


Create a Microsoft Fabric Lakehouse

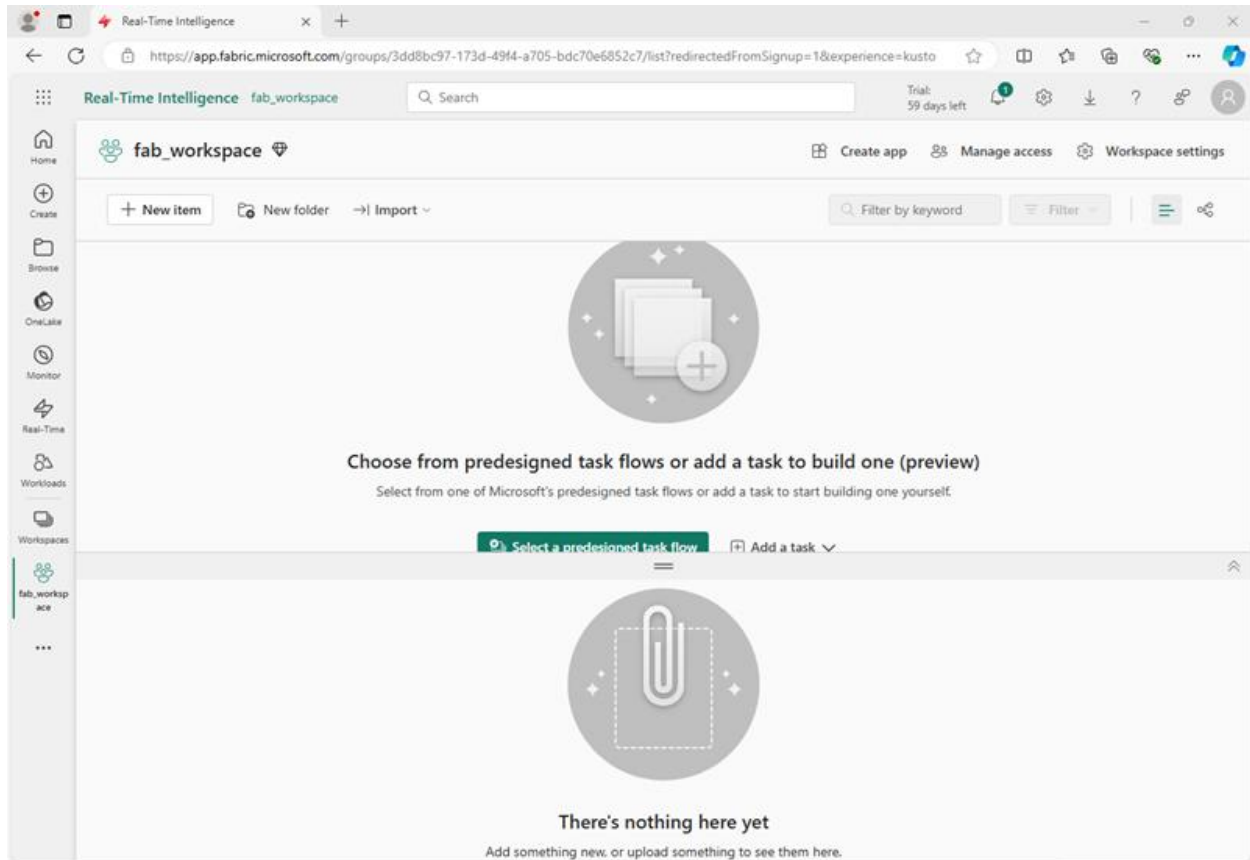
Large-scale data analytics solutions have traditionally been built around a *data warehouse*, in which data is stored in relational tables and queried using SQL. The growth in "big data" (characterized by high *volumes*, *variety*, and *velocity* of new data assets) together with the availability of 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.

Create a workspace

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

1. Navigate to the [Microsoft Fabric home page](https://app.fabric.microsoft.com/home?experience=fabric) at <https://app.fabric.microsoft.com/home?experience=fabric> in a browser, and sign in with your Fabric credentials.
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 in the **Advanced** section that includes Fabric capacity (*Trial*, *Premium*, or *Fabric*).
4. When your new workspace opens, it should be empty.



