Power Query #6 # #6 Part 2

From Folder feature and the Combine Files option

Table of Contents

Project Goal for Video 06 2

Steps in CSV File Project: 3

1) Download files 3

2) All Tables Must Have the Same Structure and Schema Before Appending Tables 3

3) Excel From Folder feature 4

4) Power BI Desktop From Folder feature 4

5) M Code Binary Values: Files in a Column 5

6) Filter to Only Import CSV File Types 5

7) Remove Other Columns and Keep Only the Content Column 5

7) Combine Files button 6

8) Delimiter 6

9) Helper Queries and New Query Steps 7

10) Invoke Custom Function 8

11) Finished Appended Table 8

Excel File Project All Worksheets With Same Name: 9

1) Steps are the same as with the Csv file project, except: 9

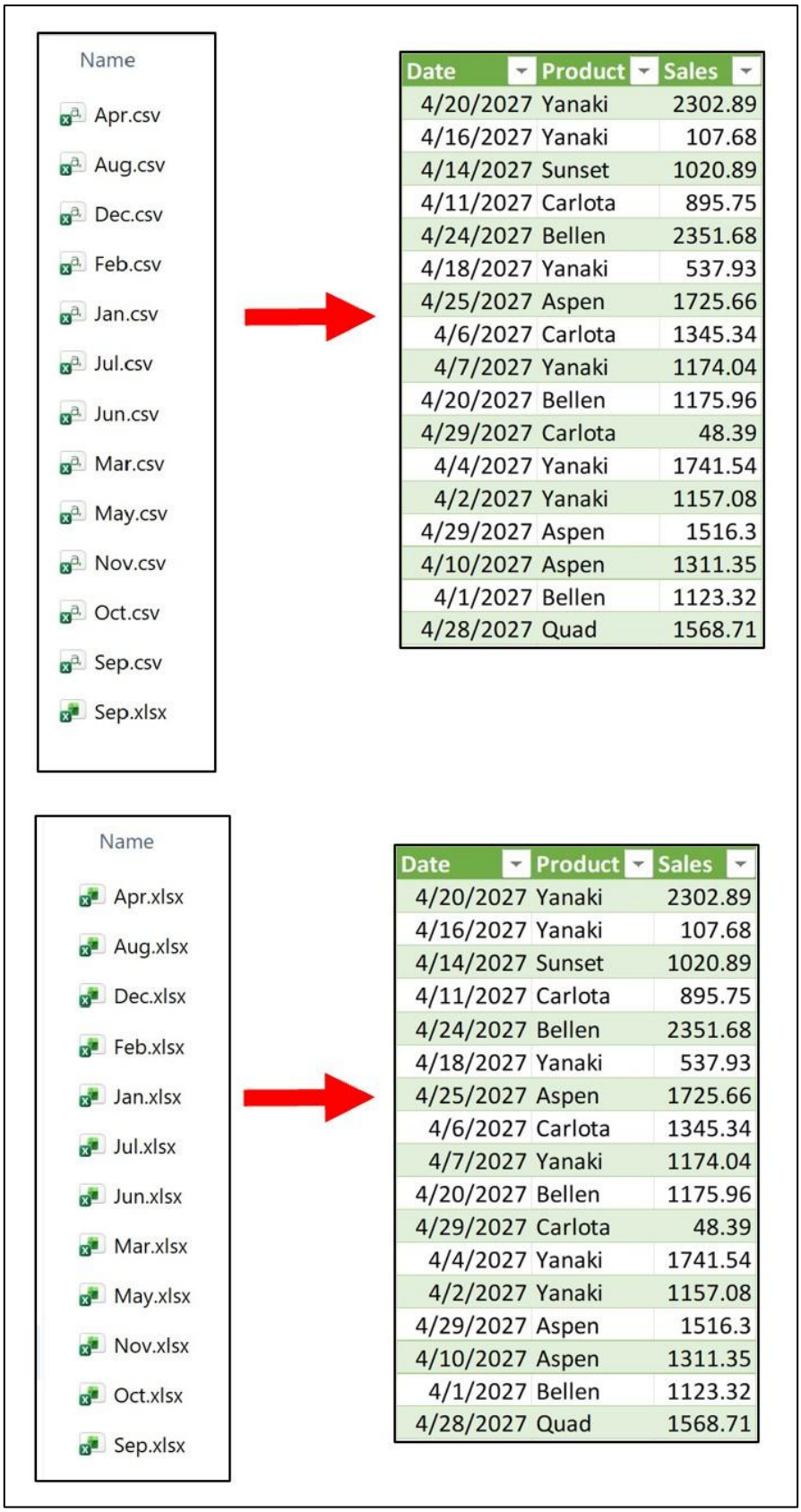
Excel File Project Worksheets With Different Names: 11

1) Steps are the same as with the Excel video #6 file project, except: 11

The difference between Excel example 1 (video #6) and Excel example 2 (video #6 Part 2): 13

Project Goal for Video 06

We will use the From Folder feature and the Combine Files option to automatically append 12 files into a single proper data table. We will see how to do it with 12 csv files and 12 Excel files.



Steps in CSV File Project:

1) Download files

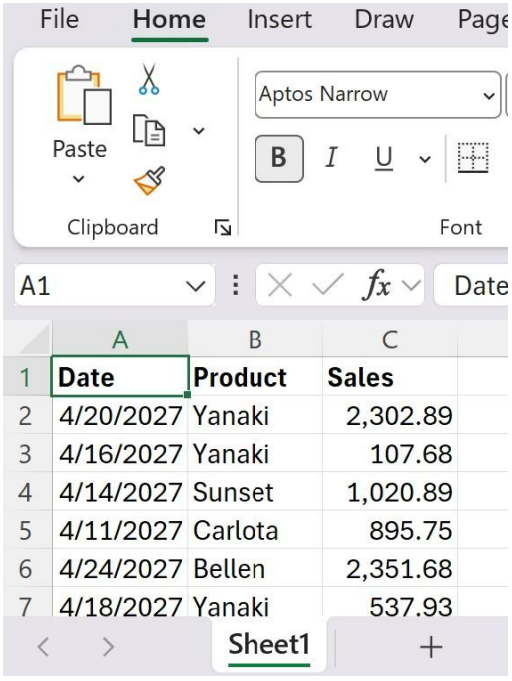
- As we learned in the last video, download the zipped folder below Video #4, then unzip it in the same location you used for the previous video.

