



Power BI Advanced

# Shape Your Data into a Data Model with M

Version: August 2019



## Module 2 Lab: Import Multiple Tables from a Single Source

If the Source mirrors the Table required by the Model, import multiple tables at once

**CampaignDim**

CampaignID	TrafficChannel	Device
1	Organic Search	Desktop
2	Organic Search	Mobile
3	Organic Search	Tablet
4	SEO	Desktop

**GeoDim**

Zip	City	State	Region	District	Country
22654	Star Tannery, VA, USA	VA	East	District 807	USA
22655	Stephens City, VA, USA	VA	East	District 807	USA
22656	Stephensan, VA, USA	VA	East	District 807	USA
22657	Stresburg, VA, USA	VA	East	District 807	USA

**ProductDim**

ProductID	Product	Category	Segment	ManufacturerID	Manufacturer
382	Maximus RP-03	Rural	Productivity	7	VanArsdel
393	Maximus RP-02	Rural	Productivity	7	VanArsdel
394	Maximus RS-01	Rural	Select	7	VanArsdel
395	Maximus UM-01	Accessory	Accessory	7	VanArsdel

**DateDim**

Date	MonthNo	MonthName	MonthID	Month	Quarter	Year
1/1/2011	1	Jan	201101	1/1/2016 Q3		2011
1/2/2011	1	Jan	201101	1/1/2016 Q3		2011
1/3/2011	1	Jan	201101	1/1/2016 Q3		2011
1/4/2011	1	Jan	201101	1/1/2016 Q3		2011

**Sales**

ProductID	Date	CustomerID	CampaignID	Units
676	9/25/2012	70069	22	1
649	9/25/2012	199389	22	1
613	5/16/2012	212645	22	1
613	5/14/2012	70646	22	1
613	5/14/2012	114439	22	1

## Step by Step

Objective: Import multiple worksheets from an Excel file in one step

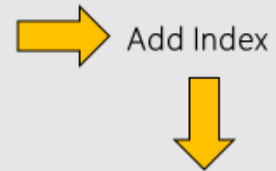
1. Get Data > from Excel  
C:\Power BI\_Adv\_M\VansArsdel\_Actuals.xlsx
2. Select the following Tables (All but CustomerDim):
  - a) CampaignDim
  - b) GeoDim
  - c) ProductDim
  - d) DateDim
  - e) Sales
3. In GeoDim, **change** the [ZIP] Data Type to Text

## Student Notes

## Module 2 Lab: Create CatSegDim (Category Segment)

Extract unique combinations of Categories & Segments from the Product Dimension

	ProductID	Product	Category	Segment	ManufacturerID	Manufacturer
1	352	Maximus RP-01	Rural	Productivity	7	VanArndel
2	353	Maximus RP-02	Rural	Productivity	7	VanArndel
3	354	Maximus RS-01	Rural	Select	7	VanArndel
4	356	Maximus UM-01	Accessory	Accessory	7	VanArndel
5	357	Maximus UM-02	Accessory	Accessory	7	VanArndel



*CatSegDim* is a **Degenerate Dimension**  
It is extracted from the data in the fact table.

	Category	Segment
1	Rural	Productivity
2	Rural	Select
3	Accessory	Accessory
4	Urban	Regular
5	Urban	Education
6	Urban	all Season
7	Urban	Productivity
8	Urban	Work
9	Urban	Commuter

## Step by Step

Objective: Create the Product Category Dimension by extracting Categories from the ProductDim

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
5. Add Column > Add Index Column starting at 1, with column name [CatSegID]
6. Reorder Columns: [CatSegID], [Category], [Segment]
7. Rename the query “CatSegDim”

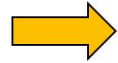
## Student Notes

## Module 2 Lab: Update Product Dim



### Apply new CatSegID field to the Product Dimension

ProductID	Product	Category	Segment	ManufacturerID	Manufacturer
392	Maximus RP-01	Rural	Productivity	7	VanKandel
393	Maximus RP-02	Rural	Productivity	7	VanKandel
394	Maximus RS-01	Rural	Select	7	VanKandel
395	Maximus UM-01	Accessory	Accessory	7	VanKandel
397	Maximus UM-02	Accessory	Accessory	7	VanKandel