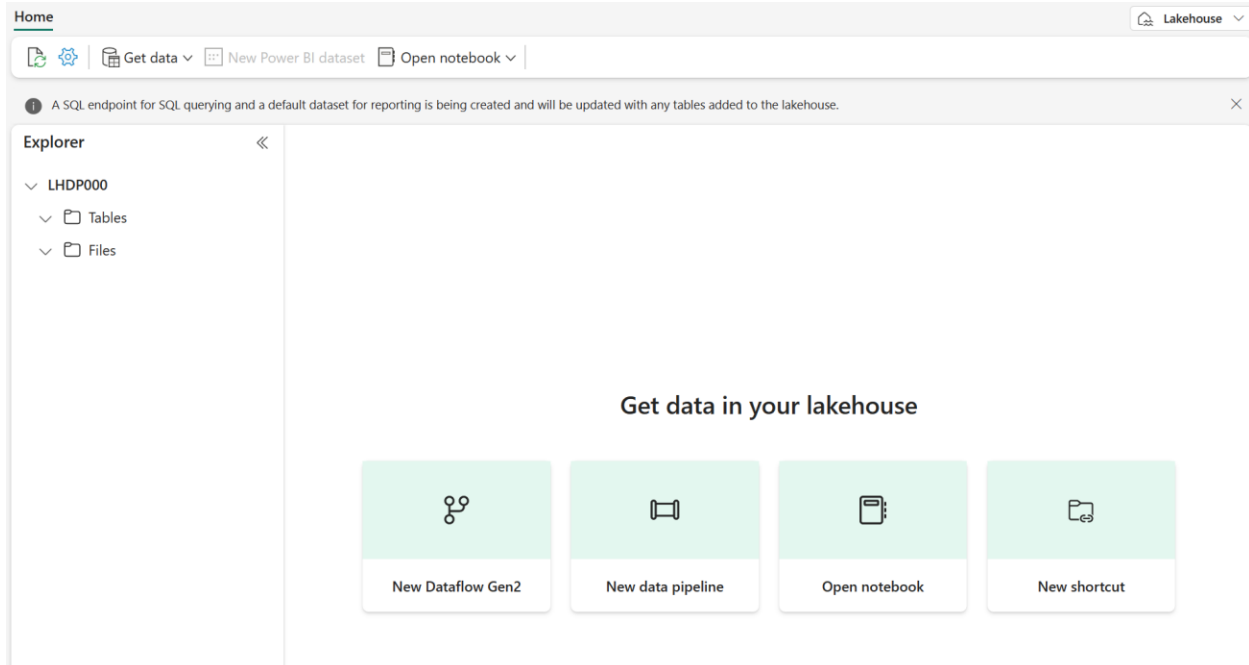
Create a lakehouse

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

1. On the menu bar on the left, select **Create**. In the *New* page, under the *Data Engineering* section, select **Lakehouse**. Give it a unique name of your choice.

Note: If the **Create** option is not pinned to the sidebar, you need to select the ellipsis (...) option first.

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



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.

