

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.

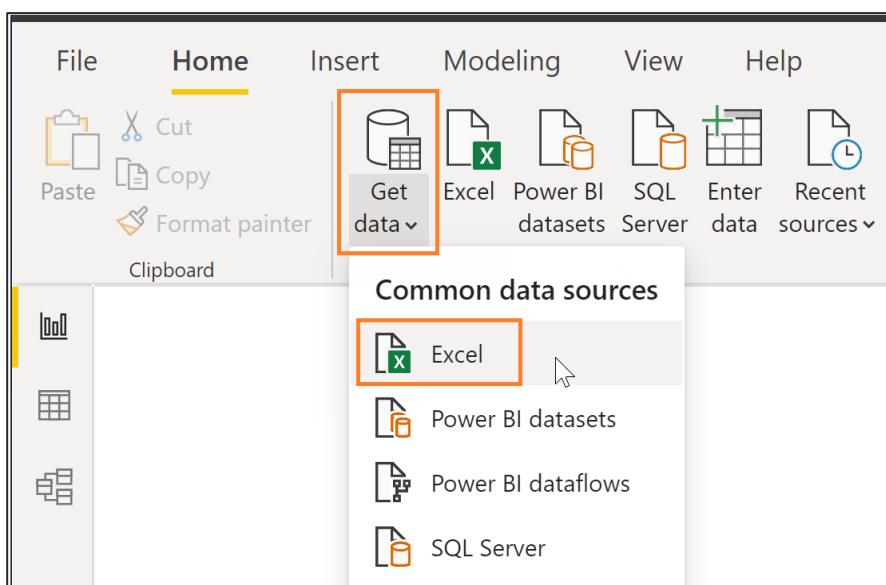
Data Shaping Labs

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

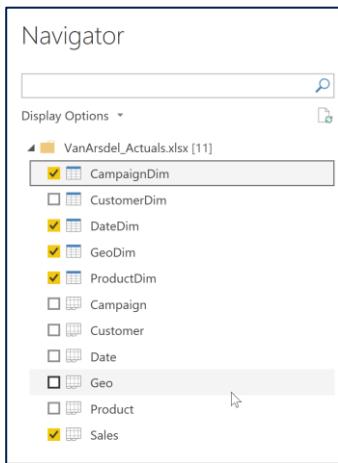
Task: Import multiple tables from a single excel source file.

The estimated time to complete this lab is 15 minutes.

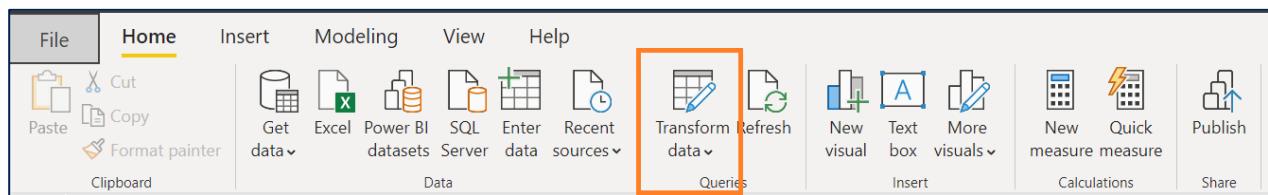
1. **GetData** > from Excel C:\Power BI_Adv_M\VansArsdel_Actuals.xlsx



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**

The screenshot shows the 'Transform Column Types' dialog for the 'Zip' column in the 'GeoDim' table. The dialog lists various data types: Decimal Number, Fixed decimal number, Whole Number, Percentage, Date/Time, Date, Time, Date/Time/Timezone, Duration, Text, True/False, Binary, and Using Locale... The 'Text' option is selected and highlighted with a red box.

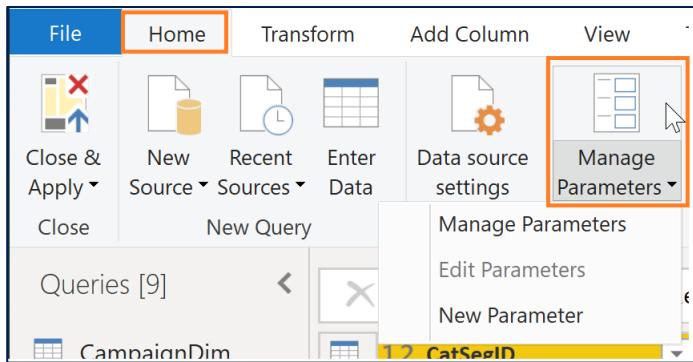
	A ^B C Zip	A ^B C City
1	1.2 Decimal Number	Star Tannery, VA, USA
2	\$ Fixed decimal number	Stephens City, VA, USA
3	123 Whole Number	Stephenson, VA, USA
4	% Percentage	Strasburg, VA, USA
5	Date/Time	Toms Brook, VA, USA
6	Date	White Post, VA, USA
7	Time	Woodstock, VA, USA
8	Date/Time/Timezone	Culpeper, VA, USA
9	Duration	Aroda, VA, USA
10	A ^B C Text	Banco, VA, USA
11	True/False	Bealeton, VA, USA
12	Binary	Boston, VA, USA
	Using Locale...	

Lab 1b: Create lab parameters

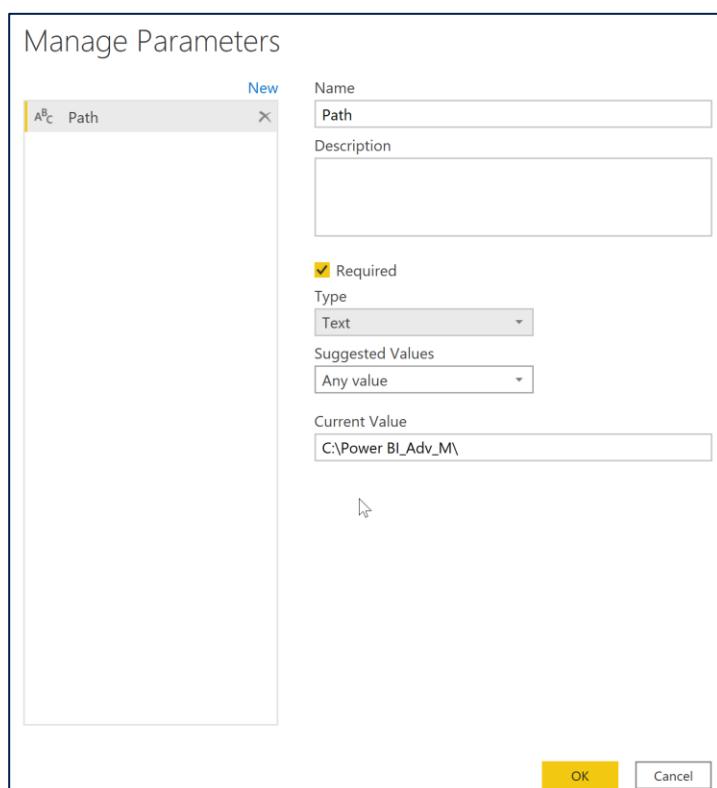
Task: Create new parameters

The estimated time to complete this lab is 15 minutes.

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

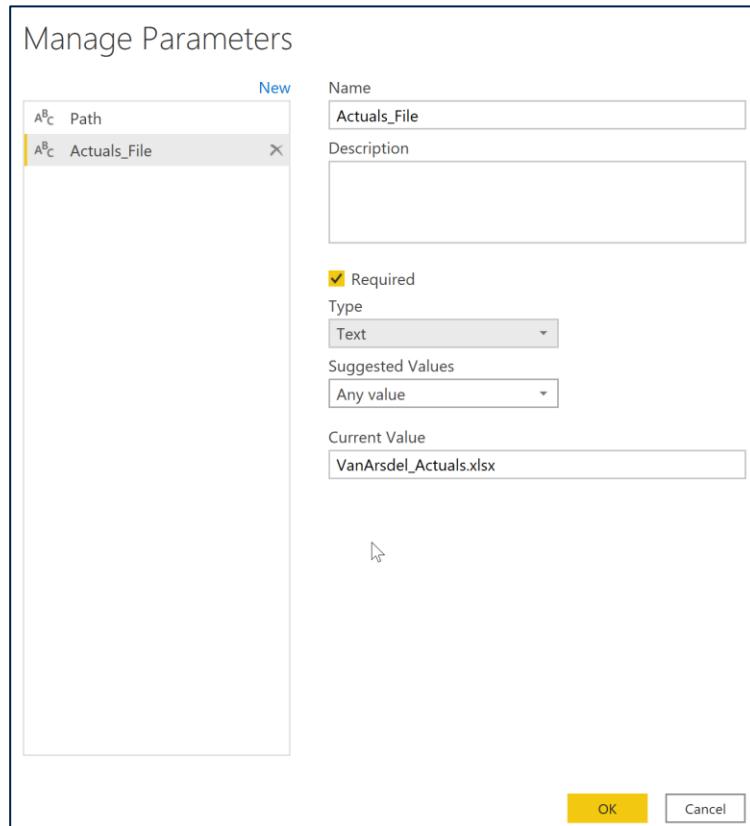


3. Create a new **Parameter**
 - a. Parameter Name: **Path**
 - b. Type: **Text**
 - c. Current Value = **C:\Power BI_Adv_M**



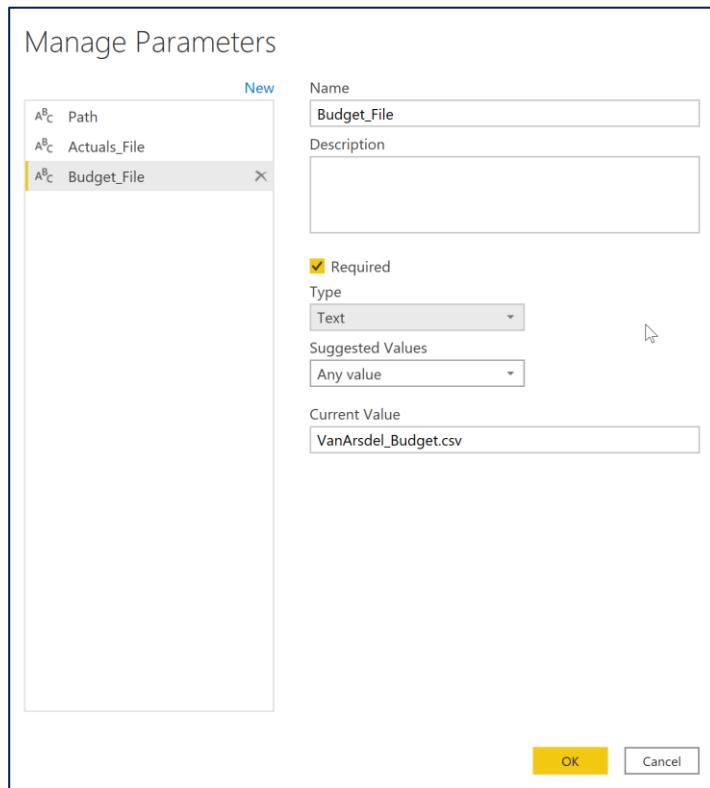
4. Create a new **Parameter**

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



5. Create a new **Parameter**

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



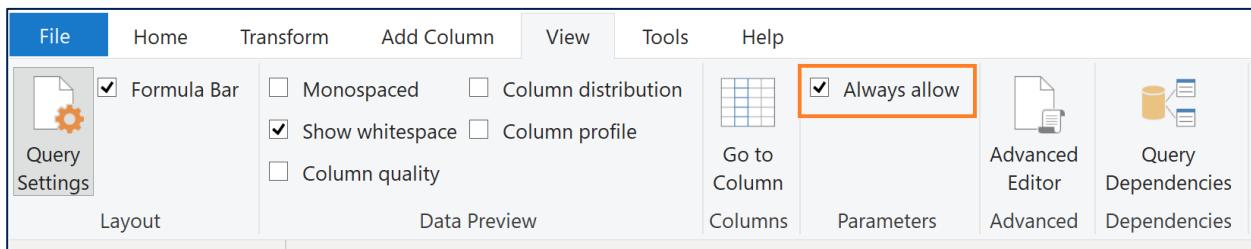
6. Update the text files to ensure the parameter names are consistent

Lab 2a: Create CatSegDim

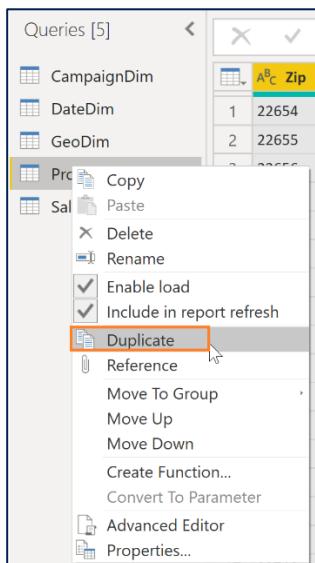
Task: Create a dimension using Category and Segment attributes from the Product dimension

The estimated time to complete this lab is 30 minutes.

