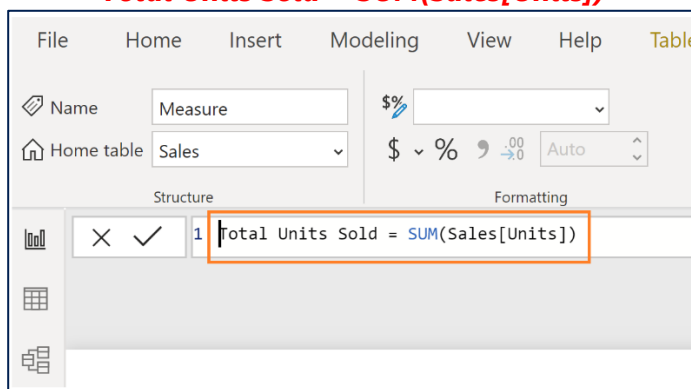
The estimated time to complete this lab is 30 minutes.

1. Select **Sales** Table. From the ribbon select **Modeling** -> **New Measure**

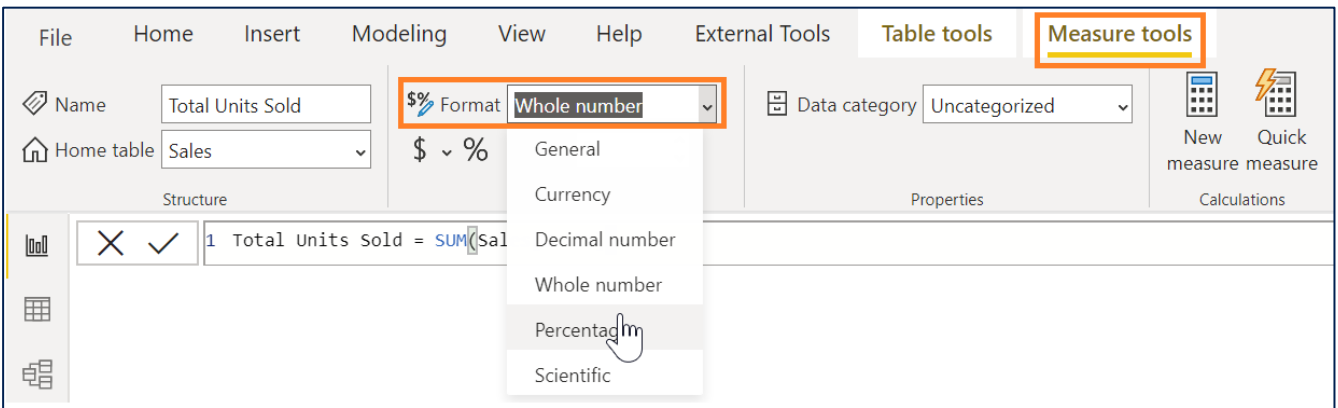


2. In the formula bar enter:

Total Units Sold = SUM(Sales[Units])