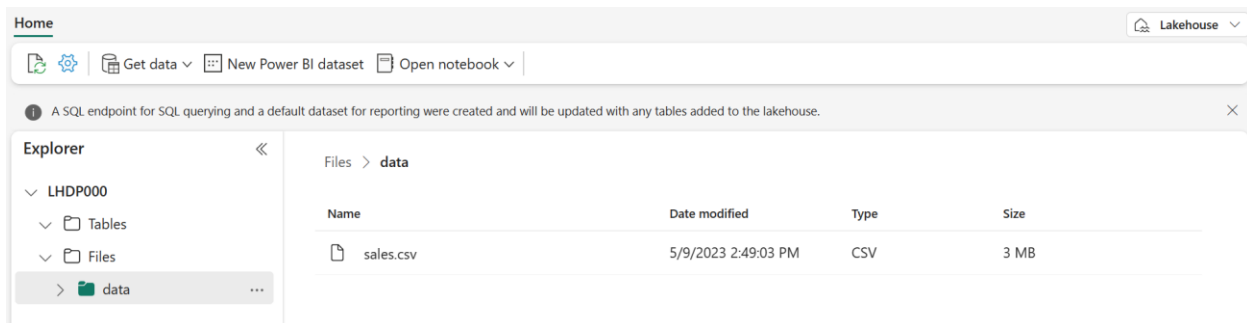
Upload a file

Fabric provides multiple ways to load data into the lakehouse, including built-in support for pipelines that copy data from 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](https://raw.githubusercontent.com/parveenkraina/fabric-solution-architect/refs/heads/main/files/sales.csv) file from raw.githubusercontent.com/parveenkraina/fabric-solution-architect/refs/heads/main/files/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 files**, 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:



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.

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

- Explorer

⏮

⏭

▼ LHDP000

▼ Tables

> sales

▼ Files

▼ 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

-

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 analytics endpoint**. Then wait a short time until the SQL analytics endpoint for your lakehouse opens in a visual interface from which you can query its tables.
2. Use the **New SQL query** button to open a new query editor, and enter the following SQL query:
3. SELECT Item, SUM(Quantity * UnitPrice) AS Revenue
4. FROM sales
5. GROUP BY Item

ORDER BY Revenue DESC;

Note: If you are in a lab VM and have any problems entering the SQL query, you can download the [01-Snippets.txt](https://raw.githubusercontent.com/parveenkraina/fabric-solution-architect/refs/heads/main/files/01-Snippets.txt) file from raw.githubusercontent.com/parveenkraina/fabric-solution-architect/refs/heads/main/files/01-Snippets.txt, saving it on the local computer (or lab VM if applicable). You can then copy the query from the text file.

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

The screenshot shows the Power BI Desktop interface. On the left, the 'Explorer' pane shows the database structure with 'dp_test_lake' selected. The main pane shows the SQL query editor with a query to calculate revenue from the 'sales' table. The Results pane displays a table with 130 rows and 2 columns: 'Item' and 'Revenue'.

	Item	Revenue
1	Road-150 Red, 48	1205876.99
2	Road-150 Red, 62	1202296.72
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	925946.496199999
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

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, expand the **New SQL query** option and 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:

Home SQL endpoint

[New SQL query](#) [New visual query](#)

Explorer

- Warehouses
- dp_test_lake
 - Schemas
 - dbo
 - Functions
 - StoredProcedures
 - Tables
 - sales
 - Views
 - guest
 - Security
- Queries
 - SQL query 1
 - Visual query 1