Replace with Index



ProductID	CatSegID	Product	Unit Price	Unit Cost
392	1	Maximus RP-01	81.05625	87.2710625
393	1	Maximus RP-02	51.05625	57.2710625
394	2	Maximus RS-01	144.05725	119.7417925
395	3	Maximus UM-01	90.74675	68.2630875
397	3	Maximus UM-02	149.41975	109.2224175
398	3	Maximus UM-03	157.48475	114.9711675
400	3	Maximus UM-05	84.48975	61.4840975
402	3	Maximus UM-07	122.54975	74.7239175
403	3	Maximus UM-08	113.85675	83.1154275
404	3	Maximus UM-09	204.26975	136.0499275
405	3	Maximus UM-10	191.41975	126.0624275
406	3	Maximus UM-11	181.41975	129.8524275

## Student Notes

## Step by Step

Objective: Update the Product Category ID in the ProductDim

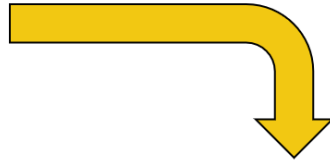
1. Select the *ProductDim* query
2. From Home Ribbon > Merge Queries > **Select** CatSegDim
  - a) From *ProductDim*, **highlight** [Category] and [Segment]
  - b) From *CatSegDim*, **highlight** [Category] and [Segment]
  - c) Note the Join Kinds available, and leave **Left Outer**
  - d) **Expand**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

## Module 2 Lab: Create Customer Dimension



### Using a *garbage* column to fill out CustomerDim

CustomerID	ZipCode	Email Name
1	90250	(Meghan.Alexander@xyz.com): Alexander, Meghan
2	90250	(Leah.Kemp@xyz.com): Kemp, Leah
3	90250	(Tamekah.Stevens@xyz.com): Stevens, Tamekah
4	90250	(Dexter.Haney@xyz.com): Haney, Dexter
5	90250	(Jonah.Moon@xyz.com): Moon, Jonah
6	90250	(Brock.Burnett@xyz.com): Burnett, Brock
7	90250	(Lamar.Daugherty@xyz.com): Daugherty, Lamar
8	90250	(Dorian.Turner@xyz.com): Turner, Dorian
9	90250	(Olympia.Rodriguez@xyz.com): Rodriguez, Olympia
10	90250	(Colby.Snow@xyz.com): Snow, Colby
11	90250	(Elizabeth.Acosta@xyz.com): Acosta, Elizabeth
12	90250	(Chase.Garrett@xyz.com): Garrett, Chase



Key Formulas:

Text.PositionOf()

Text.Start()

Text.Range()

CustomerID	Email	ZipCode	First Name	Last Name	Full Name
1	Meghan.Alexander@xyz...	90250	Alexander	Meghan	Alexander Meghan
2	Leah.Kemp@xyz.com	90250	Kemp	Leah	Kemp Leah
3	Tamekah.Stevens@xyz...	90250	Stevens	Tamekah	Stevens Tamekah
4	Dexter.Haney@xyz.com	90250	Haney	Dexter	Haney Dexter
5	Jonah.Moon@xyz.com	90250	Moon	Jonah	Moon Jonah
6	Brock.Burnett@xyz.c...	90250	Burnett	Brock	Burnett Brock
7	Lamar.Daugherty@xyz...	90250	Daugherty	Lamar	Daugherty Lamar
8	Dorian.Turner@xyz.c...	90250	Turner	Dorian	Turner Dorian
9	Olympia.Rodriguez@xy...	90250	Rodriguez	Olympia	Rodriguez Olympia
10	Colby.Snow@xyz.com	90250	Snow	Colby	Snow Colby
11	Elizabeth.Acosta@xyz...	90250	Acosta	Elizabeth	Acosta Elizabeth
12	Chase.Garrett@xyz.c...	90250	Garrett	Chase	Garrett Chase

## Student Notes

## Step by Step

Objective: Create a First, Last, and Full Name columns based on the Email Name column.

