Module 3: Power BI Data

Contents:

Module overview

Power BI offers a straightforward approach to report creation, and the ability to create and share dashboards without dependency on a report developer, or the need for Microsoft® SharePoint®. Microsoft Excel® has long been the tool of choice for data analysts who work in a self-service style. However, Excel does not offer a quick and easy way to share reports without the use of either SharePoint, or the creation of multiple copies of spreadsheets that quickly become out of date, or exist outside source control.

In recent years, power tools have been added to Excel: Power View, Power Query (known as Get & Transform in Excel 2016), and Power Pivot. Power BI brings much of this power into an integrated environment in the form of Power BI Desktop. Previously, Excel users were inconvenienced by needing to transition between the different power tools, but Power BI Desktop brings the tools together. This means that Power BI is fast becoming an obvious choice for the analysis and sharing of data.

However, analysts are likely to continue working with Excel for the near future. Power BI easily cooperates with Excel, and many other data sources. It's this ability to create reports rapidly, by using data from a combination of sources, which really puts the power into Power BI.

Objectives

After completing this module, you will be able to:

- Connect to Excel files and import data.
- Describe the data model and know how to optimize your data within the model.
- Use on-premises and cloud Microsoft SQL Server databases as data sources, along with the R script data connector.
- Take advantage of the features of the Power BI service by using Q&A to ask questions in natural query language and create apps.

Lesson 1: Using Excel as a data source for Power BI

In this lesson, you will learn how to connect to Excel from Power BI and import data. You will also learn how to update and refresh data.

Lesson objectives

After completing this lesson, you will be able to:

- Connect to files from the Power BI service and Power BI Desktop.
- Import data from Excel.
- Publish data from Power BI to Excel.
- Update files in Power BI.
-

Refresh Excel data in Power BI.

Connecting to files

- Connect to files from Power BI desktop or service:
  - Compatible file types include Excel, CSV, XML, and JSON
- From the Power BI service:
  - Connect to a local file, OneDrive Business or OneDrive Personal, or SharePoint – Team Sites
  - Maximum file size to upload to the Power BI service is 250 MB
- From Power BI Desktop:
  - On the **External data** menu, click **Get Data** from, and then choose file location from your local computer, or OneDrive
  - Connect to a folder to import multiple files
- Folder locations can contain different file formats

In Power BI, you can connect to various file formats. In addition to Excel, you import data from comma-separated values (CSV), XML, text, or JavaScript Object Notation (JSON) files, or a folder that contains multiple files in one of these formats. You can also import a Power BI report file that has the .pbix extension. When you import data directly into the Power BI service, the maximum size for any file format is 250 megabytes (MB).

You import files from your local computer, or connect to files on Microsoft OneDrive® Personal, OneDrive for Business, or SharePoint – Team Sites. Data that is imported from OneDrive or SharePoint into Power BI is automatically updated if the source file changes. For example, if additional rows are added to a table in a workbook, the changes are reflected in any reports and dashboards in Power BI, usually within an hour.

When importing CSV files, it's best to use a comma-delimited format, and include a header row. Fixed width CSV and text files are also supported. After selecting the file for import, the preview enables you to select the delimiter type, including comma, colon, semicolon, tab, fixed width, or a custom value.

Connecting to files from the Power BI Service

To connect to a file in the Power BI service, click **Get Data**. In the **Create new content** section, under **Files**, click **Get**. You then select from one of the following:

- **Local File**. Browse to a file that is stored on your local computer. Click **Open** to upload the data to Power BI.
- **OneDrive – Business** or **OneDrive – Personal**. Browse to the file that you want to upload, and then click **Connect**. Power BI creates a connection to the file; updates to the file are reflected automatically in Power BI.
- **SharePoint – Team Sites**. Click **SharePoint – Team Sites** to open the **Connection** dialog box. Either enter the URL of the SharePoint server and click **Connect**, or click **Connect** to view content that is available to you at the root level.

Connecting to files from Power BI Desktop

When you initially start Power BI Desktop, the splash screen gives you options to connect to data. If you have selected **Show this screen on startup**, you can click **Get data** to open the Get Data window. Alternatively, on the **External data** menu, click **Get Data**. This presents you with a list of the most common sources, or you can click **More** to open the Get Data window and view the full list of compatible data sources. The Get Data window breaks the data source connections into **All**, **File**, **Database**, **Power BI**, **Azure**, **Online Services**, and **Other**. Click **File** to view the list of compatible file formats, or select **Folder** to import a collection of files. When you select a file format, such as Excel, you can select a file from your local computer, or from a OneDrive location.

> **Note:** When you are using a folder location to import multiple files, you can include different file formats in the folder. After selecting the folder location, Power BI displays a list of the files that are stored in the folder. This includes any incompatible formats such as .jpg or .docx. When you click **Load** to import the data, Power BI ignores the files that it recognizes as not being data files.

Importing Excel files

- Two approaches to importing an Excel workbook:
  - Connect and use contents as datasets:
    - File size limitations
    - Data must be formatted as a table in Excel
    - Load the data and use Power Query Editor to apply transformations
  - Import an entire workbook, including Power Pivot and Power View, and explore as you would in Excel Online:
    - Import any .xlsx or .xlsm file to explore features
    - Power Pivot models are imported to created datasets
    - Power View content is imported as reports that can immediately be pinned to dashboards
    - Data source connections are imported for scheduled data refresh

If Excel is widely used in your organization, you can combine reports that have been created in Excel with the visualizations and sharing capability of Power BI, without losing the effort that went into creating the Excel workbooks in the first place. There are two approaches to importing Excel files:

1. Connect to an Excel workbook (.xlsx) and use the contents as datasets for your Power BI reports and dashboards.
2. Import a whole Excel workbook and explore the whole file, in the same way that you would by using Excel Online.

Importing Excel content as a dataset

When you import Excel content as a dataset, the workbook can consist of a data model and the core worksheet contents, but the file that you choose to upload can be no larger than 250 MB. Within the 250 MB limit, the core worksheet can be up to 10 MB, with the remainder of space used by the data model. If your workbook meets these criteria, you save the file to OneDrive for Business and connect to it from Power BI, in addition to viewing it in Excel Online.

There are several ways to reduce the size of the core workbook in a file that you want to import. Images and clip art elements can increase the size of the file, so remove these if possible. Remove cell shading and sheet background colors to reduce the size further. If the report contains a data model, you can move data from the worksheet to the data model. Furthermore, ensure that you exclude columns that are not necessary to the analysis that you want to perform. If your data originated from a data warehouse, it might include metadata columns that were added during the extract, transform, and load (ETL) process, such as Last Run Date, or Create Date. Look out for the inclusion of these columns and remove them where necessary. For more information about creating an efficient data model, see *Create a memory-efficient Data Model using Excel and the Power Pivot add-in* in the Office documentation:

Create a memory-efficient Data Model using Excel and the Power Pivot add-in

http://aka.ms/Ca9lsv

To import data from Excel into a Power BI dataset, the data must first be formatted as a table:

1. To convert columns of data into a table in Excel, first highlight the rows and columns that you want to include. Then, on the **Insert** menu, click **Table**.
2. After you have formatted your Excel workbook, return to Power BI, click **Get Data**, and then click **Excel**.
3. The navigator displays a list of worksheets and tables within the workbook. You

select the worksheets and tables that you want to import then click **Load** to import these immediately; or you click **Edit** to open the Power Query Editor to apply transformations.

4. After you have loaded the worksheets into Power BI, you can begin working with them as Power BI datasets.

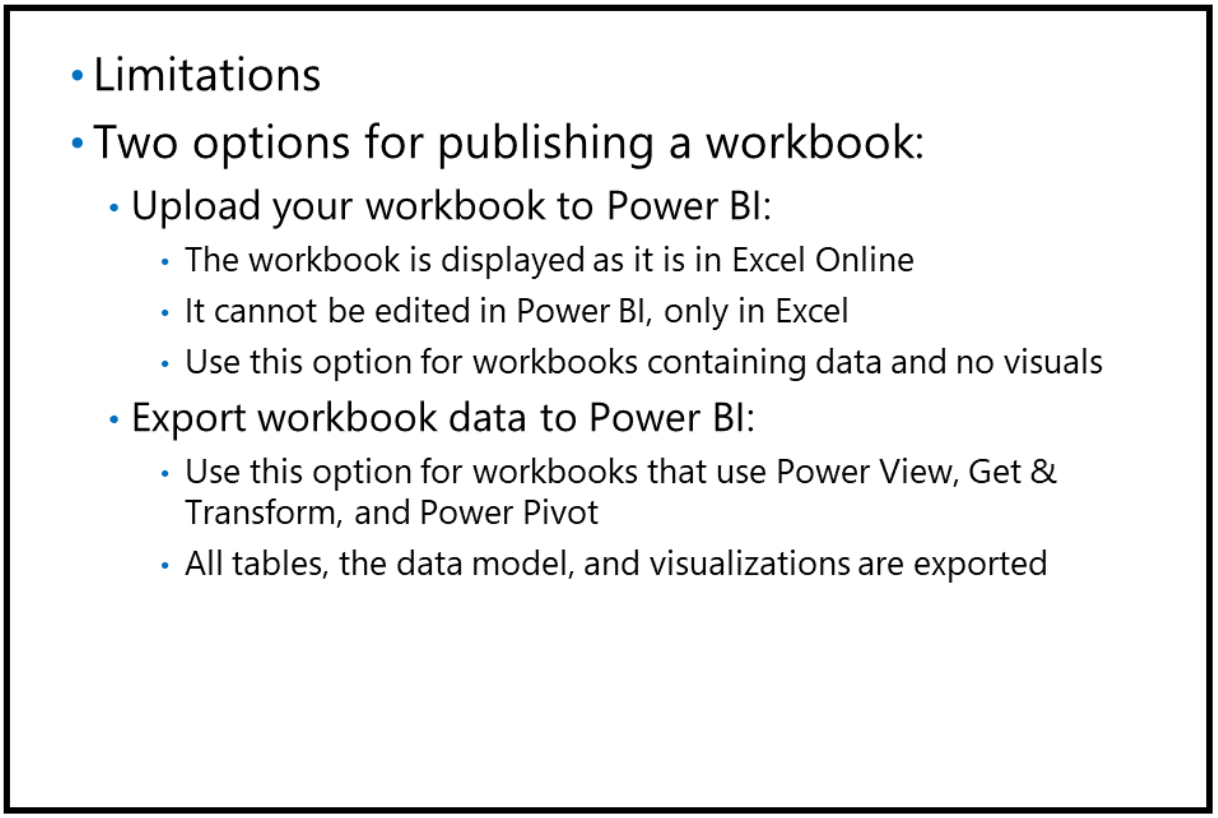
Working with a whole Excel workbook

Power BI can import any Excel .xlsx or .xls file, enabling you to explore features as if you were using Excel Online. If you have created data models by using Power Pivot, Power BI imports your tables, calculated columns, measures, and hierarchies. Power View sheets are imported and created as reports. As soon as the reports are created, you can begin pinning the visualizations to dashboards. Be aware that not all Power View visuals are supported in Power BI.