Visual query 1

Manage columns Reduce rows Sort Transform Combine View SQL

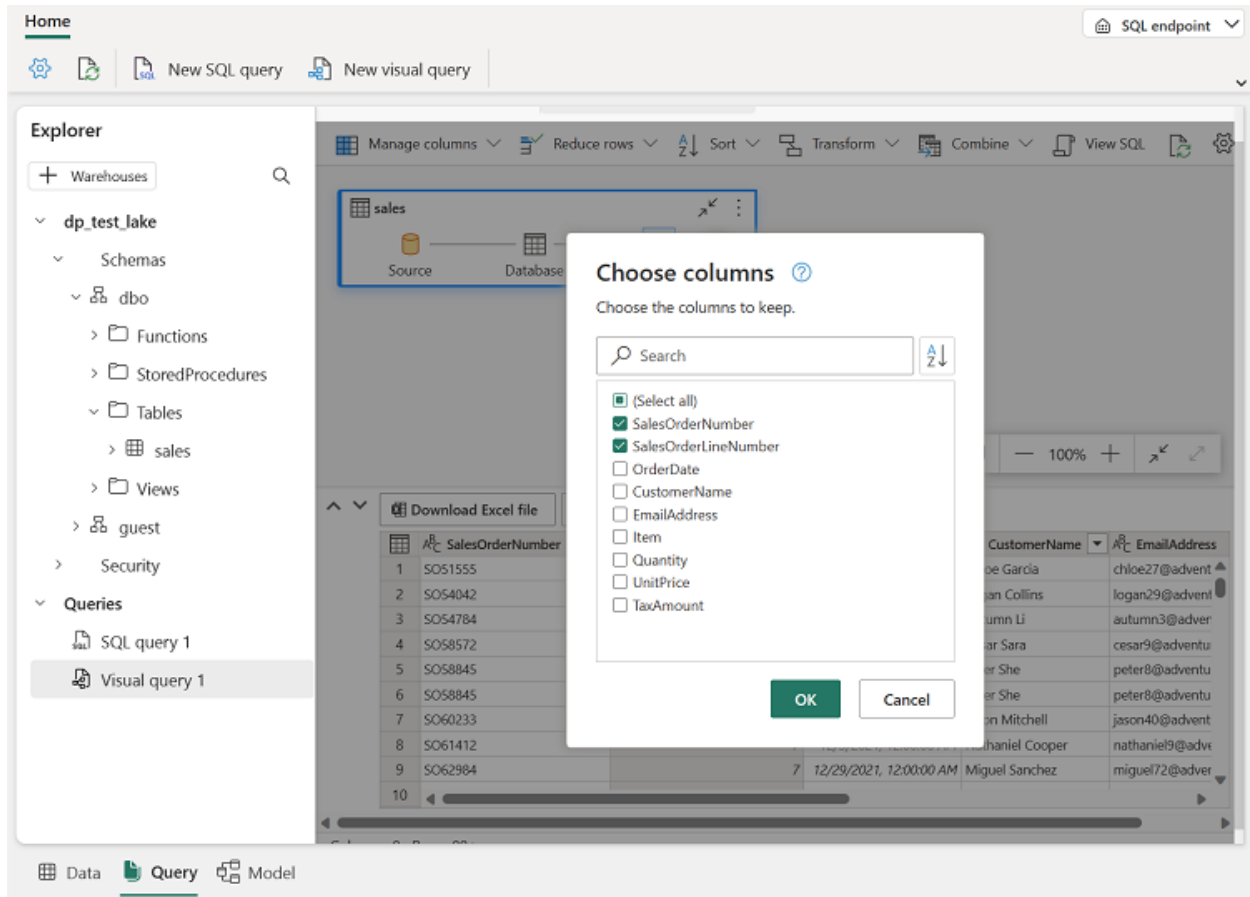
Diagram showing the data flow: Source (sales) → Database → Table → + (Join/Aggregation)

