# Get started with lakehouses in Microsoft Fabric

Lakehouses merge data lake storage flexibility with data warehouse analytics. Microsoft Fabric offers a lakehouse solution for comprehensive analytics on a single SaaS platform.

## Learning objectives

In this module, you'll learn how to:

* Describe core features and capabilities of lakehouses in Microsoft Fabric
* Create a lakehouse
* Ingest data into files and tables in a lakehouse
* Query lakehouse tables with SQL

# Introduction

The foundation of Microsoft Fabric is a **lakehouse**, which is built on top of the **OneLake** scalable storage layer and uses Apache Spark and SQL compute engines for big data processing. A lakehouse is a unified platform that combines:

* The flexible and scalable storage of a **data *lake***
* The ability to query and analyze data of a **data ware*house***

Imagine your company has been using a data warehouse to store structured data from its transactional systems, such as order history, inventory levels, and customer information. You have also collected unstructured data from social media, website logs, and third-party sources that are difficult to manage and analyze using the existing data warehouse infrastructure. Your company's new directive is to improve its decision-making capabilities by analyzing data in various formats across multiple sources, so the company chooses Microsoft Fabric.

In this module, we explore how a lakehouse in Microsoft Fabric can help address scenarios like this by providing a scalable and flexible data store for files and tables that you can query using SQL.

# Explore the Microsoft Fabric lakehouse

# A **Lakehouse** presents as a database and is built on top of a data lake using Delta format tables. Lakehouses combine the SQL-based analytical capabilities of a relational data warehouse and the flexibility and scalability of a data lake. Lakehouses store all data formats and can be used with various analytics tools and programming languages. As cloud-based solutions, lakehouses can scale automatically and provide high availability and disaster recovery.

# 

Some benefits of a lakehouse include:

* Lakehouses use Spark and SQL engines to process large-scale data and support machine learning or predictive modeling analytics.
* Lakehouse data is organized in a *schema-on-read format*, which means you define the schema as needed rather than having a predefined schema.
* Lakehouses support ACID (Atomicity, Consistency, Isolation, Durability) transactions through Delta Lake formatted tables for data consistency and integrity.
* Lakehouses are a single location for data engineers, data scientists, and data analysts to access and use data.

A lakehouse is a great option if you want a scalable analytics solution that maintains data consistency. It's important to evaluate your specific requirements to determine which solution is the best fit.

## **Microsoft Fabric lakehouses**

In Microsoft Fabric, you can create a lakehouse in any premium tier workspace. After creating a lakehouse, you can load data - in any common format - from various sources; including local files, databases, or APIs.

Data ingestion can also be automated using **Data Factory Pipelines or Dataflows (Gen2**) in Microsoft Fabric. Additionally, you can create Fabric shortcuts to data in external sources, such as **Azure Data Lake Store Gen2** or a **Microsoft OneLake** location outside of the lakehouse's own storage.

The **Lakehouse Explorer** enables you to browse files, folders, shortcuts, and tables; and view their contents within the Fabric platform.

After you've ingested the data into the lakehouse, you can use **Notebooks or Dataflows** (Gen2) to explore and transform it.

**Dataflows (Gen2)** are based on **Power Query** - a familiar tool to data analysts using Excel or Power BI that provides visual representation of transformations as an alternative to traditional programming.

**Data Factory Pipelines** can be used to orchestrate Spark, Dataflow, and other activities; enabling you to implement complex data transformation processes.

After transforming your data, you can query it using SQL, use it to train machine learning models, perform real-time analytics, or develop reports in Power BI.

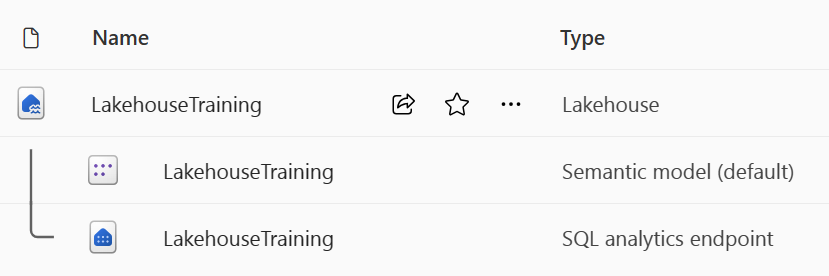
You can also apply data governance policies to your lakehouse, such as data classification and access control.

## **Create and explore a lakehouse**

Before you can create a lakehouse, you create a workspace in the Microsoft Fabric platform. Configure your workspace to allow Fabric resources.

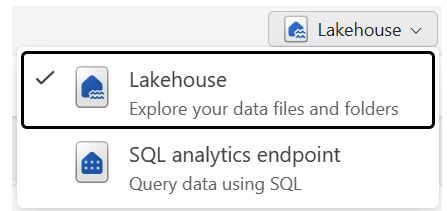
You create and configure a new lakehouse in the Data Engineering workload. Each L produces three named items in the Fabric-enabled workspace:

* **Lakehouse** is the lakehouse storage and metadata, where you interact with files, folders, and table data.
* **Semantic model (default)** is an automatically created data model based on the tables in the lakehouse. Power BI reports can be built from the semantic model.
* **SQL Endpoint** is a read-only SQL endpoint through which you can connect and query data with Transact-SQL.



