



INSIGHTSMASTERY ACADEMY

LEARNING HANDBOOK

Data Connectivity in Power BI

Connect Excel • CSV • Folder • SQL Server • Semantic Model
Step-by-step examples for Power BI Desktop

Practical guide with clear steps, best practices,
and ready-to-use patterns for reliable data loading

Power BI Desktop • Power Query • Get Data

www.insightsmastry.in



Table of Contents

1. Introduction — Get Data in Power BI Desktop
2. Connect to Excel Workbooks
3. Connect to CSV / Text Files
4. Connect to a Folder (Combine Multiple Files)
5. Connect to SQL Server
6. Connect to a Power BI Semantic Model (Live)
7. After Loading — Prepare for Time Intelligence
8. Comparison & When to Use Which Connector
9. Common Issues & Best Practices
10. Quick Reference Checklist



1. Introduction — Get Data in Power BI Desktop

Almost every Power BI solution starts with **Get data**. From the Home ribbon you choose a connector, authenticate, select objects in the Navigator, then either **Load** the data or **Transform Data** in Power Query Editor before it enters the model.

This handbook covers the five most common enterprise and learning scenarios:

- **Excel** — single workbook with sheets or tables
- **CSV / Text** — flat files, often daily exports
- **Folder** — many similar files that should be combined automatically
- **SQL Server** — relational database (Import or DirectQuery)
- **Power BI Semantic Model** — live connection to an existing published model

How to open Get Data

In Power BI Desktop, on the **Home** tab, click the **Get data** button (or the small dropdown arrow next to it). Common sources appear in a short list; choose **More...** to open the full Get Data dialog with categories (File, Database, Power Platform, Azure, Online Services, Other).

■ Screenshot cue — Get Data

Home ribbon → Data group → Get data. You will see Excel, SQL Server, and Power BI semantic models as quick buttons. Clicking More... opens the full connector list where Folder and Text/CSV also appear.



2. Connect to Excel Workbooks

Use the Excel connector when your data lives in .xlsx or .xlsm files. Prefer Excel **Tables** (Insert → Table) over plain ranges — tables keep headers and expand automatically.

Step 1: Start the connector

Home → **Get data** → **Excel workbook** (or Get data → More → File → Excel workbook → Connect).
Browse to the .xlsx / .xlsm file and click Open.

Step 2: Navigator — choose objects

The Navigator window lists sheets and any Excel Tables inside the workbook.
Check the box next to each table or sheet you need. Prefer objects with the Table icon.
Preview appears on the right. Click **Transform Data** to clean first, or **Load** to bring data straight into the model.

Step 3: Example scenario

File: Sales_2024.xlsx containing tables Sales, Products, Customers.
Select all three tables → Transform Data → set correct data types (especially Date columns) → Close & Apply.

Best practices for Excel

- Convert ranges to Excel Tables before connecting (Ctrl+T).
- Give tables meaningful names (Sales, DimProduct) instead of Table1, Table2.
- Keep one fact table and dimension tables in the same workbook for simple models.
- Avoid merged cells and total rows inside the data area.
- For files on OneDrive / SharePoint, you can also use the Web connector with the file URL.



3. Connect to CSV / Text Files

CSV files are common for system exports. Power BI's Text/CSV connector detects delimiter, encoding, and headers automatically in most cases.

Step 1: Start the connector

Home → **Get data** → **Text/CSV**.

Select the .csv (or .txt) file → Open.

Step 2: Preview & settings

A preview dialog appears. Confirm File Origin (encoding), Delimiter (Comma, Tab, etc.), and Data Type Detection.

If headers are wrong, change "Data Type Detection" or click Transform Data to fix in Power Query.

Click **Load** or **Transform Data**.

Step 3: Example scenario

Daily export: Orders_20240315.csv with columns OrderID, OrderDate, CustomerID, Amount.

Connect → confirm comma delimiter and UTF-8 → Transform Data → change OrderDate to Date type → Close & Apply.

Later create a Date table and relate OrderDate to it for time intelligence.

