Create a Lakehouse

Large-scale data analytics solutions have traditionally been built around a *data warehouse*, in which data is stored in relational tables and queried using SQL. The growth in “big data” (characterized by high *volumes*, *variety*, and *velocity* of new data assets) together with the availability of low-cost storage and cloud-scale distributed compute technologies has led to an alternative approach to analytical data storage; the *data lake*. In a data lake, data is stored as files without imposing a fixed schema for storage. Increasingly, data engineers and analysts seek to benefit from the best features of both of these approaches by combining them in a *data lakehouse*; in which data is stored in files in a data lake and a relational schema is applied to them as a metadata layer so that they can be queried using traditional SQL semantics.

In Microsoft Fabric, a lakehouse provides highly scalable file storage in a *OneLake* store (built on Azure Data Lake Store Gen2) with a metastore for relational objects such as tables and views based on the open source *Delta Lake* table format. Delta Lake enables you to define a schema of tables in your lakehouse that you can query using SQL.

This lab takes approximately **30** minutes to complete.

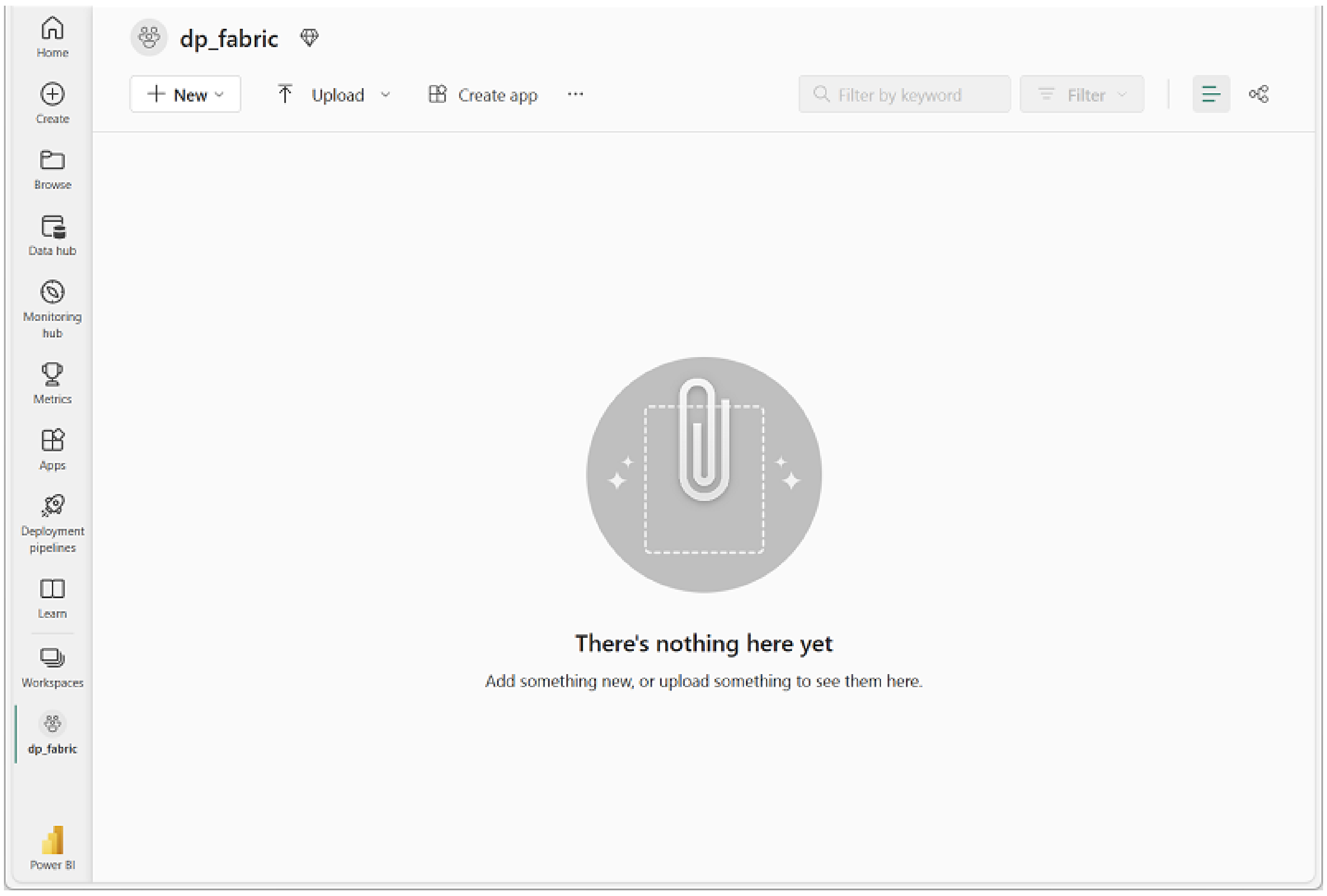
> ! **Note**: You need a Microsoft *school* or *work* account to complete this exercise. If you don’t have one, you can sign up for a trial of Microsoft Office 365 E3 or higher.

Create a workspace

Before working with data in Fabric, create a workspace with the Fabric trial enabled.

1. On the Microsoft Fabric home page, select **Synapse Data Engineering**.
2. In the menu bar on the left, select **Workspaces** (the icon looks similar to 🗇).
3. Create a new workspace with a name of your choice, selecting a licensing mode that includes Fabric capacity (*Trial*, *Premium*, or *Fabric*).

4. When your new workspace opens, it should be empty.



There's nothing here yet

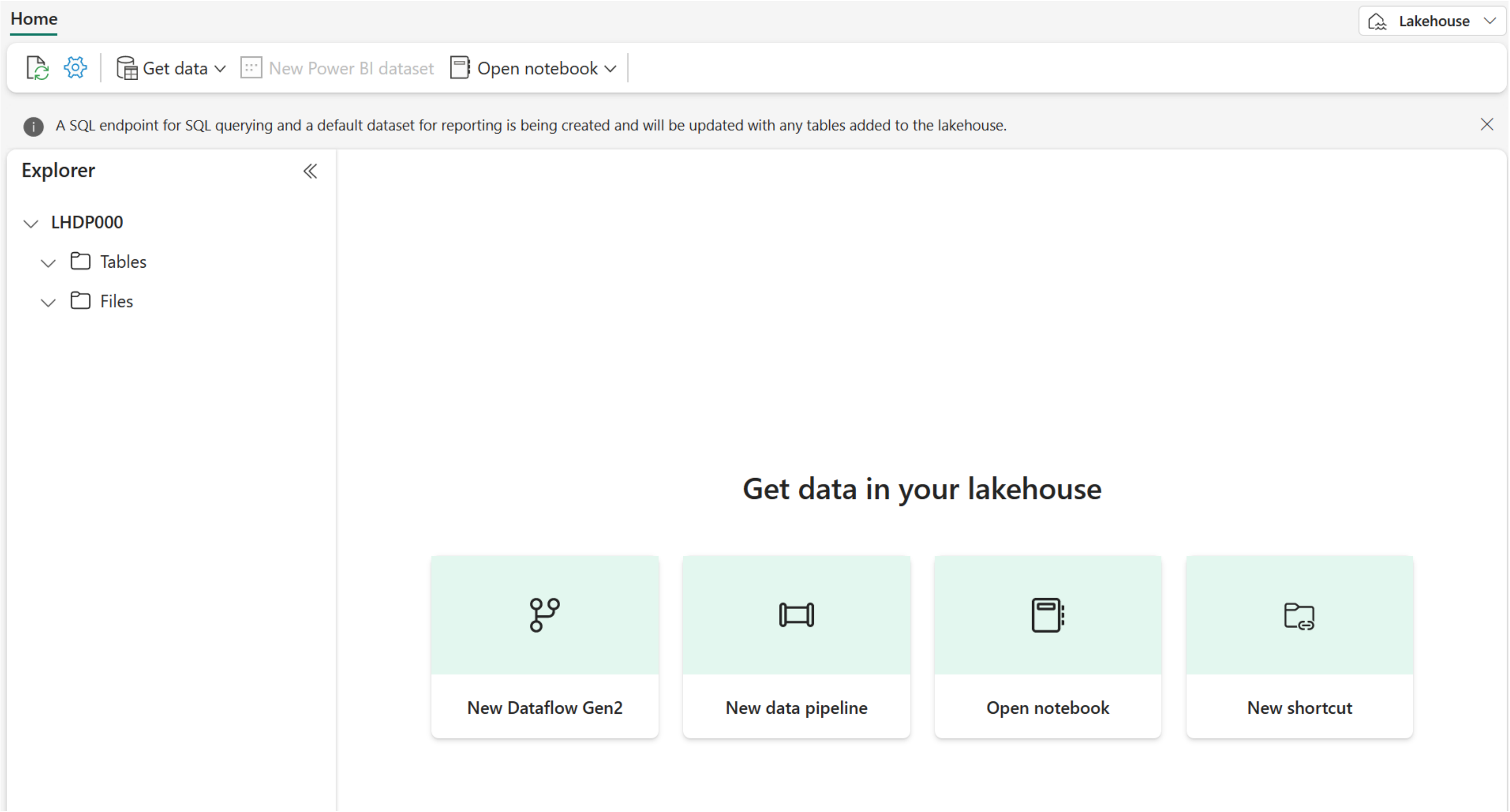
Add something new, or upload something to see them here.