2) All Tables Must Have the Same Structure and Schema Before Appending Tables

- The 12 csv files that we have each contain a single table with the field names Date, Product, and Sales. The structure and schema are the same for each table so we can append the 12 tables into one table. A picture of one of the csv files is shown below:

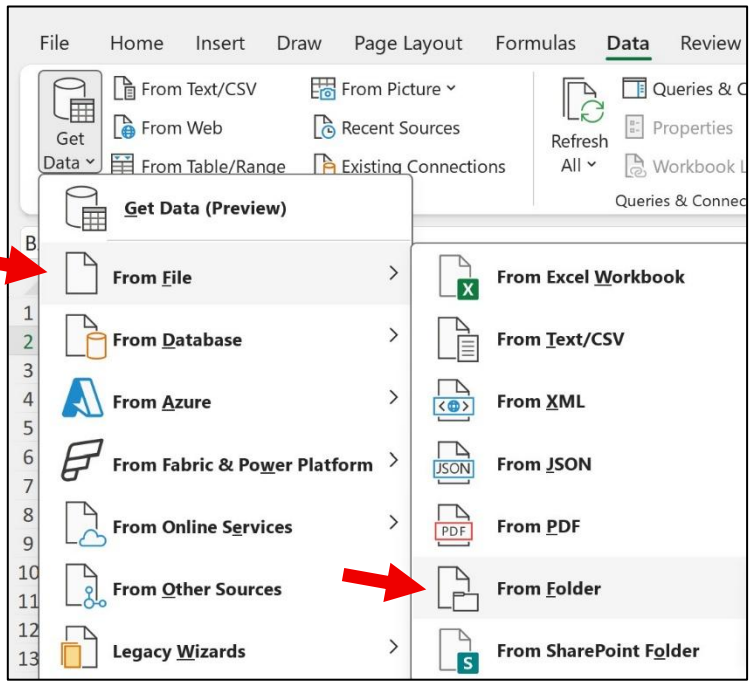
```
Date,Product,Sales
4/20/2027,Yanaki,"2,302.89"
4/16/2027,Yanaki,107.68
4/14/2027,Sunset,"1,020.89"
4/11/2027,Carlota,895.75
4/24/2027,Bellen,"2,351.68"
4/18/2027,Yanaki,537.93
4/25/2027,Aspen,"1,725.66"
4/6/2027,Carlota,"1,345.34"
4/7/2027,Yanaki,"1,174.04"
4/20/2027,Bellen,"1,175.96"
```

- The 12 Excel files that we have each contain a single worksheet named “Sheet1” and hold a single table with the field names Date, Product, and Sales. The structure and schema are the same for each table so we can append the 12 tables into one table. A picture of one of the Excel files is shown below:



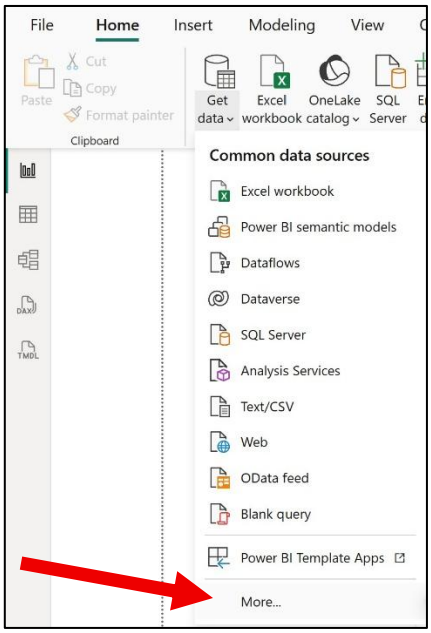
3) Excel From Folder feature

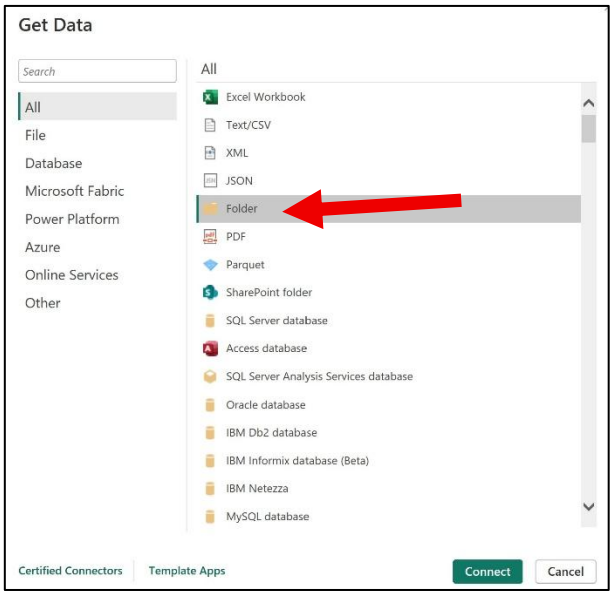
- To import files from within a folder, in the Excel Ribbon, go to the Data tab, Get & Transform group, Get Data dropdown arrow and select the From Folder option.



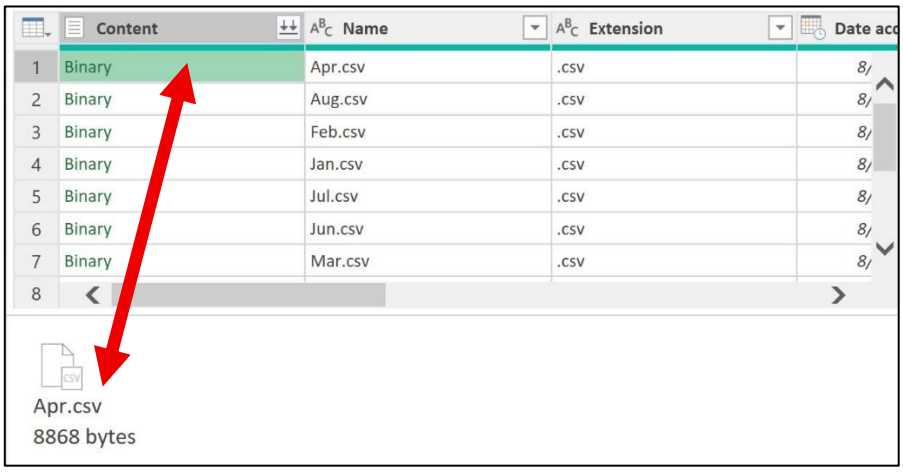
4) Power BI Desktop From Folder feature

- To import files from within a folder, in the Power BI Desktop Ribbon, go to the Data tab, Get & Transform group, then click More. Then in the Get Data dialog box, on the left select All, then on the right select Folder.

