1.Three data sources Power BI can connect to are:

1. **SQL Server Database**
2. **Excel files (.xlsx, .xls)**
3. **SharePoint Online Lists**

2. The first step to import data into Power BI Desktop is to **click the "Get Data" button** on the Home ribbon and select your desired data source.

3. You can refresh imported data in Power BI Desktop by going to the **Home** ribbon and clicking **Refresh**.
This re-runs the query and pulls the latest data from the source into your report.

4. Two file formats Power BI can import directly are:

- **Excel (.xlsx, .xls)**
- **Comma-Separated Values (.csv)**

5. In Power BI, after you select a data source, the **Navigator** window shows:

- A **list of available tables, sheets, or objects** from that source.
- A **preview of the selected data** so you can confirm it's the right dataset before loading.

7. In Power BI Desktop, you can change **OrderDate** to a date format during import by:

1. **Clicking "Transform Data"** instead of "Load" when importing.
2. In **Power Query Editor**, selecting the **OrderDate** column.
3. Going to the **Home** or **Transform** tab → **Data Type** dropdown → selecting **Date**.
4. Clicking **Close & Apply** to save changes.

This way, the column will be stored as an actual **Date** type, enabling date-specific filtering and calculations.

8. In Power BI's import dialog:

- **Load** → Brings the data directly into Power BI **as it is**, without changes.
- **Transform Data** → Opens the **Power Query Editor** so you can clean, filter, or reshape the data **before** loading it into the model.

Think of it as:
Load = "Just take it as it is."
Transform Data = "Let me tidy it up first."

9. One common reason is **incorrect server name or database credentials** — if the username, password, or server address is wrong, Power BI can't connect.
Other causes could be firewall restrictions, missing drivers, or no network access to the SQL server.

10. In Power BI Desktop, you can replace a data source like this:
1. Go to the **Home** tab.
2. Click **Transform data → Data source settings**.
3. Select the existing data source from the list.
4. Click **Change Source…**.
5. Enter the new file path, server, or database information.
6. Click **OK**, then **Close & Apply** to update your report.

12. If your **Sales_Data.csv** file has changed location or name, you can update the data source in Power BI like this:

Method 1 — Using Power BI Desktop UI
1. Go to **Home → Transform Data → Data Source Settings**.
2. In the window, find your CSV data source path.
3. Select it and click **Change Source…**.
4. Browse to the new **Sales_Data.csv** location or rename it.
5. Click **OK → Close & Apply** to save changes.

Method 2 — Editing M-code in Power Query
1. Go to **Home → Transform Data**.
2. In **Applied Steps**, select the step starting with File.Contents(...).
3. Change the file path in the formula bar, for example:
4. Source = Csv.Document(
5. File.Contents("C:\NewPath\Sales_Data.csv"),
6. [Delimiter=",", Columns=5, Encoding=65001, QuoteStyle=QuoteStyle.None]
7.)
8. Close & Apply.

If you set it up as a **parameter** for the file path, you won't need to change the M-code each time — just update the parameter value.
Do you want me to show you how to make **Sales_Data.csv** use a parameter for the path so it's easier to switch?

13. **1. In Power Query (best practice)**
- Go to **Home → Transform Data**.
- Find the problematic column (it often has an error icon in the header).
- Change its type explicitly:
  - **Right-click column header → Change Type → Text** (or Number/Date as needed).
  - If needed, use **Transform → Detect Data Type** to auto-detect first, then adjust.
- If you want to keep all values but standardize them, use: **Transform → Replace Errors** and fill with a default (e.g., 0 for numbers).

---

2. Clean data before import

- Open the CSV in Excel or a text editor.
- Make sure each column has only one type of data.
- Save and re-import.

3.In M-code directly

If you want to force all values to text, modify the step where types are set:

ChangedType = Table.TransformColumnTypes(Source, {{"Quantity", type text}})

or for numbers:
ChangedType = Table.TransformColumnTypes(Source, {{"Quantity", Int64.Type}})

15. 1️⃣ **Prepare Your Power BI Dataset**

- Publish your Power BI report to the **Power BI Service** (cloud).
- Ensure your **data source** has proper credentials stored in Power BI Service.
- If it's a file-based source (like Excel/CSV), store it in **OneDrive** or **SharePoint** so the service can access it.
- If it's SQL or another database, configure a **gateway** if needed.

2️⃣ **Create a Power Automate Flow**

1. Go to Power Automate.
2. Click **Create → Automated cloud flow**.
3. Choose a **trigger**:
    - **Schedule**: Run every X hours/days.
    - **File added/modified in folder**: Trigger when a CSV/Excel file is updated in OneDrive/SharePoint.
    - **HTTP/Webhook**: Trigger from an external system.
4. Add **Power BI → Refresh a dataset** action.
5. Select:
    - **Workspace** where the dataset is published.
    - **Dataset** you want to refresh.

3️⃣ **(Optional) Trigger on File Upload**

Example: When Sales_Data.csv is updated in SharePoint:

- **Trigger**: "When a file is created or modified (properties only)" in SharePoint.
- **Condition**: File name = Sales_Data.csv.
- **Action**: Refresh Power BI dataset.

4️⃣ **Monitor the Automation**

- In Power Automate, check **Run history** for errors.

- In Power BI Service, view **Dataset → Refresh history**.
- Add email/Teams notification in the flow to alert if refresh fails.

✅ **Result**:

- You drop or update a file → Flow runs → Power BI dataset refreshes automatically → Reports are always up to date.
- For SQL/other live sources, you can still use **scheduled refresh** but let Power Automate handle *event-based refreshes* when needed.