> **Note:** If you import an Excel workbook that uses Get & Transform or Power Pivot to connect to an external data source, you can set up a scheduled data refresh. After the import has completed, Power BI uses the connection information to make a direct connection to the data source. The data is then queried and refreshed, and visualizations are updated.

The process for importing Excel files that contain Power Pivot or Power View content is the same as for a standard data worksheet. You can import the content into Power BI Desktop or upload it to the Power BI service from your local computer, or from OneDrive.

Publishing to Power BI from Excel



You can also use Excel to publish workbooks straight to the Power BI service, where you create reports and dashboards, and then share visuals with your colleagues.

Limitations

There are several limitations that you must consider before publishing to Power BI from Excel:

- You must save the workbooks that you want to publish to OneDrive for Business.
- You must use the same account for Microsoft Office, OneDrive for Business, and Power BI.
- Before you publish a workbook, it must contain content that is supported in Power BI; you cannot publish an empty workbook.
- Encrypted or password-protected workbooks, or workbooks that have Information Protection Management, cannot be published.
- Modern authentication must be enabled. The **Publish** option is not available on the **File** menu if modern authentication is set to disabled.

Publishing a workbook

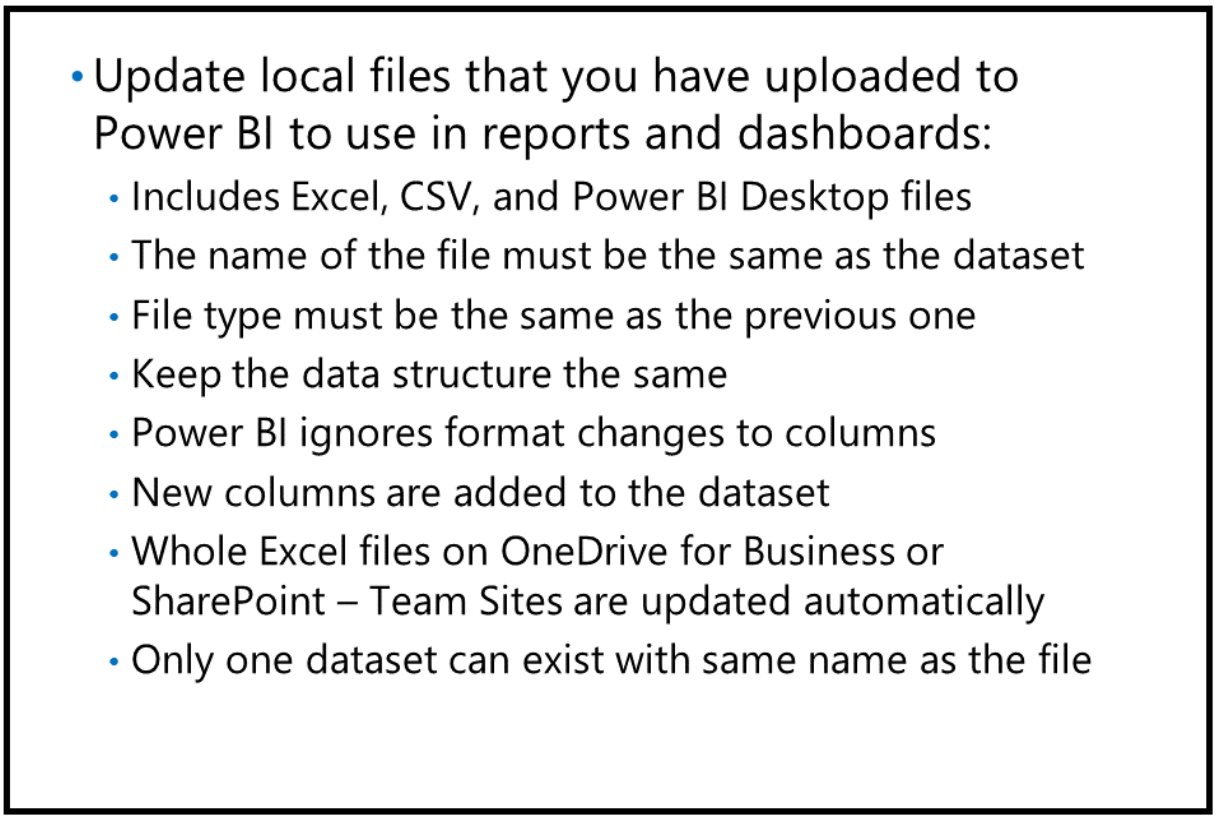
If necessary, save your Excel workbook to OneDrive for Business, open it from this location, click **File**, and then click **Publish**. This gives you two options for uploading your file to Power BI:

1. **Upload your workbook to Power BI**. If you choose this option, your workbook is displayed as it is in Excel Online, but you can still pin visuals in your worksheets to dashboards. You will not be able to edit your workbook in Power BI, but you can click **Edit** to open the workbook for editing in Excel Online, or on your computer. The changes are saved to the version on OneDrive. Uploading your workbook does not create any datasets in Power BI. Workbooks uploaded to Power BI have an Excel icon, to indicate that they are uploaded workbooks. This is the best option if you only have data in your workbooks, or PivotTables and PivotCharts that you want to view in Power BI. This option is similar to the Manage and View Excel in Power BI feature, in the Power BI service. Click **Get Data**, under **File**, click **Get**, click **OneDrive - Business**, and then click **Connect**.

2. **Export workbook data to Power BI**. Choose this option if you have a workbook that uses Get & Transform or Power Pivot to load data into a data model, or if the workbook contains Power View visualizations that you want to view in Power BI. Unlike the upload option, this option exports any supported tables and data models into new datasets in Power BI. Power View sheets are converted to Power BI reports, so you can instantly create dashboards from the visualizations. Furthermore, you can continue to edit your workbook in Excel. When you save changes, they are synchronized with the Power BI datasets, usually within an hour. For more immediate results, you can click **Publish** again to update the content without having to wait. Reports and dashboards that use the visualizations are updated. This option is similar to the Export Excel data into Power BI feature in the Power BI service. Click **Get Data**, under **File**, click **Get**, and then click **OneDrive - Business**.

When you click **Publish**, and choose the upload or export option, Excel signs in to your Power BI account by using the credentials for your Office account, and then publishes the workbook. The **Publishing to Power BI** status bar displays the progress of the operation.

Updating files in Power BI



If you upload a local file to Power BI to use as a dataset in your reports and dashboards, you can make changes to the file and upload it again. Providing the file name is the same, Power BI can update the file. This applies to Excel, CSV, and Power BI Desktop files. Several limitations apply:

- The file names must have the same name and the same type. If you have an Excel file named Finance, it will not be replaced with a Power BI Desktop file named Finance.
- The structure of the data must stay the same. Renaming or deleting columns that are used in a report or dashboard will break the dependent visuals.
- Power BI ignores any format changes to columns so, for example, you can change

a value from 75 percent to 0.75.

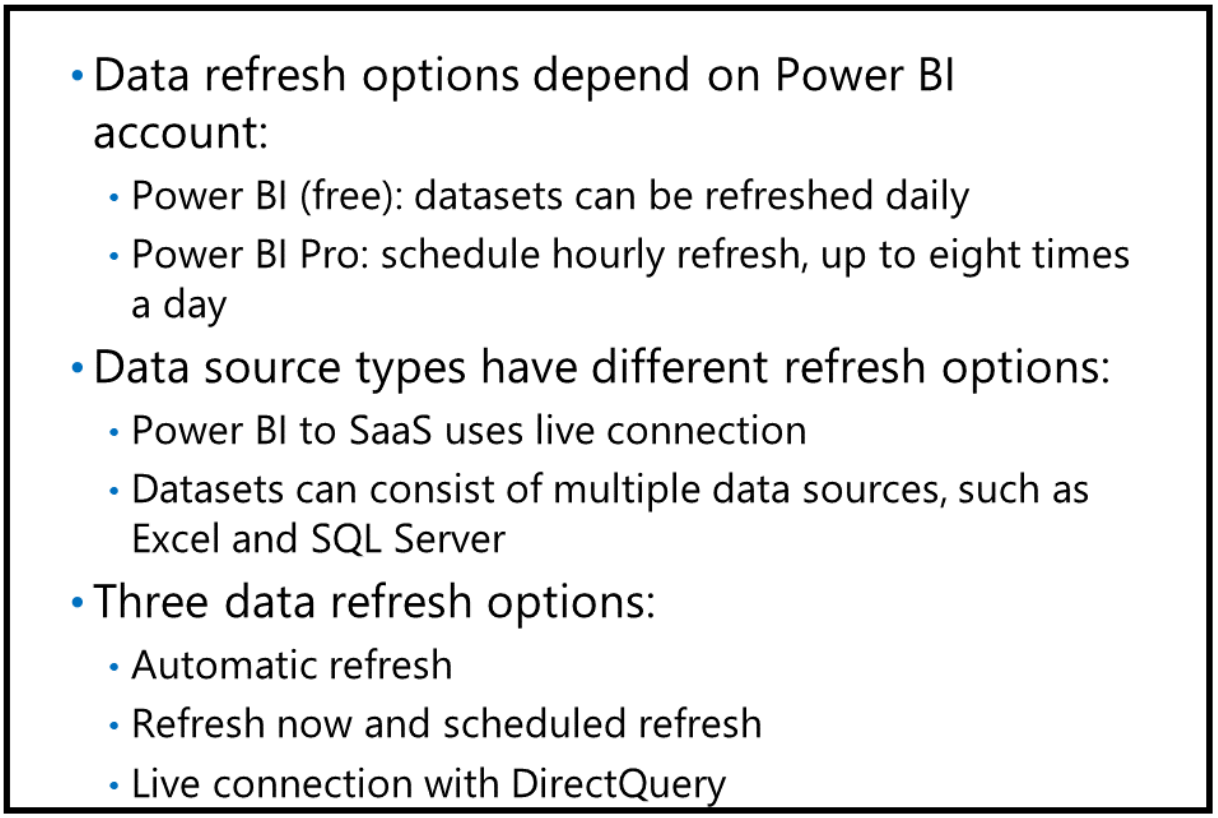
- New columns are added to the dataset, but they are ignored until they are used in a visual.
- When you import whole Excel files from OneDrive for Business or SharePoint – Team Sites, the changes to the file are automatically reflected in Power BI.

