



Power BI

Fundamentals of

Power Query

Power Query Labs

Overview

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

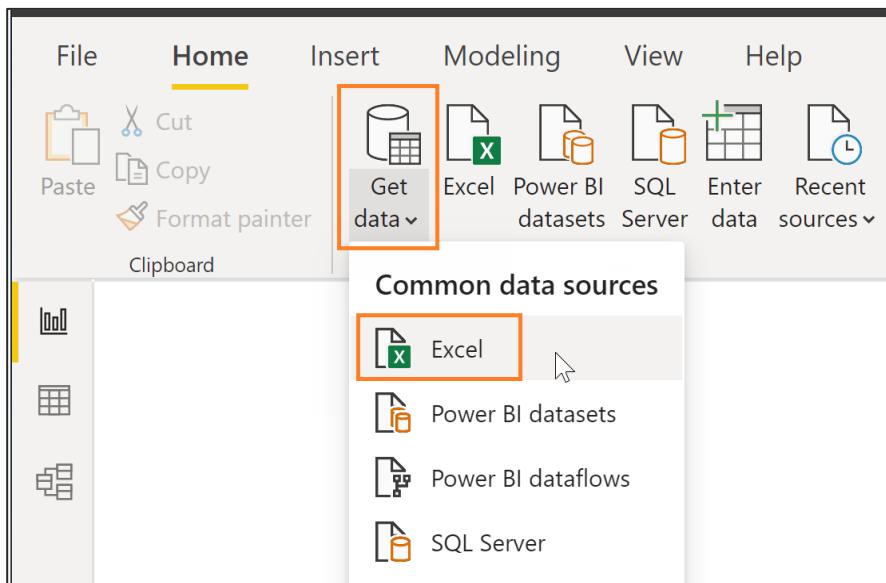
In this lab you will learn how to upload multiple tables from a single data source. As well as create a new dimension for the model and enhance an existing dimension. You will learn how to create a fact table for a budget. You will also learn how to create parameters and a dynamic path to your data source.

Lab 01: 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 10 minutes.

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



2. Select the following Tables (All but CustomerDim), then click, "Transform Data"
 - a. **CampaignDim**
 - b. **GeoDim**
 - c. **ProductDim**
 - d. **DateDim**
 - e. **Sales**

** Navigator

The screenshot shows the Power BI Navigator window. On the left, there's a tree view of the data model with several dimensions selected: CampaignDim, DateDim, GeoDim, ProductDim, and Sales. The Sales table itself is shown on the right, containing columns: ProductID, Date, CustomerID, CampaignID, and Units. There are 639 rows of data. At the bottom of the window are three buttons: Load (yellow), Transform Data (red box), and Cancel.

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

This screenshot shows the 'Transform Column Types' dialog for the 'Zip' column in the 'GeoDim' table. The column currently has a 'Decimal Number' type. A dropdown menu is open, showing options like 'Text', 'True/False', and 'Binary'. The 'Text' option is highlighted with a cursor over it.

** 4. Because the Data Type was automatically selected at import, you'll need to "Replace Current" to overwrite the automatic data type selection

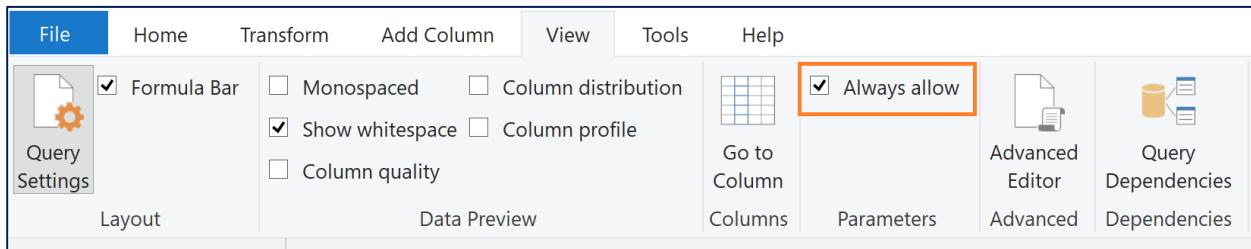
This screenshot shows a confirmation dialog titled 'Change Column Type'. It asks if the user wants to replace the existing conversion or preserve it. At the bottom are three buttons: 'Replace current' (yellow), 'Add new step', and 'Cancel'.

Lab 02.A : Create CatSegDim

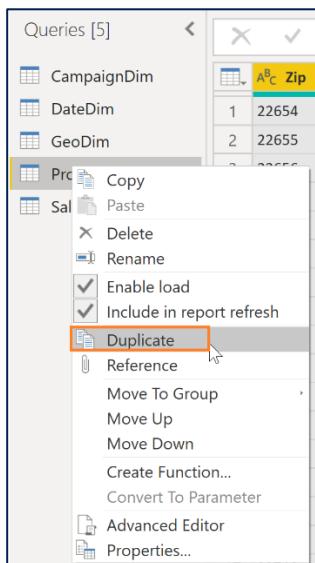
Task: Create a dimension using Category and Segment attributes from the Product dimension **The estimated time to complete this lab is 10 minutes.**

**

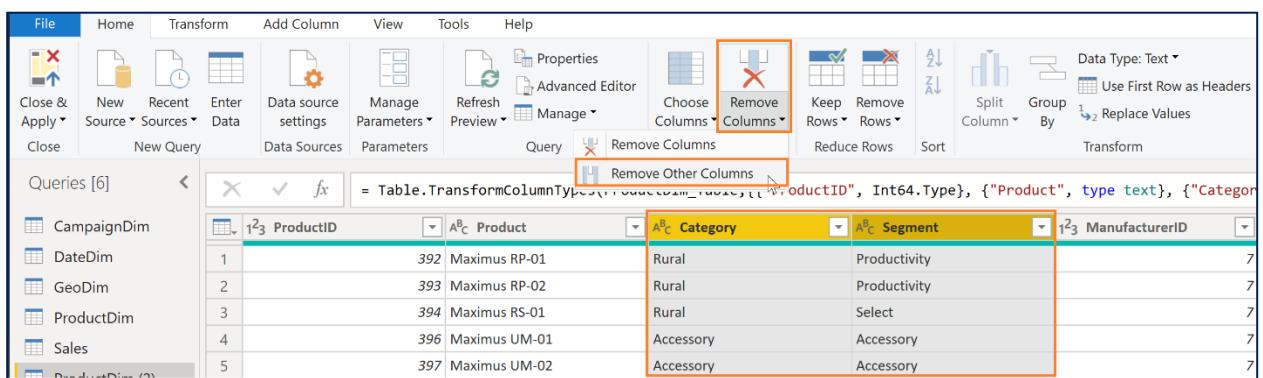
- From the **View** ribbon in Power Query check “**Always Allow**”



- Duplicate the **ProductDim** query



- Highlight **Category** and **Segment**, and Remove other columns



**

4. Highlight **Category** and **Segment**, right-click on the column header, and **Remove Duplicates**

A screenshot of a Power BI data view showing two columns: 'Category' and 'Segment'. The 'Category' column has values: 1 Rural, 2 Rural, 3 Rural, 4 Accessory, 5 Accessory, 6 Accessory, 7 Accessory, 8 Accessory, 9 Accessory, 10 Accessory, 11 Accessory, 12 Accessory, 13 Accessory. The 'Segment' column has values: Productivity, Productivity, Select, Accessory, Accessory, Accessory, Accessory, Accessory, Accessory, Accessory, Accessory, Accessory, Accessory. A context menu is open over the 'Category' column, with 'Remove Duplicates' highlighted.

