

1.1. Exercise: Setting up an Excel data source

Introduction

By now, you know what the benefits of combining data sources are and how to set up a flat data source. In this exercise, you'll put these new skills into practice by setting up a Microsoft Excel data source for Power BI, connecting to an Excel file from Power BI, selecting and importing data and creating a report in Power BI.

Case Study

Jamie, an Adventure Works executive, has asked you to prepare a report for the Sales department, displaying the details of any required order, including the product, unit price, order quantity, and discount rate for the annual review meeting. Each country manager shared a Microsoft Excel workbook containing the required data. It is your job to connect to the data sources and then combine them all into a single report!

Resources

To accomplish this exercise, you will need to ensure you have access to the following resources:

- Power BI Desktop.
- Table to import: This data will be imported into Power BI as part of the exercise.

Instructions

Step 1: Download and prepare the file

The first step is to ensure that your Excel file is saved on your computer, or that you have access to it through OneDrive for Business, personal OneDrive accounts, or SharePoint Team Sites. Ensure your files meet the following requirements:

- The dataset is provided as a flat file: Your workbook must not have any total rows or columns.
- All data in each column should be of the same type: Whether it is dates, text or currency, each column must contain the same type of data.
- The file is in an easily readable table format. It does not contain pivot tables or matrix formats
- A space-free name: Give your table a name that is easy to remember. Just remember that the name cannot contain any spaces.

Step 2: Get data

Power BI Desktop provides the ability to connect with various data sources, ranging from basic sources such as Microsoft Excel files to more complex ones.

- Since your assigned task is to connect to the data sources required, let's connect to them. Choose the required data type from the list.
- Navigate to the location where your Excel file is saved.
- Choose the worksheet(s) and table(s) you want to import in the Navigator window.

Click Load. This will import your Excel data into Power BI.

- Tip: You can inspect the data in the Table View (also known as the Data View) by clicking the table icon on the left sidebar.
- Save your Power BI project.

Conclusion

Congratulations, you have set up an Excel data source in Power BI! Remember that while Excel is a great starting point, Power BI offers many more powerful data analysis and visualization tools.

Exemplar: Setting up an Excel data source

In the exercise Setting up an Excel Data Source, you had to set up a Microsoft Excel data source for Power BI, connect to an Excel file from Power BI, select and import data and create a report in Power BI.

This reading presents the process, accompanied by screenshots for easy comparison with your own copy.

Before we start, please take note of the following important information.

Power BI Desktop User Interface

Power BI Desktop is updated and released on a monthly basis, incorporating customer feedback and new features. You might experience changes in the Power BI Desktop User Interface (UI) that have taken place after the development of this training content. As a result, the screenshots in the videos, readings, or exercises might not align exactly with how you experience the UI. However, please note that these changes do not impact the functionalities of the UI. Hence, you will still be able to perform all the steps shown in that video, reading, or exercise.

Step 1: Understand the context

The first step is to ensure that your Excel file is saved on your computer, or that you have access to it through OneDrive for Business, personal OneDrive accounts, or SharePoint Team Sites. Ensure your files meet the following requirements:

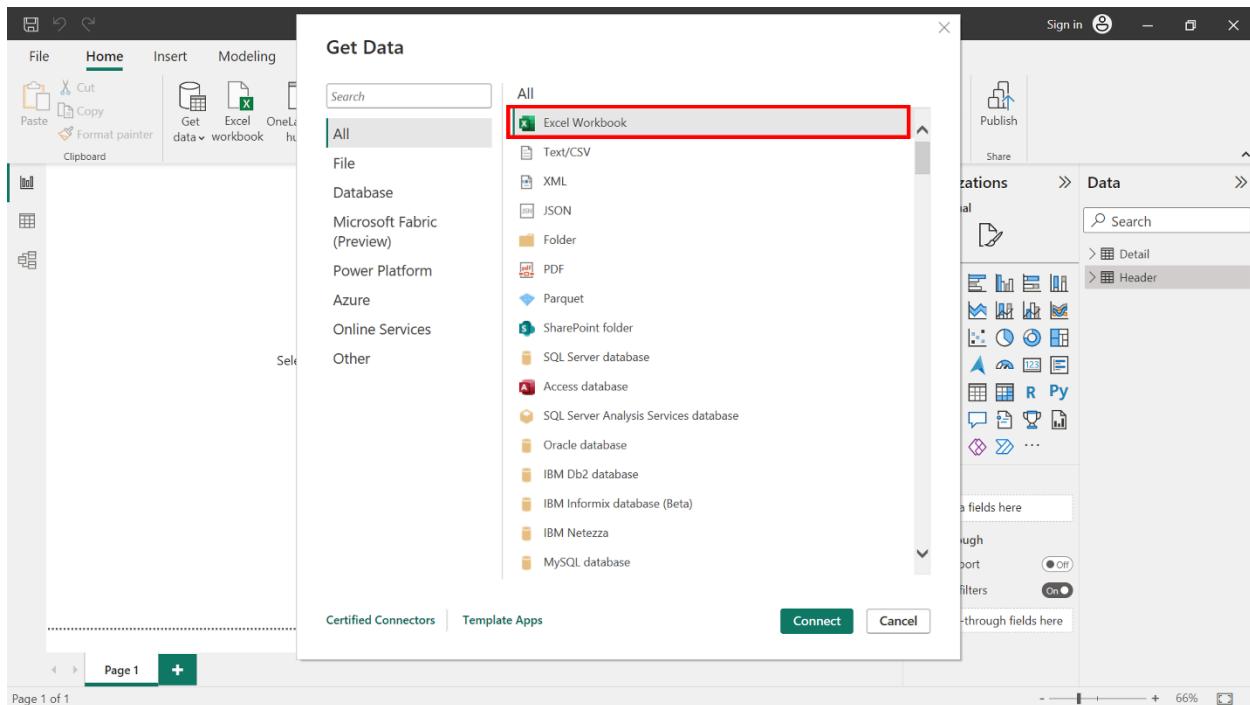
- The dataset is provided as a flat file: Your workbook must not have any total rows or columns.
- All data in each column should be of the same type: Whether it is dates, text or currency, each column must contain the same type of data.
- The file is in an easily readable table format. It does not contain pivot tables or matrix formats

- Easily readable table format: no pivot tables or matrix formats are accepted.
- A space-free name: Give your table a name that is easy to remember. Just remember that the name cannot contain any spaces.

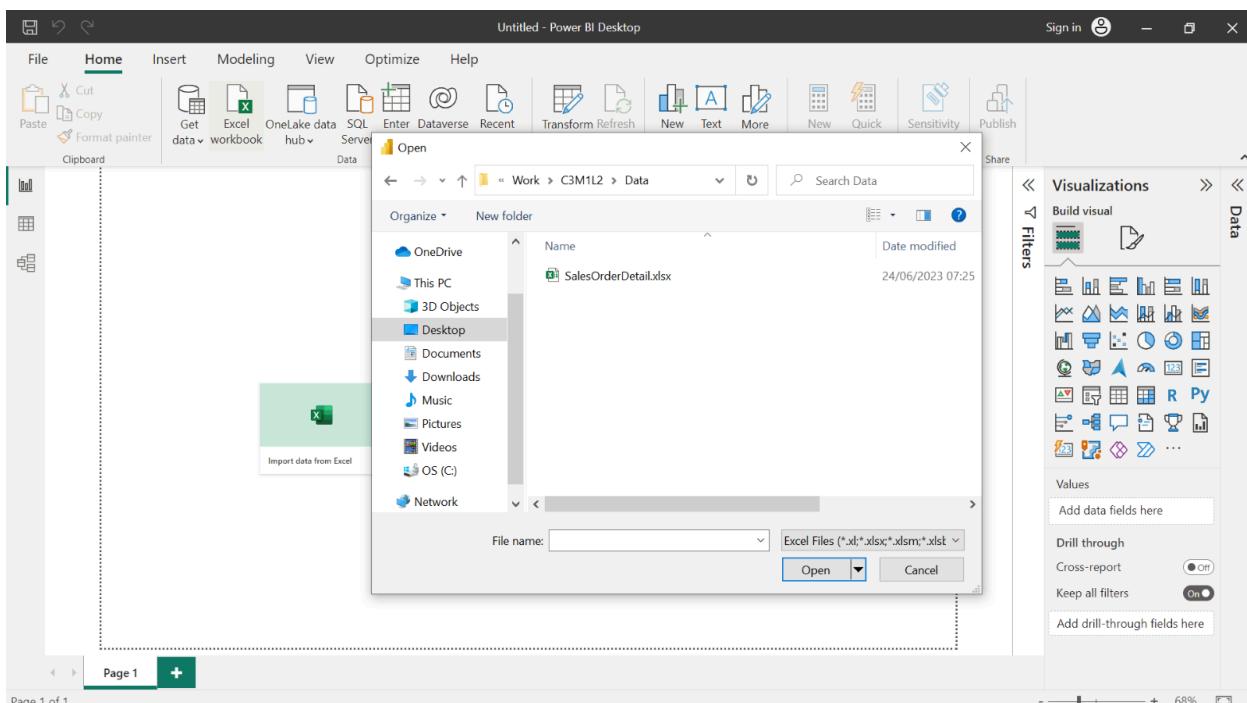
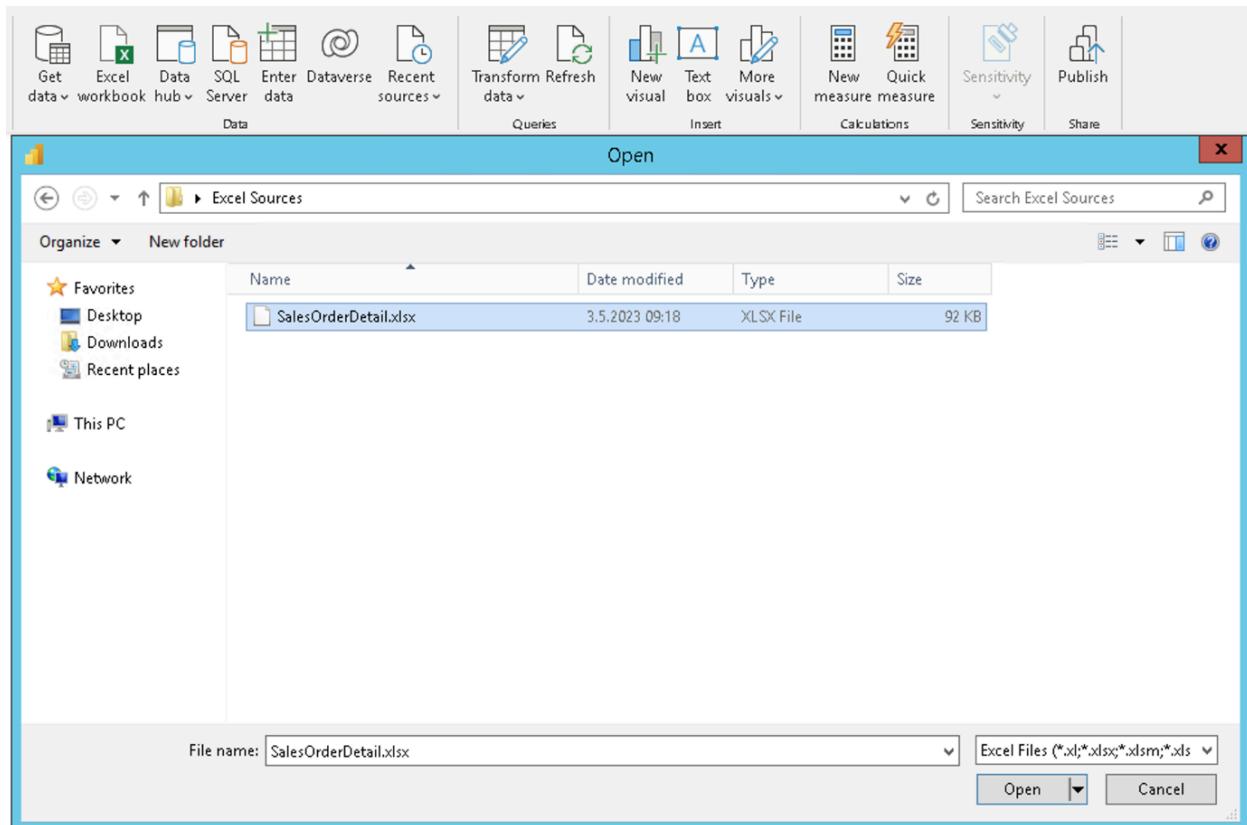
Step 2: Get data

Power BI Desktop provides the ability to connect with various data sources, ranging from basic sources such as Microsoft Excel files to more complex ones.

1. Once you have Power BI Desktop open, select the Get Data option on the Home tab. This will open the Get Data window where you can choose the data source you want to connect to.

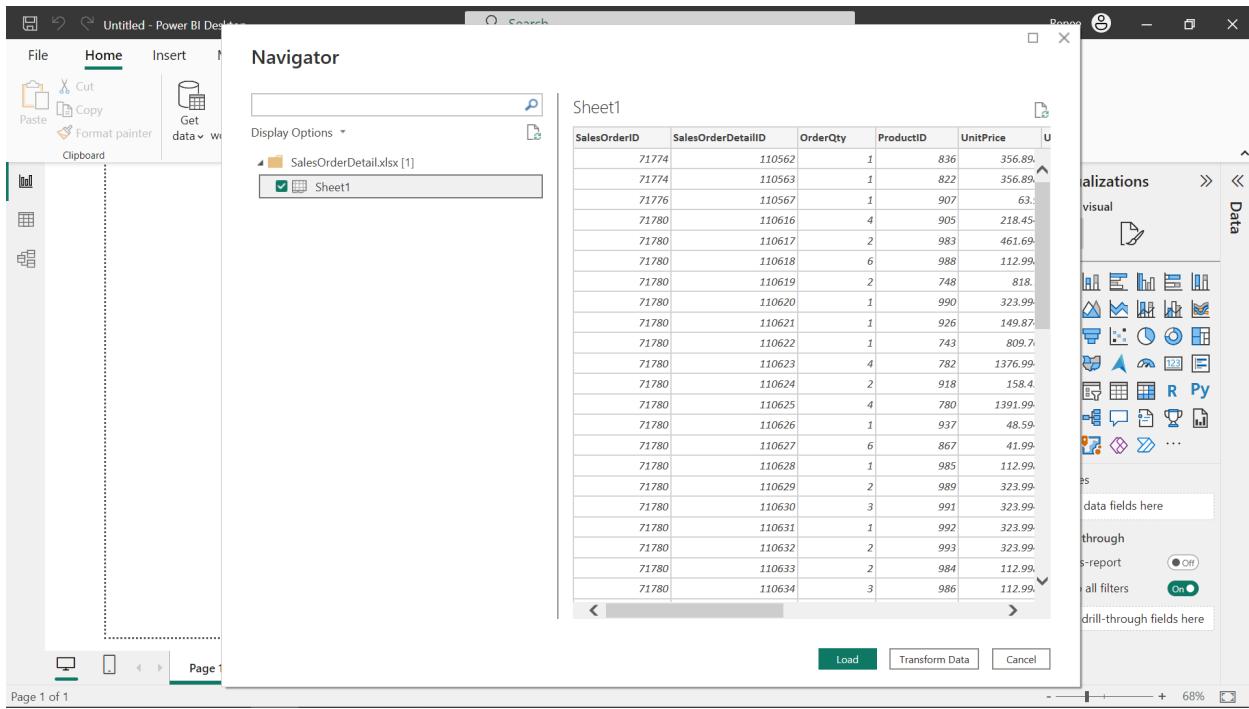


1. Choose Excel from the list of data sources where you can navigate to the location where your Excel file is saved.



From the Navigator window you can choose the worksheet(s) and table(s) you want to import.

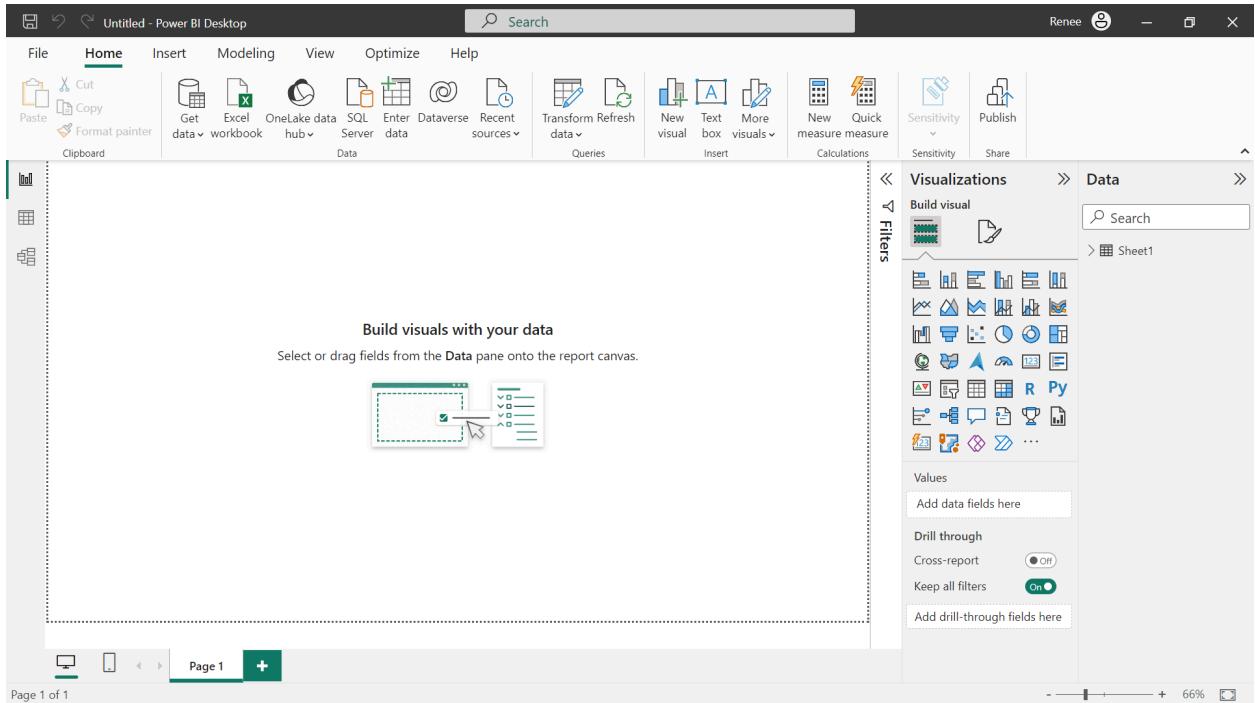
1.



Then select Load. This will import your Excel data into Power BI.

Click the Save button in the top-left corner to save the project.

Power BI will display a message indicating that you are ready to build visuals now that the data has successfully imported.



Completed examples

Compare your work with our finished version. Below is the Table View that is accessible by clicking on the table icon on the left side of the Power BI interface.

Sheet1													
SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid	ModifiedDate					
71774	110562	1	836	356.898	0	356.898	E3A1994C-7A68-4CE8-96A3-77FDD3B7D3	01/06/2008 00:00:00					
71774	110563	1	822	356.898	0	356.898	5C77F557-FDB6-43BA-90B9-9A7AE55CA32	01/06/2008 00:00:00					
71776	110567	1	907	63.9	0	63.9	6DBFE398-D15D-425E-AA58-88178FE360E5	01/06/2008 00:00:00					
71780	110620	1	990	323.994	0	323.994	F117A449-039D-44B8-A4B2-B12001DACC01	01/06/2008 00:00:00					
71780	110621	1	926	149.874	0	149.874	92E5052B-72D0-4C91-9A8C-42591803667E	01/06/2008 00:00:00					
71780	110622	1	743	809.76	0	809.76	8BD33BED-C4F6-4D44-84F8-B7D04AFC7D794	01/06/2008 00:00:00					
71780	110626	1	937	48.594	0	48.594	7F5FEB17-8EF4-4236-9F1C-15046D9638F0	01/06/2008 00:00:00					
71780	110631	1	992	323.994	0	323.994	560FEEE1-DD54-4C34-AB81-F4B841D0AA41	01/06/2008 00:00:00					
71780	110640	1	810	72.162	0	72.162	351A29FB-CEB1-4CA6-BB36-506D87B82A95	01/06/2008 00:00:00					
71780	110642	1	925	149.874	0	149.874	21624302-CA0F-402F-8A46-5A3FFFAT7DF3	01/06/2008 00:00:00					
71780	110644	1	880	32.994	0	32.994	C3FB83EC-3FF6-4E11-88CF-230E128815F3	01/06/2008 00:00:00					
71782	110669	1	954	1430.442	0	1430.442	8D3814C1-DF49-4919-942C-3BB379D7BE42	01/06/2008 00:00:00					
71782	110678	1	978	445.41	0	445.41	8BD27F52-34F9-498D-A838-366179F7075D	01/06/2008 00:00:00					
71782	110679	1	955	1430.442	0	1430.442	1D44BF7E-9C42-4CA1-AFD6-E6DDF9D85BC5	01/06/2008 00:00:00					
71782	110680	1	948	63.9	0	63.9	083CDD70-A4DF-4AFD-863A-19B03DEEFBA4	01/06/2008 00:00:00					
71782	110681	1	994	32.394	0	32.394	8ED3D829-95AD-4BED-8894-9B76D20981B9	01/06/2008 00:00:00					
71782	110682	1	970	728.91	0	728.91	74E94A04-5C2B-4B21-8FC7-069204B0BA4A	01/06/2008 00:00:00					
71782	110688	1	957	1430.442	0	1430.442	8D36D42E-5A61-4AA0-88B7-A83DEEEB3DDE	01/06/2008 00:00:00					
71782	110694	1	963	445.41	0	445.41	48EAD4A8-0A74-4CD4-BD00-E788EEBBCFDE	01/06/2008 00:00:00					
71782	110695	1	953	728.91	0	728.91	8C28E39E-66D9-437D-8D18-233702BC2785	01/06/2008 00:00:00					
71782	110699	1	916	31.584	0	31.584	31.584	20812393-5001-46EB-A7F6-69C537CB76E	01/06/2008 00:00:00				
71782	110700	1	886	200.052	0	200.052	C09D7227-B5C8-4E1F-8489-C5456C71516E	01/06/2008 00:00:00					
71782	110701	1	892	602.346	0	602.346	0233509-02F29-4F6E-B84C-E1D38536D432	01/06/2008 00:00:00					

Table: Sheet1 (542 rows)

Conclusion

Congratulations! You have successfully completed this exercise. You now know how to set up an Excel data source in Power BI.

1.2. Exercise: Implementing triggers

Introduction

Manually importing and analyzing data can be tedious and time-consuming, which is why Power BI offers Triggers and Connectors to automate workflows and save you time.

Triggers are an essential functionality in Power BI that allow you to automate tasks based on specific conditions. By configuring a trigger, you can schedule an action to execute at a particular time or when a particular event occurs. Connectors play a vital role in Power BI by connecting to various data sources and executing actions or triggering workflows based on specific events, saving time and improving the accuracy of your business data.

In this exercise, you will learn how to configure a connector and set up a trigger in Power BI to create a dataset (a collection of data that has been imported, transformed, and loaded into the Power BI service) that is updated each night at the same time from a local Excel file. By the end of this exercise, you will be able to automate tasks and actions to deliver fast, accurate data to meet specific business requirements.

Case study

Your manager, Adio Quinn, a data analyst at Adventure Works, has tasked you with analyzing daily sales reports and providing insights to the management team. However, manually importing and analyzing the data can be time-consuming and tedious.

To automate the process, you want to use Power BI's triggers and actions to configure the workflow. With Power BI, you can easily schedule an action to refresh the data, freeing up time to focus on analyzing the data and providing valuable insights to the management team.

Instructions

Step 1: Prepare the resources

Prepare data in Excel

1. Let's take an Excel file as an example. Download this sample data set and save it to your computer or to OneDrive for Business.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid	ModifiedDate				
2	71774	110562	1	836	356.898	0	356.898	E3A1994C-7A68-4CE8-96A3-77FDD3BBD730	00:00:00				
3	71774	110563	1	822	356.898	0	356.898	5C77F557-FDB6-43BA-90B9-9A7AECC5CA32	00:00:00				
4	71776	110567	1	907	63.9	0	63.9	6DBFE398-D15D-425E-AA58-88178FE360E5	00:00:00				
5	71780	110616	4	905	218.454	0	873.816	377246C9-44B3-48ED-A5B9-E56F005364E0	00:00:00				
6	71780	110617	2	983	461.694	0	923.388	43A54BCD-536D-4A1B-8E69-24D083507A14	00:00:00				
7	71780	110618	6	988	112.998	0.4	406.7928	12706FAB-F3A2-48C6-B7C7-1CCDE4081F18	00:00:00				
8	71780	110619	2	748	818.7	0	1637.4	B12F0D3B-5B4E-4F1F-B2F0-F7CD99D826	00:00:00				
9	71780	110620	1	990	323.994	0	323.994	F117A449-039D-44B8-A4B2-B12001DACC01	00:00:00				
10	71780	110621	1	926	149.874	0	149.874	92E5052B-72D0-4C91-9A8C-42591803667E	00:00:00				
11	71780	110622	1	743	809.76	0	809.76	88D33BED-C4F6-4D44-84FB-A7D04AFCD794	00:00:00				
12	71780	110623	4	782	1376.994	0	5507.976	686999FB-42E6-4D00-9A14-83FFA86833E3	00:00:00				
13	71780	110624	2	918	158.43	0	316.86	82940B03-C70B-4183-8660-6B3418908429	00:00:00				
14	71780	110625	4	780	1391.994	0	5567.976	644B0CD6-B2C3-4E4D-AB43-091C2EF6C829	00:00:00				
15	71780	110626	1	937	48.594	0	48.594	7F5FEB17-8EF4-4236-9F1C-15046D9638F0	00:00:00				
16	71780	110627	6	867	41.994	0	251.964	AC7883BD-B503-41A5-9791-480E528F028C	00:00:00				
17	71780	110628	1	985	112.998	0.4	67.7988	2C10A282-A13D-442A-8F45-F4D6B23A7D9C	00:00:00				
18	71780	110629	2	989	323.994	0	647.988	654FB79E-70D4-B92-9832-9FA67013215B	00:00:00				
19	71780	110630	3	991	323.994	0	971.982	3D6CA7AB-055E-4536-8940-76234C9BCDDE	00:00:00				
20	71780	110631	1	992	323.994	0	323.994	560FEE1-DD54-4C34-ABB1-4F8841DOAA41	00:00:00				
21	71780	110632	2	993	323.994	0	647.988	19570052-4023-4658-BC56-DC5C619BD00E	00:00:00				
22	71780	110633	2	984	112.998	0.4	135.5976	27562675-F8C3-4A3B-BD9E-B366B83E5204	00:00:00				
23	71780	110634	3	986	112.998	0.4	203.3964	E193CE39-EF33-4969-87B1-468D2F7B48AD	00:00:00				

Before you can load your Excel file into Power BI, you must organize your data in a flat table. In a flat table, each

column contains the same data type; for example, text, date, number, or currency. Your table should have a

header row but no columns or rows that display totals. Next, format your data as a table. In Excel, on the Home

tab, in the Styles group, select Format as Table.

Sign up for the Power BI Service

1. Visit the Power BI website at <https://powerbi.microsoft.com> and sign in using your organizational or school account.

Step 2: Get the data to the Power BI service

Once you have the resources ready, you can proceed to select data. The Power BI service connects to many data sources, including Excel files. You can use the Get data option to import or connect to an Excel workbook. From the File menu, select the location of your file.

The sales department at Adventure Works uploads the file every day to the company's OneDrive, so the location you should choose is OneDrive - Business. After uploading your workbook, it will be available in the Workbooks section of the workspace's navigation pane and in the workspace's content list. This action will add a dataset, and the report can be generated and published.