Create a lakehouse

Now that you have a workspace, it’s time to create a data lakehouse for your data files.

1. In the **Synapse Data Engineering** home page, create a new **Lakehouse** with a name of your choice.

   After a minute or so, a new lakehouse will be created:

   A SQL endpoint for SQL querying and a default dataset for reporting is being created and will be updated with any tables added to the lakehouse.

2. View the new lakehouse, and note that the **Lakehouse explorer** pane on the left enables you to browse tables and files in the lakehouse:

   - The **Tables** folder contains tables that you can query using SQL semantics. Tables in a Microsoft Fabric lakehouse are based on the open source *Delta Lake* file format, commonly used in Apache Spark.
   - The **Files** folder contains data files in the OneLake storage for the lakehouse that aren't associated with managed delta tables. You can also create *shortcuts* in this folder to reference data that is stored externally.

   Currently, there are no tables or files in the lakehouse.

Upload a file

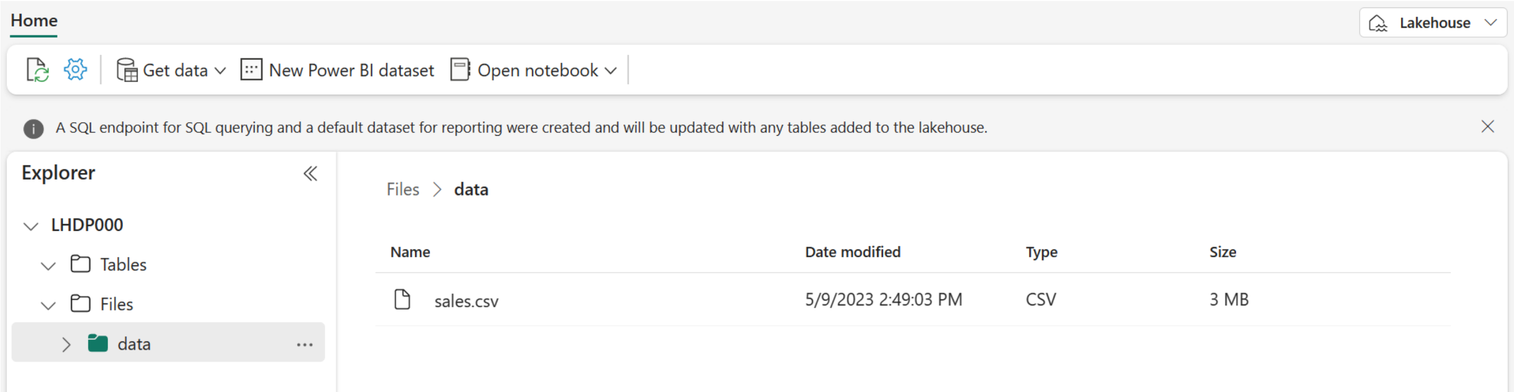
Fabric provides multiple ways to load data into the lakehouse, including built-in support for pipelines that copy data external sources and data flows (Gen 2) that you can define using visual tools based on Power Query. However one of the simplest ways to ingest small amounts of data is to upload files or folders from your local computer (or lab VM if applicable).

1. Download the **sales.csv** file from https://raw.githubusercontent.com/MicrosoftLearning/dp-data/main/sales.csv, saving it as **sales.csv** on your local computer (or lab VM if applicable).

   > ❕ **Note**: To download the file, open a new tab in the browser and paste in the URL. Right click anywhere on the page containing the data and select **Save as** to save the page as a CSV file.

2. Return to the web browser tab containing your lakehouse, and in the **...** menu for the **Files** folder in the **Lakehouse explorer** pane, select **New subfolder**, and create a subfolder named **data**.
3. In the **...** menu for the new **data** folder, select **Upload** and **Upload file**, and then upload the **sales.csv** file from your local computer (or lab VM if applicable).

4. After the file has been uploaded, select the **Files/data** folder and verify that the **sales.csv** file has been uploaded, as shown here:



A SQL endpoint for SQL querying and a default dataset for reporting were created and will be updated with any tables added to the lakehouse.

Name	Date modified	Type	Size
sales.csv	5/9/2023 2:49:03 PM	CSV	3 MB

5. Select the **sales.csv** file to see a preview of its contents.

Explore shortcuts

In many scenarios, the data you need to work with in your lakehouse may be stored in some other location. While there are many ways to ingest data into the OneLake storage for your lakehouse, another option is to instead create a *shortcut*. Shortcuts enable you to include externally sourced data in your analytics solution without the overhead and risk of data inconsistency associated with copying it.

1. In the **...** menu for the **Files** folder, select **New shortcut**.
2. View the available data source types for shortcuts. Then close the **New shortcut** dialog box without creating a shortcut.

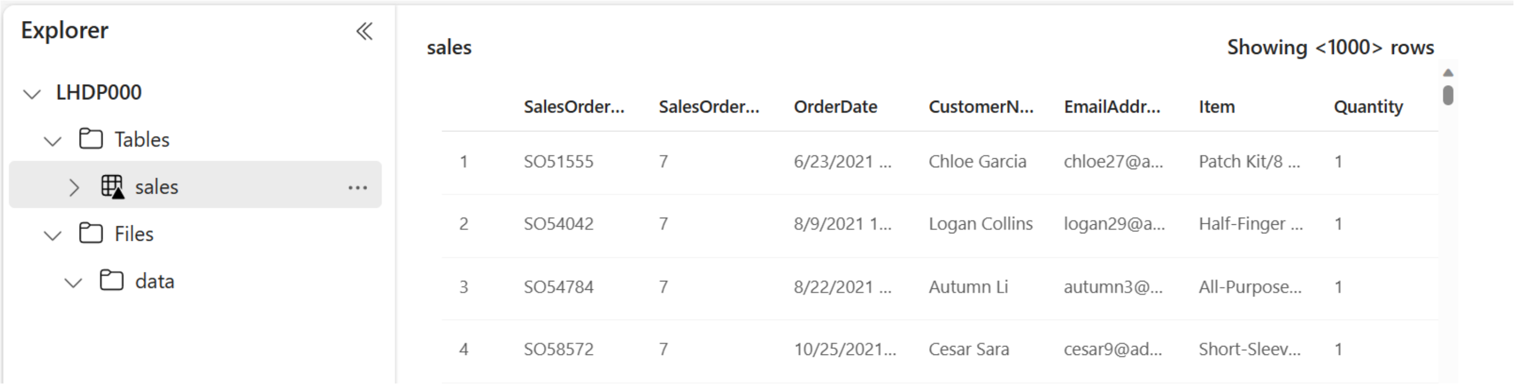
Load file data into a table

The sales data you uploaded is in a file, which data analysts and engineers can work with directly by using Apache Spark code. However, in many scenarios you may want to load the data from the file into a table so that you can query it using SQL.

