Microsoft Power Tools for Data Analysis #5

Power Query: Append All Tables in Current Workbook

Notes from Video:

Table of Contents:

1. **Goal of Video** is to learn how to use the Excel.CurrentWorkbook Function to combine or append all the Excel Tables in an Excel Workbook File into a single Proper Data Set. The idea is presented in this picture:

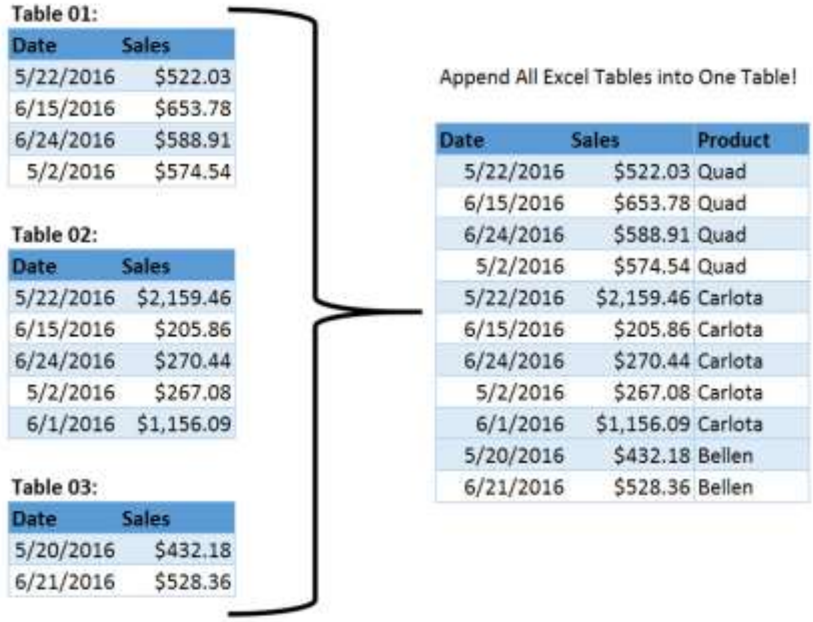


Table 01:

Date	Sales
5/22/2016	$522.03
6/15/2016	$653.78
6/24/2016	$588.91
5/2/2016	$574.54

Table 02:

Date	Sales
5/22/2016	$2,159.46
6/15/2016	$205.86
6/24/2016	$270.44
5/2/2016	$267.08
6/1/2016	$1,156.09

Table 03:

Date	Sales
5/20/2016	$432.18
6/21/2016	$528.36

Append All Excel Tables into One Table!

Date	Sales	Product
5/22/2016	$522.03	Quad
6/15/2016	$653.78	Quad
6/24/2016	$588.91	Quad
5/2/2016	$574.54	Quad
5/22/2016	$2,159.46	Carlota
6/15/2016	$205.86	Carlota
6/24/2016	$270.44	Carlota
5/2/2016	$267.08	Carlota
6/1/2016	$1,156.09	Carlota
5/20/2016	$432.18	Bellen
6/21/2016	$528.36	Bellen

2. **Each Excel Table on New Sheet**. In the file we use for the video, the Excel Tables that we want to append into a single Proper Data Set are on different sheets as seen in this picture:

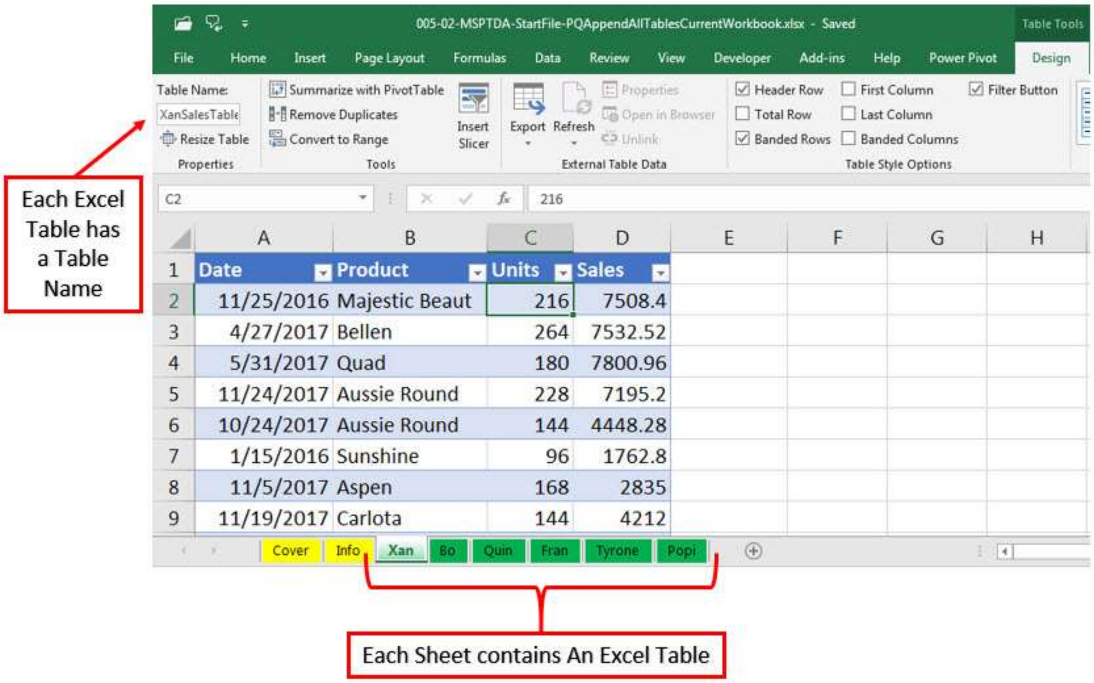


Table Name: XanSalesTable

C2 216

	A	B	C	D
1	Date	Product	Units	Sales
2	11/25/2016	Majestic Beaut	216	7508.4
3	4/27/2017	Bellen	264	7532.52
4	5/31/2017	Quad	180	7800.96
5	11/24/2017	Aussie Round	228	7195.2
6	10/24/2017	Aussie Round	144	4448.28
7	1/15/2016	Sunshine	96	1762.8
8	11/5/2017	Aspen	168	2835
9	11/19/2017	Carlota	144	4212

Each Excel Table has a Table Name

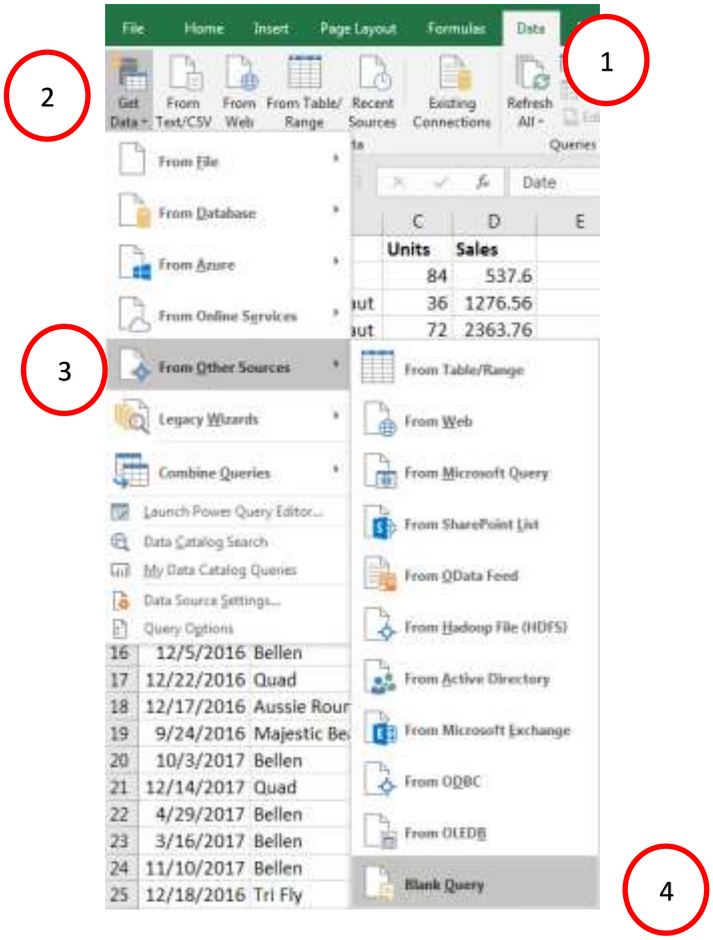
Each Sheet contains An Excel Table

3. **What does Excel.CurrentWorkbook Function Import into Query Editor?**
   1) Excel Tables (Using Excel Table feature)
   2) Defined Names
   3) Print Ranges
   4) Filtered Ranges

4. **Potential Problems with using Excel.CurrentWorkbook Function**:
    1) If you load the Appended Table to an Excel Worksheet, because Power Query Loads data to a Worksheet as an Excel Table, when you try to refresh the Power Query Load to the Worksheet is imported and the number of records will be Doubled. This is called the Recursion Problem because the Query (which imports Excel Tables) The quick fix is to filter the Table Name column so that it values are not equal to the Query Name.
    2) If you have Defined Names, Print Ranges, Filtered Ranges that you do not want appended into the new table, then you must edit these objects out, as seen in the video and shown later in the document.
5. **Example1: Append all Excel Tables with Recursion Problem in Current Workbook To Worksheet**. We will see the Recursion Problem and solve it by filtering out the Query/Table Name.

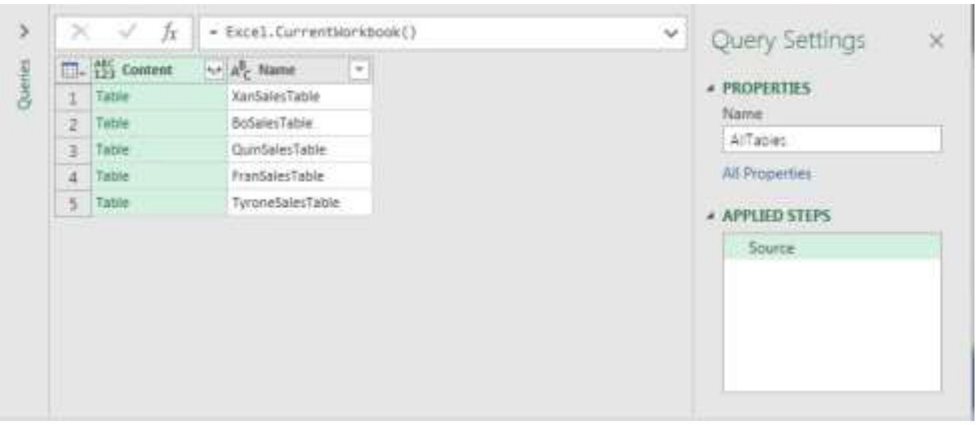
Steps for Example #1:

1) In Example #1 we have a Workbook File with Excel Tables, but no Defined Names.
2) **Create a Blank Query**.
    i. Click to the Data Ribbon Tab
    ii. Click Get Data dropdown arrow in Get & Transform group
    iii. Hover cursor over From Other Sources
    iv. From the dropdown list click on Blank Query, as seen in this picture:



3) Name the Query “AllTables”.

4) **In the Formula Bar type =Excel.CurrentWorkbook()**, then hit Enter, as seen in this picture:



= Excel.CurrentWorkbook()

	Content	Name
1	Table	XanSalesTable
2	Table	BoSalesTable
3	Table	QuinSalesTable
4	Table	FranSalesTable
5	Table	TyroneSalesTable

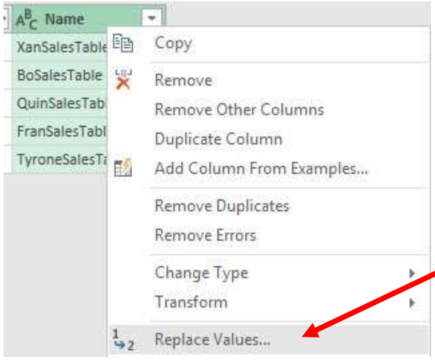
Name: AllTables

Source

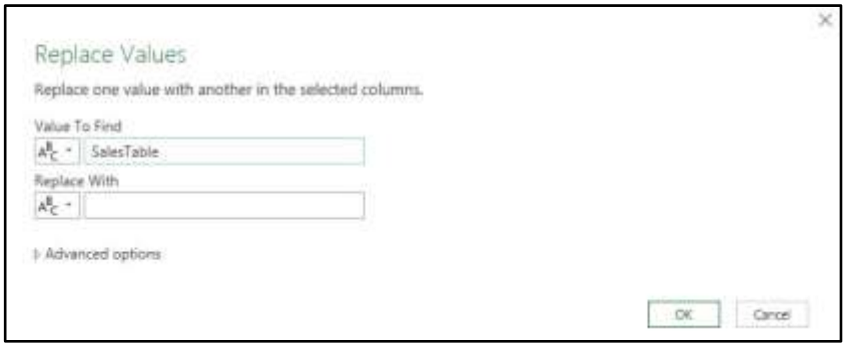
5) As seen in the above picture:
   i. The Content Column contains a full Excel Table for each row.
   ii. The Name Column contains the name of Each Excel Table.