Download Excel file Visualize results

	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					

Data Query Model

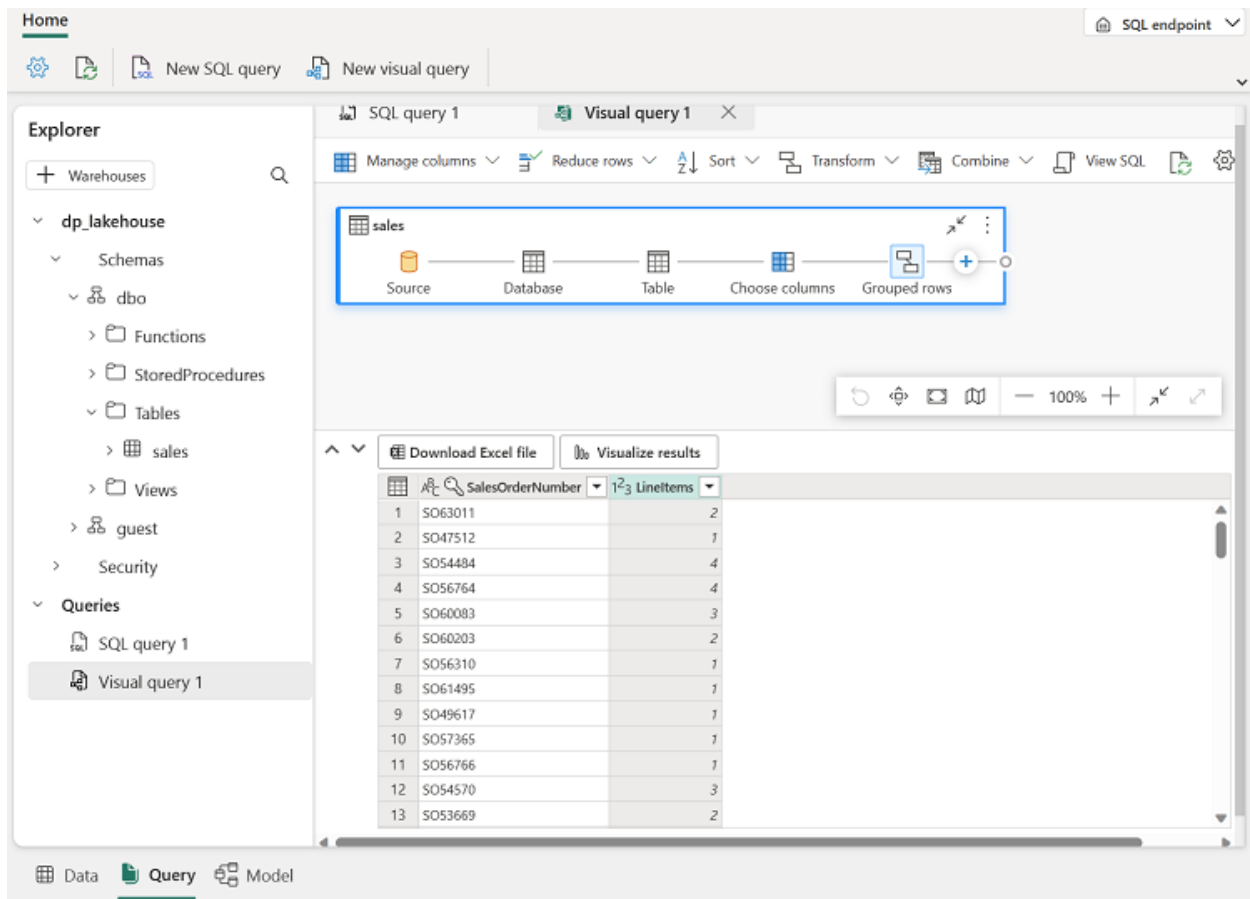
- In the **Manage columns** menu, select **Choose columns**. Then select only the **SalesOrderNumber** and **SalesOrderLineNumber** columns.



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

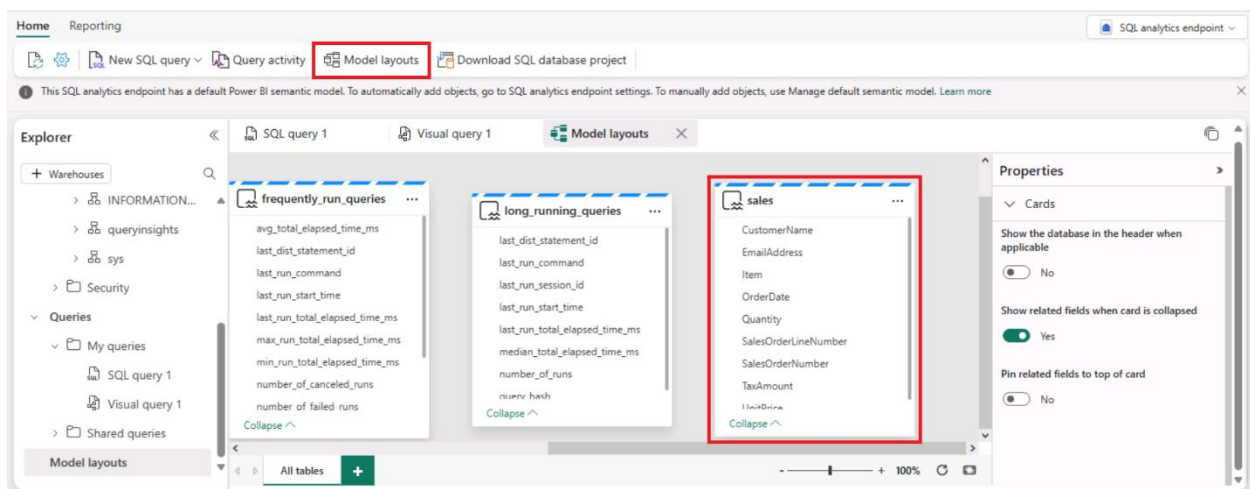
When you're done, the results pane under the visual query shows the number of line items for each sales order.



