

# Data Analysis with Power BI

Module 5 (Part 1)

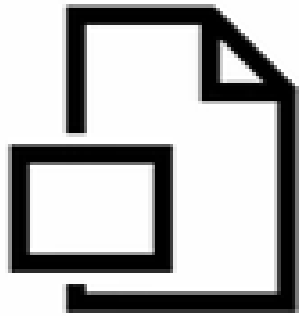
# Structure

- Part 1: Extract, Transform and Load (ETL) – identify, explain and configure multiple data sources in Power BI; clean and transform data using Power Query.
- Part 2: Data Modeling – identify and create appropriate model relationships; configuring your table and column properties; data analysis expressions (DAX) to configure and optimize your models
- Part 3: Data Analysis and Visualization – add visualizations to reports; format visuals

# Part 1

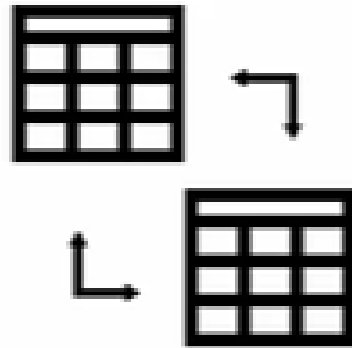
Extract, Transform and Load (ETL)

# Data Sources



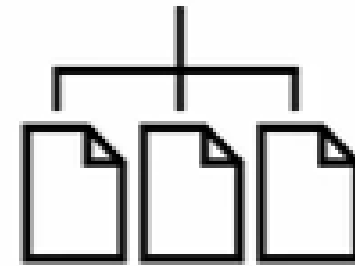
Flat files

CSV, TXT, XLSX



Relational data  
sources

SQL, mySQL, Oracle



NoSQL databases

MongoDB, Cassandra

PowerBI has the flexibility to connect to a wide range of data sources

# Data Sources (cont.)

Data source	Examples
File	XML, JSON, PDF, Excel workbook (Limited to a maximum file size of 1GB)
Database	MySQL, SQL Server, Access, PostgreSQL, Google BigQuery
Power Platform	Microsoft Power Platform Power BI datasets, Datamarts, Dataverse, Dataflows
Azure	Azure SQL Database, Azure Synapse Analytics SQL, Azure Data Explorer
Online Services	GitHub, QuickBooks Online, Stripe, Dynamics 365, Salesforce
Other	R scripts, Python scripts and Active Directory

# Dataset vs Data Source

A **dataset** is what you get when you use the Get Data feature to bring in data from a file, template app, or live source. It includes information about the data source, credentials, and a portion of the data itself. When you create reports and dashboards, you review the data from the dataset.

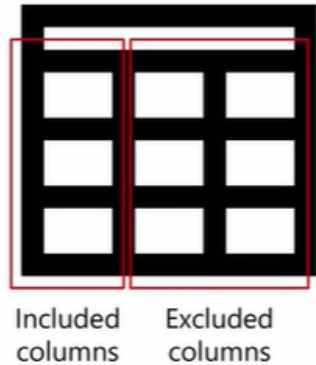
However, the data in the dataset comes from a **data source**. A data source can be an online service, a cloud database, or a local file or server. The data source is where the data comes from. An example of a data source could be a cloud database like Azure SQL, or a file on your computer.

# Exercise: Setting up an Excel data source

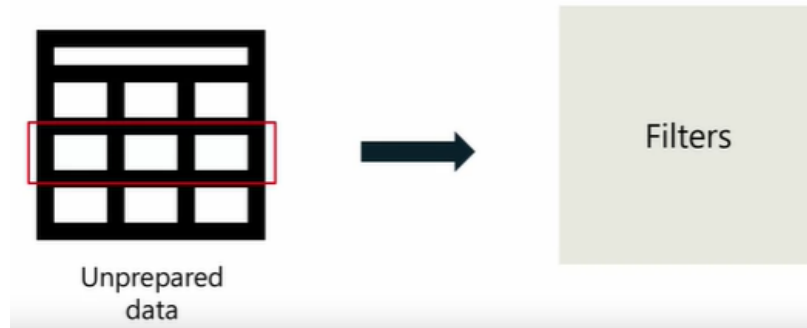
- Choose the required data source from the **Get Data** drop-down list.
- Navigate to the location where your Excel file (*SalesOrderDetail.xlsx*) is saved.
- Choose the worksheet(s) and table(s) you want to import in the Navigator window.
- Click Load. This will import your Excel data into Power BI.

# Data Transformation

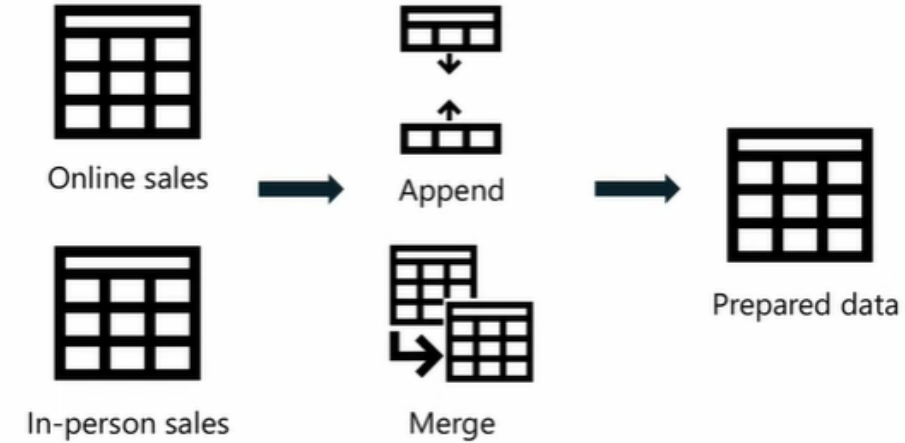
## Excluding data



## Data cleaning



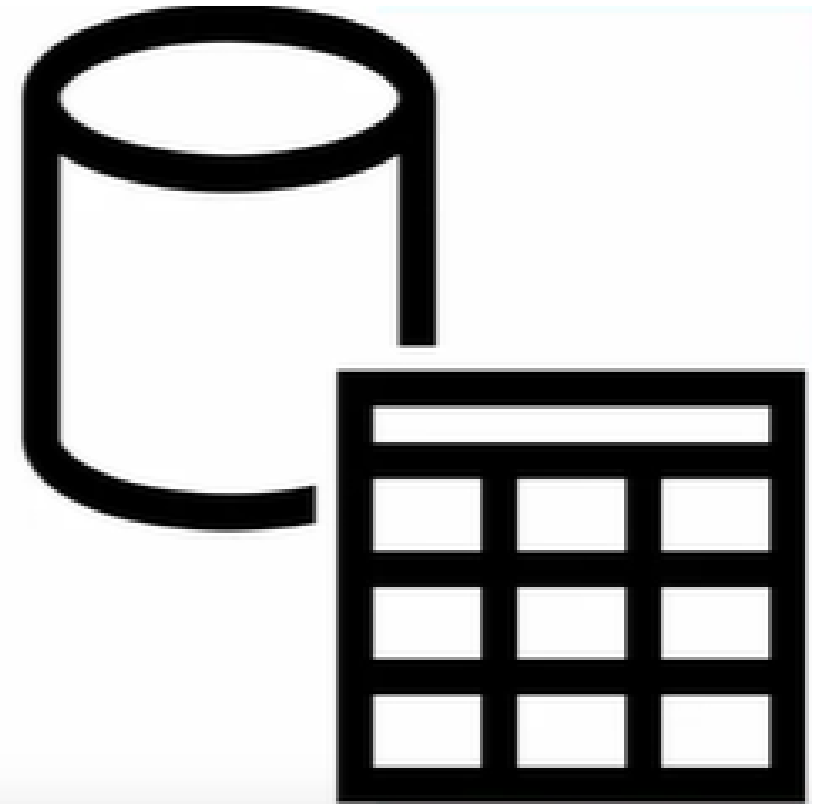
## Data merging





# Power Query

Data transformation  
and preparation tool.



# Power Query – Data Extraction and Transformation

The screenshot displays the Microsoft Power Query Editor interface. The 'Transform' tab is selected in the ribbon, with a red box highlighting it. The ribbon includes various transformation options such as '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', 'Split Column', 'Group By', 'Data Type: Text', 'Use First Row as Headers', 'Replace Values', 'Merge Queries', 'Append Queries', 'Combine Files', 'Text Analytics', 'Vision', and 'Azure Machine Learning'.

The main data view shows a table with columns: Product ID, Product Category, Product Subcategory, Product Name, and Product Description. The 'Product Subcategory' column is selected, and a context menu is open, showing options like 'Sort Ascending', 'Sort Descending', 'Clear Sort', 'Clear Filter', 'Remove Empty', and 'Text Filters'. A red box highlights the 'Sort Ascending' and 'Sort Descending' options. Below these, a list of product names is shown, with a red box highlighting the list. The list includes: (Select All), Adventurer 1000, Adventurer 2000, AeroSpeed 1000, AeroSpeed 2000, CommutePro 1000, CommutePro 2000, CrossRider 1000, CrossRider 2000, DownhillDominator 1000, DownhillDominator 2000, DuoExplorer 1000, DuoExplorer 2000, and E-Mountain 1000.