■ Screenshot cue — Text/CSV dialog

After selecting the file you see a preview grid, File Origin dropdown, Delimiter dropdown, and buttons Load / Transform Data. Always verify the first row is treated as headers.

■ Encoding & locale

If dates or numbers look wrong (e.g. 03/04/2024 interpreted as March instead of April), open Transform Data and use Data Type → Using Locale... or set the correct locale in the source step.



4. Connect to a Folder (Combine Multiple Files)

When you receive a new CSV or Excel file every day/week with the same structure, point Power BI at the **folder** instead of individual files. Power Query can combine all files automatically and will pick up new files on refresh.

Step 1: Start the Folder connector

Home → **Get data** → More... → File → **Folder** → Connect.
Browse to (or paste) the folder path that contains the files → OK.

Step 2: Choose how to proceed

You see a list of files (Name, Extension, Date modified, etc.).
Prefer **Combine & Transform Data** — this creates a sample query and combines all matching files.
Alternatively: Combine & Load (straight to model) or Transform Data (work with the file list first).

Step 3: Combine files experience

Power Query shows a sample file. Select the correct sheet/table/object that exists in every file.
Click OK. Power Query generates helper queries (Transform Sample File, Parameter, etc.) and a final combined table.
Rename the final query (e.g. Sales Combined) and clean data types, then Close & Apply.

Step 4: Example scenario

Folder: C:\Data\DailySales\ containing Sales_2024-03-01.csv, Sales_2024-03-02.csv, ...
All files have identical columns. Connect to Folder → Combine & Transform Data → select the CSV sample → OK.
After refresh, any new CSV dropped into the folder is automatically included.

■ Folder combine rules

All files must share the same format and column structure. If some files differ, first Transform Data, filter the file list (e.g. only .csv, exclude temp files), then combine. Subfolders are included by default.



5. Connect to SQL Server

The SQL Server connector supports both **Import** (data copied into the Power BI model) and **DirectQuery** (queries run live against the database). Choose Import for most analytical models; use DirectQuery when you need near-real-time data or very large tables.

Step 1: Start the connector

Home → **Get data** → **SQL Server** (or More... → Database → SQL Server database).
Enter Server name (e.g. sql-prod.contoso.com or localhost\SQLEXPRESS).
Optionally enter Database name.
Choose Data Connectivity mode: **Import** or **DirectQuery** → OK.

Step 2: Authentication

Select authentication: Windows, Database (SQL login), or Organizational account (Microsoft Entra ID) if supported.
Enter credentials and set the level (e.g. this server) → Connect.
If the connection is unencrypted you may see a warning — accept only on trusted networks or configure encryption.

Step 3: Navigator

Expand the database → select the tables or views you need.
You can also use Advanced options → SQL statement to write a native query.
Click Transform Data (recommended) or Load.

Step 4: Example scenario

Server: analytics-sql.company.local, Database: SalesDW.
Import mode. Select FactInternetSales, DimDate, DimProduct, DimCustomer.
In Power Query set DateKey columns to Date type where appropriate, then Close & Apply.
Create relationships and mark DimDate as the Date table for time intelligence.

Import vs DirectQuery (quick guide)

Aspect	Import	DirectQuery
Data location	Copied into .pbix model	Stays in SQL Server
Performance	Fast in-memory	Depends on database & network
Refresh	Scheduled / manual refresh needed	Always current
Model size	Limited by memory	No size limit in model
DAX & modeling	Full features	Some limitations
Best for	Most reports & time intelligence	Real-time / huge facts

Table 1 — Import vs DirectQuery



6. Connect to a Power BI Semantic Model (Live)

Instead of importing raw data, you can connect live to an existing published semantic model. This is the recommended pattern for self-service: one central model, many reports. You get all tables, measures, and hierarchies; RLS is enforced.

Step 1: Start the connector

Home → **Get data** → **Power BI semantic models**

(or the dedicated Power BI semantic models button on the Home ribbon).

Sign in if prompted.

Step 2: Data hub / OneLake catalog

A list of semantic models you have Build permission on appears.

Filter by My data, Endorsed, or search by name.

Select the model → **Connect**.