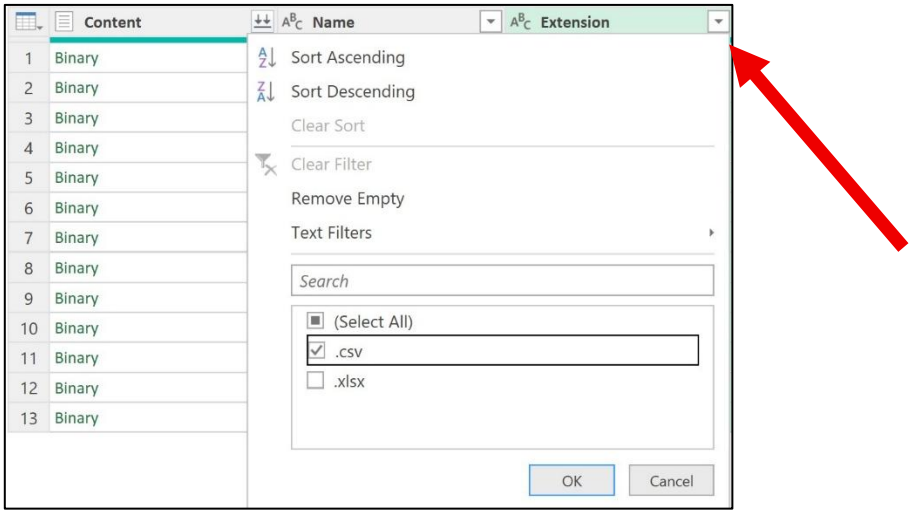




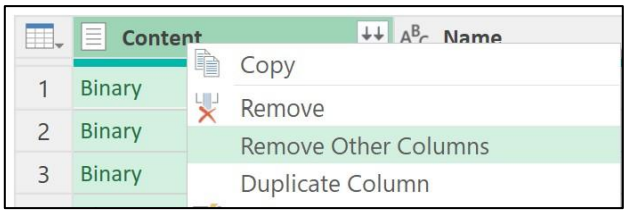
5) M Code Binary Values: Files in a Column

- An M Code Binary Value represents a sequence of bytes, such as a CSV file, a JPG file or an Excel file. Binary M Code Values allow us to work with files in the Power Query Editor. The picture below shows a column named Content which folds an M Code Binary CSV file in each row. This ability of Power Query to use the M Code Binary Value type to hold data objects like CSV files, Excel files or Tables in a column is unique to Power Query and it is not found in the Excel worksheet or in the Power Pivot or Power BI Desktop Data Model.
- 

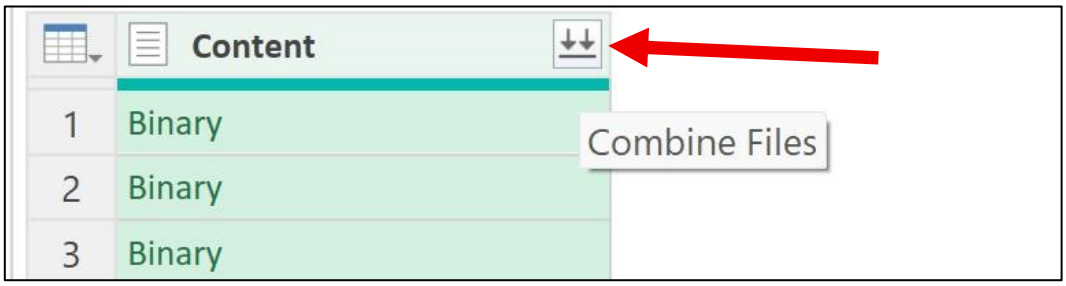
6) Filter to Only Import CSV File Types

- Use the Extension column filter dropdown arrow to filter to include only ".csv" extensions. This is protection for your query so that if there are other file types in the folder, they will not be imported. A picture is shown below:
- 

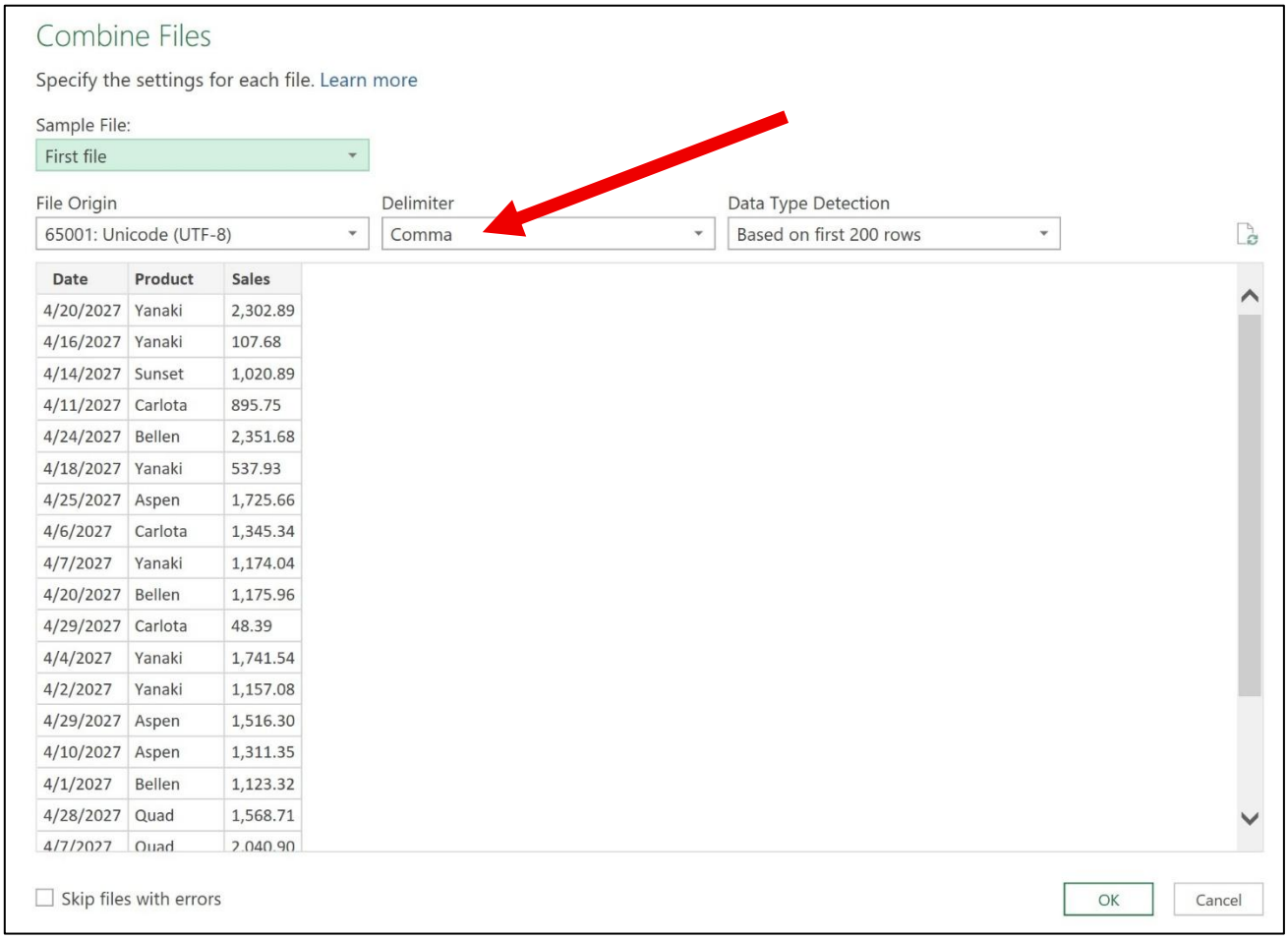
7) Remove Other Columns and Keep Only the Content Column

