**Introduction**

Completed100 XP

* 2 minutes

Like most of us, you work for a company where you're required to build Microsoft Power BI reports. The data resides in several different databases and files. These data repositories are different from each other, some are in Microsoft SQL Server, some are in Microsoft Excel, but all the data is related.

[Introduction - Training | Microsoft Learn](https://learn.microsoft.com/en-us/training/modules/get-data/1-introduction)

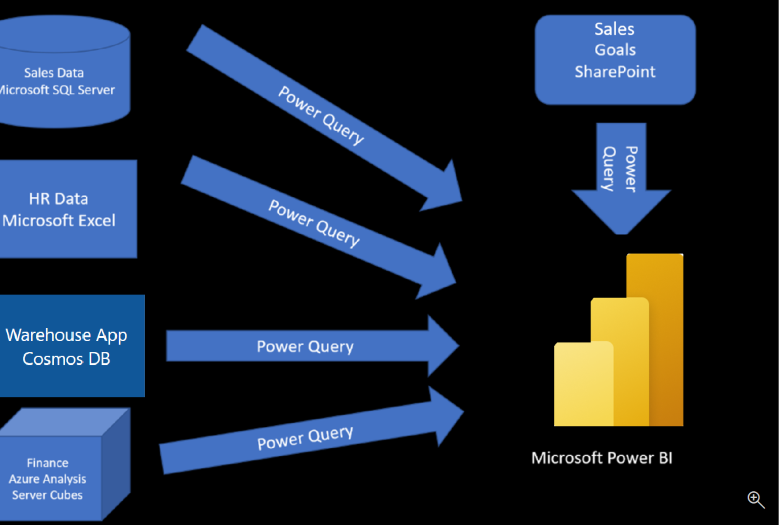
**Note**

The module sections prior to the lab are purely informational. You will be given the opportunity to work with real data during the lab.

In this module’s scenario, you work for Tailwind Traders. You’ve been tasked by senior leadership to create a suite of reports that are dependent on data in several different locations. The database that tracks sales transactions is in SQL Server, a relational database that contains what items each customer bought and when. It also tracks which employee made the sale, along with the employee name and employee ID. However, that database doesn’t contain the employee’s hire date, their title, or who their manager is. For that information, you need to access files that Human Resources keeps in Excel. You've been consistently requesting that they use an SQL database, but they haven't yet had the chance to implement it.

When an item ships, the shipment is recorded in the warehousing application, which is new to the company. The developers chose to store data in Cosmos DB, as a set of JSON documents.

Tailwind Traders has an application that helps with financial projections, so that they can predict what their sales will be in future months and years, based on past trends. Those projections are stored in Microsoft Azure Analysis Services. Here’s a view of the many data sources you're asked to combine data from.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/1-data-source-scenario-c.png#lightbox)

Before you can create reports, you must first extract data from the various data sources. Interacting with SQL Server is different from Excel, so you should learn the nuances of both systems. After gaining understanding of the systems, you can use Power Query to help you clean the data, such as renaming columns, replacing values, removing errors, and combining query results. Power Query is also available in Excel. After the data has been cleaned and organized, you're ready to build reports in Power BI. Finally, you'll publish your combined dataset and reports to Power BI service. From there, other people can use your dataset and build their own reports or they can use the reports you’ve already built. Additionally, if someone else built a dataset you'd like to use, you can build reports from that too!

This module will focus on the first step of getting the data from the different data sources and importing it into Power BI by using Power Query.

By the end of this module, you’ll be able to:

* Identify and connect to a data source
* Get data from a relational database, such as Microsoft SQL Server
* Get data from a file, such as Microsoft Excel
* Get data from applications
* Get data from Azure Analysis Services
* Select a storage mode
* Fix performance issues
* Resolve data import errors