**

5. **Add Column**> Add **Index** Column starting at 1, and change the **Index** column name to **CatSegID**

A screenshot of the Power BI ribbon showing the 'Add Column' tab selected. Below the ribbon, there is a 'General' section with a 'Conditional Column' button, an 'Index Column' button (which is highlighted), and a dropdown menu with 'From 0' and 'From 1' options, with 'From 1' selected. To the right, there are buttons for 'Merge Columns', 'Extract', 'Parse', and 'Statistics'. On the left, there is a 'Queries [6]' pane listing 'CampaignDim', 'DateDim', 'GeoDim', 'ProductDim', 'Sales', and 'ProductDim (2)'. The main area shows a table with columns 'Category' and 'Segment'. The data in the table is:

	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 02.B : Update Product dimension

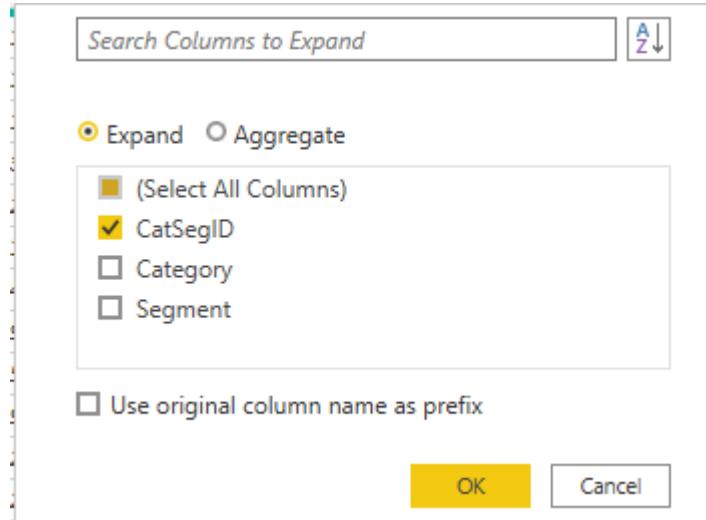
Task: Update the Product dimension

** **The estimated time to complete this lab is 10 minutes.**

- ** 1. In the Power Query Editor, 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**

The screenshot shows the 'Merge' dialog box in Power Query. It displays two tables: 'ProductDim' and 'CatSegDim'. The 'ProductDim' table has columns: ProductID, Product, Category 1, Segment 2, ManufacturerID, Manufacturer, Unit Cost, and Unit Price. The 'CatSegDim' table has columns: CatSegDim, Category 1, and Segment 2. Both the 'Category 1' and 'Segment 2' columns in both tables are highlighted with a red border. In the 'Join Kind' section, 'Left Outer (all from first, matching from second)' is selected. A note at the bottom states: 'The selection matches 212 of 212 rows from the first table.' There are 'OK' and 'Cancel' buttons at the bottom right.

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



3. Remove columns: **Category, Segment, 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, CatSegID, Product, Unit Price, Unit Cost**

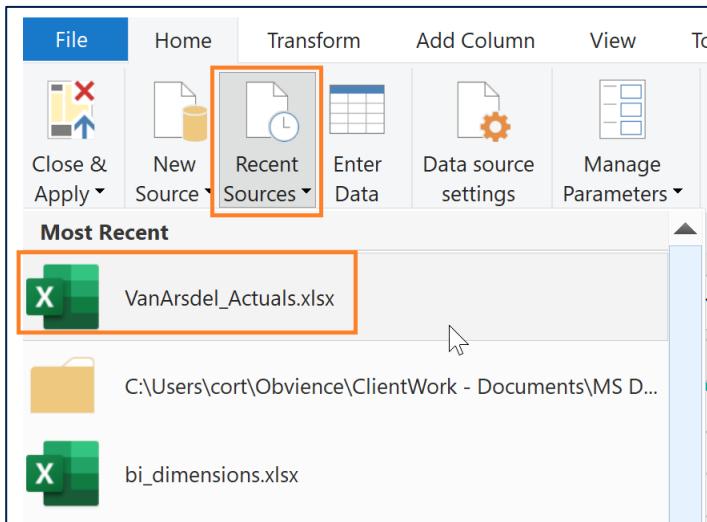
	1 ² 3 ProductID	1.2 CatSegID	A ^B C Product	1.2 Unit Price	1.2 Unit Cost
1	392	1	Maximus RP-01	51.05625	37.2710625
2	393	1	Maximus RP-02	51.05625	37.2710625
3	394	2	Maximus RS-01	164.05725	119.7617925
4	396	3	Maximus UM-01	90.79875	66.2830875
5	397	3	Maximus UM-02	149.61975	109.2224175

Lab 02.C : Create Customer dimension

Task: Create a Customer dimension

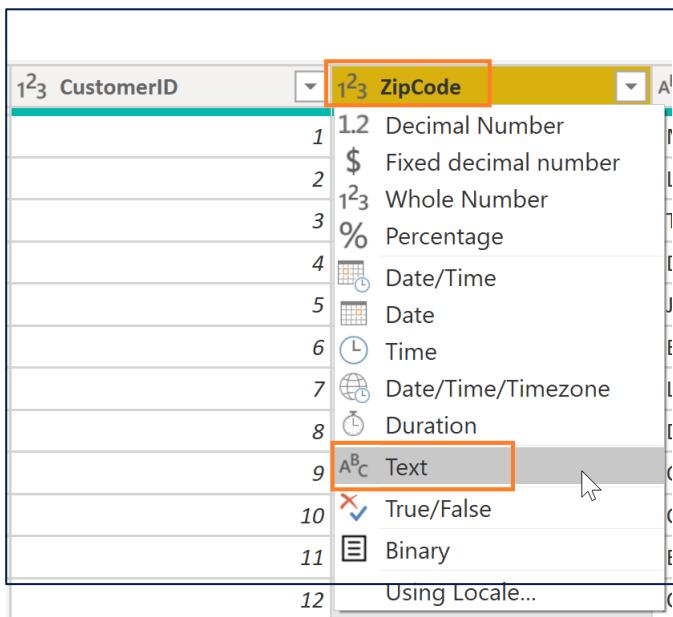
The estimated time to complete this lab is 20 minutes.

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

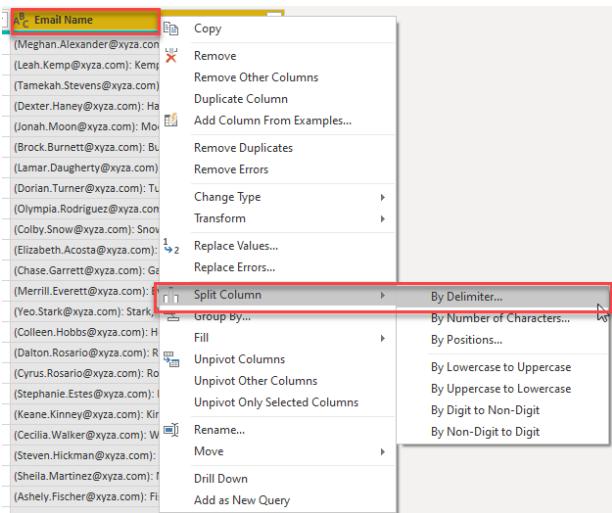


2. Change the **ZipCode** column data type to **Text**,

Note: Because leading zeros are part of a Zip code, using the *whole number* data type would cause data loss since leading zeros are trimmed when using a *whole number*.



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

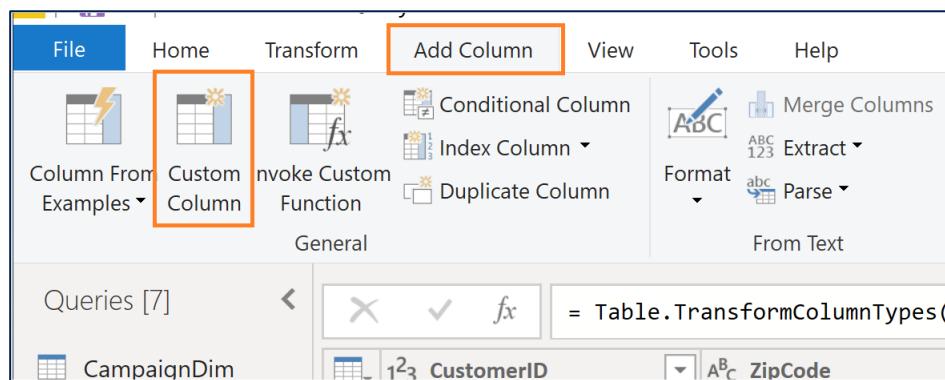
Left-most delimiter
 Right-most delimiter
 Each occurrence of the delimiter