Create a report

The tables in your lakehouse are automatically added to a default semantic model for reporting with Power BI.

1. In the toolbar, select **Model layouts**. The data model schema for the semantic model is shown.

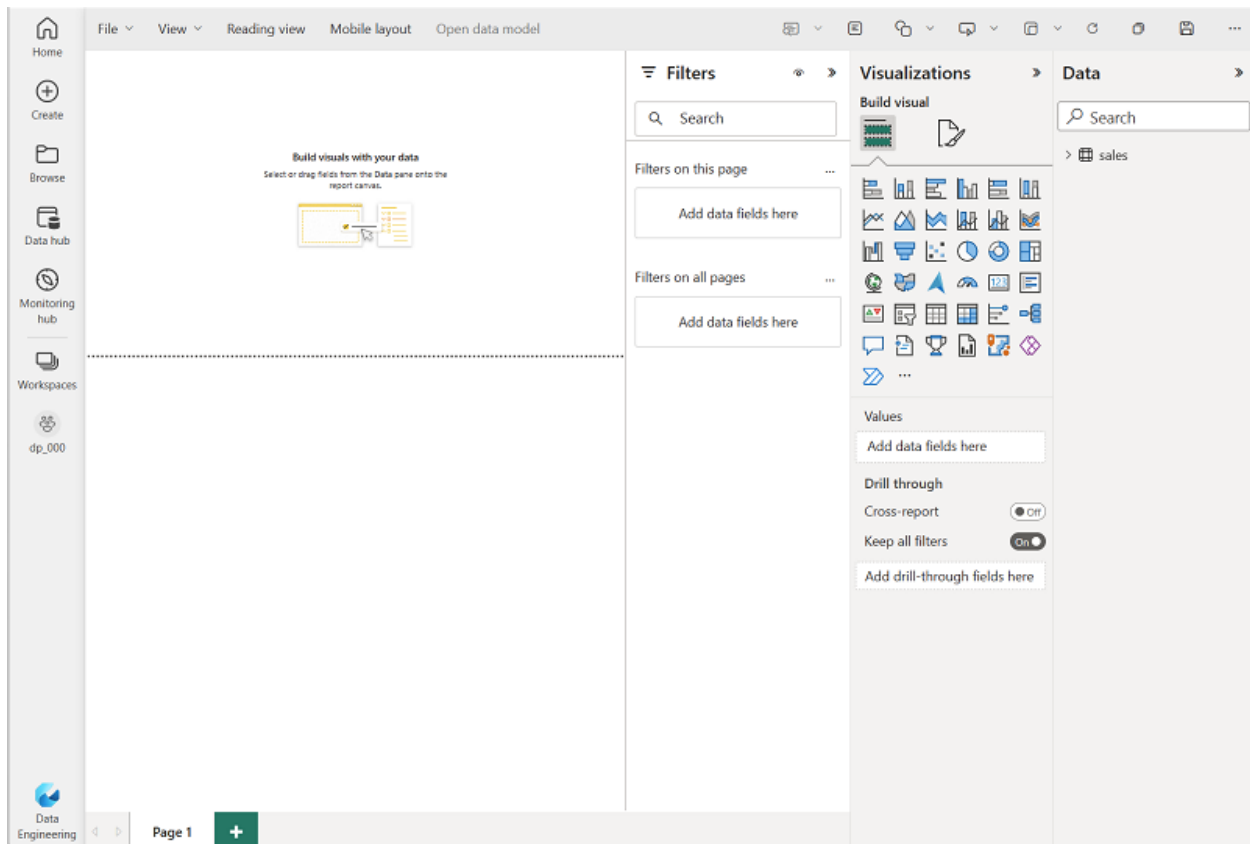


Note 1: In this exercise, the semantic model consists of a single table. In a real-world scenario, you would likely create multiple tables in your lakehouse, each of which would be included in the model. You could then define relationships between these tables in the model.