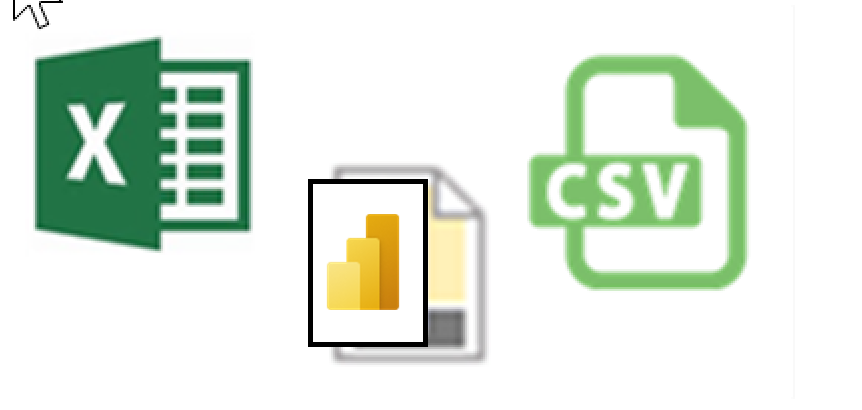
**Next unit: Get data from files**

# Get data from files

Completed100 XP

* 10 minutes

Organizations often export and store data in files. One possible file format is a flat file. A flat file is a type of file that has only one data table and every row of data is in the same structure. The file doesn't contain hierarchies. Likely, you're familiar with the most common types of flat files, which are comma-separated values (.csv) files, delimited text (.txt) files, and fixed width files. Another type of file would be the output files from different applications, like Microsoft Excel workbooks (.xlsx).

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/2-file-types-c.png#lightbox)

Power BI Desktop allows you to get data from many types of files. You can find a list of the available options when you use the **Get data** feature in Power BI Desktop. The following sections explain how you can import data from an Excel file that is stored on a local computer.

## Scenario

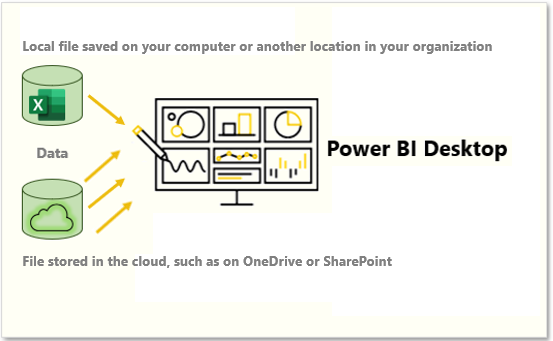
The Human Resources (HR) team at Tailwind Traders has prepared a flat file that contains some of your organization's employee data, such as employee name, hire date, position, and manager. They've requested that you build Power BI reports by using this data, and data that is located in several other data sources.

### Flat file location

The first step is to determine which file location you want to use to export and store your data.

Your Excel files might exist in one of the following locations:

* **Local** - You can import data from a local file into Power BI. The file isn't moved into Power BI, and a link doesn't remain to it. Instead, a new dataset is created in Power BI, and data from the Excel file is loaded into it. Accordingly, changes to the original Excel file aren't reflected in your Power BI dataset. You can use local data import for data that doesn't change.
* **OneDrive for Business** - You can pull data from OneDrive for Business into Power BI. This method is effective in keeping an Excel file and your dataset, reports, and dashboards in Power BI synchronized. Power BI connects regularly to your file on OneDrive. If any changes are found, your dataset, reports, and dashboards are automatically updated in Power BI.
* **OneDrive - Personal** - You can use data from files on a personal OneDrive account, and get many of the same benefits that you would with OneDrive for Business. However, you'll need to sign in with your personal OneDrive account, and select the **Keep me signed in** option. Check with your system administrator to determine whether this type of connection is allowed in your organization.
* **SharePoint - Team Sites** - Saving your Power BI Desktop files to SharePoint Team Sites is similar to saving to OneDrive for Business. The main difference is how you connect to the file from Power BI. You can specify a URL or connect to the root folder.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/2-local-vs-cloud-c.png#lightbox)

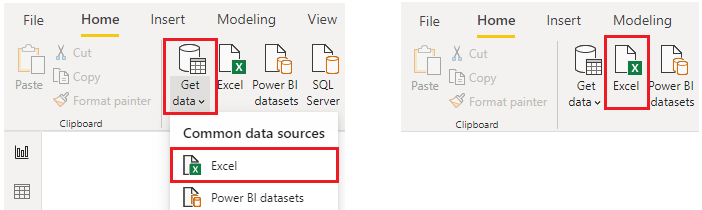
Using a cloud option such as OneDrive or SharePoint Team Sites is the most effective way to keep your file and your dataset, reports, and dashboards in Power BI in-sync. However, if your data doesn't change regularly, saving files on a local computer is a suitable option.

### Connect to data in a file

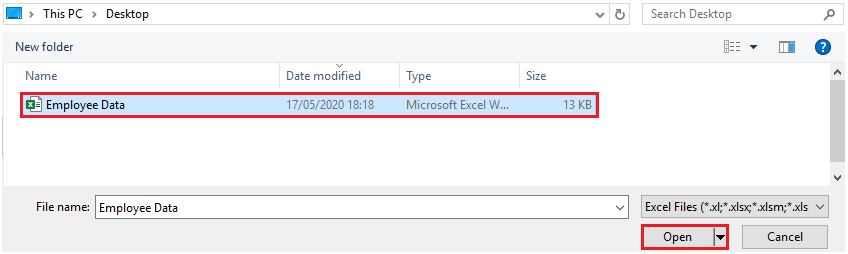
In Power BI, on the **Home** tab, select **Get data**. In the list that displays, select the option that you require, such as **Text/CSV** or **XML**. For this example, you'll select **Excel**.