The 'Query Settings' pane on the right shows the 'Properties' tab with the query name 'Sales'. The 'Applied Steps' list includes 'Source', 'Navigation', 'Promoted Headers', and 'Changed Type'.

At the bottom, the status bar indicates '15 COLUMNS, 48 ROWS' and 'Column profiling based on top 1000 rows'. The bottom right corner shows 'PREVIEW DOWNLOADED AT 11:35 PM'.

# Power Query – Query Reuseability

The screenshot displays the Microsoft Power Query Editor interface. The main area shows a table with 24 rows and 5 columns: Product ID, Product Category, Product Subcategory, Product Name, and Product Description. The 'APPLIED STEPS' pane on the right is highlighted with a red box, showing the sequence of steps: Source, Navigation, Promoted Headers, and Changed Type. The 'Query Settings' pane on the right shows the query name 'Sales' and the 'Properties' section.

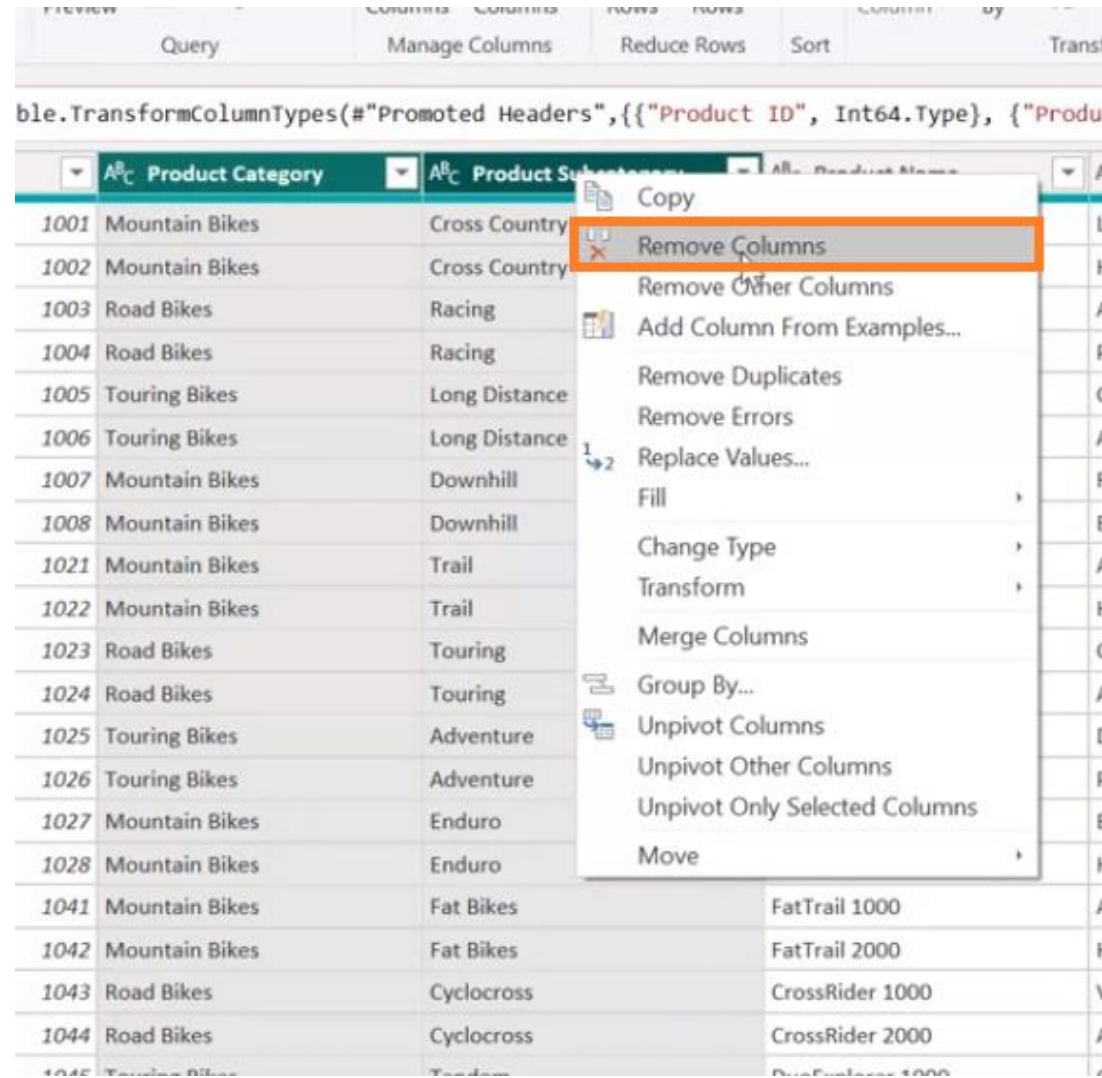
Table Data:

Product ID	Product Category	Product Subcategory	Product Name	Product Description
1001	Mountain Bikes	Cross Country	TrailBlazer 1000	Lightweight and versatile
1002	Mountain Bikes	Cross Country	TrailBlazer 2000	High-performance mountain bike
1003	Road Bikes	Racing	SpeedMaster 1000	Agile and aerodynamic road bike
1004	Road Bikes	Racing	SpeedMaster 2000	Premium racing road bike
1005	Touring Bikes	Long Distance	Explorer 1000	Comfortable and durable touring bike
1006	Touring Bikes	Long Distance	Explorer 2000	Advanced touring bike
1007	Mountain Bikes	Downhill	GravityMaster 1000	Rugged and durable downhill bike
1008	Mountain Bikes	Downhill	GravityMaster 2000	Extreme downhill performance
1021	Mountain Bikes	Trail	Pathfinder 1000	Agile trail bike for all skill levels
1022	Mountain Bikes	Trail	Pathfinder 2000	High-performance trail bike
1023	Road Bikes	Touring	Voyager 1000	Comfortable touring road bike
1024	Road Bikes	Touring	Voyager 2000	Advanced touring road bike
1025	Touring Bikes	Adventure	Adventurer 1000	Durable bike for long adventures
1026	Touring Bikes	Adventure	Adventurer 2000	Premium adventure touring bike
1027	Mountain Bikes	Enduro	EnduroMaster 1000	Endurance-focused mountain bike
1028	Mountain Bikes	Enduro	EnduroMaster 2000	High-performance enduro mountain bike
1041	Mountain Bikes	Fat Bikes	FatTrail 1000	All-terrain fat bike
1042	Mountain Bikes	Fat Bikes	FatTrail 2000	High-performance fat bike
1043	Road Bikes	Cyclocross	CrossRider 1000	Versatile cyclocross bike
1044	Road Bikes	Cyclocross	CrossRider 2000	Advanced cyclocross bike
1045	Touring Bikes	Tandem	DuoExplorer 1000	Comfortable tandem touring bike
1046	Touring Bikes	Tandem	DuoExplorer 2000	High-performance tandem touring bike
1047	Mountain Bikes	Electric	E-Mountain 1000	Electric mountain bike