- To remove all columns and keep only the Content columns, as shown in the picture below, right-click the Content column and click the Remove Other Columns option.
- 

7) Combine Files button

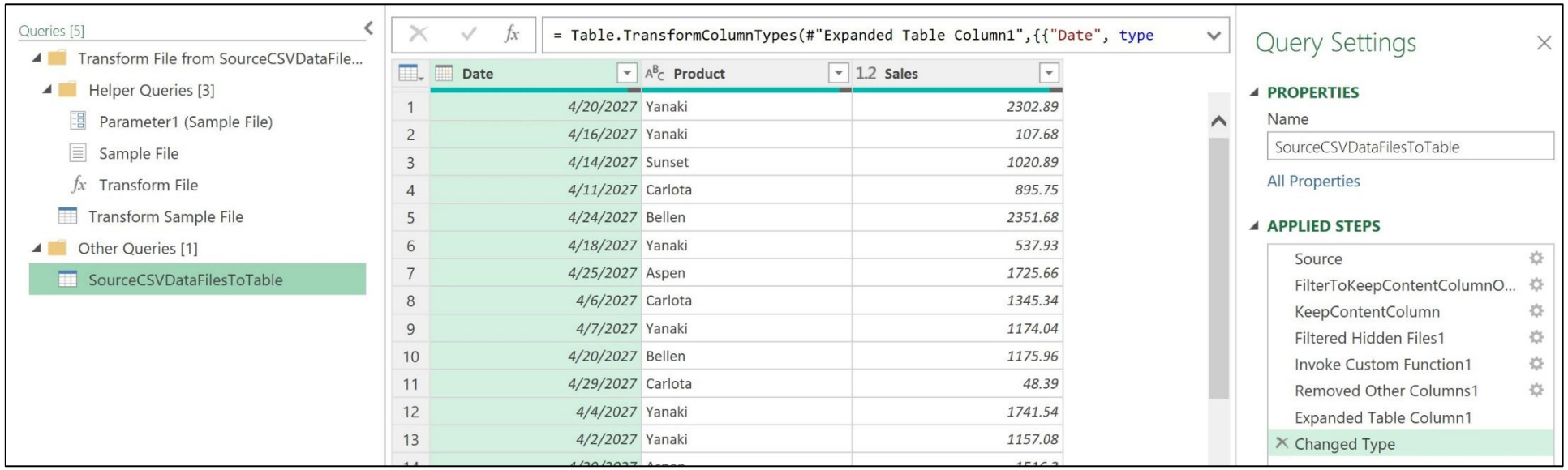
- As shown in the picture below , on the left side of a Binary file Content column, the Combine Files button appears. You can use this button to append tables that have the same structure and schema.
- 

8) Delimiter

- As shown in the picture below , In the Combine Files dialog box, make sure that the delimiter is a comma and then click the OK button..
- 

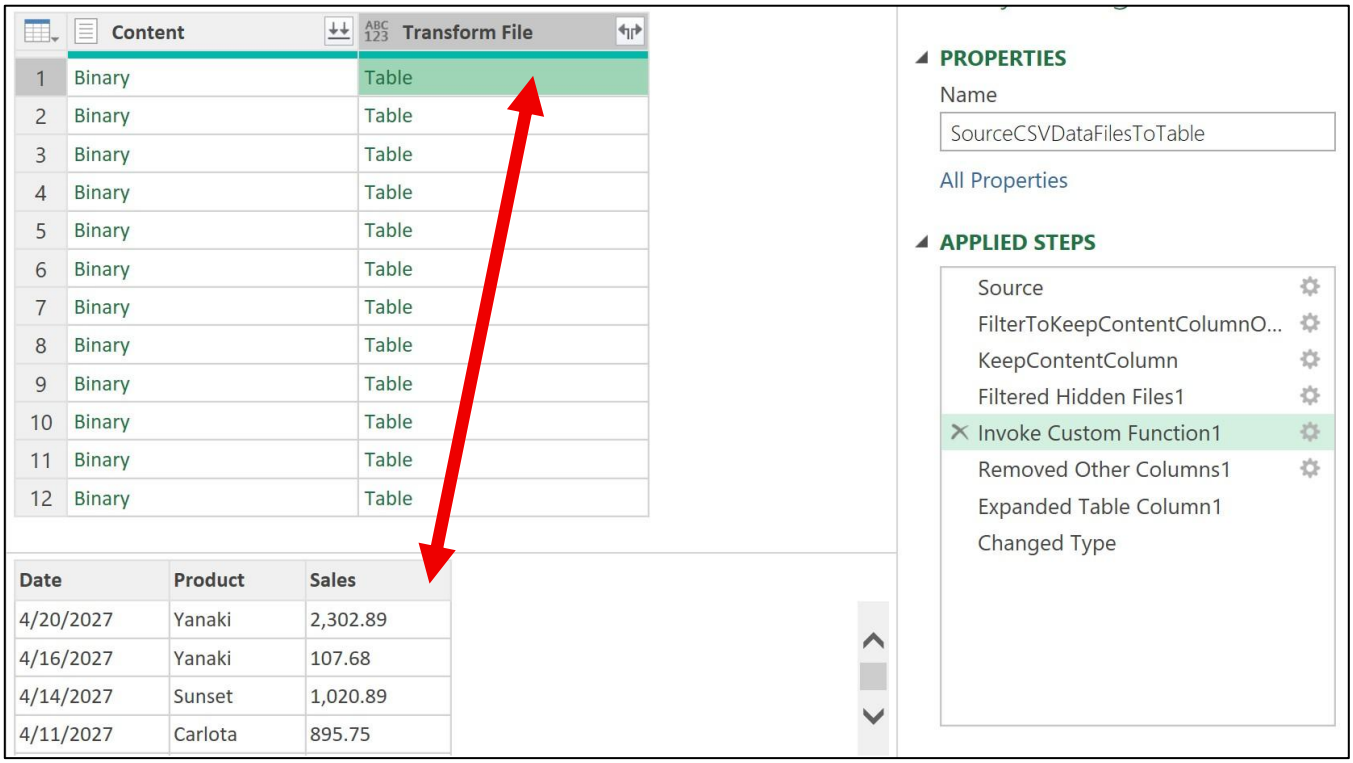
9) Helper Queries and New Query Steps

- As shown below, when you click the Combine Files button, Power Query creates four helper queries in the Queries pane on the left, adds five new steps to the Applied Steps list, and displays the new appended table in the center. In the Queries pane on the left you can see the custom function named “fx Transform File” is created. This function is used to transform the csv file in each row of the Content column to a proper table.