6) **Define Table Object**.
   i. Table = Set of records for a set of Columns or Fields.

7) **Extract SalesRep Name with Replace Feature**. Right-click the Name Column and click on Replace Values , as seen in this picture:



8) In the Replace Values dialog box, type "SalesTable" in the Value To Find textbox and then hit OK , as seen in this picture:



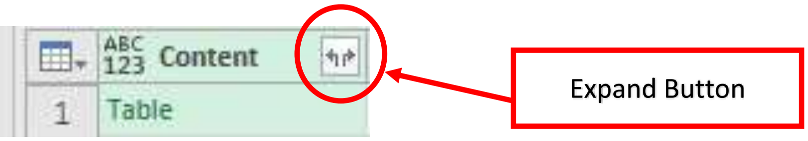
Replace Values

Replace one value with another in the selected columns.

Value To Find: SalesTable

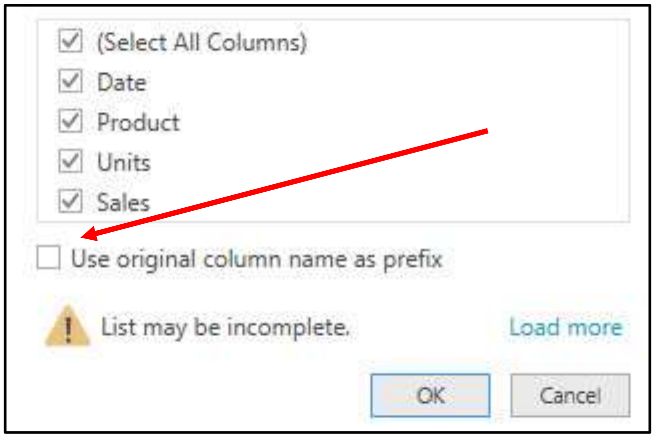
Replace With:

9) **Expand Tables**. To Append the Excel Tables into a single Proper Data Set, click the Expand Button in the Content Column Name, as seen in this picture:



Expand Button

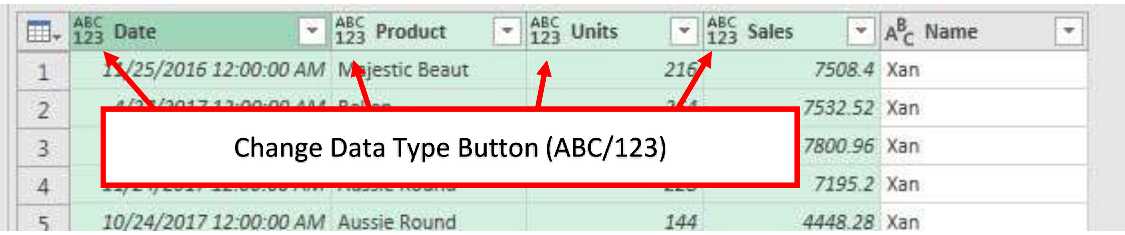
10) After clicking the Expand Button, uncheck “Use original column name as prefix”, as seen in this picture:



☑ (Select All Columns)
☑ Date
☑ Product
☑ Units
☑ Sales

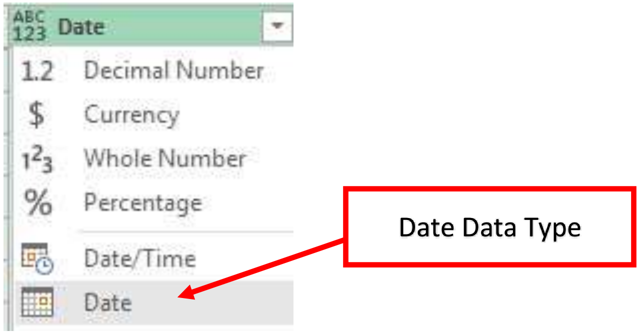
□ Use original column name as prefix

11) **To Change the Data Types** for each column, use the Change Data Type Button (ABC/123) at the top of each column, as seen in this picture:

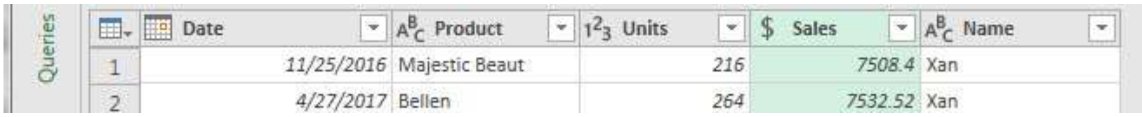


	ABC 123 Date	ABC 123 Product	ABC 123 Units	ABC 123 Sales	A B C Name
1	11/25/2016 12:00:00 AM	Majestic Beaut	216	7508.4	Xan
2	[illegible]	[illegible]	[illegible]	7532.52	Xan
3	[illegible]	[illegible]	[illegible]	7800.96	Xan
4	[illegible]	[illegible]	[illegible]	7195.2	Xan
5	10/24/2017 12:00:00 AM	Aussie Round	144	4448.28	Xan

12) For example, for the Date Column, select the Data Type “Date” , as seen in this picture:

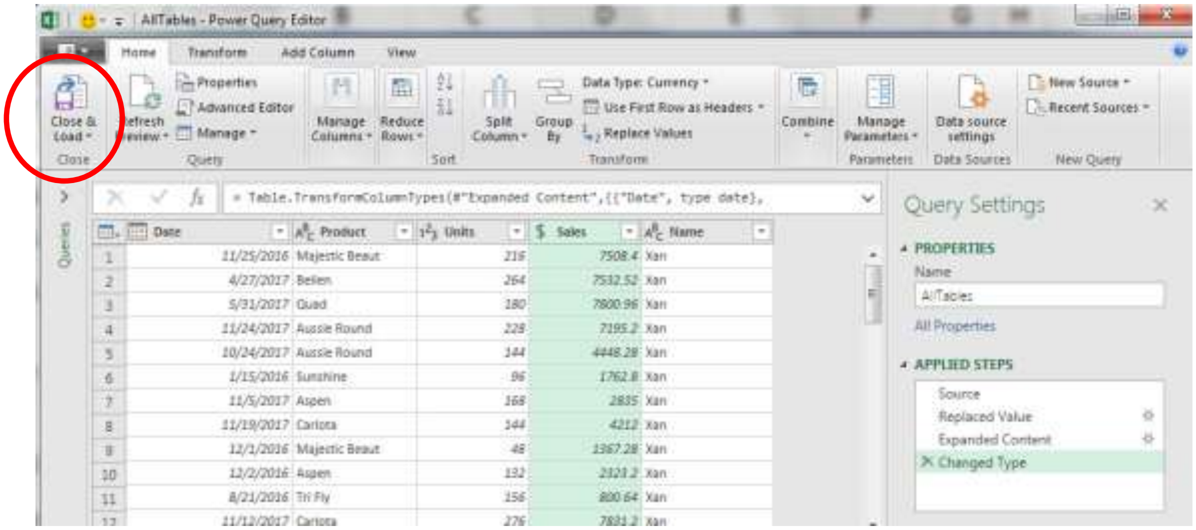


13) The Final Data Types should look like this:



	Date	A B C Product	1²₃ Units	$ Sales	A B C Name
1	11/25/2016	Majestic Beaut	216	7508.4	Xan
2	4/27/2017	Bellen	264	7532.52	Xan

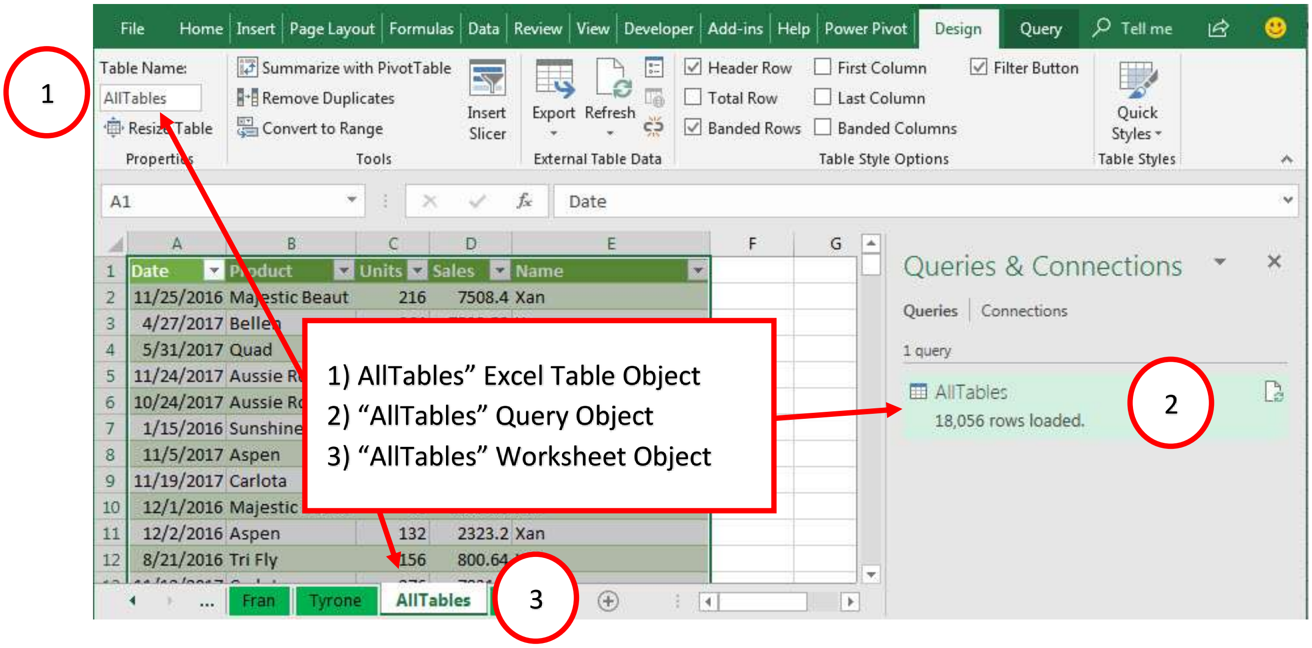
14) **Close & Load to Worksheet**. The final Appended Proper Data Set is seen in the Power Query Editor in the picture below. To Close and Load to an Excel Sheet, click the Close & Load Button:



= Table.TransformColumnTypes(#"Expanded Content",{{"Date", type date},

	Date	Product	Units	Sales	Name
1	11/25/2016	Majestic Beaut	216	7508.4	Xan
2	4/27/2017	Bellen	264	7532.52	Xan
3	5/31/2017	Quad	180	7800.96	Xan
4	11/24/2017	Aussie Round	228	7195.2	Xan
5	10/24/2017	Aussie Round	144	4448.28	Xan
6	1/15/2016	Sunshine	96	1762.8	Xan
7	11/5/2017	Aspen	168	2835	Xan
8	11/19/2017	Carlota	144	4212	Xan
9	12/1/2016	Majestic Beaut	48	1367.28	Xan
10	12/2/2016	Aspen	132	2323.2	Xan
11	8/21/2016	Tri Fly	156	800.64	Xan
12	11/12/2017	Carlota	276	7831.2	Xan

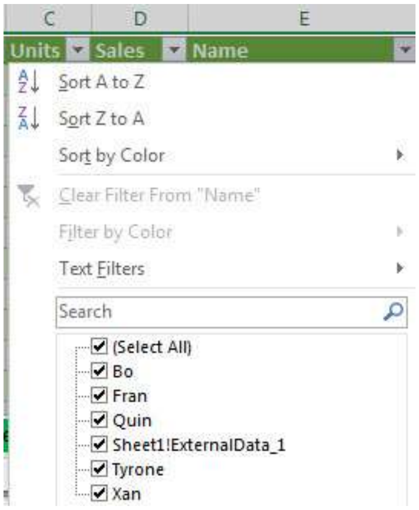
15) Here is the final Appended Proper Data Set that is loaded to an Excel Worksheet with the Worksheet Name changed to "AllTables", as seen in the picture below. Notice that the Query Name, Worksheet Name and Excel Table Name are all the same. The names are not in conflict because each one is a different Object: 1) "AllTables" Excel Table Object, 2) "AllTables" Query Object and "AllTables" Worksheet Object.