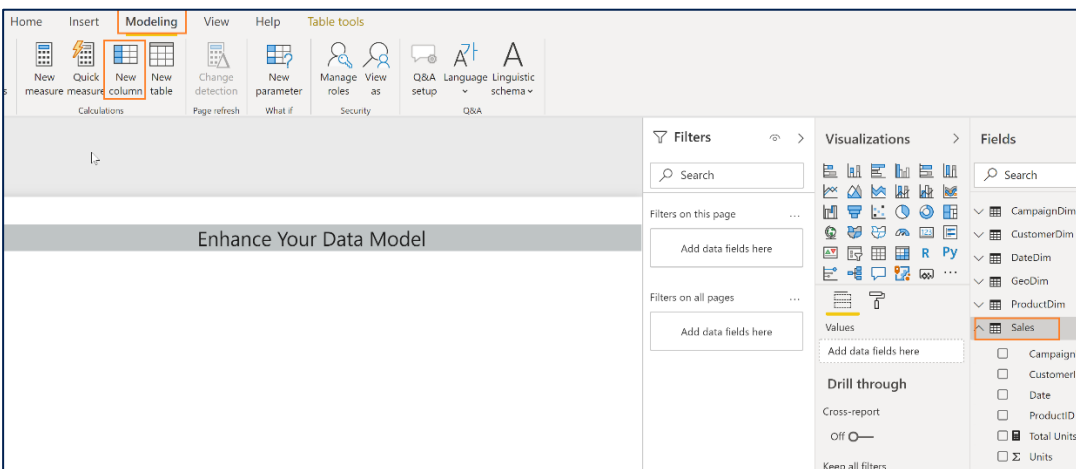


3. From the **Measure tools** ribbon select **Format** -> **Whole number** to format the measure



Task 2: Create calculated column that combines Category and TrafficChannel

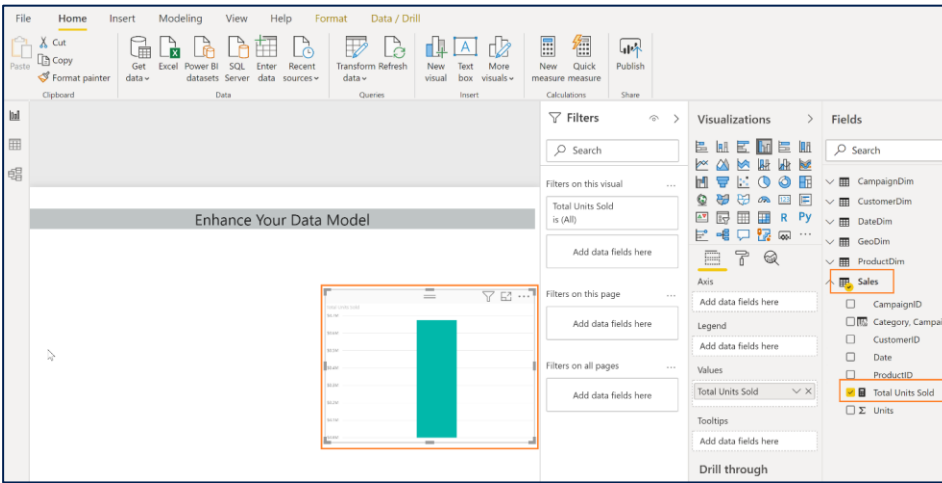
4. Select **Sales** table. From the ribbon select **Modeling** -> **New Column**.



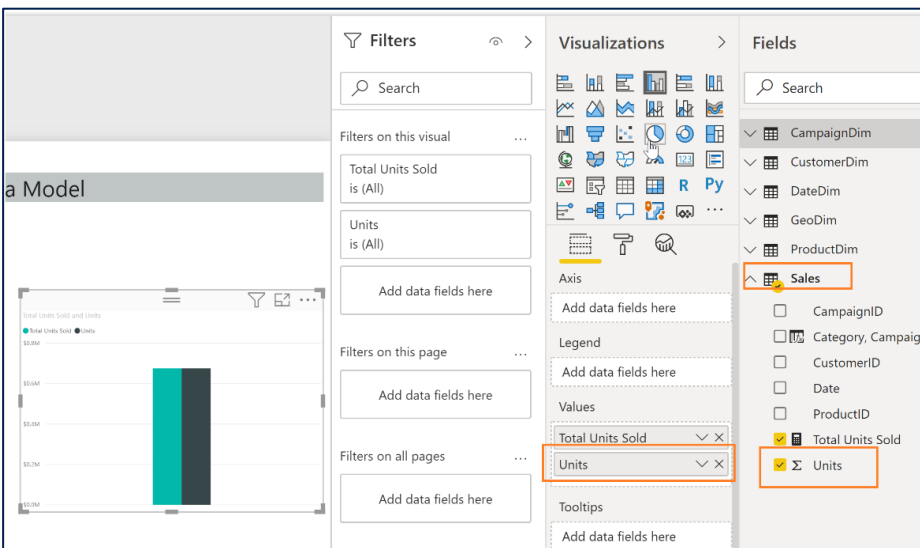
5. In the formula bar enter:

Category, Campaign = RELATED(ProductDim[Category]) & ", " & RELATED(CampaignDim[TrafficChannel])

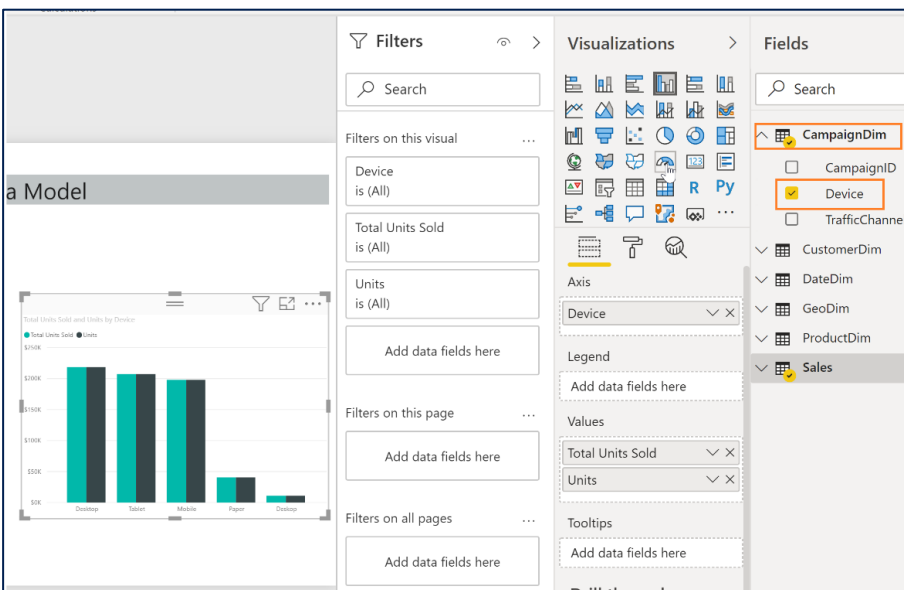
6. Drag newly created **Total Units Sold** measure to the canvas. A clustered column chart is created