Note 2: The

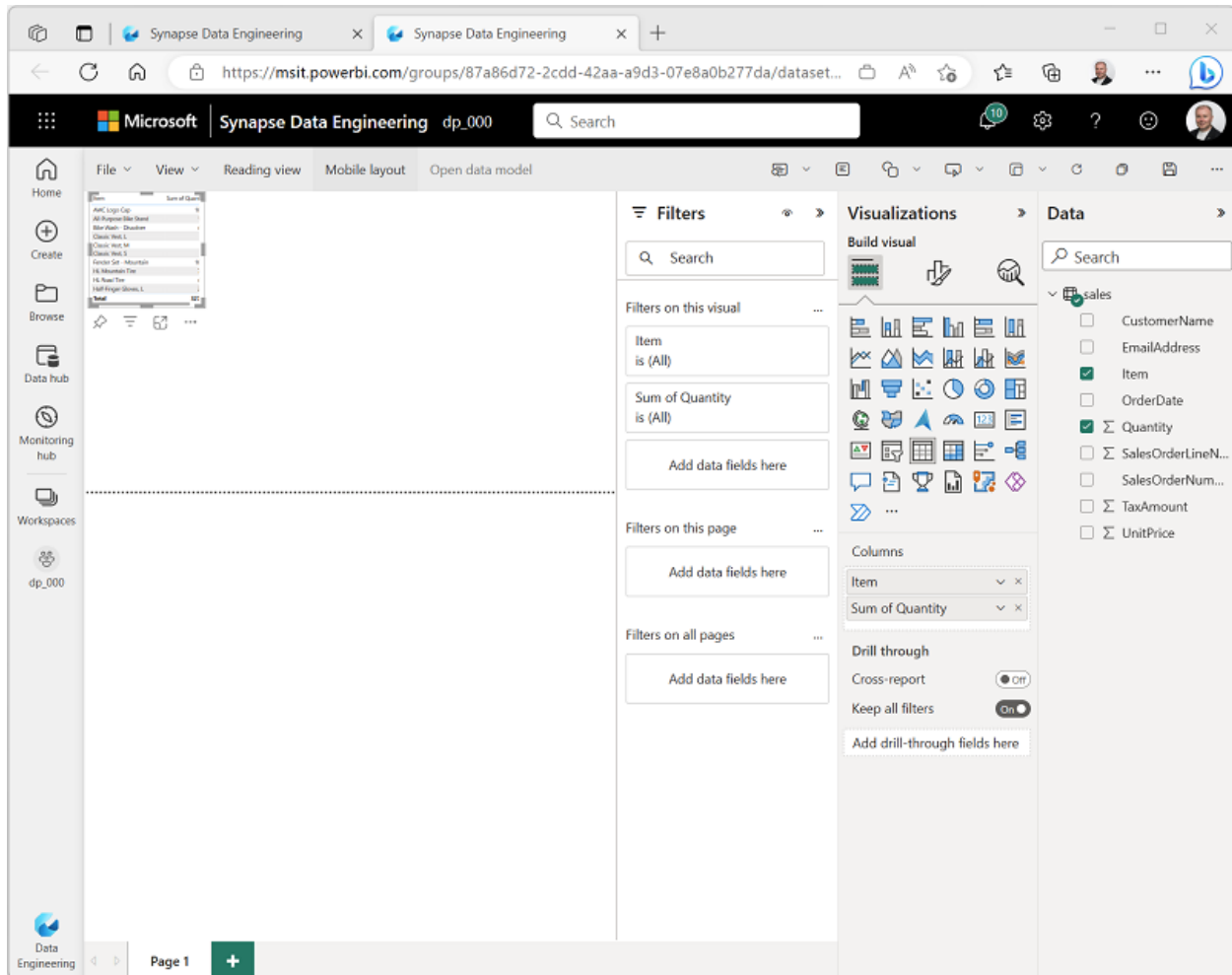
views **frequently_run_queries**, **long_running_queries**, **exec_sessions_history**, and **exec_requests_history** are part of the **queryinsights** schema automatically created by Fabric. It is a feature that provides a holistic view of historical query activity on the SQL analytics endpoint. Since this feature is out of the scope of this exercise, those views should be ignored for now.

2. In the menu ribbon, select the **Reporting** tab. Then select **New report**. Your current page will change to a report designer view.