**Tip**

The **Home** tab contains quick access data source options, such as **Excel**, next to the **Get data** button.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/2-get-data-excel-ssm.png#lightbox)

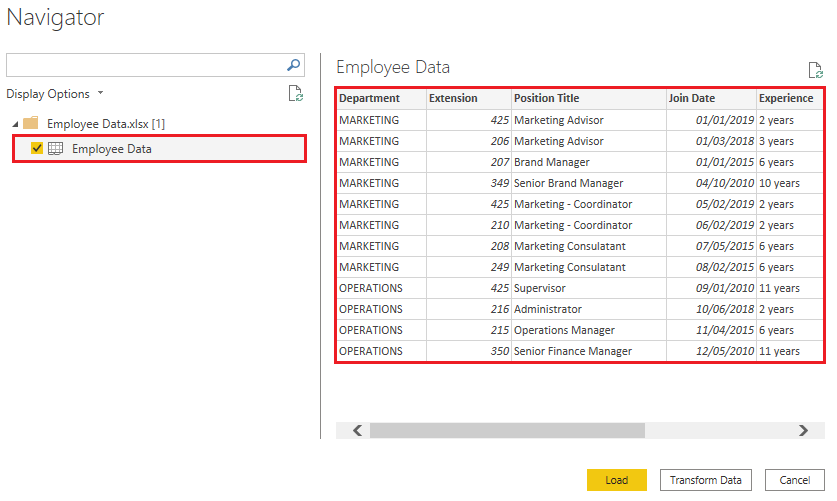
Depending on your selection, you need to find and open your data source. You might be prompted to sign into a service, such as OneDrive, to authenticate your request. In this example, you'll open the **Employee Data** Excel workbook that is stored on the Desktop (Remember, no files are provided for practice, these are hypothetical steps).

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/2-excel-save-data-ssm.png#lightbox)

### Select the file data to import

After the file has connected to Power BI Desktop, the **Navigator** window opens. This window shows you the data that is available in your data source (the Excel file in this example). You can select a table or entity to preview its contents, to ensure that the correct data is loaded into the Power BI model.

Select the check box(es) of the table(s) that you want to bring in to Power BI. This selection activates the **Load** and **Transform Data** buttons as shown in the following image.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/2-excel-worksheet-ssm.png#lightbox)

Now you can select the **Load** button to automatically load your data into the Power BI model or select the **Transform Data** button to launch the Power Query Editor, where you can review and clean your data before loading it into the Power BI model.

We often recommend that you transform data, but that process will be discussed later in this module. For this example, you can select **Load**.

### Change the source file

You might have to change the location of a source file for a data source during development, or if a file storage location changes. To keep your reports up to date, you'll need to update your file connection paths in Power BI.

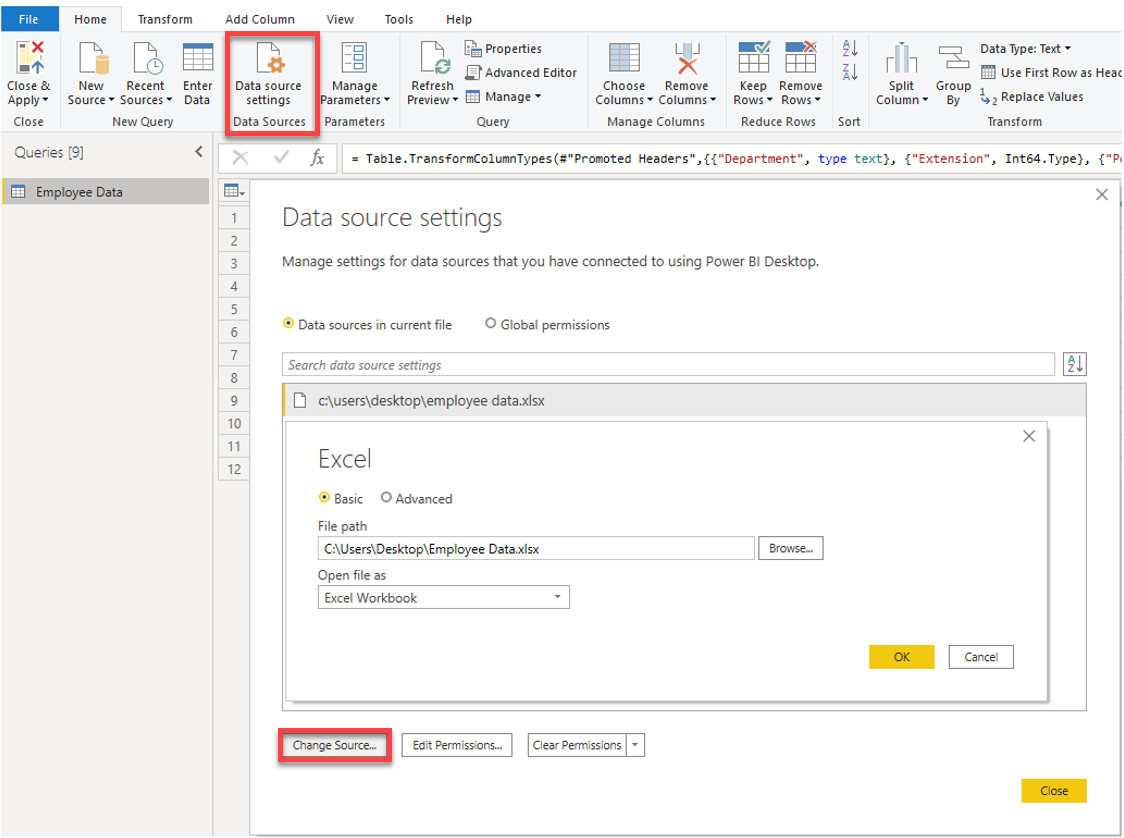
Power Query provides many ways for you to accomplish this task, so that you can make this type of change when needed.