1. On the **Home** page, select the **Files/Data** folder so you can see the **sales.csv** file it contains.
2. In the **...** menu for the **sales.csv** file, select **Load to Tables**.
3. In **Load to table** dialog box, set the table name to **sales** and confirm the load operation. Then wait for the table to be created and loaded.

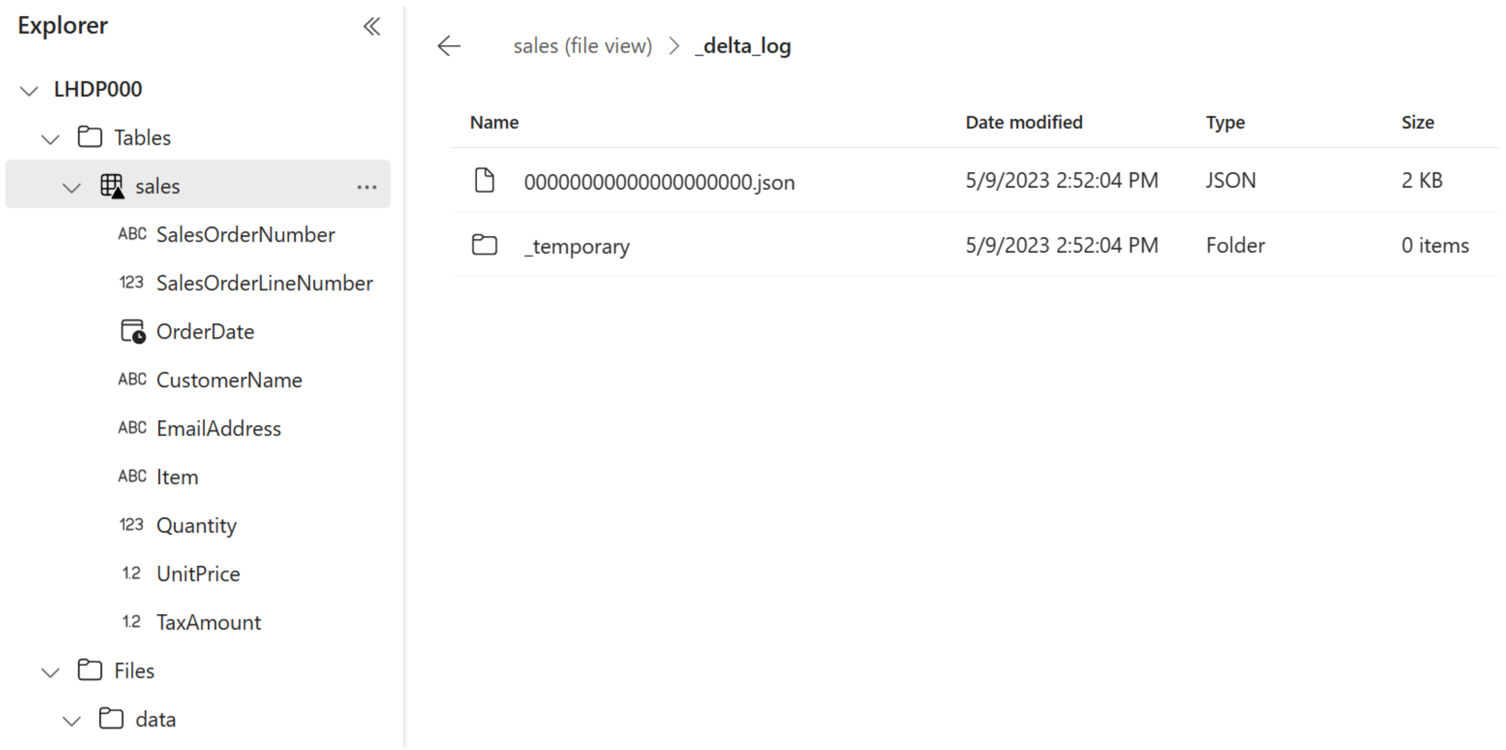
> ❕ **Tip**: If the **sales** table does not automatically appear, in the **...** menu for the **Tables** folder, select **Refresh**.

4. In the **Lakehouse explorer** pane, select the **sales** table that has been created to view the data.



	SalesOrder...	SalesOrder...	OrderDate	CustomerN...	EmailAddr...	Item	Quantity
1	SO51555	7	6/23/2021 ...	Chloe Garcia	chloe27@a...	Patch Kit/8 ...	1
2	SO54042	7	8/9/2021 1...	Logan Collins	logan29@a...	Half-Finger ...	1
3	SO54784	7	8/22/2021 ...	Autumn Li	autumn3@...	All-Purpose...	1
4	SO58572	7	10/25/2021...	Cesar Sara	cesar9@ad...	Short-Sleev...	1

5. In the **...** menu for the **sales** table, select **View files** to see the underlying files for this table



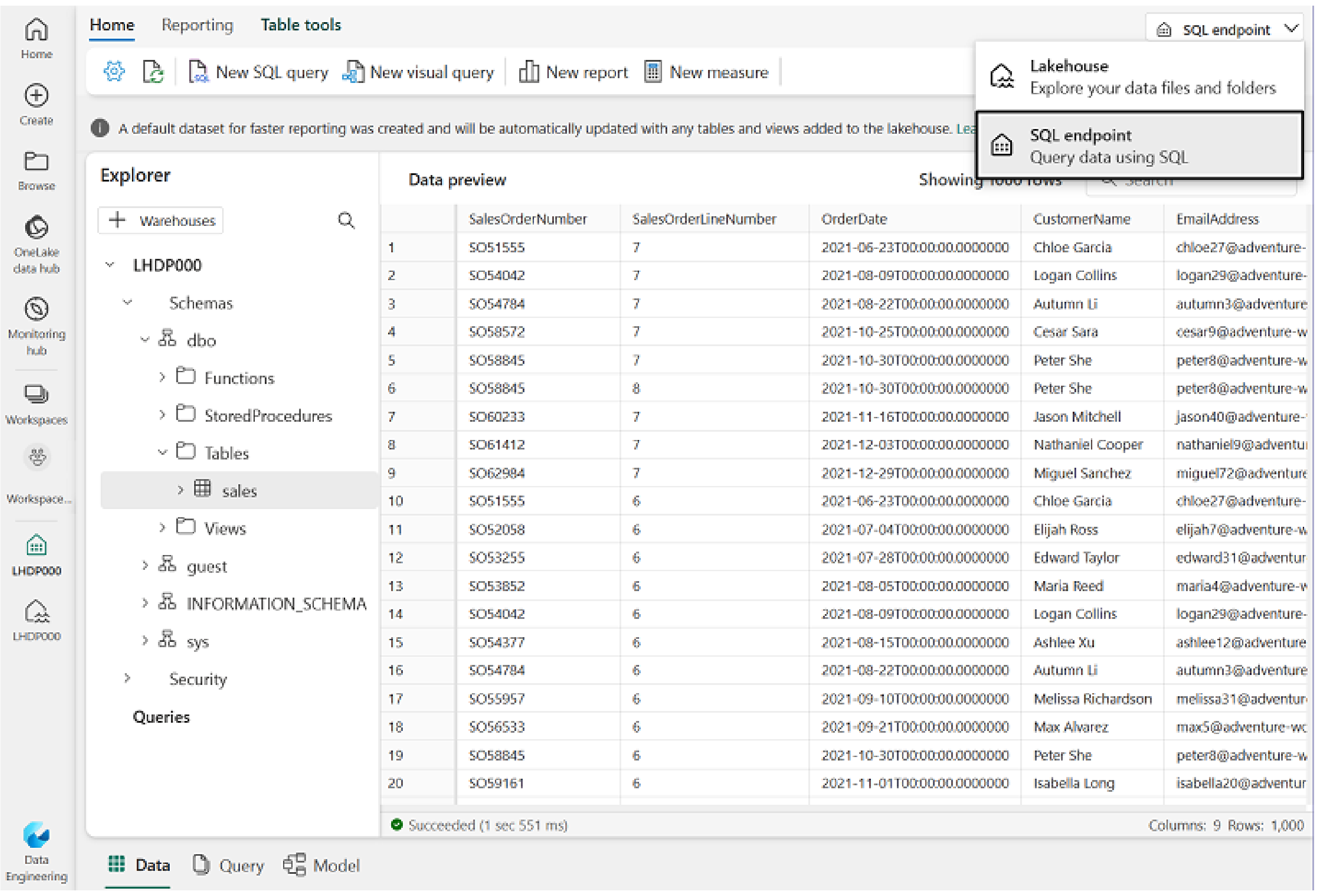
Name	Date modified	Type	Size
00000000000000000000.json	5/9/2023 2:52:04 PM	JSON	2 KB
_temporary	5/9/2023 2:52:04 PM	Folder	0 items

Files for a delta table are stored in *Parquet* format, and include a subfolder named **_delta_log** in which details of transactions applied to the table are logged.