> Advanced options

OK **Cancel**

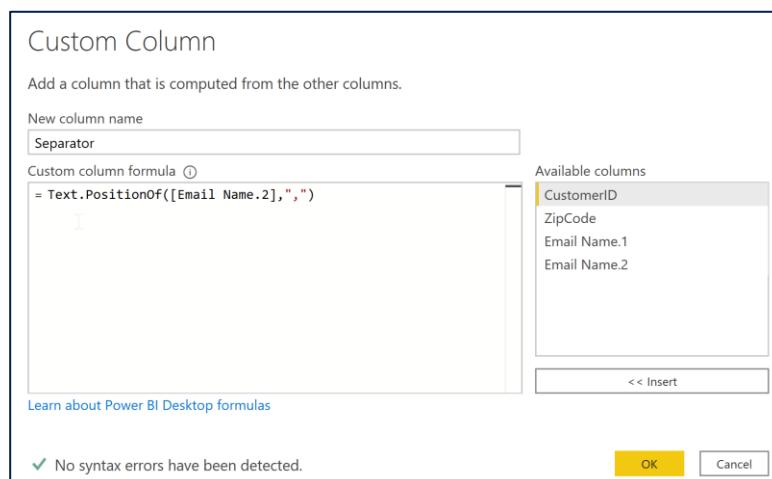
4. Add Column to find Text.PositionOf() the comma

a. Add Column > Custom Column



b. Name = "Separator"

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



***** Why is this step here!? 5. Use position of comma to split **Last Name** and **First Name**

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

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

7. Concatenate First Name and Last Name to get Full Name

Why is this here?

						= Table.AddColumn(#"Added Custom3", "Full Name", each [First Name] & " " & [Last Name])
	Name.1	Email Name.2	Separator	Last Name	First Name	Full Name
1	xander@xyz.com	Alexander, Meghan		Alexander	Meghan	Meghan Alexander
2	@xyz.com	Kemp, Leah		Kemp	Leah	Leah Kemp
3	evens@xyz.com	Stevens, Tamekah		Stevens	Tamekah	Tamekah Stevens
4	zy@xyz.com	Haney, Dexter		Haney	Dexter	Dexter Haney
5	o@xyz.com	Moon, Jonah		Moon	Jonah	Jonah Moon

8. Remove Separator column

9. Change the data type for these new columns to *Text* to help the compression of data when loaded into the dataset

The screenshot shows the Power BI Data Editor interface. A table is displayed with columns: Last Name, First Name, and Full Name. The 'Full Name' column is currently selected. A context menu is open over this column, with the 'Change Type' option highlighted. A sub-menu for 'Text' data type is also visible, with the 'Text' option selected. The 'Properties' pane on the right shows the current data type as 'Text'.

Last Name	First Name	Full Name
Alexander	Meghan	Meghan Alexander
Kemp	Leah	Leah Kemp
Stevens	Tamekah	Tamekah Stevens
Haney	Dexter	Dexter Haney
Moon	Jonah	Jonah Moon
Burnett	Brock	Brock Burnett
Daugherty	Lamar	Lamar Daugherty
Turner	Dorian	Dorian Turner
Rodriguez	Olympia	Olympia Rodriguez
Snow	Colby	Colby Snow
Acosta	Elizabeth	Elizabeth Acosta
Garrett	Chase	Chase Garrett
Everett	Merrill	Merrill Everett
Stark	Yeo	Yeo Stark
Hobbs	Colleen	Colleen Hobbs
Rosario	Dalton	Dalton Rosario
Rosario	Cyrus	Cyrus Rosario
Estes	Stephanie	Stephanie Estes
Kinney	Keane	Keane Kinney
Walker	Cecilia	Cecilia Walker
Hickman	Steven	Steven Hickman
Martinez	Sheila	Sheila Martinez
Fischer	Ashely	Ashely Fischer
Mejia	Fuller	Fuller Mejia
Railor	Iamai	Iamai Railor

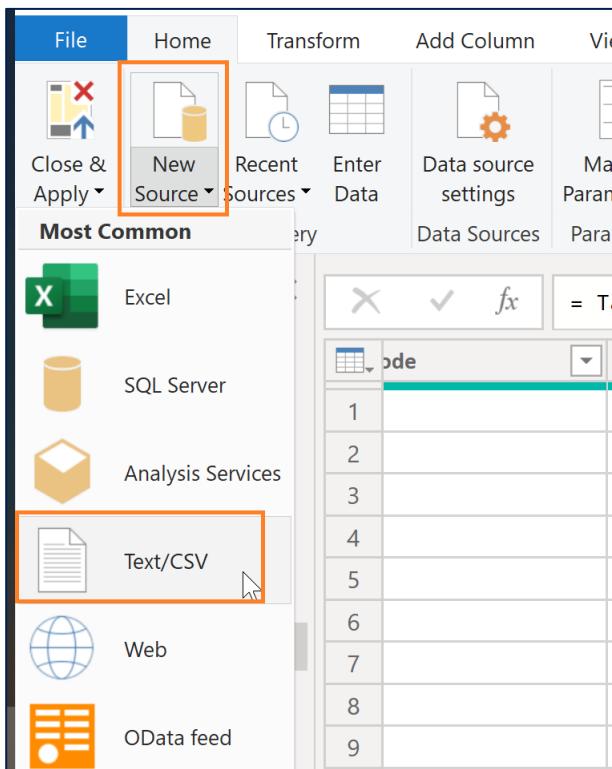
Lab 03: Create Budget Fact table

Task: Create Budget fact table

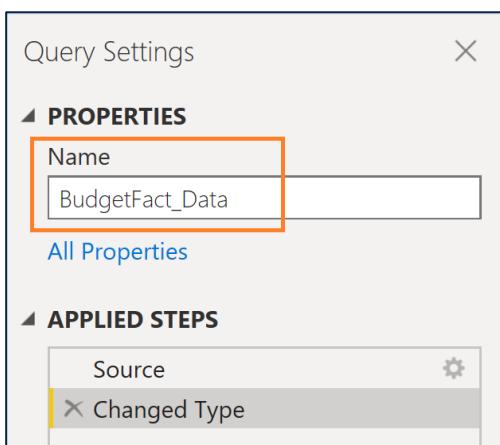
**

The estimated time to complete this lab is 20 minutes.

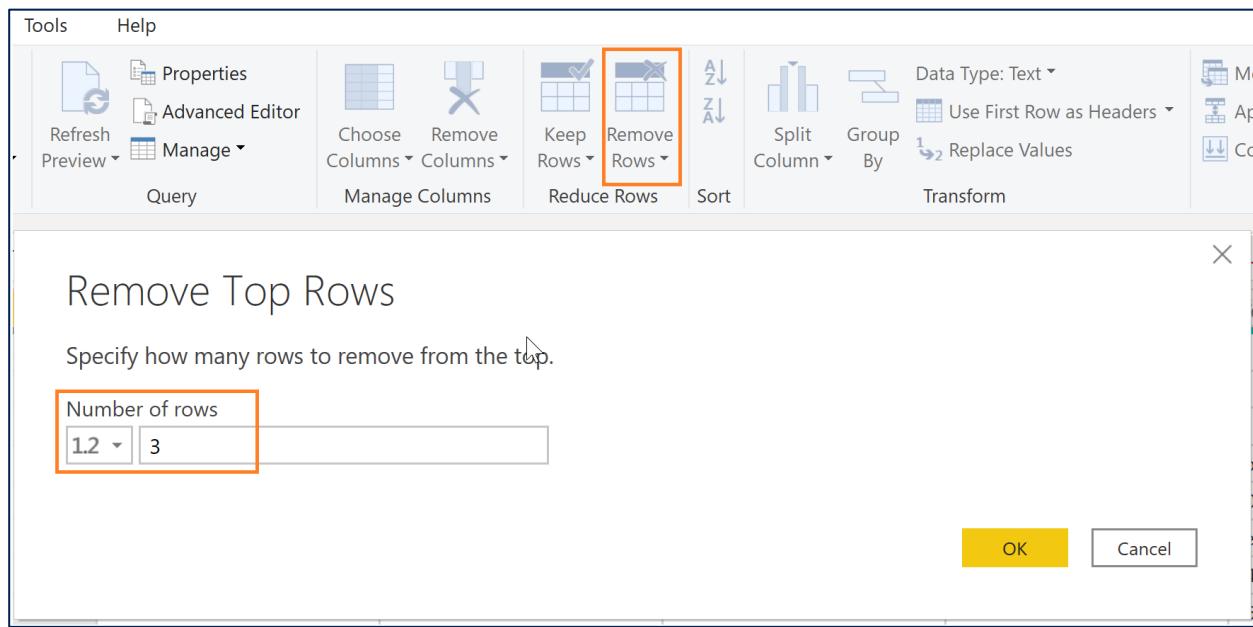
1. ImportCSV document "C:\Power BI_Adv_M\VanArsdel_Budget.csv"