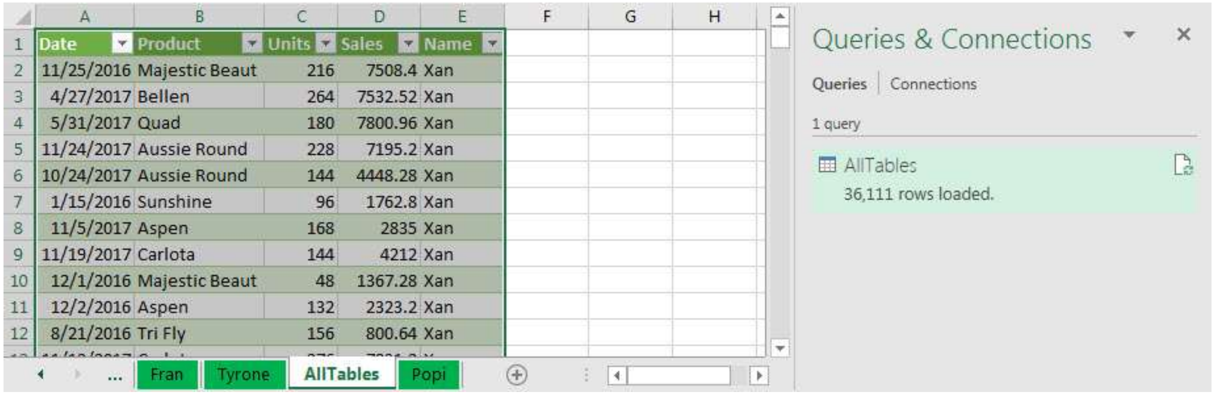
	A	B	C	D	E
1	Date	Product	Units	Sales	Name
2	11/25/2016	Majestic Beaut	216	7508.4	Xan
3	4/27/2017	Bellen			
4	5/31/2017	Quad			
5	11/24/2017	Aussie R			
6	10/24/2017	Aussie Ro			
7	1/15/2016	Sunshine			
8	11/5/2017	Aspen			
9	11/19/2017	Carlota			
10	12/1/2016	Majestic			
11	12/2/2016	Aspen	132	2323.2	Xan
12	8/21/2016	Tri Fly	156	800.64	

1) AllTables" Excel Table Object
2) "AllTables" Query Object
3) "AllTables" Worksheet Object

16) **One Record Too Many**. In the previous picture the AllTables Query loaded 18,056 rows to the Worksheet. This is one record more than there are in the combined rows from all the Excel Tables in the Excel Workbook File. The combined number of rows in all the Excel Tables in the Excel File is only 18,055. This one extra record is caused by a Recursion Problem that stems from the fact that the Excel.CurrentWorkbook Function imports all the Excel Workbooks in the Excel File, including the Excel Table that the Query loads to the Excel Worksheet. If we click the Filter dropdown in the Name Column (names from all the Excel Tables), you can see that there is one extra name that refers to the Excel Table output that the Query delivers to the Worksheet.



17) **Recursion Problem**. The Recursion Problem stems from the fact that the Excel.CurrentWorkbook Function imports all the Excel Workbooks in the Current Excel File, including the Excel Table that the Query loads to the Excel Worksheet. This means that the Query will try and import the very table that it delivers as output. If you refresh the query (right-click Query Output, right-click Query Name or use the Refresh button in Queries & Connections Pane), you will get double the number of records from the original Excel Tables because the Query import all the original Excel Tables and the Appended Excel Table that the Query loaded to the sheet. In the below picture you can see that after a refresh, there are 36,111 rows of data (double the original amount):



Date	Product	Units	Sales	Name
11/25/2016	Majestic Beaut	216	7508.4	Xan
4/27/2017	Bellen	264	7532.52	Xan
5/31/2017	Quad	180	7800.96	Xan
11/24/2017	Aussie Round	228	7195.2	Xan
10/24/2017	Aussie Round	144	4448.28	Xan
1/15/2016	Sunshine	96	1762.8	Xan
11/5/2017	Aspen	168	2835	Xan
11/19/2017	Carlota	144	4212	Xan
12/1/2016	Majestic Beaut	48	1367.28	Xan
12/2/2016	Aspen	132	2323.2	Xan
8/21/2016	Tri Fly	156	800.64	Xan

Queries & Connections

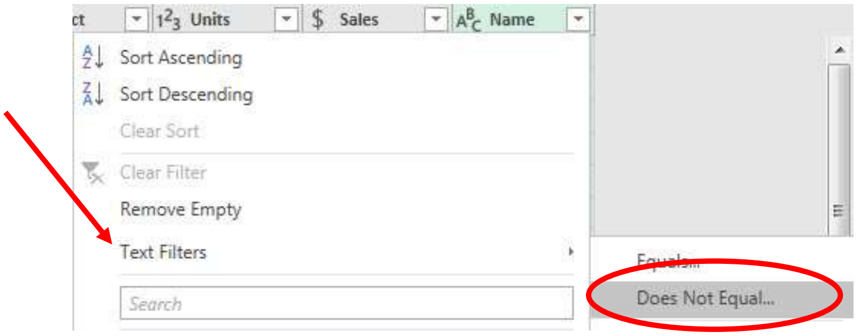
1 query

AllTables
36,111 rows loaded.

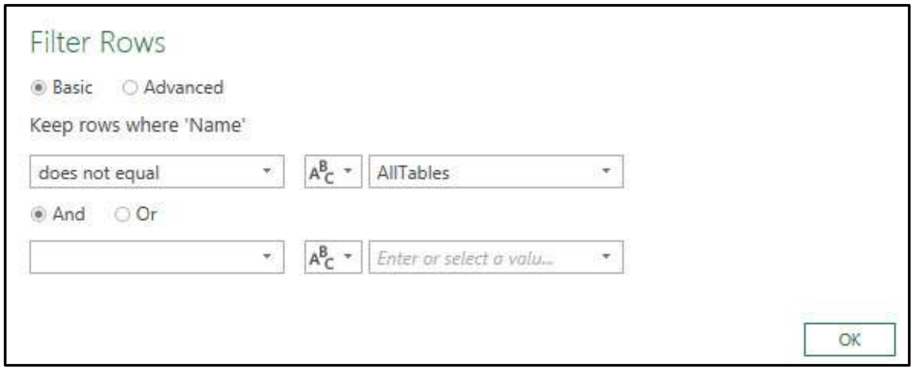
18) **Fix Recursion Problem**. To fix the Recursion Problem, we can:

i. Double click the Query Name in the Queries & Connections Pane.

ii. Click the Filter dropdown in the Name Column, Point to Text Filter, Click on "Does Not Equal", as seen in this picture:



19) Then in the Filter Dialog Box type the Query Name "AllTables", as seen in this picture:



Filter Rows

● Basic ○ Advanced

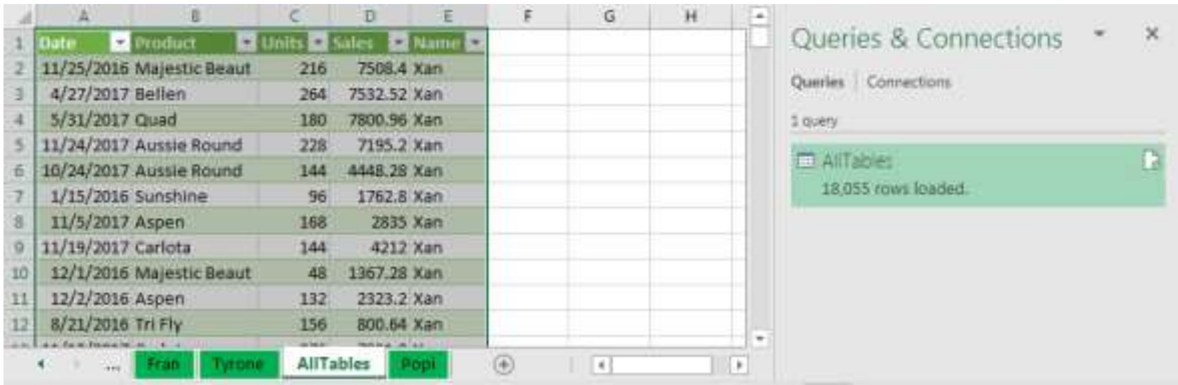
Keep rows where 'Name'

does not equal | ABC | AllTables

● And ○ Or

| ABC | Enter or select a valu...

20) Now we can see that the Query Output, an Excel Table named "AllTables" will not be imported when we refresh, and we will get the correct total number of rows of data of 18,055, as seen in this picture:



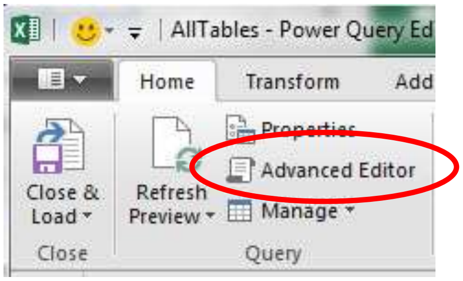
Date	Product	Units	Sales	Name
11/25/2016	Majestic Beaut	216	7508.4	Xan
4/27/2017	Bellen	264	7532.52	Xan
5/31/2017	Quad	180	7800.96	Xan
11/24/2017	Aussie Round	228	7195.2	Xan
10/24/2017	Aussie Round	144	4448.28	Xan
1/15/2016	Sunshine	96	1762.8	Xan
11/5/2017	Aspen	168	2835	Xan
11/19/2017	Carlota	144	4212	Xan
12/1/2016	Majestic Beaut	48	1367.28	Xan
12/2/2016	Aspen	132	2323.2	Xan
8/21/2016	Tri Fly	156	800.64	Xan

Queries & Connections

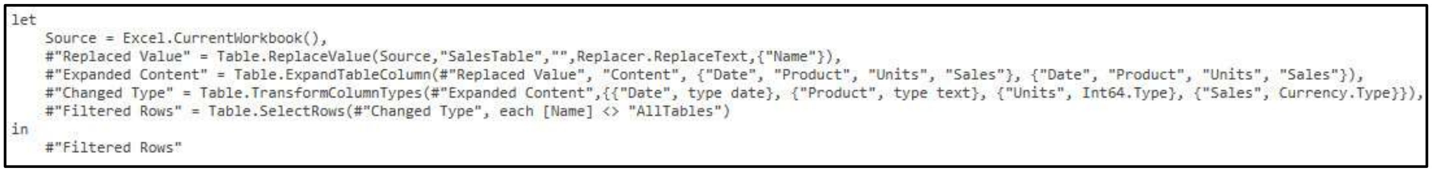
AllTables
18,055 rows loaded.

21) **Add New Excel Tables to Excel Workbook File**. Now Go to the Popi Worksheet Tab and convert the Proper Data Set to an Excel Table and name the Table (just as we did in previous videos in this class and in the prerequisite classes). Then when you refresh the Query (right-click Query Output, right-click Query Name or use the Refresh button in Queries & Connections Pane), the new Popi Data will be included in the table.