7. Drag **Units** field from **Sales** table to this visual



8. Select **Device** field from **CampaignDim** table

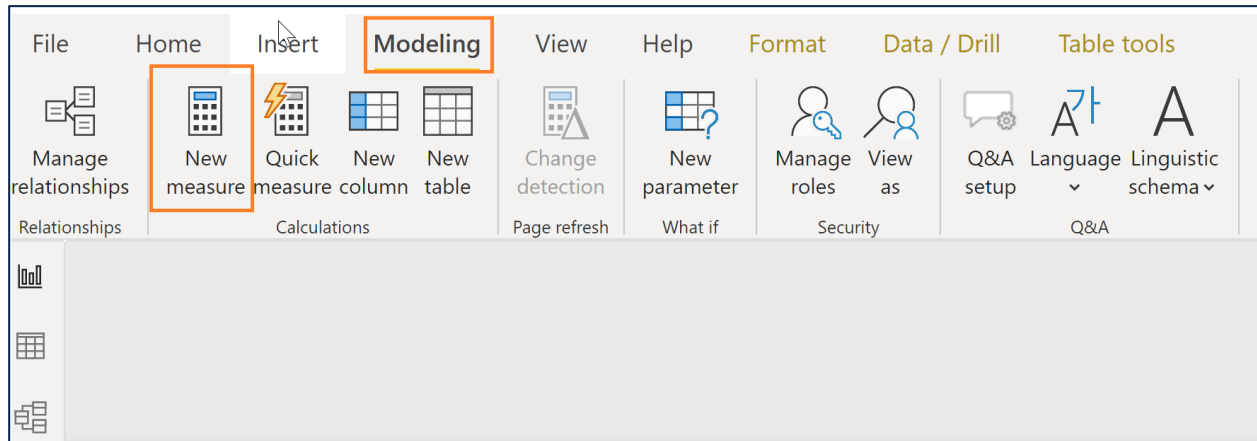


Lab 6: Create a report for the VP in charge of the Youth and Accessory Segments

Task 3: Create three new measures and a PowerBI visualization

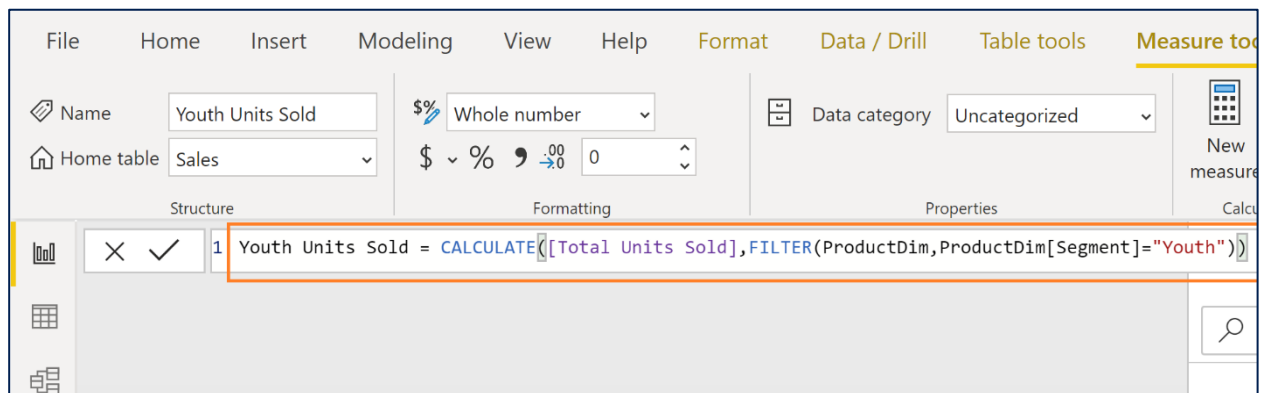
The estimated time to complete this lab is 45 minutes.