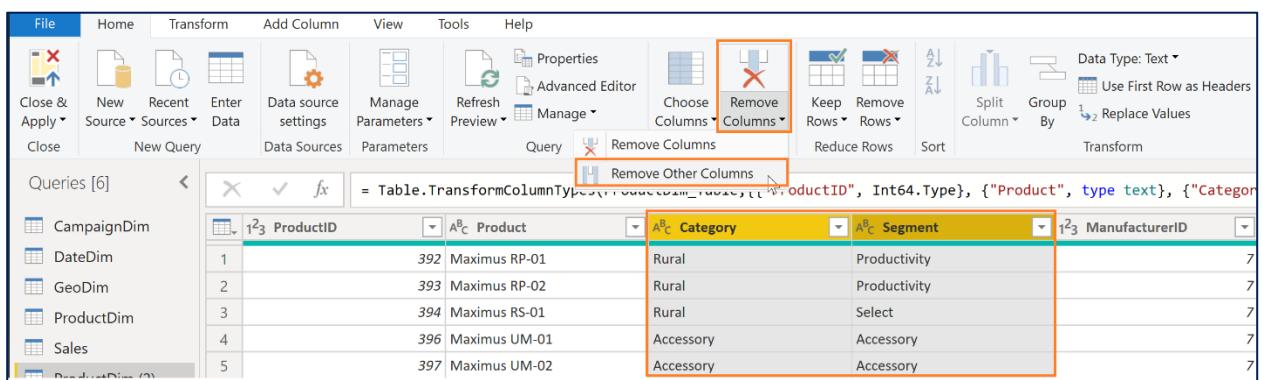
1. From the **View** ribbon check "**Always Allow**"



2. Duplicate the **ProductDim** query



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



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

The screenshot shows a table with two columns: "Category" and "Segment". The "Category" column contains values like "Rural", "Accessory", and "Urban". The "Segment" column contains values like "Productivity", "Select", "Accessory", "Moderation", "Regular", "Extreme", "All Season", "Productivity", "Youth", and "Convenience". A context menu is open, and the "Remove Duplicates" option is highlighted with a red box.

	Category	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**

The screenshot shows the Power BI ribbon with the "Add Column" tab selected. Under the "General" section, the "Index Column" button is highlighted with a red box. Below it, the "From 1" button is also highlighted with a red box. To the right, there is a preview of a table with columns "Category" and "Segment".

	Category	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]

X ✓ fx	= Table.RenameColumns(#"Reordered Columns",{{"CatSegID", "Ca	
CampaignDim		
DateDim		
GeoDim		
ProductDim		
Sales		
CatSegDim	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	Segment	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	Segment
1	Rural	Productivity
2	Rural	Select
3	Accessory	Accessory
4	Urban	Moderation
5	Urban	Regular

Join Kind

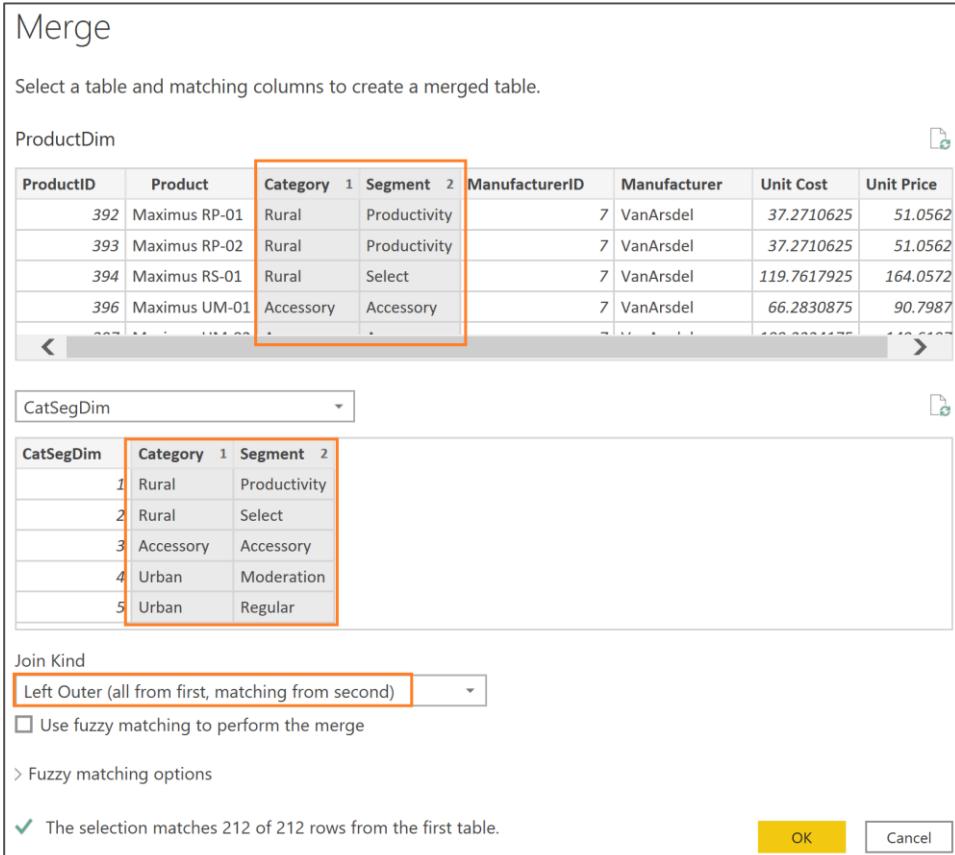
Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

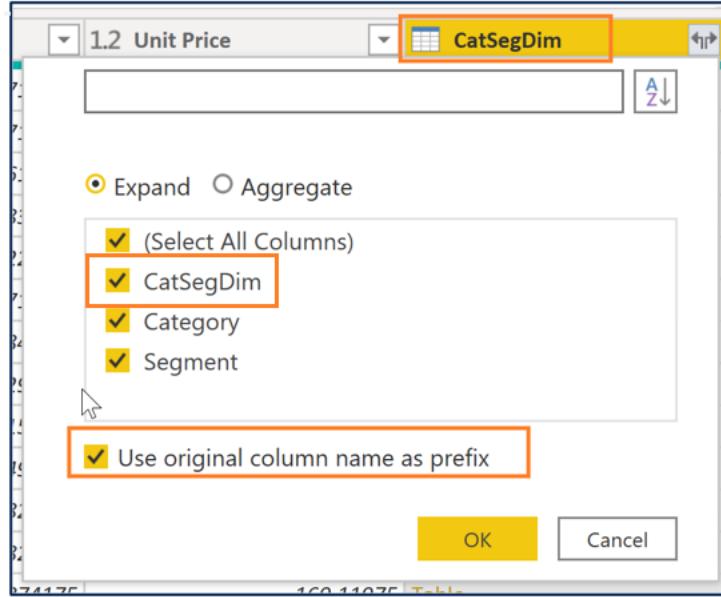
> Fuzzy matching options

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**

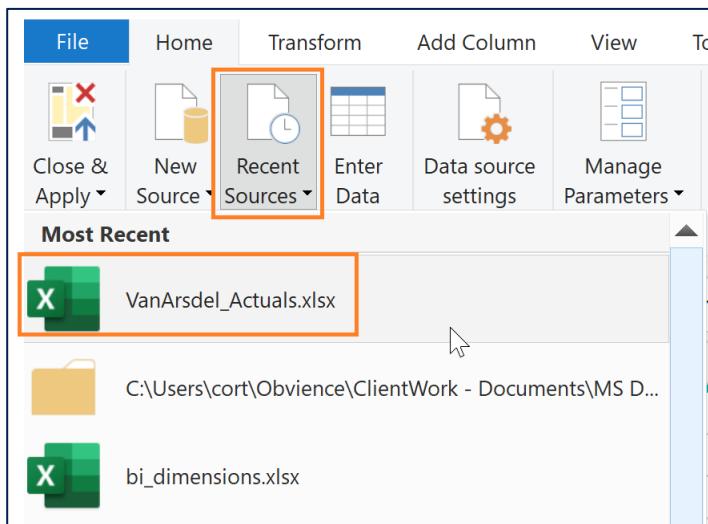
1	2	3	ProductID	1	2	CatSegID	A	B	C	Product	1	2	Unit Price	1	2	Unit Cost
1			392				1			Maximus RP-01			51.05625			37.2710625
2			393				1			Maximus RP-02			51.05625			37.2710625
3			394				2			Maximus RS-01			164.05725			119.7617925
4			396				3			Maximus UM-01			90.79875			66.2830875
5			397				3		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

Lab 2c: Create Customer dimension

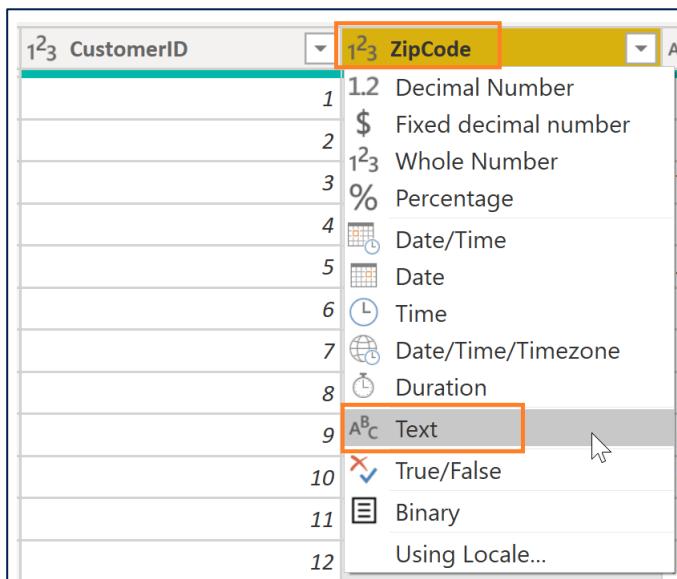
Task: Create a Customer dimension

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

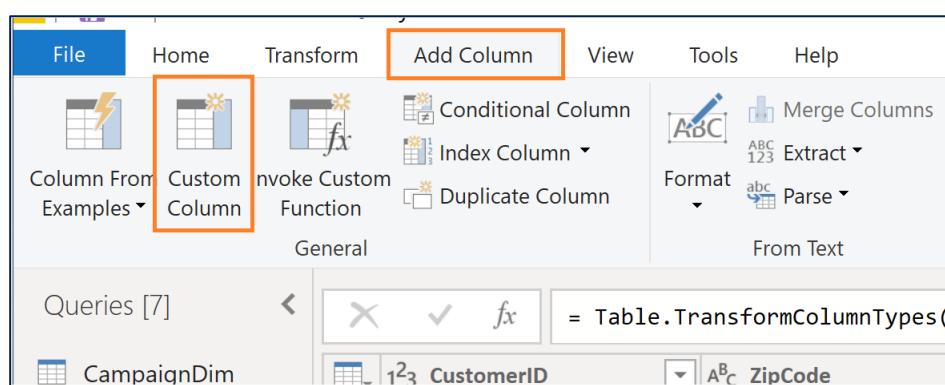
> Advanced options

OK

Cancel

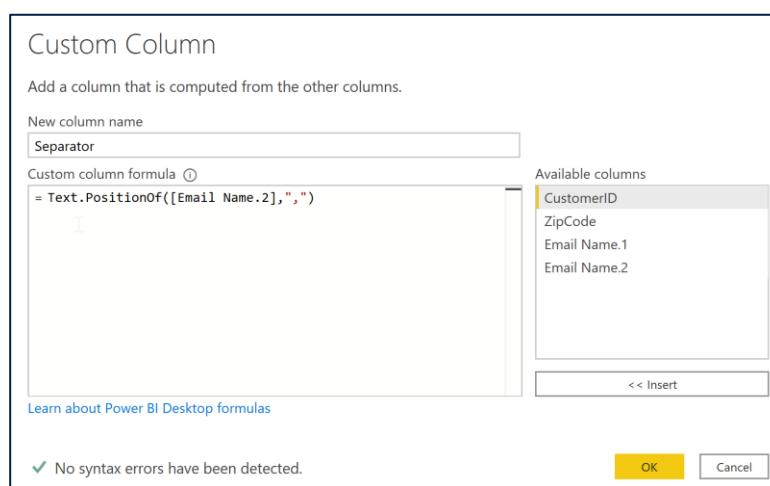
4. Add Column to find Text.PositionOf() the comma

a. Add Column > Custom Column



b. Name = "Separator"