22) **To View the Full M Code from Example #1** that was created by using the Power Query User Interface, click on the Advanced Editor button in the Query group in the Home Ribbon Tab, as seen in this picture:



23) Here in the M Code from Example #1:



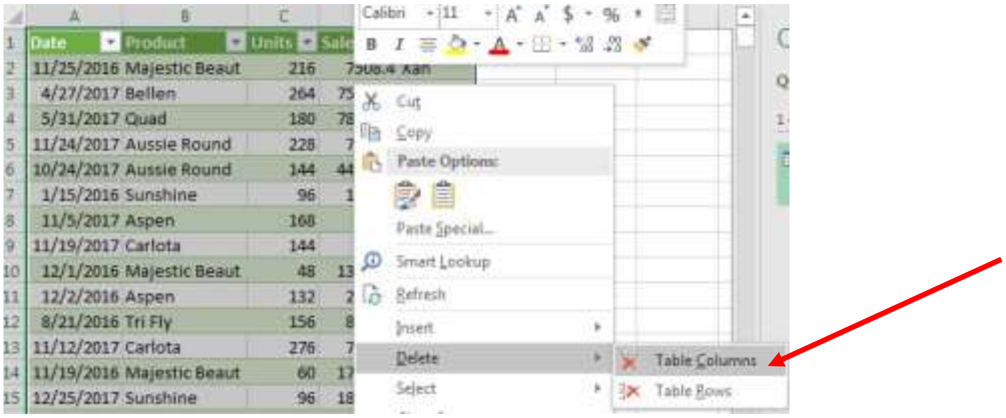
```
let
    Source = Excel.CurrentWorkbook(),
    #"Replaced Value" = Table.ReplaceValue(Source,"SalesTable","",Replacer.ReplaceText,{"Name"}),
    #"Expanded Content" = Table.ExpandTableColumn(#"Replaced Value", "Content", {"Date", "Product", "Units", "Sales"}, {"Date", "Product", "Units", "Sales"}),
    #"Changed Type" = Table.TransformColumnTypes(#"Expanded Content",{{"Date", type date}, {"Product", type text}, {"Units", Int64.Type}, {"Sales", Currency.Type}}),
    #"Filtered Rows" = Table.SelectRows(#"Changed Type", each [Name] <> "AllTables")
in
    #"Filtered Rows"
```

6. **Example2: Append all Excel Tables in Current Workbook To PivotTable Cache & make PivotTable Report**.

   1) As we saw in video number 3 in this class, when we load the Power Query Output (Proper Data Set) to the PivotTable Cache, we can build PivotTable Reports without loading the data to an Excel Worksheet. If your goal is to build Standard PivotTable Reports, then there is no need to load the Proper Data Set to an Excel Sheet. If we load to the PivotTable Cache, rather than to an Excel Worksheet, we will not have the Recursion Problem.

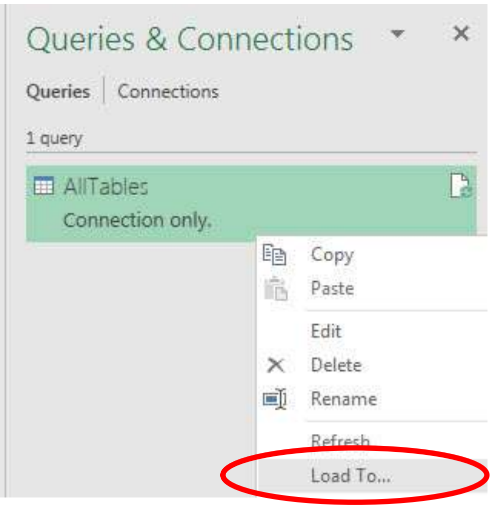
Steps for Example #2:

2) To Remove the Power Query Output (Appended Excel Tables into Proper Data Set) from the Excel Worksheet, select the entire table, then right-click, point to Delete, then in the sub-menu click on Table Columns, as seen in this picture:

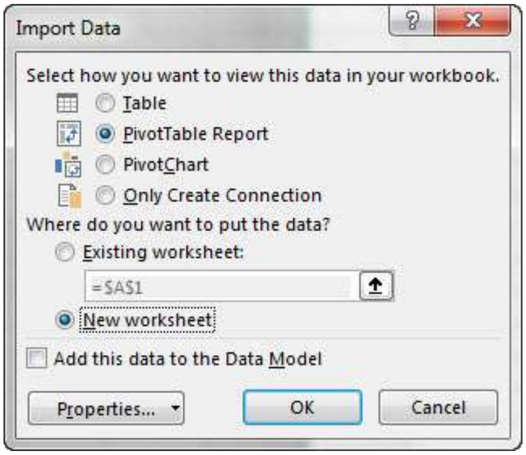


	Date	Product	Units	Sale
2	11/25/2016	Majestic Beaut	216	7508.4 Xan
3	4/27/2017	Bellen	264	75
4	5/31/2017	Quad	180	78
5	11/24/2017	Aussie Round	228	7
6	10/24/2017	Aussie Round	144	44
7	1/15/2016	Sunshine	96	1
8	11/5/2017	Aspen	168	
9	11/19/2017	Carlota	144	
10	12/1/2016	Majestic Beaut	48	13
11	12/2/2016	Aspen	132	2
12	8/21/2016	Tri Fly	156	8
13	11/12/2017	Carlota	276	7
14	11/19/2016	Majestic Beaut	60	17
15	12/25/2017	Sunshine	96	18

3) **To Load the Query Output to the PivotTable Cache**, right-click the Query Name, “AllTables” in the Queries & Connections Pane and then click on “Load To…”, as seen in this picture:



4) **Import Data dialog box**. In the Import Data dialog box, select the dialog buttons for “PivotTable Report” and “New worksheet” and then click OK, as seen in this picture:


Import Data

Select how you want to view this data in your workbook.

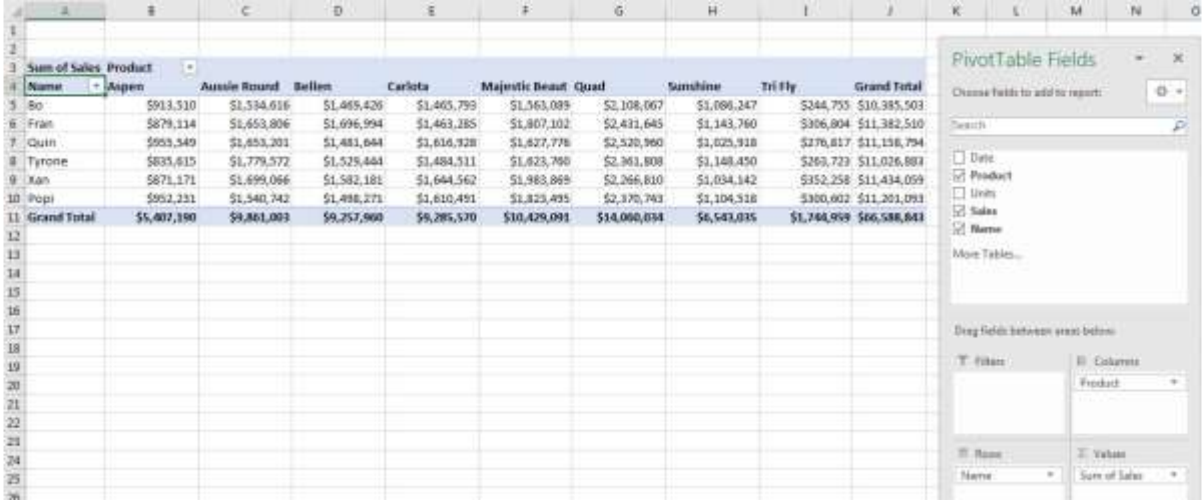
○ Table
● PivotTable Report
○ PivotChart
○ Only Create Connection

Where do you want to put the data?

○ Existing worksheet:
=A1
● New worksheet

□ Add this data to the Data Model

5) **PivotTable Report**. Then from the PivotTable Field List build a report as seen in this picture:


Sum of Sales	Product								
Name	Aspen	Aussie Round	Bellen	Carlota	Majestic Beaut	Quad	Sunshine	Tri Fly	Grand Total
Bo	$913,510	$1,534,616	$1,469,426	$1,465,793	$1,563,089	$2,108,067	$1,086,247	$244,755	$10,385,503
Fran	$879,114	$1,653,806	$1,696,994	$1,463,285	$1,807,102	$2,431,645	$1,143,760	$306,804	$11,382,510
Quin	$955,349	$1,653,201	$1,481,644	$1,616,928	$1,627,776	$2,520,960	$1,025,918	$276,817	$11,158,794
Tyrone	$835,615	$1,779,572	$1,529,444	$1,484,511	$1,623,780	$2,361,808	$1,148,450	$263,723	$11,026,883
Xan	$871,171	$1,699,066	$1,582,181	$1,644,562	$1,983,869	$2,266,810	$1,034,142	$352,258	$11,434,059
Popi	$952,231	$1,540,742	$1,498,271	$1,610,491	$1,825,495	$2,370,743	$1,104,518	$300,602	$11,201,093
Grand Total	$5,407,190	$9,861,003	$9,257,960	$9,285,570	$10,429,091	$14,060,034	$6,543,035	$1,744,959	$66,588,843

PivotTable Fields

Choose fields to add to report:

□ Date
☑ Product
□ Units
☑ Sales
☑ Name

Drag fields between areas below:

Filters

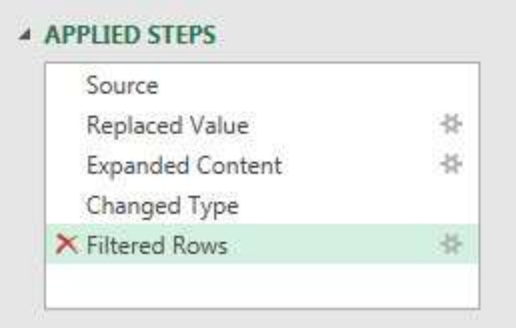
Columns: Product

Rows: Name

Values: Sum of Sales

6) **Edit Query**. Notice in the above report and PivotTable Field List that the column name for the SalesReps is “Names”. We want to edit this and change the Column Name to “SalesRep”. To edit the Query and change a Field Name in the Proper Data Set, we can double-click the “AllTables” Query Name in the Queries & Connections Pane. This will open the Query in the Power Query Editor Window.

7) Then in the Applied Steps List, delete the unnecessary last step (Filter to fix Recursion Problem) by clicking on the Red **X** next to Filtered Rows, as seen in this picture:



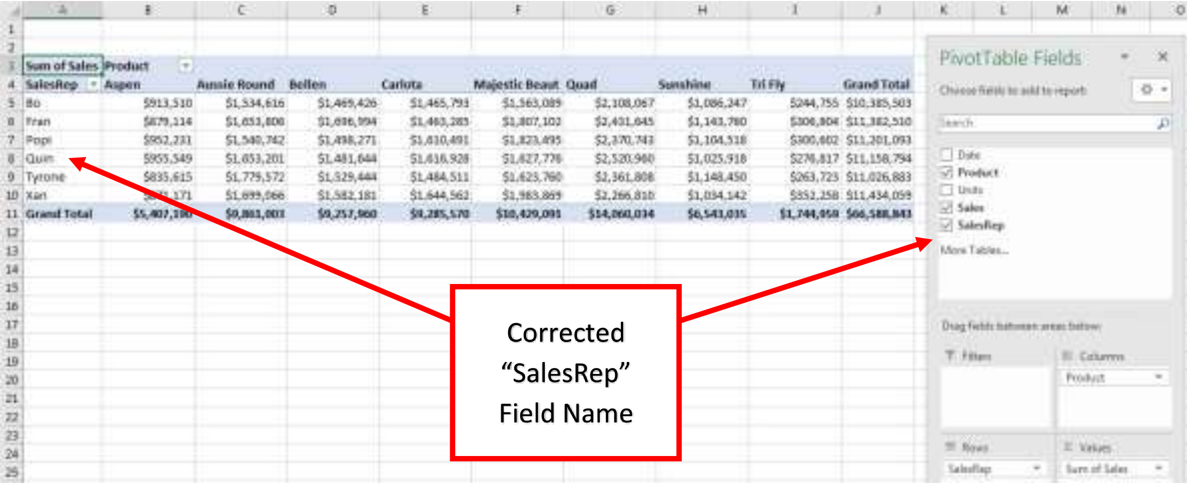
APPLIED STEPS

Source
Replaced Value
Expanded Content
Changed Type
✕ Filtered Rows

8) To re-name the Name Column, double-click the Field Name “Names”, type “Sales Rep” and then hit Enter.

9) You can then click the Close and Load button to load the corrected Query Output back to the PivotTable Cache.