1. Use **Recent Sources** to get *CustomerDim* from Excel
2. Change the ZipCode column data type to TEXT
3. Split by Delimiter Custom ": " colon space
4. Replace to Remove "(" and ")"
5. Add Column to find Text.PositionOf() the comma
  - a) Add Column > Custom Column
  - b) Name = "Separator"
  - c) Formula = Text.PositionOf([Email Name.2],",")
6. Use position of comma to split Last Name and First Name  
Add 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
8. Remove Separator column

## Module 3 Lab: Create Budget Fact



Import CSV, remove blank rows, and rename query to BudgetFact\_Data

	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
1			Forecast	Forecast	Forecast	Forecast	Forecast	Forecast	Forecast
2			2016	2016	2016	2016	2016	2016	2016
3	Category	Segment	Dec	Nov	Oct	Sep	Aug	Jul	Jun
4	Accessory	Accessory	44190.57888	50598.81566	54740.5709	64442.9079	98285.91328	61545.41064	91299.65402
5	Mix	All Season	11442.14474	14120.78693	18109.64804	26737.36704	37439.93736	45563.53694	29301.60844
6	Mix	Productivity	19538.89812	17597.55926	22835.18396	25557.336	34963.40484	17727.72593	73468.33494
7	Rural	Select	311.708775	172.2601125	662.79129	314.98992	311.708775	334.67679	167.338395
8	Urban	Convenience	120710.4406	129923.2814	169468.7696	219703.7363	333460.8569	345230.9632	262211.9656
9	Urban	Extreme	20868.84072	46971.33037	70793.02886	11957.87523	10668.924	14286.02201	5340.80274
10	Urban	Moderation	251155.7122	322984.2215	362385.6466	529188.1249	675144.698	642300.7198	651766.3129
11	Urban	Regular	689.7969225	427.4372025	2989.28376	701.48442	731.3103	1381.096238	151.17039
12	Youth	Youth	3931.03074	2891.005425	5397.748965	6437.887575	10355.73462	11738.08624	12144.36878

## Step by Step

Objective: Create the Budget Fact

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)

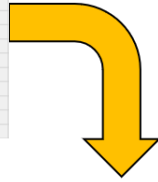
## Student Notes

## Module 3 Lab: Create Budget Fact



Create a Duplicate, and Extract just the Header Rows into a separate query

	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
1			Forecast	Forecast	Forecast	Forecast	Forecast	Forecast	Forecast
2			2016	2016	2016	2016	2016	2016	2016
3	Category	Segment	Dec	Nov	Oct	Sep	Aug	Jul	Jun
4	Accessory	Accessory	84190.57888	50598.81566	54740.5709	64442.9079	98285.91328	61545.41064	91299.65402
5	Mix	All Season	11442.14474	14120.78693	18109.64804	26737.36704	37439.93736	45563.53694	29301.60944
6	Mix	Productivity	19538.89812	17597.55926	22835.18396	25557.336	34963.40484	17727.72593	73468.33494
7	Rural	Select	311.708775	172.2401125	662.79129	314.98992	311.708775	334.67679	167.338395
8	Urban	Convenience	120710.4406	129923.2814	169468.7696	219703.7363	333460.8569	345230.9632	262211.9656
9	Urban	Extreme	20868.84072	46971.33037	70793.02886	11957.87523	10648.924	14286.02201	5340.80274
10	Urban	Moderation	251155.7122	322984.2215	362385.6466	529188.1249	675144.698	642300.7198	651766.3129
11	Urban	Regular	889.7949225	427.4772025	2989.28376	701.48442	731.3103	1381.094238	151.17039
12	Youth	Youth	3931.03074	2891.005425	5397.748945	6437.887575	10355.73462	11738.08424	12144.34878



	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8
1			Forecast	Forecast	Forecast	Forecast	Forecast	Forecast
2			2016	2016	2016	2016	2016	2016
3	Category	Segment	Dec	Nov	Oct	Sep	Aug	Jul

## Step by Step

Objective: To Create a Header which combines the first three rows

1. Duplicate query BudgetFact\_Data rename to "BudgetFact"
2. Keep Rows > Keep Top Rows, enter 3 (to keep the first three rows)

## Student Notes



## Module 3 Lab: Create Budget Fact



Transpose headers, combine into a single column, and transpose back

Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8
Forecast	Forecast	Forecast	Forecast	Forecast	Forecast	Forecast	Forecast
2016	2016	2016	2016	2016	2016	2016	2016
Category	Segment	Dec	Nov	Oct	Sep	Aug	Jul