These are the steps to follow:

1. Launch Power BI.
2. Get data to import or connect.
3. Specify location of file.
4. Load to create a dataset.

These steps will guide you to get data ready for an automated refresh.

Step 3: Configure a trigger

To schedule updates for your dataset in Power BI Service, you'll need to set up a trigger.

1. Start by selecting the dataset you want to configure. You can do this by navigating to the Datasets tab of the left sidebar and clicking on the dataset you want to work with.

2. Once you've selected your dataset, choose a function to schedule a refresh. You'll have the option to schedule a regular refresh, or to set up an incremental refresh.
3. After selecting your function, you'll need to configure the parameters. This will depend on the function you've chosen, but common parameters include the refresh frequency and the start time for the refresh. Finally, apply the trigger to your dataset. You can do this by selecting Apply in the trigger configuration panel.

Conclusion

Learning how to configure triggers and connectors in Power BI can help you automate repetitive tasks, reduce the time spent on manual data entry, and improve the accuracy of your data. This exercise demonstrated how to set up a time-based trigger to automate the process of importing data from a local Excel file into a Power BI dataset. By following the steps provided, you can create a workflow that automatically updates the dataset at a specific time on schedule. This will allow you to focus on analyzing the data and providing valuable insights to your team at Adventure Works, without having to worry about the tedious and time-consuming task of manually importing and analyzing the data.

Exemplar: Implementing triggers

Overview

In the exercise Implementing Triggers, you learned how to set up a trigger and configure a connector to create a dataset that is automatically updated each night, saving you time and providing accurate data to meet your business needs.

In this reading, you will follow the steps to learn how Adio Quinn, a data analyst at Adventure Works, automated the daily task assigned to him. With Power BI, Adio could easily schedule data refreshes, allowing him to focus on delivering valuable insights to the management team and saving time in the process.

The following steps will create a trigger to import data from an Excel file every night at the same time. The report will be updated with the latest data.

Step 1: Prepare the resources

To begin, make sure you have all the required resources for this exemplar, including:

1. Access to Power BI. Sign into Power BI Service using your organizational or school account.
2. An Excel file. A sample dataset can be downloaded from the link to follow through the exercise.
3. A designated worksheet or table within the Excel file to serve as the data source.
4. The specific time for updating the data source.

To prepare your data for loading into Power BI, you need to format it as a table. To do this in Excel, go to the

Home tab and click on the Styles group. From there, choose the Format as Table option and select a suitable

style for your worksheet. Once you've completed this step, your Excel worksheet is now properly formatted and

ready to be loaded into Power BI.

	A	B	C	D	E	F	G	H	I	J	K
1	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid	ModifiedDate		
2	71774	110562	1	836	356.898	0	356.898	E3A1994C-7A68-4CE8-96A3-77FDD3BB0730	00:00:00		
3	71774	110563	1	822	356.898	0	356.898	5C77F557-FDB6-43BA-90B9-9A7AEC55CA32	00:00:00		
4	71776	110567	1	907	63.9	0	63.9	6DBFEE398-D15D-425E-AA58-88178FE360E5	00:00:00		
5	71780	110616	4	905	218.454	0	873.816	377246C9-4483-48ED-A5B9-E56F005364EO	00:00:00		
6	71780	110617	2	983	461.694	0	923.388	43A54BCD-536D-4A1B-8E69-24D083507A14	00:00:00		
7	71780	110618	6	988	112.998	0.4	406.7928	12706FAB-F3A2-48C6-B7C7-1CCDE4081F18	00:00:00		
8	71780	110619	2	748	818.7	0	1637.4	B12F0D3B-5B4E-4F1F-B2F0-F7CDE99DD826	00:00:00		
9	71780	110620	1	990	323.994	0	323.994	F117AA49-039D-44B8-A4B2-B12001DACC01	00:00:00		
10	71780	110621	1	926	149.874	0	149.874	92E50502B-72D0-4C91-9A8C-42591803667E	00:00:00		
11	71780	110622	1	743	809.76	0	809.76	8BD33BED-C4F6-4D44-84FB-A7D04AFC7D94	00:00:00		
12	71780	110623	4	782	1376.994	0	5507.976	686999FB-42E6-4D00-9A14-83FFA86833E3	00:00:00		
13	71780	110624	2	918	158.43	0	316.86	82940B03-C70B-4183-8660-6B3418908429	00:00:00		
14	71780	110625	4	780	1391.994	0	5567.976	644B0CD6-B2C3-4E4D-AB43-091C2EF6C829	00:00:00		
15	71780	110626	1	937	48.594	0	48.594	7F5FEB17-8EF4-4236-9F1C-15046D9638F0	00:00:00		
16	71780	110627	6	867	41.994	0	251.964	A7C78838D-B503-41A5-9791-480E528F028C	00:00:00		
17	71780	110628	1	985	112.998	0.4	67.7988	2C10A282-A13D-442A-8F45-F4D6B23A7D9C	00:00:00		
18	71780	110629	2	989	323.994	0	647.988	654FB79E-70DF-4B92-9832-9FA67013215B	00:00:00		
19	71780	110630	3	991	323.994	0	971.982	3D6CA7AB-055E-4536-8940-76234CC9BCDE	00:00:00		
20	71780	110631	1	992	323.994	0	323.994	560FEE1-DD54-4C34-ABB1-4F8841D0AA41	00:00:00		
21	71780	110632	2	993	323.994	0	647.988	19570052-4023-4658-BC56-DC5C619B00E	00:00:00		
22	71780	110633	2	984	112.998	0.4	135.5976	27562675-F8C3-4A38-BD9E-B366B83E5204	00:00:00		
23	71780	110634	3	986	112.998	0.4	203.3964	E193CE39-F33-4969-87B1-468D2F7B48AD	00:00:00		

Step 2: Get the data

Before you begin preparing your data, let's review some important concepts, starting with the difference between a data source and a dataset. To help you understand these terms, let's use an analogy: think of a dataset in Power BI like a recipe in cooking, complete with instructions and ingredients. Your dataset contains all the data, metrics, and information that are displayed on your Power BI dashboard. Like a chef, you can obtain the ingredients (the data) from a store or garden (the data sources), which could be a database, web API, or CSV file, and use them to create a recipe (dataset) that produces insights and visualizations in Power BI.

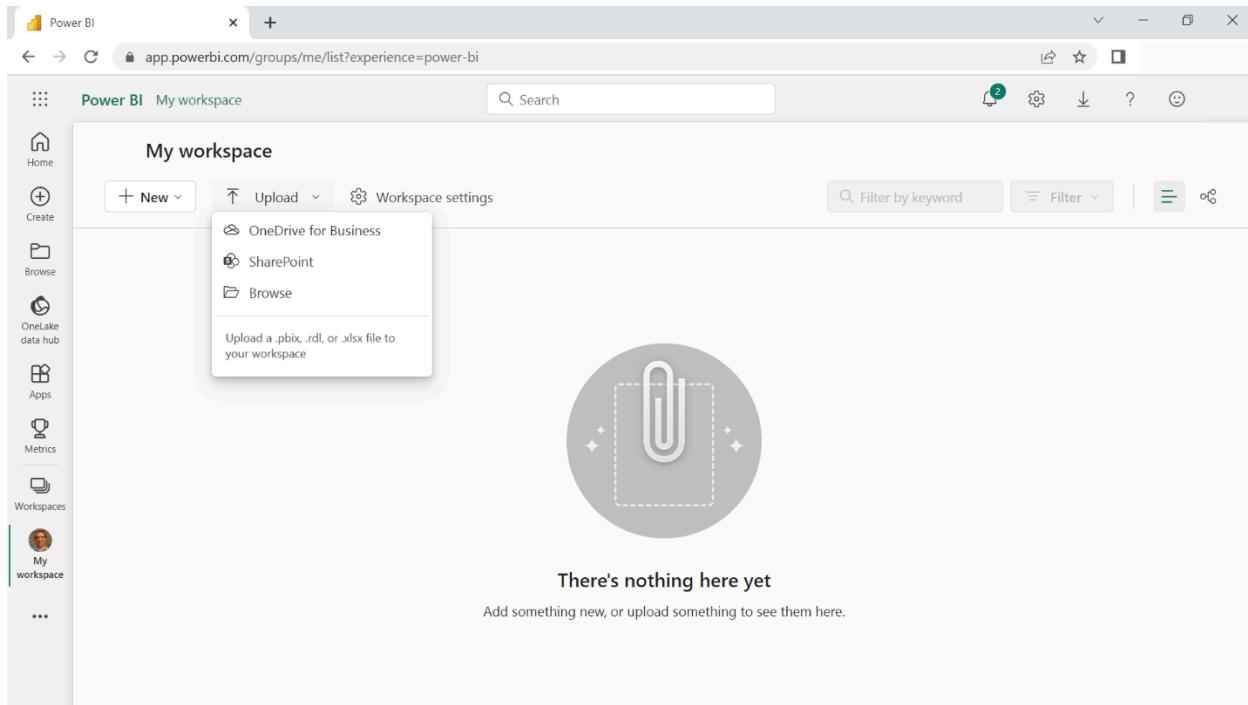
Now, let's examine the difference between Power BI Desktop and service when it comes to uploading data. In Power BI Desktop, users can create three types of models to create datasets: Import, DirectQuery, and Composite. These models are based on how external data sources are integrated with imported data.

The Power BI service offers a streamlined experience for uploading files. You can upload files from your OneDrive for Business or SharePoint sites, or directly from your computer. If you choose OneDrive or SharePoint, Power BI creates a connection to the file. If you upload local files, a copy of the file is added to your workspace.

Let's now continue with the steps to Get Data:

Launch Power BI service.

1. To import or connect to an Excel workbook, start by going to My Workspace in Power BI. From there, select the Upload button to upload your Excel file. Since the sales department at Adventure Works uploads the file every day to the company's OneDrive, select OneDrive - Business as the data source. To do this, select Upload in the top-left corner, then select OneDrive - Business. This will allow you to easily access the Excel file in Power BI.



1. Select Import and browse to the location of your Excel file and select it.

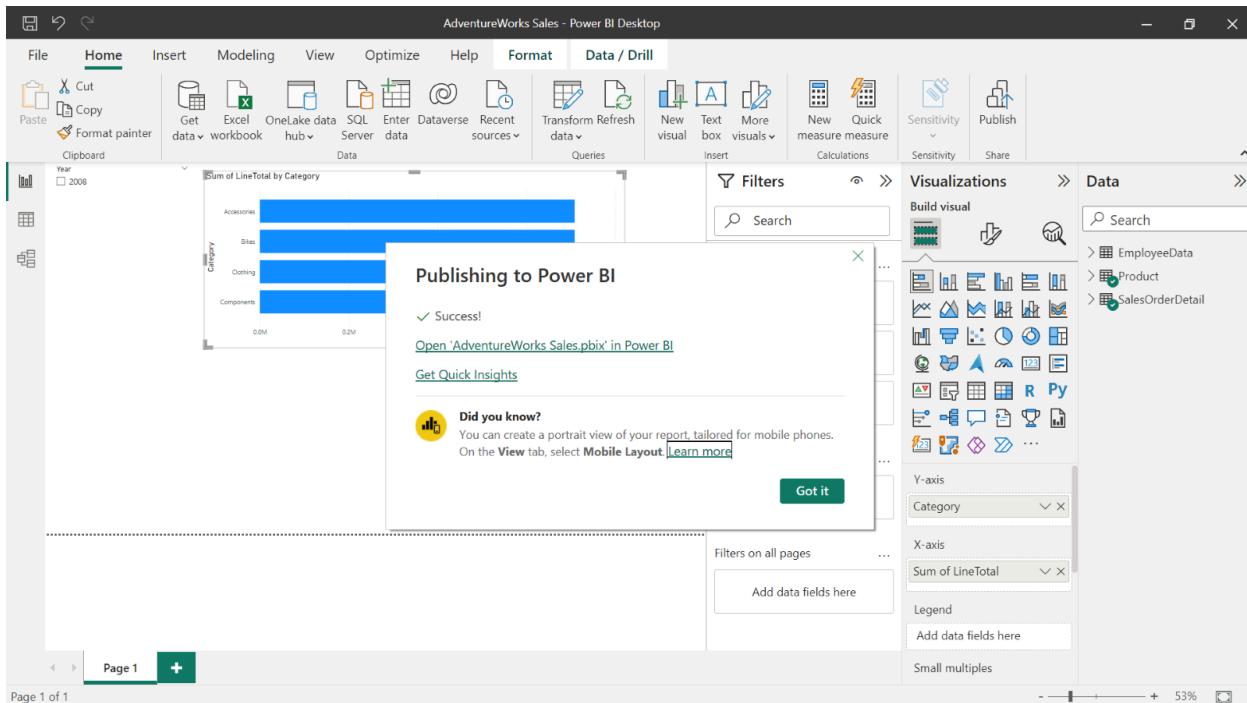
Choose Import to upload your workbook. When you choose Import, Power BI imports any supported data in tables and any data model into a new Power BI dataset.

2. When you upload an Excel file, the workbook item appears in Power BI just like it would in Excel Online, as shown in the following image.

The screenshot shows the Power BI service interface. The top navigation bar includes the Power BI logo, a search bar, and various navigation icons. The main content area is titled 'My workspace' and displays a list of items. On the left, there is a sidebar with icons for Home, Create, Browse, OneLake data hub, Apps, Metrics, Workspaces, and 'My workspace' (which is selected and highlighted with a green bar). The main content area has a header with buttons for '+ New', 'Upload', 'Workspace settings', and filters ('Filter by keyword', 'Filter', and a refresh icon). Below this is a table with columns: Name, Type, Owner, Refreshed, Next refresh, and Endorsements. A single item is listed: 'SalesOrderDetail' (Workbook, My workspace, Refreshed 6/27/23, 10:32:32 AM, Next refresh —, Endorsements —).

When you publish a Power BI Desktop file to Power BI service, you publish the data in the model to your Power BI workspace.

1. Open your Power BI Desktop application and load the dataset and report that you want to publish.
2. Choose File from the menu bar, then select Publish and select Publish to Power BI. Alternatively, you can select Publish from the Home ribbon.
3. If you are not already signed into Power BI, you will be prompted to do so. Enter your credentials to sign in.
4. Select the destination workspace where you want to publish your report. You can search your list of available workspaces by typing in the search box to filter your workspaces. Once you have found the workspace, select the Select button to publish.
5. Power BI will begin publishing your report to the selected workspace. Wait for the process to complete.
6. When publishing is complete, a dialog box will appear with a link to your report. You can then select the link to open the report in your Power BI site.



Step 3: Configure trigger and incremental refresh

The Scheduled refresh section is used to set the frequency and time slots for refreshing the dataset. While some data sources can be refreshed without a gateway configuration, others may require it. In the case of DirectQuery scenarios, if a dataset meets the criteria for performance optimization, the Refresh schedule will be relocated to the Optimize performance section. To configure the settings, turn on the Keep your data up to date slider.

Here are the steps to schedule a refresh for your dataset in Power BI:

1. Navigate to the Data Hub tab in Power BI and select the one you want to configure by clicking on it.
2. Choose a function to schedule a refresh, either a regular refresh or an incremental refresh.
3. Configure the parameters, such as refresh frequency and start time.
4. Apply the trigger by selecting Apply.

Scheduled refresh

Keep your data up to date

Configure a data refresh schedule to import data from the data source into the dataset. [Learn more](#)



On

Refresh frequency



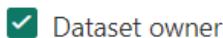
Time zone



Time

[Add another time](#)

Send refresh failure notifications to



Dataset owner



These contacts:

Enter email addresses

Apply

Discard

By following these steps, you'll be able to configure a trigger to schedule updates.

Conclusion

Learning how to configure triggers and connectors in Power BI can be a game-changer for automating repetitive tasks, reducing manual data entry, and improving data accuracy. By following the steps outlined in this exemplar, you can set up a trigger and connector to automatically import data from an Excel file into Power BI. This workflow will allow you to save time and effort by automatically updating your data at a specific time each night, giving you more time to focus on analyzing the data and providing valuable insights to your team at Adventure Works. With this knowledge, you can become more productive and efficient in your data analysis tasks and meet specific business requirements with fast and accurate data.

2.1. Exercise: Preparing a dataset

Introduction

By now, you should be familiar with common data errors and how to resolve them.

You've explored several examples of common data errors and discovered how they can be resolved using Power Query. In this exercise, you'll put your knowledge into practice by importing, cleaning, and transforming data in Power BI Desktop to ensure accurate and reliable data analysis.

Case study

Adventure Works needs your help to analyze its sales data and create a Power BI report that visualizes this data in a meaningful way. Before the analysis can begin, the raw data must be cleaned and transformed to make sure it's accurate and consistent.

The company sends you the CSV file containing the raw dataset named *SalesFile.csv*. The dataset consists of important data related to recent sales, such as product categories, manufacturing prices, sales prices, units sold, and other similar details.

However, the file also contains errors like missing values, incorrect data types, and inconsistent formatting. Help Adventure Works to resolve these issues using Power Query editor so that they can produce an accurate and reliable report.

1. This exercise aims to assist you in understanding how to address common data issues such as missing values, incorrect data types, and inconsistent formatting.
2. By the end of this exercise, you'll understand how to import, clean, and transform data in Power BI Desktop to ensure accurate and reliable analysis.

Instructions

Create a new Power BI project called Exercise – Preparing a dataset. Follow the prompts below to complete the exercise.

Step 1: Load the workbook

1. Download the Microsoft Excel workbook SalesFile.xlsx.
2. Import the SalesFile.xlsx Excel file as your dataset in Power BI.

Step 2: Open the Power Query Editor

1. Open the Power Query editor to begin editing your data.

Step 3: Address missing values

1. Locate and select the Units Sold column.
2. Identify all null values within the column and replace them with a value of 0.
3. Repeat this task for the Sale Price, Sales, and Profit columns.

Step 4: Clean the Manufacturing Price and Sale Price columns

1. Locate and select the Manufacturing Price and Sale Price columns.
2. Change the data type for both columns to Decimal Number.
3. Repeat this task for the Sales and Profit columns.

Step 5: Clean the Discount Band Column

1. Select the Discount Band column.
2. Locate each instance of value 1 in the column. Replace each instance of this value with None.
3. Then change the data type of the column to Text.

Step 6: Clean the Units Sold column

1. Select the Units Sold column. Search for and locate all instances of the text value six hundred.
2. Replace each instance of this text value with the numerical value 600.
3. Then change the column's data type to Whole Number.

Step 7: Address inconsistencies in the Date column

1. Select the Date column. Ensure that the column's data type is Date.
2. The column also contains several null values. Replace all null values with the default date of March 03rd 2023.
3. Next, select the Month Number column. Change the column's data type to Whole Number.

Step 8: Drop records with errors

1. Select the Manufacturing Price column. The column contains errors in rows 6 and 38. Use the Remove Errors feature to drop these records.
2. Repeat the same steps for the errors in the Sales and Profit columns.

Step 9: Drop duplicate rows

You need to identify several duplicate rows that the dataset contains.

1. Tip: Check the Products column
2. Once identified, remove these duplicate rows from the dataset.

Step 10: Apply the data transformations

1. Once you have completed all the above data cleaning steps, select the Close & Apply button.
2. By completing this action, you can apply your transformations and update the dataset.

Conclusion

By following these steps, you've successfully cleaned and transformed your dataset using Power BI. The dataset is now ready for further analysis and visualization in Power BI Desktop.

Remember, investing time and effort in cleaning your data will greatly enhance the quality of your analysis and ultimately lead to better decision-making and business outcomes.

Always strive for accuracy and consistency, and don't be afraid to dive into the details when working with data. With these skills, you'll be well-equipped to tackle any data challenges that come your way!

Exemplar: Preparing a dataset

Introduction

In the exercise *Preparing a data set*, you put into practice your understanding of how to prepare a data set by importing, cleaning, and transforming data in Power Query.

Your objective in this exercise was to prepare a worksheet for analysis by completing the following tasks:

- Address instances of missing values.
- Clean columns by changing data types as required.
- Replace values where required.
- Address inconsistencies in data.
- And drop records with errors.

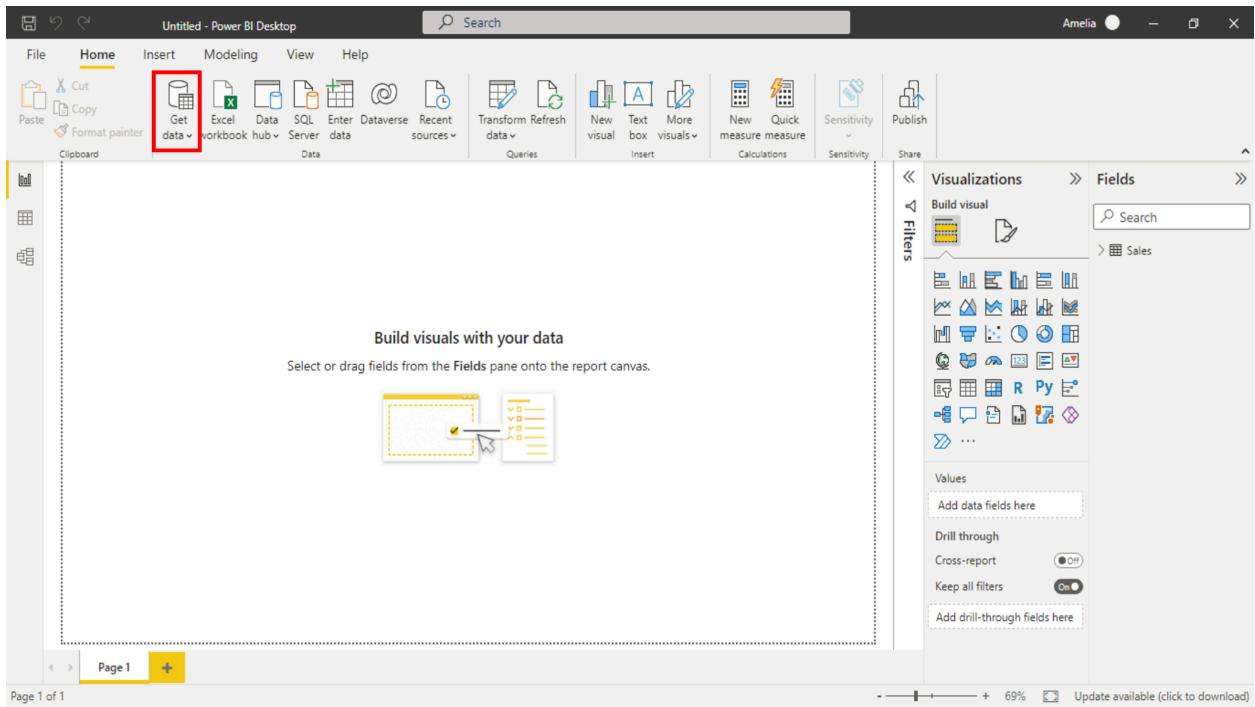
This reading provides a step-by-step guide for completing these tasks, accompanied by screenshots for easy comparison with your own copy. It also offers additional resources that provide more information on related topics.

Step 1: Load the workbook

Import your dataset.

Download the Microsoft Excel workbook *SalesFile.xlsx*.

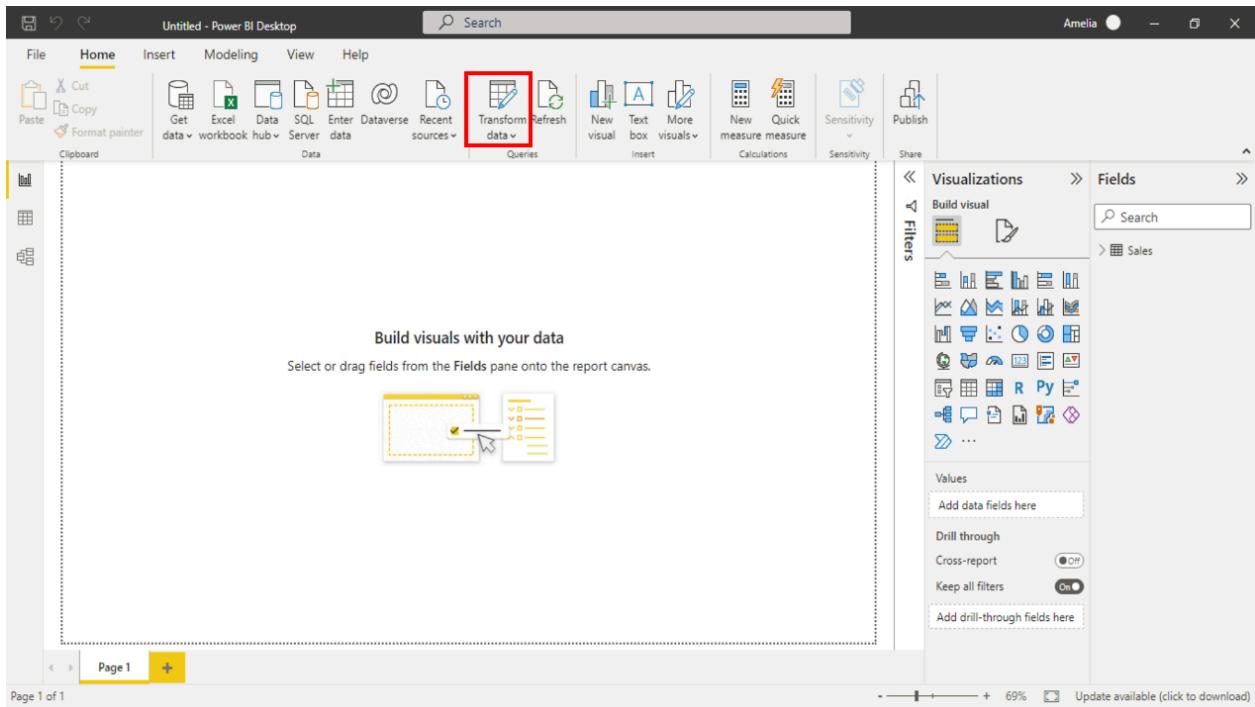
1. At the top of the Power BI window, navigate to the Home ribbon tab.
2. In the Data group, select the Get Data option. Select *SalesFile.xlsx* as the data source to import.



Step 2: Open the Power Query editor

Open the Power Query editor to begin editing your data.

1. In the Queries group, select the Transform Data button. It's located in the middle of the toolbar.
2. The Power Query Editor window opens. You can now begin cleaning the data.



Step 3: Address missing values

Locate and select the *Units Sold* column. Identify all *NULL* values within the column and replace them with a value of 0. Repeat this task for the *Sale Price*, *Sales*, and *Profit* columns.

1. In the Power Query Editor, locate the *Units Sold* column in the dataset preview.
2. Select the header of the *Units Sold* column to highlight it.
3. In the Home tab at the top of the window, select the Replace Values button.

A screenshot of the Microsoft Power BI desktop application. The ribbon at the top has several tabs: File, Home, Transform, Add Column, View, Tools, and Help. Under the Transform tab, there are icons for Close & Apply, New Source, Recent Sources, Enter Data, Data source settings, Manage Parameters, Refresh Preview, Advanced Editor, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Sort, Split Column, Group By, and Replace Values. The 'Replace Values' icon is highlighted with a red box. Below the ribbon is a 'Queries [1]' pane showing a single query named 'Sales'. The main area displays a table with columns: Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, Sales, and Profit. The 'Units Sold' column is currently selected. At the bottom left, it says '9 COLUMNS, 51 ROWS' and 'Column profiling based on top 1000 rows'. At the bottom right, it says 'PREVIEW DOWNLOADED AT 0622 PM'.

1. In the Replace Values window that appears, type null in the Value to Find field (indicating missing values) and type 0 in the Replace With field.