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



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]***

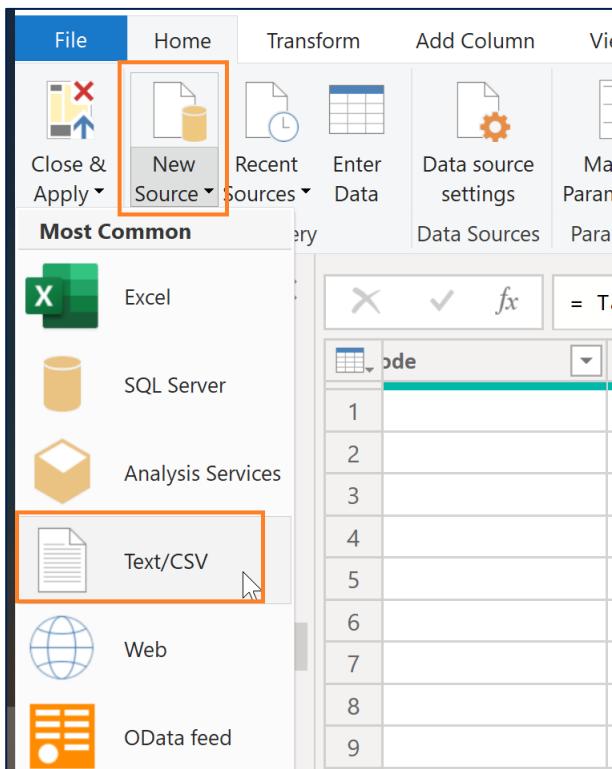
7. Remove Separator column

Lab 2d: Create Budget Fact table

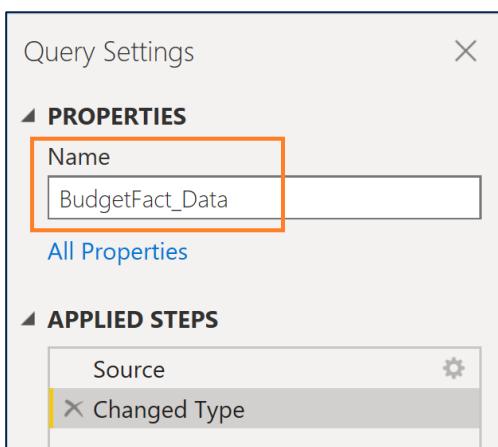
Task: Create Budget fact table

The estimated time to complete this lab is 45 minutes.

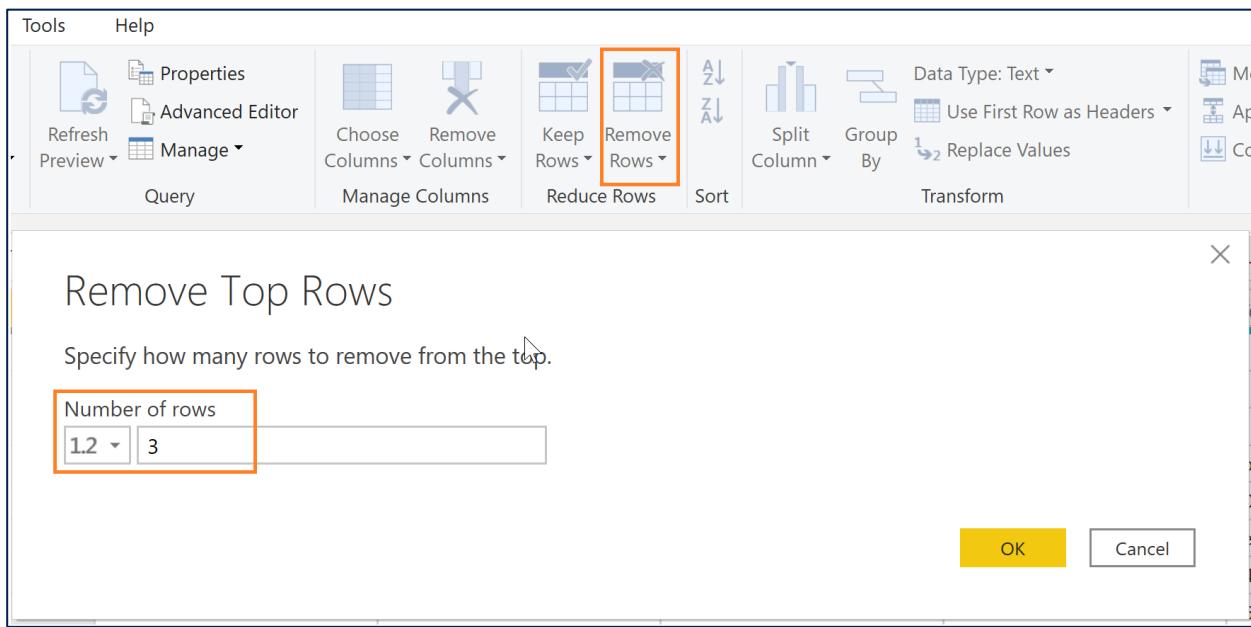
1. Import CSV document "C:/Power BI Adv_M/VanArsdel_Budget.csv"



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



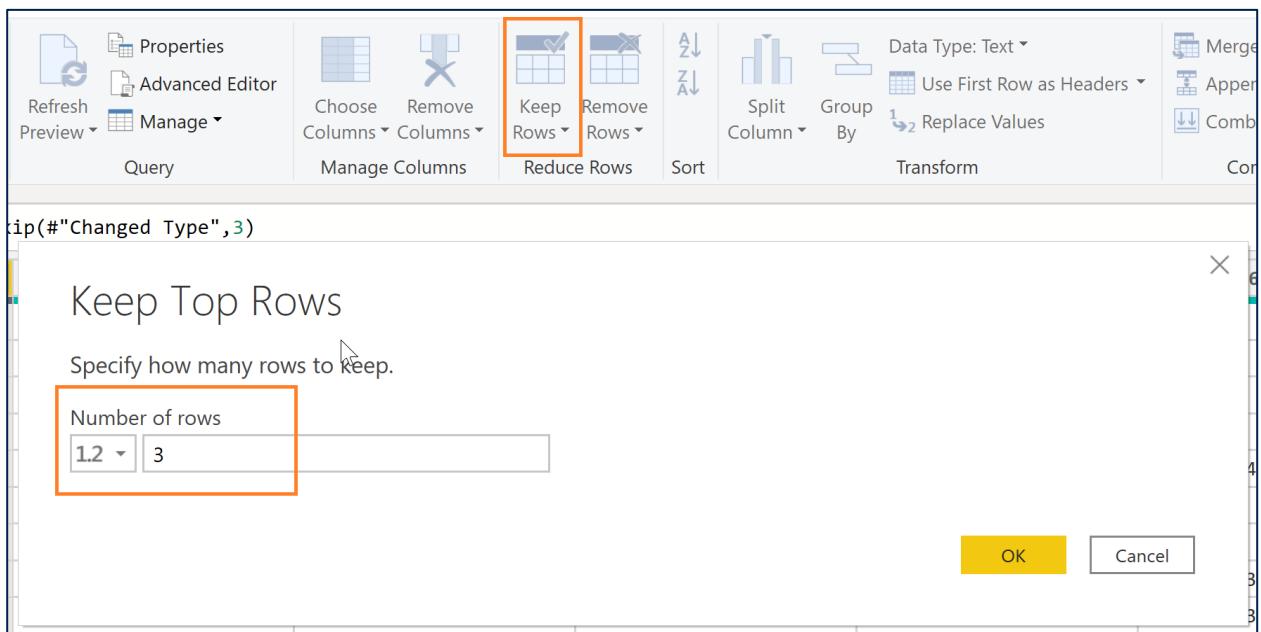
3. **Remove Rows** > Remove Top Rows, enter 3 (to remove the first 3 rows)



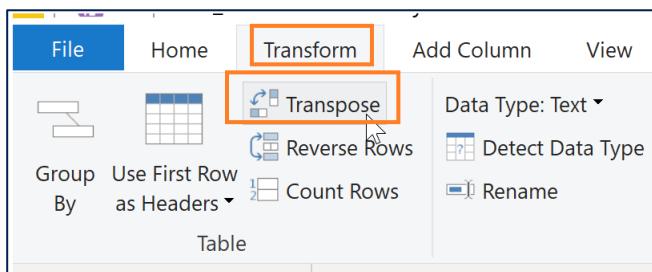
4. Duplicate query **BudgetFact_Data** rename to "**BudgetFact**"

The screenshot shows the Power BI 'Queries [8]' pane. On the left, there is a list of queries: CampaignDim, DateDim, GeoDim, ProductDim, Sales, CatSegDim, CustomerDim, and BudgetFact_Data. The 'BudgetFact_Data' query is selected and its context menu is open. The menu options include Copy, Paste, Delete, Rename, Enable load, Include in report refresh, Duplicate (which is highlighted with an orange box), Reference, Move To Group, Move Up, Move Down, Create Function..., Convert To Parameter, Advanced Editor, and Properties... . To the right of the menu, there is a preview grid showing data from the 'BudgetFact_Data' query.

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

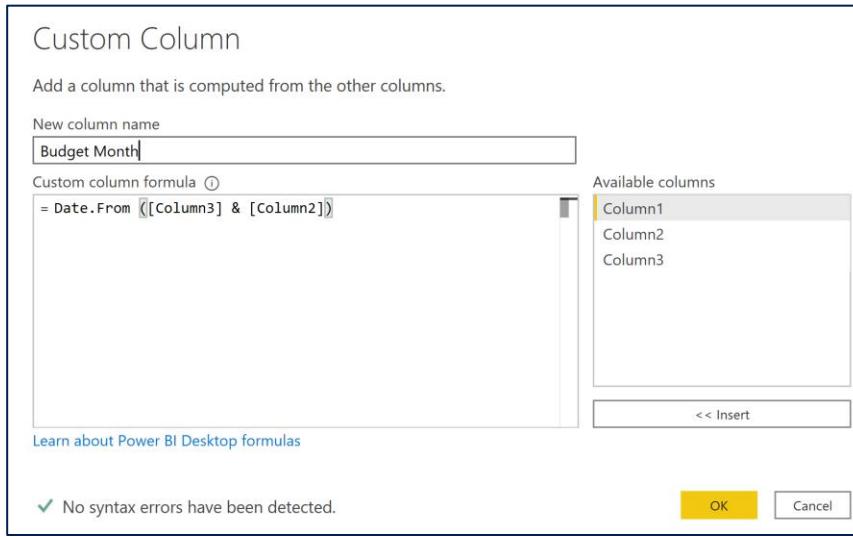


6. **Transform> Transpose**



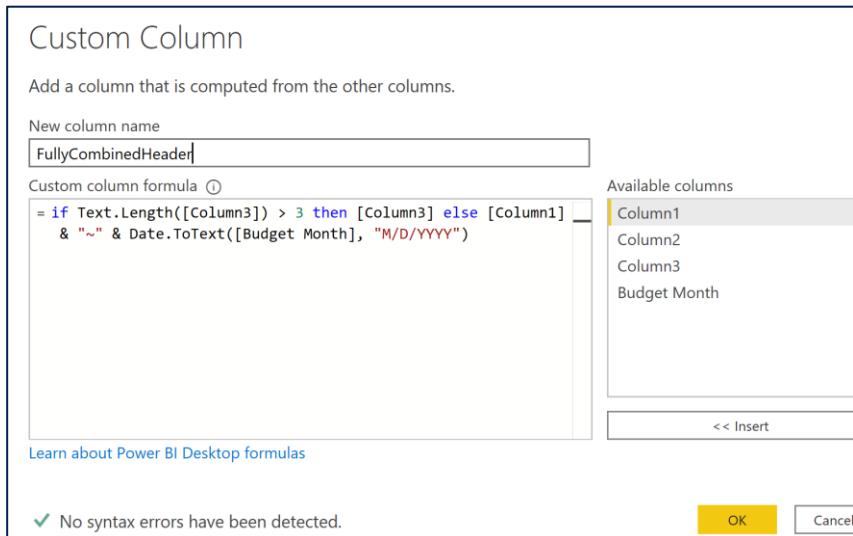
7. **Add Column** to combine month and year into a date a