You can work with the data in the lakehouse in two modes:

* **Lakehouse** enables you to add and interact with tables, files, and folders in the lakehouse.
* **SQL analytics endpoint** enables you to use SQL to query the tables in the lakehouse and manage its relational data model.



## **Access data using shortcuts**

**Shortcuts** enable you to integrate data into your lakehouse while keeping it stored in external storage.

Shortcuts are useful when you need data in your lakehouse from a different storage account or even a different cloud provider. Create shortcuts to different storage accounts, and other Fabric items like data warehouses, KQL databases, and other lakehouses.

Source data permissions and credentials are all managed by OneLake. When accessing data through a shortcut to another OneLake location, the identity of the calling user will be used to authorize access to the data in the target path of the shortcut. The user must have permissions in the target location to read the data.

*Shortcuts* can be created in both lakehouses and KQL databases, and appear as a folder in the lake. Spark, SQL, Real-Time Analytics, and Analysis Services can access data via shortcuts when querying data.

Shortcuts have limited data source connectors, so when you can't use shortcuts, you can ingest data directly into your lakehouse.

For more information on how to use shortcuts, see [**OneLake shortcuts documentation**](https://learn.microsoft.com/en-us/fabric/onelake/onelake-shortcuts) in the Microsoft Fabric documentation.

## **Ingest data into a lakehouse**

There are many ways to load data into a Fabric lakehouse, including:

* **Upload**: Upload local files or folders to the lakehouse. You can then explore and process the file data, and load the results into tables.
* **Dataflows (Gen2)**: Import and transform data from a range of sources using Power Query Online, and load it directly into a table in the lakehouse.
* **Notebooks**: Use notebooks in Fabric to ingest and transform data, and load it into tables or files in the lakehouse.
* **Data Factory pipelines**: Copy data and orchestrate data processing activities, loading the results into tables or files in the lakehouse.

## **Grant access to a lakehouse**

Fabric lakehouse permissions are granted either at the workspace or item level. In the **Workspace settings**, you can choose between four different roles. These permissions should be reserved for collaboration for all items within the workspace, not only the lakehouse.

The *lakehouse* is a single item within the workspace, so access is controlled at this level as well. You can grant access directly to the lakehouse within Fabric or you can grant access to only be used via the SQL analytics endpoint. This access means users can connect to the SQL analytics endpoint with other tools like SQL Server Management Studio (SSMS) to query the data.

You can also grant object-level security by using the SQL analytics endpoint to further control what users can access.

Security is an important aspect of the Microsoft Fabric environment. To understand what data access is available and how to grant or revoke, please review the full [**Security in Microsoft Fabric documentation**](https://learn.microsoft.com/en-us/fabric/security/security-overview#access-data).

# Explore and transform data in a lakehouse

After loading data into the lakehouse, you can use various tools and techniques to explore and transform it, including:

* **Apache Spark**: Each Fabric lakehouse can use Spark pools through Notebooks or Spark Job Definitions to process data in files and tables in the lakehouse using Scala, PySpark, or Spark SQL.
  + **Notebooks**: Interactive coding interfaces in which you can use code to read, transform, and write data directly to the lakehouse as tables and/or files.
  + **Spark job definitions**: On-demand or scheduled scripts that use the Spark engine to process data in the lakehouse.
* **SQL analytic endpoint**: Each lakehouse includes a SQL analytic endpoint through which you can run Transact-SQL statements to query, filter, aggregate, and otherwise explore data in lakehouse tables.
* **Dataflows (Gen2)**: In addition to using a dataflow to ingest data into the lakehouse, you can create a dataflow to perform subsequent transformations through Power Query, and optionally land transformed data back to the lakehouse.
* **Data pipelines**: Orchestrate complex data transformation logic that operates on data in the lakehouse through a sequence of activities (such as dataflows, Spark jobs, and other control flow logic).

## **Analyze and visualize data in a lakehouse**

The data in your lakehouse tables is included in a **semantic model** that defines a relational model for your data. You can edit this semantic model (or create other semantic models), defining custom measures, hierarchies, aggregations, and other elements of a semantic model. You can then use the semantic model as the source for a Power BI report that enables you to visualize and analyze the data.

By combining the data visualization capabilities of Power BI with the centralized storage and tabular schema of a data lakehouse, you can implement an end-to-end analytics solution on a single platform.

**Exercise**

**Create and ingest data with a Microsoft Fabric lakehouse**

In this exercise, explore Microsoft Fabric lakehouse tasks like creating a lakehouse, importing data, querying tables with SQL, and generating reports. The exercise emphasizes the importance of the lakehouse as a central component in data engineering, warehousing, and analytics, enabling users to effectively manage and analyze their data within the lakehouse environment.