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



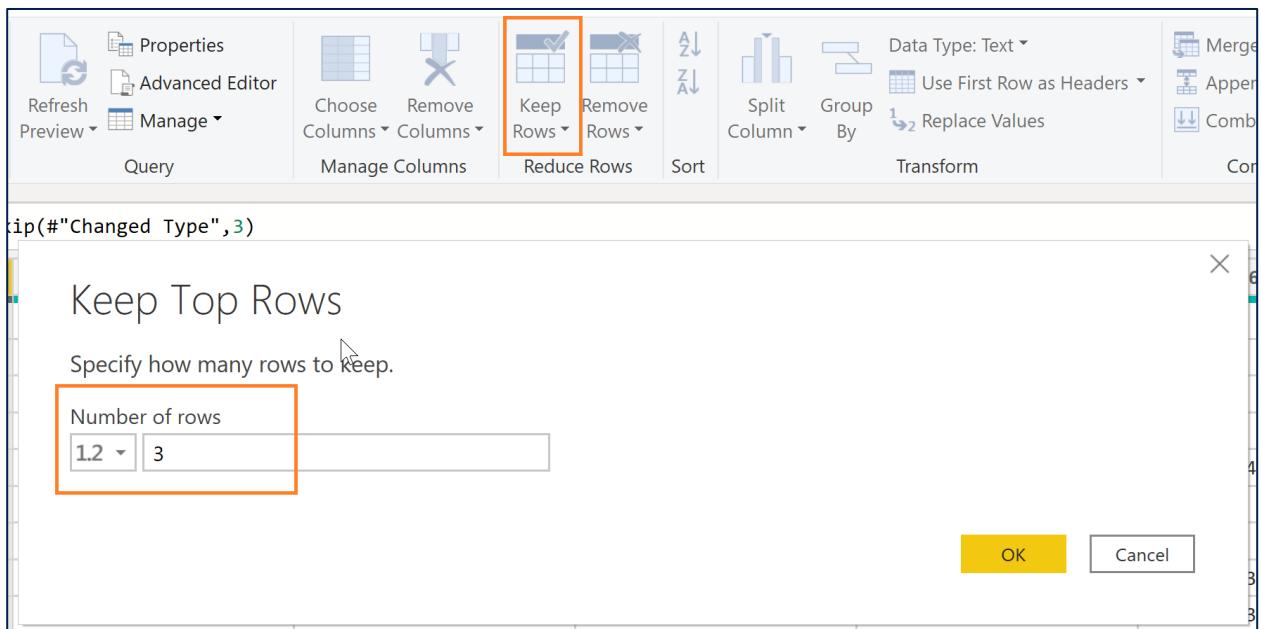
3. **Remove Rows** > Remove Top Rows, enter 3 (to remove the first 3 rows)



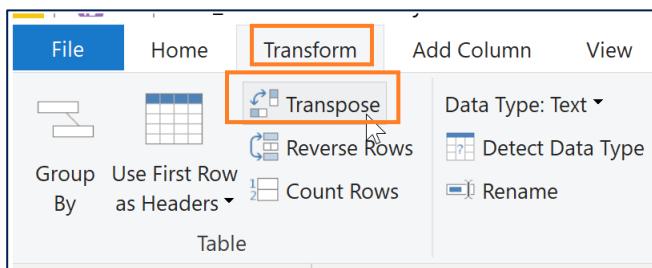
** 4. Duplicate query **BudgetFact_Data** and then rename it to "**BudgetFact**"

	Column1
1	
2	
3	Category
4	Accessory
5	Mix
6	Mix
7	Rural

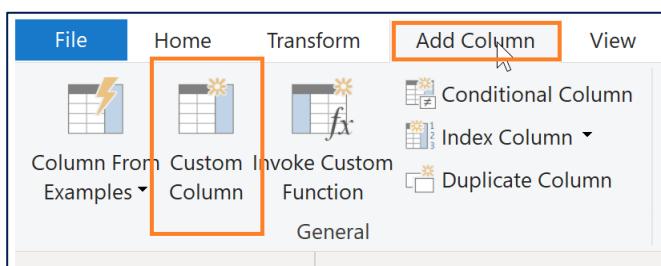
5. **Keep Rows** > Keep Top Rows, enter 3 (to keep the first three rows)



6. **Transform> Transpose**



7. **Add Column > Custom Column**

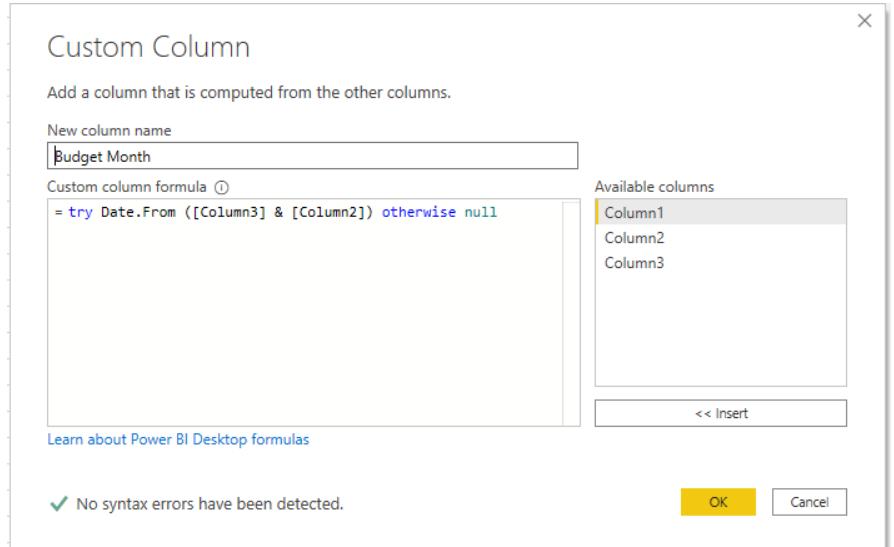


8. **Add Column** to combine month and year into a date

- Add Column > **Custom Column**
- Name = "**Budget Month**"
- Formula = **try Date.From ([Column3] & [Column2]) otherwise null**