- a. Add Column > **Custom Column**
- b. Name = "**Budget Month**"
- c. Formula = **Date.From ([Column3] & [Column2])**

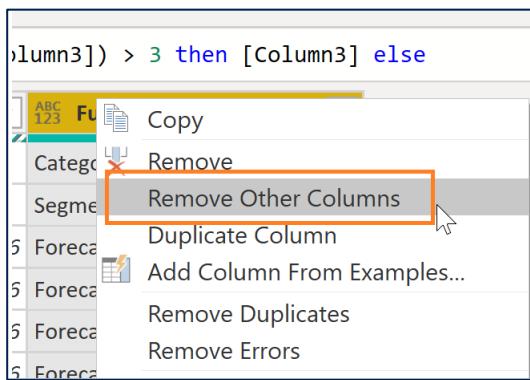


8. Add Column to combine Month and Scenario

- 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"



9. Remove all columns except for FullyCombinedHeader



10. Transform > Transpose to transpose back to wide

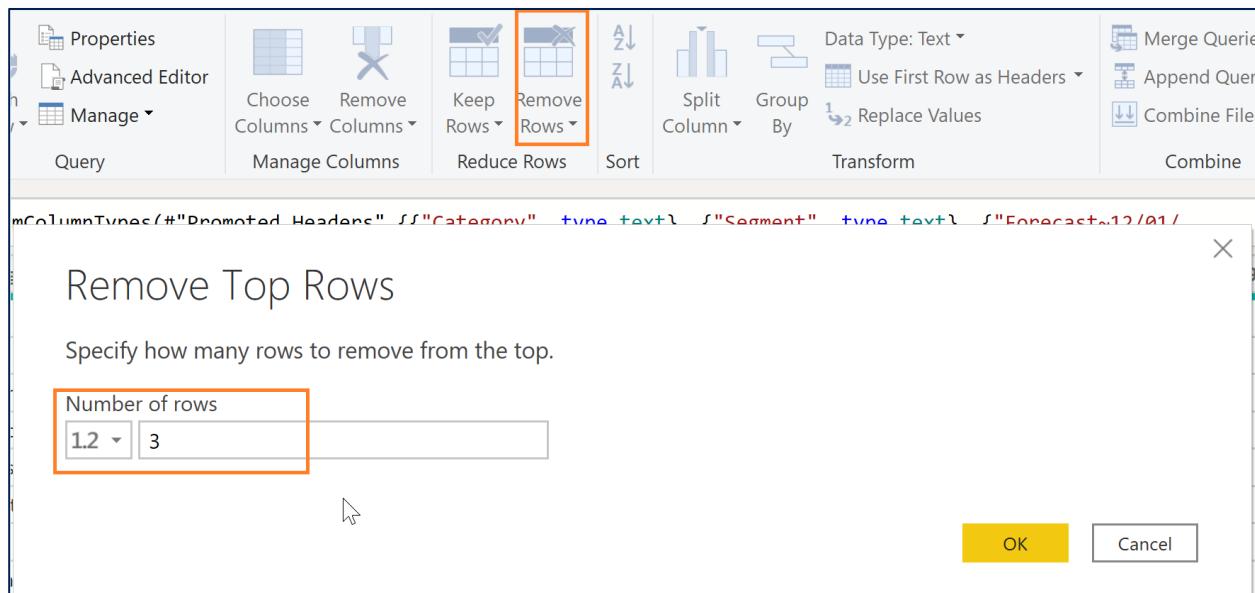
Column1	Column2	Column3	Column4
Category	Segment	Forecast~12/01/2016	Forecast~11/01/2016

11. Append query BudgetFact_Data

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

Category	Segment	Forecast~12/01/2016	Forecast~11/01/2016	Forecast~10/01/2016
Category	Segment	Dec	Nov	Oct
1		Forecast	Forecast	Forecast
2		2016	2016	2016

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

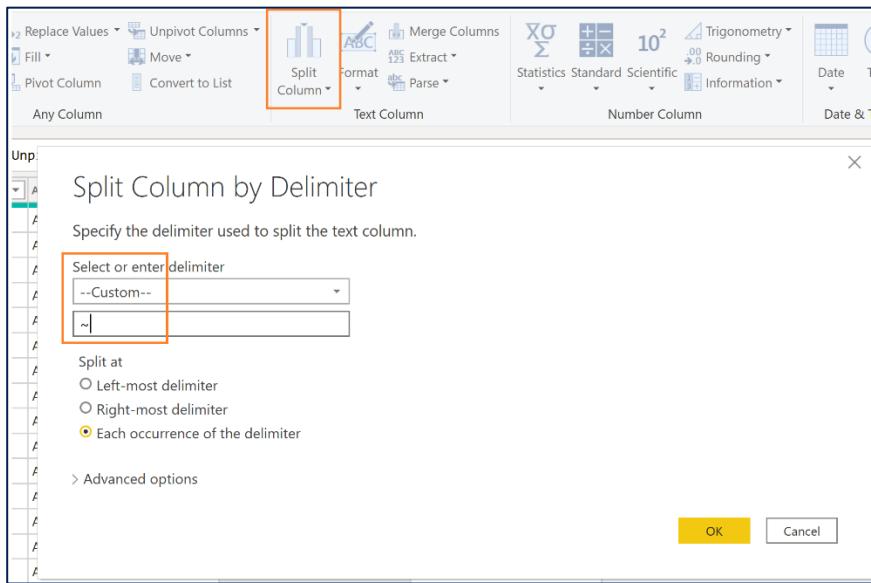


14. Highlight **Category** and **Segment** and **Transform> Unpivot Other Columns**

The screenshot shows the Power BI desktop interface with the 'Transform' ribbon tab selected. In the 'Transform' group, the 'Unpivot Columns' and 'Unpivot Other Columns' buttons are highlighted with orange boxes. Below the ribbon, a table view displays columns 'Category' and 'Segment' highlighted with an orange box. The formula bar at the top shows the formula = Table.Skip(#"Changed Type1",3). The left sidebar lists various dimensions and fact tables.

Category	Segment	Forecast^12/01/2016
Accessory	Accessory	44190.57888
Mix	All Season	11442.14474
Mix	Productivity	19538.89812
Rural	Select	311.708775
Urban	Convenience	120710.4406
Urban	Extreme	20868.84072
Urban	Moderation	251155.7122
Urban	Regular	689.7969225
Youth	Youth	3931.03074

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



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

ABC Scenario	Date	A ^B _C 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	Segment	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	Segment
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.

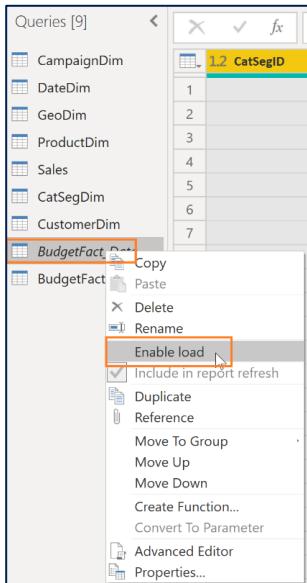
OK Cancel

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

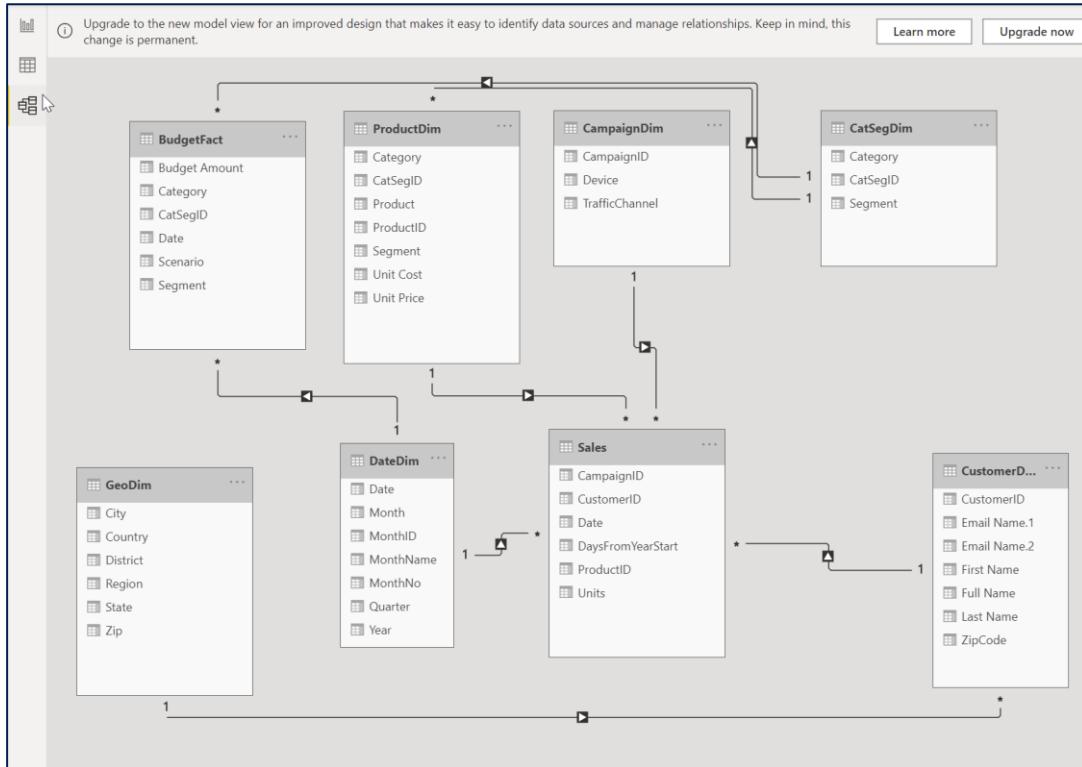


Lab 3: Create relationships between tables

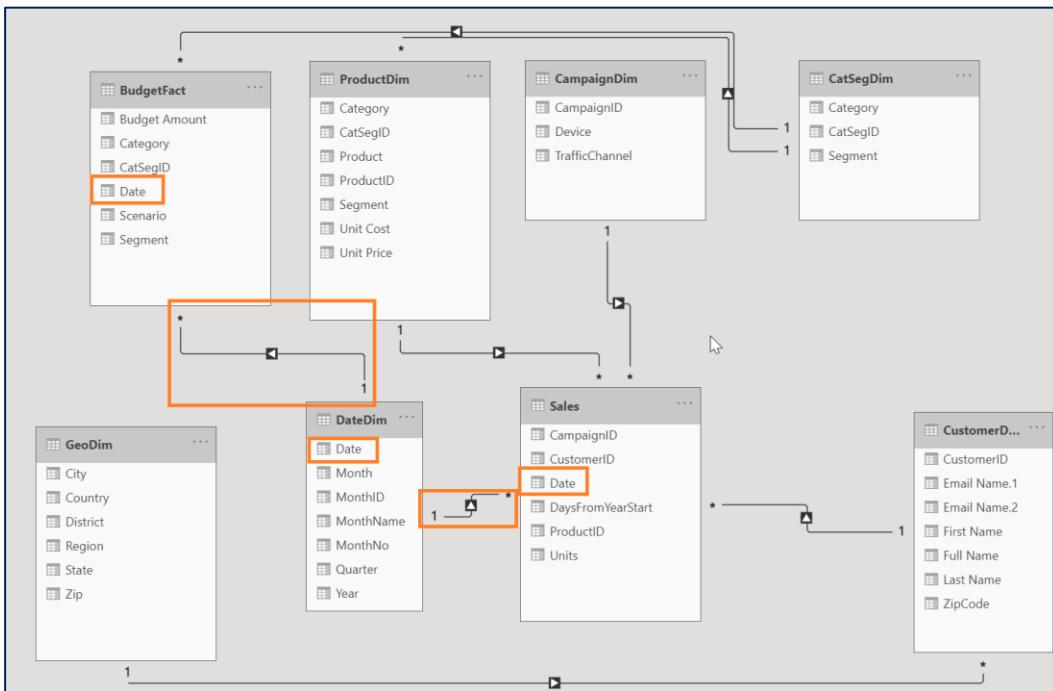
Task: Create relationships between multiple tables.