# *Create a Lakehouse*

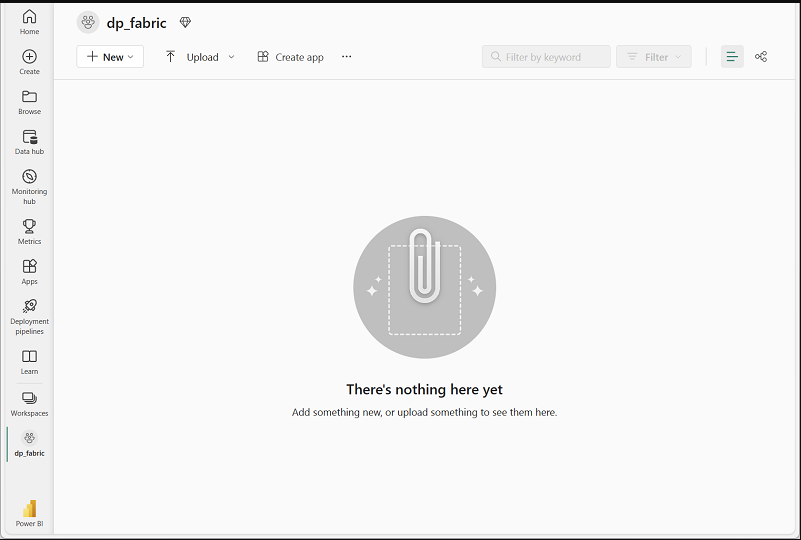
Large-scale data analytics solutions have traditionally been built around a data warehouse, in which data is stored in relational tables and queried using SQL. The growth in “big data” (characterized by high volumes, variety, and velocity of new data assets) together with the availability of low-cost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

## **Creating a workspace**

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

1. On the [Microsoft Fabric home page](https://app.fabric.microsoft.com/) at https://app.fabric.microsoft.com, 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 in the **Advanced** section that includes Fabric capacity (Trial, Premium, or Fabric).
4. When your new workspace opens, it should be empty.

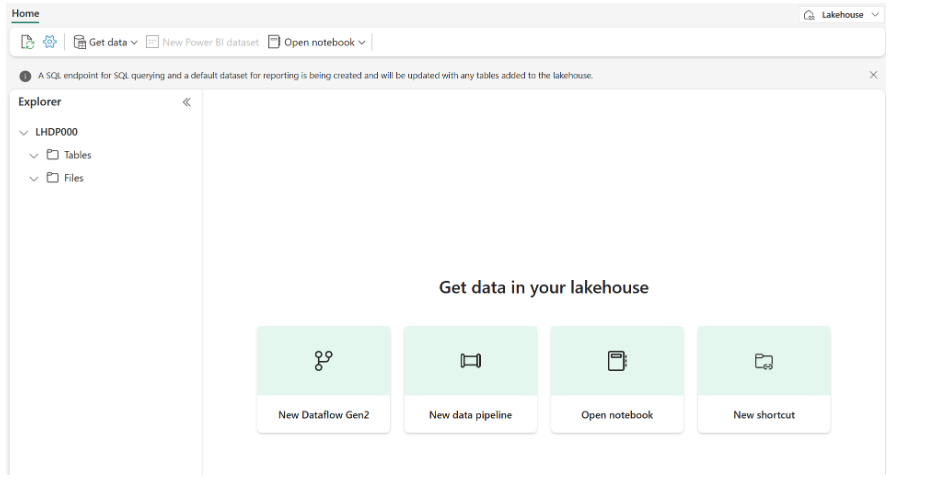


## **Create a lakehouse**

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

1. 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:



1. 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.

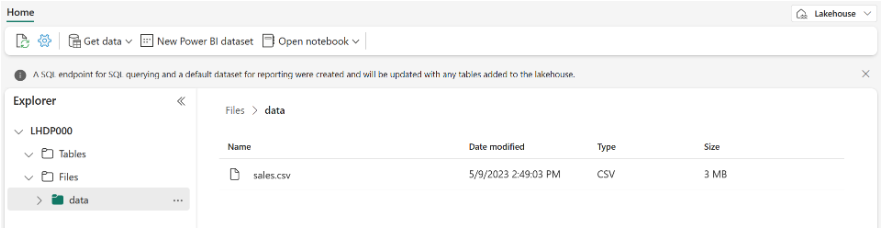
## **Upload a file**

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/MicrosoftLearning/dp-data/main/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.

1. 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**.
2. 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).
3. After the file has been uploaded, select the **Files/data** folder and verify that the **sales.csv** file has been uploaded, as shown here:



1. 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.