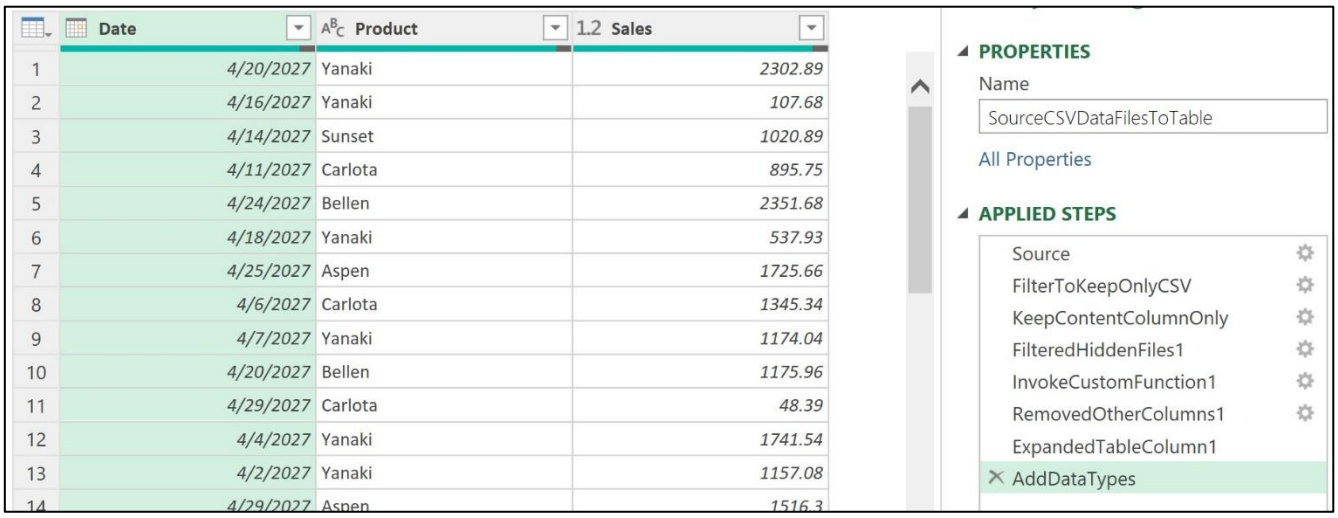
10) Invoke Custom Function

- As shown in the picture below, after selecting the query step named Invoke Custom Function 1 in the Applied Steps list, the custom function named “fx Transform File” is used to transform the csv file in each row of the Content column to a proper table..



11) Finished Appended Table

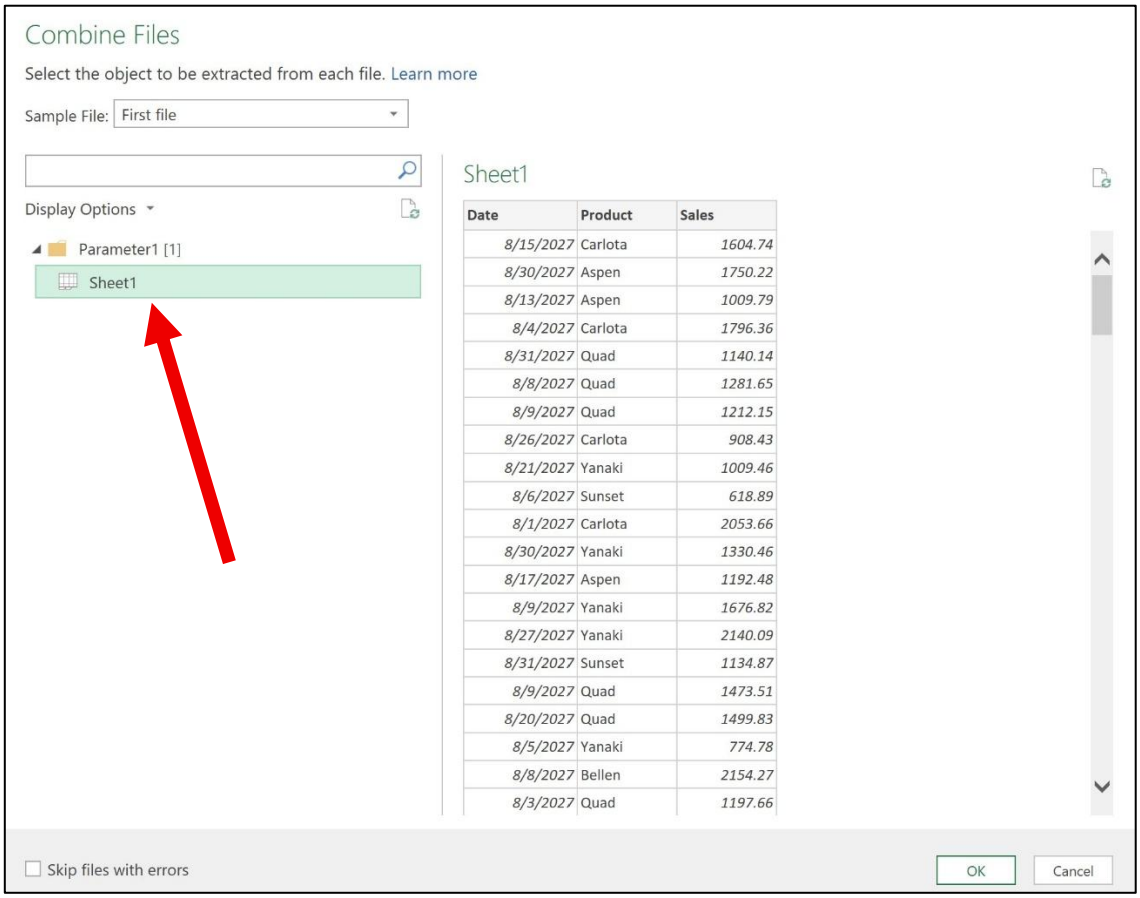
- The finished appended table is shown below.