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

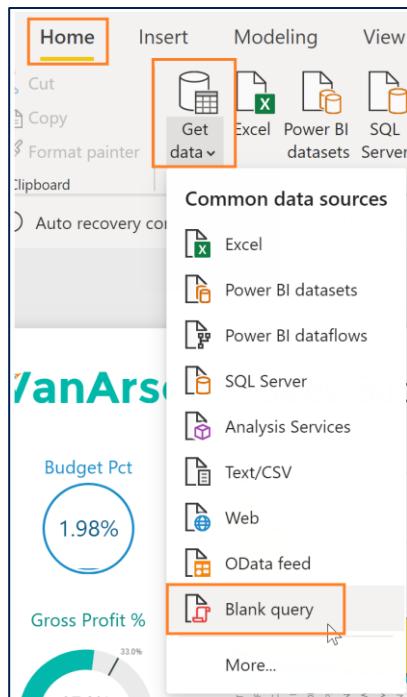


Lab 4a: Create dynamic path to excel source file

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

A screenshot of the Power BI Query Editor. On the left, the 'Queries [19]' pane shows a tree structure with 'Inputs [3]', 'Helper Queries [3]', 'Data Model [2]', and 'Fact [5]'. Under 'Helper Queries', 'Actuals_Path' is selected. The main area shows a table with 14 rows and columns for 'Name', 'Data', 'Item', 'Kind', and 'Hidden'. The formula bar at the top contains the DAX formula: `= if Text.StartsWith(FilePath,"http")`. On the right, the 'Query Settings' pane is open, showing 'PROPERTIES' with 'Name' set to 'Actuals_Path' and 'APPLIED STEPS' with 'Source' highlighted.

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

Queries [19]

= Actuals_Path

#	Name	Type	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	

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]

- DateDim
- CampaignDim
- GeoDim
- ProductDim
- CatSeqDim

Query Settings

Properties

- Name: CampaignDim
- All Properties

Applied Steps

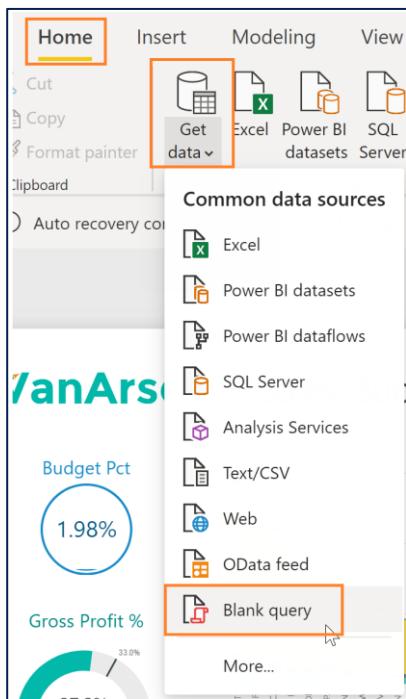
- Source
- Navigation
- Changed Type

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

A screenshot of the Power BI Data view. It shows a table with 15 rows and 5 columns. The first column is 'Column1' and the second is 'Column2'. The formula bar at the top has the text '= if Text.StartsWith(FilePath, "http")'. The table data includes:
Row 1: Budget Spreadsheet for VanArsdel
Row 2:
Row 3:
Row 4: Forecast, Forecast, Forecast
Row 5: 2016, 2016, 2016
Row 6: Category, Segment
Row 7: Accessory, Accessory
Row 8: Mix, All Season
Row 9: Mix, Productivity
Row 10: Rural, Select
Row 11: Urban, Convenience
Row 12: Urban, Extreme
Row 13: Urban, Moderation
Row 14: Urban, Regular
Row 15: Youth, Youth
On the right side, there are 'Query Settings' and two sections: 'PROPERTIES' and 'APPLIED STEPS'. In 'PROPERTIES', 'Name' is set to 'Budget_Path'. In 'APPLIED STEPS', 'Source' is listed.

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

The screenshot shows the Power BI Query Editor interface. On the left, the 'Queries [19]' pane lists various inputs and queries, including 'ActualsFile (VanArsdel_Actuals.xlsx)', 'Path (C:\PowerBI_Adv_M)', 'BudgetFile (VanArsdel_Budget.csv)', 'Helper Queries [3]', 'Data Model [2]', and 'Fact [5]'. The 'BudgetFact' query is currently selected and highlighted with a gray background.

The main area displays the 'BudgetFact' query results in a table format. The table has five columns: Column1, Column2, Column3, Column4, and Column5. The first few rows show header information, followed by data points such as Category (Accessory, Mix, Mix, Rural, Urban, Urban, Urban, Urban, Youth), Segment (Dec, All Season, Productivity, Select, Convenience, Extreme, Moderation, Regular, Youth), and various numerical values (e.g., 44190.57888, 11442.14474, 19538.89812, 311.708775, 120710.4406, 20868.84072, 251155.7122, 689.7969225, 3931.03074).

The 'Applied Steps' pane on the right is open, showing a list of steps taken during the query's creation. The 'Source' step is highlighted with an orange border, indicating it is the current step being viewed.

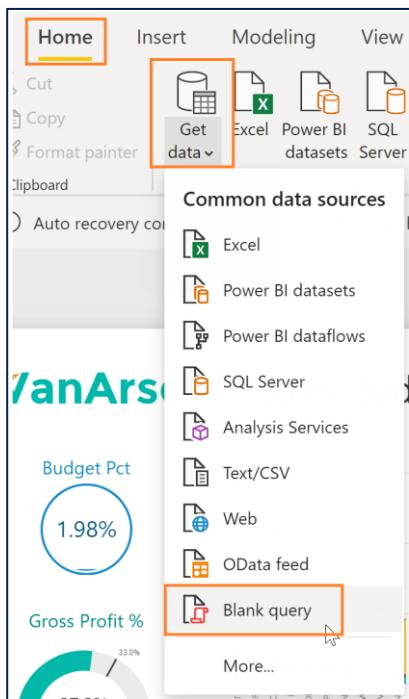
Column1	Column2	Column3	Column4	Column5
Budget Spreadsheet for VanArsdel				
1			Forecast	Forecast
2			2016	2016
3				2016
4				Oct
5				
6 Category	Segment	Dec	Nov	
7 Accessory	Accessory	44190.57888	50598.81566	54740.5709
8 Mix	All Season	11442.14474	14120.78693	18109.64804
9 Mix	Productivity	19538.89812	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

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

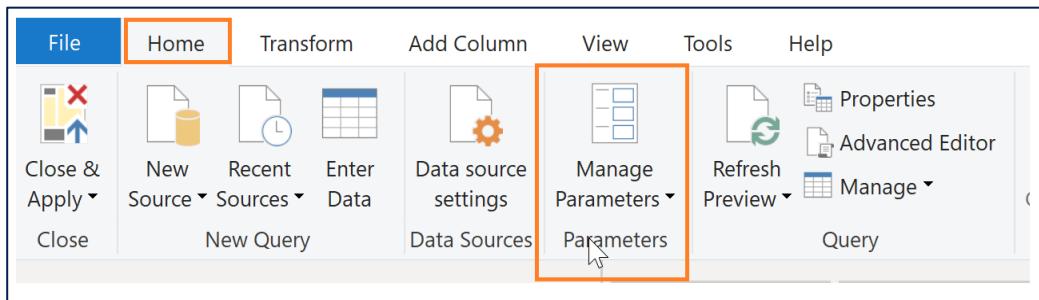


- a. Query Name: "**fn_DaySinceYearStart**"
- b. In Advanced Editor copy in text from **Number_Days.txt**

```
fn_DaySinceYearStart
let
    Source = (TransactionDate as date) => let
        YearStart = #date(Date.Year(TransactionDate),1,1),
        #"DateDiff" = Duration.From(TransactionDate-YearStart),
        #"NumberDays" = Duration.Days(#"DateDiff") + 1
    in
        #"NumberDays"
in
    Source
```

The screenshot shows the 'Advanced Editor' window of Power BI. The title bar says 'fn_DaySinceYearStart'. The code itself is a let expression. It starts with 'let' followed by a function definition '(TransactionDate as date) => let'. Inside this, it calculates 'YearStart' as the first day of the year of 'TransactionDate', then calculates 'DateDiff' as the duration from 'YearStart' to 'TransactionDate', and finally calculates 'NumberDays' as the number of days in 'DateDiff' plus one. The entire code block is enclosed in a large orange rectangle.

2. Create a new Parameter



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

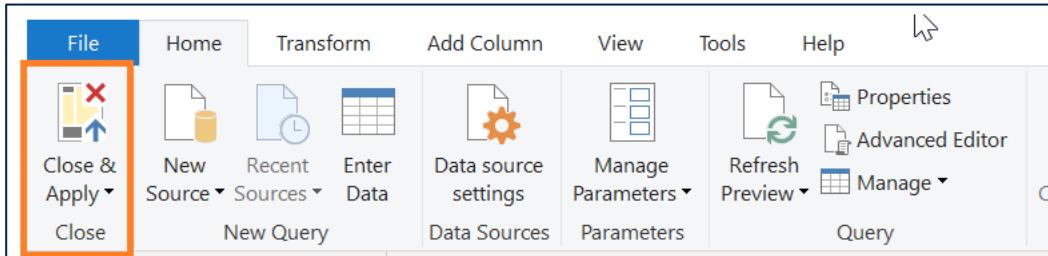
A screenshot of the 'Manage Parameters' dialog box. On the left is a list of existing parameters: TransactionDate (selected), ActualsFile, Path, and BudgetFile. On the right, for the selected 'TransactionDate' parameter, there are fields for 'Name' (set to 'TransactionDate'), 'Description' (empty), 'Required' (checked), 'Type' (set to 'Date'), 'Suggested Values' (set to 'Any value'), and 'Current Value' (set to '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**

	CustomerID	CampaignID	Units	DaysFromYearStart
1	9/25/2011	70283	22	1
2	9/25/2011	195385	22	1
3	5/14/2012	212645	22	1
4	5/14/2012	70666	22	1
5	5/14/2012	114459	22	1
6	5/14/2012	221670	22	1
7	5/14/2012	26974	22	1
8	6/3/2012	268392	22	1

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



5. Close and Save the pbi file.

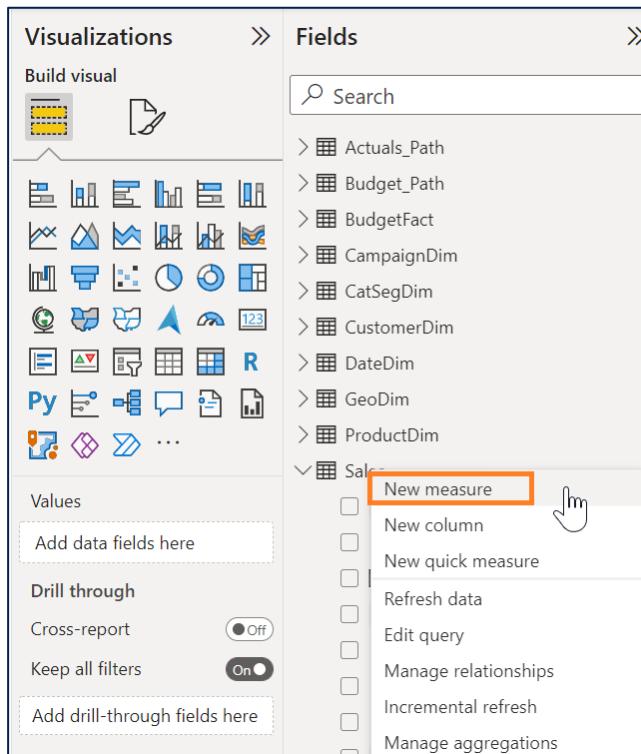
Lab 5: Create new measures and columns

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

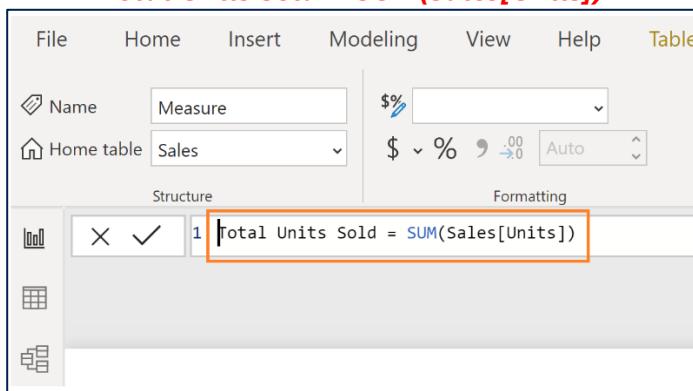
The estimated time to complete this lab is 30 minutes.