Column1	Column2	Column3
Forecast	2016	Dec
Forecast	2016	Nov
Forecast	2016	Oct
Forecast	2016	Sep
Forecast	2016	Aug
Forecast	2016	Jul
Forecast	2016	Jun
Forecast	2016	May
Forecast	2016	Apr
Forecast	2016	Mar
Forecast	2016	Feb
Forecast	2016	Jan
Budget	2016	Dec
Budget	2016	Nov

FullyCombinedHeader
Category
Segment
Forecast-12/01/2016
Forecast-11/01/2016
Forecast-10/01/2016
Forecast-09/01/2016
Forecast-08/01/2016
Forecast-07/01/2016
Forecast-06/01/2016
Forecast-05/01/2016
Forecast-04/01/2016
Forecast-03/01/2016
Forecast-02/01/2016
Forecast-01/01/2016
Budget-12/01/2016
Budget-11/01/2016

Column1	Column2	Column3	Column4	Column5
Category	Segment	Forecast-12/01/2016	Forecast-11/01/2016	Forecast-10/01/2016

### Student Notes

### Step by Step

Objective: To Create a Header which combines the first three rows

1. Transform > Transpose
2. Add Column > Custom Column
3. Add Column to combine month and year into a date
  - a) Add Column > Custom Column
  - b) Name = "Budget Month"
  - c) Formula = *try Date.From ([Column3] & [Column2]) otherwise null*
4. 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/yy"
5. Remove all columns except for [FullyCombinedHeader]
6. Transform > Transpose to transpose back to wide



## Module 3 Lab: Create Budget Fact



Append BudgetFact\_Data to the bottom of the headers, then remove the extraneous header rows, finally set first row as header

Column1	Column2	Column3	Column4	Column5	Column6	Column7
Category	Segment	Forecast-12/01/2016	Forecast-11/01/2016	Forecast-10/01/2016	Forecast-09/01/2016	Forecast-08/01/2016

+

Column1	Column2	Column3	Column4	Column5	Column6	Column7
		Forecast	Forecast	Forecast	Forecast	Forecast
		2016	2016	2016	2016	2016
Category	Segment	Dec	Nov	Oct	Sep	Aug
Accessory	Accessory	44190.57888	50598.81566	54740.5709	64442.9079	98285.91328
Mix	All Season	11442.14474	14120.78693	18109.64804	26737.36704	37439.93736
Mix	Productivity	19538.89812	17597.55926	22835.18396	25557.336	34963.40484

=

Category	Segment	Forecast-12/01/2016	Forecast-11/01/2016	Forecast-10/01/2016	Forecast-09/01/2016	Forecast-08/01/2016
Accessory	Accessory	44190.57888	50598.81566	54740.5709	64442.9079	98285.91328
Mix	All Season	11442.14474	14120.78693	18109.64804	26737.36704	37439.93736
Mix	Productivity	19538.89812	17597.55926	22835.18396	25557.336	34963.40484
Rural	Select	311.708775	172.2601125	662.79129	314.98992	311.708775
Urban	Convenience	120710.4406	129923.2814	169468.7696	219703.7363	333460.8569

## Step by Step

Objective: To Append the new header row to the data to create the new wide data table with a single header row

1. **Append** query *BudgetFact\_Data*
2. **Use First Row as Header** to promote the newly fixed header row
3. **Remove Rows > Remove Top Rows**, enter **3** (to remove the first 3 rows – the old header rows)

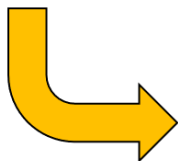
## Student Notes

## Module 3 Lab: Create Budget Fact



Unpivot Forecast Columns, Split and Rename them, then set data types

	Category	Segment	Forecast*12/01/2016	Forecast*11/01/2016	Forecast*10/01/2016	Forecast*09/01/2016	Forecast*08/01/2016
1	Accessory	Accessory	44190.57888	50598.81566	54740.5709	64442.9079	98285.91328
2	MLX	All Season	11442.14474	14120.78693	18109.64804	26737.36704	37439.93736
3	MLX	Productivity	19538.89812	17597.55926	22835.18396	25557.336	34963.40484
4	Rural	Select	311.708775	172.2601125	662.79129	314.98992	311.708775
5	Urban	Convenience	120710.4406	129923.2814	169468.7696	219703.7363	333460.8569