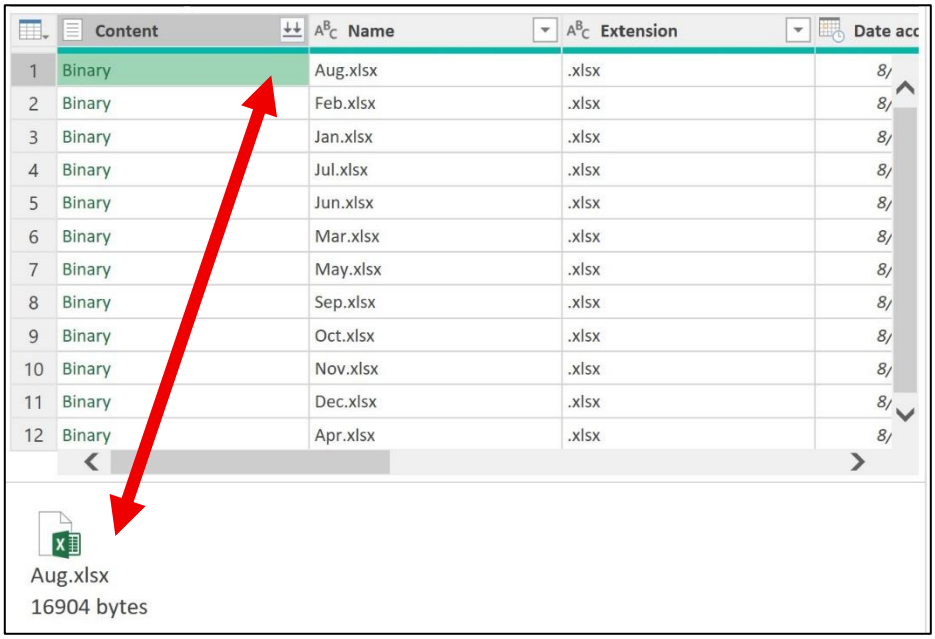
Excel File Project All Worksheets With Same Name:

1) Steps are the same as with the Csv file project, except:

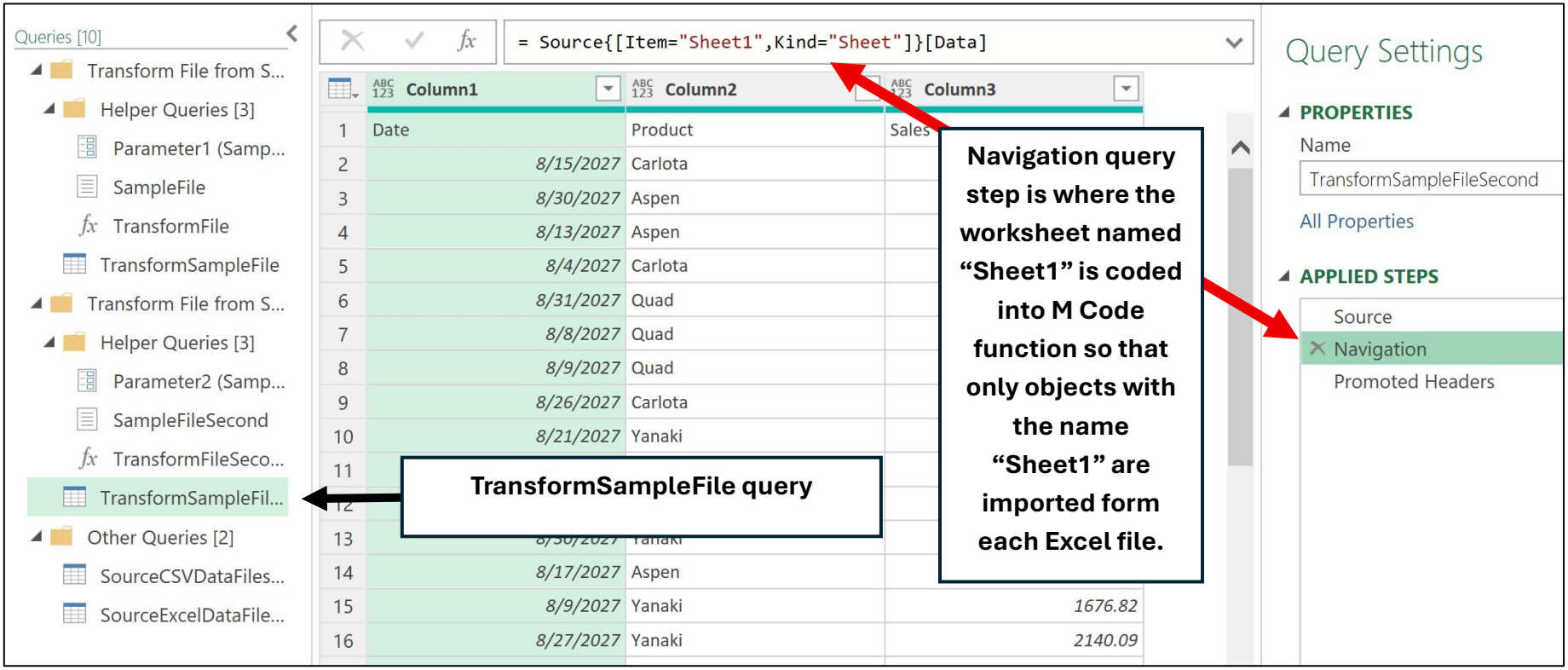
- When you use the From Folder feature and the Combine Files button with Excel files, you will see the Combine Files dialog box that asks you to select an object from the Excel file to use an example so the automatic process can build the correct custom function to convert the Excel file objects to tables and then append the table into a single table. As shown below, we select the object “Sheet1”.



- The Binary Content column contains Excel files, as shown in the picture below.