1. Data source settings
2. Query settings
3. Advanced Editor

**Warning**

If you are changing a file path, make sure that you reconnect to the same file with the same file structure. Any structural changes to a file, such as deleting or renaming columns in the source file, will break the reporting model.

For example, try changing the data source file path in the data source settings. Select **Data source settings** in Power Query. In the **Data source settings** window, select your file and then select **Change Source**. Update the **File path** or use the **Browse** option to locate your file, select **OK**, and then select **Close**.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/2-excel-data-source-settings-ssm.png#lightbox)

## Next unit: Get data from relational data sources

# Get data from relational data sources

Completed100 XP

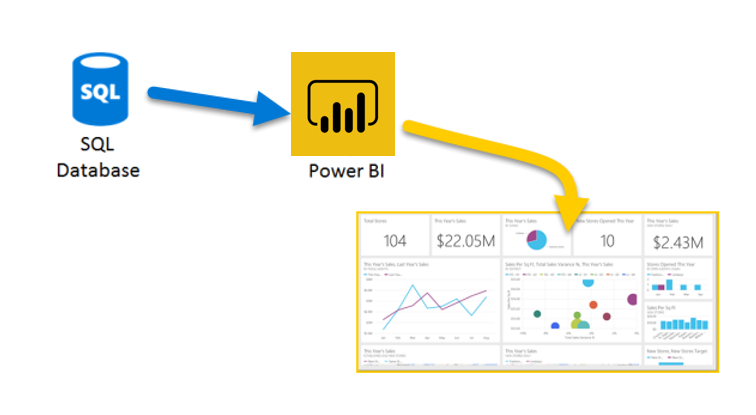
* 14 minutes

If your organization uses a relational database for sales, you can use Power BI Desktop to connect directly to the database instead of using exported flat files.

Connecting Power BI to your database will help you to monitor the progress of your business and identify trends, so you can forecast sales figures, plan budgets and set performance indicators and targets. Power BI Desktop can connect to many relational databases that are either in the cloud or on-premises.

## Scenario

The Sales team at Tailwind Traders has requested that you connect to the organization's on-premises SQL Server database and get the sales data into Power BI Desktop so you can build sales reports.

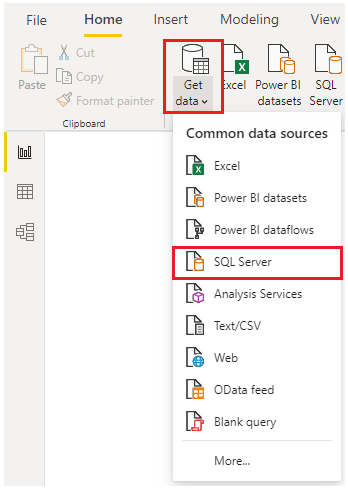
[](https://learn.microsoft.com/en-us/training/modules/get-data/media/3-get-data-sql-server-ssm.png#lightbox)