15 COLUMNS, 48 ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED AT 11:35 PM

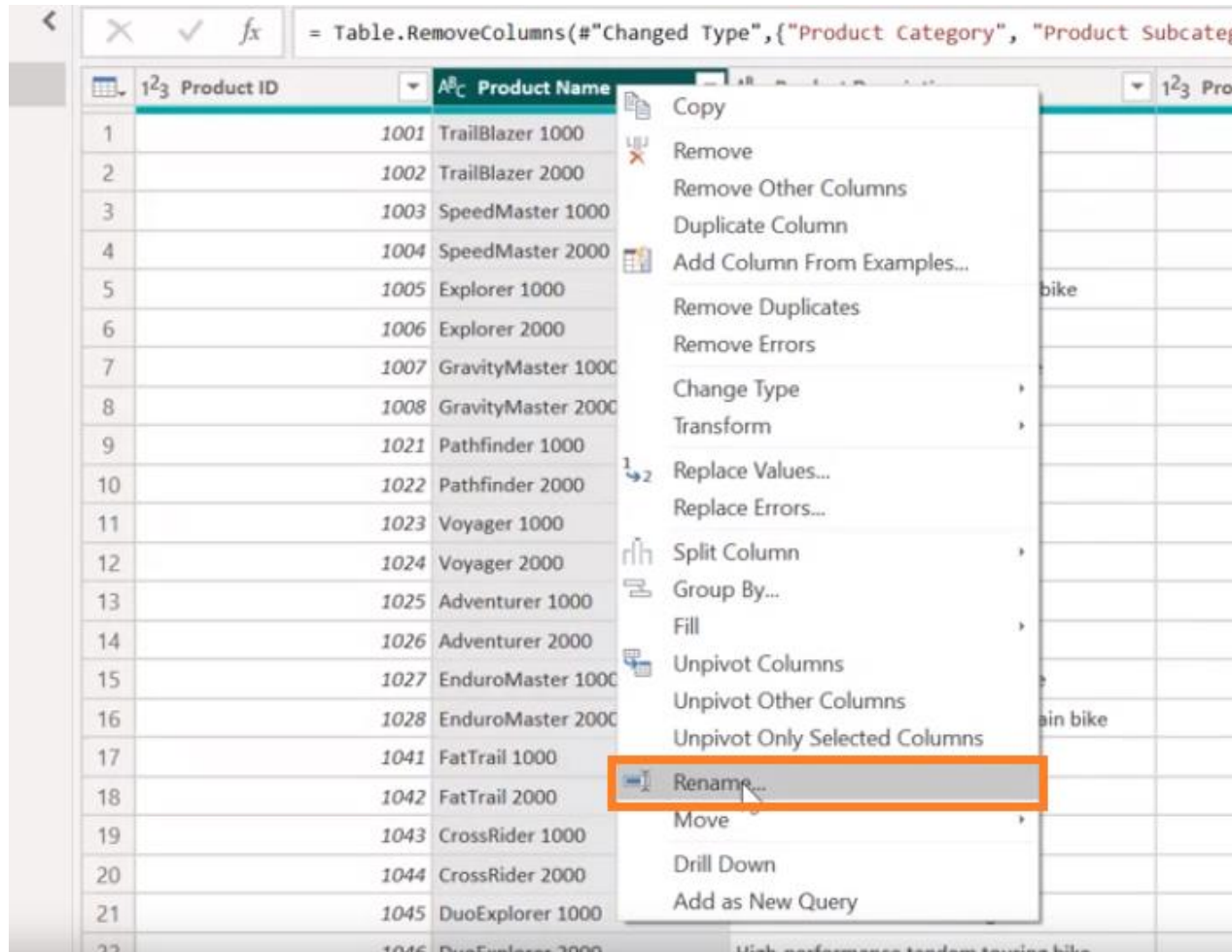
# Working with columns



The screenshot shows a data table with columns: Product ID, Product Category, Product Subcategory, and Product Name. A context menu is open over the 'Product Subcategory' column, with 'Remove Columns' highlighted. The table contains 25 rows of bike data.

Product ID	Product Category	Product Subcategory	Product Name
1001	Mountain Bikes	Cross Country	
1002	Mountain Bikes	Cross Country	
1003	Road Bikes	Racing	
1004	Road Bikes	Racing	
1005	Touring Bikes	Long Distance	
1006	Touring Bikes	Long Distance	
1007	Mountain Bikes	Downhill	
1008	Mountain Bikes	Downhill	
1021	Mountain Bikes	Trail	
1022	Mountain Bikes	Trail	
1023	Road Bikes	Touring	
1024	Road Bikes	Touring	
1025	Touring Bikes	Adventure	
1026	Touring Bikes	Adventure	
1027	Mountain Bikes	Enduro	
1028	Mountain Bikes	Enduro	
1041	Mountain Bikes	Fat Bikes	FatTrail 1000
1042	Mountain Bikes	Fat Bikes	FatTrail 2000
1043	Road Bikes	Cyclocross	CrossRider 1000
1044	Road Bikes	Cyclocross	CrossRider 2000
1045	Touring Bikes	Touring	TrailRider 1000

# Working with columns (cont.)



The screenshot shows a data table with columns 'Product ID' and 'Product Name'. A context menu is open over the 'Product Name' column, listing various actions. The 'Rename...' option is highlighted with an orange rectangle. The formula bar at the top shows a DAX formula: `= Table.RemoveColumns(#"Changed Type",{"Product Category", "Product Subcategory", "Product Color", "Product Material", "Product Weight", "Product Price", "Product Rating", "Product Reviews", "Product Images", "Product Description", "Product Category", "Product Subcategory", "Product Color", "Product Material", "Product Weight", "Product Price", "Product Rating", "Product Reviews", "Product Images", "Product Description" })`.

	Product ID	Product Name
1	1001	TrailBlazer 1000
2	1002	TrailBlazer 2000
3	1003	SpeedMaster 1000
4	1004	SpeedMaster 2000
5	1005	Explorer 1000
6	1006	Explorer 2000
7	1007	GravityMaster 1000
8	1008	GravityMaster 2000
9	1021	Pathfinder 1000
10	1022	Pathfinder 2000
11	1023	Voyager 1000
12	1024	Voyager 2000
13	1025	Adventurer 1000
14	1026	Adventurer 2000
15	1027	EnduroMaster 1000
16	1028	EnduroMaster 2000
17	1041	FatTrail 1000
18	1042	FatTrail 2000
19	1043	CrossRider 1000
20	1044	CrossRider 2000
21	1045	DuoExplorer 1000
22	1046	DuoExplorer 2000

Context Menu Options:

- Copy
- Remove
- Remove Other Columns
- Duplicate Column
- Add Column From Examples...
- Remove Duplicates
- Remove Errors
- Change Type
- Transform
- Replace Values...
- Replace Errors...
- Split Column
- Group By...
- Fill
- Unpivot Columns
- Unpivot Other Columns
- Unpivot Only Selected Columns
- Rename...**
- Move
- Drill Down
- Add as New Query



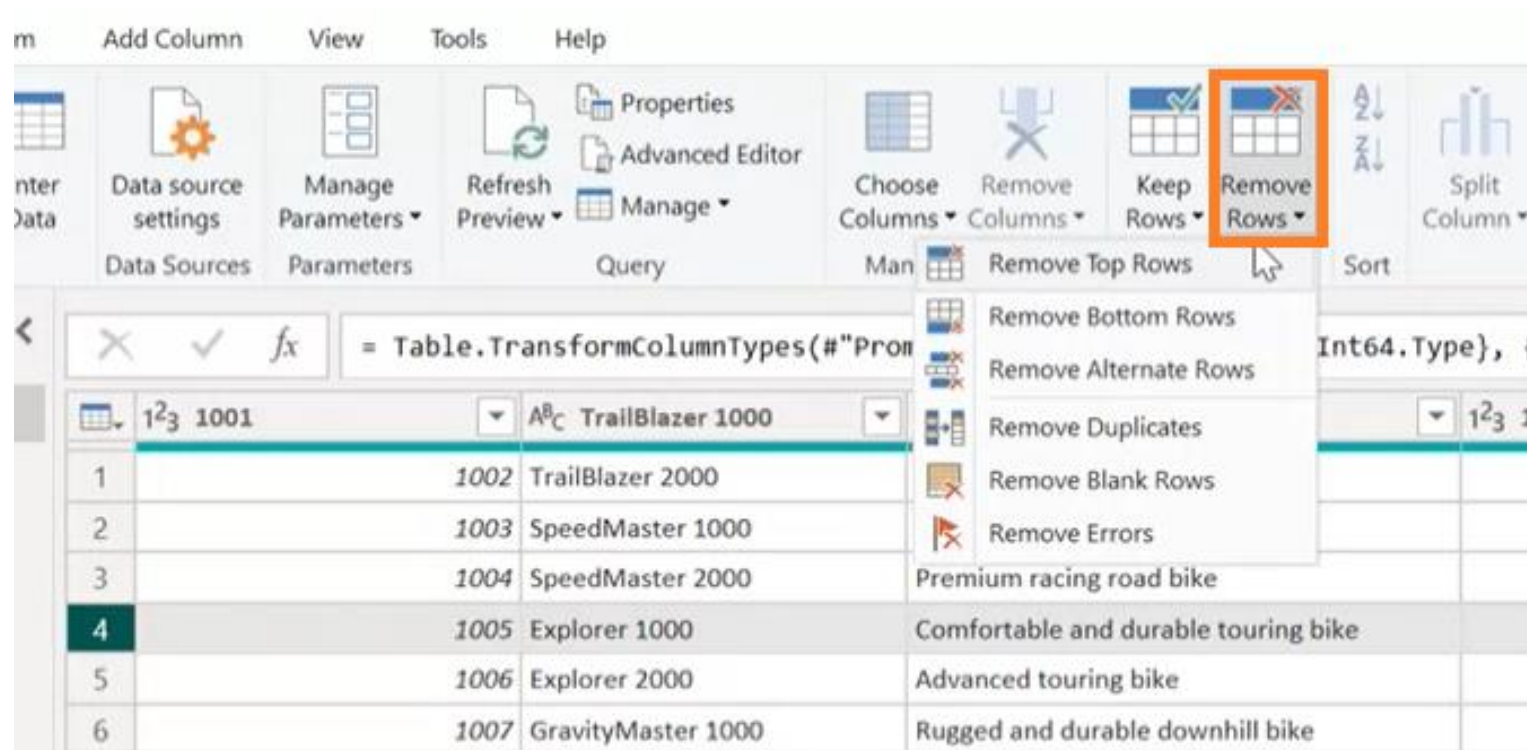
# Working with columns (cont.)