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

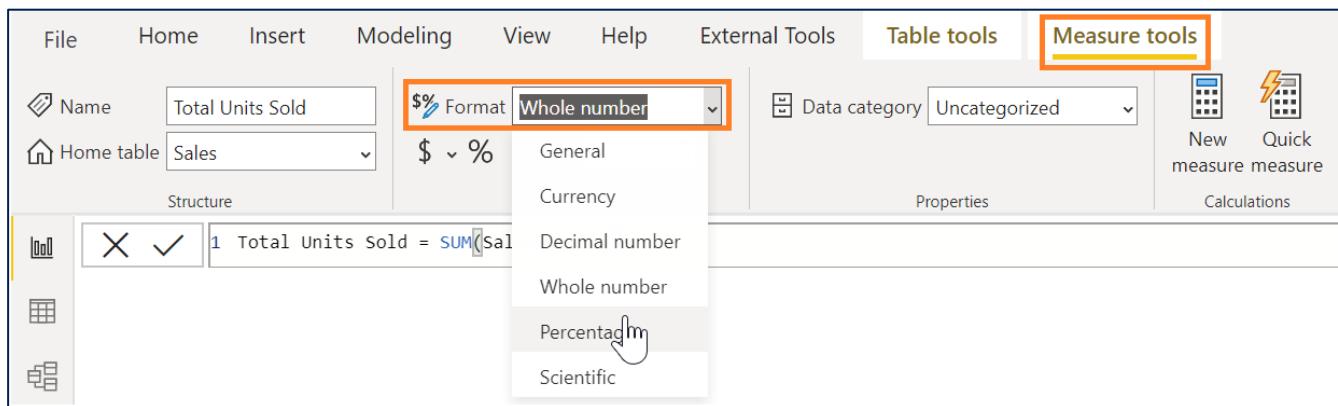


2. In the formula bar enter:

Total Units Sold = SUM(Sales[Units])

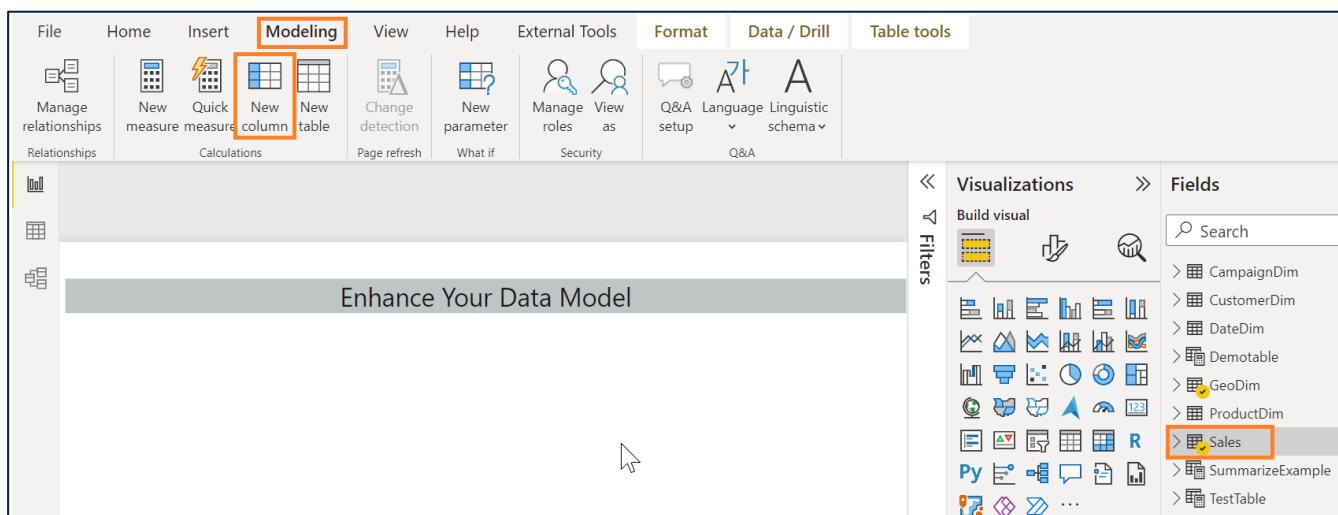


3. From the **Measure tools** ribbon select **Format -> Whole number** to format the measure



Task 2: Create calculated column that combines Category and TrafficChannel

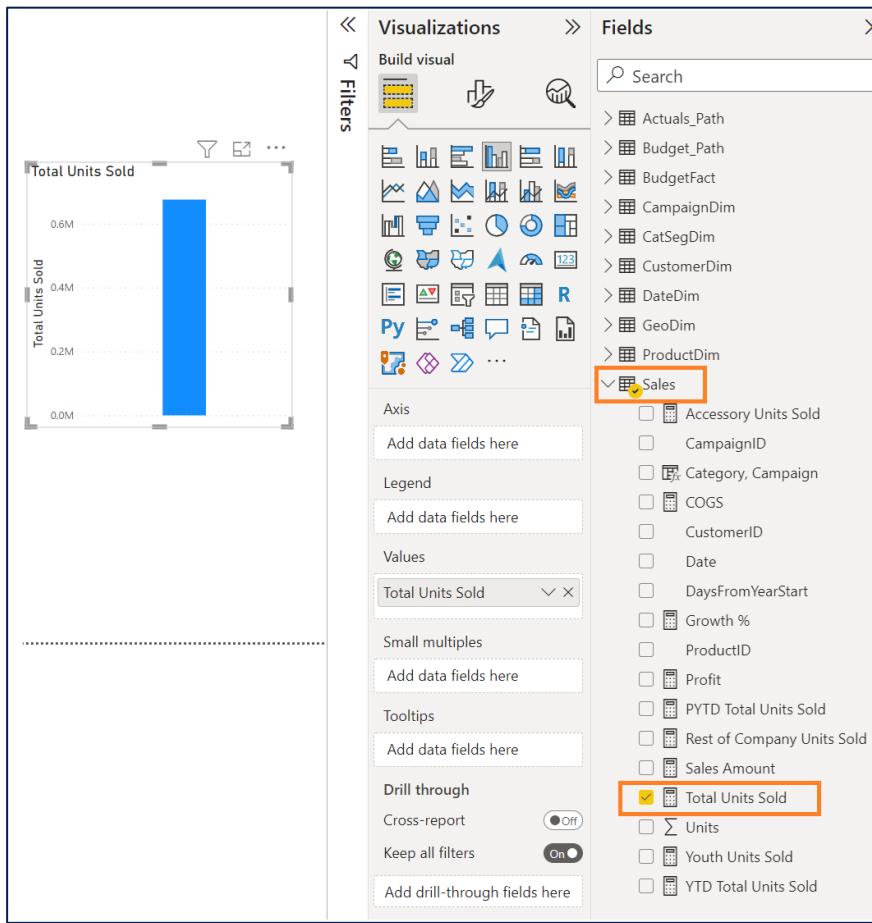
4. Select **Sales** table. From the ribbon select **Modeling -> New Column**.



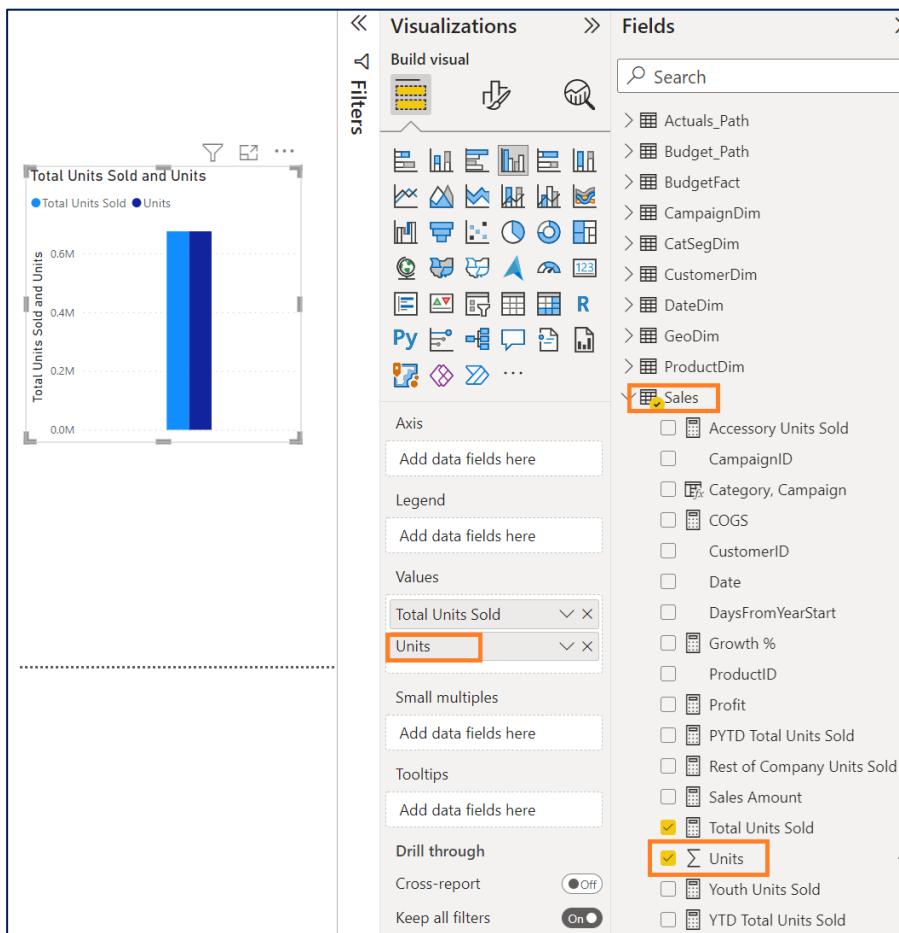
5. In the formula bar enter:

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

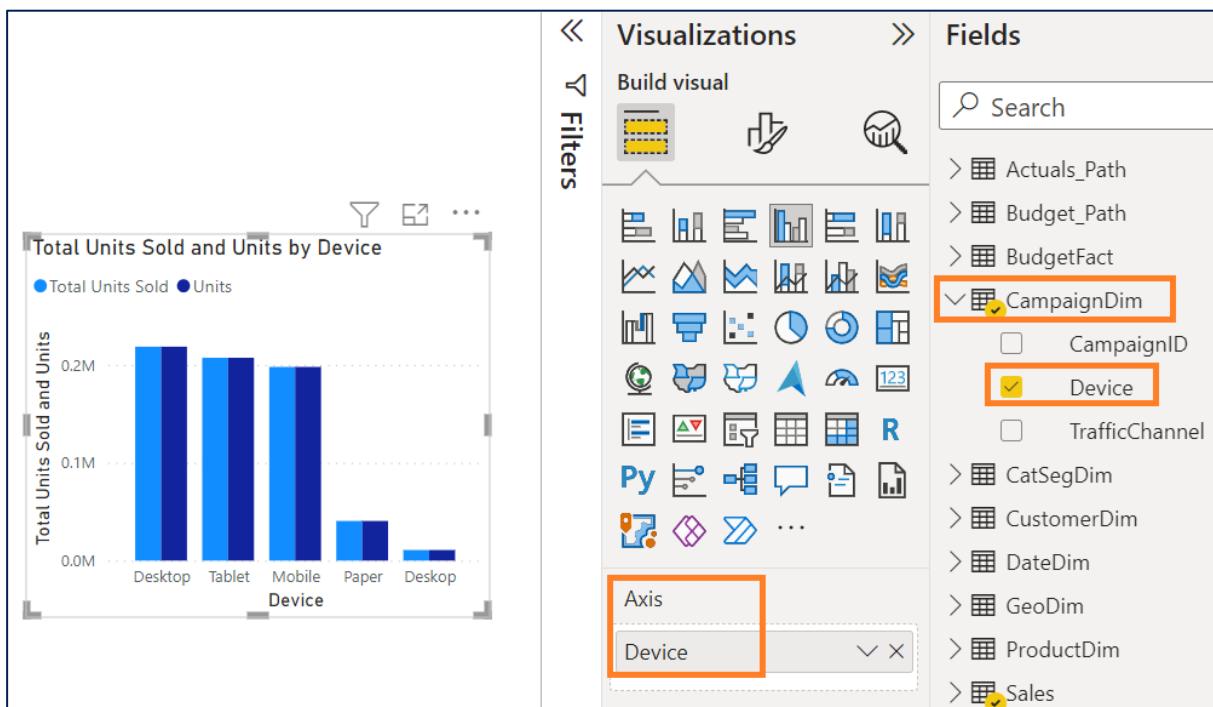
6. Drag newly created **Total Units Sold** measure to the canvas. A clustered column chart is created