1. Select **Sales** Table. From the ribbon select **Modeling** -> **New Measure**.

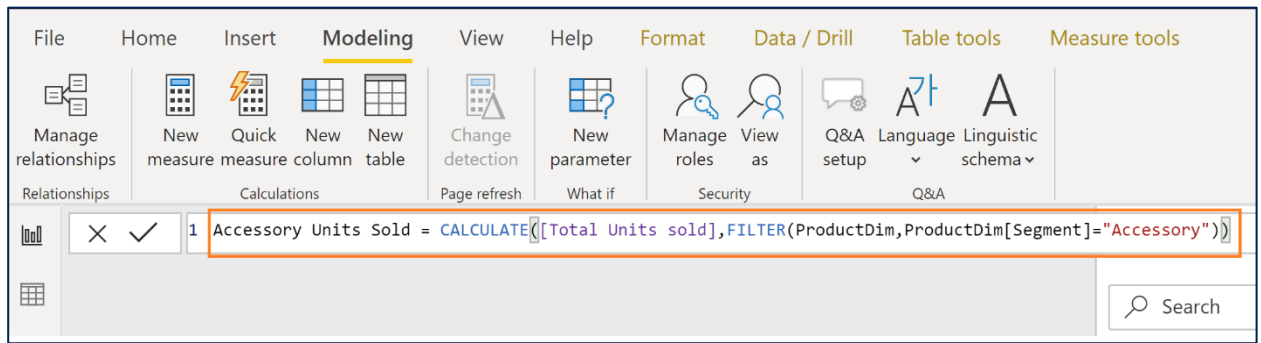


2. Create 3 measures:

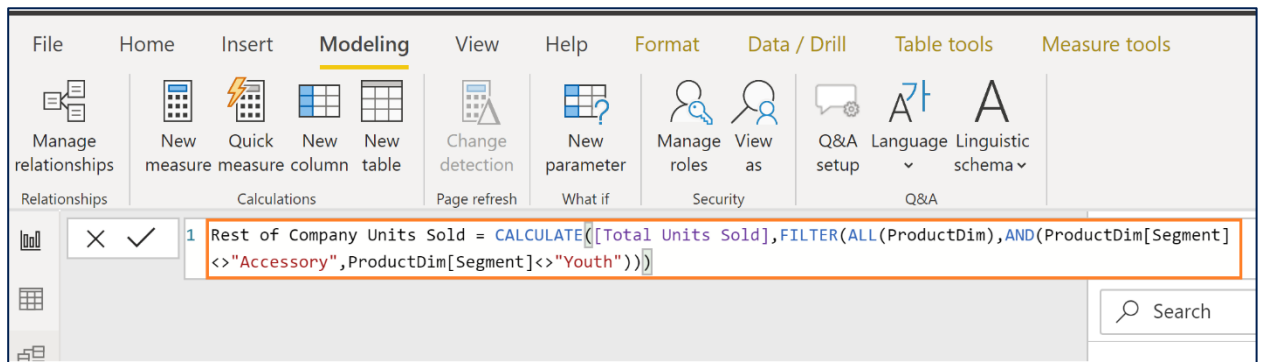
Youth Units Sold = CALCULATE([Total Units Sold],FILTER(ProductDim,ProductDim[Segment]="Youth"))



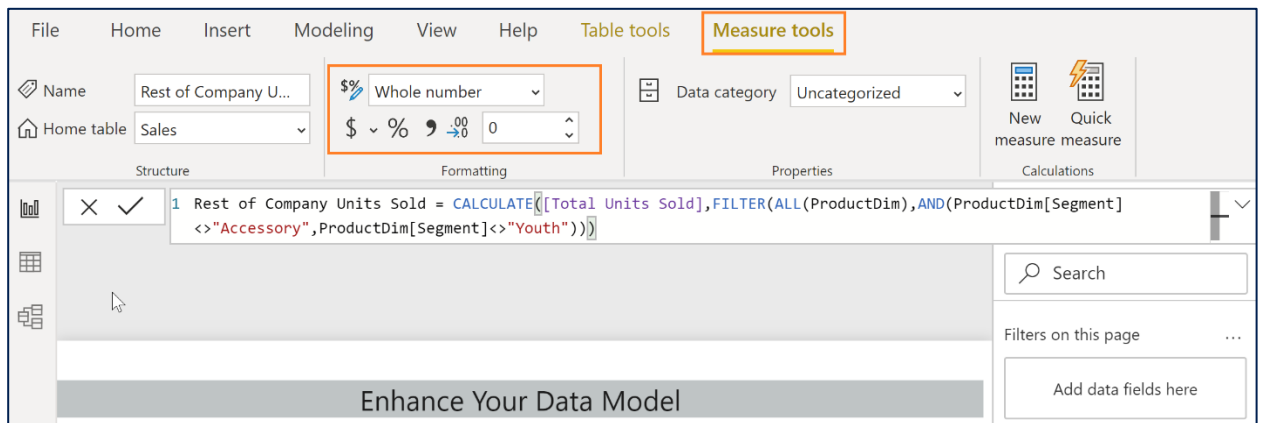
Accessory Units Sold = CALCULATE([Total Units sold],FILTER(ProductDim,ProductDim[Segment]="Accessory"))



Rest of Company Units Sold = CALCULATE([Total Units Sold],FILTER(ALL(ProductDim),AND(ProductDim[Segment]<>"Accessory",ProductDim[Segment]<>"Youth")))



- From the ribbon select **Measure tools** -> **Whole Number and Comma** to format the measure



- Add a table visual and drag **CampaignDim** -> **Device** and the **3 newly created measures**

Data Model

Device	Total Units Sold	Youth Units Sold	Rest of Company Units Sold
Desktop	610,902	223	9051
Desktop	5218,680	4533	201335
Mobile	1110,014	4427	151767
Tablet	146,514	105	17460
Tablet	5207,344	1151	159855
Total	8679,368	11541	420858

Filters

Filters on this visual

- Device is (All)
- Rest of Company Unit... is (All)
- Total Units Sold is (All)
- Youth Units Sold is (All)

Filters on this page

Filters on all pages

Visualizations

Values

- Device
- Total Units Sold
- Youth Units Sold
- Rest of Company Units

Drill through

Cross-report

Off

Keep all filters

Off

Fields

CampaignDim

- CampaignID
- Device
- TrafficChannel

CustomerDim

- CustomerID
- Date
- ProductID

DateDim

- Date
- Category, Campaign
- CustomerID

ProductDim

- ProductID
- Accessory Units Sold
- CampaignID
- Category, Campaign
- CustomerID
- Date
- ProductID

Sales

- Accessory Units Sold
- CampaignID
- Category, Campaign
- CustomerID
- Date
- ProductID
- Rest of Company Units Sold
- Total Units Sold
- Units
- Youth Units Sold