To update a file in the Power BI service:

1. Click **Get Data**, under **File**, click **Get**, and then click **Local File**.
2. Browse to the file that you want to replace, and then click **Open** to upload the file.
3. Click **Import** to connect to the data and Power BI displays a message to say that you already have a dataset with that name.
4. Click **Replace it** to upload the updated file.

> **Note:** If more than one dataset has the same name as the file that you're updating, Power BI cannot update the dataset until you rename the dataset that is not sourced from the file. There must be only one dataset with the same name as the file that you want to update.

Data refresh



The way in which data refresh works in Power BI depends on the subscription service that you are using, and the type of data source.

Subscription types

The options that are available depend on whether you have a Power BI subscription, which is free of charge, or a Power BI Pro subscription:

- **Power BI (free)**. Datasets can be scheduled to refresh daily, with a maximum of 10,000 rows per hour for streaming data in dashboards and reports by using the Microsoft Power BI REST application programming interface (API), or Microsoft Azure Stream Analytics.
- **Power BI Pro**. Using a Power BI Pro account, you can schedule an hourly refresh, with up to 1 million rows per hour for streaming data in dashboards and reports by using the Microsoft Power BI REST API, or Stream Analytics. You can have up to eight hourly data refreshes per day. Furthermore, Pro accounts include data refresh for live data sources with full interactivity (Azure SQL Database, Azure SQL Data Warehouse, Spark on Azure HDInsight®), on-premises data sources

that require a Power BI gateway, and on-premises SQL Server Analysis Services that require the Analysis Services Connector.

Data source types

The type of data source from which you are extracting the data determines how the data is refreshed. Software as a service (SaaS) data is automatically refreshed, so you don't need to do anything to update it.

Database connections in SQL Server Analysis Services use a live connection, which means that they always display the latest data.

After you have created a dataset in the Power BI service, it appears in the **Datasets** list in the **My Workspace** pane. You can use the ACTIONS list to refresh the data or schedule a refresh. During a refresh, Power BI connects to the data source by using the credentials that are stored in the dataset. The dataset data is refreshed, and the reports and dashboards that use this dataset reflect the changes immediately.

A dataset might consist of multiple data sources. For example, in Power BI Desktop, if you acquire data from an on-premises server running SQL Server, and other data from an Excel workbook, a single dataset is created when you publish to the Power BI service. However, this dataset contains two data sources that have connection information to both SQL Server and Excel. Be aware that, when you choose to refresh a dataset, Power BI connects to all of the data sources in the dataset so that it can refresh the data. This ensures that all data within reports and dashboards is consistently up to date.

Data refresh types

You can refresh most datasets in Power BI, but the type of data from which the dataset was created, and the data sources to which the dataset connects, determine whether you need to update it. Power BI has the following refresh options:

- **Automatic refresh**. Power BI configures the data refresh settings for data sources

that benefit from an automatic refresh. For example, for files that are loaded from OneDrive, the data that does not come from an external source is refreshed approximately every hour. Although you can schedule a refresh to occur more frequently, it is unlikely that this would be necessary.

- **Refresh now and scheduled refresh**. Refresh now manually refreshes a dataset, or you can configure a schedule to run on a regular basis. Use this option for Power BI Desktop (.pbix) files, and Excel workbooks that connect to on-premises and external online data sources.
- **Live connection with DirectQuery**. If you use DirectQuery, a live connection exists between Power BI and the data source, such as a database in Azure SQL Database. You always see the latest data from the source and no manual configuration is required.

For more information about data sources and the refresh options that are available to each type, see *Data refresh in Power BI* in the Power BI documentation:

Data refresh in Power BI

http://aka.ms/Bq486n

Demonstration: Importing files from a local folder

In this demonstration, you will see how to:

- Import data from an Excel file.
- Import data from a CSV file.

Check your knowledge

Select the best answer

Which of the following file formats is not a compatible data source in Power BI?

CSV

TXT

XML

SQL

JSON

Check answer Show solution Reset

Lesson 2: The Power BI data model

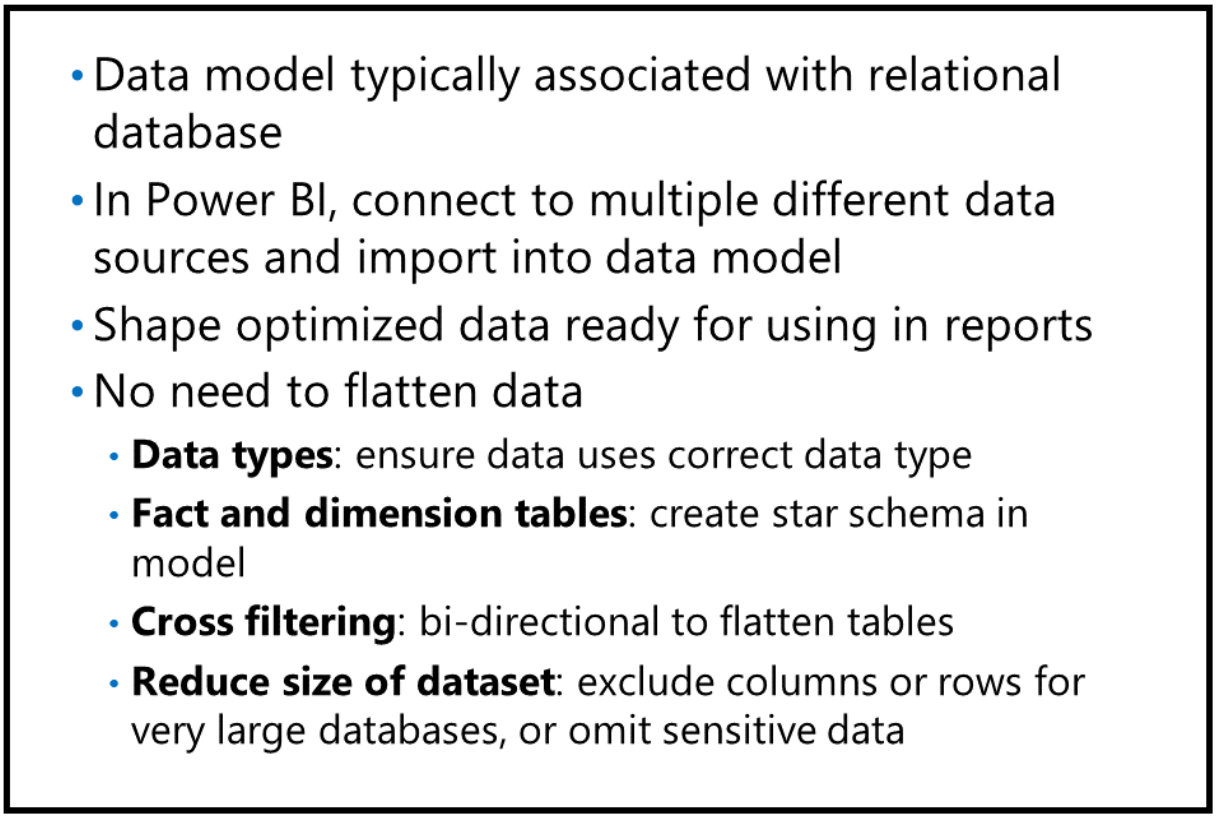
This lesson explores the Power BI data model. You will learn how you can use features within the data model to manage your data.

Lesson objectives

At the end of this lesson, you will be able to:

- Describe what a data model is.
- Create and manage relationships in your data.
- Optimize the data model so your data is ready to use in visualizations.
- Understand how hierarchies can help you analyze your data by drilling down through levels.
- Create hierarchies that support your reporting and analytical requirements.

What is a data model?



A data model is typically associated with a relational database, such as Microsoft SQL Server. In Power BI Desktop, you can connect to multiple different data sources using queries, and bring the data together in the data model. You create relationships between the tables imported from the various sources. There's no need to flatten the data you import into separate tables, you leave it in the original table structure, and create relationships. You then use these related tables for calculations or measures, and enrich the data in the model by creating calculated columns. It is best to design your model to help you build visualizations within your reports. Within the data model, you create calculated tables and columns, relationships, hierarchies, change data types, defaults, and properties. If you get all this right in your model, the creation of reports is a much smoother process, and this produces results that are more accurate.

Data types

Data that comes in from a database system such as Microsoft SQL Server is likely to be correctly typed. However, if you import from other sources, such as CSV, or the web, the data types might need to be changed. It is a good idea to correct any data types in the model so you can then begin applying the correct formatting. For example, you apply formatting to a datetime column to display the data appropriately

for your region, and then use the split column function to extract the day, week, or year part into a new calculated column.

A wrong data type might cause incorrect results within your visuals. When importing numbers, be accurate and determine whether you need precise rounding. If you have financial data that you don't want to be rounded down, you use the Fixed Decimal Number data type, which includes four digits to the right of the decimal point. This is the same as the Currency type in Power Pivot. Furthermore, if you intend to include maps within your reports, ensure that address columns have the correct geo category types—for example, set Country, State, or Region to the corresponding data. After changing any data types that need altering, it's good practice to then check that columns that Power BI has set as the default for sort orders, or aggregating, are correctly determined.

Fact and dimension tables

When we record information about an event, it is known as a **fact** in business intelligence (BI) terms. This data is stored in **fact tables**, excluding any descriptive details about that event. The details of the event are held in **dimension tables**, which are also referred to as lookup tables in a relational database. A data warehouse stores data in fact and dimension tables, and we have the same concept in Power BI. A fact table is generally on the many end of a one-to-many relationship—for example, one customer might place many orders. One fact table is usually related to many dimension tables, which creates a **star schema**, with the fact table in the center of the model, surrounded by the related dimension tables. You use dimension tables to slice and dice your data—for example, to see sales by customer, region, age range, marital status, or time.