Use SQL to query tables

When you create a lakehouse and define tables in it, a SQL endpoint is automatically created through which the tables can be queried using SQL `SELECT` statements.

1. At the top-right of the Lakehouse page, switch from **Lakehouse** to **SQL endpoint**. Then wait a short time until the SQL query endpoint for your lakehouse opens in a visual interface from which you can query its tables, as shown here:



A default dataset for faster reporting was created and will be automatically updated with any tables and views added to the lakehouse. Lea

	SalesOrderNumber	SalesOrderLineNumber	OrderDate	CustomerName	EmailAddress
1	SO51555	7	2021-06-23T00:00:00.0000000	Chloe Garcia	chloe27@adventure-
2	SO54042	7	2021-08-09T00:00:00.0000000	Logan Collins	logan29@adventure-
3	SO54784	7	2021-08-22T00:00:00.0000000	Autumn Li	autumn3@adventure
4	SO58572	7	2021-10-25T00:00:00.0000000	Cesar Sara	cesar9@adventure-w
5	SO58845	7	2021-10-30T00:00:00.0000000	Peter She	peter8@adventure-w
6	SO58845	8	2021-10-30T00:00:00.0000000	Peter She	peter8@adventure-w
7	SO60233	7	2021-11-16T00:00:00.0000000	Jason Mitchell	jason40@adventure-
8	SO61412	7	2021-12-03T00:00:00.0000000	Nathaniel Cooper	nathaniel9@adventu
9	SO62984	7	2021-12-29T00:00:00.0000000	Miguel Sanchez	miguel72@adventure
10	SO51555	6	2021-06-23T00:00:00.0000000	Chloe Garcia	chloe27@adventure-
11	SO52058	6	2021-07-04T00:00:00.0000000	Elijah Ross	elijah7@adventure-w
12	SO53255	6	2021-07-28T00:00:00.0000000	Edward Taylor	edward31@adventur
13	SO53852	6	2021-08-05T00:00:00.0000000	Maria Reed	maria4@adventure-w
14	SO54042	6	2021-08-09T00:00:00.0000000	Logan Collins	logan29@adventure-
15	SO54377	6	2021-08-15T00:00:00.0000000	Ashlee Xu	ashlee12@adventure
16	SO54784	6	2021-08-22T00:00:00.0000000	Autumn Li	autumn3@adventure
17	SO55957	6	2021-09-10T00:00:00.0000000	Melissa Richardson	melissa31@adventur
18	SO56533	6	2021-09-21T00:00:00.0000000	Max Alvarez	max5@adventure-wc
19	SO58845	6	2021-10-30T00:00:00.0000000	Peter She	peter8@adventure-w
20	SO59161	6	2021-11-01T00:00:00.0000000	Isabella Long	isabella20@adventur

Succeeded (1 sec 551 ms)

Columns: 9 Rows: 1,000

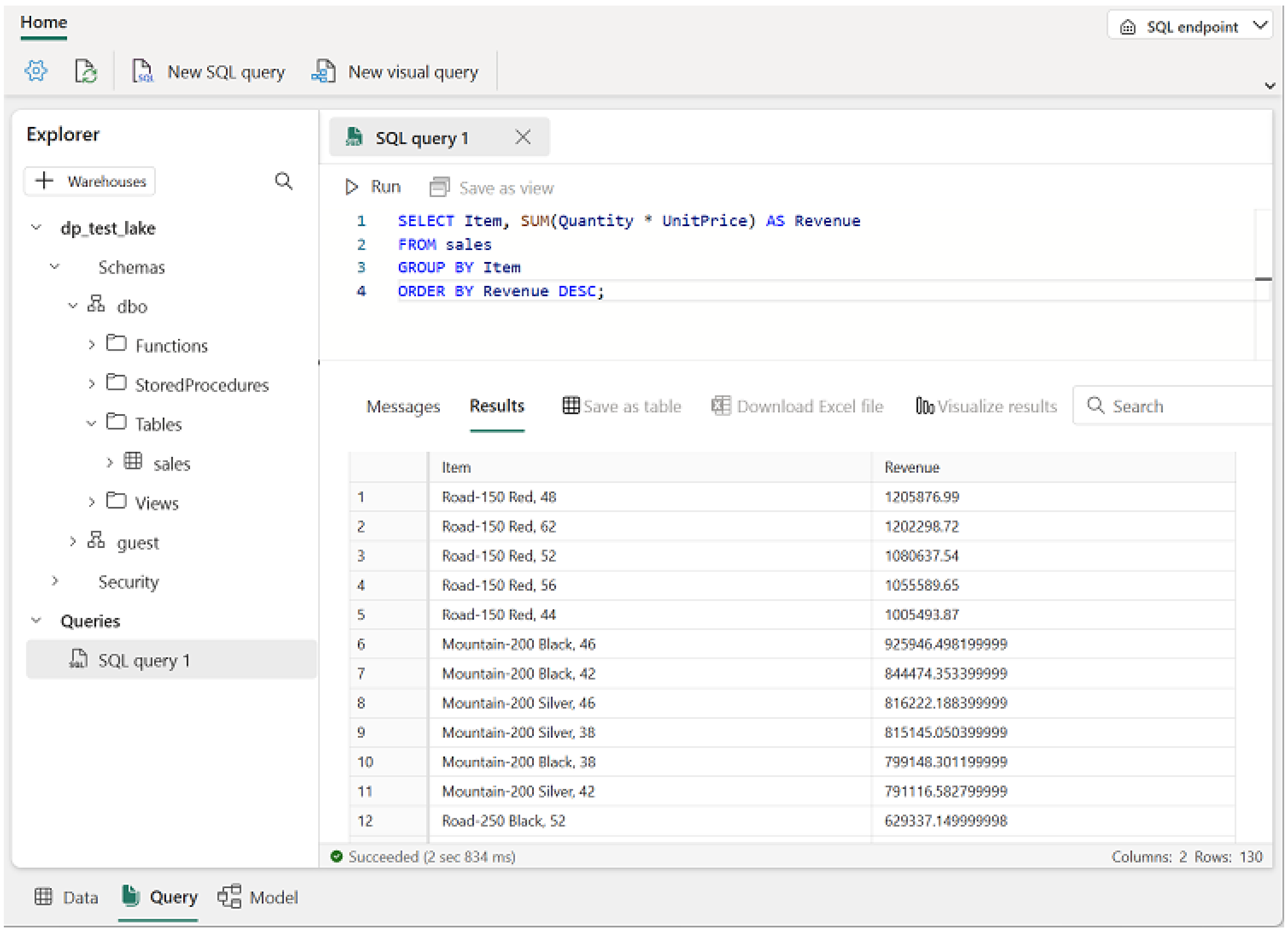
2. Use the **New SQL query** button to open a new query editor, and enter the following SQL query:

Sql

Copy

```sql
SELECT Item, SUM(Quantity * UnitPrice) AS Revenue
FROM sales
GROUP BY Item
ORDER BY Revenue DESC;
```