### Connect to data in a relational database

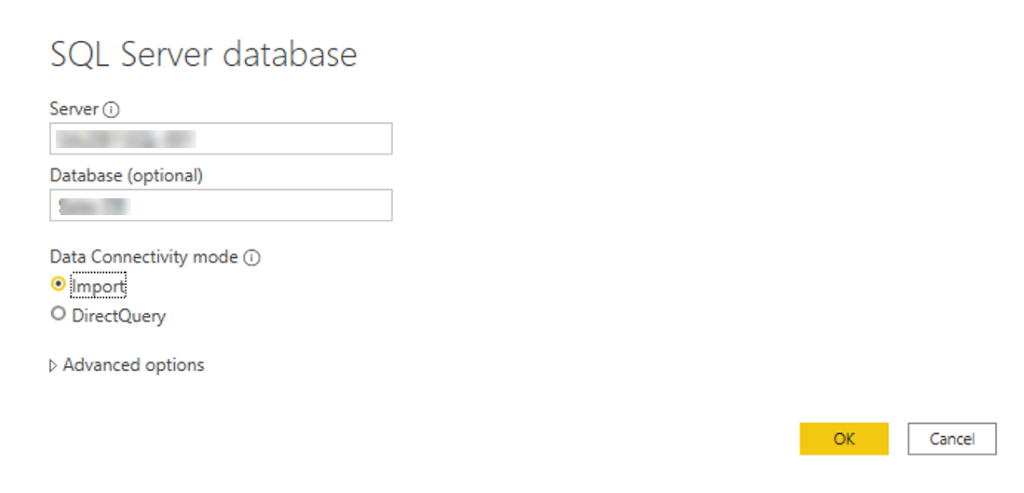
You can use the **Get data** feature in Power BI Desktop and select the applicable option for your relational database. For this example, you would select the **SQL Server** option, as shown in the following screenshot.

**Tip**

Next to the **Get Data** button are quick access data source options, such as **SQL Server.**

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/3-get-data-sql-server-dropdown-ssm..png#lightbox)

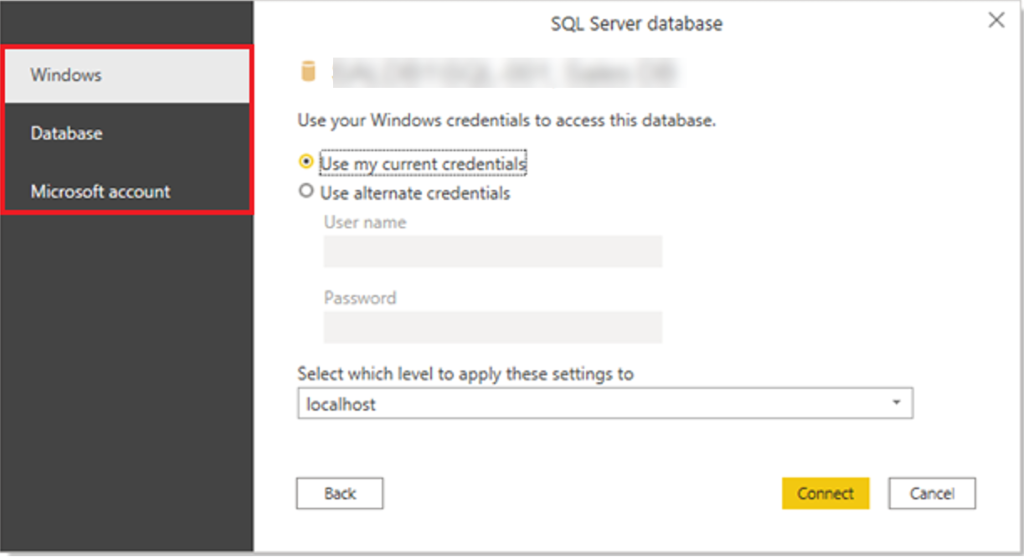
Your next step is to enter your database server name and a database name in the **SQL Server database** window. The two options in data connectivity mode are: **Import** (selected by default, recommended) and **DirectQuery**. Mostly, you select **Import.** Other advanced options are also available in the **SQL Server database** window, but you can ignore them for now.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/3-get-data-sql-server-db-ss.png#lightbox)

After you've added your server and database names, you'll be prompted to sign in with a username and password. You'll have three sign-in options:

* **Windows** - Use your Windows account (Azure Active Directory credentials).
* **Database** - Use your database credentials. For instance, SQL Server has its own sign-in and authentication system that is sometimes used. If the database administrator gave you a unique sign-in to the database, you might need to enter those credentials on the **Database** tab.
* **Microsoft account** - Use your Microsoft account credentials. This option is often used for Azure services.

Select a sign-in option, enter your username and password, and then select **Connect**.