Step 3: Live connection result

Power BI Desktop shows “Connected live to the Power BI semantic model: [Name]”.

Fields pane lists tables and measures from the model. You cannot edit the model structure in this mode.

Build visuals and reports. Changes to the central model appear after refresh of the live connection.

Step 4: Example scenario

Central model “Enterprise Sales” is published in workspace Finance Analytics and certified.

Analysts open Power BI Desktop → Get data → Power BI semantic models → select Enterprise Sales → Connect.

They build department reports without duplicating data or DAX.

■ Screenshot cue — Semantic models

After choosing Power BI semantic models you see the Data hub with search, filters (My data / Endorsed), and model cards showing name, owner, workspace, and endorsement badges. Click Connect on the chosen model.

■ Live connection limits

In a pure live connection you cannot add new tables from other sources or change relationships. To combine a live model with other data you need a composite model (DirectQuery to the semantic model + Import other sources) — advanced scenario.



7. After Loading — Prepare for Time Intelligence

Once data is loaded from any of the sources above, good modeling makes time intelligence reliable.

- Ensure every fact table has a proper Date column (Date type, not text).
- Create or load a dedicated **Date table** (contiguous dates, no gaps).
- Mark the Date table: select it → Table tools → Mark as date table → choose the Date column.
- Create a relationship from Fact[Date] to DateTable[Date] (many-to-one).
- Disable Auto date/time (File → Options → Data Load) to avoid hidden date tables.
- Then write measures such as TOTALYTD, SAMEPERIODLASTYEAR, DATEADD using DateTable[Date].

Minimal Date table (DAX) after any connection

```
DateTable =  
ADDCOLUMNS (  
    CALENDAR ( DATE ( 2020, 1, 1 ), DATE ( 2026, 12, 31 ) ),  
    "Year", YEAR ( [Date] ),  
    "Month Number", MONTH ( [Date] ),  
    "Month Name", FORMAT ( [Date], "MMMM" ),  
    "Quarter", "Q" & FORMAT ( [Date], "Q" )  
)  
// Mark as date table and relate to your fact date column
```



8. Comparison & When to Use Which Connector

Source	Best when...	Watch out for...
Excel	Small–medium curated workbooks, prototypes	Merged cells, no tables, large files
CSV / Text	System exports, simple flat files	Encoding, delimiter, locale of dates
Folder	Many same-structure files over time	Inconsistent columns across files
SQL Server	Enterprise databases, large volumes	Credentials, gateway, Import vs DQ choice
Semantic Model	Reuse certified models, governed self-service	Need Build permission; live limits

Table 2 — Choosing the right connector



9. Common Issues & Best Practices

Common issues

- Wrong data types after load → always check in Power Query (especially dates and numbers).
- CSV dates appear as text → use Change Type with locale or Date.From.
- Folder combine fails → files have different columns; filter or standardize first.
- SQL connection fails → check firewall, gateway, encryption, and credentials.
- Semantic model not listed → you need Build permission on that model.
- Refresh fails for local files → use OneDrive/SharePoint or a gateway for scheduled refresh in the service.

Best practices

- Prefer Transform Data over Load so you can clean types and names before the model.
- Name queries clearly (FactSales, DimProduct) immediately.
- Remove unused columns early to keep the model small.
- For production, move from local Excel/CSV to Folder + SharePoint or to a database.
- Document the source and refresh expectations for each query.
- After loading, build a proper Date table before writing time intelligence measures.



10. Quick Reference Checklist

Before you click Get data

- Know the source type (file, folder, database, existing model)
- Have credentials / path / server name ready
- Prefer tables over raw sheets in Excel
- Decide Import vs DirectQuery for databases

While connecting

- Use Navigator to select only needed objects
- Prefer Transform Data to fix types and names
- Confirm headers and sample rows look correct

After Close & Apply

- Check Model view relationships
- Create / mark Date table for time intelligence
- Hide technical columns from Report view
- Write base measures then time intelligence measures
- Test refresh



InsightsMastery Academy

Data Connectivity Handbook — Power BI

Connect cleanly. Model once. Analyze with confidence.

www.insightsmastry.in