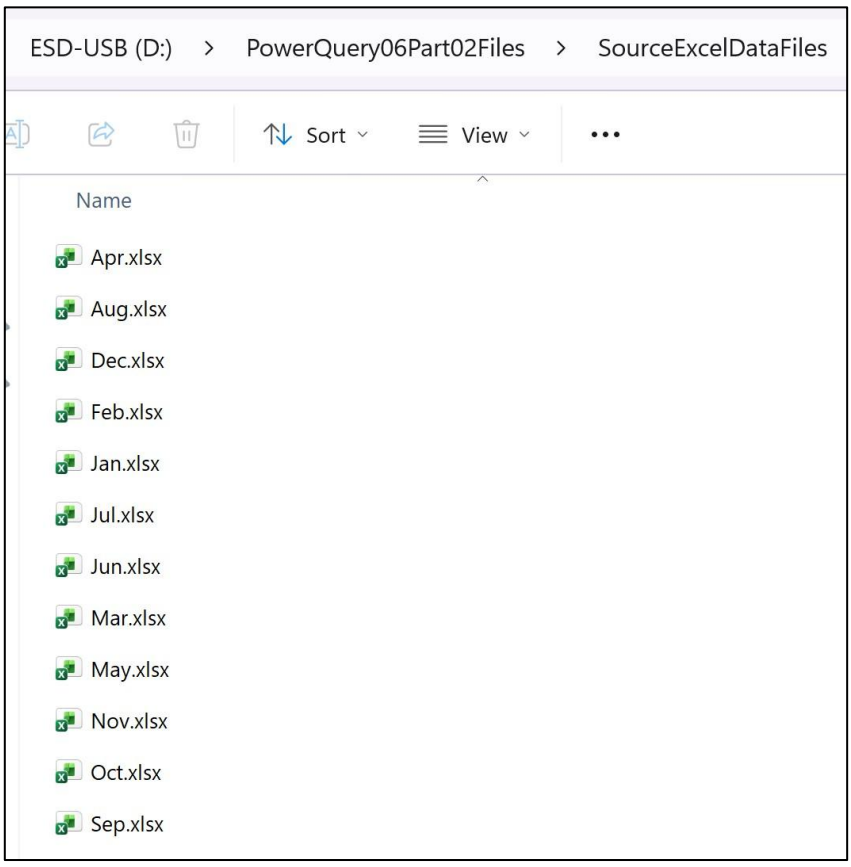
- As shown below, after we click the Combine Files button, in the TransformSampleFile query (this is the query where the M Code is created for the custom function), the Navigation step selects the worksheet named “Sheet1” from each workbook.



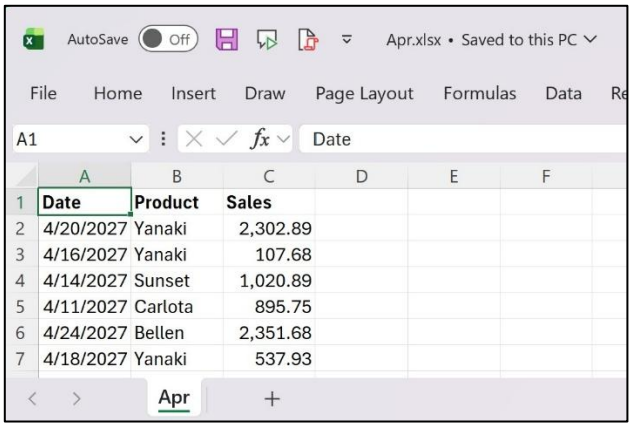
Excel File Project Worksheets With Different Names:

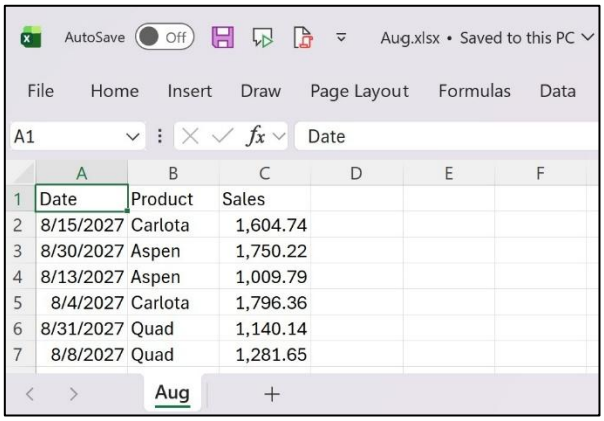
1) Steps are the same as with the Excel video #6 file project, except:

1. In video #6 Part 2, I show you how to import multiple workbooks, each with one worksheet, but each worksheet has a different name.
2. Here is the folder with the 12 files from the download zipped folder:

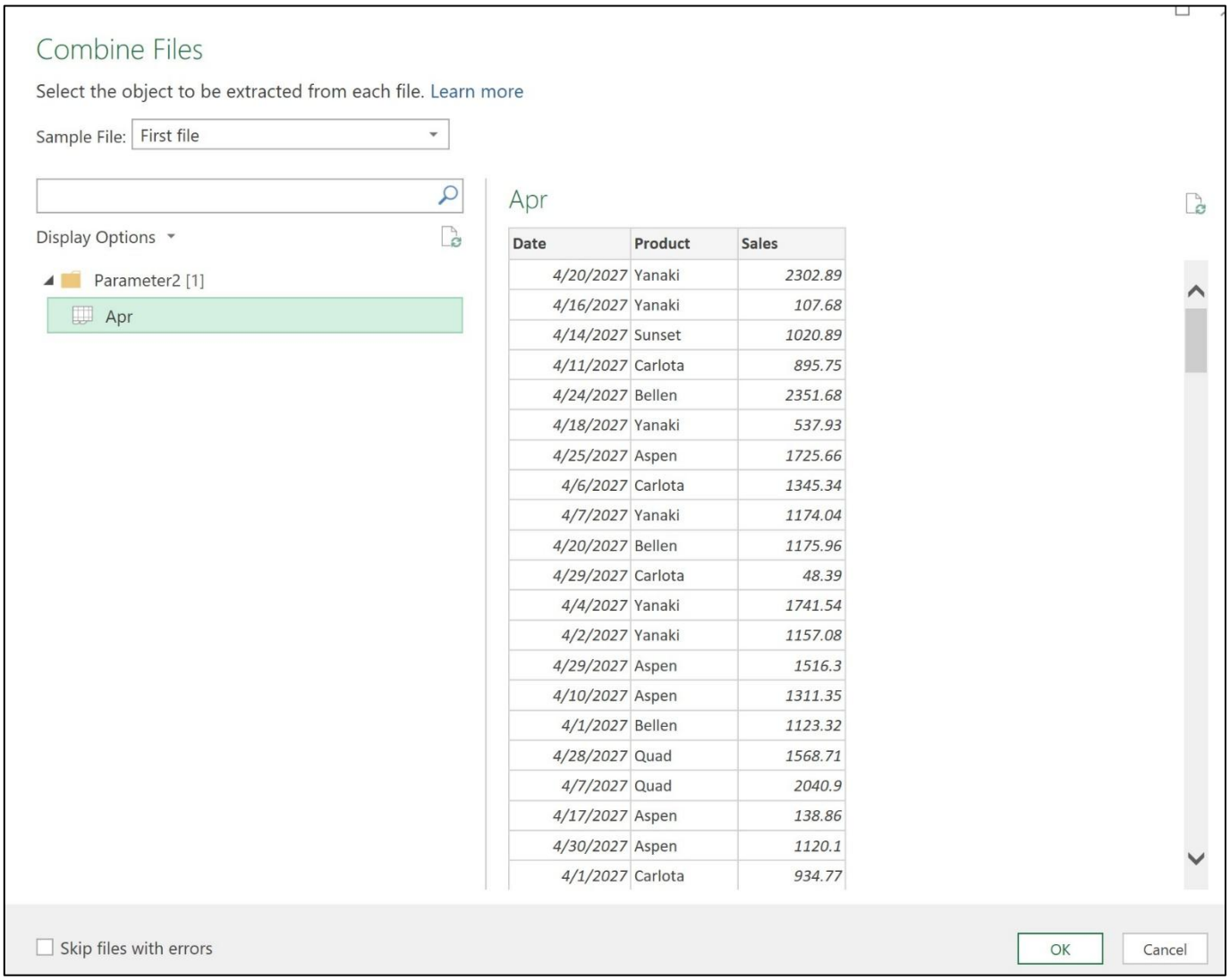


3. The files for this example (video #6, part 2) have the same file names as in video #6, but each one worksheet has a different worksheet name. For example, in the below picture, Apr and Aug files have the worksheet names Apr and Aug, respectively. However, whereas in the Excel example in video #6, you could have multiple worksheets in an Excel file, in this example (video #6 Part 2), each file can have only one worksheet.



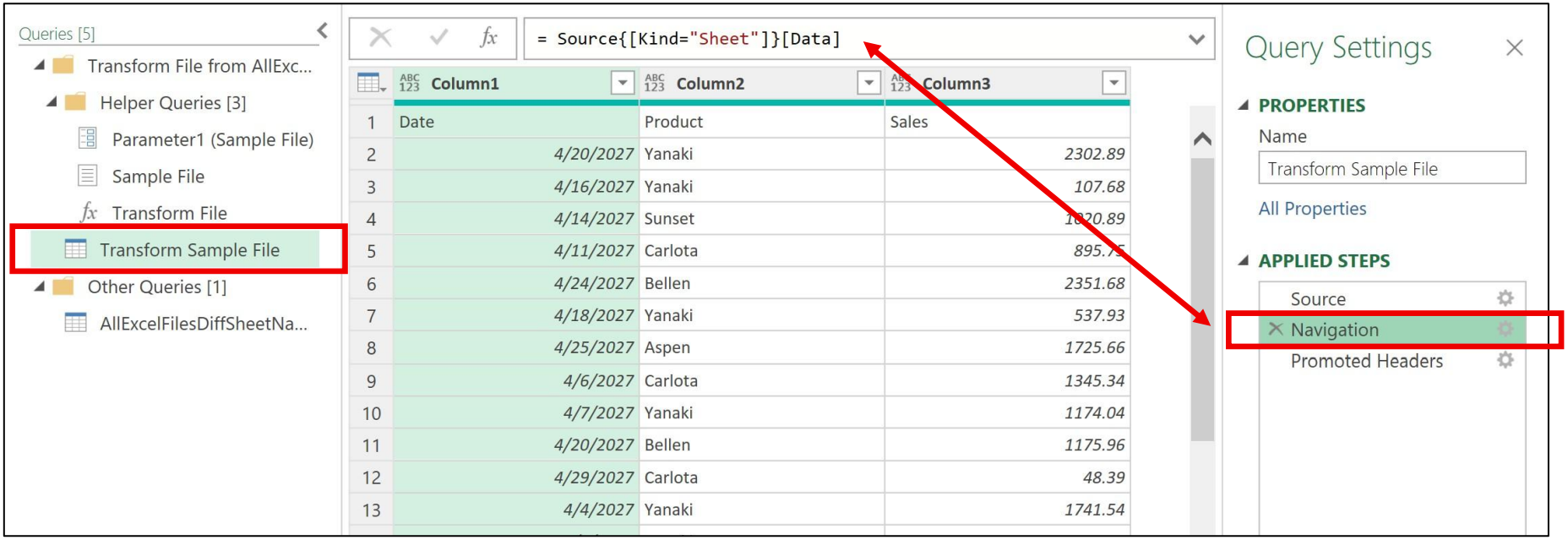


4. After you click the combine files button, in the Combine Files dialog box, you select “Apr” worksheet as the example for Power Query to build the code. This will not work, but you will change the code later in the Transform Sample File query in the Navigation step.



Date	Product	Sales
4/20/2027	Yanaki	2302.89
4/16/2027	Yanaki	107.68
4/14/2027	Sunset	1020.89
4/11/2027	Carlota	895.75
4/24/2027	Bellen	2351.68
4/18/2027	Yanaki	537.93
4/25/2027	Aspen	1725.66
4/6/2027	Carlota	1345.34
4/7/2027	Yanaki	1174.04
4/20/2027	Bellen	1175.96
4/29/2027	Carlota	48.39
4/4/2027	Yanaki	1741.54
4/2/2027	Yanaki	1157.08
4/29/2027	Aspen	1516.3
4/10/2027	Aspen	1311.35
4/1/2027	Bellen	1123.32
4/28/2027	Quad	1568.71
4/7/2027	Quad	2040.9
4/17/2027	Aspen	138.86
4/30/2027	Aspen	1120.1
4/1/2027	Carlota	934.77

5. As shown below, in the Transform Sample File query in the Navigation step, change the code from:
   = Source{[Item="Apr", Kind="Sheet"]}[Data]
   to:
   = Source{[Kind="Sheet"]}[Data]
   By changing the code you are allowing worksheets of any name to be imported, and this works as long as there is only one worksheet in each workbook.



The difference between Excel example 1 (video #6) and Excel example 2 (video #6 Part 2):

- **The M Code for the Excel example in video #6, = Source{[Item="Sheet1", Kind="Sheet"]}[Data]**, allows you to have multiple worksheets or other objects in the workbook because the Kind criteria force the table from the workbook to be a worksheet (thus eliminating other types of objects like Excel Tables and Defined Names) and the Item criteria forces the worksheet name to be "Sheet1" (thus eliminating other worksheets with different names).
- **The M Code for the Excel example in video #6 Part 2, = Source{[Kind="Sheet"]}[Data]**, requires that you only have one worksheet in the workbook because the row lookup criteria, {[Kind="Sheet"]} – not fully explained until video #9 when we learn about Key Match Lookup, yields an error if the is more than one worksheet, and because the Kind criteria only specifies "Sheet" other types of objects like Excel Tables and Defined Names are ignored.