**Power BI**

The Power BI development cycle is divided into four parts: data discovery, data modeling, data visualization, and distribution of reports. Each stage requires its own skill set.

We cover data modeling and visualization skills in Chapter 2, “Modeling and visualizing data,” and report distribution in Chapter 3, “Configure dashboards, reports, and apps in the Power BI Service.” In this chapter, we review the skills you need to consume data in Power BI Desktop. Power BI has a rich set of features available for data shaping, which enables the creation of sophisticated data models. We start with the steps required to connect to various data sources. We then review the basic and advanced transformations available in Power BI Desktop, as well as ways to combine data from distinct data sources. Finally, we review some data cleansing techniques.

**Connect to data sources**

Before you model or visualize any data, you need to prepare and load it into Power BI, creat­ing one or more connections to data sources. Power BI can connect to a wide variety of data sources, and the number of supported data sources grows every month. Furthermore, Power BI allows you to create your own connectors, making it possible to connect to virtually any data source.

**Connect to databases, files, and folders**

Databases, files, and folders are some of the most common data sources used when connect­ing to data in Power BI. Power BI can connect to the following databases:

* SQL Server database
* Access database
* SQL Server Analysis Services database
* Oracle database
* IBM DB2 database
* IBM Informix database (Beta)
* IBM Netezza (Beta)
* MySQL database
* PostgreSQL database
* Sybase database
* Teradata database
* SAP HANA database
* SAP Business Warehouse database
* Amazon Redshift
* Impala
* Snowflake
* ODBC
* OLE DB

Power BI can also connect to the following file types:

* Excel
* Text/CSV
* XML
* JSON

Files can also be connected to in bulk mode through the following folder connectors:

* Folder
* SharePoint folder
* Azure Blob Storage
* Azure Data Lake Store

To connect to a data source, you need to click the Home tab and select Get Data in the Ex­ternal Data group. Clicking the text portion of the button opens a drop-down list with the most common data sources. When you click More in the drop-down list, the full Get Data window opens.



**Data connectivity modes**

The most common way to consume data in Power BI is by importing it to the data model. When you import data in Power BI, you create a copy of it that is kept static until you refresh your dataset. Currently, data from files and folders can only be imported in Power BI. When it comes to databases, there are two ways in which you can make data connections.



First, you can import your data into Power BI, which copies data into the Power BI data model. This method offers you the greatest flexibility when you model your data because you can use all available features in Power BI.

Second, you can connect to your data directly in its original source. This method is known as DirectQuery. With DirectQuery, data is not kept in Power BI. Instead, the original data source is queried every time you interact with Power BI visuals. Not all data sources support DirectQuery.

**Importing data**

When you import data, you load a copy of it into Power BI. Because Power BI is based on an in-memory engine called VertiPaq (also known as xVelocity), the imported data consumes both the RAM and disk space, because data is stored in files. During the development phase, the imported data consumes the disk space and RAM of your development machine. Once you publish your report to a server, the imported data consumes the disk space and RAM of the server to which you publish your report. The implication of this is that you can’t load more data into Power BI than your hardware allows.

You have an option to transform data when you import it in Power BI, limited only by the functionality of Power BI. If you only load a subset of tables from your database, and you apply filters to some of the tables, only the filtered data gets loaded into Power BI.

Once data is loaded into the Power BI cache, it is kept in a compressed state, thanks to the VertiPaq engine. The compression depends on many factors, including data type, values, and cardinality of the columns. In most cases, however, data will take much less space once it is loaded into Power BI compared to its original size.

One of the advantages of this data connection method is that you can use all of the functionality of Power BI without restrictions, including all transformations available in Power Query Editor, as well as all DAX functions when you model your data.

Additionally, you can use data from more than one source in the same data model. For example, you can load some data from a database and some data from an Excel file. You can then either combine them in the same table in Power Query Editor or relate the tables in the data model.

Another advantage of this method is the speed of calculations. Because the VertiPaq engine stores data in-memory in a compressed state, there is little to no latency when accessing the data. Additionally, the engine is optimized for calculations, resulting in the best computing speed.

**DirectQuery**

When you use the DirectQuery method, you are not loading any data into Power BI. All the data remains in the data source, except for metadata, which Power BI keeps. Metadata includes column and table names, data types, and relationships. For most data sources supporting DirectQuery, when connecting to a data source, you select the structures you want to con­nect to, such as tables or views. Each structure becomes a table in your data model. With some sources, such as SAP Business Warehouse, you only select a database, not specific tables or other structures.

With this method, Power BI only serves as a visualization tool. As a result, the Power BI file size will be negligible compared to a file with imported data.

At the time of this writing, only the following databases support DirectQuery connectivity.

* Amazon Redshift
* Azure HDInsight Spark (Beta)
* Azure SQL Database
* Azure SQL Data Warehouse
* Google BigQuery (Beta)
* IBM Netezza (Beta)
* Impala (version 2.x)
* Oracle Database (versions 12 and above)
* SAP Business Warehouse (Beta)
* SAP HANA
* Snowflake
* Spark (Beta) (versions 0.9 and above)
* SQL Server
* Teradata Database
* Vertica (Beta)

The main advantage of this method is that you are not limited by the hardware of your de­velopment machine or of the server to which you will publish your report. All data is kept in the data source, and all the calculations are done in the source as well. Using DirectQuery entails some implications to the available functionality.