

## Lesson-2

1. List three data sources Power BI can connect to.
2. What is the first step to import data into Power BI Desktop?
3. How do you refresh imported data in Power BI?
4. What file formats can Power BI import directly? (Name two.)
5. What does the "Navigator" window show after selecting a data source?
6. Import Sales\_Data.csv and load only the "Product" and "Price" columns.
7. How would you change OrderDate to a date format during import?
8. What is the difference between "Load" and "Transform Data" in the import dialog?
9. Why might you see an error when connecting to a SQL database? (Name one reason.)
10. How do you replace a data source after importing it?
11. Write the M-code to import only rows where Quantity > 1.
12. How would you change the data source if Sales\_Data.csv changed?
13. Troubleshoot: Your CSV import fails due to a "mixed data type" error—how do you fix it?
14. Connect to a live SQL database with parameters (e.g., filter by year).
15. How would you automate data imports using Power BI and Power Automate?

## Answers

1. Power BI can connect to a wide variety of data sources. Here are three common ones:

Excel/CSV Files – Power BI can import data from Excel workbooks (.xlsx, .xls) or Comma-Separated Values (.csv) files.

SQL Database – It can connect to relational databases like Microsoft SQL Server, Azure SQL Database, MySQL, PostgreSQL, and others.

Cloud Services – Power BI integrates with cloud platforms such as Azure Blob Storage, Google Analytics, Salesforce, and SharePoint Online.

2. The first step to import data into Power BI Desktop is:

1. Open Power BI Desktop and Click "Get Data"

Launch Power BI Desktop.

On the Home tab, click "Get Data" (or use the shortcut Ctrl + G).

This opens the "Get Data" window, where you can select your data source (Excel, SQL, Web, etc.).

Next Steps:

Choose your data source (e.g., Excel, SQL Server, Web).

Configure the connection (file path, server name, credentials, etc.).

Load or transform the data in Power Query Editor before importing.

### 3. Refresh in Power BI Desktop

Manual Refresh:

Click Refresh on the Home tab (or press Ctrl + R).

To refresh a specific table, right-click it in the Fields pane and select Refresh data.

Advanced Options:

Go to Transform data > Data source settings to modify credentials or connection details before refreshing.

4. Excel Files (.xlsx, .xls) – Power BI can import tables, sheets, or data models from Excel workbooks.

CSV Files (.csv) – Comma-separated values files are supported for simple tabular data import

### 5. Tables/Sheets/Entities:

For Excel: Lists all worksheets and named ranges.

For SQL Databases: Shows tables, views, and sometimes stored procedures.

For CSV/Text: Displays the file as a single table.

For APIs/Folders: Lists endpoints or subfolders (e.g., JSON files in a folder).

### 6. Power Query Editor Opens

In the "Home" tab, click "Choose Columns".

A window will pop up listing all columns (e.g., "Product," "Price," "Date," "Quantity").

Check only "Product" and "Price" → Click OK.

Remove Other Columns (Alternative Method)

If you prefer, you can:

Select the "Product" and "Price" columns (hold Ctrl to multi-select).

Right-click → "Remove Other Columns".

Apply & Load

Click "Close & Apply" in the Home tab to load only the selected columns into Power BI.

7. Import the Data

Click "Get Data" → Select your data source (e.g., CSV, SQL, Excel) → Load the file/table.

In the Navigator window, click "Transform Data" to open the Power Query Editor.

Change Column Data Type

In the Power Query Editor:

Select the OrderDate column.

Go to the "Transform" tab → Click "Data Type" → Choose "Date" (or "Date/Time" if time is included).

Alternatively: Right-click the column → "Change Type" → Select "Date".

8. Key Comparison

Feature	"Load"	"Transform Data" Data
Data Modifications	None (raw import)	Full editing in Power Query Editor
Performance	Faster (good for simple data)	Slower (but essential for cleaning)
Automation	No transformations saved	Transformations auto-apply on refresh
Use Case	"Ready-to-use" data	"Fix it first" data

9. Incorrect or Expired Credentials

Why it happens:

Power BI requires valid authentication (e.g., username/password, Windows auth) to access the database.

If credentials are mistyped, expired, or lack proper permissions, the connection fails.

How to fix it:

Re-enter credentials in the Data Source Settings (ensure caps lock is off).

Verify the SQL user has read permissions on the target tables.

For Windows auth, ensure your account has database access.

10. Open Power BI Desktop.

Go to the Home tab → Click "Transform data" → Select "Data source settings".

In the pop-up window:

Select the data source you want to replace → Click "Change Source...".

Update the server name, database, file path, or credentials.

Click OK → Close.

Refresh the Data:

Click "Refresh" on the Home tab to pull data from the new source.