The screenshot shows the Power Query Editor interface. The ribbon at the top includes tabs for Transform, Add Column, View, Tools, and Help. The 'Transform' tab is active, and the 'Use First Row as Headers' option is highlighted in the 'Manage Columns' group. The main area displays a table with the following data:

	Product ID	Product Name Final	Product Description	Product Price	Product Weight
1	1001	TrailBlazer 1000	Lightweight and versatile	1200	
2	1002	TrailBlazer 2000	High-performance mountain bike	1500	
3	1003	SpeedMaster 1000	Agile and aerodynamic road bike	1800	
4	1004	SpeedMaster 2000	Premium racing road bike	2100	
5	1005	Explorer 1000	Comfortable and durable touring bike	1300	
6	1006	Explorer 2000	Advanced touring bike	1600	
7	1007	GravityMaster 1000	Rugged and durable downhill bike	2200	
8	1008	GravityMaster 2000	Extreme downhill performance	2500	
9	1021	Pathfinder 1000	Agile trail bike for all skill levels	1100	
10	1022	Pathfinder 2000	High-performance trail bike	1400	
11	1023	Voyager 1000	Comfortable touring road bike	1700	
12	1024	Voyager 2000	Advanced touring road bike	2000	
13	1025	Adventurer 1000	Durable bike for long adventures	1500	
14	1026	Adventurer 2000	Premium adventure touring bike	1800	
15	1027	EnduroMaster 1000	Endurance-focused mountain bike	2300	

The formula bar shows the query step: `= Table.RenameColumns(#'Removed Columns',{{"Product Name", "Product Name Final"}})`. The right sidebar shows the 'APPLIED STEPS' list with 'Renamed Columns' selected.

# Working with columns (cont.)



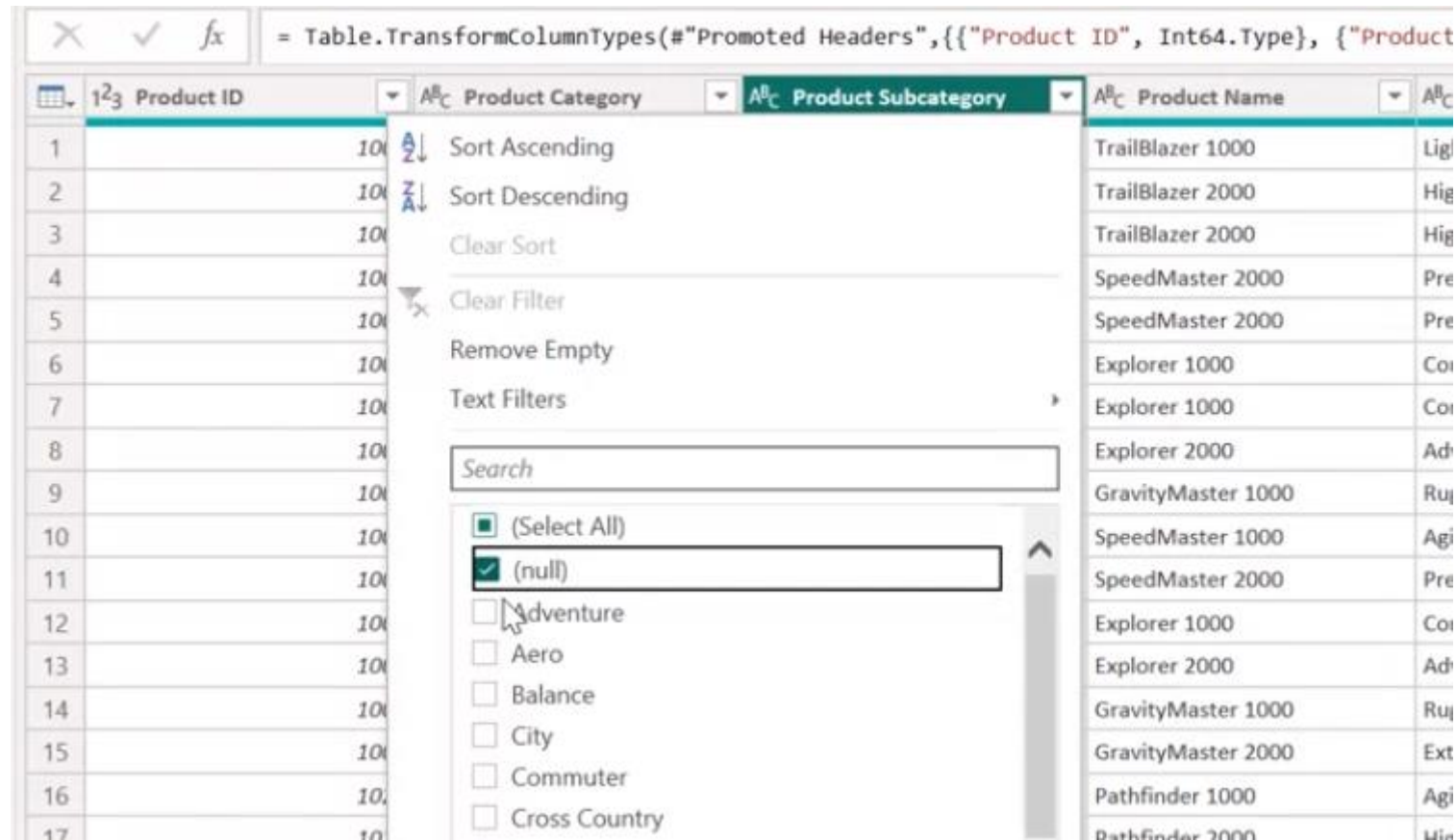
The screenshot shows the Power BI Desktop interface. The ribbon is set to the 'Query' tab, and the 'Remove Rows' button is highlighted with an orange box. A dropdown menu is open, showing options to remove rows from the table. The table below contains data for various bicycle models.

	123 1001	ABC TrailBlazer 1000
1	1002	TrailBlazer 2000
2	1003	SpeedMaster 1000
3	1004	SpeedMaster 2000
4	1005	Explorer 1000
5	1006	Explorer 2000
6	1007	GravityMaster 1000

The dropdown menu for 'Remove Rows' includes the following options:

- Remove Top Rows
- Remove Bottom Rows
- Remove Alternate Rows
- Remove Duplicates
- Remove Blank Rows
- Remove Errors

# Dealing with errors in Power Query





# Dealing with errors in Power Query (cont.)

Untitled - Power Query Editor

Transform | Add Column | View | Tools | Help

Rows | Table

Transpose | Reverse Rows | Count Rows

Data Type: Text | Detect Data | Rename

Replace Values | Replace Errors

Unpivot Columns | Move | Convert to List

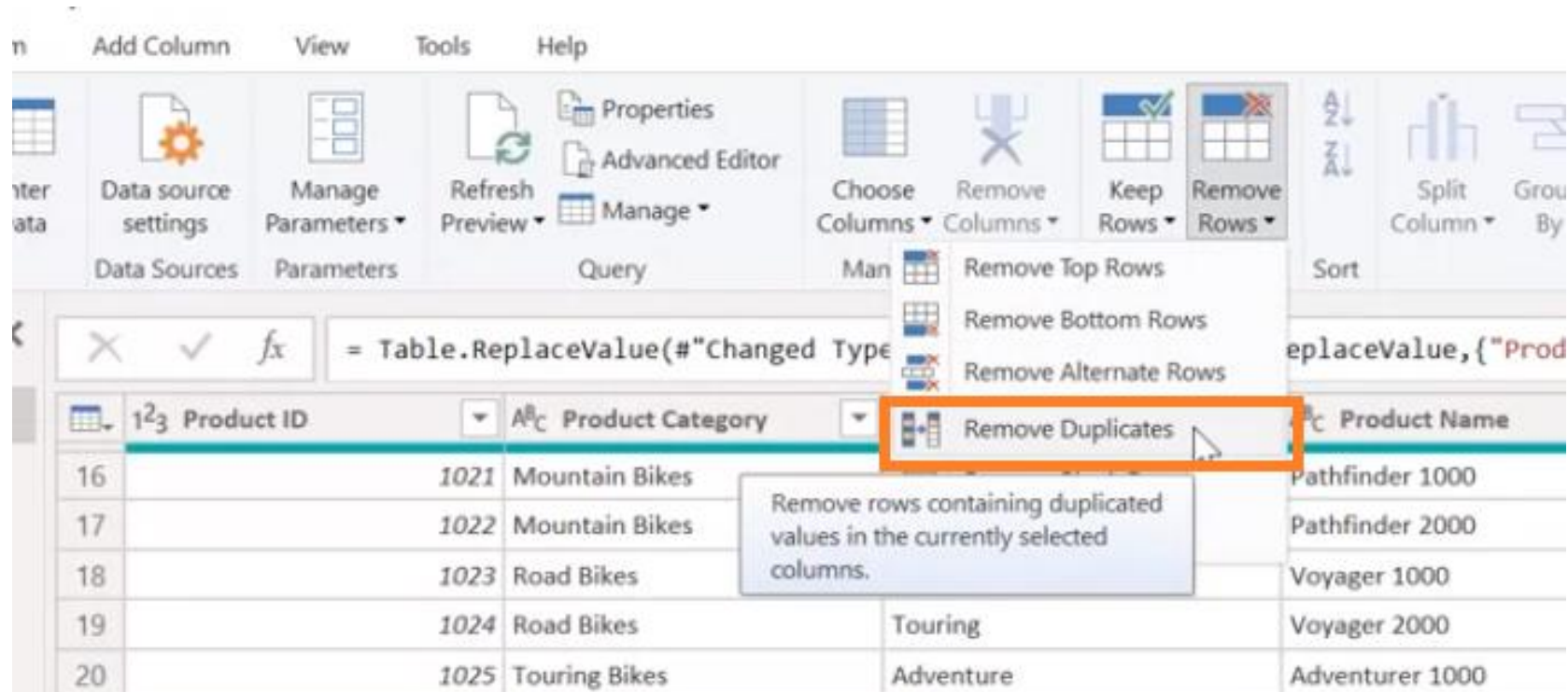
Split Column | Format | Merge Columns | Extract | Parse

Statist

fx = Table.TransformColumnTypes(#"Promoted Headers",{{"Product ID", Int64.