10) You will then need to adjust the PivotTable to get this final Report:



Sum of Sales	Product								
SalesRep	Aspen	Aussie Round	Bellen	Carlota	Majestic Beaut	Quad	Sunshine	Tri Fly	Grand Total
Bo	$913,510	$1,534,616	$1,469,426	$1,465,793	$1,563,089	$2,108,067	$1,086,247	$244,755	$10,385,503
Fran	$879,114	$1,673,806	$1,696,994	$1,463,285	$1,807,102	$2,431,645	$1,143,780	$306,804	$11,382,510
Popi	$952,231	$1,540,742	$1,498,271	$1,610,491	$1,823,495	$2,370,743	$1,104,518	$300,602	$11,201,093
Quin	$955,549	$1,653,201	$1,481,644	$1,616,928	$1,627,776	$2,520,980	$1,025,918	$276,817	$11,158,794
Tyrone	$835,615	$1,779,572	$1,529,444	$1,484,511	$1,623,760	$2,361,808	$1,148,450	$263,723	$11,026,883
Xan	$871,171	$1,699,066	$1,582,181	$1,644,562	$1,983,869	$2,266,810	$1,034,142	$352,258	$11,434,059
Grand Total	$5,407,190	$9,881,003	$9,257,960	$9,285,570	$10,429,091	$14,060,034	$6,541,035	$1,744,959	$66,588,843

PivotTable Fields

Choose fields to add to report:

Search

□ Date
☑ Product
□ Units
☑ Sales
☑ SalesRep

More Tables...

Drag fields between areas below:

Filters

Columns: Product

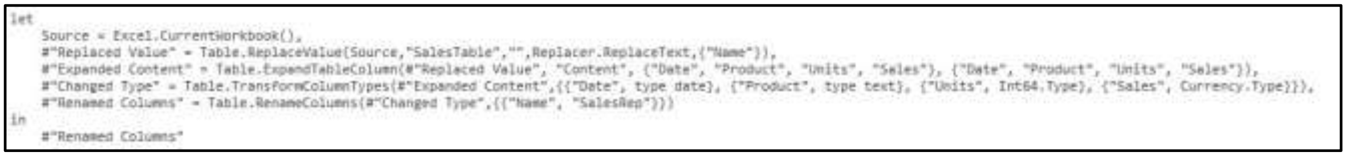
Rows: SalesRep

Values: Sum of Sales

Corrected “SalesRep” Field Name

7. The advantages to Loading the Data to the PivotTable Cache (or even the Data Model) is that we do not have a Recursion Problem when we refresh data and we only have to refresh one time when new data arrives.

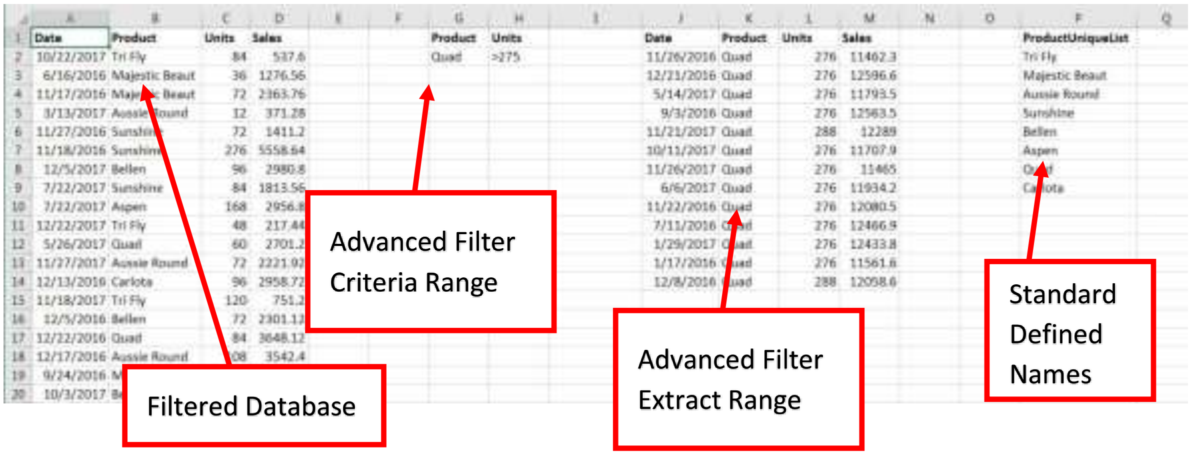
8. Here in the M Code from Example #2:



```
let
    Source = Excel.CurrentWorkbook(),
    #"Replaced Value" = Table.ReplaceValue(Source,"SalesTable","",Replacer.ReplaceText,{"Name"}),
    #"Expanded Content" = Table.ExpandTableColumn(#"Replaced Value", "Content", {"Date", "Product", "Units", "Sales"}, {"Date", "Product", "Units", "Sales"}),
    #"Changed Type" = Table.TransformColumnTypes(#"Expanded Content",{{"Date", type date}, {"Product", type text}, {"Units", Int64.Type}, {"Sales", Currency.Type}}),
    #"Renamed Columns" = Table.RenameColumns(#"Changed Type",{{"Name", "SalesRep"}})
in
    #"Renamed Columns"
```

9. **Example 3: Append all Excel Tables in Current Workbook that has Defined Names**.

1) The Data in the second file that we used in this video had Excel Tables, Defined Names, and Automatic Defined Names like Print Ranges, Filtered Ranges, Criteria Ranges and Extract Ranges. Because there are Defined Names and not just Excel Tables in our Excel Workbook File, we must be careful in building our Query so that we only import that data that we want. For our example, we only want to import Excel Tables. This means that we must filter out the Defined Names. This is tricky, because unlike the Excel.Workbook Function that we saw last video, which gives us information about the different objects that are imported in a Column called "Kind", in this video, the Excel.CurrentWorkbook does not give us information about the different objects that are imported.
2) Below is a picture of the Sheet name "Popi", which contains a number of different Defined Names. We will have to filter out the Defined Names in order to import only the Excel Tables in the Excel Workbook File.



Date	Product	Units	Sales
10/22/2017	Tri Fly	84	537.6
6/16/2016	Majestic Beaut	36	1276.56
11/17/2016	Majestic Beaut	72	2363.76
3/13/2017	Aussie Round	12	371.28
11/27/2016	Sunshine	72	1411.2
11/18/2016	Sunshine	276	5558.64
12/5/2017	Bellen	96	2980.8
7/22/2017	Sunshine	84	1813.56
7/22/2017	Aspen	168	2956.8
12/22/2017	Tri Fly	48	217.44
5/26/2017	Quad	60	2701.2
11/27/2017	Aussie Round	72	2221.92
12/13/2016	Carlota	96	2958.72
11/18/2017	Tri Fly	120	751.2
12/5/2016	Bellen	72	2301.12
12/22/2016	Quad	84	3648.12
12/17/2016	Aussie Round	108	3542.4
9/24/2016	M[illegible]	[illegible]	[illegible]
10/3/2017	B[illegible]	[illegible]	[illegible]

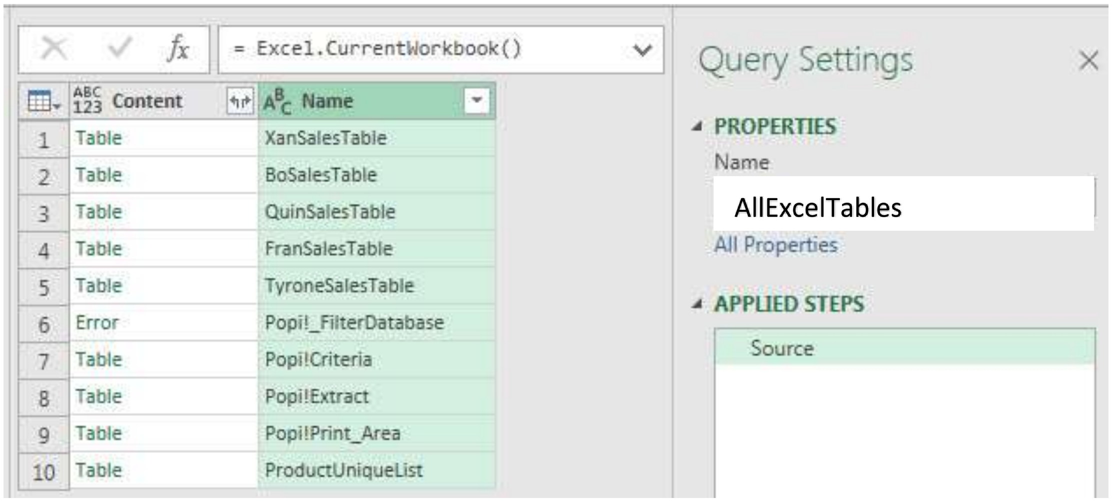
Product	Units
Quad	>275

Date	Product	Units	Sales
11/26/2016	Quad	276	11462.3
12/21/2016	Quad	276	12596.6
5/14/2017	Quad	276	11793.5
9/3/2016	Quad	276	12563.5
11/21/2017	Quad	288	12289
10/11/2017	Quad	276	11707.9
11/26/2017	Quad	276	11465
6/6/2017	Quad	276	11934.2
11/22/2016	Quad	276	12080.5
7/11/2016	Quad	276	12466.9
1/29/2017	Quad	276	12433.8
1/17/2016	Quad	276	11561.6
12/8/2016	Quad	288	12058.6

ProductUniqueList
Tri Fly
Majestic Beaut
Aussie Round
Sunshine
Bellen
Aspen
Quad
Carlota

Steps for Example #3:

3) **Create a Blank Query with the keyboard**: Alt, A, P, N, O, Q
4) Name the Query "AllExcelTables".
5) **Use the Excel.CurrentWorkbook() function to import all the Excel Tables and Defined Names**. We can see in the below picture that a number of Excel Tables and Defined Names were imported into the Power Query Editor:



```
= Excel.CurrentWorkbook()
```

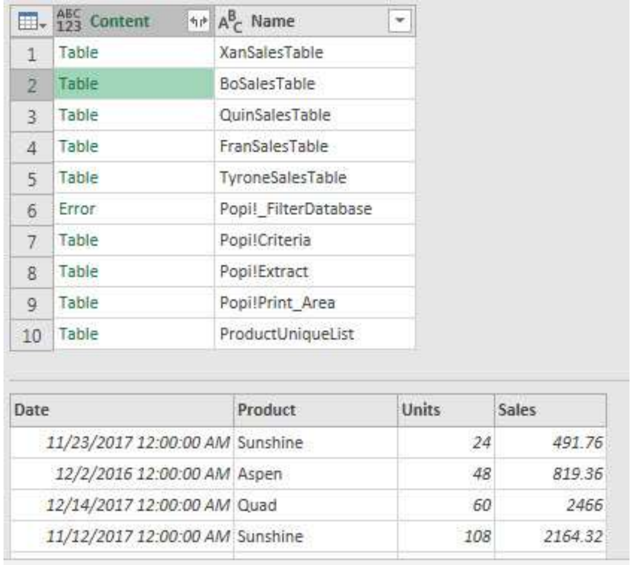
	Content	Name
1	Table	XanSalesTable
2	Table	BoSalesTable
3	Table	QuinSalesTable
4	Table	FranSalesTable
5	Table	TyroneSalesTable
6	Error	Popi!_FilterDatabase
7	Table	Popi!Criteria
8	Table	Popi!Extract
9	Table	Popi!Print_Area
10	Table	ProductUniqueList

Name: AllExcelTables

Applied Steps: Source

6) **Look at Table & Defined Name Objects in Power Query Editor Window**. In the Content Column we can click in the white area to the right of the word Table for each row and see a preview of the Table Object in the lower part of the Power Query Editor Window as see in the following pictures:

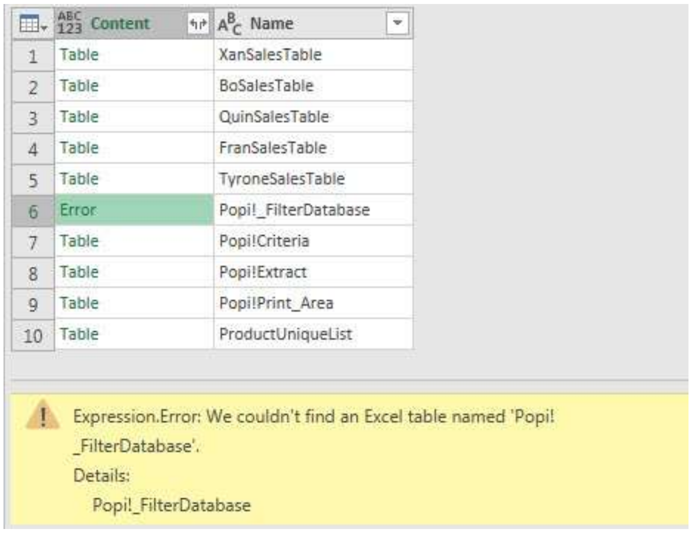
i. The Excel Table named “BoSalesTable”. We want to use this data.