3. Use the ▷ **Run** button to run the query and view the results, which should show the total revenue for each product.



```sql
SELECT Item, SUM(Quantity * UnitPrice) AS Revenue
FROM sales
GROUP BY Item
ORDER BY Revenue DESC;
```

	Item	Revenue
1	Road-150 Red, 48	1205876.99
2	Road-150 Red, 62	1202298.72
3	Road-150 Red, 52	1080637.54
4	Road-150 Red, 56	1055589.65
5	Road-150 Red, 44	1005493.87
6	Mountain-200 Black, 46	925946.498199999
7	Mountain-200 Black, 42	844474.353399999
8	Mountain-200 Silver, 46	816222.188399999
9	Mountain-200 Silver, 38	815145.050399999
10	Mountain-200 Black, 38	799148.301199999
11	Mountain-200 Silver, 42	791116.582799999
12	Road-250 Black, 52	629337.149999998

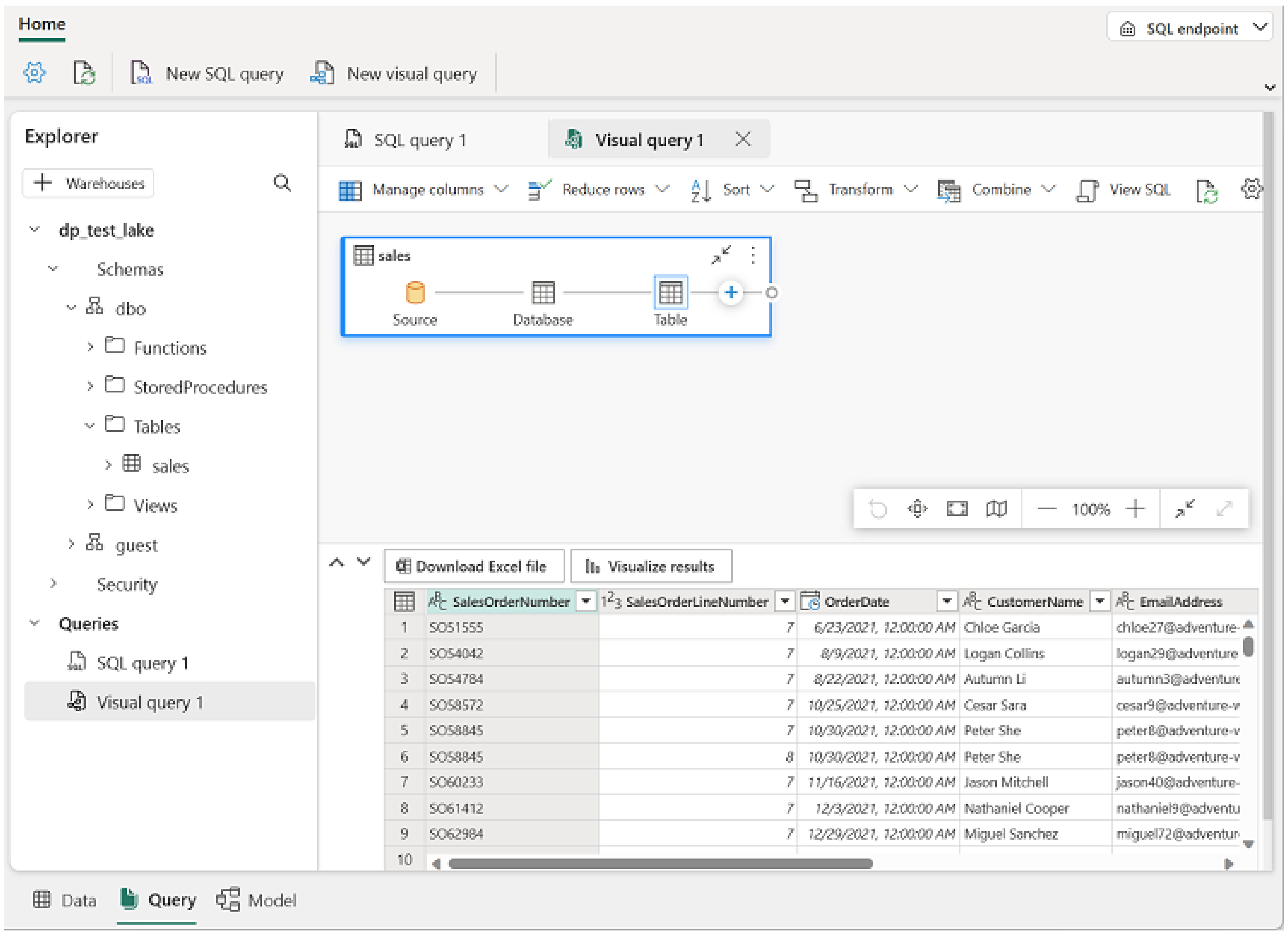
Succeeded (2 sec 834 ms)   Columns: 2 Rows: 130

Create a visual query

While many data professionals are familiar with SQL, data analysts with Power BI experience can apply their Power Query skills to create visual queries.

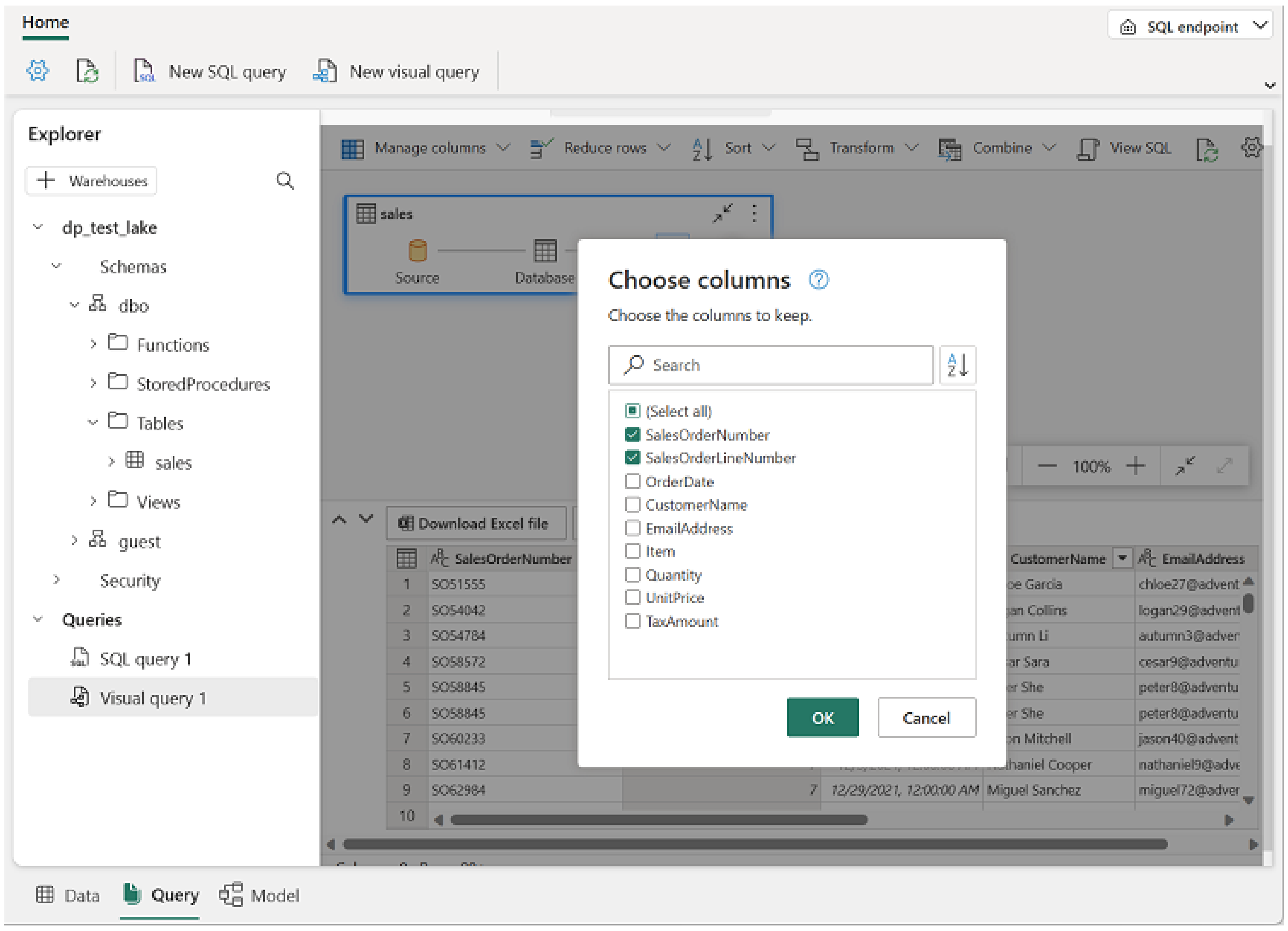
1. On the toolbar, select **New visual query**.