Cross filtering

When you turn on **bi-directional cross filtering** in your relationships, this enables the tables in your star schema to operate as if held in a single table, and you can join and aggregate values between dimension tables. There is no need for you to flatten your tables, as you can achieve this by changing the cross-filter type. When you have multiple fact tables, initially start with single direction cross filtering to relate the tables

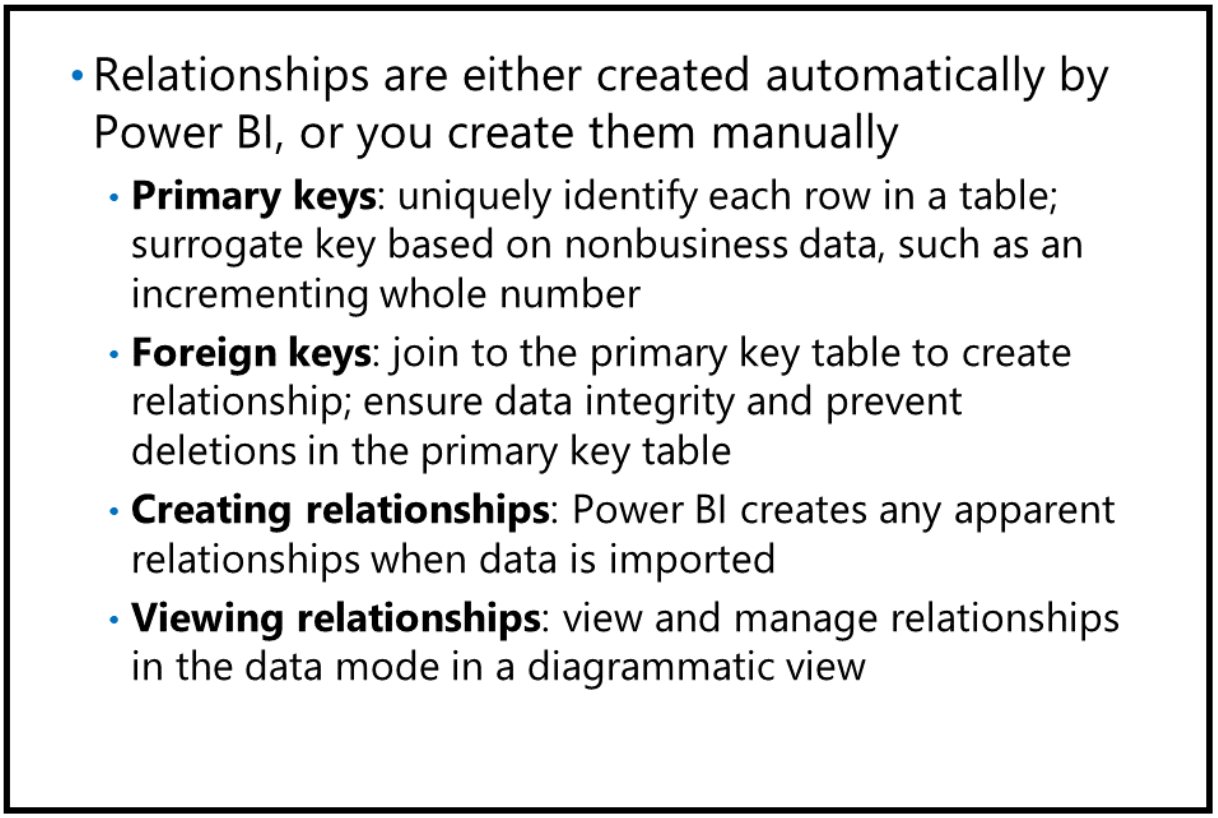
correctly, before adding bi-directional cross filtering. This method ensures you have your basic relationships working first, and prevents confusion and ambiguity in your visualizations. You can also use bi-directional cross filtering to overcome Power BI's lack of support for many-to-many relationships.

However, you might have a problem creating the relationships you need when you have two fact tables joined to one or more dimension tables. In this case, you create calculated tables using existing dimension tables and then create new relationships. To do this, delete the relationships from one of the fact tables to the dimension tables, and then create new calculated tables using the existing dimension data. You can then create relationships between these new tables to the fact table from which you deleted the relationships, and set the cross-filter direction to be bi-directional. The calculated table works the same as any other table, and is updated when the model is updated, though it will increase the size of your model.

Reducing the dataset size

While you can use bi-directional cross filtering and avoid the need to flatten your data, in some instances you might want to do some flattening of your data before import. If you have a very large database, it might be beneficial to join tables, or apply filtering within the query to reduce the volume of data you include in your model. Furthermore, your database might contain sensitive data that you don't want to import, so for security reasons, you might choose to exclude such data in the query.

Managing data relationships



Relationships might be created automatically by Power BI or created manually—a relationship describes how one table relates to another. Columns are related using key fields, comprising primary and foreign keys.

Primary keys

A primary key column uniquely identifies each row within a table. At the end of a relationship, one of the tables must contain unique values. A primary key is frequently created using a surrogate key, particularly in a database. A surrogate key is usually a sequential number, starting at 1, and incrementing by 1, for each row that is added to the table. A surrogate key uses information that is not actually business data. However, business data can be used and an **Employee** table, for example, might have a primary key column based on the social security number, which uniquely identifies each employee, as shown in the following table:

SocialSecurityNo	FirstName	LastName
123 456 ABC	Lucinda	Smith
987 321 ZYX	Elizabeth	Jones

If another employee named Lucinda Smith joins the organization, she can be uniquely identified because the data in the **SocialSecurityNo** column will differ from the data stored for the existing Lucinda Smith. In many cases, data will not have a natural key. If you have a **Category** table, the data is likely to be a list of category names, which will not necessarily be unique. To make them unique, we add a surrogate primary key column named **CategoryID**, as shown in the following table:

CategoryID	Category
1	Frozen Food
2	Pet Supplies
3	Dairy

Each time a new category is inserted into the table, the CategoryID number is incremented. The primary key forces uniqueness within the table, and becomes the basis for foreign keys.

Foreign keys

Foreign keys enforce data integrity by ensuring that the data table in one table is correctly related to data in another table. Furthermore, it prevents rows in the primary key table from being deleted, leaving orphaned records in the foreign key table. For example, a **SubCategory** table has a **SubCategoryID** primary key column to uniquely identify each subcategory; it also includes a **CategoryID** foreign key column to the **Category** table, which acts as the parent within the relationship. This relationship can be seen in the following table:

SubCategoryID	CategoryID	SubCategory
1	1	Ice Cream
2	1	Sorbet
3	2	Dog Treats
4	2	Dog Food
5	2	Cat Food
6	3	Cheese

The relationship between the tables enables you to work with the data as if it was one table:

SubCategoryID	CategoryID	Category	SubCategory
1	1	Frozen Food	Ice Cream
2	1	Frozen Food	Sorbet
3	2	Pet Supplies	Dog Treats
4	2	Pet Supplies	Dog Food
5	2	Pet Supplies	Cat Food
6	3	Dairy	Cheese

The **Frozen Food** category cannot be deleted from the **Category** table, because there is at least one related table in the **SubCategory** table. If the category were deleted, the child records in the **SubCategory** table would have incomplete data that would not make for useful reporting. By using relationships, this prevents repeated data. In the above table showing the data joined from the Category and SubCategory table, each Category name is repeated for the corresponding row in the SubCategory table, however, it only exists once in the Category table. This is important when data is updated, as only one row requires altering.

Creating relationships

You use relationships to work with data as if all the related tables were a single table. For example, if your **Product** table is related to the **SubCategory** table using a **SubCategoryID** column, you can display the product name from the Products table alongside the subcategory name from the SubCategory table. The SubCategory table then joins to the Category table making the data appear seamless in your reports, and enabling users to slice data.

When you import data into the data model, Power BI detects existing relationships. When you connect to a relational database such as SQL Server, the data is most likely to be related already, and Power BI detects the existing connections between the tables to create the relationships. However, if you import two tables with a common column but without a relationship, Power BI is likely to detect the

commonality and still create a relationship. For example, if you have two tables named **Department** and **Manager**, each containing a **DepartmentID** column, even though there is no relationship between the tables, Power BI works out that the columns and data match, and automatically creates the relationship.

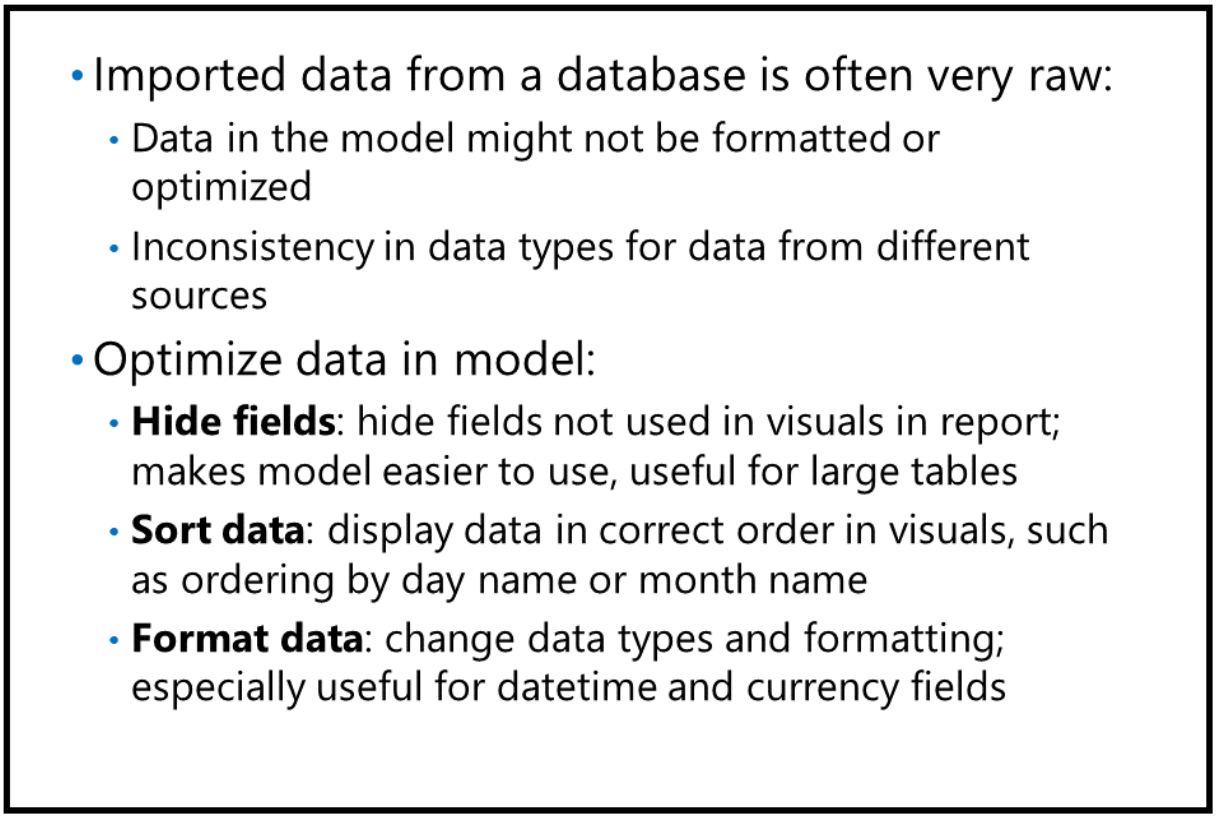
If you drag fields onto the report canvas to combine with other data in a visual, Power BI checks to see if the data is related and can be combined. If it can't relate the fields, a warning message appears, and you can click to choose whether you want Power BI to autodetect the relationships between the fields, or to manually create a new relationship.

