

Module 7: Direct Connectivity

- [Module 7: Direct Connectivity](#)
 - [Lesson 1: Cloud data](#)
 - [Demo 1: Using Azure SQL Database as a Power BI data source](#)
 - [Import data from tables in a database in Azure SQL Database](#)
 - [View relationships between the tables](#)

Lesson 1: Cloud data

Demo 1: Using Azure SQL Database as a Power BI data source

Connect to data from tables in a database in Azure SQL Database

1. Ensure that you have copied all folders from `Desktop/power-bi-quickstart` folder into `*D:*` drive before starting the lab.
2. On the Taskbar, click **Power BI Desktop**.
3. To close the getting started window, at the top-right of the window, click **X**.
4. In the **Power BI Desktop** window, click **Get data**.
5. In the **Get Data** dialog box, click **SQL Server**, and then click **Connect**.
6. In the **SQL Server database** window, in the **Server** box, type the URL of the Azure server **localhost**.
7. In the **Database (optional)** box, type **AdventureWorksLT**.
8. In the **Data Connectivity mode**, select **DirectQuery**, and then click **OK**.
9. In the **SQL Server database** dialog box, on the **Database** tab, in the **User name** box, type **Student**.
10. In the **Password** box, type **Pa55w.rd**, and then click **Connect**.
11. In the **Navigators** dialog box, select the **SalesLT.Customer**, **SalesLT.Product**, **SalesLT.ProductCategory**, **SalesLT.SalesOrderDetail**, and **SalesLT.SalesOrderHeader** check boxes, and then click **Load**.
12. In the **FIELDS** pane, notice that the five tables have been added. When the report is published to the Power BI service, the tables are combined into a single dataset.
13. The connection is now a DirectQuery connection. The data has not been imported and you are always looking at current data.

Use a Composite model

DirectQuery storage is ideal for some of your data, but some data rarely changes and you would achieve performance gains from caching that data locally. Initially, you will set the Customer and Product tables to Dual storage mode. This will improve performance, by using cached data, but will use Direct Query, when necessary, for example, when pushing join logic to the source system.

1. In the **FIELDS** pane, right-click **SalesLT Customer**, and click **Properties**.
2. In **FIELD PROPERTIES**, in **Storage mode**, select **Dual**.
3. Read the warning message and click **OK**.
4. In **FIELDS**, select **SalesLT Product**.
5. In **FIELD PROPERTIES**, in **Storage mode**, select **Dual**.

6. Read the warning message and click **OK**.
7. In **FIELDS**, select **SalesLT ProductCategory**.
8. In **FIELD PROPERTIES**, in **Storage mode**, select **Import**.
9. Read the warning message and discuss the implications. Note that the relationships are weak when the one-side of a relationship cannot be guaranteed to be unique. This issue can be resolved by using Dual storage mode. In this situation, you are happy that the data and relationships are correct and wish to continue with Import Storage mode.
10. Unselect **Set affected tables to dual**.
11. Click **OK**.
12. Note that the **Storage mode** for **SalesLT SalesOrderDetail** and **SalesLT SalesOrderHeader** remains as **DirectQuery**. This will give the most up-to-date data for frequently updated tables.

View relationships between the tables

1. In the menu on the left, click **Model**, and then increase the size of the **SalesLT SalesOrderDetail**, **SalesLT SalesOrderHeader**, and **SalesLT Customer** tables to display all columns.
2. Position the cursor on the relationship arrow between **SalesLT SalesOrderDetail** and **SalesLT SalesOrderHeader**. Notice that the related columns are highlighted.
3. Position the cursor on the relationship arrow between **SalesLT SalesOrderHeader** and **SalesLT Customer**. Point out that the related columns are highlighted.
4. Notice that the tables are still related irrespective of the Storage mode.

Create Visualizations

1. In the menu on the left, click **Report** to return to the report canvas.
2. In the **FIELDS** pane, expand **SalesLT Customer**, and drag the **CompanyName** field onto the canvas to create a table.
3. In the **FIELDS** pane, expand **SalesLT SalesOrderDetail**, and drag the **LineTotal** field to the table on the report.
4. In the **VISUALIZATIONS** pane, click **Stacked column chart**.
5. Drag the right edge of the chart to stretch it across the report and display the customers in full.
6. In the **VISUALIZATIONS** pane, click **Format**, expand **Title**, and then rename the chart **Line Total by Company Name**.
7. In the **FIELDS** pane, in **SalesLT ProductCategory**, drag the **Name** field onto the canvas to create a table below the chart.
8. In the **FIELDS** pane, in **SalesLT SalesOrderDetail**, drag the **OrderQty** field to the table on the report.
9. In the **VISUALIZATIONS** pane, click **Stacked column chart**.
10. Drag the right edge of the chart to stretch it across the report and display the customers in full.
11. In the **VISUALIZATIONS** pane, click **Format**, expand **Title**, and then rename the chart **Order Quantity by Product Category**.
12. Expand **Data colors**, and then select a different color from the **Default color** selector.
13. Click on the canvas.

14. In the **FIELDS** pane, in **SalesLT Customer**, drag the **CompanyName** field onto the **Page level filters** in the **VISUALIZATIONS** pane.
15. Close Power BI desktop without saving your changes.