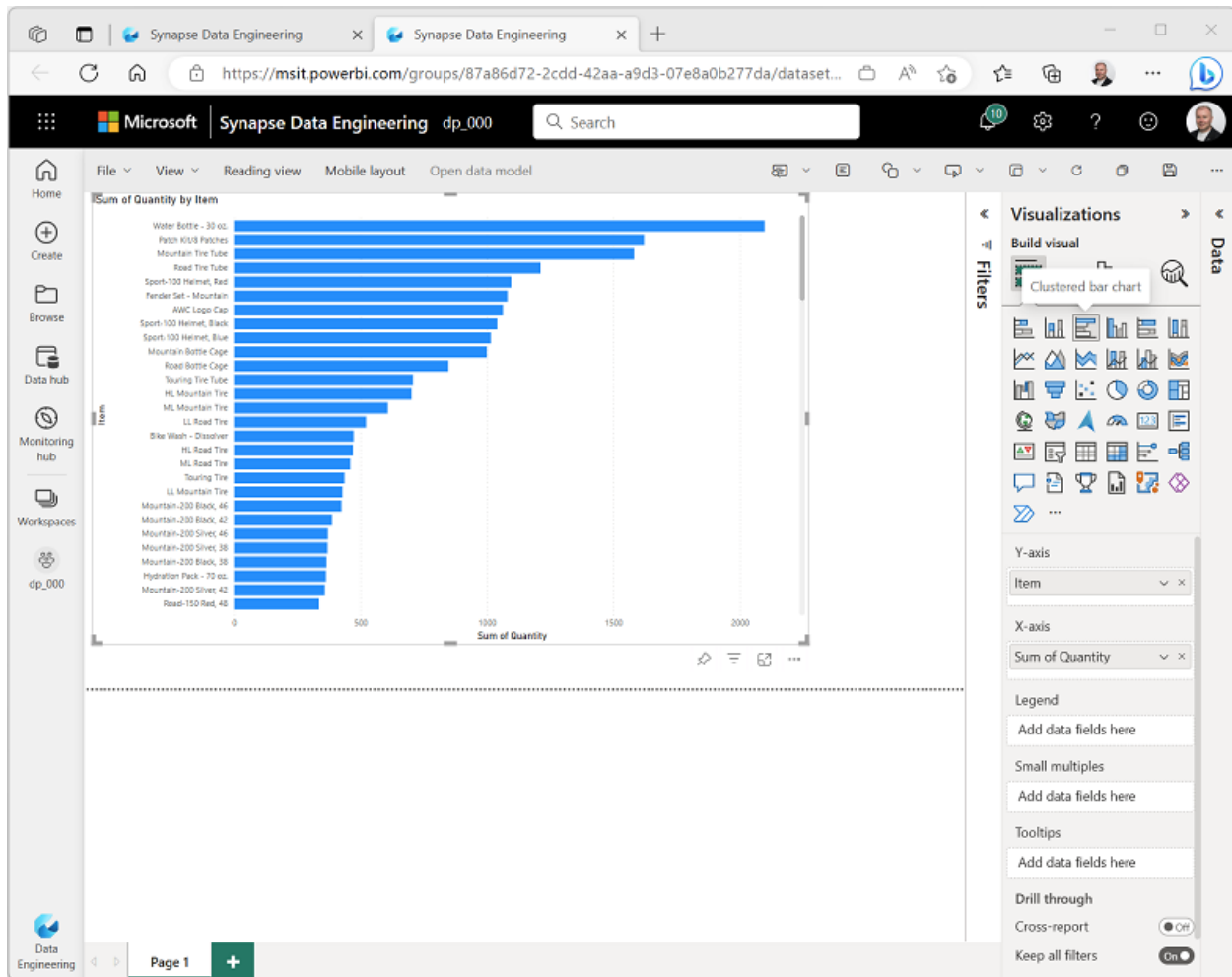


3. In the **Data** pane on the right, expand the **sales** table. Then select the following fields:
 - **Item**
 - **Quantity**

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. Now, in the hub menu bar on the left, select your workspace to verify that it contains the following items:
 - Your lakehouse.
 - The SQL analytics endpoint for your lakehouse.
 - A default semantic model for the tables in your lakehouse.
 - The **Item Sales Report** report.