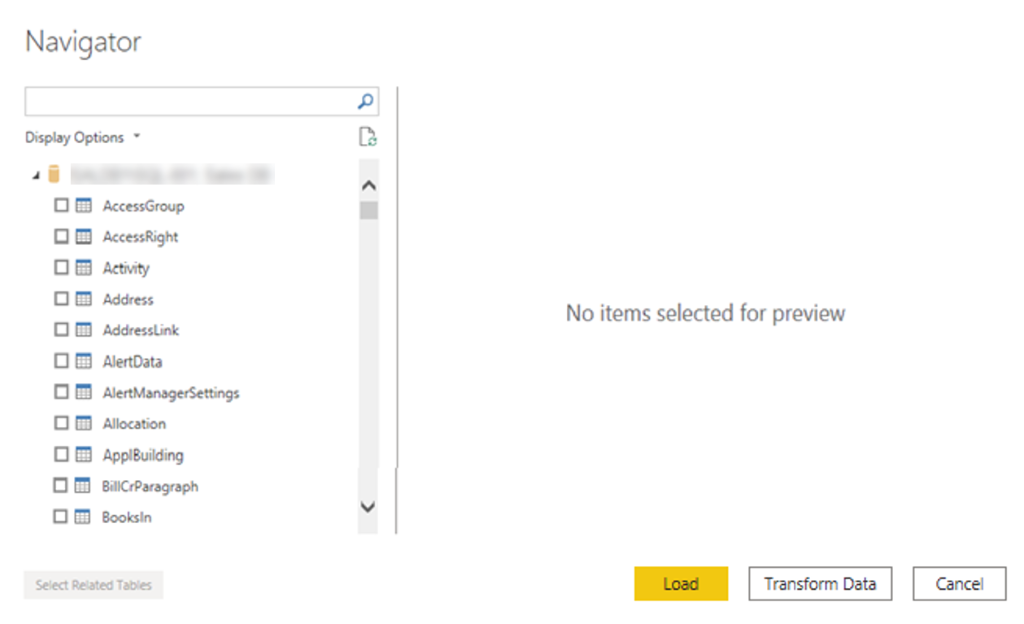
[](https://learn.microsoft.com/en-us/training/modules/get-data/media/3-sql-creds-ssm.png#lightbox)

### Select data to import

After the database has been connected to Power BI Desktop, the **Navigator** window displays the data that is available in your data source (the SQL database in this example). You can select a table or entity to preview its contents and make sure that the correct data will be loaded into the Power BI model.

Select the check box(es) of the table(s) that you want to bring in to Power BI Desktop, and then select either the **Load** or **Transform Data** option.

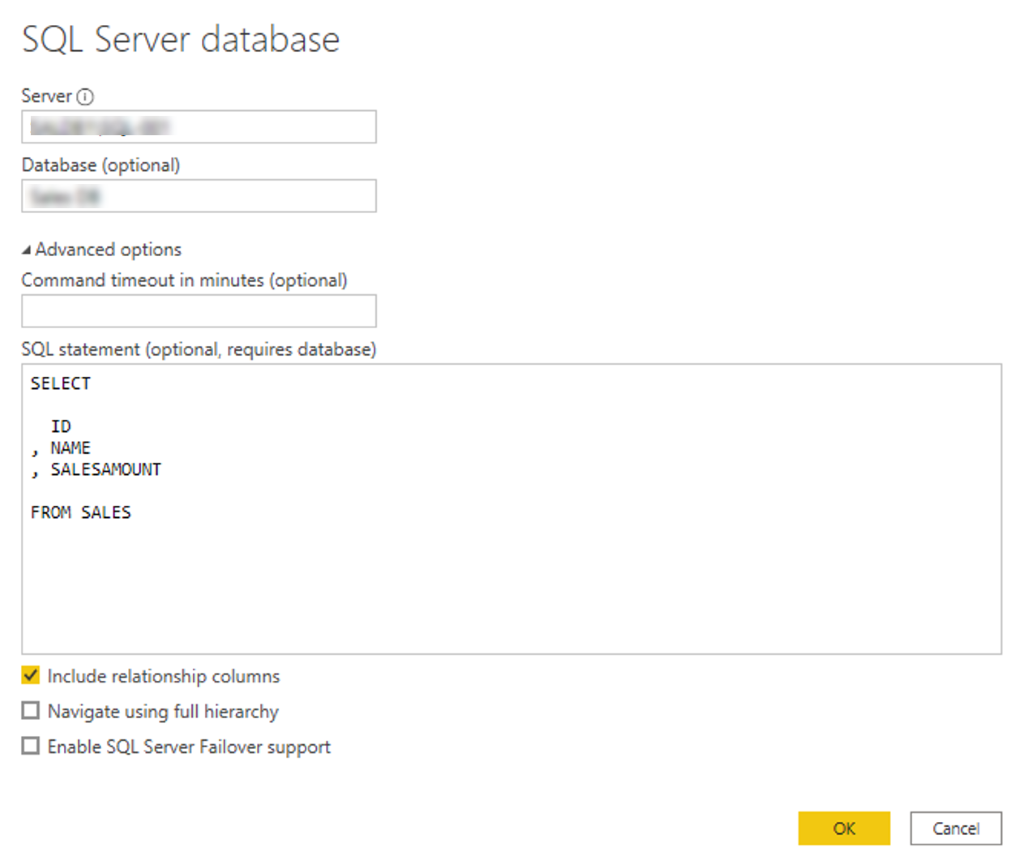
* **Load** - Automatically load your data into a Power BI model in its current state.
* **Transform Data** - Open your data in Microsoft Power Query, where you can perform actions such as deleting unnecessary rows or columns, grouping your data, removing errors, and many other data quality tasks.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/3-table-selection-ssm.png#lightbox)