5. Select **Line Chart** visual.

Data Model

Device	Total Units Sold	Youth Units Sold	Rest of Company Units Sold
Desktop	610,902	223	9051
Desktop	5218,680	4533	201335
Mobile	1110,014	4427	151767
Tablet	146,514	105	17460
Tablet	5207,344	1151	159855
Total	8679,368	11541	420858

Filters

Filters on this visual

Add data fields here

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Visualizations

Line Chart

Axis

Add data fields here

Legend

Add data fields here

Values

Add data fields here

Secondary values

Add data fields here

Fields

CampaignDim

- CampaignID
- Device
- TrafficChannel

CustomerDim

- CustomerID
- Date
- ProductID

DateDim

- Date
- Category, Campaign
- CustomerID

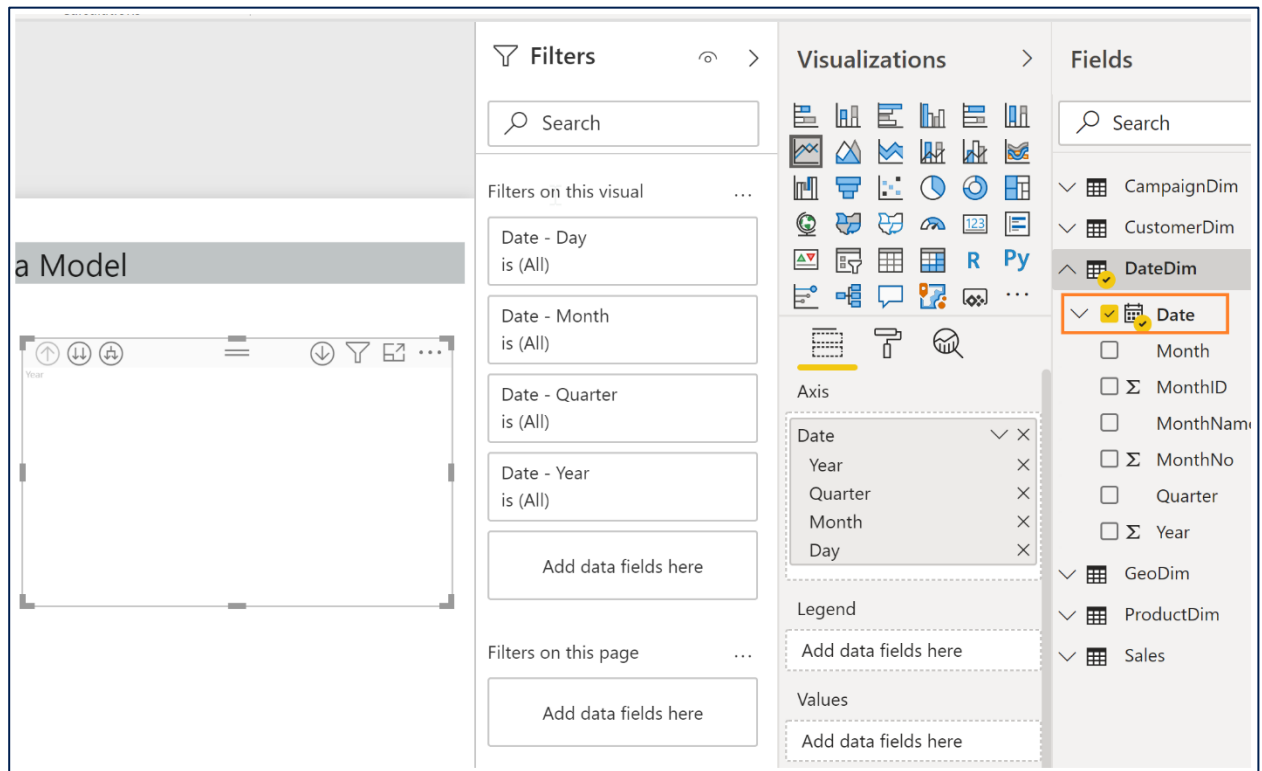
ProductDim

- ProductID
- Accessory Units Sold
- CampaignID
- Category, Campaign
- CustomerID
- Date
- ProductID