A screenshot of the Microsoft Power BI desktop application. The ribbon and query structure are identical to the previous screenshot. A 'Replace Values' dialog box is overlaid on the main data grid. The dialog has a title 'Replace Values' and a sub-instruction 'Replace one value with another in the selected columns.' It contains two input fields: 'Value To Find' with 'null' typed into it, and 'Replace With' with '0' typed into it. There are 'OK' and 'Cancel' buttons at the bottom right of the dialog. The main data grid below shows the same table structure as the first screenshot, with the 'Units Sold' column still selected.

1. Select the OK button to replace all missing values in the Units Sold column with 0.
2. Now, select the down arrow near the Units Sold column and confirm that no NULL values remain in the column. By performing this operation, you have replaced the non-numeric value NULL with the numeric value 0 in the Units Sold rows.
3. Repeat these steps for the Sale Price, Sales, and Profit columns.

Step 4: Clean the Manufacturing Price and Sale Price columns

Locate and select the Manufacturing Price and Sale Price columns. Change the data type for both columns to Decimal Number.

1. Highlight the Manufacturing Price column by selecting its header.
2. Change the data type of the Manufacturing Price column to Decimal Number by selecting the data type icon next to the column name (it may initially show as ABC or 123) and choosing Decimal Number from the drop-down menu.
3. Take note of how the appearance of the Manufacturing Price column values has changed.
4. Repeat this task for the Sale Price, Sales, and Profit columns.

The screenshot shows the Microsoft Power BI Data Editor interface. The ribbon at the top has the 'Transform' tab selected. Below the ribbon is a toolbar with various icons for data management. The main area displays a table with 9 columns and 51 rows. The columns are labeled: Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, Sales, and Profit. The 'Sale Price' column is currently selected, as indicated by a yellow background. A context menu is open over the '1.2 Decimal Number' option in the dropdown for this column. The menu includes other options like 'Text', 'True/False', 'Binary', and 'Using Locale...'. At the bottom of the screen, there is a status bar with the text '9 COLUMNS, 51 ROWS Column profiling based on top 1000 rows' and 'PREVIEW DOWNLOADED AT 06:22 PM'.

Step 5: Clean the Discount Band column

Select the *Discount Band* column. Locate each instance of value 1 in the column.

Replace each instance of this value with None. Then change the data type of the column to Text.

1. Highlight the Discount Band column by selecting its header.
2. Select the Replace Values button in the Home ribbon tab.
3. In the Replace Values window, enter 1 in the Value to Find field and None in the Replace With field.

The screenshot shows the Microsoft Power BI Data Editor interface. A 'Replace Values' dialog box is open over a table. The dialog box has two input fields: 'Value To Find' containing '1' and 'Replace With' containing 'None'. Both fields are highlighted with a red box. The background table has columns: Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, Sales, and Profit. The 'Discount Band' column contains values like 'High', 'None', and 'Low'. The 'Sale Price' column contains numerical values like 5, 300, and 500. The 'Sales' and 'Profit' columns contain large numerical values.

1. Select the OK button to replace the incorrect entry with None.
2. Select the down arrow near the Discount Band column and confirm that the None value is added to the value list.
3. Change the data type of the Discount Band column to Text by selecting the data type icon next to the column name (it may initially show as ABC or 123) and choosing Text from the drop-down menu.
4. Take note of how the appearance of the Discount Band column values has changed.

Step 6: Clean the Units Sold column

Select the Units Sold column. Search for and locate all instances of the text value six hundred. Replace each instance of this text value with the numerical value 600. Then change the column's data type to Whole Number.

1. Highlight the Units Sold column by selecting its header.
2. On the Home tab, select the Replace Values button.

- In the Replace Values window, enter six hundred (the text value to be replaced) in the Value to Find field and 600 (the new numeric value) in the Replace With field.

The screenshot shows the Microsoft Power BI Data Editor interface. A 'Replace Values' dialog box is open, overlaid on a table of sales data. The dialog box contains two input fields: 'Value To Find' with the value 'six hundred' and 'Replace With' with the value '600'. Both input fields are highlighted with a red box. Below the input fields are 'OK' and 'Cancel' buttons. The background table has columns: Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, Sales, and Profit. The 'Units Sold' column contains several non-numeric values like 'None', 'High', 'Low', etc., which will be replaced by the numeric value 600.

- Select the OK button to replace the text entry with the numerical value. By performing this operation, you have replaced another non-numeric value of six hundred with a numeric value of 600 in the Units Sold rows. Now you are ready to change the column's data type.
- Change the data type of the Units Sold column to Whole Number by selecting the data type icon next to the column name (it may initially show as ABC or 123) and choosing Whole Number from the drop-down menu.
- Take note of how the appearance of the Units Sold column values has changed.

Step 7: Address inconsistencies in the Date column

Select the Date column. Change the column's data type to Date. The column also contains several empty values. Replace all empty values with the default date of 03 March 2023. Next, select the Month Number column. Change the column's data type to Whole Number.

1. Highlight the Date column by selecting its header.
2. Change the data type of the Date column to the Date data type by selecting the data type icon next to the column name (it may initially show as ABC or 123) and choosing the Date option from the drop-down menu.
3. Replace any empty values with a default date (of 03 March 2023) using the Replace Values tool.

The screenshot shows the Microsoft Power BI desktop interface. A 'Replace Values' dialog box is open, overlaid on a table. The table has columns labeled 'Units Sold', 'Manufacturing Price', 'Sale Price', 'Sales', 'Profit', and 'Date'. The 'Date' column has a dropdown arrow icon. The 'Replace Values' dialog box contains fields for 'Value To Find' and 'Replace With', both of which are currently empty. A red box highlights these two fields. The 'OK' button is located at the bottom right of the dialog. The Power BI ribbon is visible at the top, showing various tabs like File, Home, Transform, etc.

1. Next, select the down arrow near the Month Number column. Confirm that there are no empty values and that the 03 March 2023 value is added to the value list.
2. Highlight the Month Number column by selecting its header.
3. Change the data type of the Month Number column to Whole Number by selecting the data type icon next to the column name (it may initially show as ABC or 123) and choosing Whole Number from the drop-down menu.

The screenshot shows the Microsoft Power Query Editor interface. A table is displayed with columns: ABC Units Sold, ABC Manufacturing Price, 1.2 Sale Price, ABC Sales, ABC Profit, ABC Date, and Month Number. The 'Month Number' column has a dropdown menu open, showing options like '1-3 Whole Number', '1-3 Decimal Number', '\$ Fixed decimal number', '% Percentage', 'Date/Time', 'Date', 'Time', 'Date/Time/Timezone', 'Duration', 'Text', 'True/False', and 'Binary'. The '1-3 Whole Number' option is highlighted with a red box.

1. Take note of how the appearance of the Month Number column values has changed.

Step 8: Drop records with errors

Select the Manufacturing Price column. The column contains errors in rows 6 and 38. Use the Remove Errors feature to drop these records. Repeat the same steps for the errors in the Sales and Profit columns.

1. Examine each column individually to identify any columns that may have errors. There is an error in the Manufacturing Price column. Select the column header to select it.
2. With the column selected, navigate to the Home tab on the Ribbon.
3. Locate the Remove Rows button. Select the button to reveal a drop-down menu.
4. In the drop-down menu, choose Remove Errors. Power Query scans the selected column and removes all rows it identifies within the column that contain errors.
5. Confirm that no errors remain on any rows in the Manufacturing Price column.

6. Repeat these steps for the Sales and Profit columns.

The screenshot shows the Microsoft Power Query Editor interface. The top ribbon has tabs like File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. On the far right of the ribbon, there are various icons for merging queries, text analytics, and machine learning. Below the ribbon is a toolbar with buttons for Close & Apply, New Source, Recent Sources, Enter Data, Data source settings, Manage Parameters, Refresh Preview, Properties, Manage, Choose Columns, Remove Columns, Keep Rows, Remove Rows (which is highlighted with a red box), Reduce, Split Column, Group By, Data Type (set to Any), Use First Row as Headers, Transform, Merge Queries, Append Queries, Combine Files, Combine, and AI Insights. The main area shows a table with columns: ABC Units Sold, Manufacturing Price, Sale Price, Profit, Date, and Month Number. The 'Sale Price' column is currently selected. A context menu is open over the table, with the 'Remove Duplicates' option also highlighted with a red box. At the bottom left, it says '9 COLUMNS, 51 ROWS Column profiling based on top 1000 rows'. At the bottom right, it says 'PREVIEW DOWNLOADED AT 05:46 PM'.

Step 9: Drop duplicate rows

The dataset contains several duplicate rows. Remove these duplicate rows from the dataset.

1. In the Power Query Editor, locate the Queries pane on the left side of the window. Select the table name.
2. In the Home tab, locate the Remove Rows button in the toolbar. Select the button to reveal a drop-down menu.
3. In the drop-down menu, choose Remove Duplicates. Power Query scans your dataset and automatically removes any duplicate rows it identifies.

The screenshot shows the Microsoft Power Query Editor interface. The 'Transform' tab is active in the ribbon. A context menu is open over a table, with the 'Remove Duplicates' option highlighted. The table contains data from the 'Sales' query, with columns including Product, Discount Band, Units Sold, Price, Sales, and Profit.

Step 10: Apply the data transformations

Once you have completed all the above data cleaning steps, select the Close & Apply button to apply your transformations and update the dataset.

1. Once you have completed all the data cleaning steps outlined above, in the top-left corner of the Power Query Editor window, select the Close & Apply button. This action applies all the transformations you have made and closes the Power Query Editor.

The screenshot shows the Power BI Desktop interface with the 'Sales' query selected in the Data view. The ribbon at the top has the 'Transform' tab selected. The 'Close & Apply' button in the ribbon is highlighted with a red box. The data table below contains 24 rows of product sales information, including columns for Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, Sales, and Profit.

	A ^B Product	A ^B Discount Band	A ^B Units Sold	A ^B Manufacturing Price	A ^B Sale Price	A ^B Sales	A ^B Profit
1	TrailBlazer 1000	None	958	5	300	287400	47900
2	TrailBlazer 2000	Low	2529	10	7	17525,97	4880,97
3	SpeedMaster 1000	Low	918	10	300	269892	40392
4	SpeedMaster 2000	Low	1774	10	125	215097,5	2217,5
5	Explorer 1000	Low	866	250	null	9976,32	7378,32
6	GravityMaster 1000	Medium	1679	260	350	552391	115851
7	GravityMaster 2000	Medium	588	120	20	10936,8	5056,8
8	Pathfinder 1000	Medium	1366	260	20	25134,4	11474,4
9	Voyager 1000	High	2072	260	15	27972	7252
10	Voyager 2000	High	600	5	15	8936,4	2166,4
11	Adventurer 1000	High	2641	10	20	45953,4	19543,4
12	Adventurer 2000	High	1727	5	7	10396,54	1761,54
13	EnduroMaster 1000	High	663	120	125	70443,75	-9116,25
14	FatTrail 1000	Low	1445	3	12	17166,6	12831,6
15	FatTrail 2000	None	1728	10	300	508032	76032
16	CrossRider 1000	Low	1901	10	12	22127,64	16424,64
17	CrossRider 2000	Low	349	250	350	117264	26524
18	DuoExplorer 1000	Medium	2861	120	15	40769,25	12159,25
19	DuoExplorer 2000	Medium	727	260	350	239183	50163
20	E-Mountain 1000	Medium	3244	250	null	56208,62	26475,12
21	E-Mountain 2000	Medium	2460	260	300	678960	63960
22	UrbanEco 1000	Medium	1038	10	20	18891,6	8511,6
23	UrbanEco 2000	High	853	260	300	230310	17060
24							

1. Power BI Desktop updates the dataset in the Data view with the cleaned, transformed data.

The screenshot shows the Power BI Desktop interface with the 'Sales' query selected in the Data view. The ribbon at the top has the 'Transform' tab selected. The 'Close & Apply' button in the ribbon is highlighted with a red box. The data table below contains 24 rows of product sales information, including columns for Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, Sales, and Profit.

	A ^B Product	A ^B Discount Band	A ^B Units Sold	A ^B Manufacturing Price	A ^B Sale Price	A ^B Sales	A ^B Profit
1	TrailBlazer 1000	None	958	5	300	287400	47900
2	TrailBlazer 2000	Low	2529	10	7	17525,97	4880,97
3	SpeedMaster 1000	Low	918	10	300	269892	40392
4	SpeedMaster 2000	Low	1774	10	125	215097,5	2217,5
5	Explorer 1000	Low	866	250	null	9976,32	7378,32
6	GravityMaster 1000	Medium	1679	260	350	552391	115851
7	GravityMaster 2000	Medium	588	120	20	10936,8	5056,8
8	Pathfinder 1000	Medium	1366	260	20	25134,4	11474,4
9	Voyager 1000	High	2072	260	15	27972	7252
10	Voyager 2000	High	600	5	15	8936,4	2166,4
11	Adventurer 1000	High	2641	10	20	45953,4	19543,4
12	Adventurer 2000	High	1727	5	7	10396,54	1761,54
13	EnduroMaster 1000	High	663	120	125	70443,75	-9116,25
14	FatTrail 1000	Low	1445	3	12	17166,6	12831,6
15	FatTrail 2000	None	1728	10	300	508032	76032
16	CrossRider 1000	Low	1901	10	12	22127,64	16424,64
17	CrossRider 2000	Low	349	250	350	117264	26524
18	DuoExplorer 1000	Medium	2861	120	15	40769,25	12159,25
19	DuoExplorer 2000	Medium	727	260	350	239183	50163
20	E-Mountain 1000	Medium	3244,5	250	null	56208,62	26475,12
21	E-Mountain 2000	Medium	2460	260	300	678960	63960
22	UrbanEco 1000	Medium	1038	10	20	18891,6	8511,6
23	UrbanEco 2000	High	853	260	300	230310	17060
24							

9 COLUMNS, 44 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 06:22 PM

Conclusion

Your objective for this exercise was to apply techniques for importing, cleaning, and transforming data. In this context, you learned how to use Power Query Editor to change data types, replace missing or inconsistent values, remove duplicate rows, and eliminate rows with errors.

2.2. Activity: Apply a pivot

Introduction

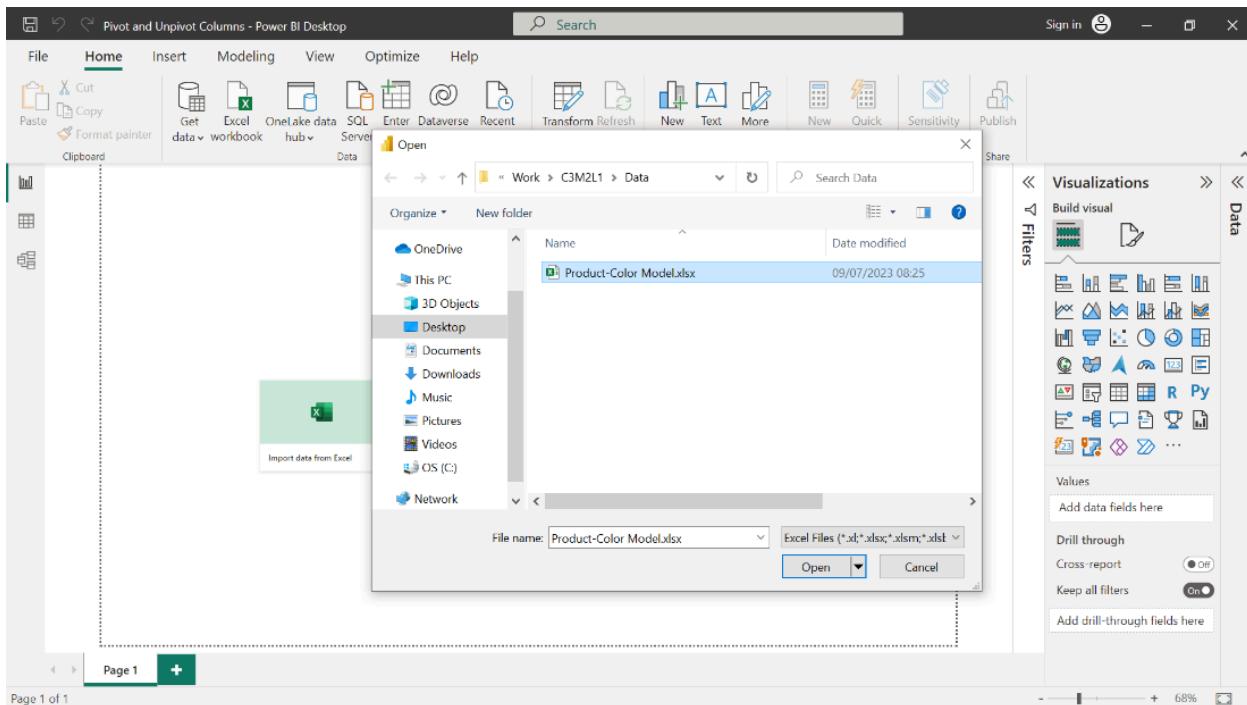
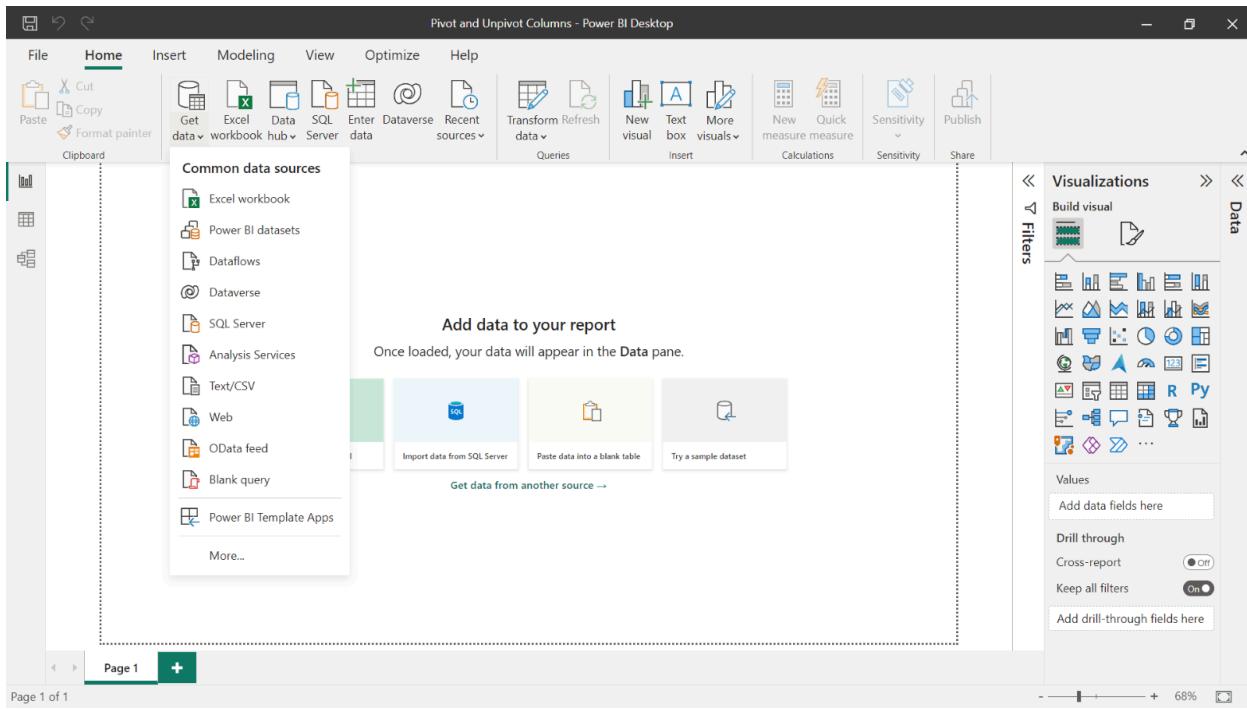
So far, you've learned that it's challenging to identify patterns within unstructured flat data that lacks organization or grouping. Using Power Query, you can transform your flat data into a structured tabular format, aggregating values for each unique value in a designated column. Power Query conducts grouping and aggregate calculations for every individual value in the column, and then pivots the column into a new table, the transformation enabling you to structure your data and analyze it more effectively. In this step-by-step activity, you'll gain hands-on experience in applying the pivot operation in Power BI.

Apply a pivot

In this activity, you will convert an Adventure Works Excel file called *Product-Color Model.xlsx* that includes Product ID, Color, and Model data. You need to present the product count per color in a tabular format. To do this, you need to import the Excel data first and then pivot columns by using aggregate functions. Follow the steps below to apply a pivot in Power BI.

Step 1: Select the data source type

1. Open Power BI Desktop.
2. On the Home ribbon tab, inside the Data group, select the Get Data down arrow followed by Excel to find *Product-Color Model.xlsx*.



Step 2: Import Excel data

1. Import the Excel data to add the Product-Color Model query to the Queries pane.

2. Observe the 3 columns in the table: Product Name, Color and Model.

Remove the Product Name column.

3. Note: Because your objective is to present the product count by color, you do not need this column.

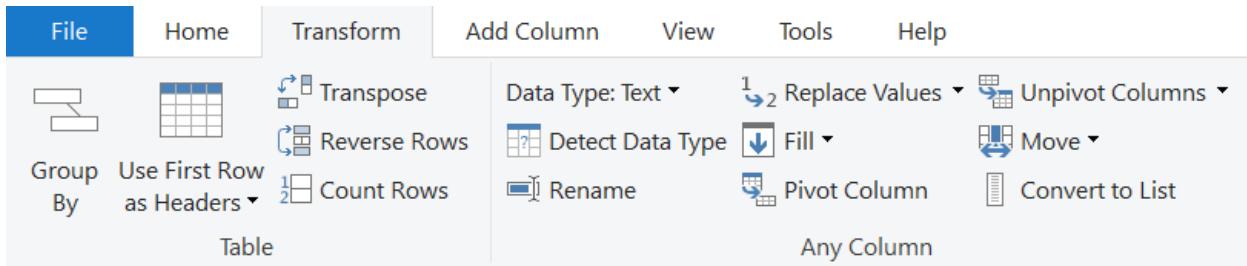
The screenshot shows the Power Query Editor interface. The ribbon at the top includes Home, Transform, Add Column, View, Tools, and Help tabs. The main area displays a table with two columns: 'Color' and 'Model'. The table contains 23 rows of data. The 'Model' column has values like M1, M2, M3, etc. The 'Color' column has values like Black, Blue, Red, etc. To the right of the table, the 'Applied Steps' pane shows the steps taken: Source, Navigation, Changed Type, Promoted Headers, and Changed Type1, followed by a step labeled 'Removed Columns'. The 'Properties' pane on the far right shows the query name as 'Product Color Model'.

Step 3: Pivot columns

1. To pivot the table columns, select the Product-Color Model query on the left menu.

Select the Transform ribbon tab, followed by Pivot Column.

- 2.



1. On the Pivot Column window that displays, select the Color column in the query and keep the value in Values column dropdown list.
2. Expand the Advanced options and select option Count (All) from the Aggregate Value Function dropdown list, and then select OK.

The screenshot shows the Power Query Editor interface. In the center, the 'Pivot Column' dialog box is open, prompting the user to use names in the 'Color' column to create new columns. The 'Values Column' dropdown is set to 'Model'. Below it, the 'Advanced options' section is expanded, with a link to 'Learn more about Pivot Column'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons. To the right of the dialog, the 'Query Settings' pane is visible, showing the query name 'Product Color Model'. The 'APPLIED STEPS' pane lists the steps taken: Source, Navigation, Changed Type, Promoted Headers, Changed Type1, and Removed Columns. The status bar at the bottom indicates '2 COLUMNS, 28 ROWS' and 'Column profiling based on top 1000 rows'.

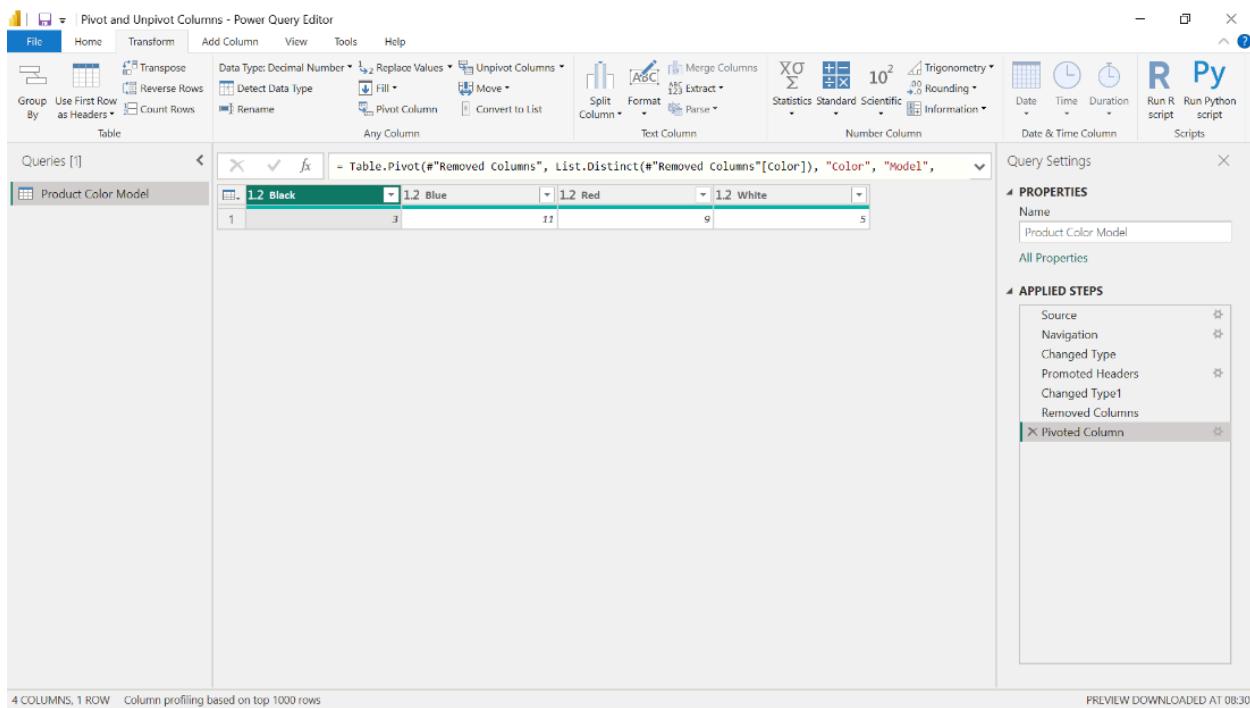
Note: Count is an aggregate function and Count(All) counts every row and returns an integer value. Generally,

it's recommended to use Count as the aggregate function. In some cases, you may use SUM, which calculates

the summation of all the rows. MIN, MAX or AVG functions are rarely used.

Step 4: Examine the changes

By applying the Pivot Column, you change the way the data is organized. All distinct color values are converted to the column headers. There is just one row in the table, where every column has a matching count of that category in the previous rows. You can use this data to visualize in reports or dashboards, primarily in pie chart reports.



The screenshot shows the Power Query Editor interface with the following details:

- File** tab is selected.
- Transform** ribbon tab is active.
- Queries [1]** pane shows a single query named "Product Color Model".
- Table** view shows a table with four columns: "1.2 Black" (containing value 3), "1.2 Blue" (containing value 11), "1.2 Red" (containing value 9), and "1.2 White" (containing value 5).
- Applied Steps** pane on the right lists the following steps:
 - Pivoted Column
 - Removed Columns
 - Changed Type1
 - Promoted Headers
 - Changed Type
 - Navigation
 - Source
- Properties** pane shows the query name as "Product Color Model".
- Query Settings** pane is visible.
- Bottom status bar: "4 COLUMNS, 1 ROW" and "Column profiling based on top 1000 rows".
- Bottom right: "PREVIEW DOWNLOADED AT 08:30"

Conclusion

In this step-by-step activity, you practiced applying a pivot operation in Power BI — an important operation especially when working with Excel data. You observed that the pivot operation converts row values to columns and sets aggregate function results for the column rows. Remember that both pivoting and unpivoting operations are useful where you need rows converted to columns and columns converted to rows respectively. Now that you've applied the pivoting technique, you're better prepared to use them in a data analysis task.