	Category	Segment	Scenario	Date	Value
1	Accessory	Accessory	Forecast	12/1/2016	44190.5789
2	Accessory	Accessory	Forecast	11/1/2016	50598.8157
3	Accessory	Accessory	Forecast	10/1/2016	54740.5709
4	Accessory	Accessory	Forecast	9/1/2016	64442.9079
5	Accessory	Accessory	Forecast	8/1/2016	98285.9133
6	Accessory	Accessory	Forecast	7/1/2016	61545.4106
7	Accessory	Accessory	Forecast	6/1/2016	91299.654
8	Accessory	Accessory	Forecast	5/1/2016	120579.7261
9	Accessory	Accessory	Forecast	4/1/2016	103436.9095

## Step by Step

Objective: Create the Budget Fact

1. Highlight [Category] and [Segment] and **Transform > Unpivot Other Columns**
2. Highlight [Attribute] and navigate to **Home > Split Column > By Delimiter > "~"**
3. **Rename:** [Attribute.1] = "Scenario", [Attribute.2] = "Date", [Value] = "Budget Amount"
4. **Change the Data Types:** [Budget Amount] = Fixed Decimal, [Date] = Date

## Student Notes

## Module 3 Lab: Create Budget Fact



Merge in CatSegID and remove extraneous columns

	Category	Segment	Scenario	Date	Value
1	Accessory	Accessory	Forecast	12/1/2016	44190.5789
2	Accessory	Accessory	Forecast	11/1/2016	50598.8157
3	Accessory	Accessory	Forecast	10/1/2016	54740.5709
4	Accessory	Accessory	Forecast	9/1/2016	64442.9079
5	Accessory	Accessory	Forecast	8/1/2016	98285.9133
6	Accessory	Accessory	Forecast	7/1/2016	61545.4106
7	Accessory	Accessory	Forecast	6/1/2016	91299.654
8	Accessory	Accessory	Forecast	5/1/2016	120579.7261
9	Accessory	Accessory	Forecast	4/1/2016	103436.9095



	CatSegID	Scenario	Date	Value
1	3	Forecast	12/1/2016	44190.5789
2	3	Forecast	11/1/2016	50598.8157
3	3	Forecast	10/1/2016	54740.5709
4	3	Forecast	9/1/2016	64442.9079
5	3	Forecast	8/1/2016	98285.9133
6	3	Forecast	7/1/2016	61545.4106
7	3	Forecast	6/1/2016	91299.654
8	3	Forecast	5/1/2016	120579.7261
9	3	Forecast	4/1/2016	103436.9095



### Student Notes

### Step by Step

Objective: Merge the queries together to create the final Budget Fact table

1. Home > **Merge Queries** > Select CatSegDim
  - a) From the CatSegDim **highlight** both [Category] and [Segment]
  - b) Go back up to BudgetFact, **highlight** both [Category] and [Segment]
  - c) Show the Join Kinds available, and leave “Left Outer”
  - d) **Expand** [NewColumn] > **Select** “CatSegID” and **deselect** “Use Original column name as prefix”
2. **Remove:** [Category], [Segment]
3. **Reorder:** [CatSegID], [Scenario], [Date], [Budget Amount]
4. **Disable** the load of BudgetFact\_Data



### Create Parameters

- Actual File
- Budget File
- Path

The screenshot shows the 'Parameters' dialog box with the following details:

- Name:** Actuals\_File
- Description:** (empty)
- Required:** ☒
- Type:** Text
- Suggested Values:** Any value
- Current Value:** VanArsdel\_Actuals.xlsx

The left pane lists existing parameters: Path, Actuals\_File, and Budget\_File.