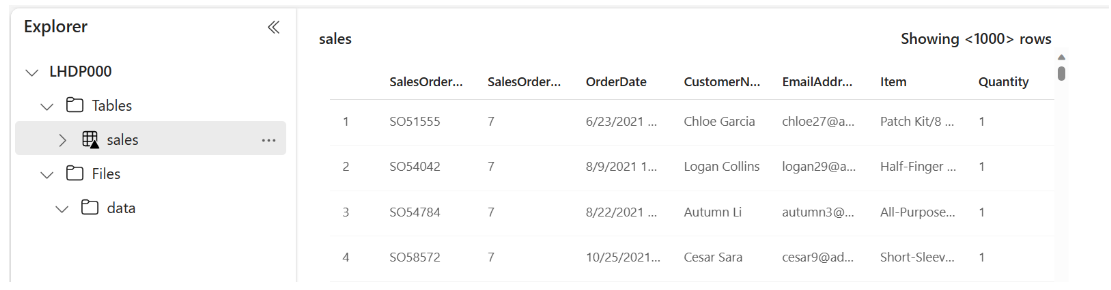
## **Load file data into a table**

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.

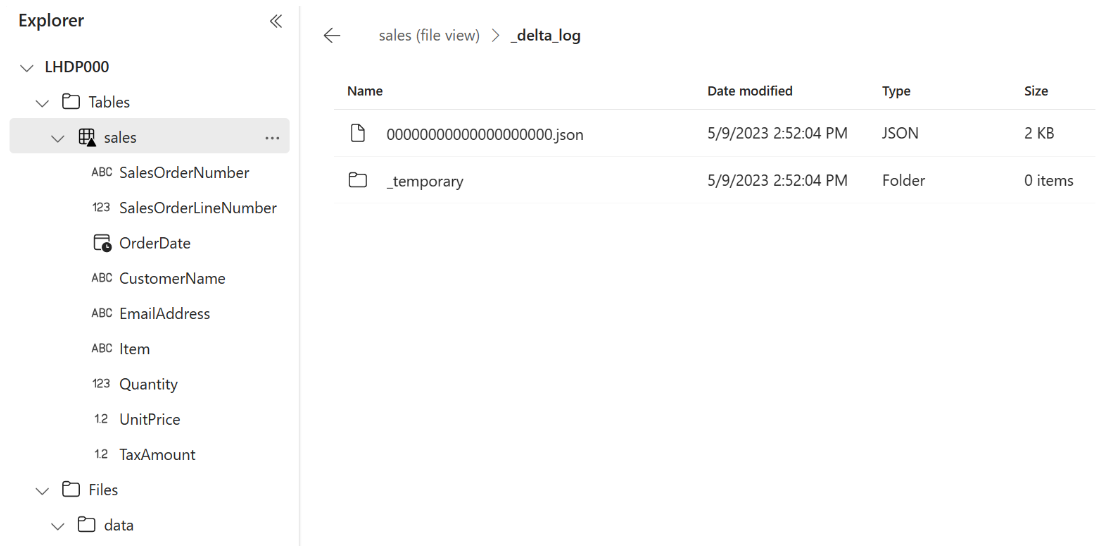
**Tip**: If the **sales** table does not automatically appear, in the **…** menu for the **Tables** folder, select **Refresh**.

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



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

table.

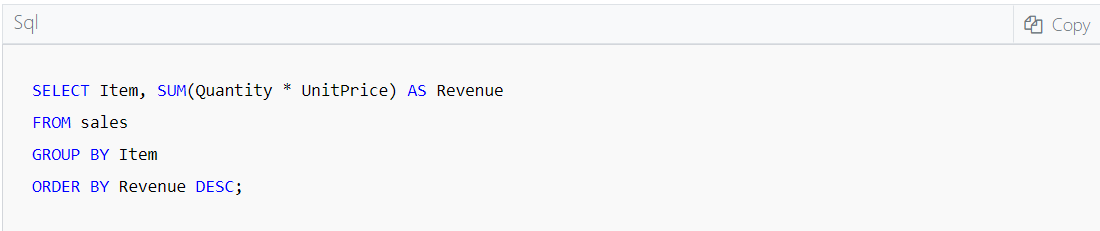


Files for a delta table are stored in Parquet format, and include a subfolder named **\_delta\_log** in which details of transactions applied to the table are logged.