Viewing relationships

You view and manage the relationships within your data model by clicking **Model** on the left side of the Power BI Desktop window. This displays your tables diagrammatically, and you instantly see how one table relates to another. Furthermore, you can move the tables around while maintaining the links between them, so they are easier to understand. If you have many tables within your model, you use the zoom feature to zoom in or out and see a closer view of the tables and columns. The **Fit to screen** button lays out the tables so you see all tables and relationships in a single view on the screen. After moving your tables around, you use the **Reset layout** button to return to the default layout.

Before you begin adding visuals to your report and hooking up the data, ensure you have correctly established the relationships, because this facilitates the ability to present accurate data.

Optimizing the model for reporting



Data is frequently in a raw format when you import it into Power BI, especially if you have taken it directly from a database. Furthermore, you might have fields you don't need if you have not been able to use a query to reduce columns in the dataset you have imported. When you combine data from different data sources into your data model, it is highly probable that the different source systems or files have applied different formats or data types. There's a few things you can do to optimize your data, and make it more consistent. This helps you to work with your data more efficiently, focusing on the information you need. It is also helpful to colleagues or anyone with whom you might share the data.

Hiding fields

It's a good idea to hide fields that you know you are not going to use in your visuals. Hiding a field removes the field name from the **Fields** pane, and neither the column nor the underlying data is deleted. To hide a field, in **Report** view, right-click the field in the **Fields** pane, and click **Hide**. Right-click anywhere on a table or field to open the menu and choose **View hidden**, so hidden fields reappear in the list of fields, and show in their original position in the order of the fields. Hidden fields are shown in grey text to indicate their hidden status. You can also select **Unhide all** to make

hidden fields visible again. If you switch to the data view, you will see that hidden columns also display with the column header and data in grey text.

Sorting data

If you drag a column such as **DayOfWeek**, or **MonthName** to a column chart visual to display sales figures, the values are automatically ordered alphabetically. However, it's unlikely you want the data to be ordered in this way, and require the columns to be in day or month order, rather than by name. In **Report** view, in the **Fields** pane, click the column to sort and then, from the **Sort** group on the **Modeling** ribbon, click **Sort by Column**. When you select a column from the Sort by Column list, you will see that Power BI has selected one column as the default for the sort, so when this best guess is incorrect, it is quick and easy to change. Choose another column, such as **DayNumber**, or **MonthNumber**, so the columns are ordered correctly. For example, rather than showing as April, August, December, and so on, month names now appear as January, February, March, and so on. This makes data analysis much easier as the user can read the data in the correct time order.

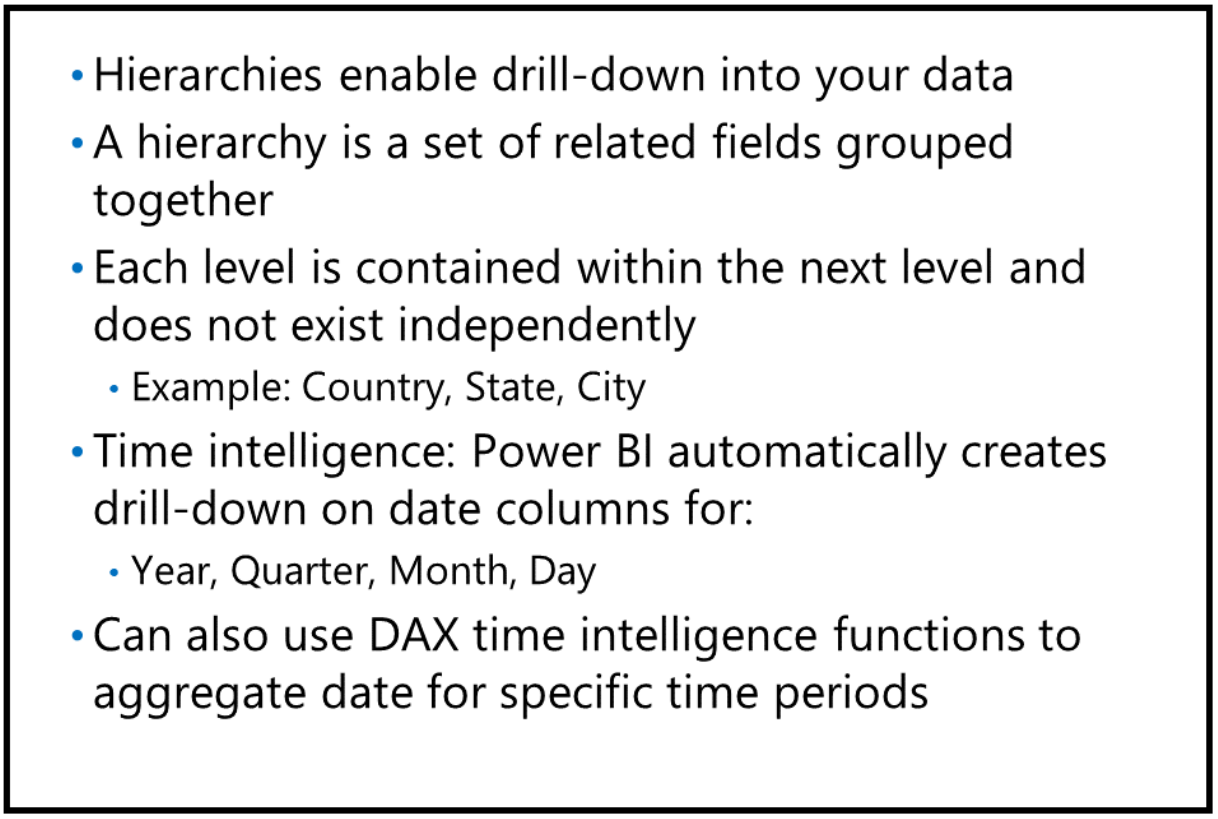
> **Note:** This ability to sort data is also applicable to other items you usually refer to in a specific order, such as product categories, projects, or departments. When working with visuals, try to present data in the optimum way for enabling the end user to quickly digest the presented information.

Formatting data

Changing data types and formatting data are good ways of optimizing your data. When you drag a datetime or currency field onto the report canvas, you might find the style of the date is not easy to read, or not in your local formatting, or that sales figures display without a currency symbol. From the **Formatting** group on the **Model** tab, you select the **Data type** and **Format** menus to customize your data. Power BI enables menu formatting items that are relevant to the data type of the column that you highlight in the **FIELDS** pane. You might wish to change a data type first, and then change the available formatting options. You can format how datetime data types are displayed, in addition to setting financial columns to always include your

local currency symbol. This presents the data with clarity in your reports and dashboards.

What are hierarchies?



You use hierarchies to drill down into your data. A hierarchy is a set of related fields grouped together that you can use to drill from one level in the hierarchy to the next. Each level within the hierarchy is contained within the next level—it cannot exist independently. Power BI will automatically create a hierarchy for you on datetime fields; however, you can also create your own hierarchies within the model to suit your requirements for analysis. The following is an example of an address hierarchy you could create in Power BI and use to drill through from top to bottom:

- Country
- State/Province
- City/Town
- Street

- House Number or House Name

Time intelligence

If you have date columns in your data that you have used in a report in Power BI, you might have noticed that the ability to drill through is created for you. Power BI automatically splits date data into:

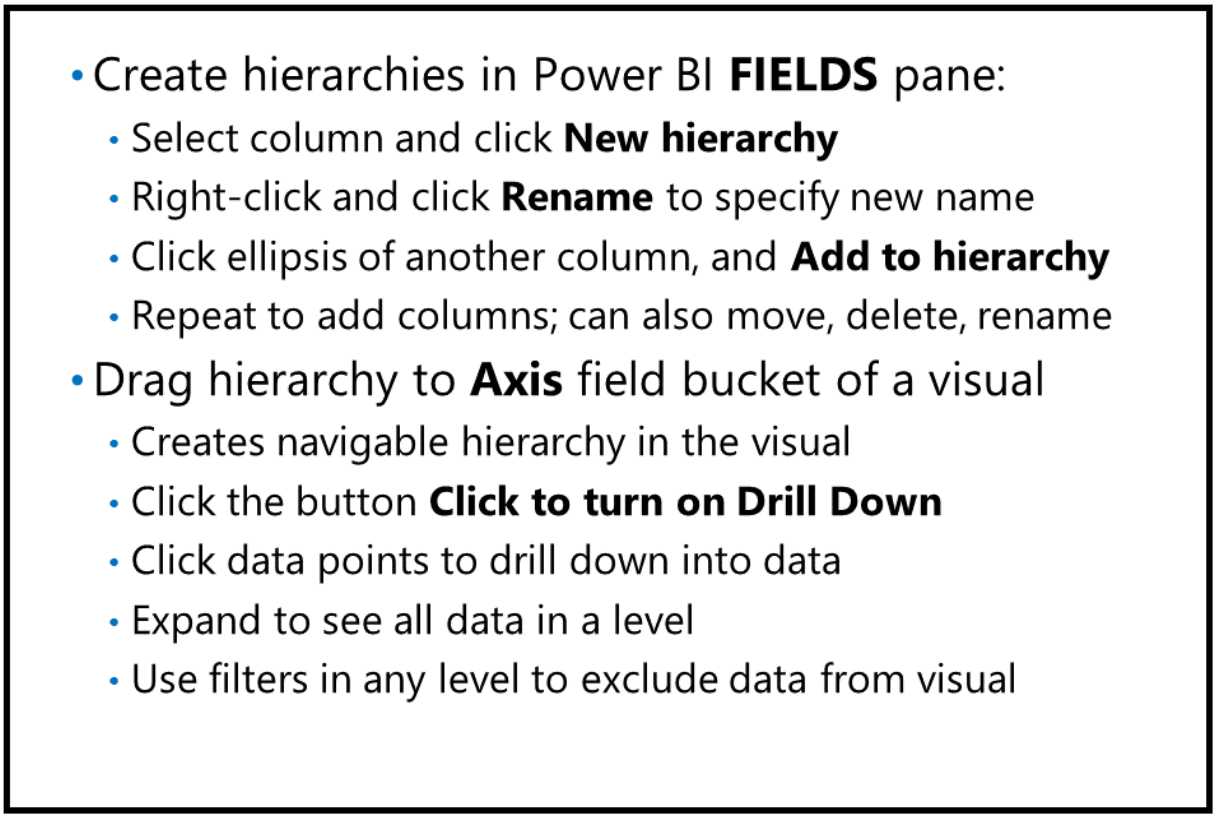
- Year
- Quarter
- Month
- Day

If you drag a date column to the axis field bucket of a visual, you see this in action, because you can immediately start drilling into the data over time. You use DAX time intelligence functions to aggregate your data over different time periods. DAX is covered in detail in Module 5 of this course, but to find out more information, see *Timeintelligence functions* in Microsoft Docs:

Time-intelligence functions

https://aka.ms/umjsqc

Creating hierarchies



You use hierarchies to drill down into your data. By creating custom hierarchies in Power BI Desktop, you can drill through levels applicable to your data, to support your exact analytical requirements. Furthermore, you can add multiple hierarchies to a table in your data model. Use the following steps to create a new hierarchy:

1. In the **FIELDS** pane, expand the table where you want to create the hierarchy.
2. Click the ellipsis on the column you want to use as the top level of your hierarchy, and then click **New hierarchy**. This creates a new field below the column you selected. By default, this is given the name of the column—for example, **Product Hierarchy**. You right-click the new hierarchy and click **Rename** to give it another description.
3. Click the ellipsis on the column where you want to add to the hierarchy at the next level down. Click **Add to hierarchy**, and then click the name of the hierarchy. If you have already created a hierarchy, it will be listed in the menu.
4. Repeat step 3 for each column you want to add to the hierarchy.
5. If you have added a column incorrectly, you click it in the hierarchy and click **Delete**, or drag the column and move it up or down in the hierarchy to reposition

it. To rename a field in the hierarchy, right-click the field and then click **Rename**.