2.3. Exercise: Appending two tables

Introduction

By now, you should have a good understanding of the concept of combining data and its practical applications. You covered the two primary data combining operations, append and merge. You initially focused on appending data. In this exercise, you will have the opportunity to apply your knowledge by using Power Query to append rows, allowing you to consolidate data and simplify data management.

Case study

Adventure Works has recently acquired another bicycle business. Adventure Works' CEO, Jamie Lee, has assigned a task to the sales department to ensure that the sales data from the newly acquired business is reported in the Adventure Works sales reports. Your manager, Adio Quinn, has tasked you with creating a Power BI query that merges the data.

The company provided the two Excel files containing the Adventure Works sales data, *AdventureWorksSales.xlsx* and the new company's data, *OtherSales.xlsx*. The datasets have some common data such as SalesOrderID, SalesOrderDetailID, UnitPrice and SalesData as well as columns with different names. As you learned in previous videos, in tables that will be combined, the column names that you want to stack vertically should be the same. For columns with matching names, their row values are stacked vertically to merge them, while for columns that exist separately in each set, they will be added horizontally, with null values for tables where those columns were not present before. To complete your task successfully you have to rename the common columns and give them the same names. You also have to remove any unnecessary columns.

This exercise aims to assist you in understanding how to combine data by appending data. By the end of this exercise, you'll understand how to append data in Power Query, add rows together and consolidate data.

Instructions

Create a new Power BI project called Exercise – Appending a dataset. Follow the prompts below to complete the exercise.

Step 1: Download the Excel files

- Download the *AdventureWorksSales.xlsx* and *OtherSales.xlsx* files, which you will use in this exercise.

The screenshot shows the Power BI Power Query Editor interface with two queries: "AdventureWorksSales" and "OtherSales". A transformation step is visible between them, titled "Table.TransformColumnTypes(#'Promoted Headers', {{"SalesOrderID", Int64.Type}, {"SalesOrderDetailID", Int64.Type}, {"OrderQty", Int64.Type}, {"ProductID", type text}, {"UnitPrice", type number}, {"UnitPriceDiscount", type number}, {"LineTotal", type number}, {"SalesDate", type date}})". The "AdventureWorksSales" query contains 20 rows of sales data, and the "OtherSales" query contains 25 rows of sales data. Both datasets include columns for SalesOrderID, SalesOrderDetailID, OrderQty, ProductName, UnitPrice, UnitPriceDiscount, LineTotal, and SalesDate.

SalesOrderID	SalesOrderDetailID	OrderQty	ProductName	UnitPrice	UnitPriceDiscount	LineTotal	SalesDate
75084	121222	1	Hitch Rack - 4-Bike	120	0	12000000	06/30/2014
75085	121223	1	AWC Logo Cap	8.99	0	8990000	06/30/2014
75085	121224	1	Bike Wash - Dissolver	7.95	0	7950000	06/30/2014
75086	121225	1	Bike Wash - Dissolver	7.95	0	7950000	06/30/2014
75087	121226	1	Sport-100 Helmet, Red	34.99	0	34990000	06/30/2014
75088	121227	1	Road Tire Tube	3.99	0	3990000	06/30/2014
75088	121228	1	Mt. Road Tire	24.99	0	24990000	06/30/2014
75088	121229	1	Sport-100 Helmet, Blue	34.99	0	34990000	06/30/2014
75088	121230	1	Long-Sleeve Logo Jersey, S	49.99	0	49990000	06/30/2014
75089	121231	1	LL Road Tire	21.49	0	21490000	06/30/2014
75089	121232	1	Road Tire Tube	3.99	0	3990000	06/30/2014
75089	121233	1	Sport-100 Helmet, Blue	34.99	0	34990000	06/30/2014
75090	121234	1	Mountain Tire Tube	4.99	0	4990000	06/30/2014
75090	121235	1	Mt. Mountain Tire	.35	0	3500000	06/30/2014
75090	121236	1	Sport-100 Helmet, Black	34.99	0	34990000	06/30/2014
75091	121237	1	LL Road Tire	21.49	0	21490000	06/30/2014
75091	121238	1	Road Tire Tube	3.99	0	3990000	06/30/2014
75091	121239	1	Hydration Pack - 70 oz.	54.99	0	54990000	06/30/2014
75092	121240	1	LL Road Tire	21.49	0	21490000	06/30/2014
75092	121241	1	Half-Finger Gloves, M	24.49	0	24490000	06/30/2014

The screenshot shows the Power BI Power Query Editor interface with the "AdventureWorksSales" query selected. A transformation step is visible, titled "Table.TransformColumnTypes(#'Promoted Headers', {{"SalesOrderID", Int64.Type}, {"SalesOrderDetailID", Int64.Type}, {"Quantity", Int64.Type}, {"Name", type text}, {"UnitPrice", type number}, {"Total", type number}, {"SalesDate", type date}})". The transformed "AdventureWorksSales" query contains 25 rows of sales data, matching the structure of the "OtherSales" query. It includes columns for SalesOrderID, SalesOrderDetailID, Quantity, Name, UnitPrice, Total, and SalesDate.

SalesOrderID	SalesOrderDetailID	Quantity	Name	UnitPrice	Total	SalesDate
101000	1	1	Hitch Rack - 4-Bike	120	0	12000000
101000	2	1	Women's Mountain Shorts, S	69.99	0	69990000
101000	3	1	Classic Vest, L	63.5	0	63500000
101001	4	1	Women's Mountain Shorts, L	69.99	0	69980000
101001	5	1	Long-Sleeve Logo Jersey, L	49.99	0	49980000
101002	6	1	Water Bottle - 30 oz.	4.99	0	4990000
101003	7	1	Sport-100 Helmet, Black	34.99	0	34990000
101003	8	1	Water Bottle - 30 oz.	4.99	0	4990000
101003	9	1	Bike Wash - Dissolver	7.95	0	7950000
101003	10	1	Mountain Tire Tube	4.99	0	4990000
101004	11	1	Sport-100 Helmet, Red	34.99	0	34990000
101005	12	1	AWC Logo Cap	8.99	0	8990000
101005	13	1	Long-Sleeve Logo Jersey, L	49.99	0	49990000
101005	14	1	Mountain Tire Tube	4.99	0	4990000
101005	15	1	Patch Kit/Patches	2.29	0	2290000
101005	16	1	Fender Set - Mountain	21.98	0	21980000
101005	17	1	Sport-100 Helmet, Black	34.99	0	34990000
101005	18	1	Mountain Tire Tube	4.99	0	4990000
101005	19	1	Fender Set - Mountain	21.98	0	21980000
101005	20	1	Classic Vest, S	63.5	0	63500000
101005	21	1	Fender Set - Mountain	21.98	0	21980000
101005	22	1	Hydration Pack - 70 oz.	54.99	0	54990000
101005	23	1	Bike Wash - Dissolver	7.95	0	7950000
101006	24	1	Mountain Tire Tube	4.99	0	4990000
101006	25	1	Sport-100 Helmet, Black	34.99	0	34990000

Step 2: Open the Power Query Editor

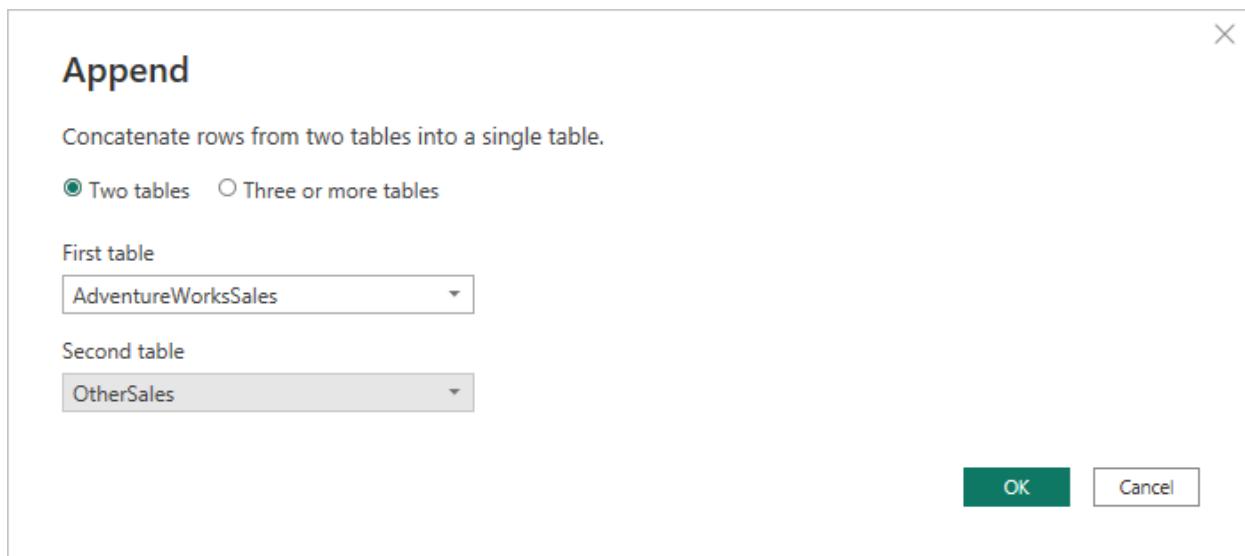
- Open the Power Query editor and import your datasets, *AdventureWorksSales* and *OtherSales*.

Step 3: Format Excel files

- You have to append *OtherSales* data to *AdventureWorksSales* data. So, you will use *AdventureWorksSales* data as the first table and *OtherSales* data as second table.
- For this reason, format the *OtherSales* data and rename the column names, using the *AdventureWorksSales* data, for example, Quantity to OrderQty, Name to ProductName, and Total to LineTotal.

Step 4: Append queries

- Append queries in a new master table. In the newly created query, check the column names, row number and the values appended. Make sure that the operation has been completed successfully.



Step 5: Rename new query

- In the left menu, select the new query and change its name to *Consolidated Sales* and select Enter on the right pane, named Properties.

Conclusion

You have now successfully combined your datasets using Power Query. The data sets are now consolidated in one file, ready for further analysis in Power BI Desktop.

Exemplar: Appending two tables

Overview

In the exercise *Appending two tables*, you put into practice your understanding of how to combine a data set by importing, formatting, and appending data in Power Query.

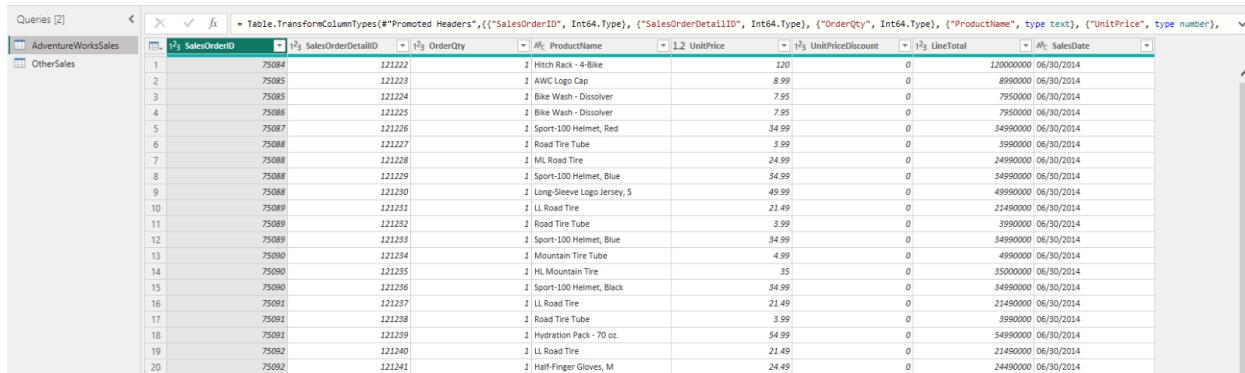
Your objective for this exercise was to prepare a worksheet for analysis by completing the following tasks:

- Download Excel Files
- Open Power Query Editor
- Format Excel Files
- Append Queries
- Rename the New Query

This reading provides a step-by-step guide for completing these tasks, accompanied by screenshots for easy comparison with your own copy.

Step 1: Download Excel files

- Download *AdventureWorksSales.xlsx* and *OtherSales.xlsx* files, which will be used in this exercise.



The screenshot shows the Power Query Editor interface with two queries listed in the 'Queries [2]' pane: 'AdventureWorksSales' and 'OtherSales'. The 'AdventureWorksSales' query is currently selected and displays a table with 20 rows of data. The columns are: SalesOrderID, SalesOrderDetailID, OrderQty, ProductName, UnitPrice, UnitPriceDiscount, LineTotal, and SalesDate. The data includes various products like Hitch Rack - 4-Bike, AWC Logo Cap, Bike Wash - Dissolver, Sport-100 Helmet, Red, Road Tire Tube, MT Road Tire, Sport-100 Helmet, Blue, Long-Sleeve Logo Jersey, S, LL Road Tire, Road Tire Tube, Sport-100 Helmet, Blue, Mountain Tire Tube, HL Mountain Tire, Sport-100 Helmet, Black, LL Road Tire, Road Tire Tube, Hydration Pack - 70 oz., LL Road Tire, and Half-Finger Gloves, M. The 'SalesDate' column shows dates from 06/30/2014. The 'OtherSales' query is also visible in the pane.

SalesOrderID	SalesOrderDetailID	OrderQty	ProductName	UnitPrice	UnitPriceDiscount	LineTotal	SalesDate
1	75084	121222	Hitch Rack - 4-Bike	120	0	12000000	06/30/2014
2	75085	121223	AWC Logo Cap	8.99	0	899000	06/30/2014
3	75085	121224	Bike Wash - Dissolver	7.95	0	795000	06/30/2014
4	75086	121225	Bike Wash - Dissolver	7.95	0	795000	06/30/2014
5	75087	121226	Sport-100 Helmet, Red	34.99	0	3499000	06/30/2014
6	75088	121227	Road Tire Tube	3.99	0	399000	06/30/2014
7	75088	121228	MTL Road Tire	24.99	0	2499000	06/30/2014
8	75088	121229	Sport-100 Helmet, Blue	34.99	0	3499000	06/30/2014
9	75088	121230	Long-Sleeve Logo Jersey, S	49.99	0	4999000	06/30/2014
10	75089	121231	LL Road Tire	21.49	0	2149000	06/30/2014
11	75089	121232	Road Tire Tube	3.99	0	399000	06/30/2014
12	75089	121233	Sport-100 Helmet, Blue	34.99	0	3499000	06/30/2014
13	75090	121234	Mountain Tire Tube	4.99	0	499000	06/30/2014
14	75090	121235	HL Mountain Tire	35	0	3500000	06/30/2014
15	75090	121236	Sport-100 Helmet, Black	34.99	0	3499000	06/30/2014
16	75091	121237	LL Road Tire	21.49	0	2149000	06/30/2014
17	75091	121238	Road Tire Tube	3.99	0	399000	06/30/2014
18	75091	121239	Hydration Pack - 70 oz.	54.99	0	5499000	06/30/2014
19	75092	121240	LL Road Tire	21.49	0	2149000	06/30/2014
20	75092	121241	Half-Finger Gloves, M	24.49	0	2449000	06/30/2014

Queries [2]

AdventureWorksSales

OtherSales

	SalesOrderID	SalesOrderDetailID	Quantity	Name	UnitPrice	UnitPriceDiscount	Total	SaleDate
1	101000	1	2	Hitch Rack - 4-Bike	120	0	12000000	06/20/2014
2	101000	2	1	Women's Mountain Shorts, S	69.99	0	69990000	06/20/2014
3	101000	3	1	Classic Vent, L	63.5	0	63500000	06/20/2014
4	101001	4	1	Women's Mountain Shorts, L	69.99	0	69990000	06/21/2014
5	101001	5	1	Long-Sleeve Logo Jersey, L	49.99	0	49990000	06/21/2014
6	101002	6	1	Water Bottle - 30 oz.	4.99	0	49990000	06/22/2014
7	101003	7	1	Sport-100 Helmet, Black	34.99	0	34990000	06/23/2014
8	101003	8	1	Water Bottle - 30 oz.	4.99	0	49990000	06/23/2014
9	101003	9	1	Bike Wash - Dissolver	7.95	0	7950000	06/23/2014
10	101003	10	1	Mountain Tire Tube	4.99	0	49990000	06/24/2014
11	101004	11	1	Sport-100 Helmet, Red	34.99	0	34990000	06/24/2014
12	101005	12	1	AWC Logo Cap	8.99	0	8990000	06/25/2014
13	101005	13	1	Long-Sleeve Logo Jersey, L	49.99	0	49990000	06/25/2014
14	101005	14	1	Mountain Tire Tube	4.99	0	49990000	06/25/2014
15	101005	15	1	Patch Kit/8 Patches	2.29	0	2290000	06/25/2014
16	101005	16	1	Fender Set - Mountain	21.98	0	21980000	06/25/2014
17	101005	17	1	Sport-100 Helmet, Black	34.99	0	34990000	06/25/2014
18	101005	18	1	Mountain Tire Tube	4.99	0	49990000	06/25/2014
19	101005	19	1	Fender Set - Mountain	21.98	0	21980000	06/25/2014
20	101005	20	1	Classic Vent, S	63.5	0	63500000	06/25/2014
21	101005	21	1	Fender Set - Mountain	21.98	0	21980000	06/25/2014
22	101005	22	1	Hydration Pack - 70 oz.	54.99	0	54990000	06/25/2014
23	101005	23	1	Bike Wash - Dissolver	7.95	0	7950000	06/25/2014
24	101006	24	1	Mountain Tire Tube	4.99	0	49990000	06/26/2014
25	101006	25	1	Sport-100 Helmet, Black	34.99	0	34990000	06/26/2014

Step 2: Open the Power Query Editor

Open the Power Query editor and import your datasets, *AdventureWorksSales* and *OtherSales*.

1. Navigate to the Home ribbon tab at the top of the Power BI window.
2. Select the Excel Workbook button inside the Data group, in the middle of the toolbar.
3. Select *AdventureWorksSales.xlsx* and *OtherSales.xlsx* files respectively and select Transform Data in the opened window.

Navigator

Display Options ▾

AdventureWorksSales

SalesOrderID	SalesOrderDetailID	OrderQty	ProductName	UnitPrice	UnitPriceDiscount	LineTotal
75084	121222	1	Hitch Rack - 4-Bike	120	0	0
75085	121223	1	AWC Logo Cap	8.99	0	0
75085	121224	1	Bike Wash - Dissolver	7.95	0	0
75086	121225	1	Bike Wash - Dissolver	7.95	0	0
75087	121226	1	Sport-100 Helmet, Red	34.99	0	0
75088	121227	1	Road Tire Tube	3.99	0	0
75088	121228	1	ML Road Tire	24.99	0	0
75088	121229	1	Sport-100 Helmet, Blue	34.99	0	0
75088	121230	1	Long-Sleeve Logo Jersey, S	49.99	0	0
75089	121231	1	LL Road Tire	21.49	0	0
75089	121232	1	Road Tire Tube	3.99	0	0
75089	121233	1	Sport-100 Helmet, Blue	34.99	0	0
75090	121234	1	Mountain Tire Tube	4.99	0	0
75090	121235	1	HL Mountain Tire	35	0	0
75090	121236	1	Sport-100 Helmet, Black	34.99	0	0
75091	121237	1	LL Road Tire	21.49	0	0
75091	121238	1	Road Tire Tube	3.99	0	0
75091	121239	1	Hydration Pack - 70 oz.	54.99	0	0
75092	121240	1	LL Road Tire	21.49	0	0
75092	121241	1	Half-Finger Gloves, M	24.49	0	0
75092	121242	1	Road Tire Tube	3.99	0	0
75093	121243	1	Road Tire Tube	3.99	0	0
75093	121244	1	HL Road Tire	32.6	0	0

Load Transform Data Cancel

Navigator

Display Options ▾

OtherSales

SalesOrderID	SalesOrderDetailID	Quantity	Name	UnitPrice	UnitPriceDiscount
101000	1	1	Hitch Rack - 4-Bike	120	0
101000	2	1	Women's Mountain Shorts, S	69.99	0
101000	3	1	Classic Vest, L	63.5	0
101001	4	1	Women's Mountain Shorts, L	69.99	0
101001	5	1	Long-Sleeve Logo Jersey, L	49.99	0
101002	6	1	Water Bottle - 30 oz.	4.99	0
101003	7	1	Sport-100 Helmet, Black	34.99	0
101003	8	1	Water Bottle - 30 oz.	4.99	0
101003	9	1	Bike Wash - Dissolver	7.95	0
101003	10	1	Mountain Tire Tube	4.99	0
101004	11	1	Sport-100 Helmet, Red	34.99	0
101005	12	1	AWC Logo Cap	8.99	0
101005	13	1	Long-Sleeve Logo Jersey, L	49.99	0
101005	14	1	Mountain Tire Tube	4.99	0
101005	15	1	Patch Kit/8 Patches	2.29	0
101005	16	1	Fender Set - Mountain	21.98	0
101005	17	1	Sport-100 Helmet, Black	34.99	0
101005	18	1	Mountain Tire Tube	4.99	0
101005	19	1	Fender Set - Mountain	21.98	0
101005	20	1	Classic Vest, S	63.5	0
101005	21	1	Fender Set - Mountain	21.98	0
101005	22	1	Hydration Pack - 70 oz.	54.99	0
101005	23	1	Bike Wash - Dissolver	7.95	0

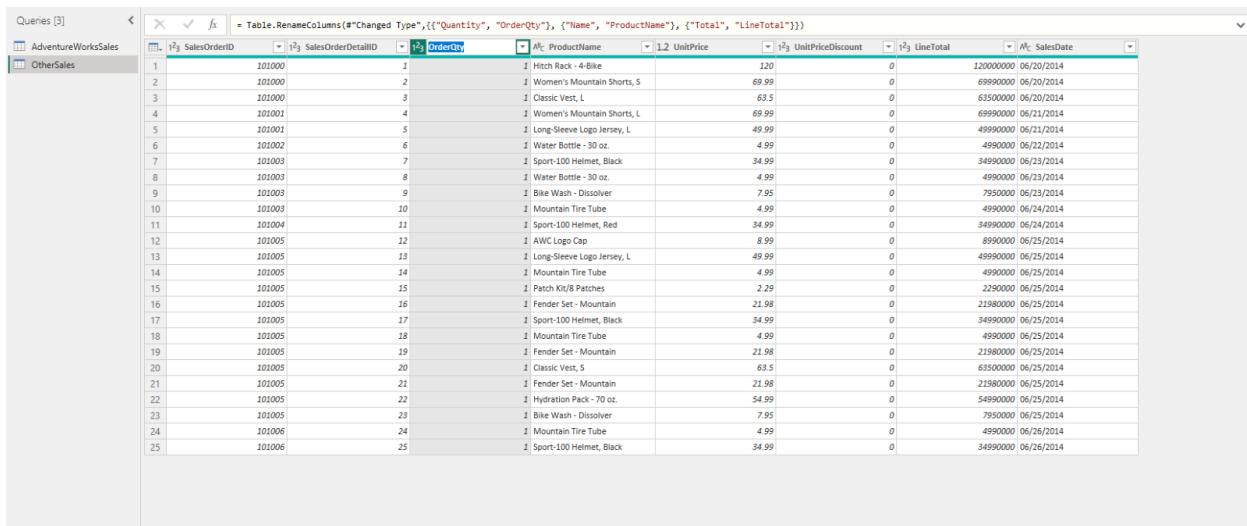
Load Transform Data Cancel

1. The Power Query Editor window opens. You can now begin formatting the data.

Step 3: Format Excel files

You will append *OtherSales* data to *AdventureWorksSales* data. So, you will use *AdventureWorks* sales data as the first table and *OtherSales* data as the second table. For this reason, format *OtherSales* data and rename the column names to the same names as the column names in *AdventureWorksSales* data.

1. Select the *OtherSales* query in the Query pane at the left menu of the Power Query window.
2. Rename the Quantity column to OrderQty by selecting the column.



The screenshot shows the Power Query Editor interface with two queries listed in the Queries [3] pane: "AdventureWorksSales" and "OtherSales". The "OtherSales" query is currently selected. The main area displays the data from the "OtherSales" table, which has 25 rows and 7 columns. The columns are labeled: SalesOrderID, SalesOrderDetailID, OrderQty, ProductName, UnitPrice, UnitPriceDiscount, and LineTotal. The data consists of various product details and their prices. A formula bar at the top indicates the transformation: `= Table.RenameColumns(#"Changed Type",{{{"Quantity", "OrderQty"}, {"Name", "ProductName"}, {"Total", "LineTotal"}}})`.

1. Repeat the same changing operation for the column Name to ProductName.
2. Change the column Total to LineTotal.

Queries [2]

AdventureWorksSales

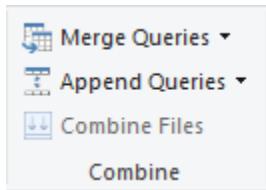
OtherSales

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductName	UnitPrice	UnitPriceDiscount	LineTotal	Sale
1	101000	1	1	Hitch Rack - 4-Bike	120	0	12000000	06/20/2
2	101000	2	2	Women's Mountain Shorts, S	69.99	0	69990000	06/20/2
3	101000	3	1	Classic Vest, L	63.5	0	63500000	06/20/2
4	101001	4	1	Women's Mountain Shorts, L	69.99	0	69990000	06/21/2
5	101001	5	1	Long-Sleeve Logo Jersey, L	49.99	0	49990000	06/21/2
6	101002	6	1	Water Bottle - 30 oz.	4.99	0	49900000	06/22/2
7	101003	7	1	Sport-100 Helmet, Black	34.99	0	34990000	06/23/2
8	101003	8	1	Water Bottle - 30 oz.	4.99	0	49900000	06/23/2
9	101003	9	1	Bike Wash - Dissolver	7.95	0	79500000	06/23/2
10	101003	10	1	Mountain Tire Tube	4.99	0	49900000	06/24/2
11	101004	11	1	Sport-100 Helmet, Red	34.99	0	34990000	06/24/2
12	101005	12	1	AWC Logo Cap	8.99	0	89900000	06/25/2
13	101005	13	1	Long-Sleeve Logo Jersey, L	49.99	0	49990000	06/25/2
14	101005	14	1	Mountain Tire Tube	4.99	0	49900000	06/25/2
15	101005	15	1	Patch Kit/8 Patches	2.29	0	22900000	06/25/2
16	101005	16	1	Fender Set - Mountain	21.98	0	21980000	06/25/2
17	101005	17	1	Sport-100 Helmet, Black	34.99	0	34990000	06/25/2
18	101005	18	1	Mountain Tire Tube	4.99	0	49900000	06/25/2
19	101005	19	1	Fender Set - Mountain	21.98	0	21980000	06/25/2
20	101005	20	1	Classic Vest, S	63.5	0	63500000	06/25/2
21	101005	21	1	Fender Set - Mountain	21.98	0	21980000	06/25/2
22	101005	22	1	Hydration Pack - 70 oz.	54.99	0	54990000	06/25/2
23	101005	23	1	Bike Wash - Dissolver	7.95	0	79500000	06/25/2

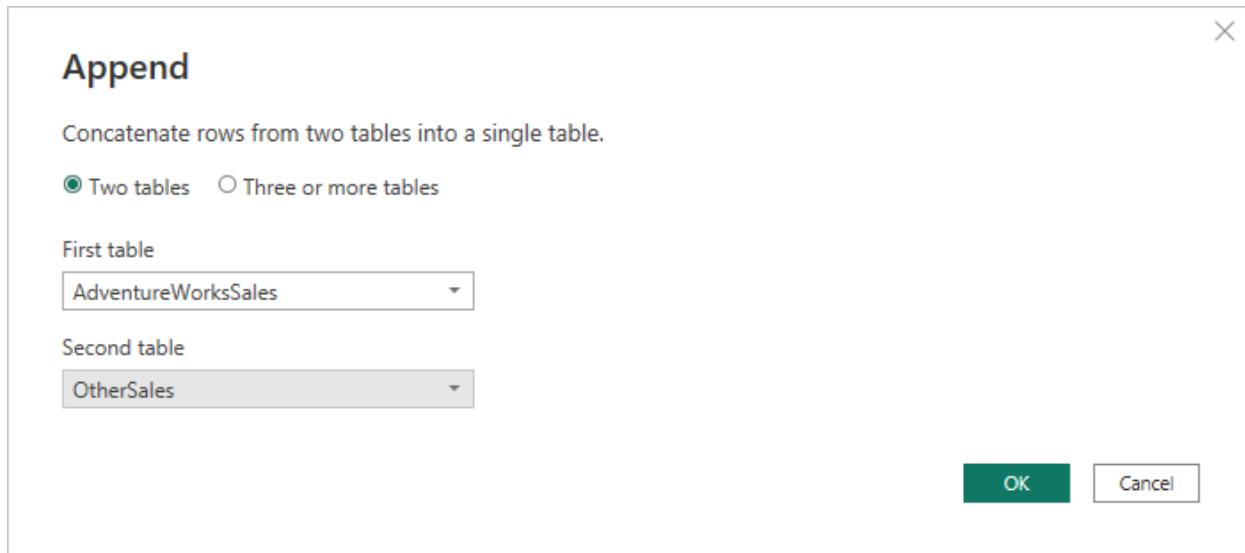
Step 4: Append queries

Append queries in a new master table.