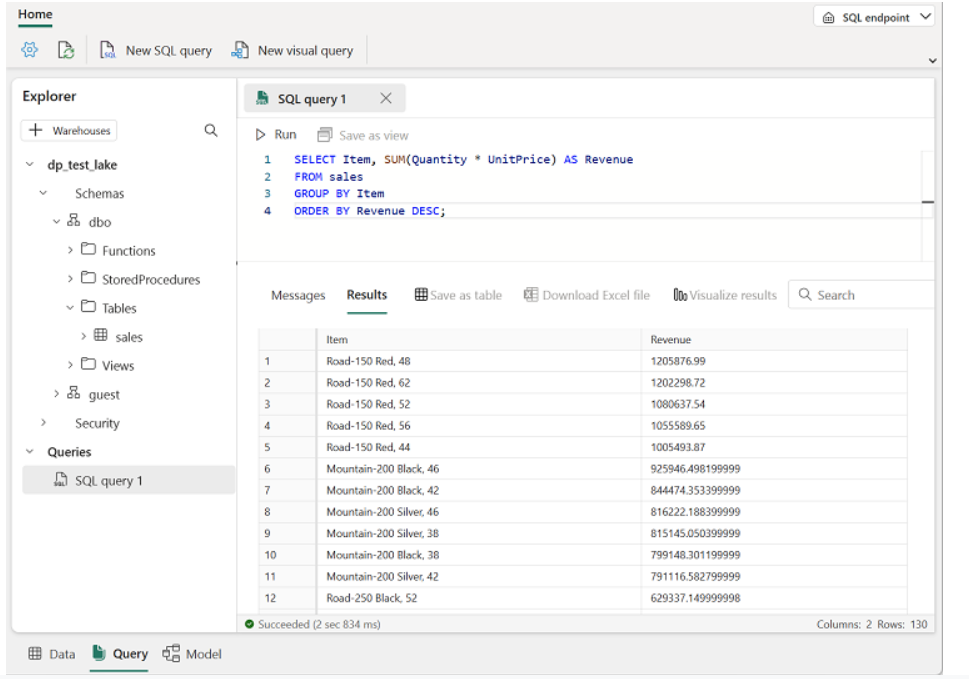
## **Use SQL to query tables**

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:



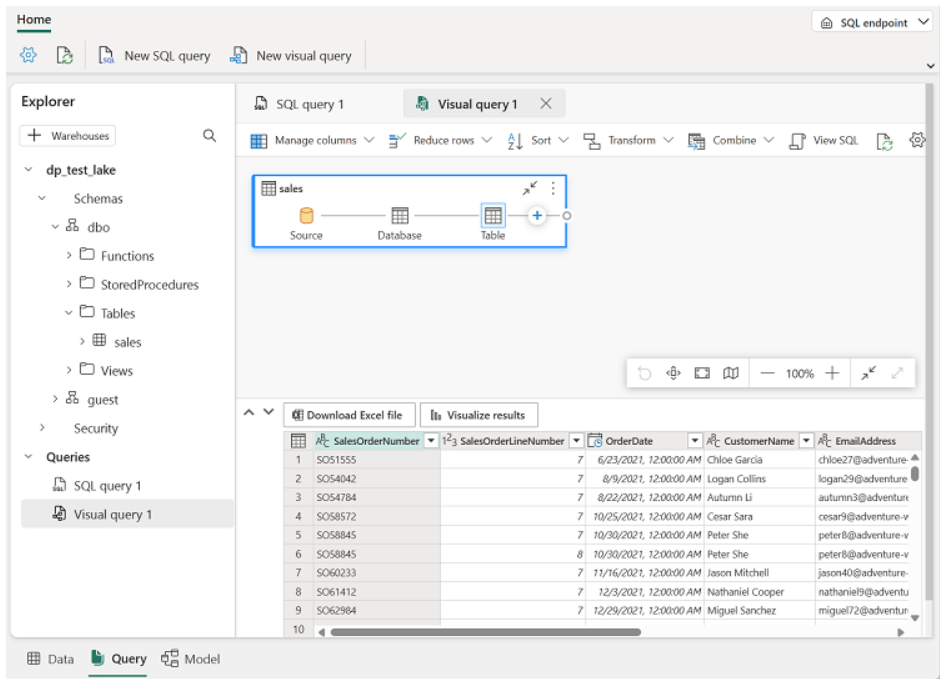
1. Use the **▷ Run** button to run the query and view the results, which should show the total revenue for each product.