	Content	Name
1	Table	XanSalesTable
2	Table	BoSalesTable
3	Table	QuinSalesTable
4	Table	FranSalesTable
5	Table	TyroneSalesTable
6	Error	Popi!_FilterDatabase
7	Table	Popi!Criteria
8	Table	Popi!Extract
9	Table	Popi!Print_Area
10	Table	ProductUniqueList

Date	Product	Units	Sales
11/23/2017 12:00:00 AM	Sunshine	24	491.76
12/2/2016 12:00:00 AM	Aspen	48	819.36
12/14/2017 12:00:00 AM	Quad	60	2466
11/12/2017 12:00:00 AM	Sunshine	108	2164.32

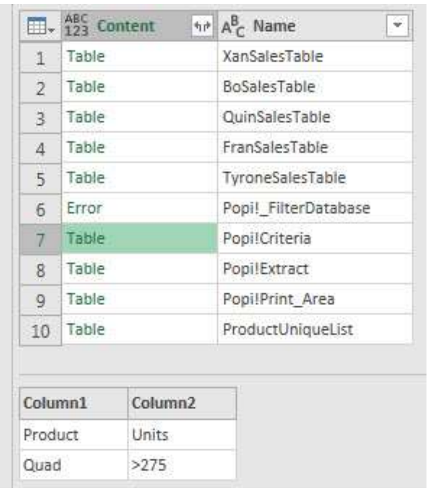
ii. The Filtered Database Error. We need to filter this out.



	Content	Name
1	Table	XanSalesTable
2	Table	BoSalesTable
3	Table	QuinSalesTable
4	Table	FranSalesTable
5	Table	TyroneSalesTable
6	Error	Popi!_FilterDatabase
7	Table	Popi!Criteria
8	Table	Popi!Extract
9	Table	Popi!Print_Area
10	Table	ProductUniqueList

Expression.Error: We couldn't find an Excel table named 'Popi!_FilterDatabase'.
Details:
Popi!_FilterDatabase

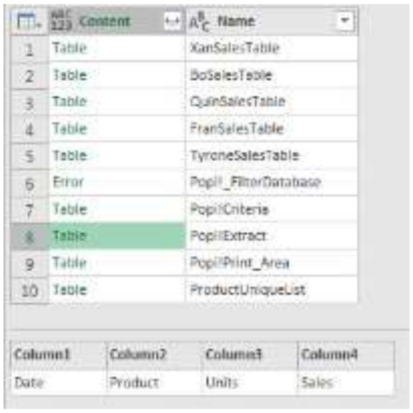
iii. The automatic Defined Name for the Advanced Filter Criteria Range. We need to filter this out. Notice that the Defined Name has two columns that are named “Column1” and “Column2”.



	Content	Name
1	Table	XanSalesTable
2	Table	BoSalesTable
3	Table	QuinSalesTable
4	Table	FranSalesTable
5	Table	TyroneSalesTable
6	Error	Popi!_FilterDatabase
7	Table	Popi!Criteria
8	Table	Popi!Extract
9	Table	Popi!Print_Area
10	Table	ProductUniqueList

Column1	Column2
Product	Units
Quad	>275

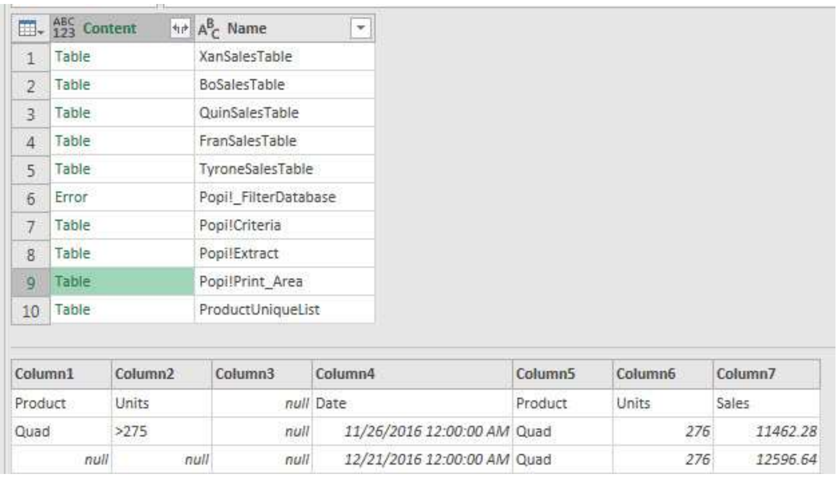
iv. The automatic Defined Name for the Advanced Filter Extract Range. We need to filter this out. Notice that the Defined Name has four columns that are named “Column1”, “Column2”, “Column3” and “Column4”.



	Content	Name
1	Table	XanSalesTable
2	Table	BoSalesTable
3	Table	QuinSalesTable
4	Table	FranSalesTable
5	Table	TyroneSalesTable
6	Error	Popi!_FilterDatabase
7	Table	Popi!Criteria
8	Table	Popi!Extract
9	Table	Popi!Print_Area
10	Table	ProductUniqueList

Column1	Column2	Column3	Column4
Date	Product	Units	Sales

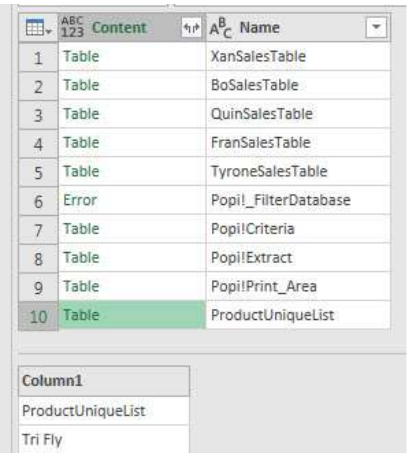
v. The automatic Defined Name for a Print Range. We need to filter this out. Notice that the Defined Name has multiple columns that are named “Column1”, “Column2” and so on.



	Content	Name
1	Table	XanSalesTable
2	Table	BoSalesTable
3	Table	QuinSalesTable
4	Table	FranSalesTable
5	Table	TyroneSalesTable
6	Error	Popi!_FilterDatabase
7	Table	Popi!Criteria
8	Table	Popi!Extract
9	Table	Popi!Print_Area
10	Table	ProductUniqueList

Column1	Column2	Column3	Column4	Column5	Column6	Column7
Product	Units	null	Date	Product	Units	Sales
Quad	>275	null	11/26/2016 12:00:00 AM	Quad	276	11462.28
null	null	null	12/21/2016 12:00:00 AM	Quad	276	12596.64

vi. The Defined Name. We need to filter this out. Notice that the Defined Name has two columns that are named “Column1” and “Column2”:

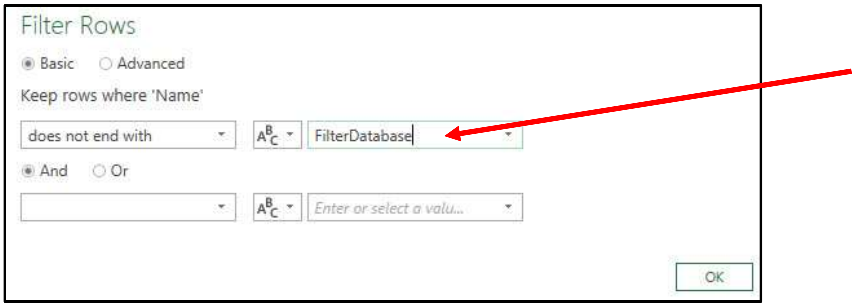


	Content	Name
1	Table	XanSalesTable
2	Table	BoSalesTable
3	Table	QuinSalesTable
4	Table	FranSalesTable
5	Table	TyroneSalesTable
6	Error	Popi!_FilterDatabase
7	Table	Popi!Criteria
8	Table	Popi!Extract
9	Table	Popi!Print_Area
10	Table	ProductUniqueList

Column1
ProductUniqueList
Tri Fly

vii. Notice that for each Defined Name, Power Query imports the Defined Names with the generic Column Headers (Field Names) like “Column1”, “Column2” and so on. We can use this fact to filter out the column name “Column1” to remove all the Defined Names. However, if order to filter out the error for the Filtered Database (Row 6), we will have to use a “Does Not End With” Filter to remove that error.

7) **Filter out the error for the Filtered Database**. To filter out the error for the Filtered Database, we can click on the Filter dropdown arrow in the Name Column, then point to Text Filters, then click on “Does Not End With”. In the Filter Rows dialog box, enter the criteria: “FilterDatabase”. Remember, in Power Query, the case of the letters matters.



Filter Rows

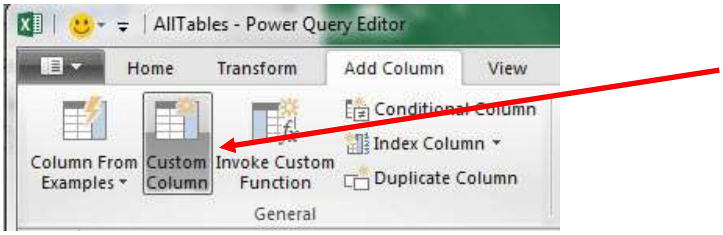
● Basic ○ Advanced

Keep rows where 'Name'

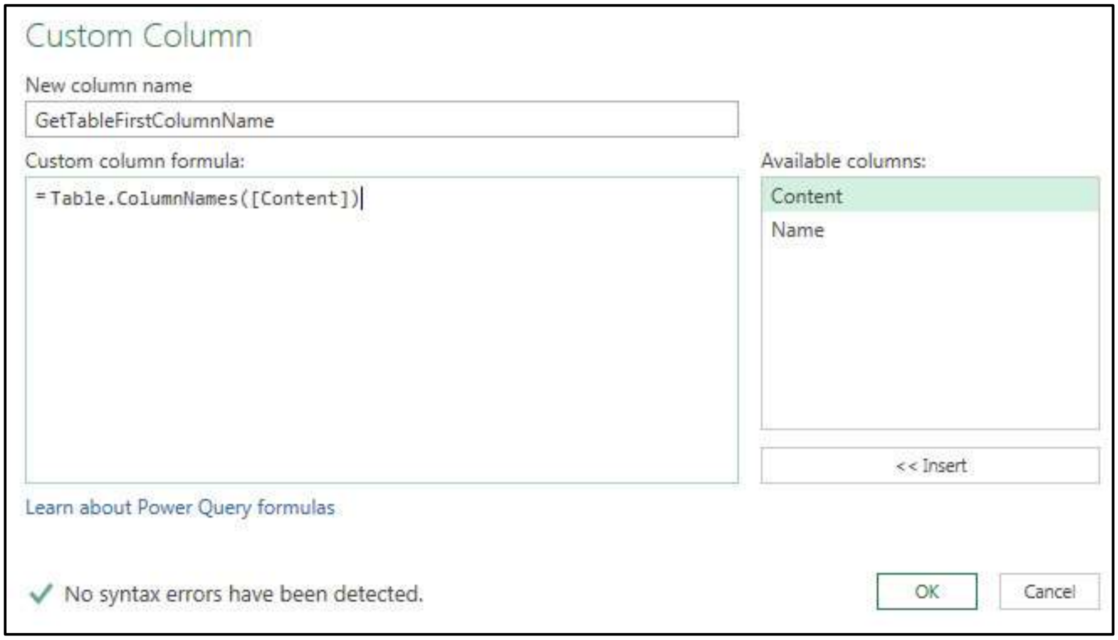
does not end with | FilterDatabase

● And ○ Or

8) **Create Custom Column to Extract Defined Name Generic Column Names**. To create a Custom Column to extract the first Column Name (Field Name) for each Table in each row, click the Custom Column button in the General group in the Add Column Ribbon Tab, as seen here:



9) Name the new column “GetTableFirstColumnName”. Then create the M Code Formula using the Power Query Function, Table.ColumnNames, as seen in the below picture. Then click OK.