**

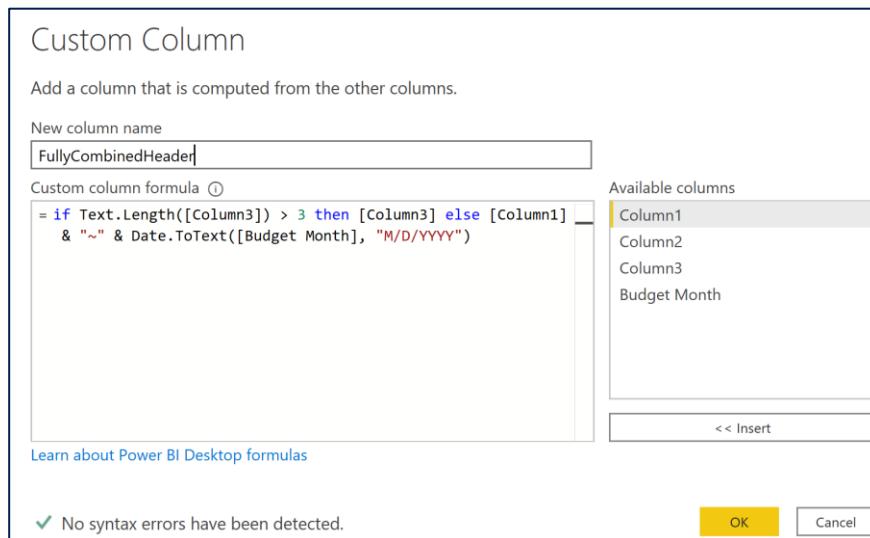
**



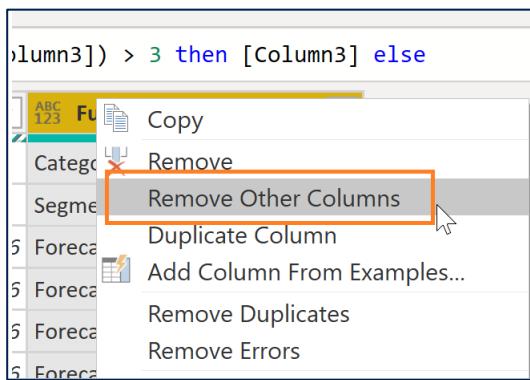
9. Add Column to combine Month and Scenario

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

**



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



11. Transform > Transpose to transpose back to wide

The Power BI ribbon is shown with the 'Transform' tab selected. The 'Transpose' button is highlighted with a red box. Below the ribbon, the formula bar shows '= Table.Transpose(#"Removed Other Columns")'. The data grid contains four columns: 'Column1' (Category), 'Column2' (Segment), 'Column3' (Forecast~12/01/2016), and 'Column4' (Forecast~11/01/2016).

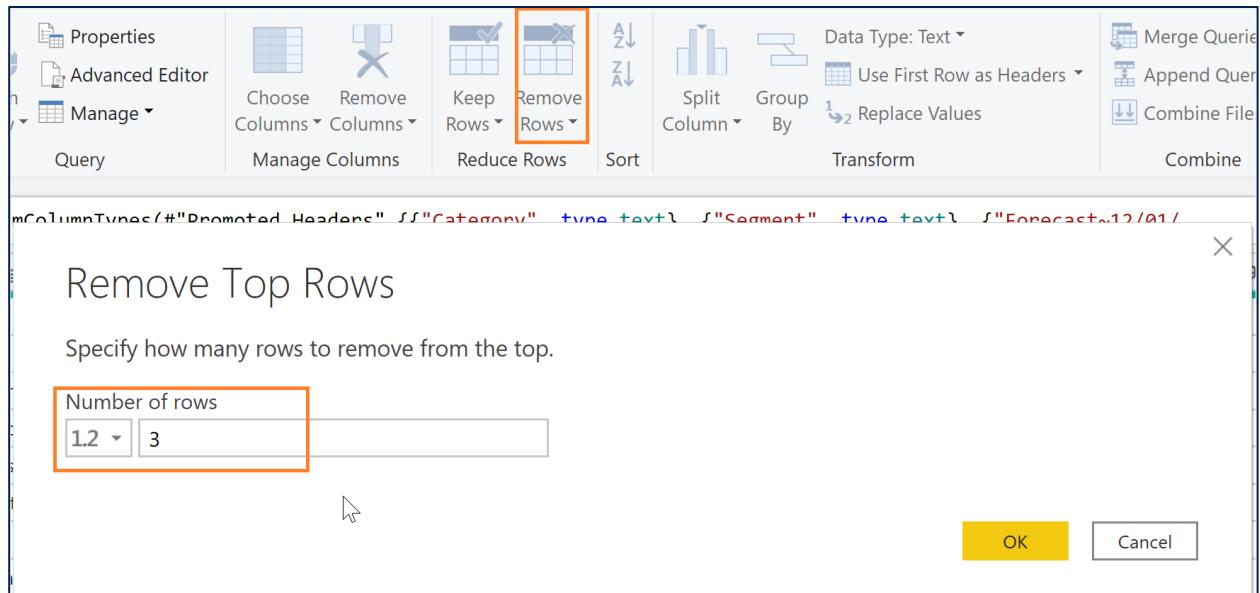
12. Home > Append Queries and choose BudgetFact_Data

The 'Append' dialog box is shown. It has two radio buttons: 'Two tables' (selected) and 'Three or more tables'. A dropdown menu labeled 'Table to append' contains 'BudgetFact_Data'. At the bottom are 'OK' and 'Cancel' buttons.

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

The Power BI ribbon is shown with the 'Transform' tab selected. The 'Use First Row as Headers' button is highlighted with a red box. Below the ribbon, the formula bar shows '= Table.TransformColumnTypes(#"Promoted Headers",{{"Category", type text}, {"Segment", type text}, {"Forecast~12/01/2016", type text}, {"Forecast~11/01/2016", type text}, {"Forecast~10/01/2016", type text}})'. The data grid shows five columns: 'Category', 'Segment', 'Forecast~12/01/2016', 'Forecast~11/01/2016', and 'Forecast~10/01/2016'. The first row is bolded and serves as the header.

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

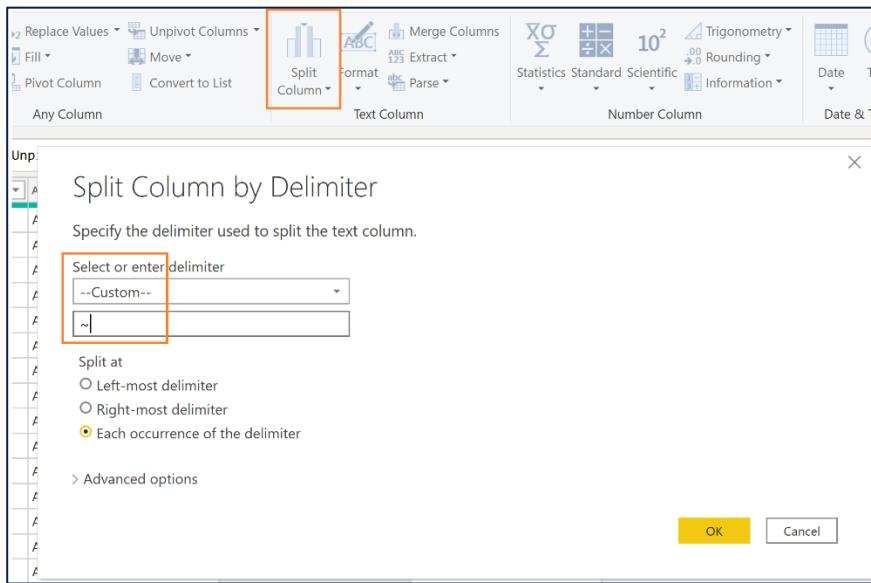


15. Highlight **Category** and **Segment** and **Transform> Unpivot Other Columns**

The screenshot shows the Power BI Query Editor interface. The 'Home' tab is selected in the ribbon. In the 'Transform' group, the 'Unpivot Columns' button is highlighted with an orange box. A dropdown menu is open, and the 'Unpivot Other Columns' option is highlighted with an orange box. To the right of the table, the formula bar shows the formula = Table.Skip(#"Changed Type1",3). The table itself has two columns: 'Category' and 'Segment', which are highlighted with an orange box. The data in the table is as follows:

	Category	Segment	Forecast^12/01/2016
1	Accessory	Accessory	44190.57888
2	Mix	All Season	11442.14474
3	Mix	Productivity	19538.89812
4	Rural	Select	311.708775
5	Urban	Convenience	120710.4406
6	Urban	Extreme	20868.84072
7	Urban	Moderation	251155.7122
8	Urban	Regular	689.7969225
9	Youth	Youth	3931.03074

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



17. 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

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

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

- From the CatSegDim highlight both **Category** and **Segment**
- Go back up to BudgetFact, highlight both **Category** and **Segment**
- Show the Join Kinds available, and leave "**Left Outer**"
- 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