	Product ID	Product Category	Product Subcategory	Product Name
1	1001	Mountain Bikes	Cross Country	TrailBlazer 10
2	1002	Mountain Bikes	Cross Country	TrailBlazer 20
3	1002	Mountain Bikes	Cross Country	TrailBlazer 20
4	1004	Road Bikes	Racing	SpeedMaster
5	1004	Road Bikes	Racing	SpeedMaster
6	1005	Touring Bikes	Long Distance	Explorer 1000
7	1005	Touring Bikes	Long Distance	Explorer 1000

# Dealing with errors in Power Query (cont.)



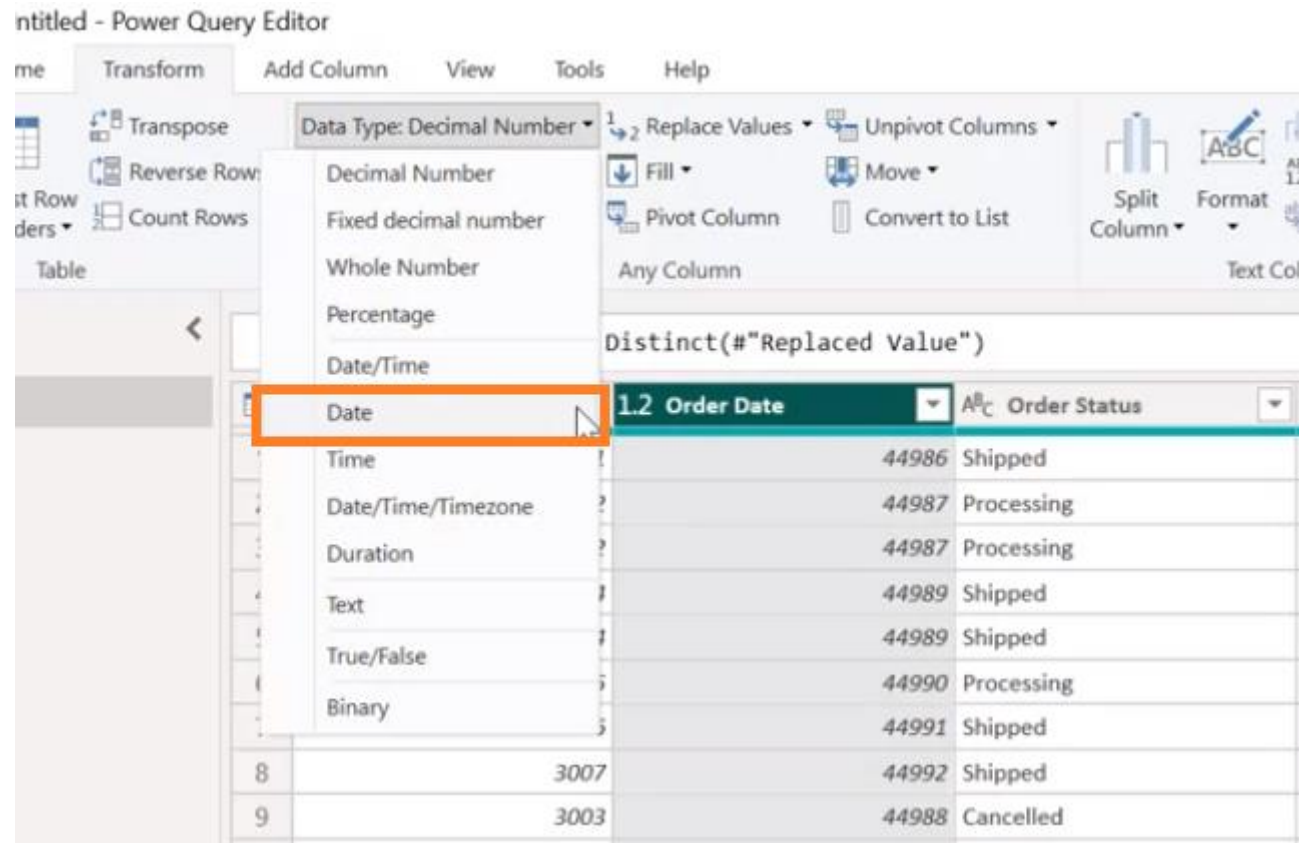
The screenshot shows the Power Query ribbon with the 'Remove Rows' dropdown menu open. The 'Remove Duplicates' option is highlighted with an orange box. A tooltip explains: 'Remove rows containing duplicated values in the currently selected columns.'

The ribbon includes the following tabs: Data, Add Column, View, Tools, and Help. The 'Data' tab is active, showing options like 'Data source settings', 'Manage Parameters', 'Refresh Preview', 'Properties', 'Advanced Editor', 'Manage', 'Choose Columns', 'Remove Columns', 'Keep Rows', and 'Remove Rows'. The 'Remove Rows' dropdown menu is open, showing options: 'Remove Top Rows', 'Remove Bottom Rows', 'Remove Alternate Rows', and 'Remove Duplicates' (highlighted).

The formula bar shows: `= Table.ReplaceValue(#"Changed Type", " ", "", ReplaceWith, Trim)`

	Product ID	Product Category	Product Name
16	1021	Mountain Bikes	Pathfinder 1000
17	1022	Mountain Bikes	Pathfinder 2000
18	1023	Road Bikes	Voyager 1000
19	1024	Road Bikes	Voyager 2000
20	1025	Touring Bikes	Adventure

# Dealing with errors in Power Query (cont.)



# Exercise: Preparing a Dataset

## **Load the workbook**

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

## **Open the Power Query Editor**

Click on the Transform Data button to open the Power Query Editor.

# Exercise: Preparing a Dataset (cont.)

## 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.

# Exercise: Preparing a Dataset (cont.)

## **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.

# Exercise: Preparing a Dataset (cont.)

## 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**.

# Exercise: Preparing a Dataset (cont.)

## 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**.



# Exercise: Preparing a Dataset (cont.)

## 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 03<sup>rd</sup> 2023**.
3. Next, select the **Month Number** column. Change the column's data type to **Whole Number**.

# Exercise: Preparing a Dataset (cont.)

## Drop records with errors

1. Select the **Manufacturing Price** column. The column contains errors in rows **6** and **38**. Drop these rows.
2. Repeat the same steps for the errors in the **Sales** and **Profit** columns.

## Drop duplicate rows

Use the **Drop Duplicates** feature to remove duplicate rows.

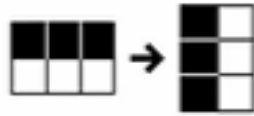
# Exercise: Preparing a Dataset (cont.)

## **Apply the data transformations**

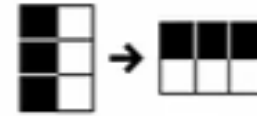
Once you have completed all the above data cleaning steps, select the **Close & Apply** button on the top left.