After you have added all columns to complete the hierarchy, you drag the hierarchy to the Axis field bucket, such as on a column chart. You will see the data aggregated for the values in the top level of your hierarchy. To begin drilling down into the lower layers, click the **Click to turn on Drill Down** arrow icon in the top right corner of the visual to enable drill-down capability. When you click a data point, the visual drills into the next level in the hierarchy. In the left corner of the visual, the title shows the levels in the hierarchy the data is displayed for—for example, using an address hierarchy that might be **Sales by Country and State**. When you drill into the next level, this becomes **Sales for Country, State and City**. You will see which filters have been applied in the **FILTERS** list. This is particularly useful for hierarchies with deeper levels.

Above the title are further buttons you can use to traverse the data. Click the **Drill Up** button to move up one level in the hierarchy. Use the button with two down arrows, **Go to the next level in the hierarchy**, to go to the next level down, without filtering on data, so you don't have to click a data point and drill into one value within a level. When the data cannot be drilled into any further, the button is disabled and displays **At the lowest level of data**.

The **Expand all down one level of the hierarchy** button enables you to expand the hierarchy: for example, with a hierarchy of Country, State, City, at the State level, rather than clicking into each state to see all sales for every city, you can expand to see all values at the next level. So if the visual currently displays all states within the United States that have sales data, rather than clicking into each state to see all the cities, use the **Expand all down one level of the hierarchy** button to show sales for every state and every city.

In the **Axis** field bucket, click the **X** icon to remove a level in the hierarchy; for example, click to remove **State**, and go from **Country** directly to **City**. To add levels back into the hierarchy, click the arrow alongside the hierarchy name in the Axis field bucket, and then click **Show all levels**. You will achieve the same result by clicking the check boxes in the hierarchy in the **FIELDS** pane. You can also right-click a data

point and select **Exclude** to remove it from the visual. You can include or exclude values using **FILTERS**.

Furthermore, you can add filters to the columns included in the hierarchy, hide columns, and create new measures and columns, so the hierarchy is fully customized for your reporting requirements.

Demonstration: Creating a hierarchy

In this demonstration, you will see how to:

- Create a hierarchy.
- Use the hierarchy to navigate data.

Check your knowledge

Discovery

Discuss some of the different data sources that you might use in your organization to import data into the data model in Power BI. What problems would you need to overcome? How easy would it be to relate the data in tables from different sources? How easy do you think it would be to use data from the web within your model?

Show solution Reset

Lesson 3: Using databases as a data source for Power BI

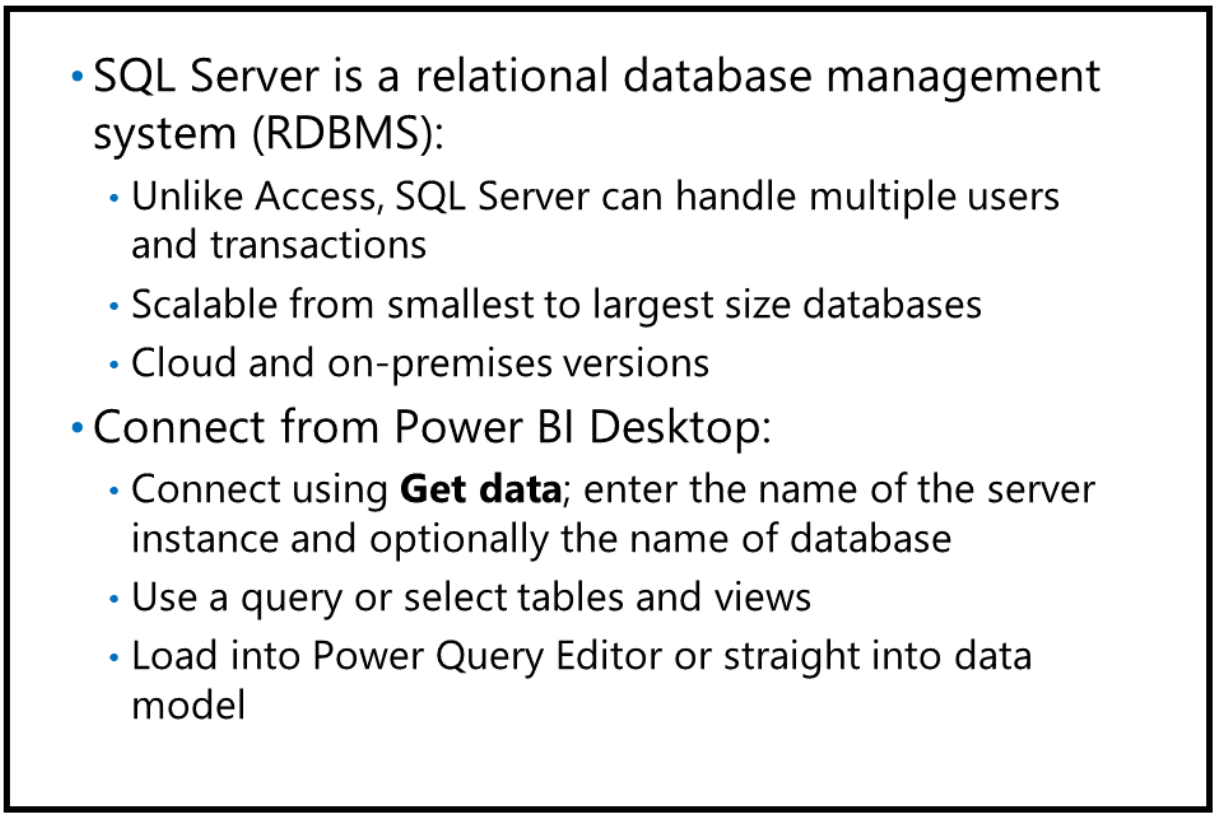
In this lesson, you will learn how to connect to on-premises and cloud instances of SQL Server, SaaS connectors, the R script data connector, and other data sources.

Lesson objectives

At the end of this lesson, you will be able to:

- Connect to SQL Server databases.
- Import data from other databases, web pages, and files.
- Use the R script data connector to import predictive data.

SQL Server



Microsoft SQL Server is a popular relational database management system (RDBMS). Unlike Microsoft Access, which is designed for a single user or very small company, SQL Server can handle multiple user connections, high volumes of transactions, and scales from the smallest to largest of databases. SQL Server can run in the cloud, but on-premises instances remain widespread, particularly in medium to large enterprises. The steps for connecting to SQL Server from Power BI are much the same as connecting to other database systems, such as Oracle, MySQL, and IBM DB2.

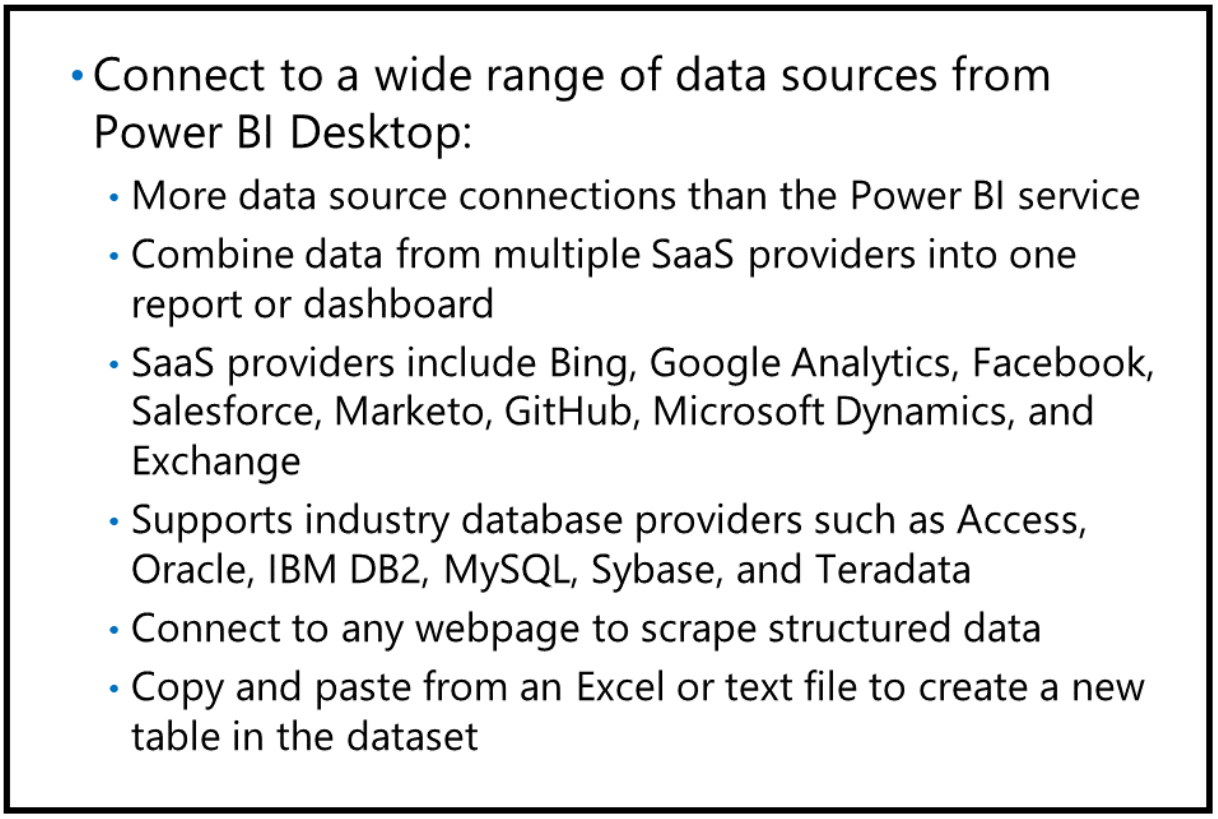
Connecting from Power BI Desktop

To connect from Power BI Desktop:

1. Open Power BI Desktop, and click **Get data**.
2. Click **SQL Server database** and then click **Connect**. This opens the SQL Server database connection dialog.
3. Type the name of the SQL Server instance into the **Server** text box. If more than one instance is running on the server, you might need to type in **<servername>\<instancename>**.
4. Optionally, you can type the name of the database you want to connect to in the **Database** text box. If you don't include this, you will connect using the default database associated with your account. However, if you expand **Advanced options** and want to use a query to return data, you will need to specify the name of the database in the Database text box.
5. You can choose to **Import** the data into Power BI, or use **DirectQuery**, which does not import the data and queries the underlying data source as you create and interact with a visualization. DirectQuery is useful for very large databases that are likely to import big datasets that would be slow to work with. If you are uncertain, leave **Import** selected, and click **OK**.
6. If you have chosen to use a query, you will see a sample of the results in the next window. Otherwise, you can select tables and views from the list to see a preview of the data. To import data, select each table and preview what you want to include.
7. Optionally, you can click **Select Related Tables** to import data from any tables that have a relationship to the one you have selected. This feature is particularly useful for fact tables that comprise multiple foreign key relationships to dimension tables.
8. After selecting the tables that you want to import, click **Edit** to load the data and begin working on it in the Power Query Editor; or click **Load** to load the data and return to the main Power BI window.