### Import data by writing an SQL query

Another way you can import data is to write an SQL query to specify only the tables and columns that you need.

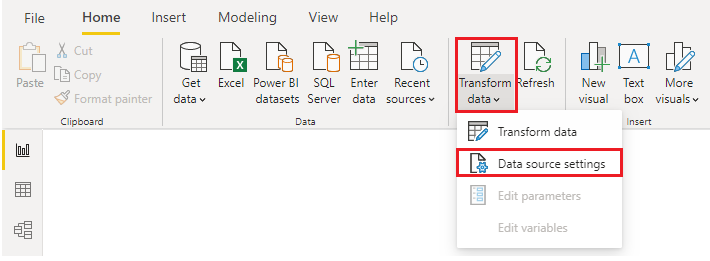
To write your SQL query, on the **SQL Server database** window, enter your server and database names, and then select the arrow next to **Advanced options** to expand this section and view your options. In the **SQL statement** box, write your query statement, and then select **OK**. In this example, you'll use the **Select** SQL statement to load the ID, NAME and SALESAMOUNT columns **from** the SALES table.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/3-sql-statement-ss.png#lightbox)

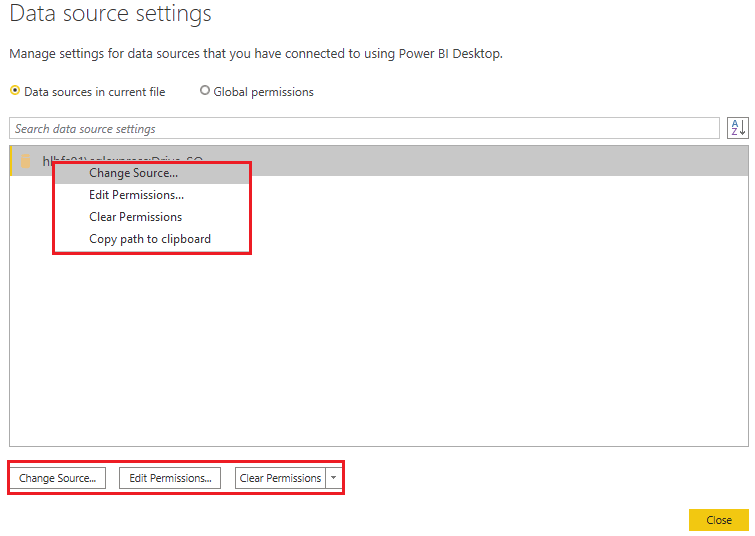
### Change data source settings

After you create a data source connection and load data into Power BI Desktop, you can return and change your connection settings at any time. This action is often required due to a security policy within the organization, for example, when the password needs to be updated every 90 days. You can change the data source, edit permissions or clear permissions.