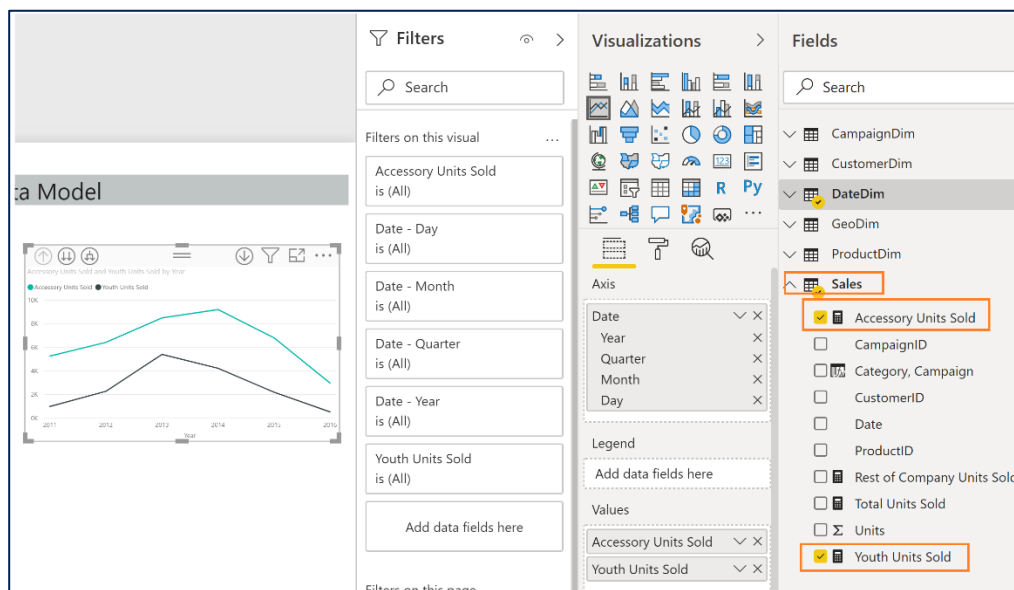
Sales

- Accessory Units Sold
- CampaignID
- Category, Campaign
- CustomerID
- Date
- ProductID
- Rest of Company Units Sold
- Total Units Sold
- Units
- Youth Units Sold

6. Select **Date** from Date table

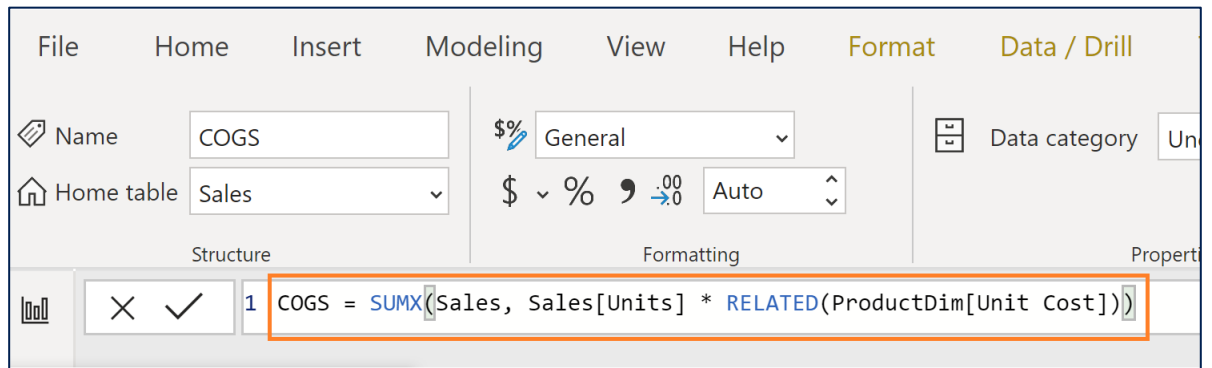


7. Select **Youth Units Sold** and **Accessory Units sold** measures.

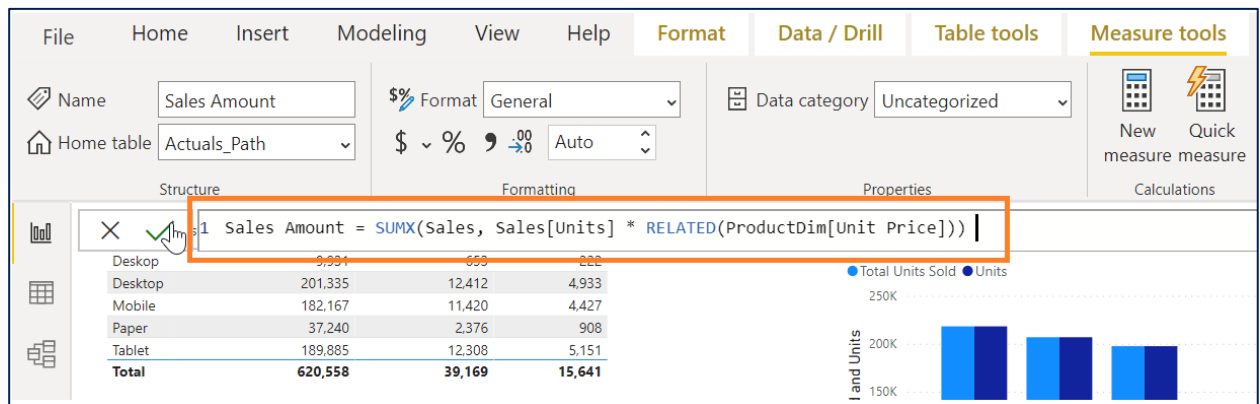


8. Create following measures and use a visual to analyze data.

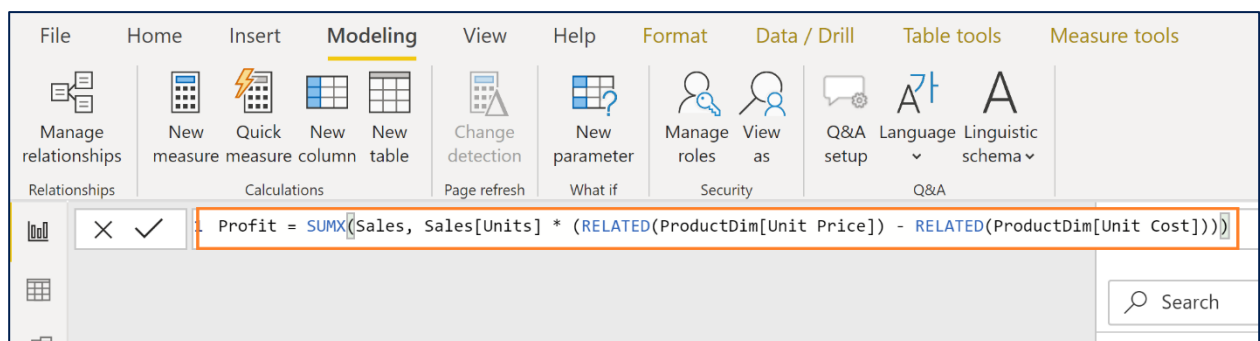
$$COGS = SUMX(Sales, Sales[Units]) * RELATED(ProductDim[Unit Cost]))$$