After your data has been loaded into the Power BI data model, you can begin shaping and transforming the data, and applying other optimizations.

Other data sources



Power BI offers a wide choice of compatible data sources that you can use for creating datasets in your reports and dashboards. You have more choice of data sources when you use Power BI Desktop than when you use the Power BI service. After you have imported data into Power BI Desktop from a source that you cannot directly connect to by using the Power BI service, you upload the dataset to work with it on the Power BI portal.

SaaS connections

You connect to an increasing number of SaaS providers to import data from the third-party online solutions that your organization uses. From Power BI Desktop, you import data from different SaaS providers and combine the data in reports and dashboards. For example, you could create a report that showed marketing data from Facebook and MailChimp campaigns, combined with the resulting sales that used data from Salesforce. SaaS providers include, but are not limited, to:

- Google Analytics
- QuickBooks Online
- MailChimp
- Facebook
- Dynamics 365
- Microsoft Exchange
- Active Directory®
- Salesforce
- Marketo
- GitHub
- Zendesk

Other databases

Power BI includes support for the main industry databases for importing data. Database connectors include:

- Microsoft Access®
- Oracle
- IBM Db2
- MySQL
- SAP HANA
- PostgreSQL
- Sybase

- Teradata

Web page data

From Power BI Desktop, you can connect to any webpage to extract the data. Power BI scrapes the data into tables in the dataset. Depending on the webpage that you are scraping, you might not be able to determine table names or the structure of the data, but you can perform operations to rename the fields and tables after importing the data into Power BI Desktop. To do this:

1. On the ribbon menu, click **Get Data**.
2. In the **Get Data** window, click **Other**, click **Web**, and then click **Connect**.
3. Type or paste the webpage address into the **URL** box, and then click **OK**. If you have previously created a parameter, Power BI gives you the option to use a parameter value for the URL. Power BI imports the structured data, and ignores page titles and text.

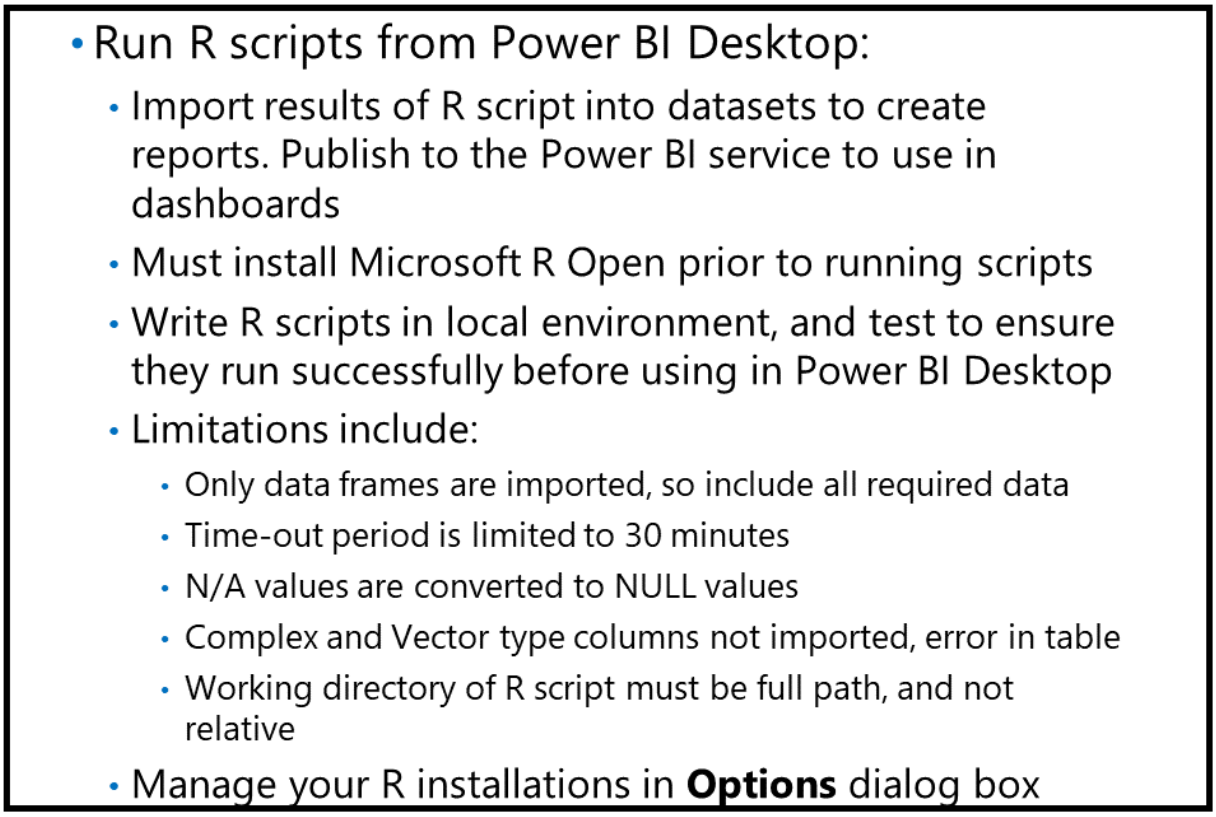
Copy and paste

You can quickly create a table in Power BI by copying and pasting data directly from Excel or from a text file:

1. Copy the data you want to use to the Windows clipboard.
2. On the **Home** ribbon, click **Enter Data** to open the Create Table window.
3. Right-click in the first cell and then click **Paste** to paste the data from the Windows clipboard. The table is created within your dataset and you can work with it just as you do with other tables. If you include column headers, Power BI detects these and uses them as the column headers in the new table. You can also manually enter data and add columns.

4. In the **Name** box, type the name of the table, and then click **Load**.

R script data connector



The highly popular statistical R programming language has been integrated into the Transact-SQL language so that data scientists can develop predictive applications in R and deploy them in a SQL Server production environment. This feature was introduced with SQL Server 2016 and is known as SQL Server R Services. This service enables you to run R scripts in Power BI Desktop and import the results into a Power BI Desktop data model. You create reports by using this data that is then uploaded to the Power BI service.

Installing R

To run R scripts from Power BI Desktop, you must install a local instance of R. For further information about downloading and installing R Services, see *Set up a data science client for R development on SQL Server* in Microsoft Docs:

Set up a data science client for R development on SQL Server

http://aka.ms/r2r8xh

Running R scripts from Power BI

After installing R on your local workstation, you can begin running R scripts to import data and create reports. You must first write and test the scripts in your local development environment, to ensure that the scripts run successfully. There are several limitations that should be observed before you run a script:

- Only data frames are imported, so all of the data that you want should be included in the data frame.
- The time-out for the query is limited to a maximum of 30 minutes. The script stops executing if it has to wait for user input.
- N/A values are converted to NULL values.
- Complex and Vector type columns cannot be imported, and will be replaced with error values in the table.
- When you set the working directory of the R script, you must use a full path, not a relative path.

To run R scripts from Power BI:

1. Open Power BI Desktop, and then on the ribbon menu, click **Get Data**.
2. In the Get Data window, click **Other**, click **R Script**, and then click **Connect**.
3. Type or paste your script into the script box, and then check that the location where the R script is installed is correct—for example, C:\Program Files\Microsoft\MRO\R-3.2.2.
4. If you have multiple installations, select the one that you want or explicitly provide the full location, and then click **OK**.

R script options

You can also manage your R installations in the **Options** dialog box:

1. In Power BI Desktop, click **File**, click **Options and settings**, and then click **Options**.
2. Click **R scripting**, select your R home directory from the list, and then click **OK**. The option that you choose here is then used as the default in the R script data connector.

Demonstration: Importing data from SQL Server

In this demonstration, you will see how to:

- Connect to SQL Server from Power BI Desktop.
- Import data into the Power Query Editor.

Check your knowledge

True or false question

True or false? You can use the Power BI Q&A natural language to ask questions of your data when using DirectQuery.

True False

Check answer Reset

Lesson 4: The Power BI Service

This lesson explores some of the advanced features in the Power BI Service—including how to use natural query language with Power BI Q&A to ask questions of

your data and how to create apps to share collections of dashboards and reports.

Lesson objectives

After completing this lesson, you will be able to:

- Configure your data to use the Q&A feature to ask natural language questions in dashboards.
- Create apps.
- Define workspace roles to share apps with colleagues.

Configuring your data for Q&A

- Power BI Q&A:
  - The Q&A text box appears at the top of every dashboard
  - Enables users to ask questions of their data by using the natural query language
  - Q&A returns answers based on the dataset in the dashboard, using an appropriate or user-specified visual
- Q&A depends on entity names for searches:
  - Can use structured data and uploaded Excel workbooks
  - Upfront data cleaning and optimizations boost the performance of Q&A to deliver better results
  - Tables, columns, and calculated fields should be named appropriately, by using words you would search on

The Q&A box appears at the top of your dashboards, and enables you to ask questions of your data by using natural query language. Q&A recognizes the words in your questions, and works out where in your dataset it can find an answer. Furthermore, Q&A helps you to formulate your question by using autocomplete, restatement, and dimming of words that it does not understand. Q&A displays the

answer as an interactive visualization. Unless you specify the type of visual that you want, Q&A uses the one that it determines is most appropriate. For example, if you asked, “What were last year’s sales by territory?” Q&A would know to use a map. However, you could ask, “What were last year’s sales by salesperson as a pie chart?” so that you specify the exact visual that you want to represent your answer.