Custom Column

New column name

GetTableFirstColumnName

Custom column formula:

```
=Table.ColumnNames([Content])
```

Available columns:

Content

Name

✓ No syntax errors have been detected.

10) **Aspects of Table.ColumnNames Power Query Function**:

   i. Table.ColumnNames can pull the Column Names / Field Names from a Table Object.

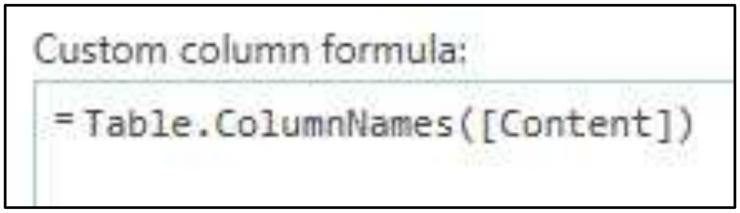
   ii. The Table.ColumnNames delivers a list of Column Names / Field Names as a List Object.

11) **Difference between a Table Object and a List Object**:

   i. Table = Set of records for a set of Columns or Fields.

   ii. List = Ordered Sequence of Values

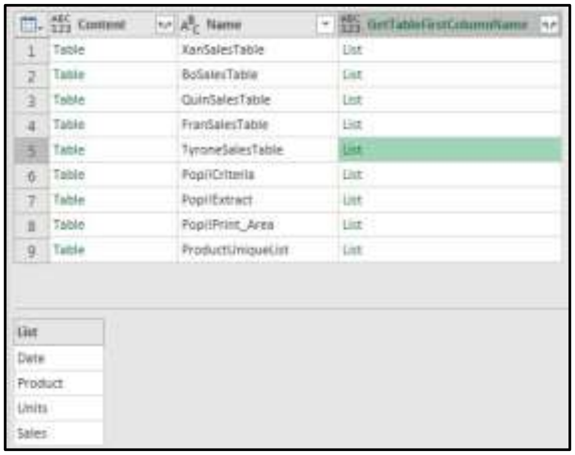
12) **Lookup Operator or Field Access Operator**. In the below formula we used in our Custom Column, the Square Brackets are called the Lookup Operator or Field Access Operator because it tells the formula to lookup the item in the Content column for each row as the formula is copied down the column. In our example, the Lookup Operator looks up a new Table Object for each row.



Custom column formula:

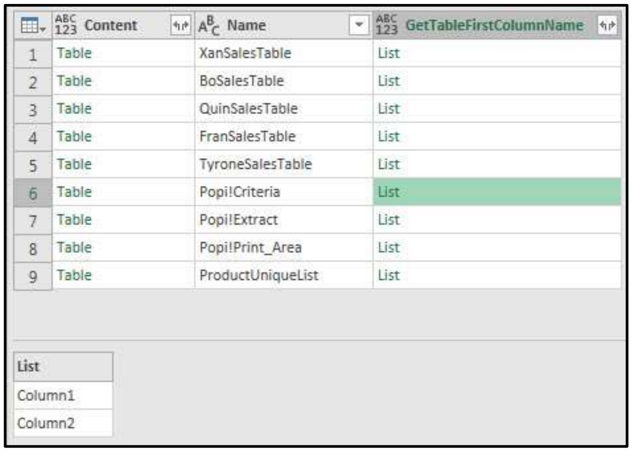
```
= Table.ColumnNames([Content])
```

13) **List Objects delivered to each row**. In the Below pictures you can see that the Table.ColumnNames function delivers a List Object to each row in the table. If you click of to the right side of the word List and look in the bottom part of the Query Editor, you can bee that each List in each row delivers the Column Names / Field Names.



	Content	Name	GetTableFirstColumnName
1	Table	XanSalesTable	List
2	Table	BoSalesTable	List
3	Table	QuinSalesTable	List
4	Table	FranSalesTable	List
5	Table	TyroneSalesTable	List
6	Table	Popi!Criteria	List
7	Table	Popi!Extract	List
8	Table	Popi!Print_Area	List
9	Table	ProductUniqueList	List

List
Date
Product
Units
Sales



	Content	Name	GetTableFirstColumnName
1	Table	XanSalesTable	List
2	Table	BoSalesTable	List
3	Table	QuinSalesTable	List
4	Table	FranSalesTable	List
5	Table	TyroneSalesTable	List
6	Table	Popi!Criteria	List
7	Table	Popi!Extract	List
8	Table	Popi!Print_Area	List
9	Table	ProductUniqueList	List

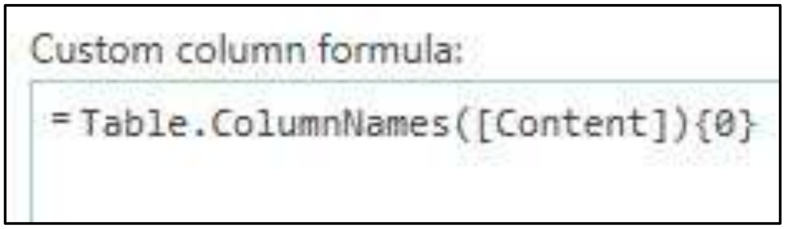
List
Column1
Column2

14) **Positional Index Operator to Extract First Column Name**. In order to extract just the first column name for each table, we need to edit our formula and use a Positional Index Operator. To edit the Custom Column, double-click the “Added Custom” step in the Applied Steps list. Then edit the formula as seen in the below picture.

i. The Positional Index Operator consists of the position of the item in the list that you want to extract, surrounded by open and close curly brackets such as { and }.

ii. Power Query is base zero, which means the first item in a list is zero, 0; the second item in a list is one, 1; the third item in a list is two, 2; and so on.

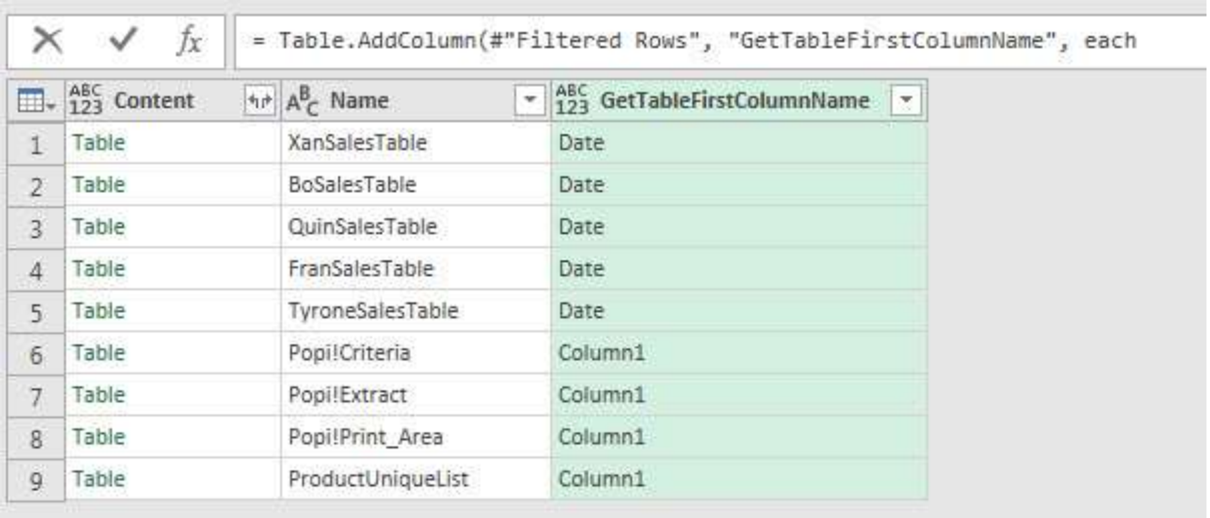
iii. For Excel people you can think of this process similar to the way the MATCH Function can look up the relative position of an item in a list. MATCH of course would deliver a one, 1, for the first item in a list, whereas, in Power Query, zero, 0, represent the first item in a list.



Custom column formula:

```
= Table.ColumnNames([Content]){0}
```

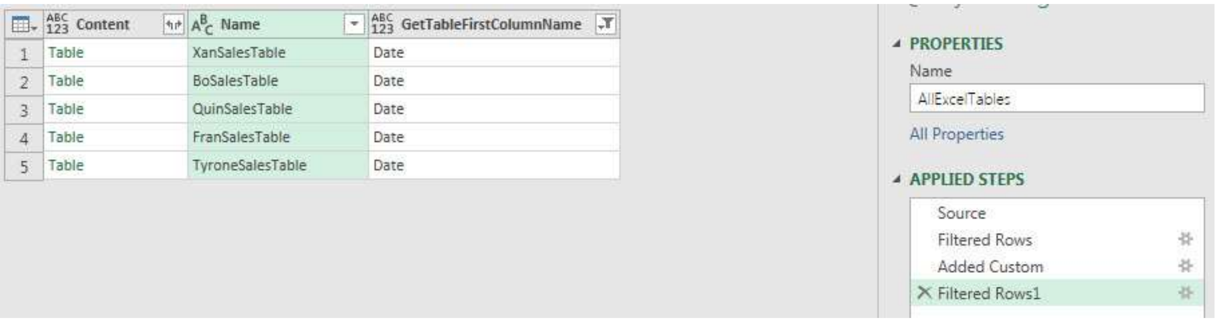
15) The result is just what we want: the first column name for each table, as seen in this picture:



```
= Table.AddColumn(#"Filtered Rows", "GetTableFirstColumnName", each
```

	Content	Name	GetTableFirstColumnName
1	Table	XanSalesTable	Date
2	Table	BoSalesTable	Date
3	Table	QuinSalesTable	Date
4	Table	FranSalesTable	Date
5	Table	TyroneSalesTable	Date
6	Table	Popi!Criteria	Column1
7	Table	Popi!Extract	Column1
8	Table	Popi!Print_Area	Column1
9	Table	ProductUniqueList	Column1

16) **Filter Out Column1 Field Name**. Now we can use the Filter dropdown to perform a Text Filter, Does Not Equal "Column1", and we will be left with just rows that contain Excel Tables, as seen here:



	Content	Name	GetTableFirstColumnName
1	Table	XanSalesTable	Date
2	Table	BoSalesTable	Date
3	Table	QuinSalesTable	Date
4	Table	FranSalesTable	Date
5	Table	TyroneSalesTable	Date