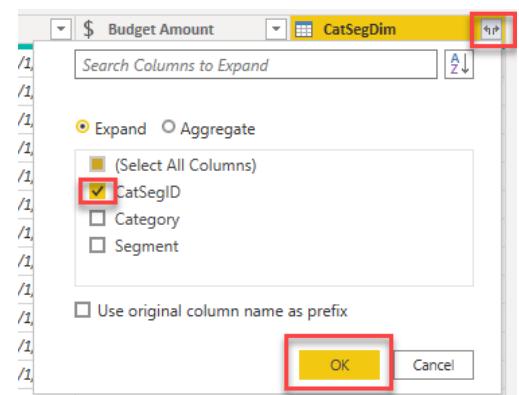
Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

> Fuzzy matching options

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

OK Cancel

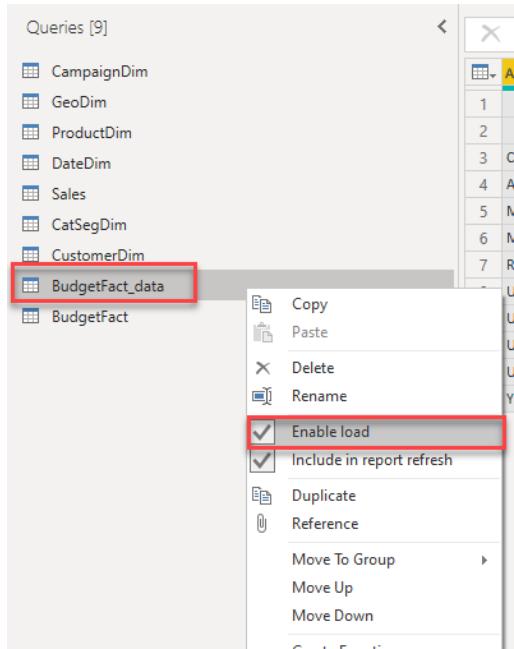


20. Remove: Category, Segment

21. Reorder: CatSegID, Scenario, Date, Budget Amount

	1.2 CatSegID	A ^B _C 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

22. Disable the load of BudgetFact_Data

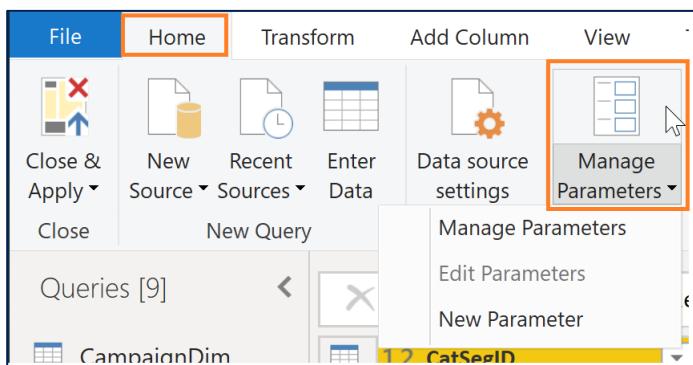


Lab 04.A : Create lab parameters

Task: Create new parameters

The estimated time to complete this lab is 15 minutes.

1. From the **Home** Ribbon > **Manage Parameters**

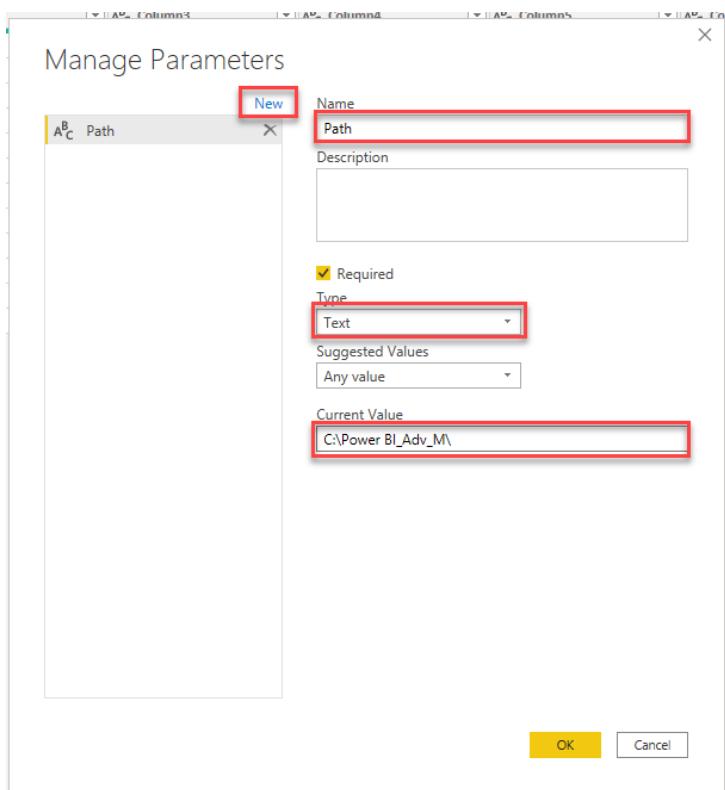


**

2. Click **New** to create a new parameter

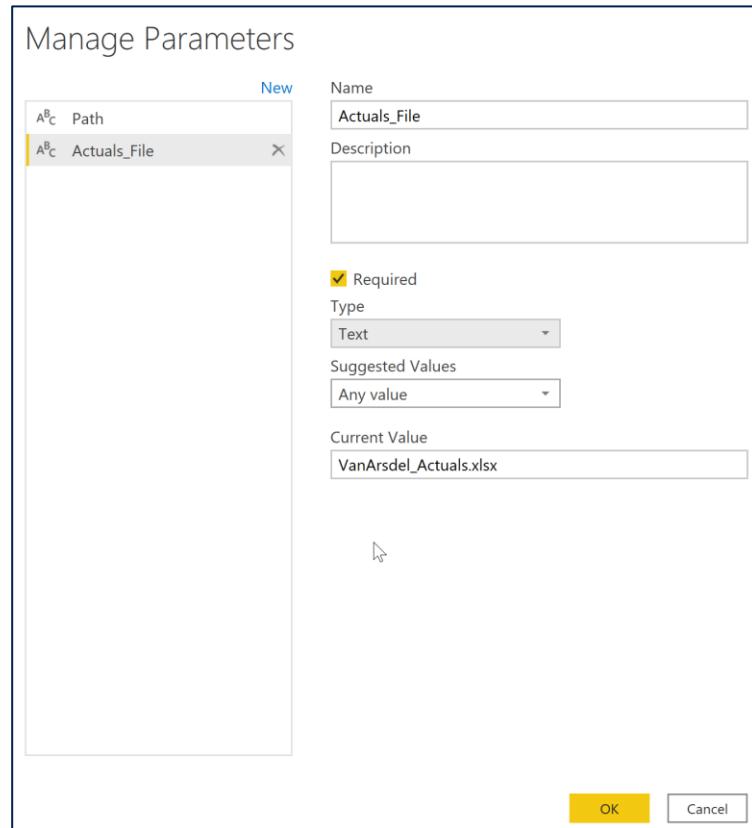
- a. Parameter Name: **Path**
- b. Type: **Text**
- c. Current Value = **C:\Power BI_Adv_M**

**



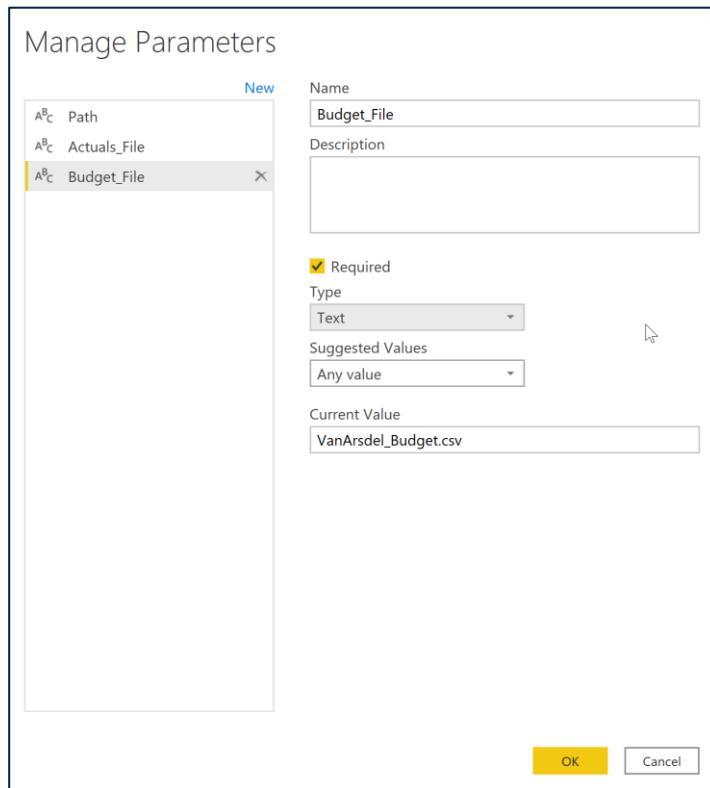
3. Create another new **Parameter**

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



4. Create another new **Parameter**

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



MAke this
more clear
so tha the
user
knows
what
they're
updating

5. Lastly, update the new *Path*, *Actuals_File*, and *Budget_File* parameters to match the location and names of your student course material files.