Sales Amount = SUMX(Sales, Sales[Units] * RELATED(ProductDim[Unit Price]))



Profit = SUMX(Sales, Sales[Units] * (RELATED(ProductDim[Unit Price]) - RELATED(ProductDim[Unit Cost])))



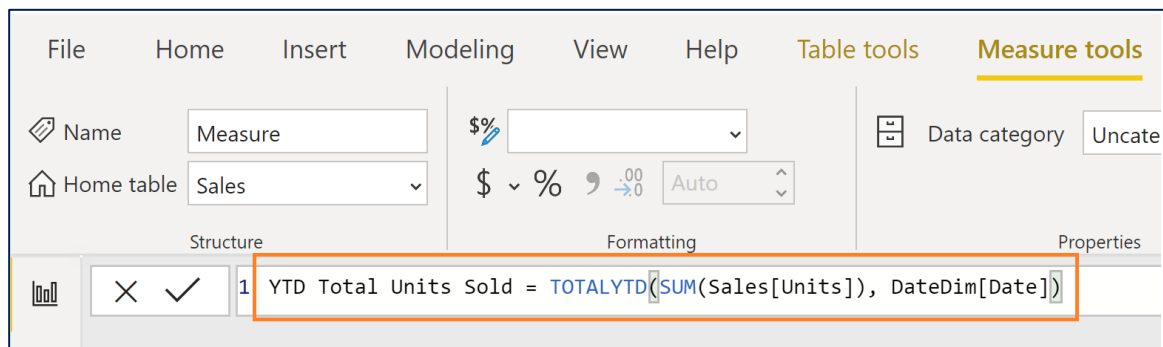
Lab 7a: Performance Best Practices

Task: Analyze DAX formulas to ensure you're using the best practices

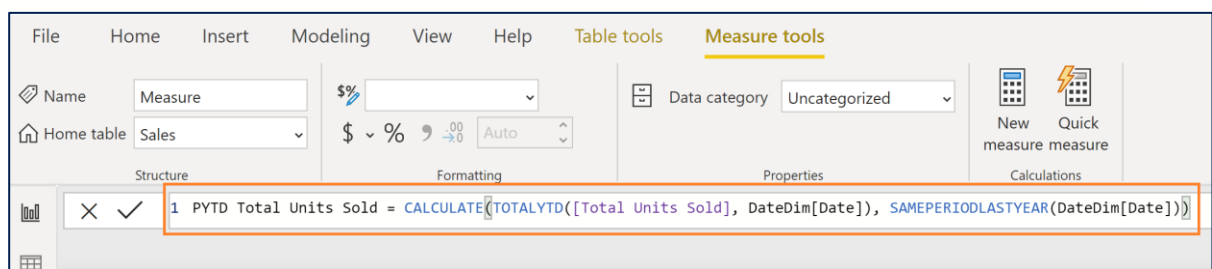
The estimated time to complete this lab is 30 minutes.

1. Create the three measures below.

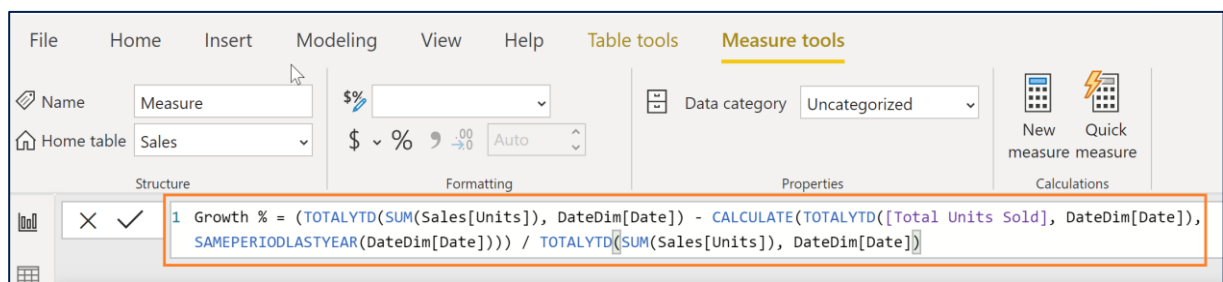
YTD Total Units Sold = TOTALYTD(SUM(Sales[Units]), DateDim[Date])



PYTD Total Units Sold = CALCULATE(TOTALYTD([Total Units Sold], DateDim[Date]), SAMEPERIODLASTYEAR(DateDim[Date]))