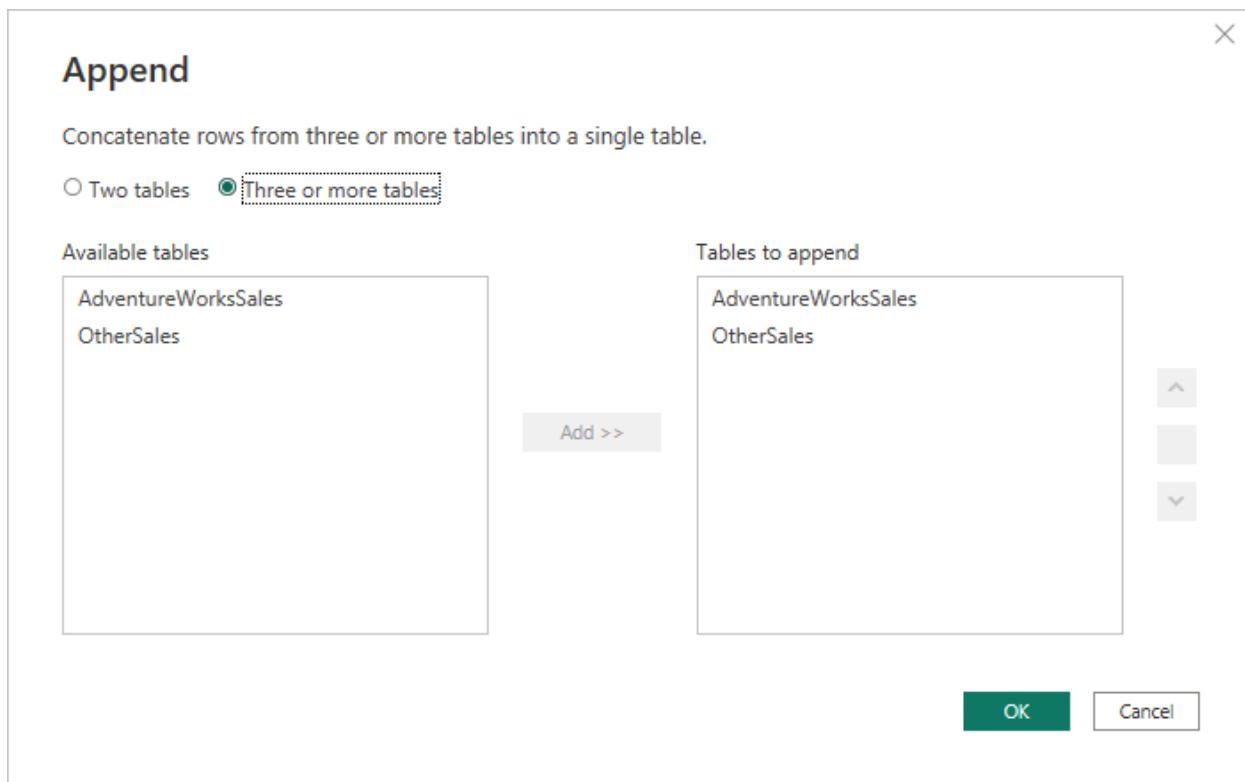
1. Select the Append Queries dropdown menu inside the Combine group, in the Home ribbon bar.



1. Select Append Queries as New from the Append Queries dropdown menu.
2. Select the Two Tables radio button. Select AdventureWorksSales as the first table and OtherSales as the second table.



1. Alternatively, you can select the Three or more tables radio button and use the drag and drop functionality. You can also select the tables you want from Available Tables and add them to Tables to Append.



1. Find the new query named *Append1* on the left menu Queries pane.

- Check the column names, row numbers, and the format of the appended query.
- Verify that there are 8 columns and 124 rows at the bottom left in the status bar.

37
38
39

8 COLUMNS, 124 ROWS Column profiling based on top 1000 rows

Step 5: Rename new query

Change the name of the new query to *Consolidated Sales*.

- In the left menu, select the new query and change its name by typing its name as *Consolidated Sales* and press Enter on the right pane, named Properties.

Queries [3]

- AdventureWorksSales
- OtherSales
- Consolidated Sales**

= Table.Combine({AdventureWorksSales, OtherSales})

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductName	UnitPrice
1	75084	121222	1	Hitch Rack - 4-Bike	1.2
2	75085	121223	1	AWC Logo Cap	
3	75085	121224	1	Bike Wash - Dissolver	
4	75086	121225	1	Bike Wash - Dissolver	
5	75087	121226	1	Sport-100 Helmet, Red	
6	75088	121227	1	Road Tire Tube	
7	75088	121228	1	ML Road Tire	
8	75088	121229	1	Sport-100 Helmet, Blue	
9	75088	121230	1	Long-Sleeve Logo Jersey, S	
10	75089	121231	1	LL Road Tire	
11	75089	121232	1	Road Tire Tube	
12	75089	121233	1	Sport-100 Helmet, Blue	
13	75090	121234	1	Mountain Tire Tube	
14	75090	121235	1	HL Mountain Tire	
15	75090	121236	1	Sport-100 Helmet, Black	
16	75091	121237	1	LL Road Tire	
17	75091	121238	1	Road Tire Tube	
18	75091	121239	1	Hydration Pack - 70 oz.	
19	75092	121240	1	LL Road Tire	
20	75092	121241	1	Half-Finger Gloves, M	
21	75092	121242	1	Road Tire Tube	
22	75093	121243	1	Road Tire Tube	
23	75093	121244	1	HL Road Tire	

11 COLUMNS, 124 ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED ON TUESDAY 16 MAY 2023

Conclusion

Your objective for this exercise was to apply techniques for importing, formatting, and combining data. In this context, you learned how to use Power Query Editor to import more than one data source, change the names or formats of the columns when needed before appending, append tables and queries, and create a new combined master table with a given name.

2.4. Exercise: Merging two data sources

Introduction

By now, you should understand the concept of data merging and its practical applications. During this week you've learned about the concepts of join types, what a join key is and where it can be used. Remember, joins are one of the most basic and important operations in Microsoft Power BI and generally in all database systems.

In this exercise, you'll have the opportunity to apply your knowledge by using Power Query to join two data sources; explore the relationships between two tables and match the related data.

Case study

Adventure Works has data with details, such as Sales Order ID, Order Date, Product Key, Quantity, Unit Price, Reseller Key, Product, Reseller and some other fields are stored in separate tables. These tables have a relationship with the *Sales* table because of the relational data architecture.

As you have learned before, although this relational data structure provides data integrity, it only shows the key values. To list the other details of the related tables, you must merge tables with joins.

Your manager, Adio Quinn, has assigned a task to you to list the *Sales* data with the Sales Order ID, Order Date, Product Name, Quantity and Unit Price columns. To do this, you must merge the *Sales* and *Product* tables by creating a Power BI query that merges the data.

Adio sends you the two .xlsx files containing the Adventure Works company sales data named *Sales.xlsx*, and product data named *Product.xlsx*. The data sets have a common column named ProductKey. You will use the common column and match the columns. You need to read data from the two tables.

1. This exercise aims to assist you in understanding how to combine data by merging data.
 2. By the end of this exercise, you'll understand how to merge data in Power Query, to explore the relationships between two tables and match the related data.

Resources

There are two resources for you to use in this exercise:

Instructions

- Create a new Power BI project called *Exercise – Merging two tables*. Follow the prompts below to complete the exercise.

Step 1: Download the Excel files

- Download the *Sales.xlsx* and *Product.xlsx* files, which will be used in this exercise.

SalesOrderNumber	OrderDate	ProductKey	ResellerKey	EmployeeKey	SalesTerritoryKey	Quantity	Unit Price	Sales	Cost
SO43897F	August 25 2017	23531228242528.84\$57.68\$63.45							
SO43897F	August 25 2017	3513024.99\$4	049.98\$3	796.19					
SO43897F	August 25 2017	3483024.99\$4	049.98\$3	796.19					
SO43897F	August 25 2017	23231228242528.84\$57.68\$63.45							
SO44544S	November	2017	2923637.4\$1	413.62					
SO44544S	November	2017	22031228242520.19\$40.38\$24.06						
SO44544S	November	2017	3513024.99\$4	049.98\$3	796.19				
SO44544S	November	2017	3493024.99\$4	049.98\$3	796.19				
SO44544S	November	2017	3443039.99\$4	079.98\$3	824.31				
SO45321S	February	2018	3463039.99\$4	079.98\$3	824.31				
SO45321S	February	2018	3473039.99\$4	079.98\$3	824.31				
SO46082V	May 23	2018	22031228242520.19\$40.38\$24.06						
SO46082V	May 23	2018	3463039.99\$4	079.98\$3	824.31				
SO46082V	May 23	2018	3453039.99\$4	079.98\$3	824.31				
SO46082V	May 23	2018	23231228242528.84\$57.68\$63.45						
SO46082V	May 23	2018	3443039.99\$4	079.98\$3	824.31				
SO46082V	May 23	2018	3483024.99\$4	049.98\$3	796.19				
SO46082V	May 23	2018	21231228242520.19\$40.38\$24.06						
SO47028F	August 24	2018	41031228242536.45\$72.95\$33.94						
SO47028F	August 24	2018	46431228242514.13\$28.26\$19.43						
SO47028F	August 24	2018	412312282425180.13\$360.26\$266.59						
SO47028F	August 24	2018	420312282425141.62\$283.24\$209.59						

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	ProductKey	ProductStandardCost	Color	Subcategory	Category	Background Color Format	Font Color Format												
2	210HL Roa	585868.63	Black	Road Frames	Components	#000000#FFFFFF													
3	215Sport-	Black\$12.03	Black	Helmets	Accessories	#000000#FFFFFF													
4	216Sport-	Black\$13.88	Black	Helmets	Accessories	#000000#FFFFFF													
5	217Sport-	Black\$13.09	Black	Helmets	Accessories	#000000#FFFFFF													
6	253LL Roa	585176.28	Black	Road Frames	Components	#000000#FFFFFF													
7	254LL Roa	58\$170.14	Black	Road Frames	Components	#000000#FFFFFF													
8	255LL Roa	585204.63	Black	Road Frames	Components	#000000#FFFFFF													
9	256LL Roa	60\$176.28	Black	Road Frames	Components	#000000#FFFFFF													
10	257LL Roa	60\$170.14	Black	Road Frames	Components	#000000#FFFFFF													
11	258LL Roa	605204.63	Black	Road Frames	Components	#000000#FFFFFF													
12	259LL Roa	625176.28	Black	Road Frames	Components	#000000#FFFFFF													
13	260LL Roa	62\$170.14	Black	Road Frames	Components	#000000#FFFFFF													
14	261LL Roa	625204.63	Black	Road Frames	Components	#000000#FFFFFF													
15	279LL Roa	445176.28	Black	Road Frames	Components	#000000#FFFFFF													
16	280LL Roa	44\$170.14	Black	Road Frames	Components	#000000#FFFFFF													
17	281LL Roa	445204.63	Black	Road Frames	Components	#000000#FFFFFF													
18	282LL Roa	485176.28	Black	Road Frames	Components	#000000#FFFFFF													
19	283LL Roa	48\$170.14	Black	Road Frames	Components	#000000#FFFFFF													
20	284LL Roa	485204.63	Black	Road Frames	Components	#000000#FFFFFF													
21	285LL Roa	525176.28	Black	Road Frames	Components	#000000#FFFFFF													
22	286LL Roa	525170.14	Black	Road Frames	Components	#000000#FFFFFF													
23	287LL Roa	525204.63	Black	Road Frames	Components	#000000#FFFFFF													

Step 2: Create a Power BI project

1. Create a Power BI project and open the Power Query editor.
2. Create a new Power BI project called *Exercise – Merging two tables*.

Step 3: Open the Power Query Editor

- Open the Power Query editor and import your datasets, *Sales* and *Product*.

Step 4: Merge queries

1. After selecting the *Sales* data in the Queries pane, select Merge Queries.
2. In the opened window, the *Sales* table will be shown automatically in the upper section of the window.
3. Choose the next table for merging, which is *Product*.
4. *ProductKey* is the common column between the tables, so click on the *ProductKey* columns in each table.

5. For the Join Kind dropdown, choose the join type Left Outer Join, which selects all records from the left table and matching records from the right table.

Step 5: Select column(s) from Product

1. After you merged the tables, a new column, named Product is added to the right side of the *Sales* data. This allows you to choose columns from the *Product* table.
2. Select the column named Product from the *Product* table.

Step 6: Choose and reorder columns from Sales

1. After you add the new column, Product, it is added to the *Sales* query as Product.Product. You must rename this column as Product to avoid confusion.
2. Move the Product field from right to left.
3. Remove the unwanted columns, Product Key (name of product is added by merge, so you will not need the key value of product), Reseller Key, Employee Key and Sales Territory Key columns.
4. Reorder the final list as indicated in your task to Sales Order Number, Order Date, Product, Quantity and UnitPrice.

Conclusion

You have successfully merged your datasets using Power Query by following these steps. The datasets are now joined and ready for further analysis in Power BI Desktop. As mentioned earlier, you can explore the relationships between two tables and match their related data by doing this operation.

Exemplar: Merging two data sources

Overview

In the exercise *Merging two data sources*, you put into practice your understanding of how to merge two data sources and by importing, merging and choosing join kind in Power Query.

Your objective in this exercise was to merge data sources by completing the following tasks:

- Download CSV Files
- Open *Power Query Editor*
- Choose Queries
- Merge Queries
- Select Column(s) from *Product*
- Choose and reorder columns from *Sales*

This reading provides a step-by-step guide for completing these tasks, accompanied by screenshots for comparison with your own copy.

Step 1: Download CSV files

- Download *Sales.csv* and *Product.csv* files.

Sales.csv Data:

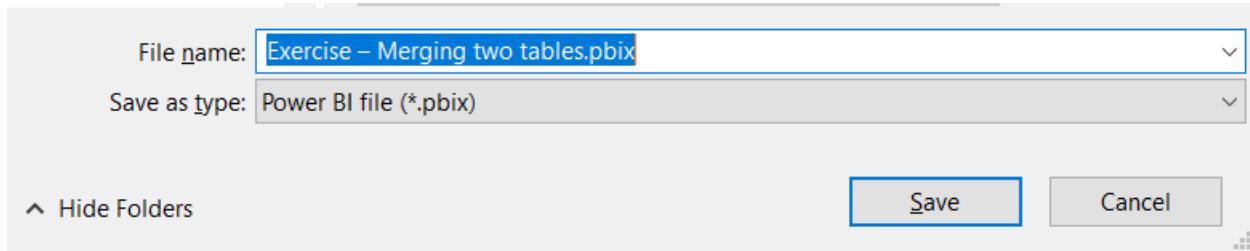
	SalesOrderNumber	OrderDate	ProductKey	ResellerKey	EmployeeKey	SalesTerritoryKey	Quantity	Unit Price	Sales	Cost
1	SO43897F	August 25 2017	23531228242528.84	57.68	\$63.45					
2	SO43897F	August 25 2017	3513024.99	54	049.98	53	796.19			
3	SO43897F	August 25 2017	3483024.99	54	049.98	53	796.19			
4	SO43897F	August 25 2017	23231228242528.84	57.68	\$63.45					
5	SO44544S	November 2017	2923637.45	1	413.62					
6	SO44544S	November 2017	22031228242520.19	40.38	\$24.06					
7	SO44544S	November 2017	3513024.99	54	049.98	53	796.19			
8	SO44544S	November 2017	3493024.99	54	049.98	53	796.19			
9	SO44544S	November 2017	3443039.99	54	079.98	53	824.31			
10	SO445321S	February 2018	3463039.99	54	079.98	53	824.31			
11	SO45321S	February 2018	3473039.99	54	079.98	53	824.31			
12	SO46082V	May 23	201822031228242520.19	40.38	\$24.06					
13	SO46082V	May 23	3463039.99	54	079.98	53	824.31			
14	SO46082V	May 23	3453039.99	54	079.98	53	824.31			
15	SO46082V	May 23	23231228242528.84	57.68	\$63.45					
16	SO46082V	May 23	3443039.99	54	079.98	53	824.31			
17	SO46082V	May 23	3443039.99	54	079.98	53	824.31			
18	SO46082V	May 23	3483024.99	54	049.98	53	796.19			
19	SO46082V	May 23	21231228242520.19	40.38	\$24.06					
20	SO47028F	August 24 2018	41031228242536.45	57.2	\$953.94					
21	SO47028F	August 24 2018	46431228242514.13	58.26	\$19.43					
22	SO47028F	August 24 2018	41231228242518.13	58.36	\$26.59					
23	SO47028F	August 24 2018	42031228242514.62	58.28	\$24.52	09.59				

Product.csv Data:

	ProductKey	StandardCost	Color	Subcategory	Category	Background Color	Format	Font	Color	Format
1	210HL.Ro	585868.63	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
2	215Sport.	Black\$12.03	Black	Helmets	Accessories	#000000	FFFF	FFF	FFF	FFF
3	216Sport.	Black\$13.88	Black	Helmets	Accessories	#000000	FFFF	FFF	FFF	FFF
4	217Sport.	Black\$13.09	Black	Helmets	Accessories	#000000	FFFF	FFF	FFF	FFF
5	253UL.Roa	585176.28	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
6	254LL.Roa	585170.14	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
7	255LL.Roa	585204.63	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
8	256LL.Roa	605176.28	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
9	257LL.Roa	605170.14	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
10	258LL.Roa	605204.63	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
11	259LL.Roa	625176.28	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
12	260LL.Roa	625170.14	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
13	261LL.Roa	625204.63	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
14	262LL.Roa	625204.63	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
15	279LL.Roa	445176.28	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
16	280LL.Roa	445170.14	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
17	281LL.Roa	445204.63	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
18	282LL.Roa	485176.28	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
19	283LL.Roa	485170.14	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
20	284LL.Roa	485204.63	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
21	285LL.Roa	525176.28	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
22	286LL.Roa	525170.14	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF
23	287LL.Roa	525204.63	Black	Road Frames	Components	#000000	FFFF	FFF	FFF	FFF

Step 2: Create a Power BI project

1. Create a Power BI project and open the Power Query editor.
2. Create a new Power BI project called *Exercise – Merging two tables*.



Step 3: Open the Power Query Editor

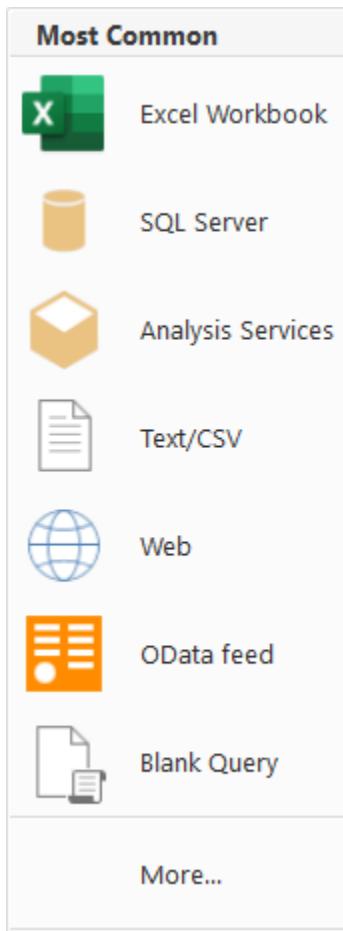
Import your datasets, *Sales* and *Product*.

1. Select the Text/CSV option from Get Data menu, in the Data group when the Home ribbon tab is selected.
2. Select the *Sales.csv* file and then select Transform Data in the window that opened.

A screenshot of the Microsoft Power Query Editor. The title bar says 'Sales .csv'. The interface includes 'File Origin' (1252: Western European (Windows)), 'Delimiter' (Tab), and 'Data Type Detection' (Based on first 200 rows). Below is a table with 19 columns: SalesOrderNumber, OrderDate, ProductKey, ResellerKey, EmployeeKey, SalesTerritoryKey, Quantity, Unit Price, Sales, and Cost. The table contains approximately 20 rows of sample data. At the bottom are buttons for 'Extract Table Using Examples', 'Load' (highlighted in green), 'Transform Data', and 'Cancel'.

3. The Power Query Editor window opens. You can now select the queries that you want to merge.

4. In Power Query Editor with the Home ribbon selected, select the New Source menu in the New Query group and choose Text/CSV.



5. Select *Product.csv* and then select OK.

Product.csv

File Origin: 1252: Western European (Windows) Delimiter: Tab Data Type Detection: Based on first 200 rows

ProductKey	Product	Standard Cost	Color	Subcategory	Category	Background Color Format	Font Color Format
210	HL Road Frame - Black, 58	\$868.63	Black	Road Frames	Components	#000000	#FFFFFF
215	Sport-100 Helmet, Black	\$12.03	Black	Helmets	Accessories	#000000	#FFFFFF
216	Sport-100 Helmet, Black	\$13.88	Black	Helmets	Accessories	#000000	#FFFFFF
217	Sport-100 Helmet, Black	\$13.09	Black	Helmets	Accessories	#000000	#FFFFFF
253	LL Road Frame - Black, 58	\$176.2	Black	Road Frames	Components	#000000	#FFFFFF
254	LL Road Frame - Black, 58	\$170.14	Black	Road Frames	Components	#000000	#FFFFFF
255	LL Road Frame - Black, 58	\$204.63	Black	Road Frames	Components	#000000	#FFFFFF
256	LL Road Frame - Black, 60	\$176.2	Black	Road Frames	Components	#000000	#FFFFFF
257	LL Road Frame - Black, 60	\$170.14	Black	Road Frames	Components	#000000	#FFFFFF
258	LL Road Frame - Black, 60	\$204.63	Black	Road Frames	Components	#000000	#FFFFFF
259	LL Road Frame - Black, 62	\$176.2	Black	Road Frames	Components	#000000	#FFFFFF
260	LL Road Frame - Black, 62	\$170.14	Black	Road Frames	Components	#000000	#FFFFFF
261	LL Road Frame - Black, 62	\$204.63	Black	Road Frames	Components	#000000	#FFFFFF
279	LL Road Frame - Black, 44	\$176.2	Black	Road Frames	Components	#000000	#FFFFFF
280	LL Road Frame - Black, 44	\$170.14	Black	Road Frames	Components	#000000	#FFFFFF
281	LL Road Frame - Black, 44	\$204.63	Black	Road Frames	Components	#000000	#FFFFFF
282	LL Road Frame - Black, 48	\$176.2	Black	Road Frames	Components	#000000	#FFFFFF
283	LL Road Frame - Black, 48	\$170.14	Black	Road Frames	Components	#000000	#FFFFFF
284	LL Road Frame - Black, 48	\$204.63	Black	Road Frames	Components	#000000	#FFFFFF
285	LL Road Frame - Black, 52	\$176.2	Black	Road Frames	Components	#000000	#FFFFFF

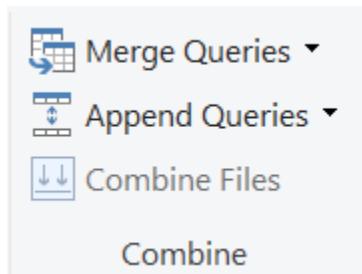
The data in the preview has been truncated due to size limits.

Extract Table Using Examples OK Cancel

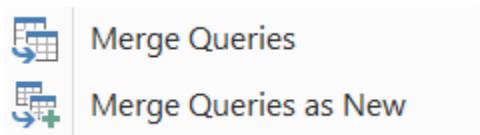
Step 4: Merge queries

Merge the *Sales* and *Product* queries, using Left Outer Join in the common column ProductKey.

- After selecting the Sales data in the Queries pane, under the Home ribbon, click Merge Queries in the Combine group.



2. Choose Merge Queries inside the Merge Queries dropdown menu.



3. In the opened window, the *Sales* table will be automatically shown in the upper parts choose *Product* in the

dropdown and click ProductKey columns in each table to mark the common column between the tables.

4. For the Join Kind dropdown, choose the Left Outer Join which selects all records from the left table and

matching records from the right table.

Merge

Select a table and matching columns to create a merged table.

Sales

SalesOrderNumber	OrderDate	ProductKey	ResellerKey	EmployeeKey	SalesTerritoryKey	Quantity	
SO43897	25/08/2017	235	312	282		4	2
SO43897	25/08/2017	351	312	282		4	2
SO43897	25/08/2017	348	312	282		4	2
SO43897	25/08/2017	232	312	282		4	2
SO43897	25/08/2017	232	312	282		4	2

Product

ProductKey	Product	Standard Cost	Color	Subcategory	Category	Background Color Form
210	HL Road Frame - Black, 58	\$868.63	Black	Road Frames	Components	#000000
215	Sport-100 Helmet, Black	\$12.03	Black	Helmets	Accessories	#000000
216	Sport-100 Helmet, Black	\$13.88	Black	Helmets	Accessories	#000000
217	Sport-100 Helmet, Black	\$13.09	Black	Helmets	Accessories	#000000

Join Kind

Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

> Fuzzy matching options

The selection matches 57851 of 57851 rows from the first table.

OK **Cancel**

Step 5: Select column(s) from Product

After you merge the tables, a new column, named *Product*, is added to the right side of the *Sales* table, which allows you to choose columns from the *Product* table.

- Find the column named *Product* at the most right of the table and click on the expand button to the right of the column.
- Choose the *Product* column by clicking on it and deselecting any other columns.
Note, the *Product* query and column have the same names, so make sure you don't get confused.