The screenshot shows the Power BI interface with the 'Queries [13]' pane on the left. A list of queries is shown, with 'Path (C:\Power BI_Adv_M\)' currently selected. To the right, there's a details panel:

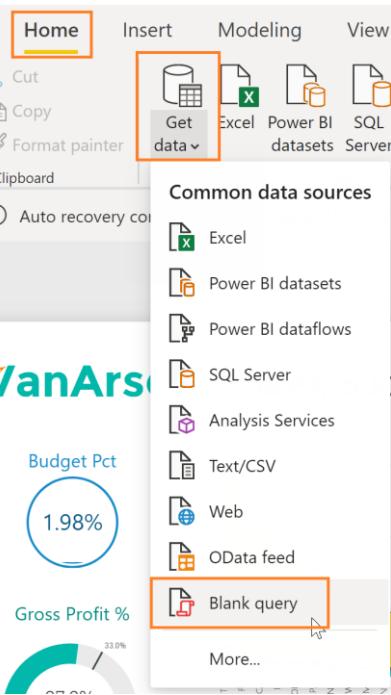
- Current Value:** C:\Power BI_Adv_M\
- Manage Parameter** button

Lab 04.B : Create dynamic path to excel source file

Task: Create a dynamic path to the excel source file

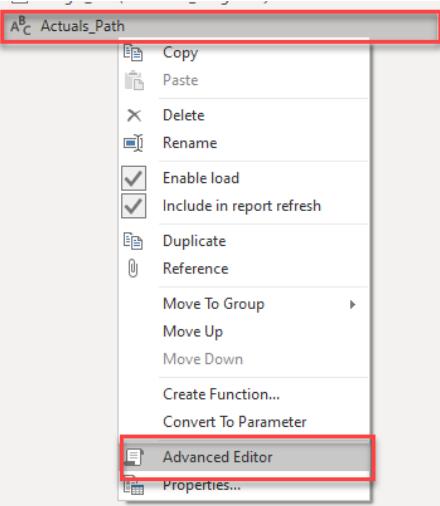
**** The estimated time to complete this lab is 10 minutes.**

- ****
1. From Power BI Desktop > Create a new **blank query**
 - a. Rename the Query to "Actuals_Path"



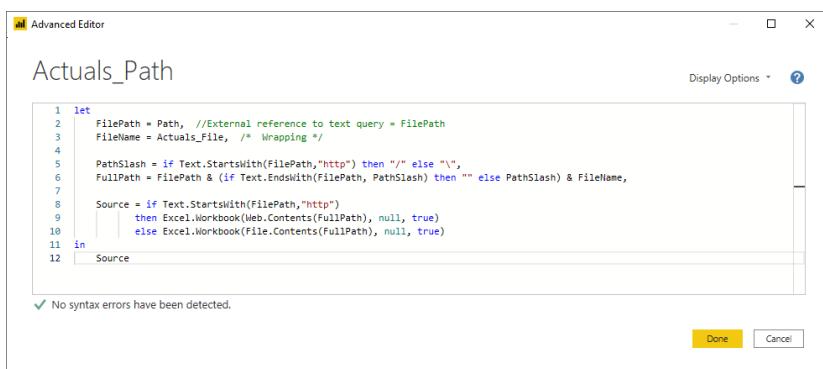
The screenshot shows the Power BI Desktop interface. The ribbon at the top has 'Home' selected. Below it, the 'Get data' button is highlighted with an orange box. A dropdown menu titled 'Common data sources' is open, listing various options like Excel, Power BI datasets, and OData feed. At the bottom of this list, 'Blank query' is also highlighted with an orange box. In the background, there are some dashboard cards and a watermark for 'JanArs'.

b. Right-click on the new query, and select "Advanced Editor"



A context menu is open over the 'Actuals_Path' query. The 'Advanced Editor' option is highlighted with a red box. Other options in the menu include Copy, Paste, Delete, Rename, Enable load, Include in report refresh, Duplicate, Reference, Move To Group, Move Up, Move Down, Create Function..., Convert To Parameter, and Properties... .

- ****
- c. From your student course materials, Copy in text from Actuals_Path.txt and paste it into this new query



The screenshot shows the 'Advanced Editor' window for the 'Actuals_Path' query. The code pane contains the following M code:

```
let
    FilePath = Path, //External reference to text query = FilePath
    FileName = Actuals_File, /* Wrapping */

    PathSlash = if Text.StartsWith(FilePath,"http") then "/" else "\",
    FullPath = FilePath & (if Text.EndsWith(FilePath, PathSlash) then "" else PathSlash) & FileName,
    Source = if Text.StartsWith(FilePath,"http")
        then Excel.Workbook(Web.ContentsFullPath, null, true)
        else Excel.Workbook(File.ContentsFullPath, null, true)
in
    Source
```

The status bar at the bottom left says 'No syntax errors have been detected.' There are 'Done' and 'Cancel' buttons at the bottom right.

** 2. For each of the following queries, Update the "Source" Applied Step to use the Resolved Path = "Actuals_Path"

- CampaignDim
- CustomerDim
- ProductDim
- CatSegDim
- DateDim
- GeoDim
- SalesFact

**

The screenshot shows the Power BI Advanced Data Shaping interface. A red box labeled 1 highlights the 'CampaignDim' query in the 'Queries [13]' list. A red box labeled 3 highlights the formula bar with the code: `= Excel.Workbook(File.Contents("C:\Users\July\Student Content\VanArsdel_Actuals.xlsx"), null, true)`. A red box labeled 2 highlights the 'APPLIED STEPS' section in the 'Query Settings' pane, which shows 'Source' selected. A red box labeled 4 highlights the 'Actuals_Path' query in the 'Queries [13]' list. A red arrow points from the 'Actuals_Path' query to the formula bar, with the text "... becomes..." placed between them.

3. Because this new query is only being used as a reference, right-click on the name of the query and select, "Enable Load"