## Step by Step

**Objective:** Update file paths to use dynamic variables

1. From the Home Ribbon > Manage Parameters
2. **Create** a new Parameter
  - a) Parameter Name: Path
  - b) Type: Text
  - c) Current Value = C:\Power BI\_Adv\_M\
3. **Create** a new Parameter
  - a) Parameter Name: Actuals\_File
  - b) Type: Text
  - c) Current Value = VanArsdel\_Actuals.xlsx
4. **Create** a new Parameter
  - a) Parameter Name: Budget\_File
  - b) Type: Text
  - c) Current Value = VanArsdel\_Budget.csv
5. **Update** the text files to ensure the parameter names are consistent

## Student Notes

## Module 4 Lab: Dynamic Path to Excel Source File



### Use blank query to create Dynamic Path

- Uses Blank Query
- Populate Advanced Editor of query with text from file provided
- For each query which uses the Excel source, update the Source (applied step) with the new variable name **=Actuals\_Path**

The screenshot shows the Advanced Editor with the following M code for the 'Actuals\_Path' query:

```
let
    FilePath = Path, //External reference to text query = FilePath
    FileName = Actuals_Path, // 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.Contents(FullPath), null, true)
    else Excel.Workbook(File.Contents(FullPath), null, true)
in
    Source
```

Below the editor is a table listing data items:

APC Name	Data	APC Item	APC Kind	Hidden
1	Date	Date	Sheet	FALSE
2	Campaign	Campaign	Sheet	FALSE
3	Customer	Customer	Sheet	FALSE
4	Product	Product	Sheet	FALSE
5	Geo	Geo	Sheet	FALSE
6	Sales	Sales	Sheet	FALSE
7	DateDim	DateDim	Table	FALSE
8	CampaignDim	CampaignDim	Table	FALSE
9	CustomerDim	CustomerDim	Table	FALSE
10	ProductDim	ProductDim	Table	FALSE

On the right, the Query Settings pane shows the 'Source' step highlighted in the 'APPLIED STEPS' section.

## Step by Step

Objective: Create a query to validate if source is Web or Local and resolve path

1. Create a new blank query
  - a) Query Name: "Actuals\_Path"
  - b) Copy in text from Actuals\_Path.txt
2. Update Source Applied Step to use Resolved Path = "Actuals\_Path" to the following Queries:
  - CampaignDim
  - CustomerDim
  - ProductDim
  - CatSegDim
  - DateDim
  - GeoDim
  - SalesFact

## Student Notes



Use the **Actuals\_Path** query as a variable in other queries

- In Advanced Editor, update the Excel source to the new Actuals\_Path source.

CampaignDim	Original
<pre>let     Source = Excel.Workbook(File.Contents("C:\PowerBI_200_M\PowerBI_200_M_Data.xlsx"), null, true),     CampaignDim_Table = Source[{"Item="CampaignDim",Kind="Table"}][Data],     #"Changed Type" = Table.TransformColumnTypes(CampaignDim_Table,{"CampaignId", type text}, {"Traffic Chann in     #"Changed Type"</pre>	
CampaignDim	Updated
<pre>let     Source = Actuals_Path,     Campaign_Sheet = Source[{"Item="CampaignDim",Kind="Table"}][Data],     #"Changed Type" = Table.TransformColumnTypes(Campaign_Sheet,{"CampaignID", Int64.Type}, {"Traffic in     #"Changed Type"</pre>	

## Step by Step

## Student Notes

## Module 4 Lab: Dynamic Path to CSV Source File



Follow a similar pattern to make the Budget CSV file dynamic

- Uses Blank Query
- Populate Advanced Editor of query with text from file provided
- For each query which uses the Excel source, update the Source (applied step) with the new variable name  
=Budget\_Path

```
Budget_Path

let
    filepath = Path, //Internal reference to text query = filepath
    BudgetFactData = Budget_Fact, // wrapping comment line
    PathSlash = if Text.StartsWith(filepath, "\\") then "/" else "\",
    FullPath = filepath & (if Text.EndsWith(filepath, PathSlash) then "" else PathSlash) & BudgetFactData,
    Source = if Text.StartsWith(filepath, "http")
        then Csv.Document(Web.Contents(FullPath), [Delimiter=";", Encoding=1252, QuoteStyle=QuoteStyle.None])
        else Csv.Document(File.Contents(FullPath), [Delimiter=";", Encoding=1252, QuoteStyle=QuoteStyle.None])
in
    Source
```

Column1	Column2	Column3	Column4	Column5
Budget Spread...				
		Forecast	Forecast	Forecast
		2016	2016	2016
Category	Segment	Dec	Nov	Oct
Accessory	Accessory	44190.57888	50598.81566	54740.57
Mix	All Season	11442.14474	14120.78693	18109.64
Productivity	Productivity	19538.89812	17597.55926	22835.18
Mix	Select	311.708775	172.2601125	662.7912