For more information about asking questions by using Power BI Q&A, see *Get started with Power BI Q&A* in the Power BI Documentation:

Get started with Power BI Q&A

http://aka.ms/AA4yc31

Q&A searches structured data, and can work on any Excel workbook that you upload. However, upfront data cleaning and optimizations can help to boost the performance of Q&A, and deliver the answers that you need. Consider the names that you give to your entities. For example, if you have a table named **Internet Sales**, columns named **Category**, **Product**, **Units Sold**, **Cost Price**, **Gross Sales**, **Month**, and **Year**, and a calculated column named **Profit**, you see how easy it is to find answers to questions such as, “What were the sales last year by category and month?” Q&A also understands how to filter, sort, aggregate, and group data, which you can include in your question, so you could ask, “What were last year’s sales by month sorted by profit?” By being clear in your naming conventions, you see how Q&A can more easily deliver answers to your questions.

> **Note:** The Power BI Q&A natural language only works with cloud-based datasets that have been uploaded to the service, so you cannot use it with an on-premises tabular model in SQL Server Analysis Services.

Creating apps

- Power BI content packs are packaged reports, dashboards, and datasets:
  - Can be shared with other Power BI users
  - After importing, the contents are merged into the My Workspace lists
  - Packs can be customized for different users
  - Give access to specific groups, or entire organizations
  - Add title, description, and image or company logo
  - Datasets for the selected reports and dashboards cannot be excluded
- Import content packs from SaaS providers such as Bing, MailChimp, Insightly, Marketo, and Twilio

Apps are packaged reports, dashboards, and datasets that can be shared with other Power BI users in your organization. When you install an app on the Power BI portal, the report items are merged into your workspace lists. Users who have a free Power BI account can only view apps that are published to Power BI Premium capacity—they cannot create them. You can create apps to customize reports or dashboards for users in different departments within your organization. For example, you could create a set of reports with targeted visuals for finance, sales, and manufacturing, because each department is likely to want different data with which to measure performance.

When you publish an app, you choose who you want to give access to. You choose specific groups, such as sales or human resources, or you give access to your entire organization. The app can be customized with a title and a description to help users to determine whether the app is applicable to their needs. Furthermore, you can upload an image or company logo for the app. You choose the reports, dashboards, and datasets that you want to include, but when you choose a report or dashboard, it automatically includes any required datasets—these cannot be excluded. The app is then available in your organization's content gallery. Users who have access to the app can create new dashboards from the contents.

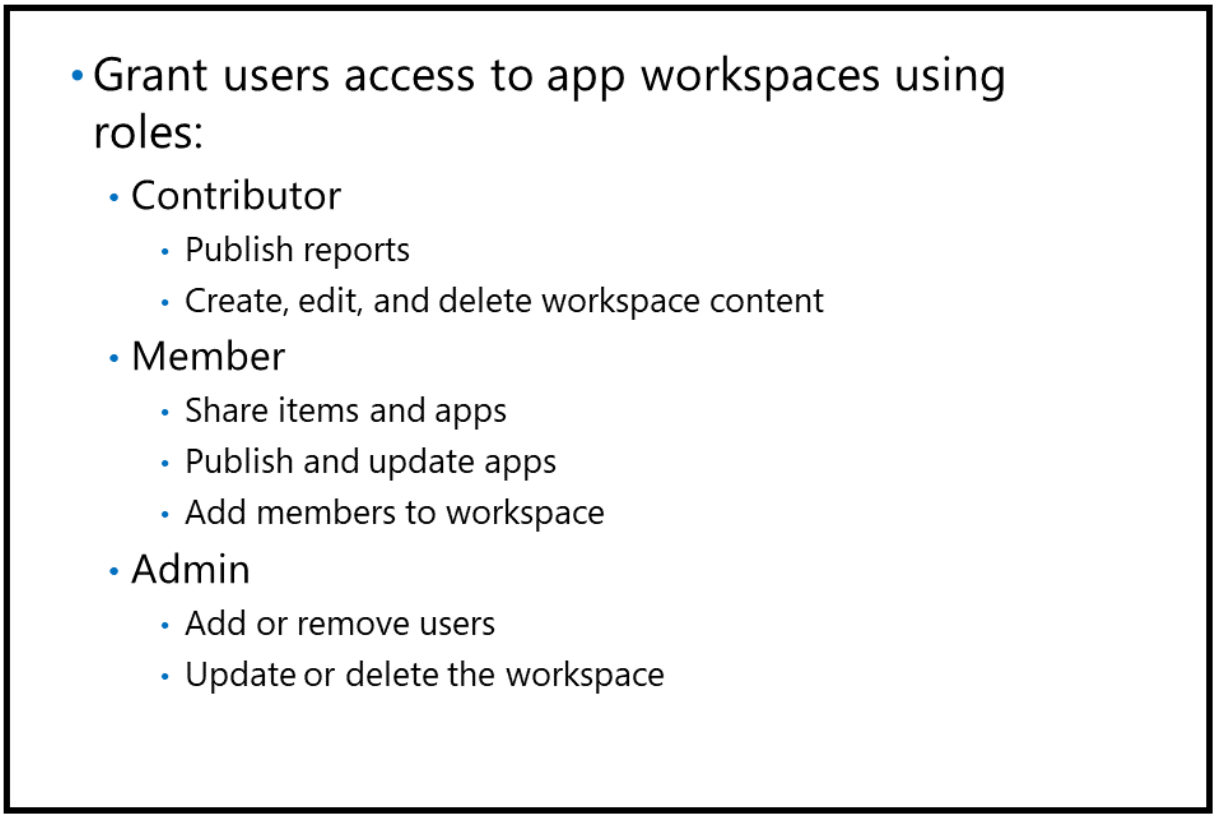
Importing content packs

You can install apps from SaaS providers such as Adobe Analytics, Alpine Metrics Sales Predictions, Insightly, Marketo, and Twilio. To add an app from an SaaS provider with whom you have an account:

1. Sign in to the Power BI service, click **Get Data**, and then under **Services**, click **Get**.
2. Click the provider's SaaS logo, and then click **GET IT NOW**. You will be prompted to enter your login details for the service.
3. After you have been authenticated, you can import an app that contains reports and dashboards that have been designed to visualize your data without you needing to do any work.

> **Note:** Only users who have a Power BI Pro subscription can create and share apps. You do not need a Power BI Pro account to view apps from your organization or from SaaS providers.

Defining workspace roles



When you create an app workspace, you grant access to it by adding user groups or individuals to workspace roles. These roles define which users can view, create, and share information in the app, and which users can administer the app workspace.

Three roles are available: Admin, Member, and Contributor. The information in the following table describes the capabilities assigned to users in each role.

Capability	Contributor	Member	Admin
Publish reports to the workspace	Yes	Yes	Yes
Create, edit, and delete content in the workspace	Yes	Yes	Yes
Share an item or an app		Yes	Yes
Publish or update an app		Yes	Yes
Add members to the workspace		Yes	Yes
Add/remove users, including admins			Yes
Update or delete the workspace			Yes

To add users to the workspace:

1. Sign in to the Power BI service, and then click the arrow next to **Workspaces**.
2. In the expanded pane, click the ellipsis next to your workspace name, and then click **Workspace Access**.
3. In the **Access** pane, in the **Enter email addresses box**, type an email address or security group name, in the dropdown list select **Member**, **Contributor**, or **Admin**, and then click **Add**.
4. When you have finished adding all the required users, click **Close**.

Demonstration: Querying data by using Q&A

In this demonstration, you will see how to:

- Ask a question by using Q&A.
- Pin the answer to a question to an existing dashboard.
- Ask a question and specify the visual to represent the data.

Check your knowledge

Discovery

How could your organization use apps? What are the major advantages of apps?

Show solution Reset

Lab: Importing data into Power BI

Scenario

Adventure Works employees are increasingly frustrated by the time that it takes to implement managed BI services. The existing managed BI infrastructure, including a data warehouse, enterprise data models, and reports and dashboards, are valued

sources of decision-making information. However, users increasingly want to explore relationships with other, currently unmanaged data, and it takes too long for the IT department to incorporate these requirements into the corporate BI solution.

As a BI professional, you have been asked to explore ways in which Adventure Works can empower business users to augment their managed enterprise BI solution with self-service BI.

Objectives

After completing this lab, you will be able to:

- Alter an Excel file to reduce its size, and then import the data into Power BI Desktop.
- View existing Excel Power View worksheets as reports in Power BI.

> **Note:** Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual − but you can access them on GitHub.

Lab setup

Estimated time: 60 minutes

Virtual machine: **20778C-MIA-CLI**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_03.md.

Exercise 1: Importing Excel files into Power BI

Scenario

As a data analyst for Adventure Works, you will be using Power BI to create reports that the business analysts can use to create dashboards in the Power BI service. One of the business analysts has asked you to import an Excel file as the basis for a report. The file contains formatting that needs to be removed before you can import it. You will remove the formatting, and then import the data in the workbook to create a new dataset. As part of this exercise, you will alter the column names so that they are more suitable for Q&A to find answers within the dataset.

The main tasks for this exercise are as follows:

1. Prepare the lab environment.
2. Reduce the size of Excel files.
3. Import Excel files.

Result: After this exercise, the data in Excel will be available as a dataset in Power BI.

Exercise 2: Viewing reports from Excel files

Scenario

A business analyst has emailed to you an Excel workbook that contains a Power View report. The analyst wants you to upload the file to Power BI so that the sales department can reuse the work that has already been done on creating the interactive visuals. You will sign in to Power BI and upload the report.

The main task for this exercise is as follows:

- View Excel Power View sheets as Power BI reports.

Result: At the end of this exercise, the Power View report will be available as a Power BI report.

Review question(s)

Check your knowledge

Discovery

Discuss the different data sources that your organization could use to create Power BI reports. Can you think of a scenario where users perhaps have Excel workbooks for one set of reports, and reports in SQL Server Reporting Services for another set of data? Could this be combined into a single dataset in Power BI?

Show solution Reset

Module review and takeaways

In this module, you have learned how to use Power BI to enable users to easily access data and create reports. You have seen how to publish data from Excel and from SQL Server and other types of database. In addition, you have seen how to use Q&A to ask questions in natural query language and how to share your reports with colleagues.

Review question(s)

Check your knowledge

Discovery

Discuss the different ways in which Power BI could reduce your organization's dependency on shared Excel files. How would having a central location for data, reports, and dashboards benefit different departments? How could each department make use of features such as apps and the natural query language in Q&A?

Show solution Reset

Este documento pertenece a Monica Rodriguez.
monica.rodriguez@annardx.com
No están permitidas las copias sin autorización.

Este documento pertenece a Monica Rodriguez.
monica.rodriguez@annardx.com
No están permitidas las copias sin autorización.