# Un-pivot and Pivot Columns

Unpivot

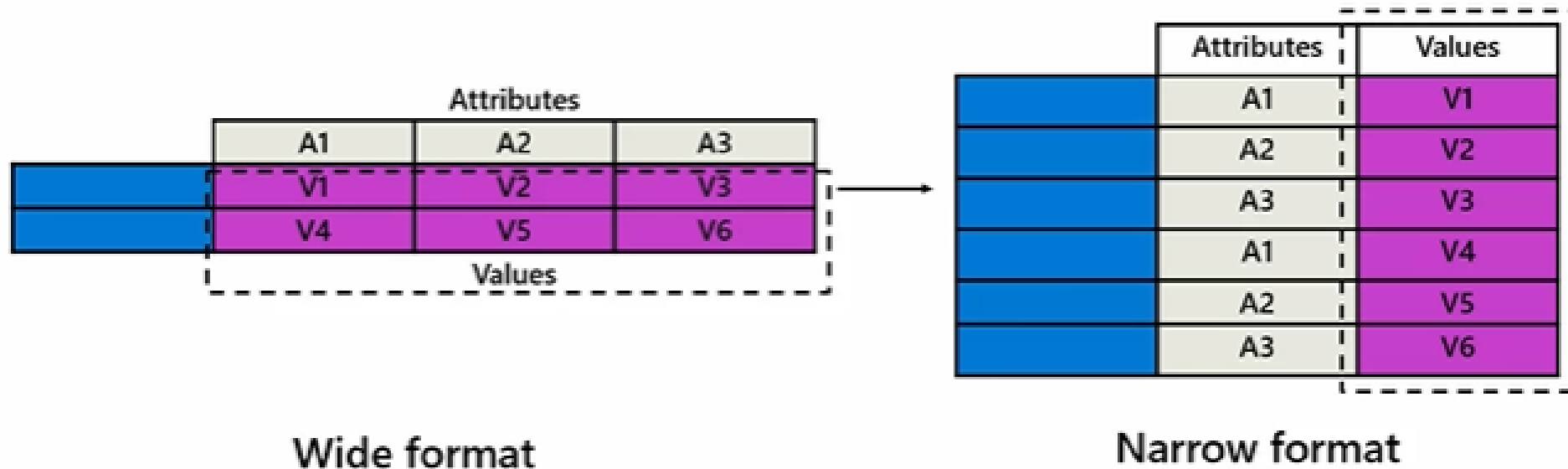


Pivot



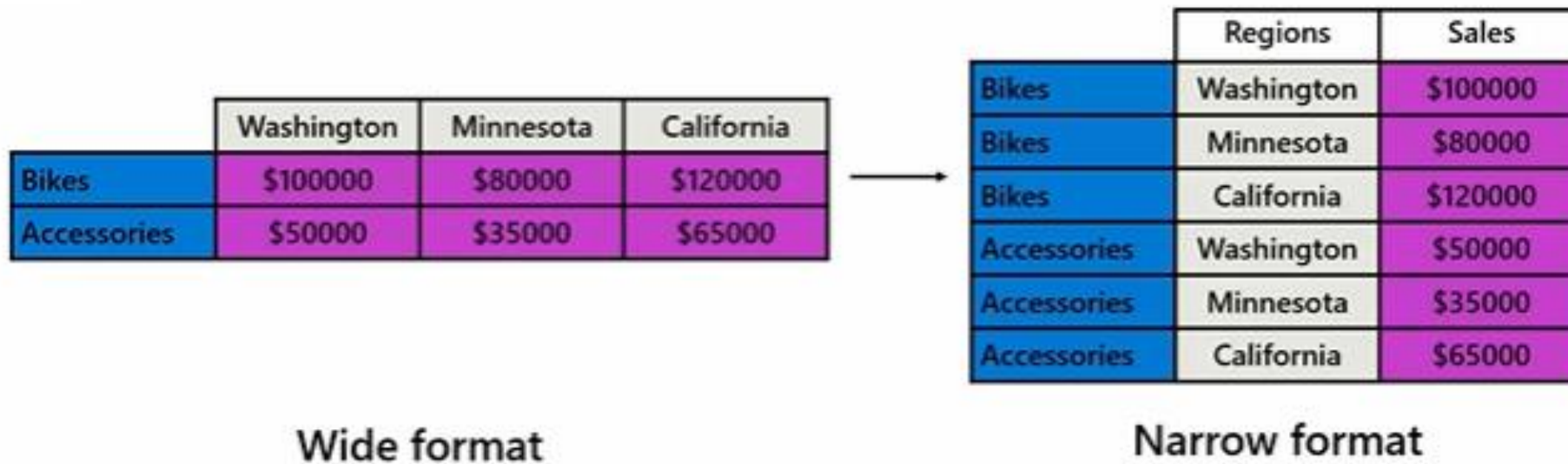
# Un-pivot and Pivot Columns (cont.)

## Unpivot columns



# Un-pivot and Pivot Columns (cont.)

## Unpivot columns



# Un-pivot and Pivot Columns (cont.)

## Pivot columns

	Products	Sales
	Bikes	\$100000
	Accessories	\$50000
	Components	\$25000
	Bikes	\$80000
	Accessories	\$35000
	Components	\$15000

Narrow format



	Bikes	Accessories	Components
	\$100000	\$50000	\$25000
	\$80000	\$35000	\$15000

Wide format

# Activity: Apply a pivot

## 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*.

## Import Excel data

1. Import the **Excel data** to add the **Color Model** query to the Queries pane.
2. Observe the 3 columns in the table: **Product Name**, **Color** and **Model**.
3. Remove the **Product Name** column.



# Activity: Apply a pivot (cont.)

## Pivot columns

1. To pivot the table columns, select the **Color Model** query on the left menu.
2. Select the **Transform ribbon** tab, followed by **Pivot Column**.
3. On the **Pivot Column** window that displays, select **Model** as the Values Column.
4. Expand the **Advanced options** and select option **Count (All)** from the **Aggregate Value Function** dropdown list, and then select **OK**.

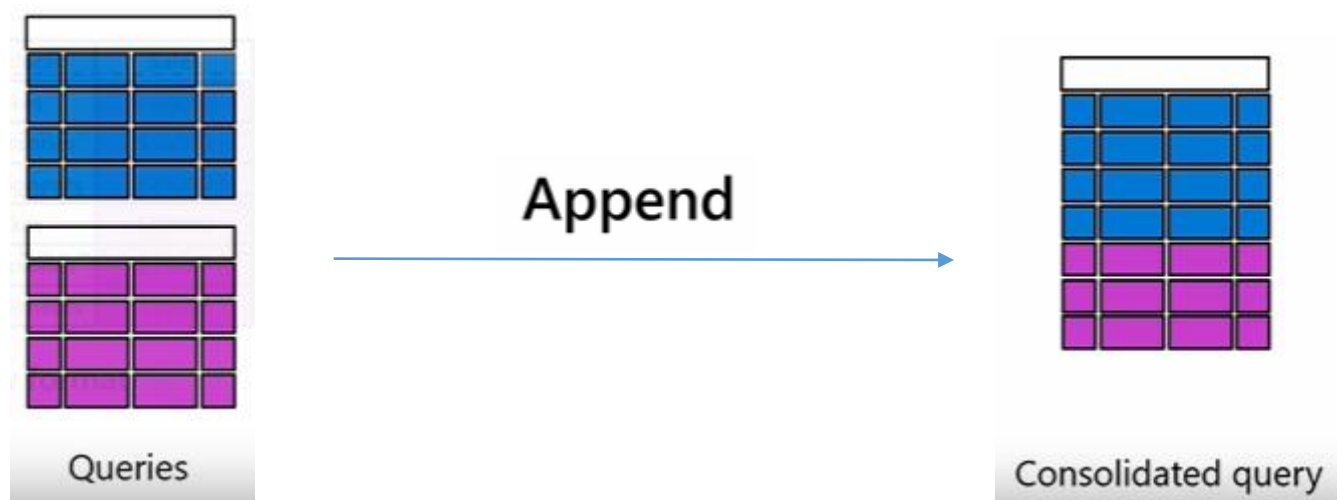
# Quiz

- Which of the following operations are steps in the data transformation process? Select all that apply.
  - a) Creating insights from data
  - b) Shaping data
  - c) Removing data
  - d) Cleaning data
- Which of the following data types are part of the number type group? Select all that apply.
  - a) Fixed decimal number
  - b) Binary
  - c) Whole number
  - d) Text

# Quiz (cont.)

- Which one of the following features are used to track, re-order or delete the steps completed in Power Query?
  - a) Applied Steps
  - b) Queries
  - c) New Source
  - d) Properties
- Which of the following options can be used for Power Query Optimization? Select all that apply.
  - a) Filter rows in the queries.
  - b) Choose only the columns that you will use in the data model.
  - c) Choose the right data types for columns.

# Combining tables with append



# Exercise: Appending two tables

## **Download the Excel files**

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

## **Open the Power Query Editor**

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

# Exercise: Appending two tables (cont.)

## Format Excel files

1. 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.
2. 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**.