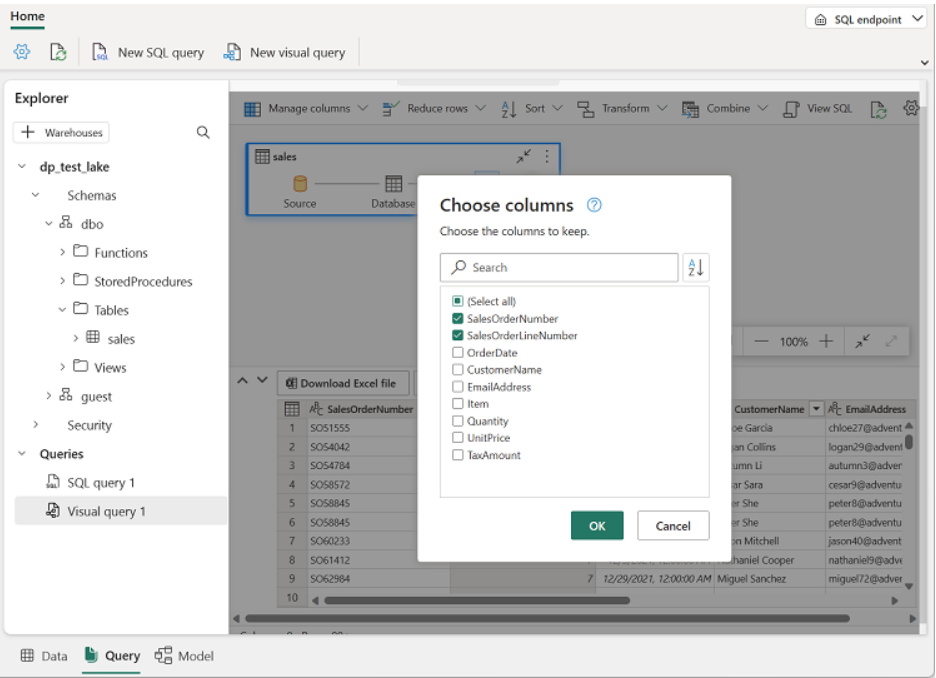
## **Create a visual query**

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:



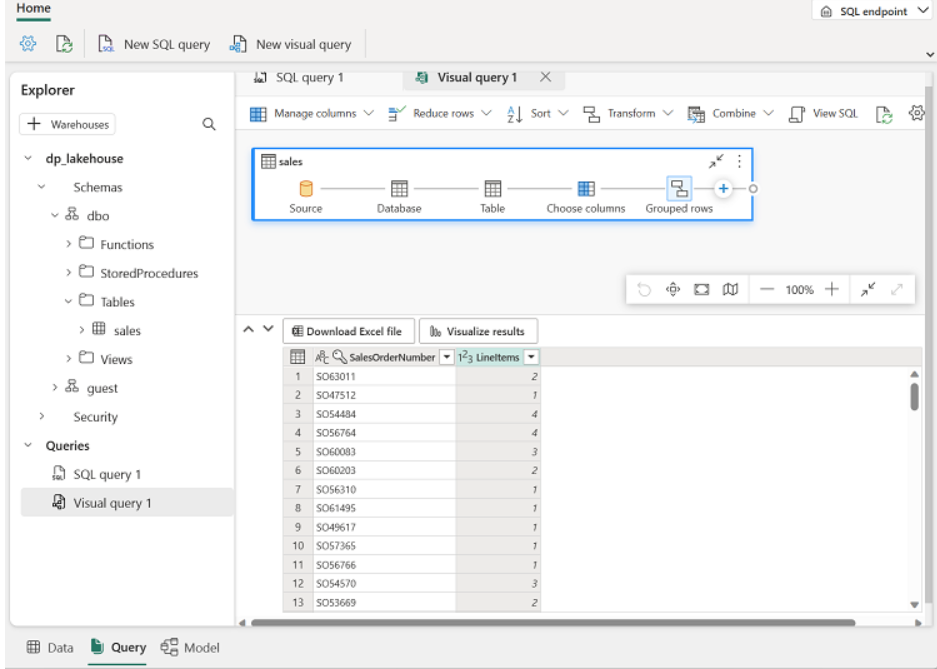
1. In the **Manage columns** menu, select **Choose columns**. Then select only the **SalesOrderNumber** and **SalesOrderLineNumber** columns.



1. 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

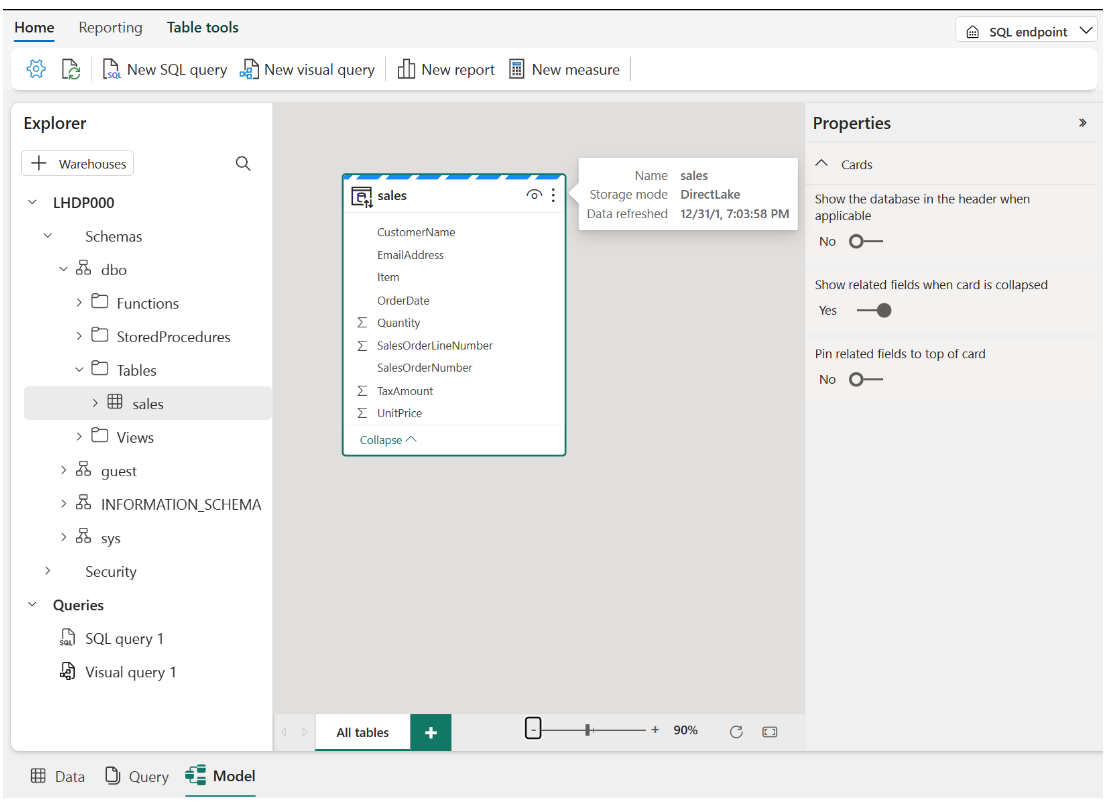
When you’re done, the results pane under the visual query shows the number of line items for each sales order.