11. To import only rows where **Quantity > 1** from your data source using **Power Query's M language**, you can add a filter step in the **Advanced Editor**. Here's the M code for different scenarios:

### **1. For CSV/Excel Files**

let

    // Import the file (replace path with your file location)

    Source = Csv.Document(File.Contents("C:\Data\Sales\_Data.csv"),

    // Promote headers (if first row contains column names)

    Headers = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),

    // Filter rows where Quantity > 1 (change "Quantity" to your column name)

    FilteredRows = Table.SelectRows(Headers, each [Quantity] > 1)

in

    FilteredRows

12. Update Source Path via Power Query Editor (GUI)

Open Power Query Editor

In Power BI Desktop, go to the Home tab → Click Transform data → Transform data.

Modify the Source

In the Queries pane, select your Sales\_Data query.

Go to the Home tab → Click Data source settings → Select the CSV source → Click Change Source....

Browse to the new file location (e.g., updated path or filename like Sales\_Data\_v2.csv).

Click OK → Close.


Apply Changes

Click Close & Apply to reload data from the new file.

### 13. **Open Power Query Editor**

- Go to **Home** → **Transform data** → **Transform data**.

### 2. **Identify the Problem Column**

- Look for the column flagged with an error (  icon) in the preview.
- Common culprits: `Price`, `Quantity`, or `Date` columns with mixed formats (e.g., `"100"` vs. `"\$100"`).

### 3. **Standardize the Data Type**

- **Option 1: Force a Specific Type**
  - Select the column → Go to **Transform** → **Data Type** → Choose **Decimal** (for numbers) or **Text** (if mixed).
  - **Note**: Forcing a type may convert incompatible values to `null` (e.g., `"N/A"` becomes blank).
- **Option 2: Clean the Data First**
  - Use **Replace Values** to standardize entries (e.g., replace `"\$"` or `"N/A"` with `""`):  
```m

```
CleanedColumn = Table.ReplaceValue(  
    Source,  
    each [Price],  
    each if Text.Contains([Price], "$") then Text.Remove([Price], {"$"}) else [Price],  
    Replacer.ReplaceText,  
    {"Price"}  
)
```

)

- Then convert the cleaned column to Decimal.

#### 4. Handle Errors (Optional)

- Right-click the column → Replace Errors → Set a default value (e.g., 0 or null).

#### 5. Reimport the CSV (If Needed)

- If the CSV itself has formatting issues (e.g., commas in numbers), pre-clean it in Excel or use delimited settings in Power Query.

#### ### Prevent Future Issues

- Modify the CSV Source: Ensure consistent data types (e.g., no text in numeric columns).
- Use Power Query's Locale Settings: If numbers use different decimal separators (e.g., 1,000 vs. 1.000), set the correct locale under Data Type → Using Locale.

#### ### Example M Code

m

let

```
Source = Csv.Document(File.Contents("C:\Data\Sales.csv"),
PromotedHeaders = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
CleanedPrice = Table.ReplaceValue(
    PromotedHeaders,
    each [Price],
    each if [Price] = "N/A" then null else Text.Replace([Price], "$", ""),
    Replacer.ReplaceText,
    {"Price"}
),
ChangedType = Table.TransformColumnTypes(CleanedPrice, {"Price", type number})
```

in

ChangedType

#### 14. Using SQL Query with Parameters (Recommended for Performance)

##### Step 1: Create a Parameter

Go to the Home tab → Manage Parameters → New Parameter.

Name: YearFilter

Type: Whole Number

Suggested Values: List of values (e.g., 2020, 2021, 2022)

Default Value: 2023

Step 2: Connect to SQL Database

Get Data → SQL Server (or Azure SQL/MySQL/PostgreSQL).

Enter server/database credentials.

Select Advanced options and paste a parameterized SQL query:

sql

```
SELECT * FROM Sales
```

```
WHERE Year = @YearFilter -- Parameterized filter
```

Click OK → Power BI will prompt you to input the parameter value (@YearFilter maps to your YearFilter parameter).

Step 3: Refresh with Dynamic Filtering

Change the YearFilter parameter value from the Parameters pane or create a slicer for user interaction.

## 15. Auto-Refresh Power BI Dataset on Schedule

Use Case: Refresh a Power BI dataset daily or when a file updates.

Steps:

Prepare Power BI:

Publish your report to the Power BI Service.

Ensure the data source has gateway access (if on-premises).

Create a Power Automate Flow:

Go to Power Automate → Create → Automated cloud flow.

Choose a trigger:

Recurrence: For scheduled refreshes (e.g., daily at 8 AM).

When a file is created/modified (OneDrive/SharePoint): To trigger on file updates.

Add the Power BI action:

Search for "Refresh a dataset" → Select your workspace and dataset.

(Optional) Add a notification (e.g., email/SMS) on success/failure.

Save and Test:

Run the flow manually to verify it refreshes the dataset.