Sales

	Quantity	Avg Unit Price	Avg Sales	Cost
1	2	\$28.84	\$57.68	
2	2	\$2,024.99	\$4,049.98	
3	2	\$2,024.99	\$4,049.98	
4	2	\$28.84	\$57.68	
5	2	\$818.7	\$1,637.4	
6	2	\$20.19	\$40.38	
7	2	\$2,024.99	\$4,049.98	
8	2	\$2,024.99	\$4,049.98	
9	2	\$2,039.99	\$4,079.98	
10	2	\$2,039.99	\$4,079.98	
11	2	\$2,039.99	\$4,079.98	
12	2	\$20.19	\$40.38	
13	2	\$2,039.99	\$4,079.98	
14	2	\$2,039.99	\$4,079.98	
15	2	\$28.84	\$57.68	
16	2	\$2,039.99	\$4,079.98	\$3,824.31
17	2	\$2,024.99	\$4,049.98	\$3,796.19
18	2	\$20.19	\$40.38	\$24.06
19	2	\$36.45	\$72.9	\$53.94
20	2	\$14.13	\$28.26	\$19.43
21	2	\$180.13	\$360.26	\$266.59
22	2	\$141.62	\$283.24	\$209.59
23	2	\$222.70	\$445.58	\$311.34

Product

Properties

Query Settings

APPLIED STEPS

- Merged Queries

3. Rename Product.Product column to Product.

Sales

	Avg SalesOrderNumber	OrderDate	ProductKey	Avg Product	ResellerKey
1	SO43897	25/08/2017	235	Long Sleeve Logo Jersey, XL	
2	SO44113	23/09/2017	235	Long-Sleeve Logo Jersey, XL	
3	SO44520	13/11/2017	235	Long-Sleeve Logo Jersey, XL	
4	SO46057	15/05/2018	235	Long-Sleeve Logo Jersey, XL	
5	SO44115	23/09/2017	235	Long Sleeve Logo Jersey, XL	
6	SO43897	25/08/2017	351	Mountain-100 Black, 48	
7	SO44544	18/11/2017	351	Mountain-100 Black, 48	
8	SO43864	12/08/2017	351	Mountain-100 Black, 48	
9	SO44481	01/11/2017	351	Mountain-100 Black, 48	
10	SO44505	07/11/2017	351	Mountain-100 Black, 48	
11	SO44551	23/11/2017	351	Mountain-100 Black, 48	
12	SO44776	23/12/2017	215	Sport-100 Helmet, Black	
13	SO44521	13/11/2017	215	Sport-100 Helmet, Black	
14	SO43897	25/08/2017	348	Mountain-100 Black, 38	
15	SO46082	23/05/2018	348	Mountain-100 Black, 38	
16	SO44776	23/12/2017	348	Mountain-100 Black, 38	
17	SO43901	26/08/2017	348	Mountain-100 Black, 38	
18	SO45290	09/02/2018	348	Mountain-100 Black, 38	
19	SO43897	25/08/2017	232	Long-Sleeve Logo Jersey, L	
20	SO46082	23/05/2018	232	Long Sleeve Logo Jersey, L	
21	SO44776	23/12/2017	232	Long Sleeve Logo Jersey, L	
22	SO46362	18/06/2018	232	Long-Sleeve Logo Jersey, L	
23	SO46057	15/11/2018	232	Long-Sleeve Logo Jersey, L	

Product

Properties

Query Settings

APPLIED STEPS

- Reordered Columns

Step 6: Choose and reorder columns from Sales

Reorder the columns from the *Sales* table after merging them with *Product* table.

- Your manager asked you to show the name of the product, so you won't need the key anymore. Remove

ProductKey column by right clicking on the column and select Remove.

The screenshot shows the Power Query Editor interface with the 'Sales' query selected. A context menu is open over the 'ProductKey' column, with the 'Remove' option highlighted. The 'APPLIED STEPS' pane on the right shows the history of steps taken, including 'Renamed Columns'.

SalesOrderNumber	OrderDate	ProductKey	ResellerKey
SO43897	25/08/2017		
SO44113	23/09/2017		
SO44520	13/11/2017		
SO46057	15/05/2018		
SO44115	23/09/2017		
SO43897	25/08/2017		
SO44544	18/11/2017		
SO43864	12/08/2017		
SO44481	01/11/2017		
SO44505	07/11/2017		
SO44551	23/11/2017		
SO44776	23/12/2017		
SO44521	13/11/2017		
SO43897	25/08/2017		
SO46082	23/05/2018		
SO44776	23/12/2017		
SO43901	26/08/2017		
SO45290	09/02/2018		
SO43897	25/08/2017		
SO46082	23/05/2018		
SO44776	23/12/2017		
SO46362	18/06/2018		
SO44557	15/11/2018		

- Move the newly added Product column between OrderDate and Quantity columns by dragging and dropping them in place.
- Remove Reseller, Employee and Sales Territory Key columns from the query.
- Your final query will look like the image below after you have renamed, removed and reordered columns.

The screenshot shows the Microsoft Power Query Editor interface. In the center, there's a preview of a table named "Table.RemoveColumns(#"Renamed Columns", {"ProductKey"})". The table contains data from two sources: "Sales" and "Product". The "Sales" table has columns: SalesOrderNumber, OrderDate, Product, ResellerKey, and EmployeeKey. The "Product" table has columns: ProductID, Name, and Size. The preview shows the merged data with columns: SalesOrderNumber, OrderDate, ProductName, ResellerKey, EmployeeKey, and Size. On the right side, there's a "Query Settings" pane with sections for "PROPERTIES" (Name: Sales) and "APPLIED STEPS". The "APPLIED STEPS" section lists several steps: Source, Promoted Headers, Changed Type, Merged Queries, Expanded Product, Reordered Columns, Renamed Columns, and Removed Columns (which is currently selected). At the bottom of the editor, it says "10 COLUMNS" and "PREVIEW DOWNLOADED AT 15:40".

Conclusion

Your objective for this exercise was to apply techniques for importing, formatting, and merging data. In this context, you learned how to use Power Query Editor to import more than one data source, merge them by using the common columns, choose the columns from both tables and format the columns as needed.

You can refer to [Combine multiple tables into a single table](#) article on the *Microsoft Learn* site to review more details on the merging data processes in Power BI.

3.1. Exercise: Profiling a dataset

Introduction

By now, you should have a good understanding of the concept of profiling data and its practical applications to identify data anomalies. You covered the three primary data profiling operations, column quality, column distribution and column profile. In this exercise, you can apply your knowledge by using Microsoft Power Query to examine the valid, error, empty, min, max, unique, and distinct values in Excel spreadsheet rows, allowing you to identify the anomalies in the data.

Case study

Adventure Works has recently acquired another bicycle business. Adventure Works' CEO, Jamie Lee, has assigned a task to the sales department to ensure that the product data from the newly acquired company is validated, revised in quality factors, and ready for importing to the current company products. Your manager, Adio Quinn, has requested that you examine the data using Power Query by the factors of profile, quality, and distribution.

Adventure Works has provided you with an Excel file containing the newly acquired company's product data called *Other Company Products.xlsx*. The dataset has some empty data in its ProductKey column. You also need to assess the distribution of the products by the product categories and detect potential anomalies in the Price column. To complete your task successfully, you must import the Excel file, transform the data in Power Query and assess the Column quality, Column distribution, and Column profile options in the Data Preview group.

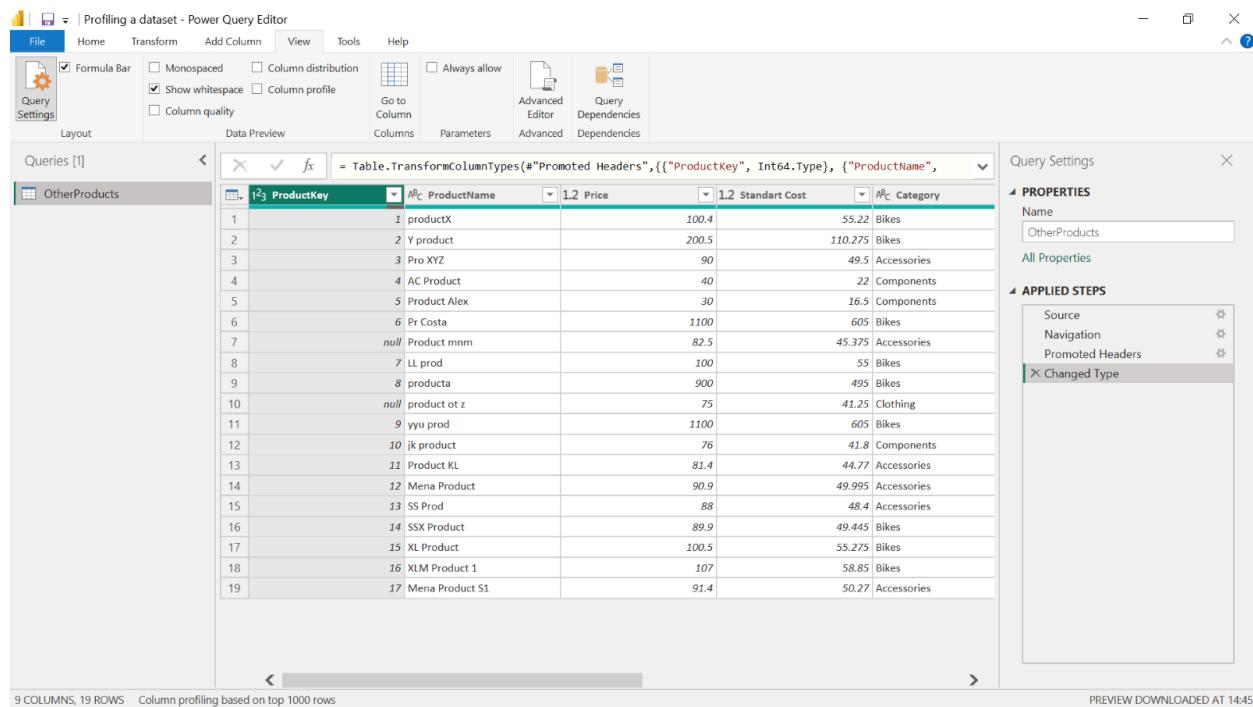
- This exercise aims to help you to understand how to identify data anomalies by profiling data.
- By the end of this exercise, you'll understand how to profile data in Power Query, and how to identify data anomalies.

Instructions

Create a new Power BI project called Exercise – Profiling a dataset. Follow the prompts below to complete the exercise.

Step 1: Download the Excel files

Download the *Other Company Products.xlsx* file, which you will use in this exercise. This file is available at the top of this exercise.



The screenshot shows the Power Query Editor interface. On the left, the 'Queries [1]' pane lists 'OtherProducts'. The main area displays a table with columns: ProductKey, ProductName, Price, Standard Cost, and Category. The 'Applied Steps' pane on the right shows a history of steps: 'Source', 'Navigation', 'Promoted Headers', and a highlighted 'Changed Type' step. The status bar at the bottom indicates '9 COLUMNS, 19 ROWS' and 'Column profiling based on top 1000 rows'.

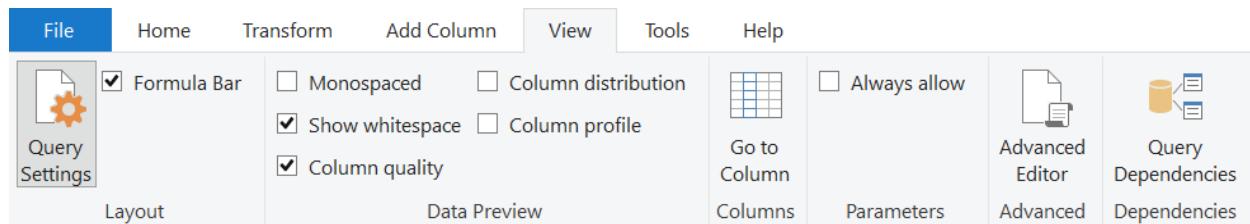
Step 2: Open the Power Query Editor

In Power BI, select Get Data and then select Transform data to open the Power Query editor and import your dataset, *Other Company Products*.

Step 3: Detect empty values in ProductKey column:

1. There are some empty values in the spreadsheet's ProductKey column.

- To detect empty and invalid values, you need to assess column quality, on the View ribbon tab, from inside the Data Preview group, check Column Quality. The column quality feature allows you to easily determine the percentage of valid, error, or empty values found in columns.



- Note amount Valid, Error and Empty rows percentage values of the ProductKey column

Step 4: Assess the distribution of product categories:

- There are many categories in Product list and you need to find out how the data is distributed by the category data.
- To assess column distribution, on the View ribbon tab, from inside the Data Preview group, check Column Distribution and note the amount of distinct values and unique values. Check Column Profile while keeping Column Distribution checkbox as checked and note the number of Bikes, Accessories, Components and Clothing.

Step 5: Detect potential anomalies in the Price column

- You have to assess Price column in the Product list and you need to find out min, max, mean values and the distribution of the values.
- To detect potential anomalies and assess column distribution for the Price column, on the View ribbon tab, from inside the Data Preview group, check Column Profile while keeping Column Distribution checkbox as checked.
- Note the min, max, mean values for the Price column and also note assess the column distribution.

Conclusion

Well done. You have now successfully assessed your datasets by the factors of profile, quality and distribution and detected potential anomalies using Power Query.

Exemplar: Profiling a dataset

Overview

In the exercise *Profiling a dataset*, you put into practice your understanding of how to profile a data set and detect potential anomalies in Power Query.

Your objective for this exercise was to prepare a worksheet for analysis by completing the following tasks:

- Download the Excel Files.
- Open the Power Query Editor.
- Detect empty values in the ProductKey column.
- Assess the distribution of product categories.
- Detect potential anomalies in the Price column.

This reading provides a step-by-step guide for completing these tasks, accompanied by screenshots for easy comparison with your own copy.

Step 1: Download Excel files

- Download the *Other Company Products.xlsx* file, which you will use in the exercise. The file is available at the top of this reading.

The screenshot shows the Power Query Editor interface. The ribbon at the top has tabs for File, Home, Transform, Add Column, View, Tools, and Help. The 'View' tab is selected. The toolbar below the ribbon includes options like Formula Bar, Monospaced, Column distribution, Show whitespace, Column profile, and Column quality. The 'Layout' group contains Query Settings, Go to Column, Advanced Editor, and Dependencies. The 'Data Preview' group contains Columns, Parameters, Advanced, and Dependencies.

The main area displays a table titled 'OtherProducts'. The table has 19 rows and 5 columns. The columns are: ProductKey, ProductName, Price, Standart Cost, and Category. The data includes various products like 'productX', 'Y product', 'Pro XYZ', etc., with their respective prices and categories like 'Bikes', 'Accessories', 'Components', 'Clothing', etc.

The 'Applied Steps' pane on the right shows the steps taken to transform the data. The 'Promoted Headers' step is highlighted. Other steps listed include 'Source', 'Navigation', and 'Changed Type'.

Step 2: Open the Power Query Editor

Open the Power Query editor and import your dataset, *Other Company Products*.

1. Navigate to the Home ribbon tab at the top of the Power BI window.
2. Select the Excel Workbook button in the Data group, in the middle of the toolbar.
3. Select *Other Company Products.xlsx* and select Transform Data in the opened window.

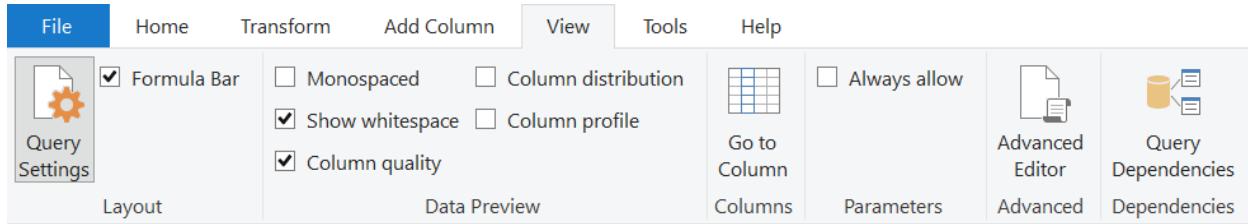
The screenshot shows the Microsoft Power Query Editor interface. On the left, the Navigator pane lists a connection named 'Other Company Products.xlsx [1]' with a table selected named 'OtherProducts'. Below it, there's a section for 'Suggested Tables [1]' which includes 'Table 1 (OtherProducts)'. The main area, titled 'OtherProducts', displays a data preview of 17 rows. The columns are labeled 'ProductKey', 'ProductName', 'Price', 'Standard Cost', 'Category', and 'Color'. The data includes various product names like 'productX', 'Y product', 'Pro XYZ', etc., with prices ranging from 200.5 to 1100. The 'Category' column shows categories such as 'Bikes', 'Components', 'Accessories', and 'Clothing'. The 'Color' column has entries like 'Black', 'White', 'Blue', and 'Yellow'. At the bottom right of the preview area are three buttons: 'Load', 'Transform Data', and 'Cancel'.

ProductKey	ProductName	Price	Standard Cost	Category	Color
1	productX	100.4	55.22	Bikes	Black
2	Y product	200.5	110.275	Bikes	White
3	Pro XYZ	90	49.5	Accessories	Blue
4	AC Product	40	22	Components	Blue
5	Product Alex	30	16.5	Components	White
6	Pr Costa	1100	605	Bikes	Black
null	Product mnm	82.5	45.375	Accessories	Black
7	LL prod	100	55	Bikes	Black
8	producta	900	495	Bikes	White
null	product ot z	75	41.25	Clothing	White
9	yyu prod	1100	605	Bikes	Blue
10	jk product	76	41.8	Components	Yellow
11	Product KL	81.4	44.77	Accessories	Blue
12	Mena Product	90.9	49.995	Accessories	Blue
13	SS Prod	88	48.4	Accessories	White
14	SSX Product	89.9	49.445	Bikes	Black
15	XL Product	100.5	55.275	Bikes	Black
16	XLM Product 1	107	58.85	Bikes	White
17	Mena Product S1	91.4	50.27	Accessories	Black

1. The Power Query Editor window opens. You can now begin profiling the data.

Step 3: Detect empty values in ProductKey column

1. There are empty values in ProductKey column.
2. To detect empty and invalid values, you need to assess column quality, on the View ribbon tab, from inside the Data Preview group, and select the Column Quality checkbox. The column quality feature allows you to easily determine the percentage of valid, error, or empty values found in columns.



1. Note that 89% of the values are Valid, 0% of the values are Error and 11% of the values are Empty rows in the ProductKey column.

	ProductKey	ProductName	Price	Standard Cost	Category
1	productX		100.4	55.22	Bikes
2	Y product		200.5	110.275	Bikes
3	Pro XYZ		90	49.5	Accessories
4	AC Product		40	22	Components
5	Product Alex		30	16.5	Components
6	Pr Costa		1100	605	Bikes
7	null Product mmm		82.5	45.375	Accessories
8	7 LL prod		100	55	Bikes
9	8 producta		900	495	Bikes
10	null product ot z		75	41.25	Clothing
11	9 yyu prod		1100	605	Bikes
12	10 jk product		76	41.8	Components
13	11 Product KL		81.4	44.77	Accessories
14	12 Mena Product		90.9	49.995	Accessories
15	13 SS Prod		88	48.4	Accessories
16	14 SSX Product		89.9	49.445	Bikes
17	15 XL Product		100.5	55.275	Bikes
18	16 XLM Product 1		107	58.85	Bikes
19	17 Mena Product S1		91.4	50.27	Accessories

Step 4: Assess the distribution of product categories

1. There are many categories in the Product list, and you need to find out how the data is distributed by the category data.
2. To assess column distribution, on the View ribbon tab, from inside the Data Preview group, check the Column Distribution checkbox. Note that there are 4 distinct values and 1 unique value.

The screenshot shows the Power Query Editor interface. The top navigation bar includes File, Home, Transform, Add Column, View, Tools, and Help. The View tab is selected, displaying the 'Query Settings' ribbon with options like Formula Bar, Monospaced, Column distribution, Show whitespace, Column profile, and Column quality. Below the ribbon are tabs for Layout, Data Preview, Columns, Parameters, Advanced, and Dependencies.

The main area displays a table titled 'OtherProducts' with 19 rows and 9 columns. The columns are: ProductKey, id, or, ipty, ProductName, 1.2 Price, 1.2 Standart Cost, 1.2 Category, and 1.2c. Each column has a histogram and a summary table below it. For example, the 'id' column has 89% Valid values, 0% Error values, and 11% Empty values. The 'ProductName' column has 19 distinct values and 19 unique values. The '1.2 Price' column has 18 distinct values and 17 unique values. The '1.2 Standart Cost' column has 18 distinct values and 17 unique values. The '1.2 Category' column has 4 distinct values and 1 unique value.

On the right side, there are two panes: 'Query Settings' and 'APPLIED STEPS'. The 'Query Settings' pane shows the query name 'OtherProducts' and all properties. The 'APPLIED STEPS' pane lists the steps taken: 'Source', 'Navigation', 'Promoted Headers', and 'Changed Type' (which is highlighted).

1. Check the Column Profile checkbox, while keeping Column Distribution checkbox as checked. Note that there are 9 Bikes, 6 Accessories, 3 Components and 1 Clothing categories.

Profiling a dataset - Power Query Editor

File Home Transform Add Column View Tools Help

Formula Bar Monospaced Column distribution Show whitespace Column profile Column quality

Layout Go to Column Columns Parameters Advanced Editor Advanced Dependencies Dependencies

Queries [1] OtherProducts

ProductKey ProductName Price Standard Cost Category

	ProductKey	ProductName	Price	Standard Cost	Category
1	productX		100.4	55.22	Bikes
2	Yproduct		200.5	110.275	Bikes
3					

Column statistics

Count	19
Error	0
Empty	0
Distinct	4
Unique	1
Empty string	0
Min	Accessori...
Max	Compon...

Value distribution

Query Settings

PROPERTIES

- Name: OtherProducts
- All Properties

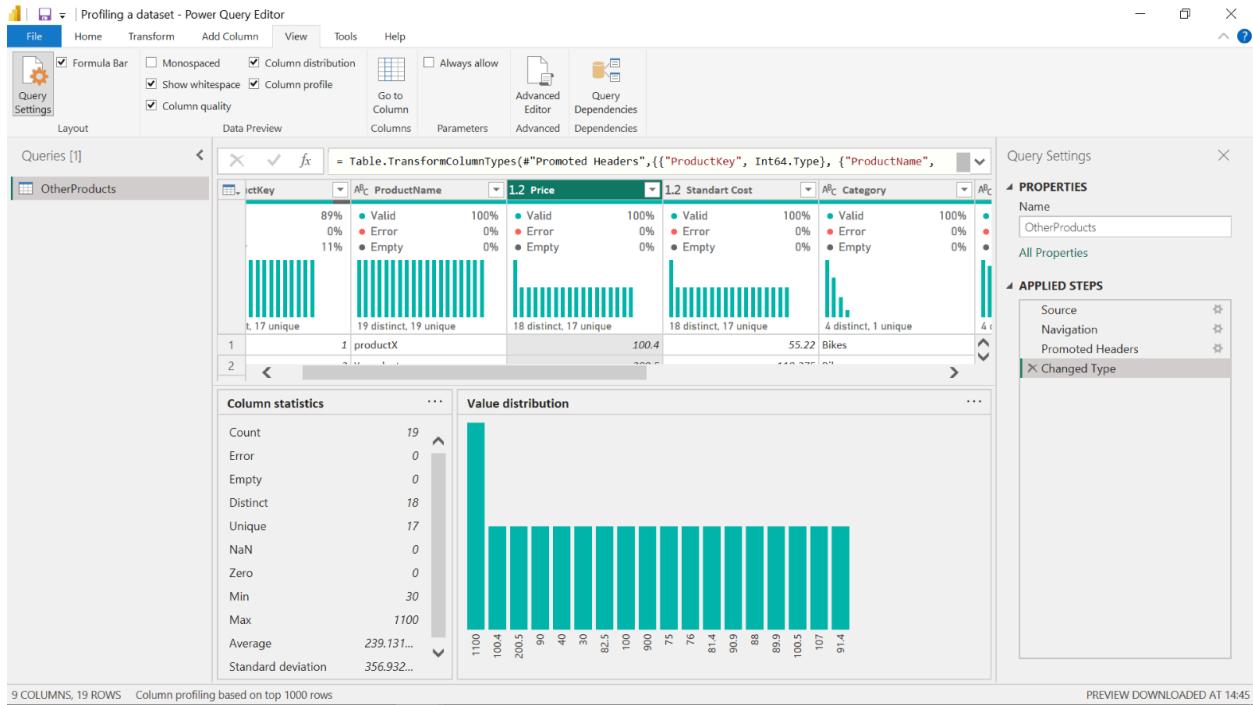
APPLIED STEPS

- Source
- Navigation
- Promoted Headers
- Changed Type

9 COLUMNS, 19 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 14:45

Step 5: Detect potential anomalies in the price column

1. You have to assess the Price column in Product list and you need to find out min, max, mean values and the distribution of the values. To detect potential anomalies and assess column distribution for the Price column, on the View ribbon tab, from inside the Data Preview group, check Column Profile while keeping Column Distribution checkbox as checked.
2. Note that min value is 30, max value is 1100, and average is 239.13 for the Price column.
3. There are 18 distinct and 17 unique values, which means there are 2 products with the same price.
4. When you assess the value distribution, it can be considered as normal distribution and there are some outliers like 30, 40, 900 and 1100. The aim of this assessment is to find a potential anomaly in the Price column.



Conclusion

Your objective for this exercise was to apply techniques for profiling data to detect potential anomalies. In this context, you learned how to use Power Query Editor to check column quality, column distribution and column profile.

3.2. Activity: Import a dataset with best practices

Introduction

As a data analyst, you play a vital role in analyzing large datasets to gain insights and improve business operations. Microsoft Power Query is an essential tool in your workflow, allowing you to transform and integrate data from various sources. At this point, you should have a good understanding of the best practices when using Microsoft Power Query and its practical applications. You gained valuable knowledge on planning, documentation, connectors, data profiling, performance, error handling, using groups, and optimizing data types. In this step-by-step exercise, you have the opportunity to apply best practices when importing datasets in Power BI.

Case study

You are working as a data analyst at Adventure Works which sells various consumer products. As the business expands, the management team wants to gain deeper insights into their sales performance across different regions. You use Power BI to import and analyze the sales data.

The company uses CSV files containing the Adventure Works company sales, reseller and product data. You will be applying the best practices for importing data.

By the end of this activity, you'll have a good understanding of how to apply best practices when importing datasets into Power BI.

Instructions

Create a new Power BI project called *Exercise – Import a dataset with Best Practices*. Follow the prompts below to complete the exercise.

Step 1: Download the CSV files

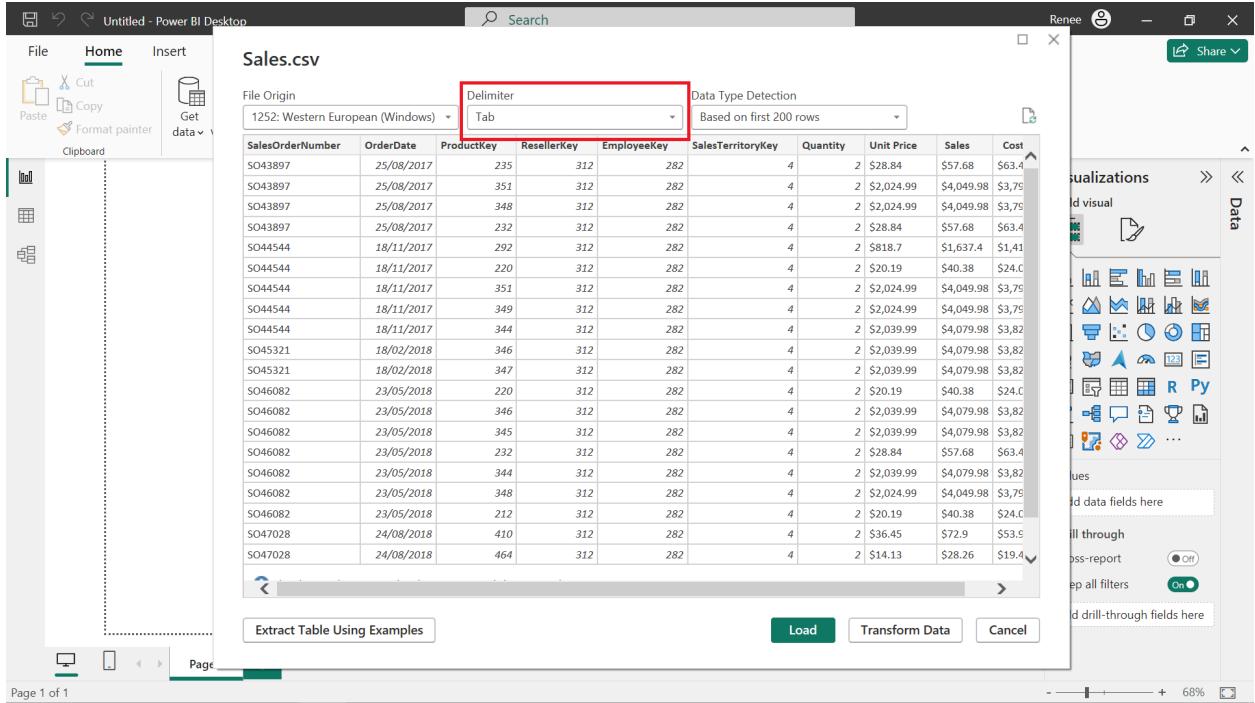
- Download the *Sales.csv*, *Product.csv* and *Reseller.csv* files, which will be used in this exercise. These files are available at the top of this article.