## **Create a report**

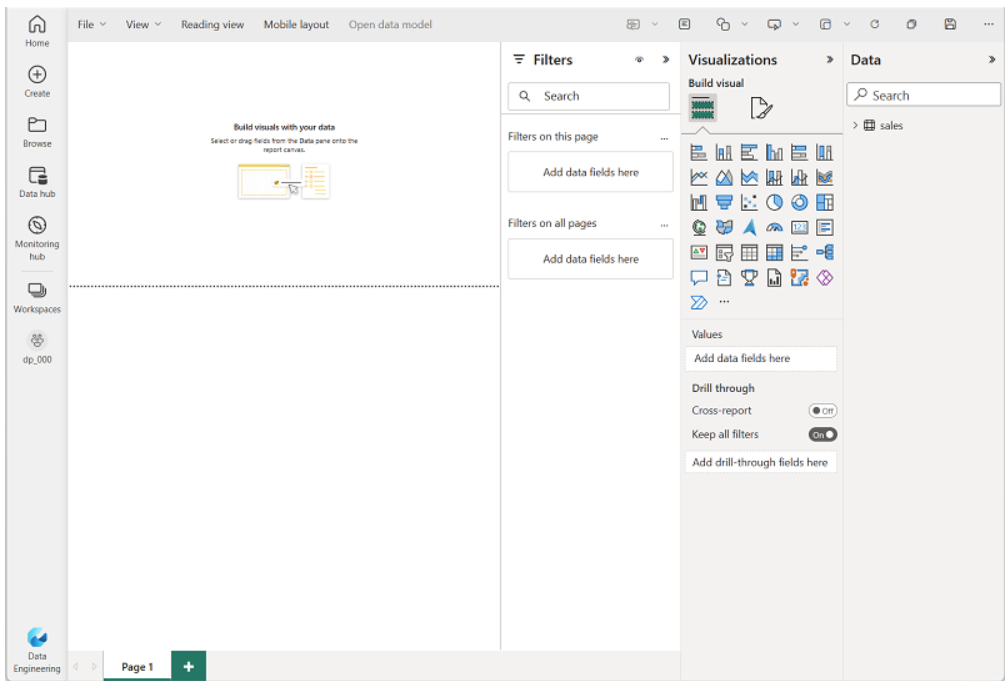
The tables in your lakehouse are automatically added to a default semantic model for reporting with Power BI.