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



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

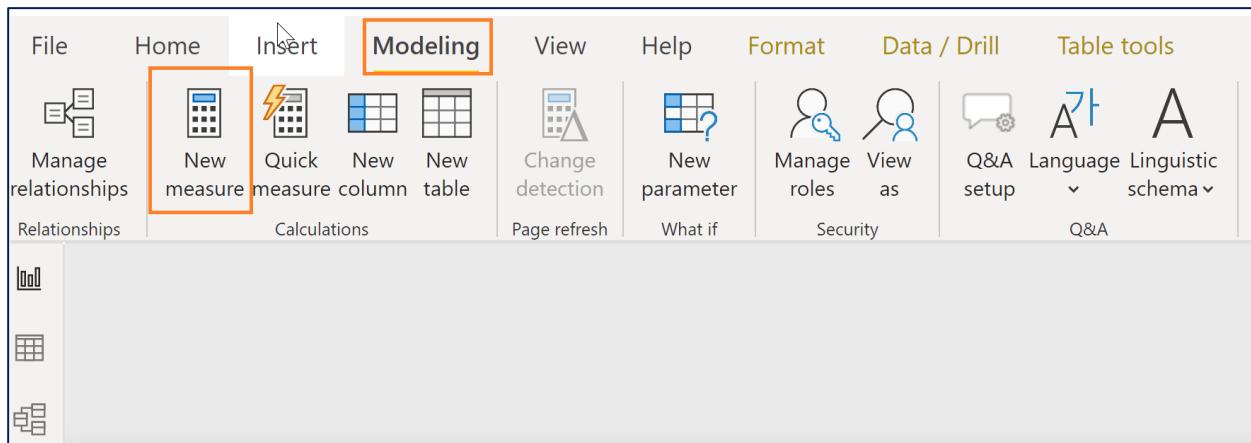


Lab 6: Create a report for the VP in charge of the Youth and Accessory Segments

Task 3: Create three new measures and a PowerBI visualization

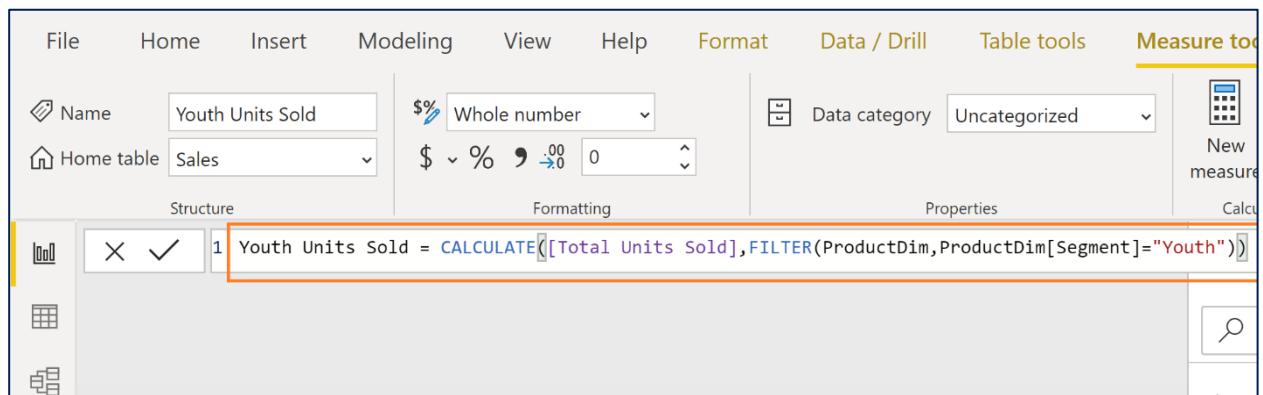
The estimated time to complete this lab is 45 minutes.

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

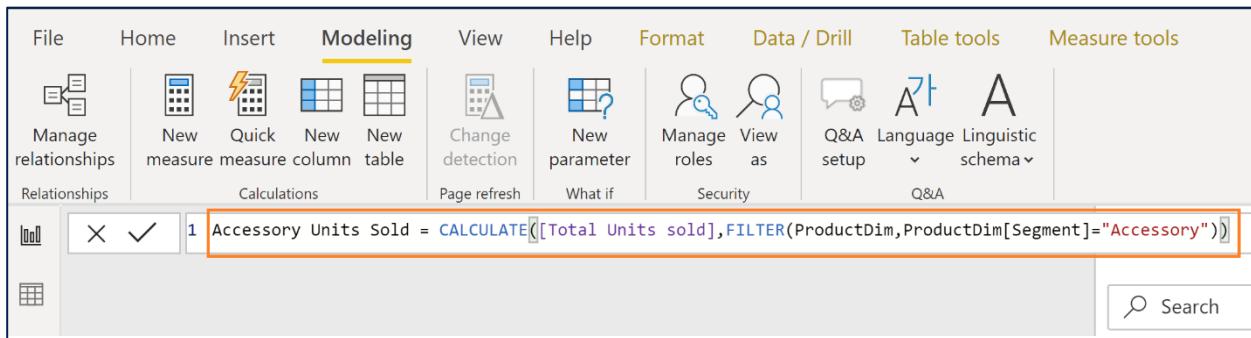


2. Create 3 measures:

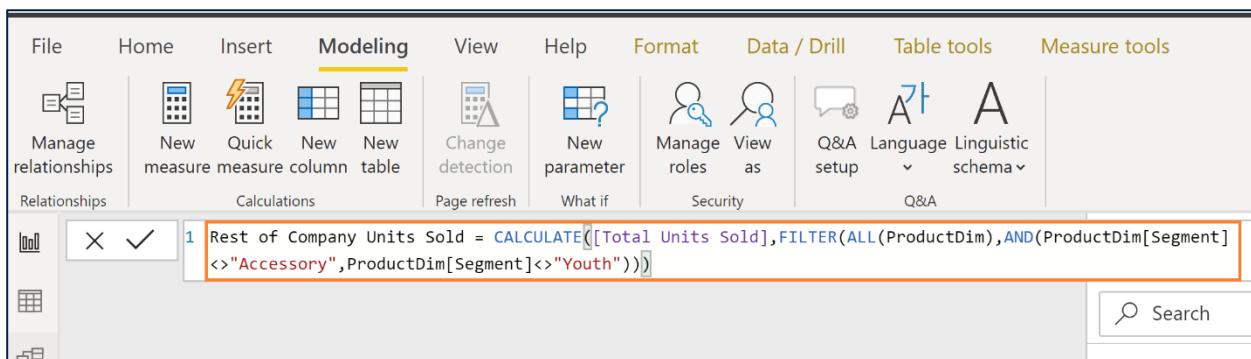
Youth Units Sold = CALCULATE([Total Units Sold], FILTER(ProductDim, ProductDim[Segment] = "Youth"))



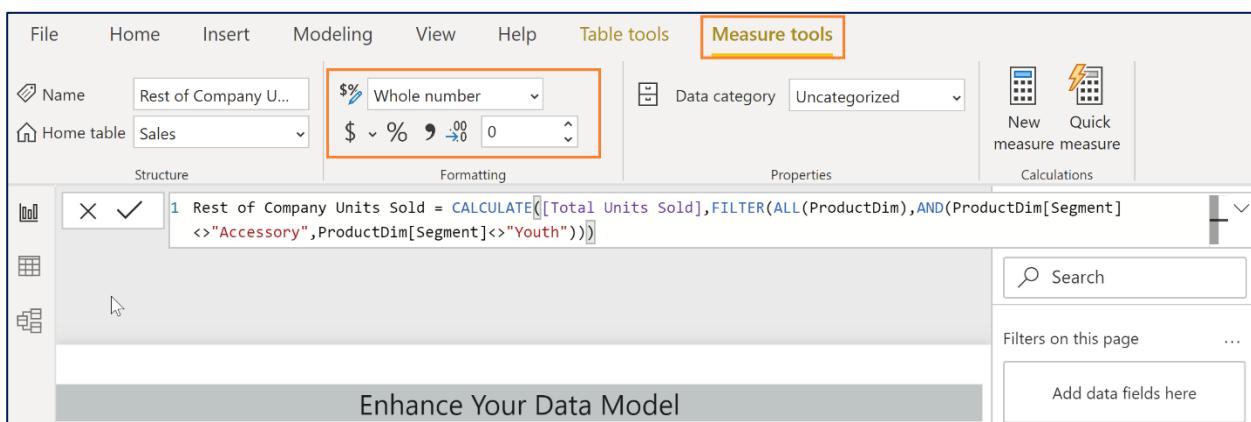
Accessory Units Sold = CALCULATE([Total Units Sold], FILTER(ProductDim, ProductDim[Segment] = "Accessory"))



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**

The screenshot shows the Power BI 'Build visual' interface. On the left, there is a table visualization with the following data:

Device	Rest of Company Units Sold	Youth Units Sold	Accessory Units Sold
Desktop	9931	222	653
Desktop	201335	4,933	12412
Mobile	182167	4,427	11420
Paper	37240	908	2376
Tablet	189885	5,151	12308
Total	620558	15,641	39169

The 'Fields' pane on the right lists various dimensions and measures. The following fields are selected (highlighted with orange boxes):

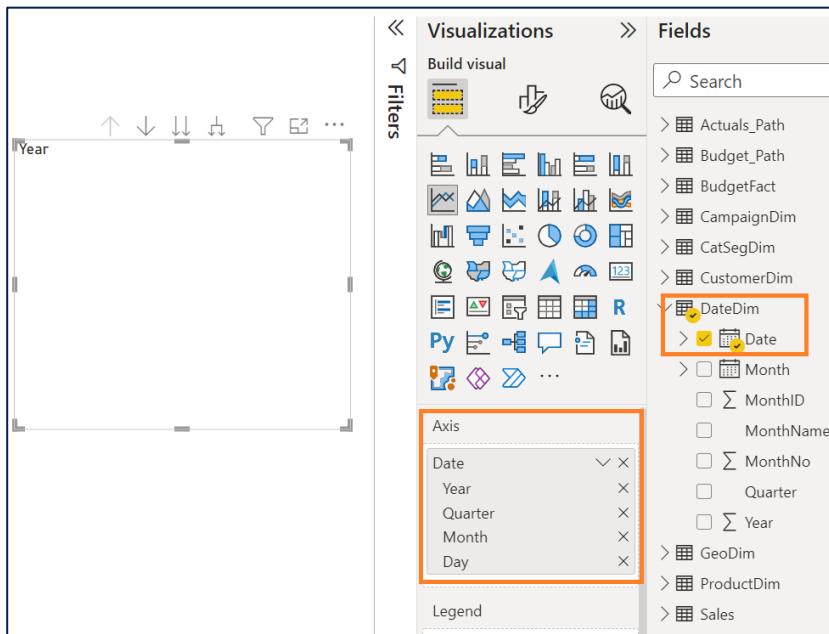
- Sales
 - Accessory Units Sold
 - Rest of Company Units Sold
 - Youth Units Sold