2. Drag the **sales** table to the new visual query editor pane that opens to create a Power Query as shown here:



	SalesOrderNumber	SalesOrderLineNumber	OrderDate	CustomerName	EmailAddress
1	SO51555	7	6/23/2021, 12:00:00 AM	Chloe Garcia	chloe27@adventure-
2	SO54042	7	8/9/2021, 12:00:00 AM	Logan Collins	logan29@adventure
3	SO54784	7	8/22/2021, 12:00:00 AM	Autumn Li	autumn3@adventure
4	SO58572	7	10/25/2021, 12:00:00 AM	Cesar Sara	cesar9@adventure-v
5	SO58845	7	10/30/2021, 12:00:00 AM	Peter She	peter8@adventure-v
6	SO58845	8	10/30/2021, 12:00:00 AM	Peter She	peter8@adventure-v
7	SO60233	7	11/16/2021, 12:00:00 AM	Jason Mitchell	jason40@adventure-
8	SO61412	7	12/3/2021, 12:00:00 AM	Nathaniel Cooper	nathaniel9@adventu
9	SO62984	7	12/29/2021, 12:00:00 AM	Miguel Sanchez	miguel72@adventur
10					

3. In the **Manage columns** menu, select **Choose columns**. Then select only the **SalesOrderNumber** and **SalesOrderLineNumber** columns.



Choose columns

Choose the columns to keep.

Search

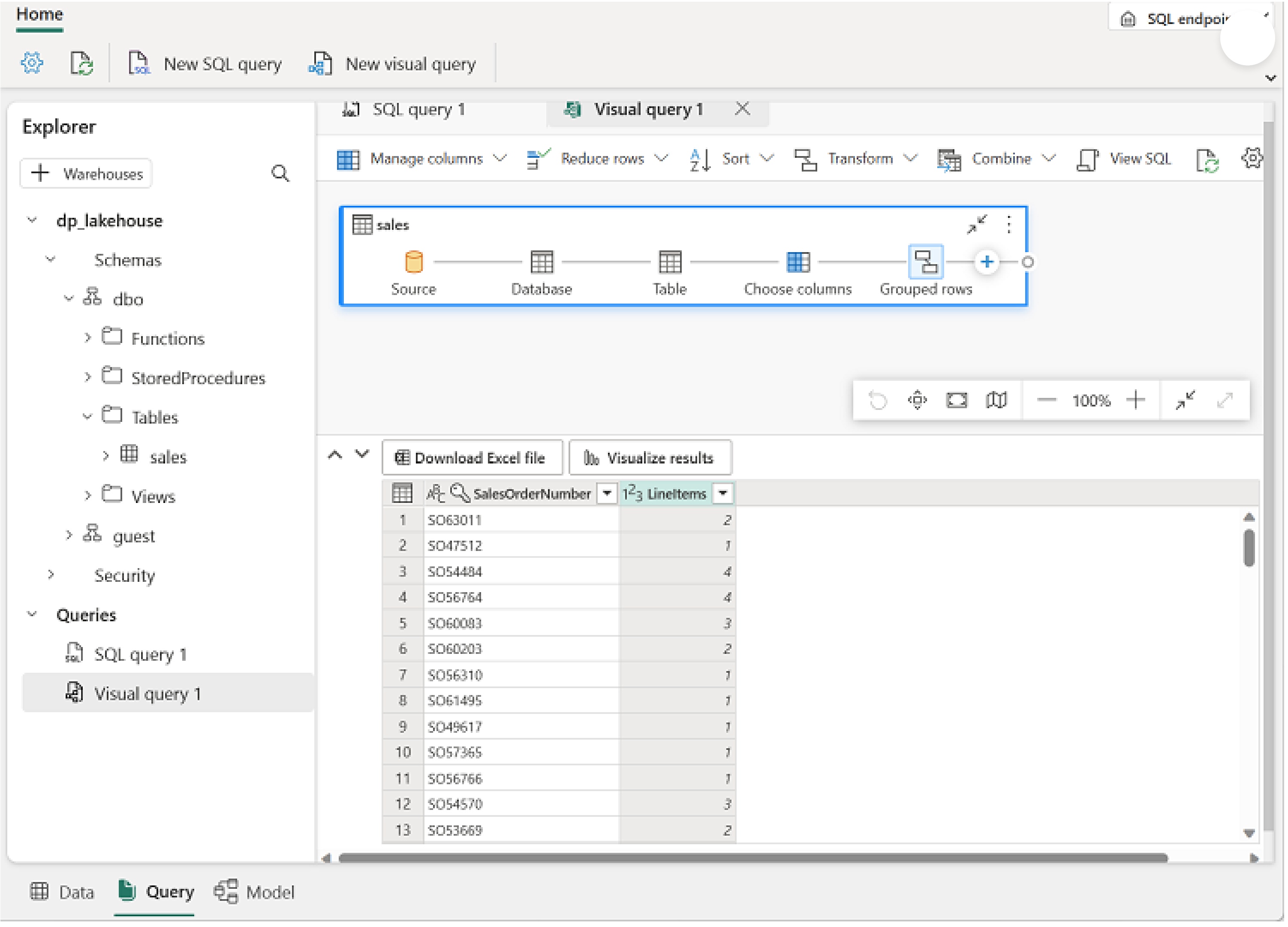
- ■ (Select all)
- ☑ SalesOrderNumber
- ☑ SalesOrderLineNumber
- ☐ OrderDate
- ☐ CustomerName
- ☐ EmailAddress
- ☐ Item
- ☐ Quantity
- ☐ UnitPrice
- ☐ TaxAmount

OK Cancel

4. in the **Transform** menu, select **Group by**. Then group the data by using the following **Basic** settings:

    - **Group by**: SalesOrderNumber
    - **New column name**: LineItems
    - **Operation**: Count distinct values
    - **Column**: SalesOrderLineNumber

   When you're done, the results pane under the visual query shows the number of line items for each sales order.