## Step by Step

Objective: Update CSV file paths to use dynamic variables

1. **Create** a new blank query
  - a) Query Name: "Budget\_Path"
  - b) Copy in text from Budget\_Path.txt
2. **Update** Source Applied Step to use ResolvedBudgetPath = "Budget\_Path" to the following Queries:
  - BudgetFact
  - BudgetFact\_Data

## Student Notes



## Module 4 Lab: Custom Function



Create a function to get number of days from start of year

- Uses Blank Query
- Populate Advanced Editor of query with text from file provided
- Invoke the custom function from Sales table. This will provide the number of days from start of year for each transaction

Advanced Editor

```
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
```

✓ No syntax errors have been detected.

Done Cancel

Queries (10)

Inputs (3)

- Hqize Queries (2)
- Data Model (2)
- Fact (4)
- Sales
- BudgetFact
- TransactionDate (1/1/2011)
- fn\_DaySinceYearStart
- Dimensions (2)
- Other Queries (2)

Table: AddColumn("Changed Type", "fn\_DaySinceYearStart", each fn\_DaySinceYearStart([Date]))

	Product	Date	Category	Subcategory	Item	Quantity	Unit Price	DaysFromYearStart
1	436	5/25/2011	7020	22	2	26		
2	436	5/25/2011	22185	22	2	26		
3	613	5/14/2012	22184	22	2	23		
4	613	5/14/2012	7088	22	2	23		
5	613	5/14/2012	22185	22	2	23		
6	613	5/14/2012	22187	22	2	23		
7	613	5/14/2012	20174	22	2	23		
8	487	6/10/2012	20824	22	2	23		
9	487	6/10/2012	22477	22	2	23		
10	613	6/10/2012	20820	22	2	23		

Query Settings

PROPERTIES

Name: fn\_DaySinceYearStart

APPLIED STEPS

Source: fn\_DaySinceYearStart

Promoted Headers: fn\_DaySinceYearStart

Output Type: Int64 (Long)

## Step by Step

Objective: Create a custom function

1. **Create** a new blank query
  - a) Query Name: "fn\_DaySinceYearStart"
  - b) In Advanced Editor
  - c) 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
3. **Update** Sales query. Add Column -> Invoke Custom Function
  - a) New Column Name: DaysFromYearStart
  - b) Function query: fn\_DaySinceYearStart
  - c) Transaction Date = Date

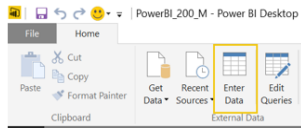
## Student Notes

## Module 4 Lab: Create a Change Log



### Use Enter Data to create a Change Log

- Create a table within query to house the change log
- Name the query **Change Log**
- Update table each time a change is made to the code
- If you decide to load file to data model, then you have the ability to report on current version and last update date



#### Create Table

Create a table by typing or pasting content.

	Version	Date	Made By	Change	
1	2.0	12/15/2016	V-Barran (Barbara)	Version 2 Updates	
2	2.01	1/3/2017	V-Barran (Barbara)	Lab Updates	
*					

#### Report

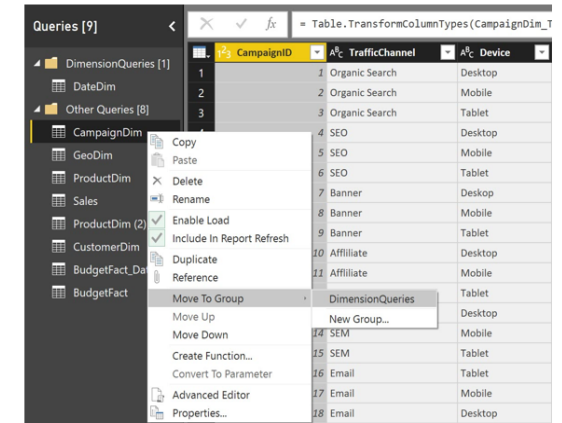
Version	Change	Date
2.00	Version 2 Updates	12/15/16
2.01	Lab Updates	1/3/17

## Module 4 Lab: Organize your Queries



### Use Folders to Group Queries

- Create logical groups to Manage your queries
  - Can create sub-groups
- Use Drag and Drop to move queries to folders



## Student Notes

## Student Notes