# Exercise: Appending two tables (cont.)

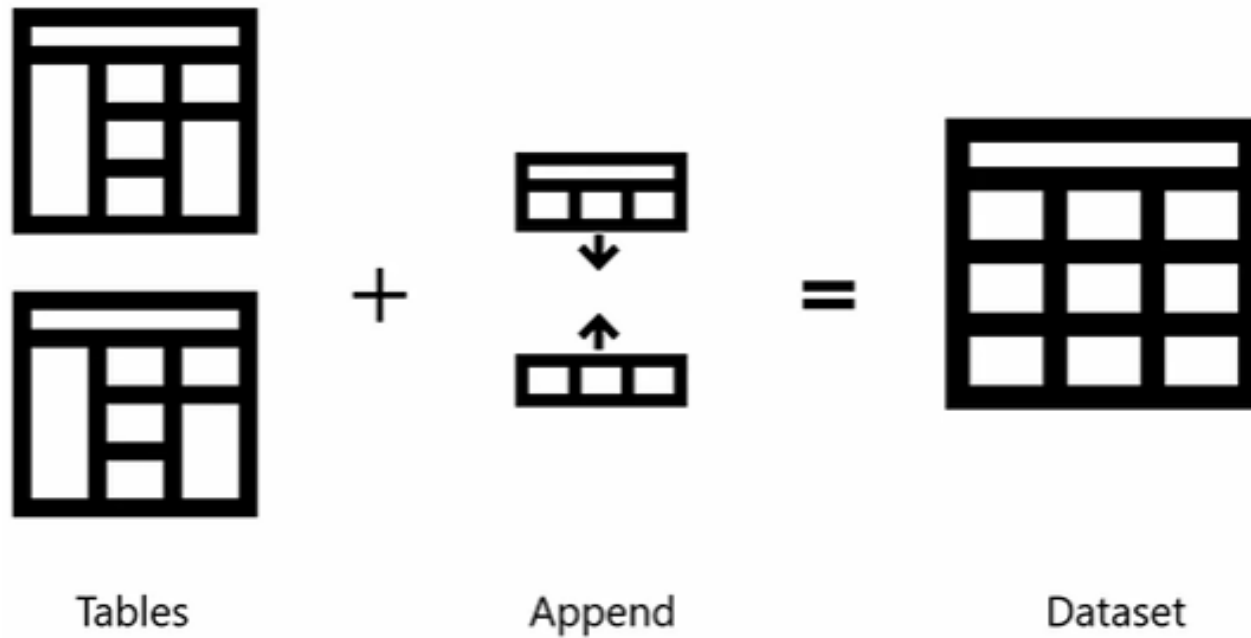
## **Append queries**

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

## **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**.

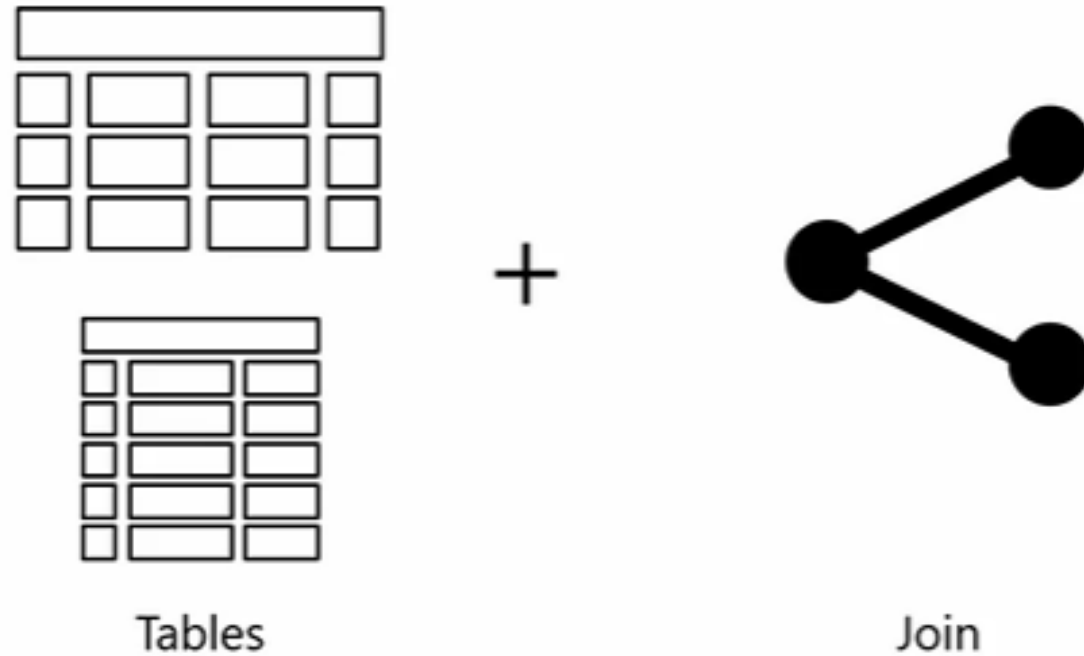
# What is a join?



When you have two tables with the same structure, merging them is straightforward.

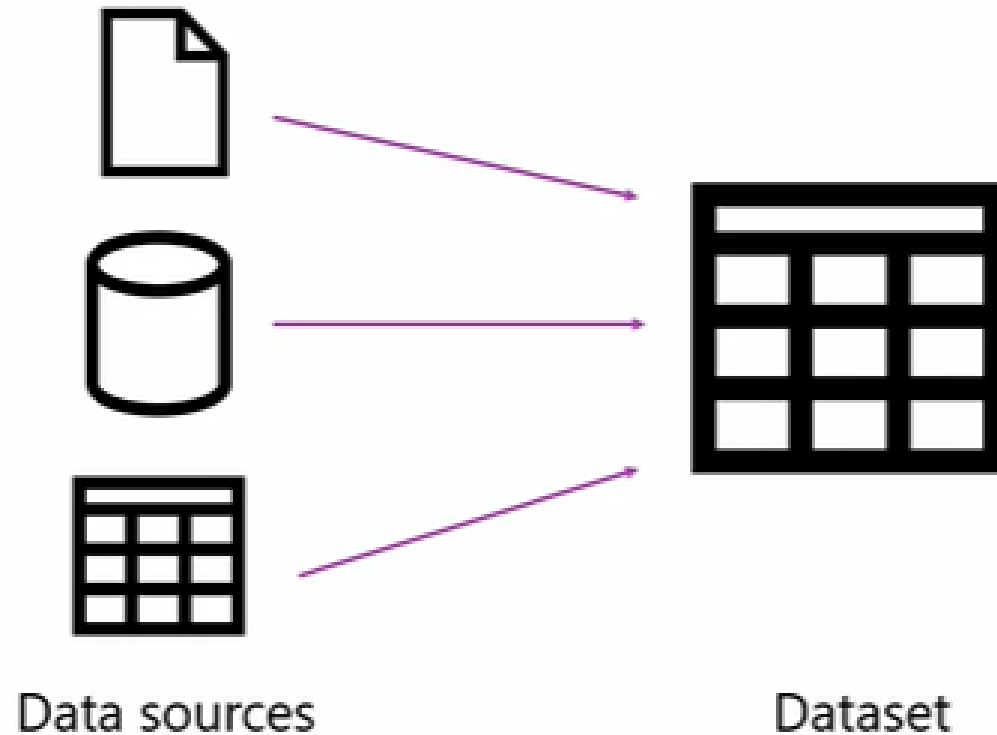


# What is a join? (cont.)



When you have two tables with different structures, you need to specify the method of joining them.

# What is a join? (cont.)



'Join' is when you combine multiple data sources to get a bigger dataset.

# What is a join? (cont.)

Products	
CategoryKey	
1	
2	

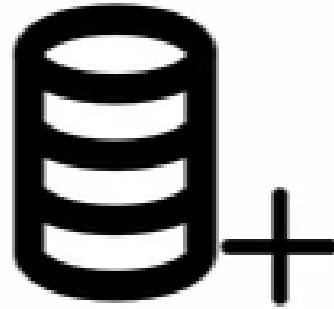
Categories	
CategoryKey	CategoryName
1	Bikes

Here, you have two tables with the same column, but the column has different distributions in the two tables. Why is that?

# What is a join? (cont.)



Match related data



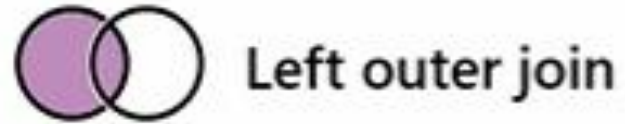
Integrate data



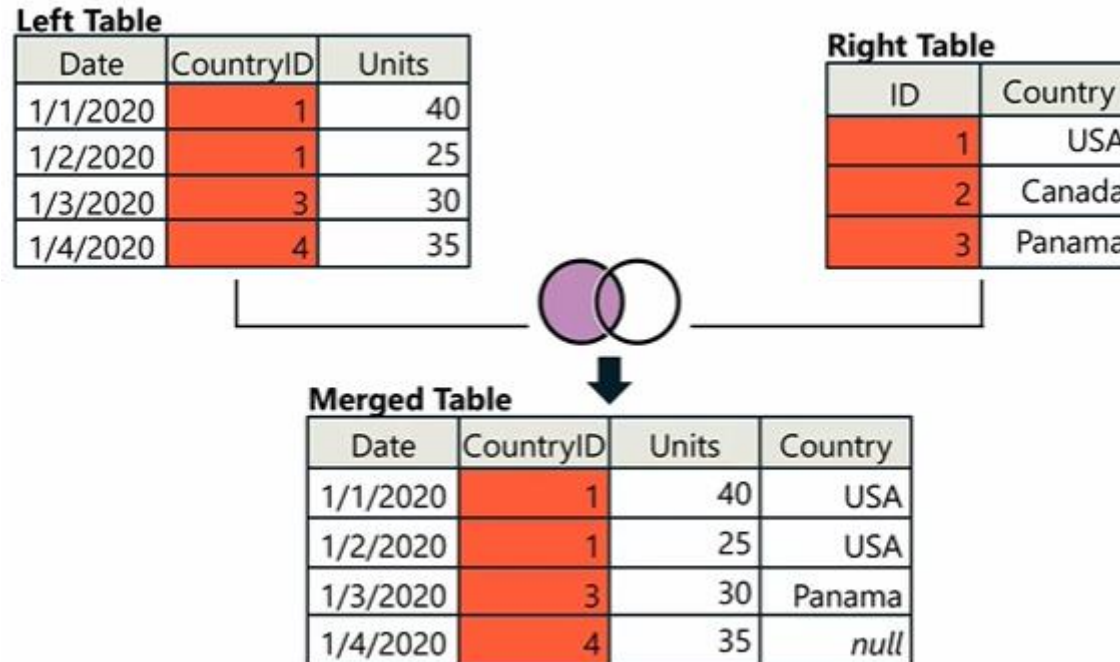
Relationships

When you merge data with join, it allows you to match and integrate related data.  
It also allows you explore relationships between tables.