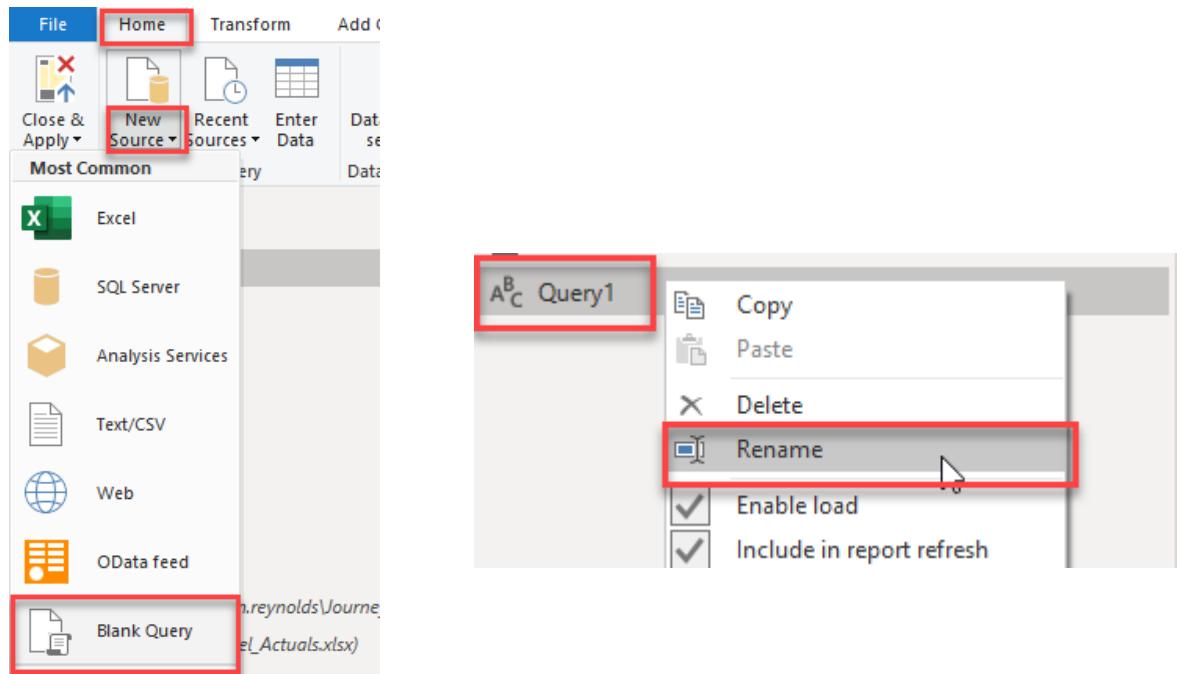
The screenshot shows the Power BI Query Editor. A red box highlights the 'Actuals_Path' query in the 'Queries' list. A context menu is open over the 'Actuals_Path' query, with a red box highlighting the 'Enable load' option under the 'More...' section. The 'Enable load' option has a checked checkbox next to it. Other options in the menu include 'Copy', 'Paste', 'Delete', 'Rename', 'Include in report refresh', and 'fn_DaySinceYearStart'.

Lab 04.C : Create dynamic path to csv source file

Task: Create a dynamic path to csv source file

** **The estimated time to complete this lab is 5 minutes.**

- ** 1. From the Power Query Editor window, Create a new blank query. Rename the Query to "Budget_Path"



- ** 2. From your student course materials, Copy in text from Budget_Path.txt and paste it into this new query

The screenshot shows the Power Query Editor with a table loaded. A formula bar at the top contains the expression: = if Text.StartsWith(FilePath, "http") . The table has five columns: Column1, Column2, Column3, Column4, and Column5. The first column contains category names like 'Category', 'Accessory', 'Mix', etc. The second column contains segment names like 'Segment', 'Accessory', 'All Season', etc. The third column contains various numerical values. The fourth and fifth columns also contain numerical values. To the right of the table, the 'Query Settings' pane is visible, showing the 'Properties' section with 'Name' set to 'Budget_Path'. Below that, the 'Applied Steps' section lists 'FilePath', 'BudgetFilename', 'PathSlash', and 'FullPath', with 'Source' being the currently selected step.

**

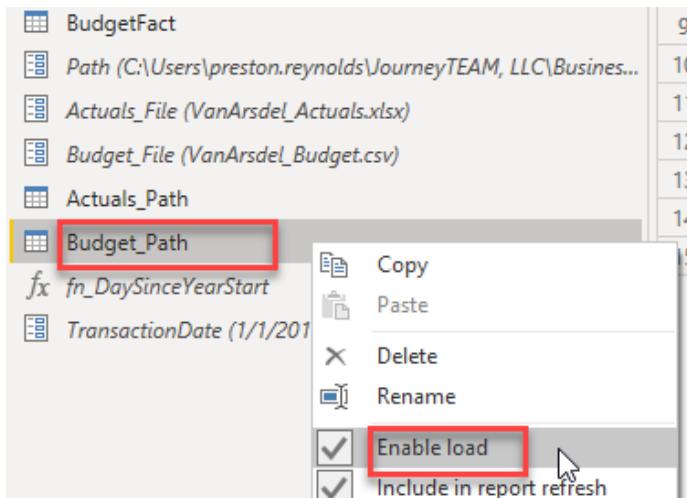
3. UpdateSource Applied Step to use ResolvedBudgetPath = "Budget_Path" to the following

Queries:

- BudgetFact
- BudgetFact_Data

The screenshot shows the Power BI Query Editor interface. On the left, the 'Queries [19]' pane lists various inputs, helper queries, and data models. The 'BudgetFact' query is selected and highlighted with a red box. In the main area, the 'Budget_Path' query is displayed as a table with columns: Column1, Column2, Column3, Column4, and Column5. The table contains 15 rows of data. On the right, the 'Query Settings' pane is open, showing the 'APPLIED STEPS' section. The 'Source' step is highlighted with a red box. Other steps listed include 'Changed Type', 'Removed Top Rows', 'Added Index', 'Filtered Rows', 'Removed Columns', 'Transposed Table', 'Added Custom', 'Added Custom1', 'Removed Other Columns', and 'Transposed Table1'. The 'Name' field in the settings is set to 'BudgetFact'.

4. Because this new query is only being used as a reference, right-click on the name of the query and select, "Enable Load"

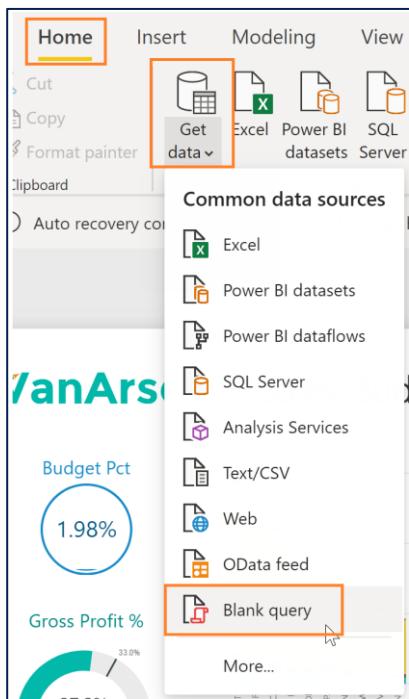


Lab 04.D : Create a custom function

Task: Create a custom function

The estimated time to complete this lab is 15 minutes.

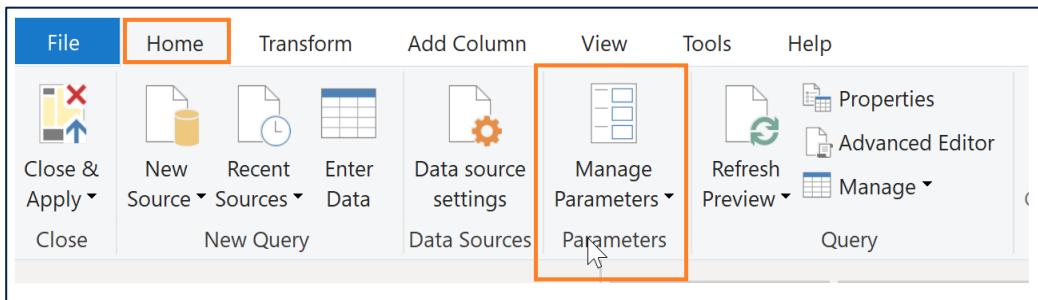
- ** 1. From Power BI Desktop, Create a new blank query



- Query Name: "**fn_DaySinceYearStart**"
- In Advanced Editor paste in the 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
```

- Create a new Parameter



We should actually use this if we're going to make it

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

The screenshot shows the 'Manage Parameters' dialog box. A new parameter named 'TransactionDate' is being created. The 'Name' field contains 'TransactionDate'. The 'Type' dropdown is set to 'Date'. The 'Required' checkbox is checked. The 'Current Value' field contains '1/1/2011'. The 'Suggested Values' dropdown is set to 'Any value'.

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

- b. New Column Name: **DaysFromYearStart**
- c. Function query: **fn_DaySinceYearStart**
- d. Transaction Date = **Date**

Note that you need to use the dropdown

4. Change the date type of the new DaysFromYearStart column to Whole Number

4. Save your file, and hit close and apply

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.