The screenshot shows the Microsoft Power Query Editor interface. The title bar reads "Exercise - Import a dataset with Best Practices - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The toolbar contains various icons for file operations, data sources, parameters, and query management. The main area displays a table titled "Sales" with columns: SalesOrderNumber, OrderDate, ProductKey, ResellerKey, and EmployeeKey. The table has 23 rows of data. The status bar at the bottom left indicates "10 COLUMNS, 999+ ROWS" and "Column profiling based on top 1000 rows". On the right side, there is a "Query Settings" pane with sections for "PROPERTIES" (Name: Sales) and "APPLIED STEPS" (Source: Promoted Headers, Changed Type). The status bar at the bottom right says "PREVIEW DOWNLOADED AT 06:33".

Step 2: Open the Power Query editor

Open the Power Query Editor and import your datasets, *Sales*, *Product*, and *Reseller* using the Text/CSV option.

- Tip: If the columns do not appear correctly, ensure that the Delimiter is set to Tab.



Step 3: Optimize data extraction

1. To optimize data extraction, apply filters to retrieve the sales data for the past five years only.
2. Select the OrderDate column, select the down arrow at the right side of the column and select the Date Filters menu.
3. Then select Custom Filter on the bottom of the list to select the last five years' records.
4. In the *Keep rows where the 'OrderDate' field boxes*, select "is in the previous," "5" and "years" in the respective sections to filter the dataset.

The screenshot shows the Power Query Editor interface. A 'Filter Rows' dialog is open, set to 'Basic' mode, filtering rows where 'OrderDate' is in the previous 5 years. The preview pane shows a table with columns: SalesOrderNumber, OrderDate, ProductKey, ResellerKey, EmployeeKey. The main interface shows the query = Table.SelectRows(#"Changed Type", each Date.IsInPreviousNYears([OrderDate], 4)).

Step 4: Transform and clean data

Using Power Query, transform and clean the data.

1. Find the empty rows in the OrderDate column
2. Find the error rows in the ProductKey column
3. Remove errors in these columns.

Note the results of performing these operations.

Step 5: Profile data

1. Examine the data in distribution, quality and profile.
2. Find error and empty rows, calculate minimum, maximum, average, count and other statistics, and compare distinct and unique values in different sales columns.
3. Find the minimum, maximum, and average values of the UnitPrice column

4. Find distinct and unique values in the Product Name column. Compare the distinct and unique values and find the reason for that.
5. Find the Count value of the SalesOrderID column.
6. Write down the values that you found.

Conclusion

By following these best practices, you successfully import Adventure Works sales data into Power BI, enabling you to analyze and profile sales performance across different regions. You gain best practices for optimizing data extraction, transforming, cleansing, and profiling data.

3.3. Activity: Using reference queries

Introduction

Microsoft Power BI provides a robust set of features that enables efficient data modeling and analysis. One of these powerful features is the ability to create reference queries. Reference queries allow you to reuse and reproduce changes across multiple queries. This reduces duplication of efforts and helps to ensure data consistency. In this reading, you will explore the benefits of using reference queries in Power BI and engage in a hands-on activity to reinforce how reference queries work.

Reference queries

With reference queries, you can create a master query that contains common data transformations and apply it to multiple dependent queries. Any changes made to the master query automatically replicate to all referencing queries. By leveraging reference queries, you can save significant time and effort when dealing with complex data transformations. Instead of recreating the same transformations multiple times, you can simply reference the master query and apply it to different datasets. This streamlined approach accelerates the data preparation process and allows you to focus on analysis and insights.

Reference queries offer a centralized approach to query maintenance. Instead of modifying transformations in each query, you can make updates in the master query, and those changes will be reflected in all referencing queries. Centralized maintenance simplifies query management, reduces the risk of errors, and improves overall query consistency. When you reference a query, the new query will have a single step: sourcing from the original query. The referenced query does not include the applied steps of the original query. If you make changes to the original query, then this new query will be impacted.

Case study

Adventure Works has recently acquired another bicycle business. Adventure Works' lead data analyst and your manager, Adio Quinn, tasks you with creating a Power BI query that merges data from this new business. There are two Excel files containing sales data: Adventure Works' sales data, named *AdventureWorksSales.xlsx*, and the other company's data, named *OtherSales.xlsx*. The datasets have some common data such as SalesOrderID, SalesOrderDetailID, UnitPrice, and SalesData, and some other columns which do not have the same names.

Ordinarily, you'd have to rename the common columns in queries so that they match, remove unnecessary columns if they exist, and finally append them. However, your manager asked you not to change any of the existing queries, because they want to use those as the source for other operations. As a result, you are going to create references from the queries, update the new queries, and then append them. Any changes in the base query will impact the new queries. By doing this, you will preserve the original queries, update the reference queries, and reflect the changes to them.

Instructions

Create a new Power BI project called *Activity—Using reference queries*. Follow the prompts below to complete the activity.

Step 1: Download the Excel files

Download the *AdventureWorksSales.xlsx* and *OtherSales.xlsx* files to use in this activity.

AdventureWorksSales.xlsx

SalesOrderID	SalesOrderDetailID	OrderQty	ProductName	UnitPrice	UnitPriceDiscount	LineTotal	SalesDate
75084	121222	1	Hitch Rack - 4-Bike	120	0	120,000,000	06/30/2014
75085	121223	1	AWC Logo Cap	8.99	0	8,990,000	06/30/2014
75085	121224	1	Bike Wash - Dissolver	7.95	0	7,950,000	06/30/2014
75086	121225	1	Bike Wash - Dissolver	7.95	0	7,950,000	06/30/2014
75087	121226	1	Sport-100 Helmet, Red	34.99	0	34,990,000	06/30/2014
75088	121227	1	Road Tire Tube	3.99	0	3,990,000	06/30/2014
75088	121228	1	ML Road Tire	24.99	0	24,990,000	06/30/2014
75088	121229	1	Sport-100 Helmet, Blue	34.99	0	34,990,000	06/30/2014
75088	121230	1	Long-Sleeve Logo Jersey, S	49.99	0	49,990,000	06/30/2014
75089	121231	1	LL Road Tire	21.49	0	21,490,000	06/30/2014
75089	121232	1	Road Tire Tube	3.99	0	3,990,000	06/30/2014
75089	121233	1	Sport-100 Helmet, Blue	34.99	0	34,990,000	06/30/2014
75090	121234	1	Mountain Tire Tube	4.99	0	4,990,000	06/30/2014
75090	121235	1	HL Mountain Tire	35	0	35,000,000	06/30/2014
75090	121236	1	Sport-100 Helmet, Black	34.99	0	34,990,000	06/30/2014
75091	121237	1	LL Road Tire	21.49	0	21,490,000	06/30/2014
75091	121238	1	Road Tire Tube	3.99	0	3,990,000	06/30/2014
75091	121239	1	Hydration Pack - 70 oz.	54.99	0	54,990,000	06/30/2014
75092	121240	1	LL Road Tire	21.49	0	21,490,000	06/30/2014
75092	121241	1	Half-Finger Gloves, M	24.49	0	24,490,000	06/30/2014
75092	121242	1	Road Tire Tube	3.99	0	3,990,000	06/30/2014

OtherSales.xlsx

SalesOrderID	SalesOrderDetailID	Quantity	Name	UnitPrice	UnitPriceDiscount	Total	SalesDate
101000	1	1	Hitch Rack - 4-Bike	120	0	120,000,000	06/20/2014
101000	2	1	Women's Mountain Shorts, S	69.99	0	69,990,000	06/20/2014
101000	3	1	Classic Vest, L	63.5	0	63,500,000	06/20/2014
101001	4	1	Women's Mountain Shorts, L	69.99	0	69,990,000	06/21/2014
101001	5	1	Long-Sleeve Logo Jersey, L	49.99	0	49,990,000	06/21/2014
101002	6	1	Water Bottle - 30 oz.	4.99	0	4,990,000	06/22/2014
101003	7	1	Sport-100 Helmet, Black	34.99	0	34,990,000	06/23/2014
101003	8	1	Water Bottle - 30 oz.	4.99	0	4,990,000	06/23/2014
101003	9	1	Bike Wash - Dissolver	7.95	0	7,950,000	06/23/2014
101003	10	1	Mountain Tire Tube	4.99	0	4,990,000	06/24/2014
101004	11	1	Sport-100 Helmet, Red	34.99	0	34,990,000	06/24/2014
101005	12	1	AWC Logo Cap	8.99	0	8,990,000	06/25/2014
101005	13	1	Long-Sleeve Logo Jersey, L	49.99	0	49,990,000	06/25/2014
101005	14	1	Mountain Tire Tube	4.99	0	4,990,000	06/25/2014
101005	15	1	Patch Kit/8 Patches	2.29	0	2,290,000	06/25/2014
101005	16	1	Fender Set - Mountain	21.98	0	21,980,000	06/25/2014
101005	17	1	Sport-100 Helmet, Black	34.99	0	34,990,000	06/25/2014
101005	18	1	Mountain Tire Tube	4.99	0	4,990,000	06/25/2014
101005	19	1	Fender Set - Mountain	21.98	0	21,980,000	06/25/2014
101005	20	1	Classic Vest, S	63.5	0	63,500,000	06/25/2014
101005	21	1	Fender Set - Mountain	21.98	0	21,980,000	06/25/2014

Step 2: Open the Power Query Editor

- Open the Power Query editor and import your datasets, *AdventureWorksSales.xlsx* and *OtherSales.xlsx*.

Step 3: Reference queries

Create reference queries from the datasets you imported using the instructions that follow:

1. Navigate to the left menu of the Power Query interface.
2. In the Queries pane, right-click the *AdventureWorksSales* file and select Reference in the menu. Reference will create a new query which is a copy of the *AdventureWorksSales*, but only contains one step.
3. Rename it as *AdventureWorksSales-Referenced*.
4. Repeat the previous steps for the *OtherSales* query.

Step 4: Format Excel files

To append *OtherSales-Referenced* data to *AdventureWorksSales-Referenced* data, you will use *AdventureWorksSales-Referenced* data as first table and *OtherSales-Referenced* data as the second table. For this reason:

1. Format the *OtherSales-Referenced* data.
2. Rename the columns using the *AdventureWorksSales-Referenced* data. For example, change Quantity to OrderQty, Name to ProductName, and Total to LineTotal.

Step 5: Append queries

1. Select the queries you want to append the queries to a new master table by selecting them.
2. To append queries, navigate to the Home tab and choose Append Queries as New in the Append Queries menu.

3. Select the queries you want to append together and press OK to create a new master table.
4. In the Queries pane, select the newly created query and check its column names, row number, and the values appended to ensure that the operation is completed successfully.

Step 6: Rename the new query

1. In the left menu in Power Query, select the new query and change its name to *Consolidated Sales*.
2. In the Properties pane on the right, press Enter.

Step 7: Uncheck the Enable Load

1. You can prevent the original queries from loading, and saving memory in Power BI.
2. To do so, right-click the *AdventureWorksSales* query and uncheck the Enable Load option.
3. Repeat for the *OtherSales* query.

Conclusion

Reference queries are a powerful feature in Power BI that enables reusability, consistency, and centralized maintenance. By leveraging reference queries, you can streamline your data preparation process, save time and effort, and ensure consistency across multiple queries. The activity provided in this reading will help you get started with creating and using reference queries in Power BI. Reference queries will always reflect the latest data from the master query, allowing you to focus on analysis and generating valuable insights. Remember to regularly refresh the data in Power BI to ensure that your reports and visualizations are up to date with the latest sales information from the Adventure Works database. By using reference queries in your Power BI projects, you can unlock the full potential of query reusability and maintenance and experience enhanced efficiency and productivity in your data modeling and analysis endeavors.

3.4. Activity: Disabling helper queries

Introduction

As a member of the data analytics team at Adventure Works, your job involves wrangling a large amount of data from different regions. One morning, you receive an email containing CSV files. You must combine these multiple files into a coherent report for the quarterly review meeting at noon.

Feeling under pressure, you turn to your most trusted ally in these situations –Microsoft Power BI. As you navigate through Power BI's features and begin the data integration process, you notice a peculiar pattern. Power BI is generating many additional queries alongside your primary one. These helper queries are automatically crafted by Power Query when you combine files from a folder.

First, it feels like a smart feature, but then you soon notice unfavorable consequences. Your Query Editor is gradually filling up, cluttered with more queries than you find useful, thereby slowing you down. You find yourself wanting a cleaner, more streamlined process. You ask yourself: "Is there a way to manage or disable these extra queries?" Let's uncover the mystery behind these helper queries and equip you with the knowledge to modify their behavior.

This reading helps you to understand the role of helper queries in Power BI Desktop. It explains how and why you might want to disable these helper queries.

What are Helper Queries?

In Power BI Desktop, a helper query is a sequence of steps used to perform some sort of data preparation or transformation. Helper queries run every time you refresh your data, ensuring that your data is always in the right shape for analysis. Despite their utility, there could be scenarios when you want to disable these helper queries.

Performance could be one reason. Every query in Power BI Desktop consumes computer memory and processing power. If you're working with large datasets or complex transformations, running too many queries can slow down your report refreshes. There may also be times when the data shape required by a specific part of your report changes, and the associated helper query is no longer required. In such cases, you can disable these helper queries to ensure they don't consume unnecessary resources.

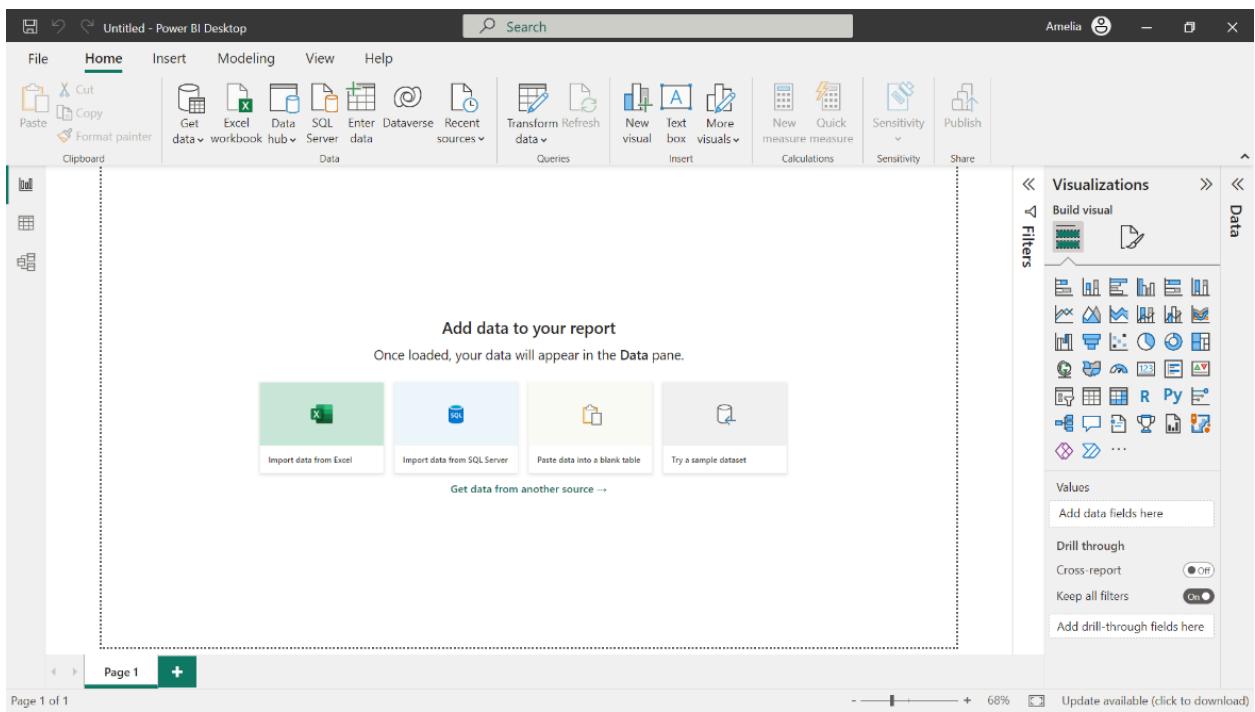
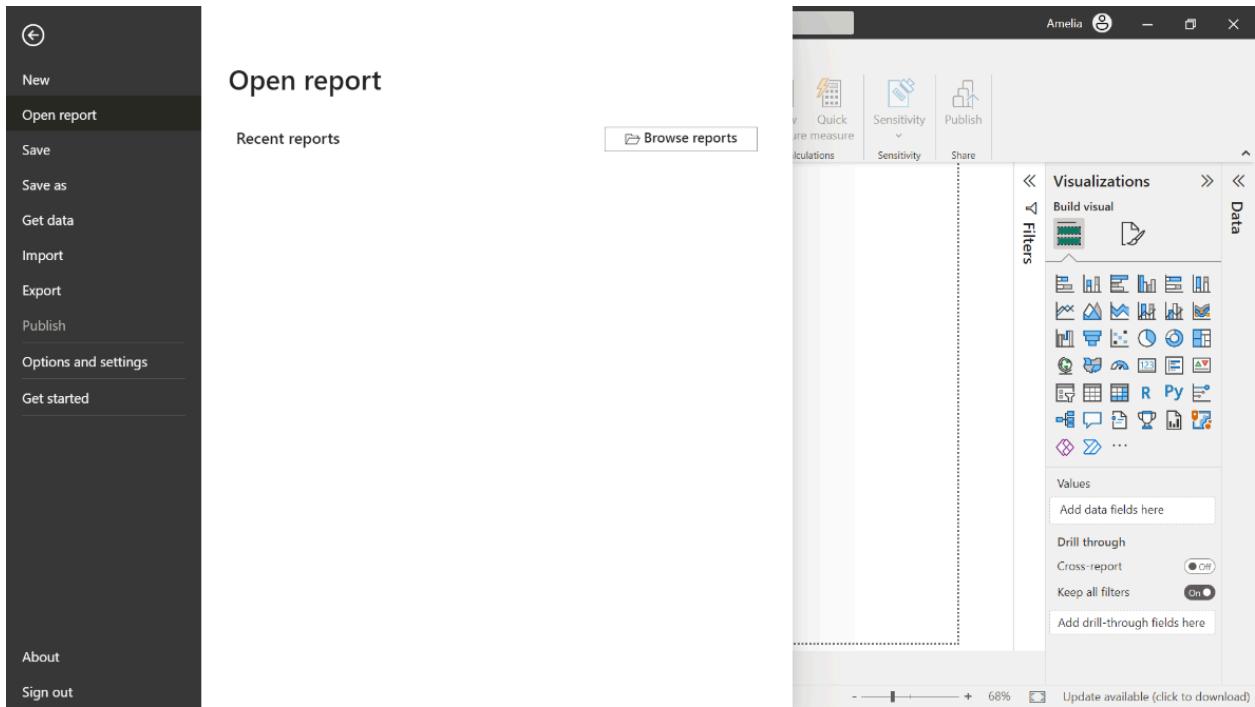
Activity: Disabling Helper Queries

Now, let's follow the steps to disable helper queries.

First, download the *AdventureWorksSales.zip* and unzip the file to a folder.

Tip: Right-click the zip file and select Extract All.

1. In this extracted folder there will be a *Disabling Helper Queries* Power BI file and a *Data* folder.
2. From the Microsoft Power BI interface's top left corner select File, and then select Open Report. Locate and select the *Disabling Helper Queries* report file on your system. Once the report is loaded, you will see multiple tabs at the top of the screen, including Home, View, Insert, Modeling, etc. Select the Home tab.



On the Home tab, select Transform Data dropdown and then Data Source Settings.

Select Change Source and change the source to the *Data* folder in the *AdventureWorksSales* folder.

1. Click Close.
2. On the Home tab, in Queries, select Transform Data. The Power Query Editor opens in a new window. On the left side of this window hover your mouse over the Queries pane. It lists all the queries in your report.

The screenshot shows the Power Query Editor interface with the following details:

- File** tab is selected.
- Queries [5]** pane on the left lists:
 - Transform File from AdventureWorks...
 - Helper Queries [3]
 - Sample File
 - Parameter 1 (Sample File)
 - fx Transform File
 - Transform Sample File
 - Other Queries [1]
 - AdventureWorksSales** (highlighted in green)
- Query Settings** pane on the right shows:
 - PROPERTIES**: Name = AdventureWorksSales
 - APPLIED STEPS** list includes:
 - Source
 - Filtered Hidden Files1
 - Invoke Custom Function1
 - Renamed Columns1
 - Removed Other Columns1
 - Expanded Table Column1
 - Changed Type** (highlighted in red)
- Preview** pane at the bottom shows the first 144 rows of the 'Customer' table.

1. Select the *AdventureWorksSales* query to highlight it. You've just selected a combined query named after the imported folder, *AdventureWorksSales*.
2. Move your pointer over the query name and right-click to open a shortcut menu.

The screenshot shows the Power Query Editor interface with the 'Transform' tab active. A context menu is open over a table containing data from 'Customers.csv'. The 'Enable load' option in the menu is checked. The 'APPLIED STEPS' pane on the right lists actions such as 'Source', 'Filtered Hidden Files1', 'Invoke Custom Function1', 'Renamed Columns1', 'Removed Other Columns1', 'Expanded Table Column1', and 'Changed Type'.

1. Among the various options in the shortcut menu, notice the Enable Load option. If this option has a checkmark next to it, it means the query is currently set to load its data into your model.

Select the Enable Load option in the Context menu. This removes the check mark next to the option, indicating that the query is not to load its data into your model, thereby disabling helper queries.

2. If a new alert box called Possible Data Loss Warning appears, select Continue.

Queries [5]

Transform File from AdventureWorksSales...

Helper Queries [3]

- Sample File
- Parameter1 (Sample File)
- Transform File
- Transform Sample File

Other Queries [1]

AdventureWorksSales

Query Settings

PROPERTIES

Name: AdventureWorksSales

All Properties

APPLIED STEPS

- Source
- Filtered Hidden Files1
- Invoke Custom Function1
- Renamed Columns1
- Removed Other Columns1
- Expanded Table Column1

Changed Type

PREVIEW DOWNLOADED AT 02:13

- When you've finished disabling all unneeded queries, on the extreme left of the Home tab of the Power Query Editor select Close & Apply to save your changes and close the Power Query Editor.

File Home Transform Add Column View Tools Help

Close & Apply New Recent Enter Data Data source settings Manage Parameters Refresh Preview Advanced Editor Choose Columns Remove Rows Keep Rows Reduce Rows Sort Split Column Group By Data Type: Text Use First Row as Headers Merge Queries Append Queries Combine Files Text Analytics Vision Azure Machine Learning AI Insights

Queries [5]

Transform File from AdventureWorksSales...

Helper Queries [3]

- Sample File
- Parameter1 (Sample File)
- Transform File
- Transform Sample File

Other Queries [1]

AdventureWorksSales

Query Settings

PROPERTIES

Name: AdventureWorksSales

All Properties

APPLIED STEPS

- Source
- Filtered Hidden Files1
- Invoke Custom Function1
- Renamed Columns1
- Removed Other Columns1
- Expanded Table Column1

Changed Type

PREVIEW DOWNLOADED AT 02:13

1. Now you're back on the main screen of Power BI Desktop. You can continue creating or modifying visualizations, setting up relationships, and so forth. The data from the disabled queries will no longer be loaded into your model. The queries you've disabled are still listed in the Queries pane on the left in the Power Query Editor. However, they are no longer loading data into your model.
2. If you decide you want the data from these queries in your model again, you can easily re-enable them. Simply go back into the Power Query Editor, right-click on the query, and choose Enable Load to put a checkmark next to it.

Conclusion

With data analysis, the path to insightful results often involves a considerable amount of data transformation and cleanup. In Power BI Desktop, this process is simplified and made efficient by the use of helper queries. Their background work ensures your data is always in the right shape for analysis. Although useful, it's essential to consider the resource implications of these tools. Every query, including a helper query, consumes memory and processing power. Unmanaged, they can unintentionally slow down your report refreshes. Therefore, learning how to disable unnecessary helper queries gives you control of your reports' performance. As you continue your journey with Power BI and beyond, embrace the adventure, and think of every data set, every query, and every insight as a step in your journey to master data analysis.

4. Final Course Project: Transforming multiple data sources

Introduction

In this course, using Power BI, you covered the data extract, transform, and load process (ETL). You should now have a good understanding of:

- Connecting to data sources
- Column data types
- Common data errors
- Combining tables with merge and append
- Advanced data transformation techniques
- Loading and staging data
- Profiling data by using column quality
- Column distribution and column profile, and its practical application in identifying data anomalies.

In this exercise, you will apply your knowledge in an end-to-end scenario by using Power Query to clean and transform multiple data sources and join and merge them. You'll also examine the valid, error, empty, min, max, unique, and distinct values in the rows. This will allow you to identify the anomalies in the data. Finally, you will remove the data sources with anomalies. This exercise will help you understand how to clean, transform, join, and merge data sources in Power Query, and identify potential data anomalies by using data profiling tools.

Case study

You are working as a data analyst at Adventure Works. Sales data is contained in two main tables, Order and OrderDetails.

Data Sources

The Order data table includes general information about the sales such as:

Order			
OrderDate	TotalDue	TerritoryID	SalesPersonID

The OrderDetails data contains related details of each main sales record such as:

OrderDetails			
ProductID	OrderQty	UnitPrice	UnitPrice Discount

Files

Adventure Works trades internationally and generates a large volume of sales data. To manage file sizes, the active Order table only includes data for the year 2023. Older data is stored in separate files for each year with the same fields and table structure.

The task

Your manager, Adio Quinn, asks you to conduct a detailed analysis of store sales. In the detail table, OrderDetails, there are multiple fields, but you only need ProductID, the quantity sold (which is in the field OrderQty), and the UnitPrice. Therefore, you are expected to remove unnecessary fields, and also eliminate empty rows, and identify any anomalies to remove those rows if necessary. After performing these tasks, you will append the two separate sales data sources together and then merge that with OrderDetails. Follow the steps below to complete the exercise.

Instructions

Step 1: Set up the project

1. Create a new Power BI project called *Exercise – Transforming Multiple Data Sources*.
2. Download the *Order2022.xlsx*, *Order2023.xlsx* and *OrderDetails.xlsx* files, which you will use in this exercise.