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



**Note**: 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.

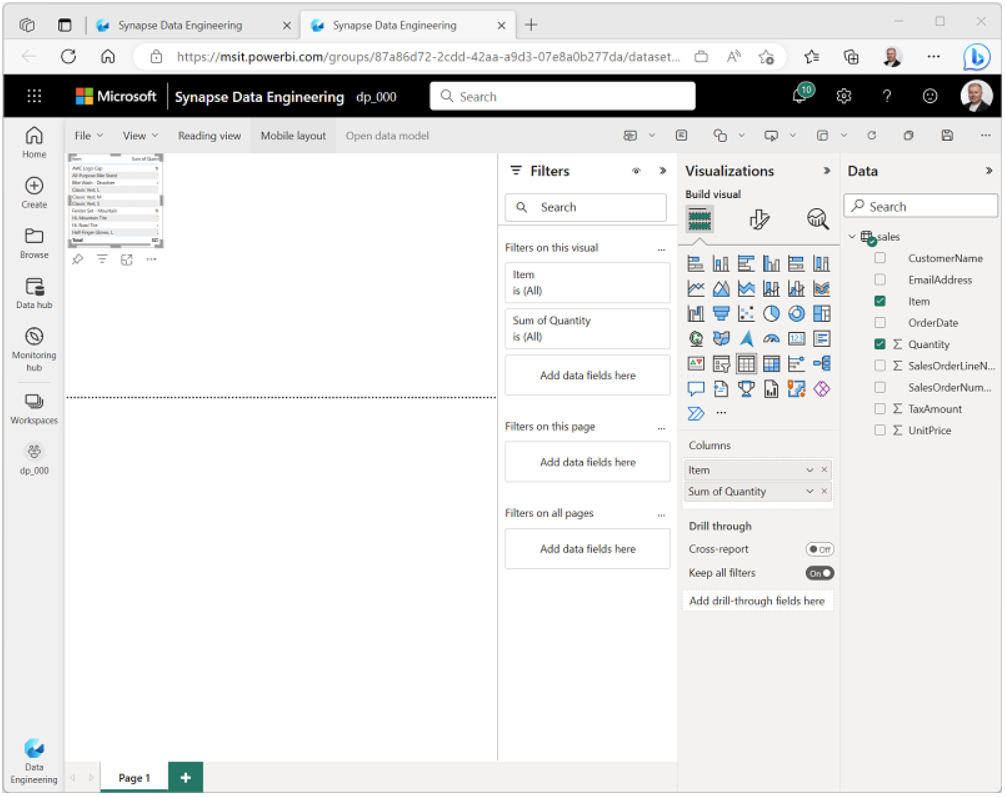
1. In the menu ribbon, select the **Reporting** tab. Then select **New report**. A new browser tab opens in which you can design your report.



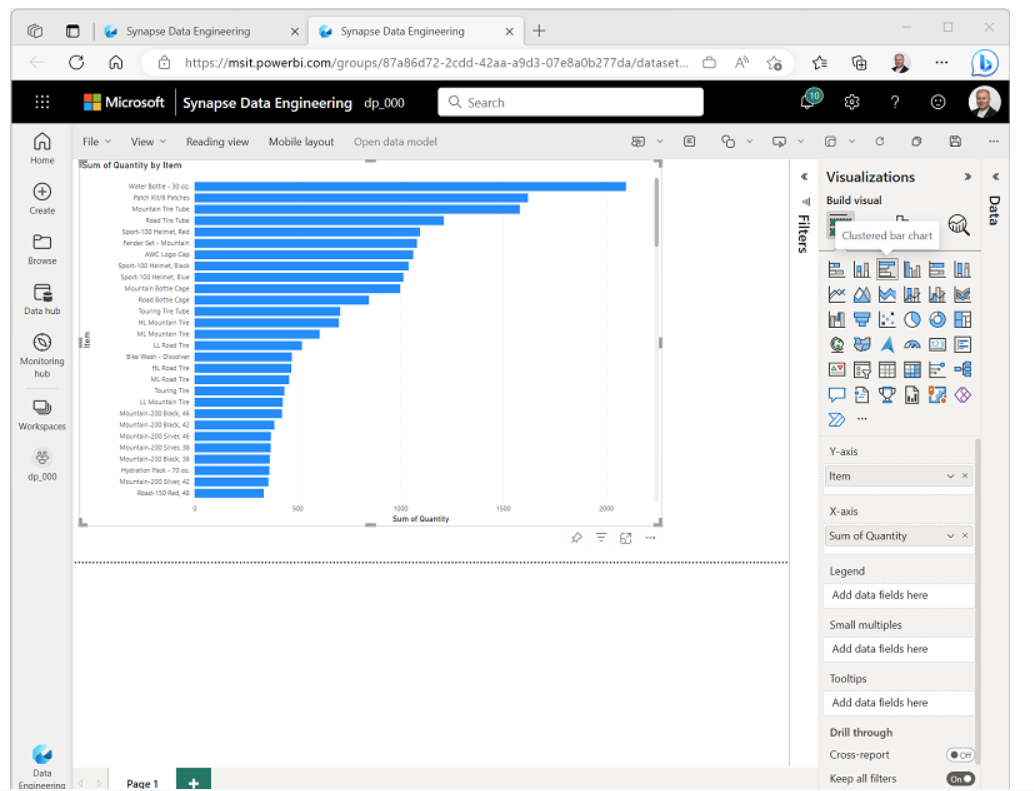
1. In the **Data** pane on the right, expand the **sales** table. Then select the following fields:

* **Item**
* **Quantity**

A table visualization is added to the report:



1. 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.



1. On the **File** menu, select **Save**. Then save the report as **Item Sales Report** in the workspace you created previously.
2. 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 analytics endpoint for your lakehouse.
   * A default semantic model 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 semantic model 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**.

**Summary**

Microsoft Fabric lakehouses provide data engineers and analysts with the combined benefits of data lake storage and a relational data warehouse. You can use a lakehouse as the basis of an end-to-end data analytics solution that includes data ingestion, transformation, modeling, and visualization.

**Tip :**

To learn more about lakehouses in Microsoft Fabric, see the [**What is a lakehouse?**](https://learn.microsoft.com/en-us/fabric/data-engineering/lakehouse-overview) reference documentation.