5. Select **Line Chart** visual.

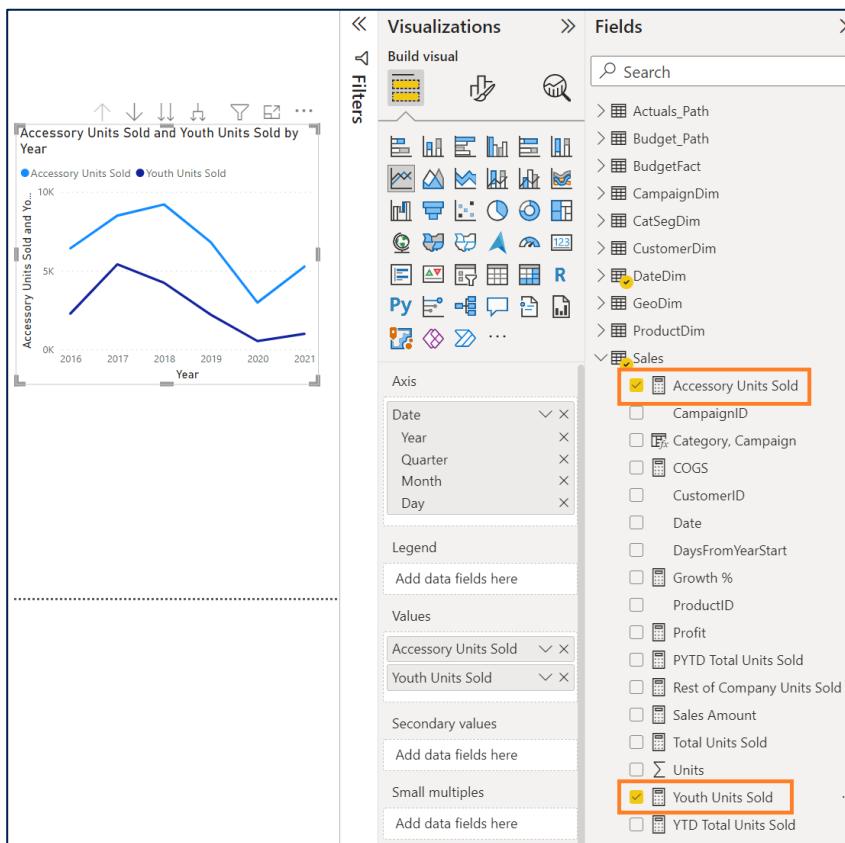
The screenshot shows the Power BI 'Build visual' interface. On the left, there is a line chart visualization with a single data series. The 'Fields' pane on the right lists various dimensions and measures. The following fields are selected (highlighted with orange boxes):

- Visualizations
 - Line chart

6. Select **Date** from Date table

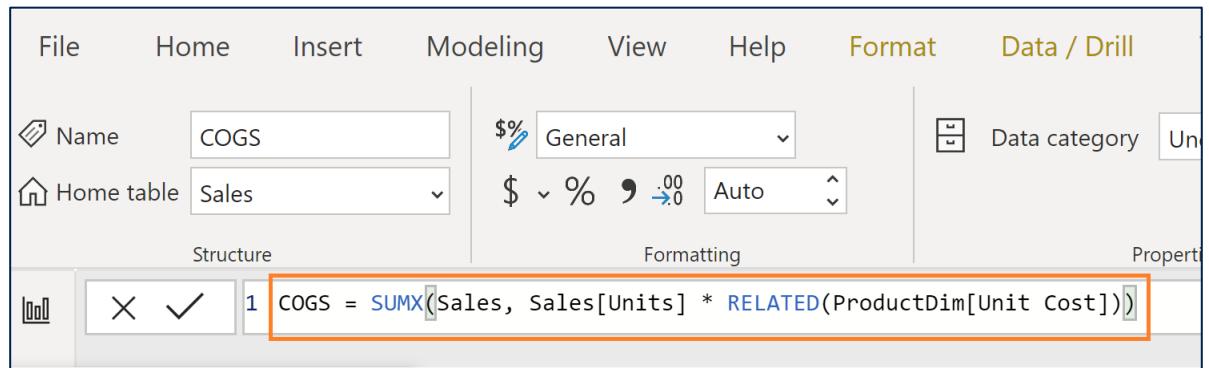


7. Select Youth Units Sold and Accessory Units sold measures.

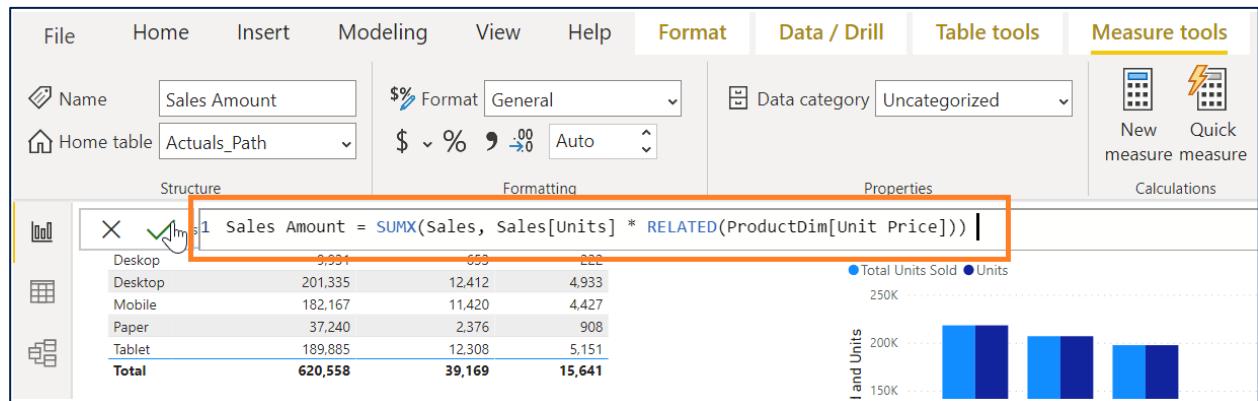


8. Create following measures and use a visual to analyze data.

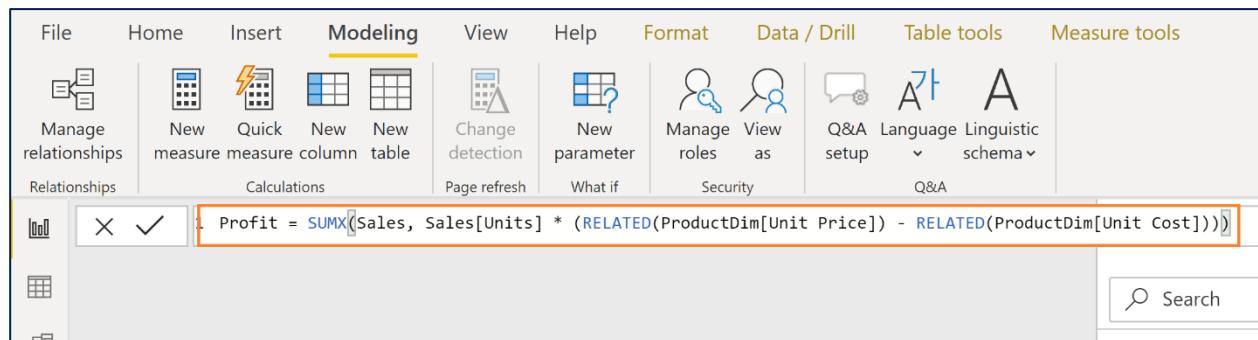
$$\text{COGS} = \text{SUMX}(\text{Sales}, \text{Sales}[Units]) * \text{RELATED}(\text{ProductDim}[Unit Cost]))$$



Sales Amount = $\text{SUMX}(\text{Sales}, \text{Sales}[Units] * \text{RELATED}(\text{ProductDim[Unit Price]}))$



Profit = $\text{SUMX}(\text{Sales}, \text{Sales}[Units] * (\text{RELATED}(\text{ProductDim[Unit Price]}) - \text{RELATED}(\text{ProductDim[Unit Cost]})))$



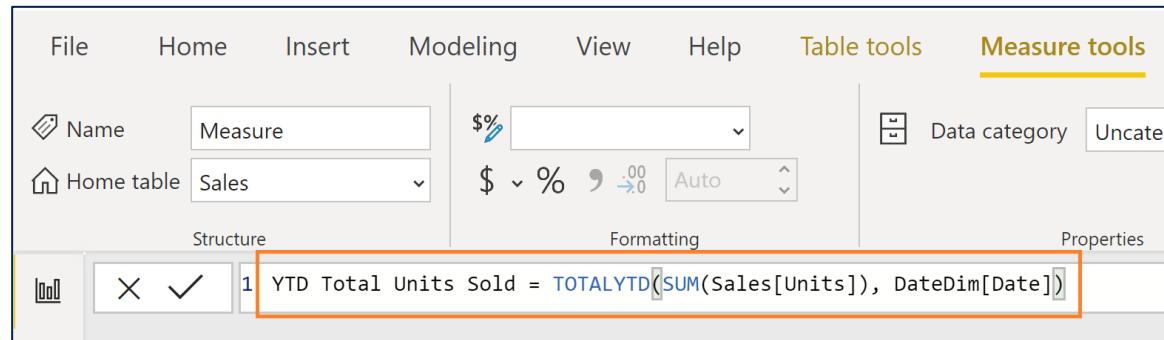
Lab 7a: Performance Best Practices

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

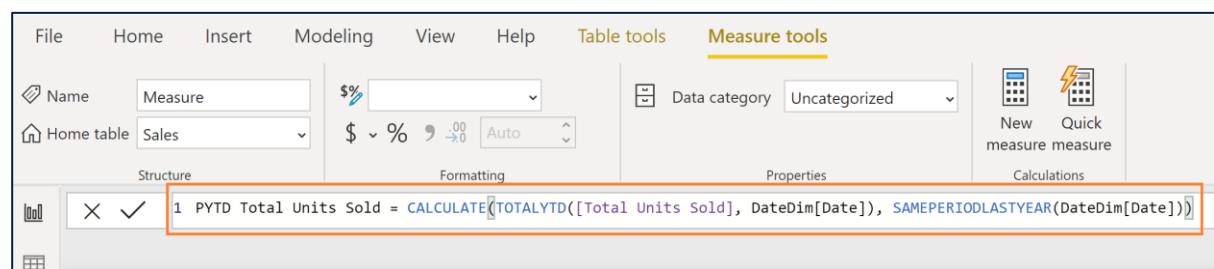
The estimated time to complete this lab is 30 minutes.

1. Create the three measures below.

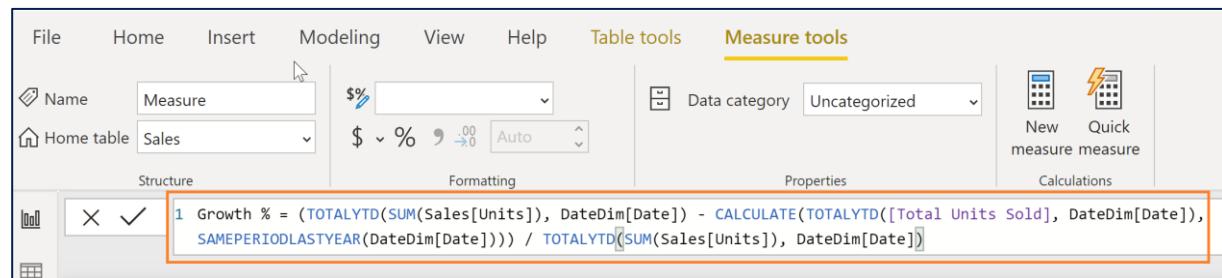
YTD Total Units Sold = TOTALYTD(SUM(Sales[Units]), DateDim[Date])



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]

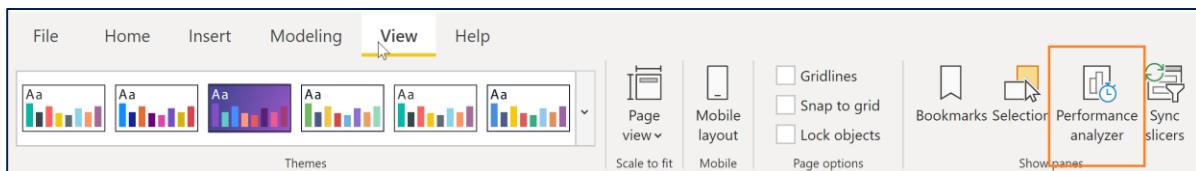
3. **Challenge:** how were the formulas changed to increase performance?

Lab 7b: Using Performance Analyzer

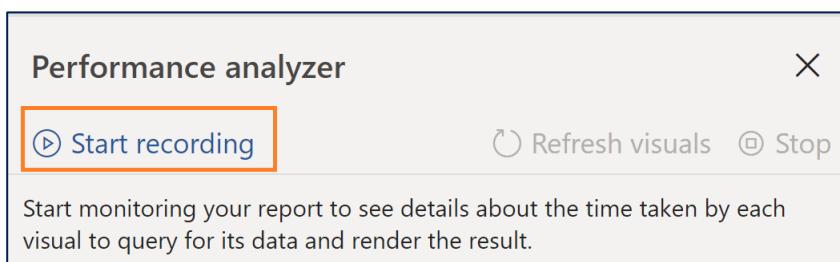
Task: Use performance analyzer to test the performance of individual charts

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

Name	Duration (ms)
⌚ Recording started (6/23/2020 10:16:50 AM)	-
⌘ Cross-highlighted	-
⊕ Total Units Sold by Year	492
⊕ Total Units Sold by Category	484
⊕ Total Units Sold by Email	483
⊕ Table	1015

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 FUNCITONALITY 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.