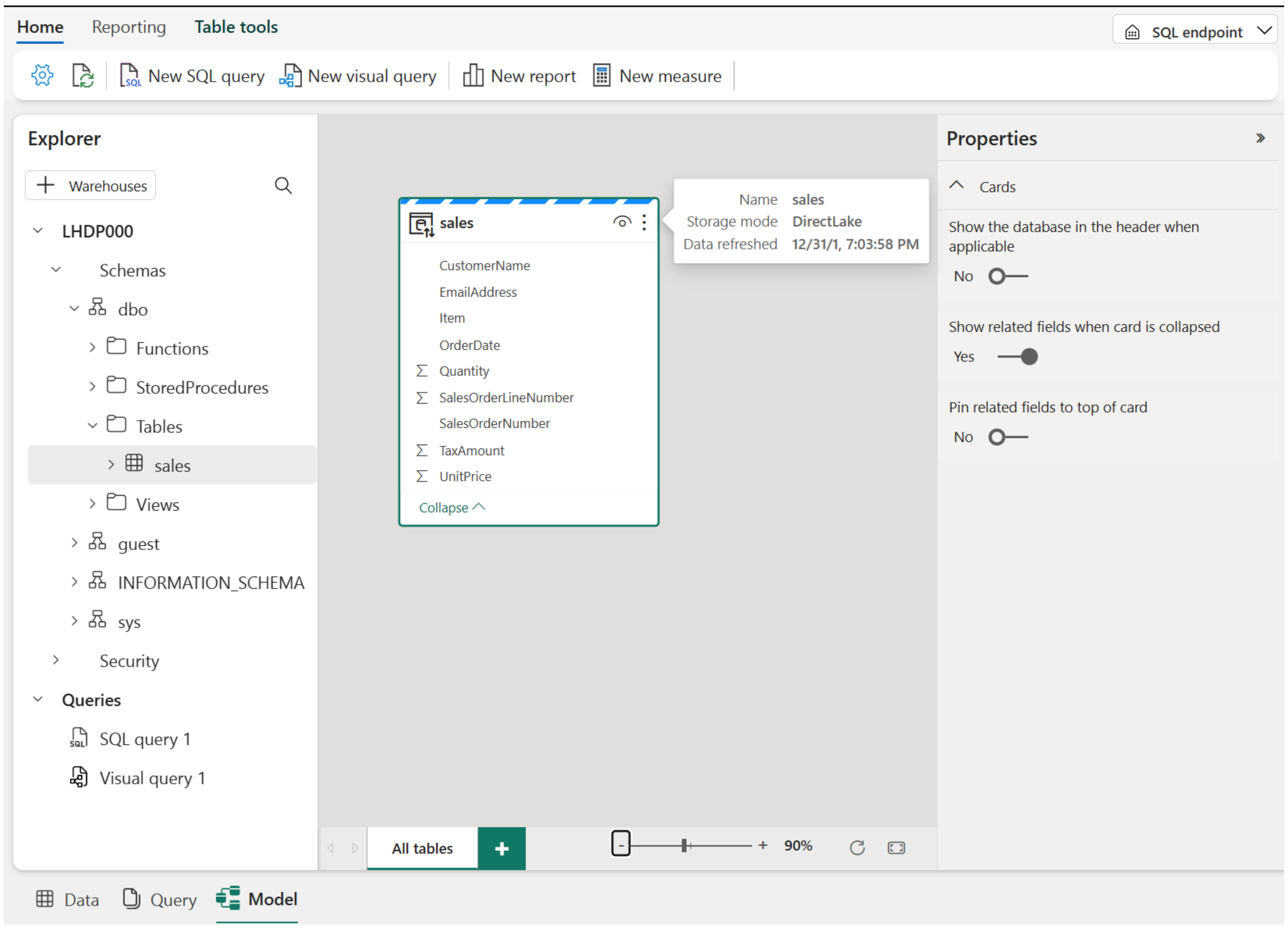


	SalesOrderNumber	LineItems
1	SO63011	2
2	SO47512	1
3	SO54484	4
4	SO56764	4
5	SO60083	3
6	SO60203	2
7	SO56310	1
8	SO61495	1
9	SO49617	1
10	SO57365	1
11	SO56766	1
12	SO54570	3
13	SO53669	2

Create a report

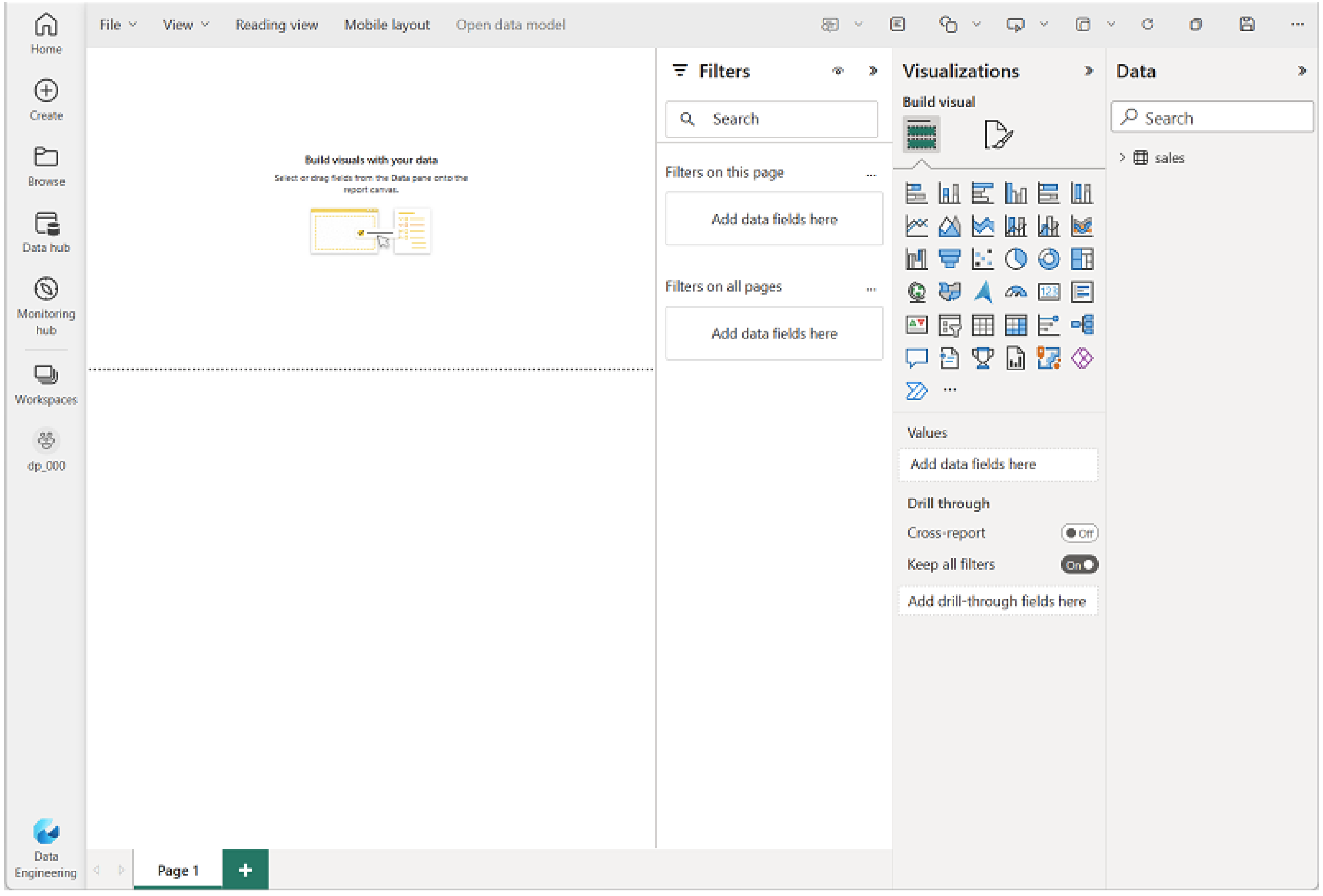
The tables in your lakehouse are automatically added to a default dataset that defines a data model for reporting with Power BI.

1. At the bottom of the SQL Endpoint page, select the **Model** tab. The data model schema for the dataset is shown.



> **Note**: In this exercise, the data model consists of a single table. In a real-world scenario, you would likely create multiple tables in your lakehouse, each of which would be included in the model. You could then define relationships between these tables in the model.