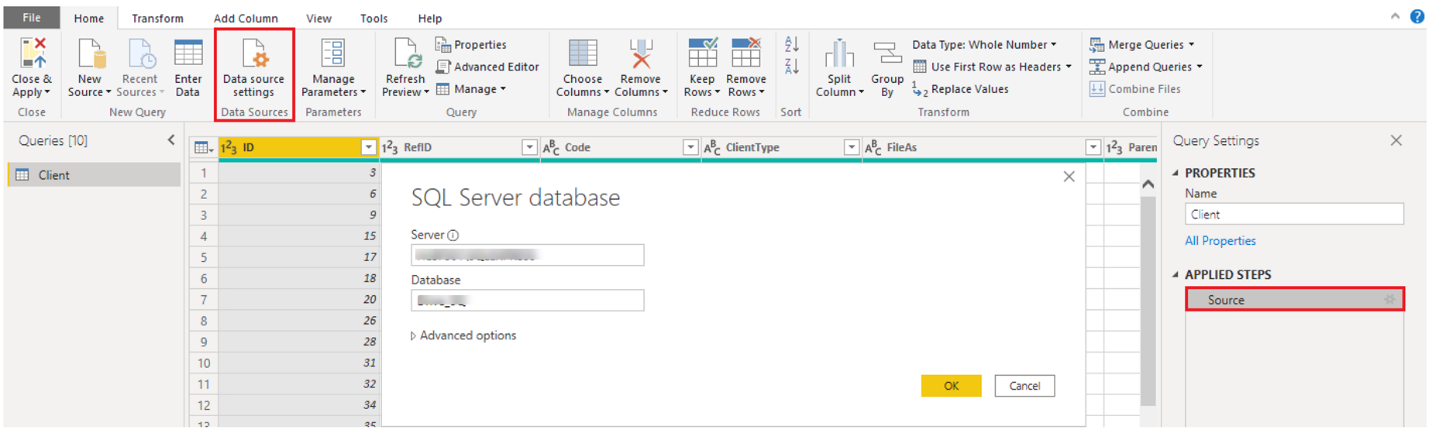
On the **Home** tab, select **Transform data,** and then select the **Data source settings** option.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/3-change-sql-settings-ssm.png#lightbox)

From the list of data sources that displays, select the data source that you want to update. Then, you can right-click that data source to view the available update options or you can use the update option buttons on the lower left of the window. Select the update option that you need, change the settings as required, and then apply your changes.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/3-sql-data-source-edit-ssm.png#lightbox)

You can also change your data source settings from within Power Query. Select the table, and then select the **Data source settings** option on the **Home** ribbon. Alternatively, you can go to the **Query Settings** panel on the right side of the screen and select the settings icon next to Source (or double Select Source). In the window that displays, update the server and database details, and then select **OK**.

[](https://learn.microsoft.com/en-us/training/modules/get-data/media/3-edit-creds-ssm.png#lightbox)

After you have made the changes, select **Close and Apply** to apply those changes to your data source settings.

### Write an SQL statement

As previously mentioned, you can import data into your Power BI model by using an SQL query. SQL stands for Structured Query Language and is a standardized programming language that is used to manage relational databases and perform various data management operations.

Consider the scenario where your database has a large table that is comprised of sales data over several years. Sales data from 2009 isn't relevant to the report that you're creating. This situation is where SQL is beneficial because it allows you to load only the required set of data by specifying exact columns and rows in your SQL statement and then importing them into your data model. You can also join different tables, run specific calculations, create logical statements, and filter data in your SQL query.

The following example shows a simple query where the ID, NAME and SALESAMOUNT are selected from the SALES table.

The SQL query starts with a **Select** statement, which allows you to choose the specific fields that you want to pull from your database. In this example, you want to load the ID, NAME, and SALESAMOUNT columns.

SQLCopy

SELECT

ID

, NAME

, SALESAMOUNT

FROM

FROM specifies the name of the table that you want to pull the data from. In this case, it's the SALES table. The following example is the full SQL query:

SQLCopy

SELECT

ID

, NAME

, SALESAMOUNT

FROM

SALES

When using an SQL query to import data, try to avoid using the wildcard character (\*) in your query. If you use the wildcard character (\*) in your SELECT statement, you import all columns that you don't need from the specified table.

The following example shows the query using the wildcard character.

SQLCopy

SELECT \*

FROM

SALES

The wildcard character (\*) will import all columns within the **Sales** table. This method isn't recommended because it will lead to redundant data in your data model, which will cause performance issues and require extra steps to normalize your data for reporting.

All queries should also have a **WHERE** clause. This clause will filter the rows to pick only filtered records that you want. In this example, if you want to get recent sales data after January 1st, 2020, add a **WHERE** clause. The evolved query would look like the following example.

SQLCopy

SELECT

ID

, NAME

, SALESAMOUNT

FROM

SALES

WHERE

OrderDate >= ‘1/1/2020’

It's a best practice to avoid doing this directly in Power BI. Instead, consider writing a query like this in a view. A view is an object in a relational database, similar to a table. Views have rows and columns, and can contain almost every operator in the SQL language. If Power BI uses a view, when it retrieves data, it participates in query folding, a feature of Power Query. Query folding will be explained later, but in short, Power Query will optimize data retrieval according to how the data is being used later.

## Next unit: Get data from a NoSQL database