PROPERTIES

Name: AllExcelTables

All Properties

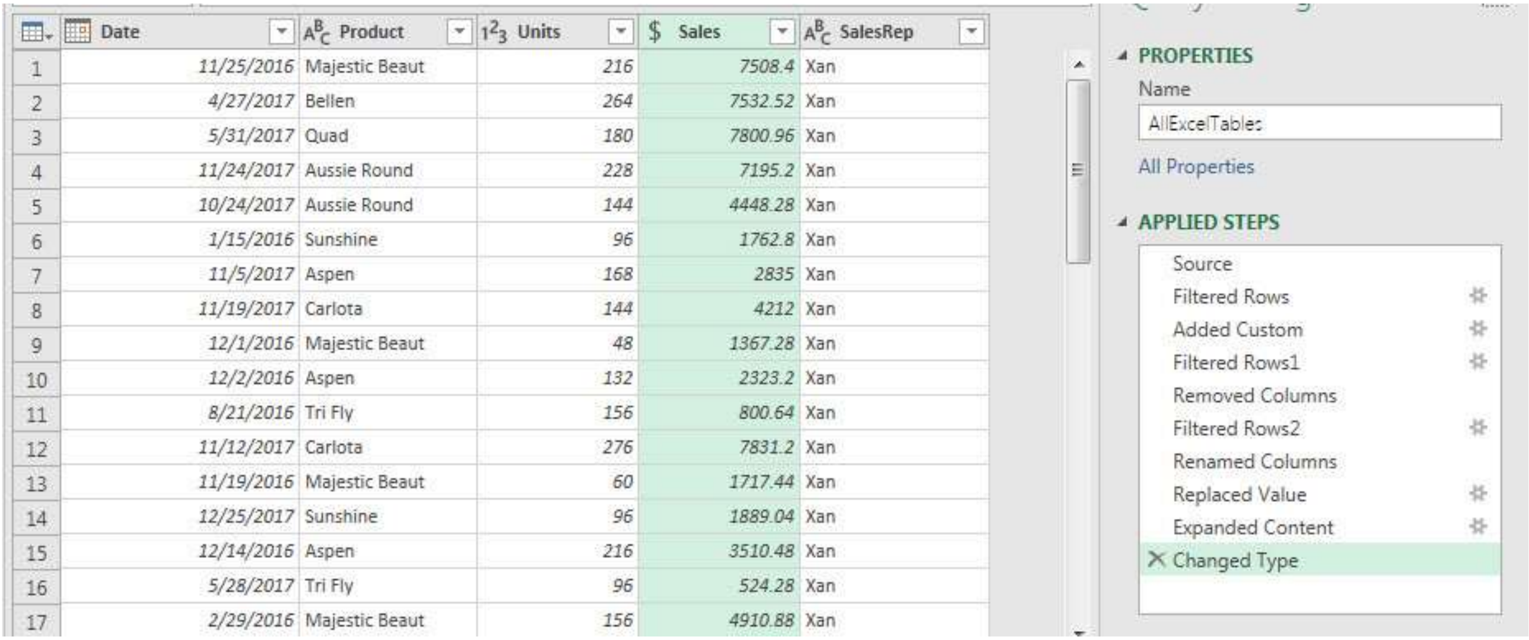
APPLIED STEPS

- Source
- Filtered Rows
- Added Custom
- Filtered Rows1

<u>**Next steps are as follows (no pictures):**</u>

17) **Remove Custom Column**.
18) **Filter Out Query Name in the Name Column (prevent Recursion Problem)**.
19) **Rename Name Column to "SalesRep"**.
20) **Use Replace feature to extract the SalesRep name from the Excel Table Name**.
21) **Expand Columns and Change Data Types**.

22) Here is what the final Query looks like in the Power Query Editor:



	Date	Product	Units	Sales	SalesRep
1	11/25/2016	Majestic Beaut	216	7508.4	Xan
2	4/27/2017	Bellen	264	7532.52	Xan
3	5/31/2017	Quad	180	7800.96	Xan
4	11/24/2017	Aussie Round	228	7195.2	Xan
5	10/24/2017	Aussie Round	144	4448.28	Xan
6	1/15/2016	Sunshine	96	1762.8	Xan
7	11/5/2017	Aspen	168	2835	Xan
8	11/19/2017	Carlota	144	4212	Xan
9	12/1/2016	Majestic Beaut	48	1367.28	Xan
10	12/2/2016	Aspen	132	2323.2	Xan
11	8/21/2016	Tri Fly	156	800.64	Xan
12	11/12/2017	Carlota	276	7831.2	Xan
13	11/19/2016	Majestic Beaut	60	1717.44	Xan
14	12/25/2017	Sunshine	96	1889.04	Xan
15	12/14/2016	Aspen	216	3510.48	Xan
16	5/28/2017	Tri Fly	96	524.28	Xan
17	2/29/2016	Majestic Beaut	156	4910.88	Xan

PROPERTIES

Name

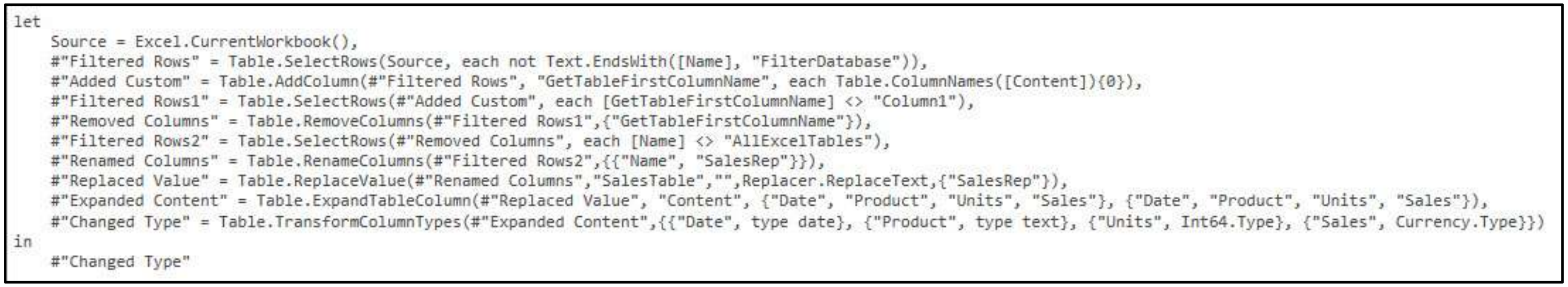
AllExcelTables

All Properties

APPLIED STEPS

- Source
- Filtered Rows
- Added Custom
- Filtered Rows1
- Removed Columns
- Filtered Rows2
- Renamed Columns
- Replaced Value
- Expanded Content
- Changed Type

23) Here is the Final M Code for Example #3:

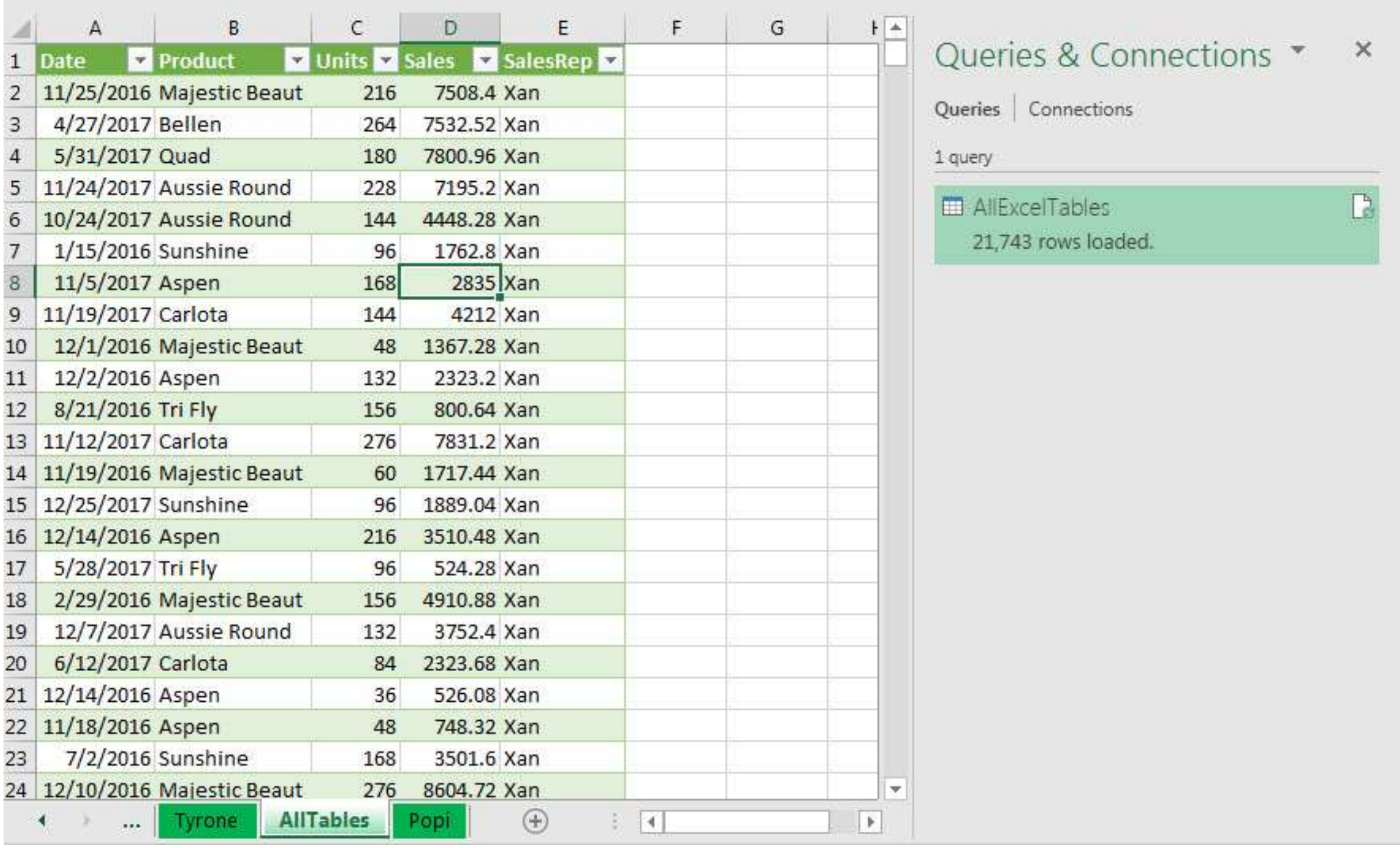


```
let
    Source = Excel.CurrentWorkbook(),
    #"Filtered Rows" = Table.SelectRows(Source, each not Text.EndsWith([Name], "FilterDatabase")),
    #"Added Custom" = Table.AddColumn(#"Filtered Rows", "GetTableFirstColumnName", each Table.ColumnNames([Content]){0}),
    #"Filtered Rows1" = Table.SelectRows(#"Added Custom", each [GetTableFirstColumnName] <> "Column1"),
    #"Removed Columns" = Table.RemoveColumns(#"Filtered Rows1",{"GetTableFirstColumnName"}),
    #"Filtered Rows2" = Table.SelectRows(#"Removed Columns", each [Name] <> "AllExcelTables"),
    #"Renamed Columns" = Table.RenameColumns(#"Filtered Rows2",{{"Name", "SalesRep"}}),
    #"Replaced Value" = Table.ReplaceValue(#"Renamed Columns","SalesTable","",Replacer.ReplaceText,{"SalesRep"}),
    #"Expanded Content" = Table.ExpandTableColumn(#"Replaced Value", "Content", {"Date", "Product", "Units", "Sales"}, {"Date", "Product", "Units", "Sales"}),
    #"Changed Type" = Table.TransformColumnTypes(#"Expanded Content",{{"Date", type date}, {"Product", type text}, {"Units", Int64.Type}, {"Sales", Currency.Type}})
in
    #"Changed Type"
```

24) **Closes and Load To Worksheet**.

25) **Add new Excel Table and Refresh**.

26) Here is what the final Query Load as an Excel Table to the Excel Worksheet looks like:



	Date	Product	Units	Sales	SalesRep
2	11/25/2016	Majestic Beaut	216	7508.4	Xan
3	4/27/2017	Bellen	264	7532.52	Xan
4	5/31/2017	Quad	180	7800.96	Xan
5	11/24/2017	Aussie Round	228	7195.2	Xan
6	10/24/2017	Aussie Round	144	4448.28	Xan
7	1/15/2016	Sunshine	96	1762.8	Xan
8	11/5/2017	Aspen	168	2835	Xan
9	11/19/2017	Carlota	144	4212	Xan
10	12/1/2016	Majestic Beaut	48	1367.28	Xan
11	12/2/2016	Aspen	132	2323.2	Xan
12	8/21/2016	Tri Fly	156	800.64	Xan
13	11/12/2017	Carlota	276	7831.2	Xan
14	11/19/2016	Majestic Beaut	60	1717.44	Xan
15	12/25/2017	Sunshine	96	1889.04	Xan
16	12/14/2016	Aspen	216	3510.48	Xan
17	5/28/2017	Tri Fly	96	524.28	Xan
18	2/29/2016	Majestic Beaut	156	4910.88	Xan
19	12/7/2017	Aussie Round	132	3752.4	Xan
20	6/12/2017	Carlota	84	2323.68	Xan
21	12/14/2016	Aspen	36	526.08	Xan
22	11/18/2016	Aspen	48	748.32	Xan
23	7/2/2016	Sunshine	168	3501.6	Xan
24	12/10/2016	Majestic Beaut	276	8604.72	Xan

Queries & Connections

Queries | Connections

1 query

AllExcelTables
21,743 rows loaded.

Important Keyboards Seen in this Video:

1. Create a Blank Query = **Alt, A, P, N, O, Q**