# Join Types

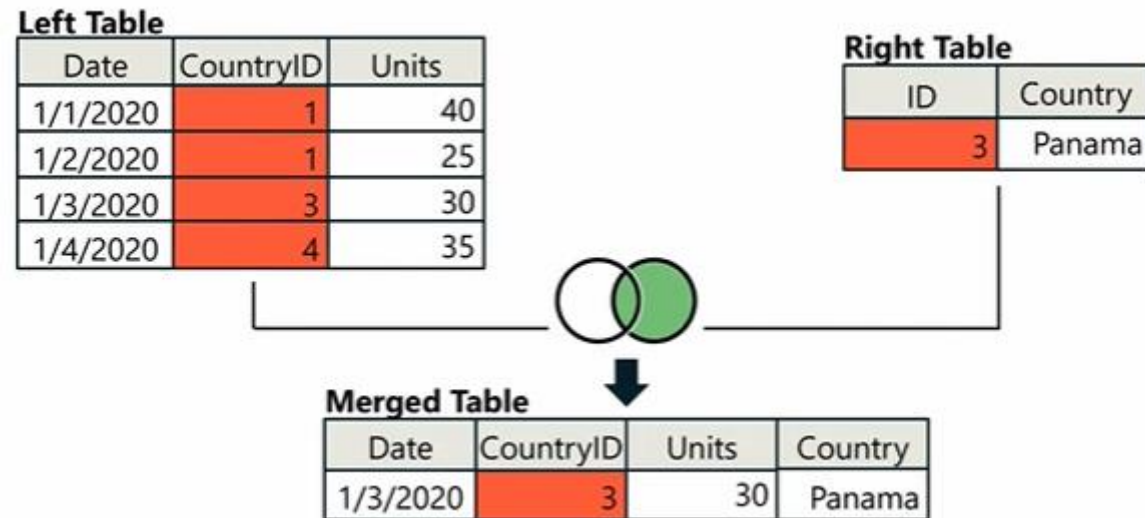


# Join Types: Left Outer



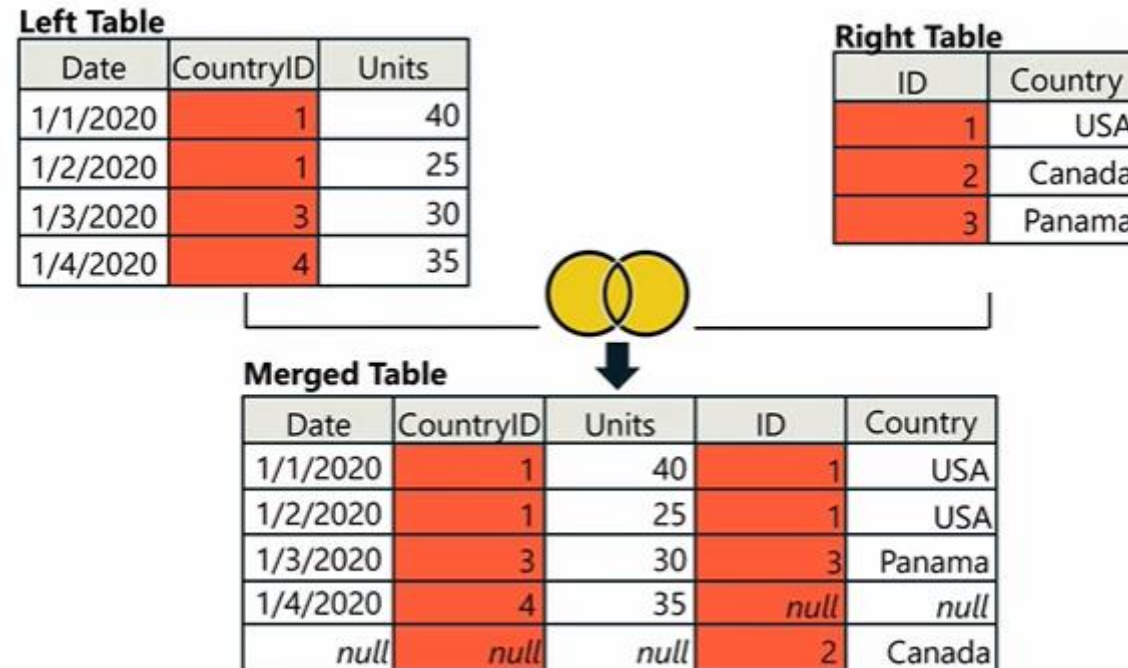
If there is no match between the tables, default/null values will be used.

# Join Types: Right Outer



If there is no match between the tables, default/null values will be used.

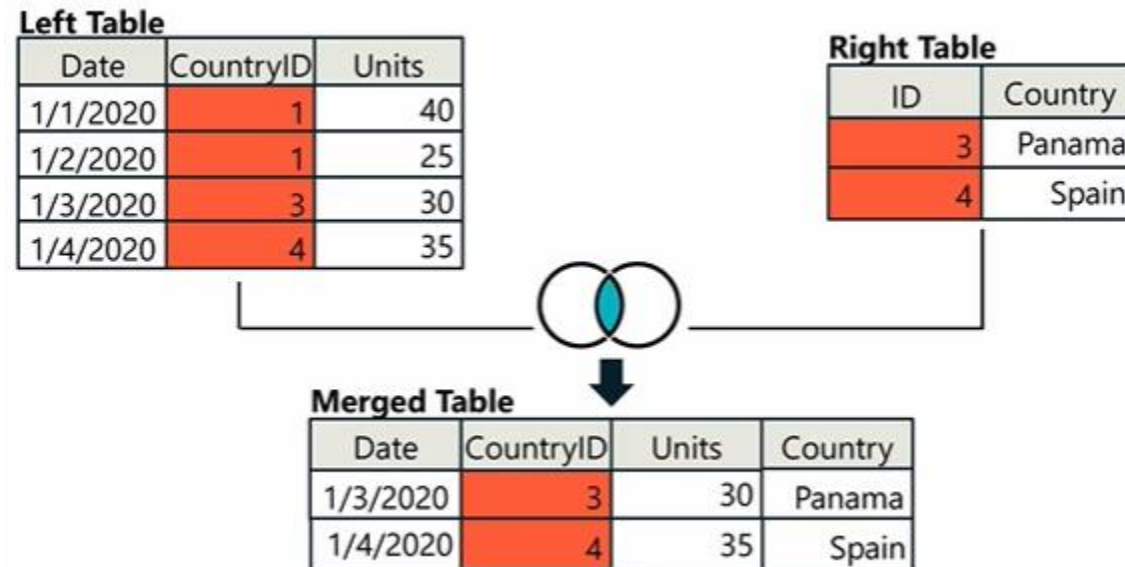
# Join Types: Full Outer



The full outer join is used when you want to retrieve all records from both tables, regardless of whether they have matching values in the join condition.



# Join Types: Inner



For inner join, only matching rows from both left and right tables are merged together.

# Exercise: Merging two data sources

## **Download the Excel files**

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

## **Create a Power BI project**

1. Create a Power BI project and open the Power Query editor.
2. Import your datasets, *Sales* and *Product*.

# Exercise: Merging two data sources (cont.)

## 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.

# Exercise: Merging two data sources (cont.)

## 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.

# Exercise: Merging two data sources (cont.)

## 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**.

# Quiz

- Which feature allows you to combine related data between differently structured data sources in Power Query?
  - a) Appending
  - b) Merging
  - c) Grouping
- Which of the following can be considered as a purpose of merging data with joins? Select all that apply:
  - a) Integrating Data
  - b) Exploring Relationships
  - c) Expanding Data
  - d) Matching Related Data

# Quiz (cont.)

- The full outer join is useful when you want to retrieve all the records from both tables, regardless of whether they have matching values in the join condition. True or False?
- You import two Microsoft Excel tables named *Product* and *Categories* into Power Query. There are 319 rows in the *Product* table. Nine of the total rows in the *Product* table do not have Categories data, so the **CategoryKey** of these rows has **NULL** values. Your manager asked you to list Product data by showing their category names including the rows which have NULL values in **CategoryKey** column. What should you do to accomplish this task?
  - a) Merge *Product* and *Categories* tables based on **ResellerKey** column.
  - b) Merge *Product* and *Categories* tables based on **CategoryKey** column by choosing **Inner Join** in the join kind dropdown.
  - c) Merge *Product* and *Categories* tables based on **CategoryKey** column by choosing **Left Outer Join** in the join kind dropdown.