Growth % = (TOTALYTD(SUM(Sales[Units]), DateDim[Date]) - CALCULATE(TOTALYTD([Total Units Sold], DateDim[Date]), SAMEPERIODLASTYEAR(DateDim[Date]))) / TOTALYTD(SUM(Sales[Units]), DateDim[Date])



2. Rewrite the DAX formulas to ensure performance best practices

YTD Total Units Sold = TOTALYTD([Total Units Sold], DateDim[Date])

**PYTD Total Units Sold = CALCULATE([YTD Total Units Sold],
SAMEPERIODLASTYEAR(DateDim[Date]))**

Growth % = ([YTD Total Units Sold] – [PYTD Total Units Sold]) / [PYTD Total Units Sold]

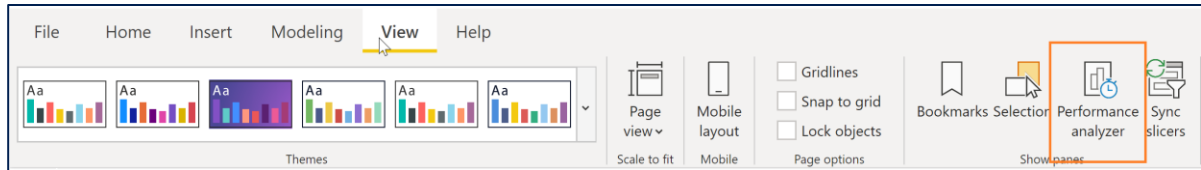
3. **Challenge:** how were the formulas changed to increase performance?

Lab 7b: Using Performance Analyzer

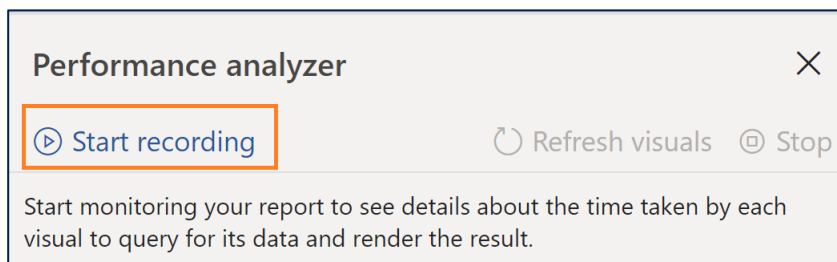
Task: Use performance analyzer to test the performance of individual charts

The estimated time to complete this lab is 15 minutes.

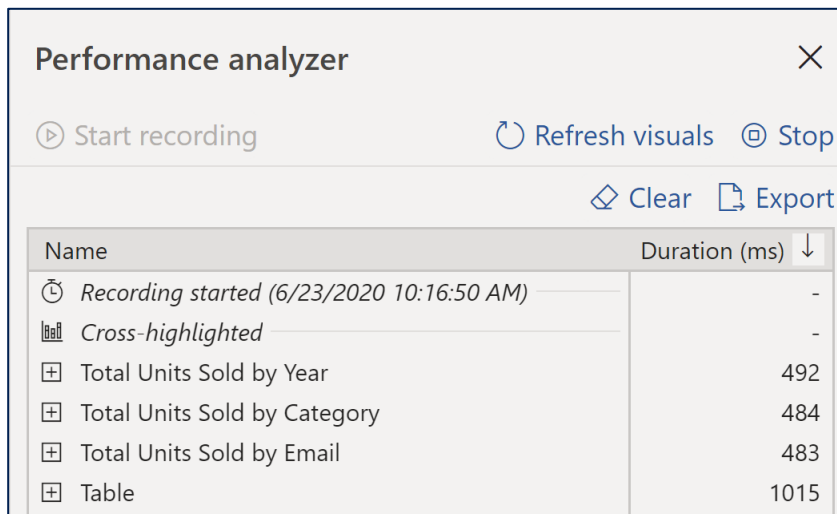
1. Click on the View tab and open the Performance Analyzer



2. Start recording and click on Refresh visuals



3. Analyze the data collected



4. **Challenge:** What changes can you make to improve performance?

Summary

In this lab, you uploaded multiple tables from a single data source. You learned how to create a brand new dimension for the model as well as enhance existing dimensions. You will have created a new budget fact table for the model. In the end you will have created new parameters and dynamic paths to your data sources.

In this lab, you have also enhanced the model by adding additional measures and columns. In the end you will have tested the new measures and columns using Power BI data visualizations as well as worked with best practices and analyzed report performance.

Terms of Use

© 2021 Microsoft. All rights reserved.

By using this hands-on lab, you agree to the following terms:

The technology/functionality described in this hands-on lab is provided by Microsoft Corporation in a “sandbox” testing environment for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the hands-on lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. Without written permission, you may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this hands-on lab or any portion thereof.

COPYING OR REPRODUCTION OF THE HANDS-ON LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION WITHOUT WRITTEN PERMISSION IS EXPRESSLY PROHIBITED.

THIS HANDS-ON LAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS HANDS-ON LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCTIONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

FEEDBACK If you give feedback about the technology features, functionality and/or concepts described in this hands-on lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE HANDS-ON LAB, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

DISCLAIMER This lab contains only a portion of new features and enhancements in Microsoft Power BI. Some of the features might change in future releases of the product.