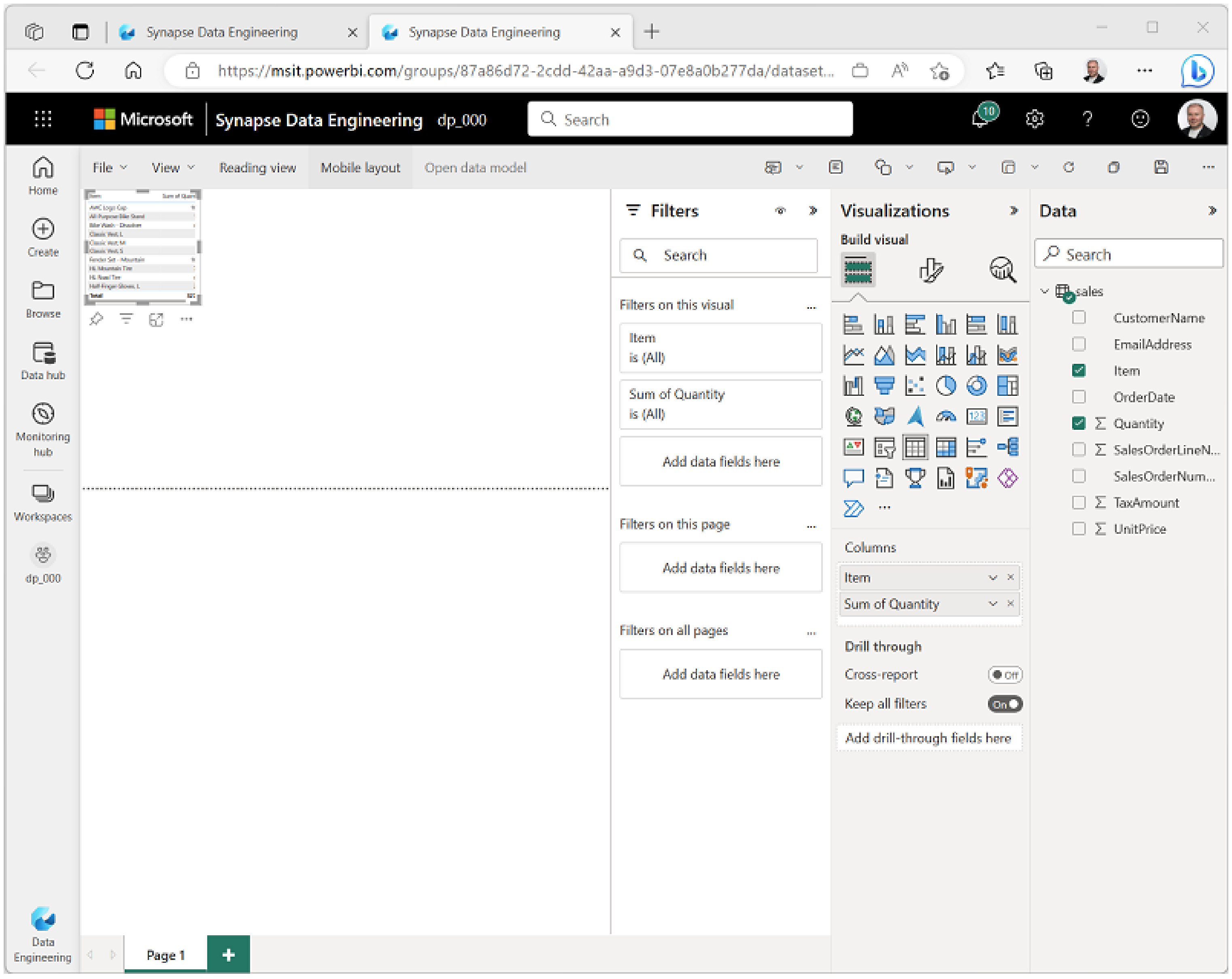
2. In the menu ribbon, select the **Reporting** tab. Then select **New report**. A new browser tab opens in which you can design your report.



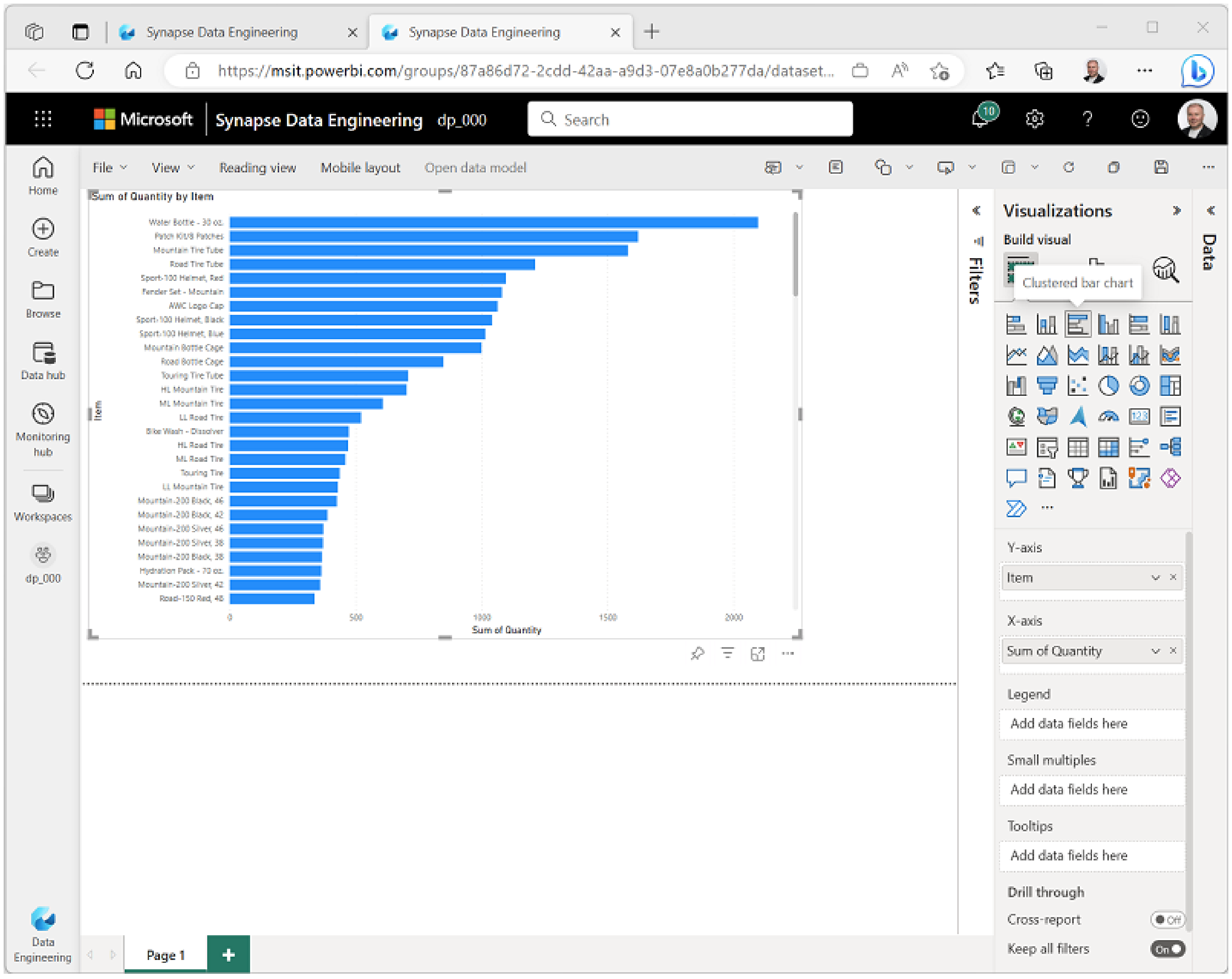
3. In the **Data** pane on the right, expand the **sales** table. Then select the following fields:

    - **Item**
    - **Quantity**

    A table visualization is added to the report:



4. Hide the **Data** and **Filters** panes to create more space. Then ensure the table visualization is selected and in the **Visualizations** pane, change the visualization to a **Clustered bar chart** and resize it as shown here.



5. On the **File** menu, select **Save**. Then save the report as **Item Sales Report** in the workspace you created previously.
6. Close the browser tab containing the report to return to the SQL endpoint for your lakehouse. Then, in the hub menu bar on the left, select your workspace to verify that it contains the following items:
   - Your lakehouse.
   - The SQL endpoint for your lakehouse.
   - A default dataset for the tables in your lakehouse.
   - The **Item Sales Report** report.

Clean up resources

In this exercise, you have created a lakehouse and imported data into it. You've seen how a lakehouse consists of files and tables stored in a OneLake data store. The managed tables can be queried using SQL, and are included in a default dataset to support data visualizations.

If you've finished exploring your lakehouse, you can delete the workspace you created for this exercise.

1. In the bar on the left, select the icon for your workspace to view all of the items it contains.
2. In the **...** menu on the toolbar, select **Workspace settings**.
3. In the **Other** section, select **Remove this workspace**.