The screenshot shows the Power Query Editor interface with the title "Exercise – Transforming Multiple Data Sources - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The ribbon tools include Close & Apply, New Query, Data Sources, Parameters, Properties, Advanced Editor, Manage, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Sort, Split Column, Group By, Use First Row as Headers, Transform, Merge Queries, Append Queries, Text Analytics, Vision, Combine Files, Azure Machine Learning, and AI Insights. The "Queries [3]" pane on the left lists "Order2022", "Order2023", and "Order Details". The main area displays a table with 23 rows of data. The first few rows are:

	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID
1	43659	1	4911-403C-98	1	
2	43659	2	4911-403C-98	3	
3	43659	3	4911-403C-98	1	
4	43659	4	4911-403C-98	1	
5	43659	5	4911-403C-98	1	
6	43659	6	4911-403C-98	2	
7	43659	7	4911-403C-98	1	
8	43659	8	4911-403C-98	3	
9	43659	9	4911-403C-98	1	
10	43659	10	4911-403C-98	6	
11	43659	11	4911-403C-98	2	
12	43659	12	4911-403C-98	4	
13	43660	13	6431-4057-83	1	
14	43660	14	6431-4057-83	1	
15	43661	15	4E0A-4F89-AE	1	
16	43661	16	4E0A-4F89-AE	1	
17	43661	17	4E0A-4F89-AE	2	
18	43661	18	4E0A-4F89-AE	4	
19	43661	19	4E0A-4F89-AE	4	
20	43661	20	4E0A-4F89-AE	2	
21	43661	21	4E0A-4F89-AE	3	
22	43661	22	4E0A-4F89-AE	2	
23	43661	23	4E0A-4F89-AE	2	

Below the table, it says "11 COLUMNS, 999+ ROWS" and "Column profiling based on top 1000 rows". On the right, there are "Query Settings" sections for "PROPERTIES" (Name: Order Details) and "APPLIED STEPS" (Source, Navigation, Promoted Headers, Changed Type). At the bottom right, it says "PREVIEW DOWNLOADED AT 20:11".

Step 2: Open the Power Query Editor

1. Use the Get Data feature in Power BI.
2. Select Transform to open the Power Query editor.
3. Import your .xlsx datasets, *Order2022*, *Order2023* and *OrderDetails*.

Step 3: Choose columns from Order Details

1. Open the Order Details query by selecting it from the Queries pane.
2. Keep SalesOrderID, ProductID, OrderQty, UnitPrice columns.
3. Remove the other columns by simply right clicking and selecting the Remove option in the shortcut-menu.

Step 4: Profile data in Order Details

1. To profile data, select Order Details.
2. On the View tab, in the Data Preview group, check Column Distribution, Column Quality and Column Profile checkboxes.

3. Note the amount of distinct values and unique values in Column Distribution.
4. Also check the valid, error and empty values of all columns in Column Quality.

Step 5: Detect potential anomalies in the price

To detect potential anomalies and assess column distribution for the price, in this step you will assess the UnitPrice column in Order Details list and find out the min, max, mean values and the distribution of the values.

1. In the View tab, in the Data Preview group, check Column Profile while keeping Column Distribution checkbox as checked.
2. Note the min, max, and mean values for the UnitPrice column and also assess the Column Distribution.
3. You should find that three rows are outliers in the UnitPrice column.
4. Consider these three rows as data anomalies (they are most probably mistypes when data was entered) . Remove them by filtering and unchecking these values to avoid confusion and incorrect calculations.

Step 6: Append queries

1. Append Order2022 and Order2023 queries in a new master table.
2. Check the newly created query, its column names, row number, and the values appended.
3. Make sure that the operation has been completed successfully.
4. Rename it as Orders.

Step 7: Merge queries

1. Select the Order Details data in the Queries pane and choose Merge Queries.
2. In the opened window, the Order Details table will be automatically shown in the upper part.

3. You will be choosing Order as the next table for merging. SalesOrderID is the common column between the tables. To begin establishing a connection select the SalesOrderID column in each table.
4. For the Join Kind dropdown, choose the join type Inner Join, which selects the matching records from the left table and the right table.
5. Choose Expand near the newly added Orders table column and choose only the OrderDate column from Orders table in the opened window.
6. Rename the Orders.OrderDate column to OrderDate by simply double-clicking on it.

Conclusion

You have now successfully completed an end-to-end scenario by cleaning and transforming multiple data sources, joining and merging them, and identifying potential anomalies in the data using Power Query.

Exemplar: Transforming multiple data sources

Introduction

In the exercise *Transforming multiple data sources*, you put into practice your understanding of how to clean and transform multiple data sources, profile data and detect potential anomalies in Microsoft Power Query.

You were asked to complete tasks, including:

- Clean and transform multiple data sources.
- Join and merge multiple data sources.
- Examine the valid, error, empty, min, max, unique, and distinct values in the rows, to identify the anomalies in the data.
- Remove the data sources with anomalies.

This reading provides a step-by-step guide to completing these tasks, accompanied by screenshots for each step. Follow the steps below to compare the outcome with your own work on the exercise.

Instructions

Step 1: Setup the project

1. You created a new Power BI project called *Exercise – Transforming Multiple Data Sources*.
2. Then you downloaded the *Order2022.xlsx*, *Order2023.xlsx* and *OrderDetails.xlsx* files, that were needed for this exercise.

Step 2: Open the Power Query Editor

1. You used the Get Data feature in Power BI.
2. You selected Transform to open the Power Query editor.
3. Your .xlsx datasets, *Order2022*, *Order2023* and *OrderDetails* were imported. The Power Query Editor window opens. You can now begin working with the data in Power Query.

The screenshot shows the Microsoft Power BI Data Editor interface. On the left, the 'Navigator' pane displays a list of queries: 'OrderDetails.xlsx [1]' and 'Order Details'. The 'Order Details' query is selected, indicated by a checkmark. The main area shows a table titled 'Order Details' with columns: SalesOrderID, SalesOrderDetailID, CarrierTrackingNumber, OrderQty, and ProductID. The data consists of approximately 25 rows of order details. On the right, the 'Data' pane is open, showing the 'Header' section. At the bottom, there are buttons for 'Load', 'Transform Data', and 'Cancel'. The status bar at the bottom right shows '66%'.

Step 3: Choose columns from OrderDetails

1. The OrderDetails query was opened by selecting it from the Queries pane.
2. You retained the columns: SalesOrderID, ProductID, OrderQty, UnitPrice
3. You removed the other columns by selecting them and using a right-click to choose the Remove option.

The screenshot shows the Power Query Editor interface with the title "Exercise – Transforming Multiple Data Sources - Power Query Editor". The ribbon has tabs like File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The ribbon also includes Close & Apply, New Query, Data Sources, Parameters, Properties, Advanced Editor, Manage, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Sort, Split Column, Group By, Data Type: Any, Use First Row as Headers, Transform, Merge Queries, Append Queries, Text Analytics, Vision, Combine Files, Azure Machine Learning, and AI Insights.

The main area shows a table with three columns: SalesOrderID, SalesOrderDetailID, and CarrierTrackingNumber. The CarrierTrackingNumber column has a context menu open with the option "Remove Columns" highlighted. Other options in the menu include Copy, Remove Other Columns, Add Column From Examples..., Remove Duplicates, Remove Errors, Replace Values..., Fill, Change Type, Merge Columns, Group By..., Unpivot Columns, Unpivot Other Columns, Unpivot Only Selected Columns, and Move.

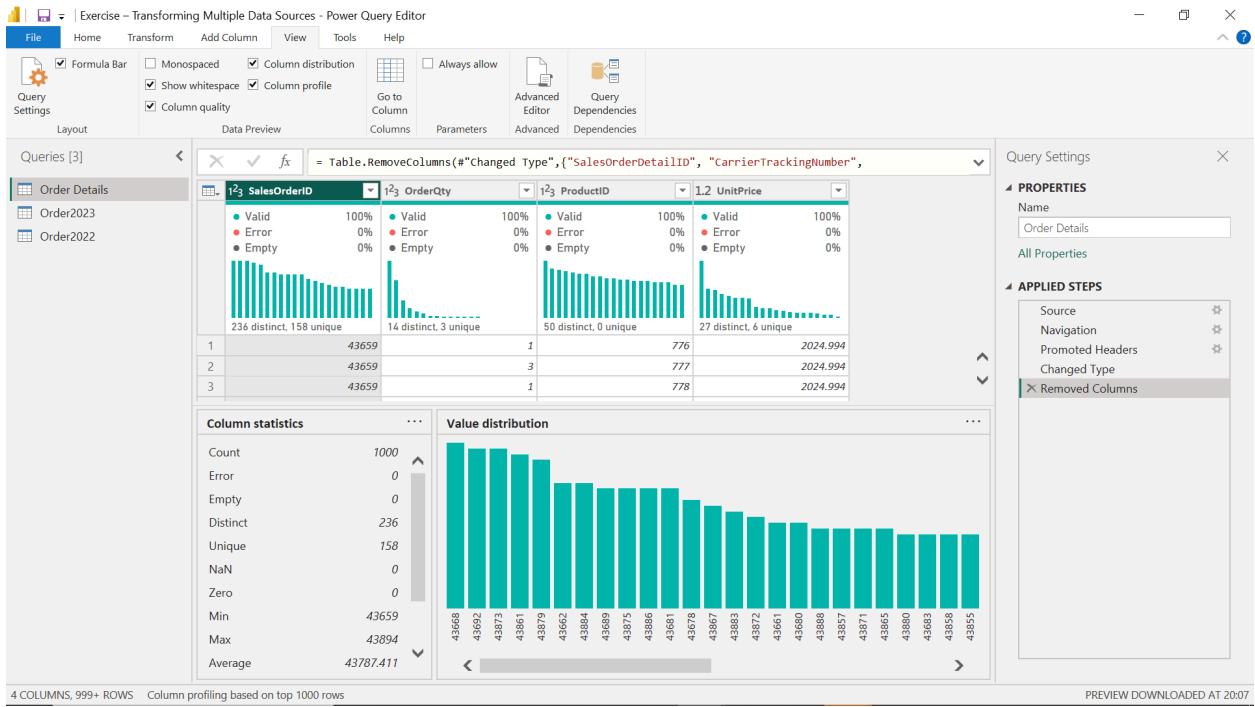
The "APPLIED STEPS" pane on the right lists the following steps:

- Source
- Navigation
- Promoted Headers
- Changed Type

At the bottom left, it says "11 COLUMNS, 999+ ROWS" and "Column profiling based on top 1000 rows". At the bottom right, it says "PREVIEW DOWNLOADED AT 20:07".

Step 4: Profile data in OrderDetails

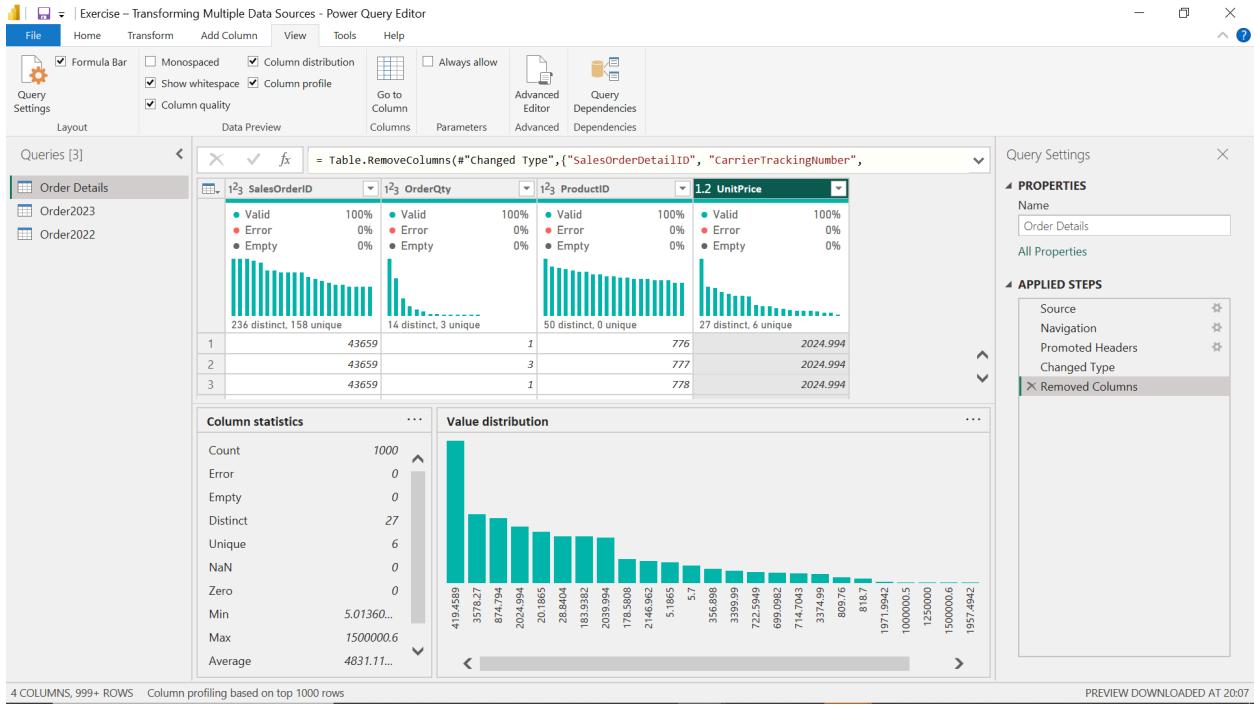
1. To profile the data, you selected OrderDetails
2. Then on the View tab, in the Data Preview group, you ensured there was a check on Column Distribution, Column Quality and Column Profile checkboxes.
3. You took note of the number of distinct values and unique values in Column Distribution.
4. You also checked the valid, error and empty values of all columns in Column Quality.



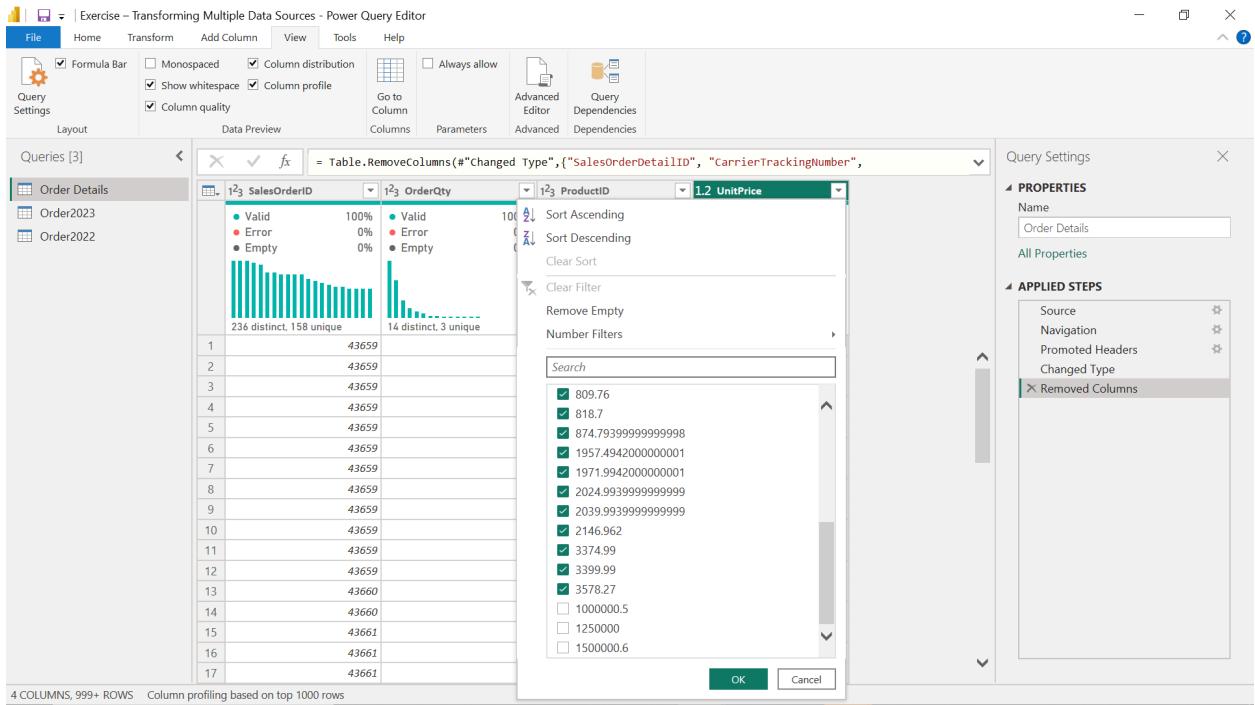
Step 5: Detect potential anomalies in the price

To detect potential anomalies and assess column distribution for the price, in this step you assessed the UnitPrice column in OrderDetails list and found the min, max, mean values and the distribution of the values.

1. In the View tab, in the Data Preview group, you checked Column Profile while keeping Column Distribution checkboxes as checked.



1. You noted the min, max, and mean values for the UnitPrice column and also assessed the Column Distribution.
2. You filtered the UnitPrice column by simply clicking the down arrow near the UnitPrice heading and find out that 3 rows are outliers in the UnitPrice column. You should have found that these three rows were the outliers:
3. You considered these three rows as data anomalies (they were probably mistypes when data was entered). Therefore, you removed them by filtering and unchecking these values to avoid confusion and incorrect calculations.



Step 6: Append Queries

In your task statement, you are asked to append the separate sales data together after performing the other tasks. To do this, you need to append the queries in a new query.

1. You select *Order2022* then on the Home tab in the Combine group you selected Append Queries as New option in the Append Queries drop-down menu.

The screenshot shows the Power Query Editor interface with the title "Exercise – Transforming Multiple Data Sources - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The toolbar contains various icons for file operations, data sources, parameters, and query management. The main area displays a table with columns: SalesOrderID, RevisionNumber, OrderDate, DueDate, and ShipDate. The table has 26 columns and 500 rows. The status bar at the bottom indicates "26 COLUMNS, 500 ROWS" and "Column profiling based on top 1000 rows". On the right side, there is a "Query Settings" pane and an "APPLIED STEPS" pane. The "APPLIED STEPS" pane shows the step "Changed Type" for the "Source" column, which was promoted from Headers.

You selected *Order2022* as the first table and *Order2023* as the second table.

The screenshot shows the Power Query Editor interface with the title "Exercise – Transforming Multiple Data Sources - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The toolbar contains various icons for file operations, data sources, parameters, and query management. A modal dialog box titled "Append" is open, asking "Concatenate rows from two tables into a single table." with two options: "Two tables" (selected) and "Three or more tables". Below this, there are dropdown menus for "First table" (set to "Order2022") and "Second table" (set to "Order2023"). At the bottom of the dialog are "OK" and "Cancel" buttons. The background shows the same table structure as the previous screenshot. The status bar at the bottom indicates "26 COLUMNS, 500 ROWS" and "Column profiling based on top 1000 rows". On the right side, there is a "Query Settings" pane and an "APPLIED STEPS" pane. The "APPLIED STEPS" pane shows the step "Changed Type" for the "Source" column, which was promoted from Headers.

1. You then checked the newly created query, its column names, row number, and the values appended.

- You checked to make sure that the operation has been completed successfully.

The screenshot shows the Power Query Editor interface with the following details:

- File** tab is selected.
- Queries [4]** pane shows four queries: Order2022, Order2023, Order Details, and Append1.
- Transform** ribbon tab is selected.
- Table** ribbon tab is selected.
- Table** ribbon tab dropdown shows the formula: `= Table.Combine({Order2022, Order2023})`.
- Context Menu** (right-clicked on the Append1 query):
 - Rename** is selected.
 - Other options include: Copy, Paste, Delete, Enable load, Include in report refresh, Duplicate, Reference, Move To Group, Move Up, Move Down, Create Function..., Convert To Parameter, Advanced Editor, and Properties...
- Properties** pane shows the Name field set to "Append1".
- Applied Steps** pane shows the "Source" step.
- Query Settings** pane is visible on the right.
- Data Preview** shows the combined data from Order2022 and Order2023, with columns SalesOrderID, RevisionNumber, OrderDate, DueDate, and ShipDate.
- Bottom Status Bar** indicates 26 COLUMNS, 999+ ROWS, Column profiling based on top 1000 rows, and PREVIEW DOWNLOADED ON TUESDAY.

- You right-clicked on the newly added query and selected Rename to rename it as Orders.

The screenshot shows the Power Query Editor interface with the following details:

- File**: Home, Transform, Add Column, View, Tools, Help
- Close & Apply**, **New Query**, **Data Sources**, **Parameters**, **Properties**, **Advanced Editor**, **Query**
- Manage Columns**: Choose Columns, Remove Columns, Keep Rows, Remove Rows, Reduce Rows, Sort
- Transform**: Split Column, Group By, Use First Row as Headers, Replace Values, Merge Queries, Merge Queries as New, Merge Queries as New
- Text Analytics**, **Vision**, **Azure Machine Learning**, **AI Insights**
- Queries [4]**: Order2022, Order2023, Order Details, Orders
- Table**: `= Table.Combine({Order2022, Order2023})`
- Properties**: Name set to "Orders", All Properties
- Applied Steps**: Source
- Query Settings**
- Table Preview** (26 COLUMNS, 999+ ROWS): Shows columns SalesOrderID, RevisionNumber, OrderDate, DueDate, ShipDate. The preview shows rows 1 through 23.
- Bottom Status Bar**: PREVIEW DOWNLOADED ON TUESDAY

Step 7: Merge Queries:

In the final step, you are asked to merge the tables OrderDetails and Orders by using the common column between the tables and take the needed columns from each table.

1. You selected the OrderDetails data in the Queries pane and from Home > Combine you chose Merge Queries in the Merge Queries drop-down menu.

The screenshot shows the Power Query Editor interface with the title "Exercise – Transforming Multiple Data Sources - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The toolbar includes Close & Apply, New Source, Recent Sources, Enter Data, Data source settings, Parameters, Refresh, Advanced Editor, Properties, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Reduce Rows, Sort, Split Column, Group By, and Data Type: Decimal Number. A context menu for the "Order Details" query is open, showing options like "Merge Queries" and "Text Analytics". The "Properties" pane on the right shows the "Name" as "Order Details". The "Applied Steps" pane shows the last step as "Renamed Columns". The main area displays two tables: "Order Details" and "Orders". The "Order Details" table has columns SalesOrderID, OrderQty, ProductID, and UnitPrice. The "Orders" table has columns SalesOrderID, CustomerID, OrderDate, and ShipAddress. The "SalesOrderID" column is highlighted in both tables, indicating it is the common key for merging.

1. In the opened window, OrderDetails table was automatically shown in the upper part.
2. You selected the next table for merging, which was Order using SalesOrderID as the common column between the tables to establish connection.

Screenshot of the Power Query Editor showing the 'Merge' step. The 'Order Details' query is being merged with the 'Orders' query. The 'Join Kind' is set to 'Inner (only matching rows)'. A message at the bottom indicates 997 rows from the first table and 236 from the second were matched.

SalesOrderID	OrderQty	ProductID	UnitPrice
43659	1	776	2024.994
43659	3	777	2024.994
43659	1	778	2024.994
43659	1	771	2039.994
43659	1	772	2039.994

SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrder
43659	8	31/05/2022 00:00:00	12/06/2022 00:00:00	07/06/2022 00:00:00	5	
43660	8	31/05/2022 00:00:00	12/06/2022 00:00:00	07/06/2022 00:00:00	5	
43661	8	31/05/2022 00:00:00	12/06/2022 00:00:00	07/06/2022 00:00:00	5	
43662	8	31/05/2022 00:00:00	12/06/2022 00:00:00	07/06/2022 00:00:00	5	

- For the Join Kind dropdown, you selected the join type of Inner Join, which selects the matching records from the left table and the right table.
- Then you selected Append near the newly added Orders table column and selected only the OrderDate column from the Orders table in the opened window.

Screenshot of the Power Query Editor showing the 'Transform' tab selected. A 'NestedJoin' step is being edited, joining the 'Order Details' query with the 'Orders' query on the 'SalesOrderID' column. The 'Orders' query is expanded, and the 'OrderDate' column is selected for renaming.

OrderID	OrderQty	ProductID	UnitPrice	OrderDate
1	43659	1	204.994	31/05/2022 00:00:00
2	43659	3	204.994	31/05/2022 00:00:00
3	43659	1	204.994	31/05/2022 00:00:00
4	43659	1	204.994	31/05/2022 00:00:00
5	43659	1	204.994	31/05/2022 00:00:00
6	43659	2	204.994	31/05/2022 00:00:00
7	43659	1	204.994	31/05/2022 00:00:00
8	43659	3	204.994	31/05/2022 00:00:00
9	43659	1	204.994	31/05/2022 00:00:00
10	43659	6	204.994	31/05/2022 00:00:00
11	43659	2	204.994	31/05/2022 00:00:00
12	43659	4	204.994	31/05/2022 00:00:00
13	43660	1	204.994	31/05/2022 00:00:00
14	43660	1	204.994	31/05/2022 00:00:00
15	43661	1	204.994	31/05/2022 00:00:00
16	43661	1	204.994	31/05/2022 00:00:00
17	43661	2	204.994	31/05/2022 00:00:00
18	43661	4	204.994	31/05/2022 00:00:00
19	43661	4	204.994	31/05/2022 00:00:00
20	43661	2	204.994	31/05/2022 00:00:00
21	43661	3	204.994	31/05/2022 00:00:00
22	43661	2	204.994	31/05/2022 00:00:00
23	43661	2	204.994	31/05/2022 00:00:00

- Using a double-click you then renamed the Orders.OrderDate column to its new name of OrderDate.

Screenshot of the Power Query Editor showing the 'Transform' tab selected. The 'ExpandTableColumn' step is being edited, expanding the 'Merged Queries' query on the 'Orders' column and selecting the 'OrderDate' column for renaming.

OrderID	OrderQty	ProductID	UnitPrice	OrderDate
1	43659	1	204.994	31/05/2022 00:00:00
2	43659	3	204.994	31/05/2022 00:00:00
3	43659	1	204.994	31/05/2022 00:00:00
4	43659	1	204.994	31/05/2022 00:00:00
5	43659	1	204.994	31/05/2022 00:00:00
6	43659	2	204.994	31/05/2022 00:00:00
7	43659	1	204.994	31/05/2022 00:00:00
8	43659	3	204.994	31/05/2022 00:00:00
9	43659	1	204.994	31/05/2022 00:00:00
10	43659	6	204.994	31/05/2022 00:00:00
11	43659	2	204.994	31/05/2022 00:00:00
12	43659	4	204.994	31/05/2022 00:00:00
13	43660	1	204.994	31/05/2022 00:00:00
14	43660	1	204.994	31/05/2022 00:00:00
15	43661	1	204.994	31/05/2022 00:00:00
16	43661	1	204.994	31/05/2022 00:00:00
17	43661	2	204.994	31/05/2022 00:00:00
18	43661	4	204.994	31/05/2022 00:00:00
19	43661	4	204.994	31/05/2022 00:00:00
20	43661	2	204.994	31/05/2022 00:00:00
21	43661	3	204.994	31/05/2022 00:00:00
22	43661	2	204.994	31/05/2022 00:00:00
23	43661	2	204.994	31/05/2022 00:00:00

Completed exercise

Your final worksheet should look like this:

The screenshot shows the Microsoft Power Query Editor interface with the following details:

- File** tab is selected.
- Home** tab is active.
- Transform** tab is visible in the ribbon.
- Queries [4]** pane on the left lists four queries: Order2022, Order2023, Order Details, and Orders. Order Details is currently selected.
- Table Preview** pane shows a table with columns: SalesOrderID, OrderQty, ProductID, UnitPrice, and OrderDate. The data consists of 997 rows, with the first few rows being: 1, 43659, 1, 776, 2024.994, 31/05/2022; 2, 43659, 3, 777, 2024.994, 31/05/2022; 3, 43659, 1, 778, 2024.994, 31/05/2022; etc.
- Properties** pane on the right shows the query name is "Order Details".
- Applied Steps** pane on the right lists the steps taken: Source, Navigation, Promoted Headers, Changed Type, Removed Columns, Filtered Rows, Merged Queries, Expanded Orders, and Renamed Columns. The "Renamed Columns" step is highlighted.
- Query Settings** pane is partially visible.
- Bottom Status Bar**: 5 COLUMNS, 997 ROWS - Column profiling based on top 1000 rows. PREVIEW DOWNLOADED ON TUESDAY.

Conclusion

Your objective for this exercise was to complete an end-to-end scenario. In this context, you learned how to clean and transform multiple data sources, append, join and merge them and identify the potential anomalies using Power Query.