END TO END DATA ANALYTICS SOLUTION USING MICROSOFT FABRIC WITH WWI DATASET.

A step-by-step guide to ingest, transform, model, and visualize business data using the Lakehouse paradigm.

Gosala Kasi Raju
kasirajugosala@outlook.com

LAKEHOUSE END TO END SCENORIO

PROJECT OVERVIEW

Implement a full end-to-end data analytics solution using Microsoft Fabric Lakehouse to ingest, transform, store, and visualize data from the **Wide World Importers (WWI) dataset.**

PROJECT PURPOSE

To demonstrate how organizations can use Microsoft Fabric to build scalable, collaborative analytics solutions with minimal infrastructure setup, integrating raw data ingestion, Delta Lake storage, semantic modeling, and Power BI reporting.

PREREQUISITES

1. **Environment:**
   - Power BI account with Microsoft Fabric Trial enabled.
   - A Fabric-enabled workspace created in Power BI.
2. **Tools Required:**
   - Lakehouse – for storing raw and transformed data.
   - **Dataflows Gen2** – to ingest .csv files.
   - **Pipelines** – to automate ZIP data ingestion.
   - **Notebooks** – for data transformation using PySpark or SQL.
   - **Power BI Reports** – for visualizing the data.
   - **SQL Analytics Endpoint** – to run SQL queries.
3. **Sample Data:**
   - `dimension_customer.csv`:
     Download CSV
   - `wwi-sample-dataset.zip`:
     Download ZIP

PROJECT STEPS

- **STEP-1: Environment Setup**
  - Create a Fabric – enabled workspace
  - Create a Workspace named as Project WS1
  - Create a new Lakehouse (wwiLakehouse) in **Project WS1**
- **STEP-2: Ingest Initial Data (CSV)**

- Use Dataflow Gen2 to load dimension_customer.csv
- Transform : Promote headers, save, and load as table.

➢ **STEP-3: Ingest Full Dataset (ZIP)**

- Create a Pipeline to ingest wwi-sample-dataset.zip via HTTP.
- Enable binary copy and extract ZIP into the Lakehouse Files.
- Rename the imported folder to wwi-raw-data.

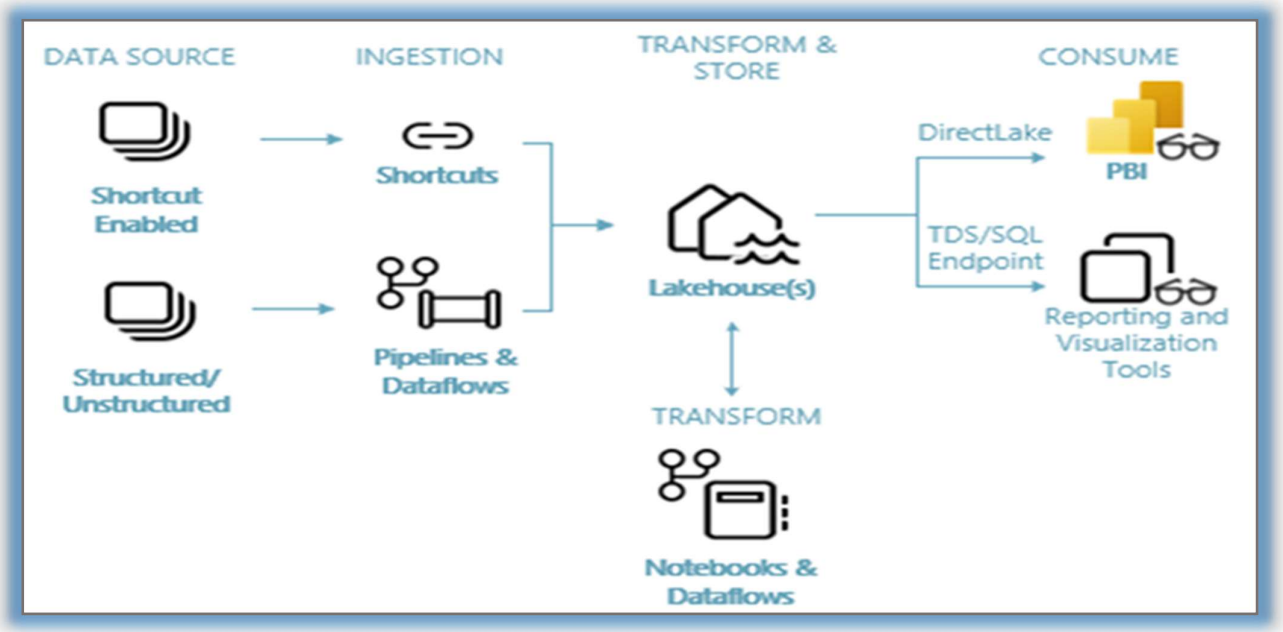
➢ **STEP-4: Data Transformation**

- Import and run Notebook 01 – Create Delta Tables:
  - Read raw files, transform, write to Delta tables.
- Run Notebook 02 – Transformation – Business**:**
  - Create business aggregates using PySpark or SQL

➢ **STEP-5: Build Report**

- Add tables to semantic model via SQL endpoint.
- Enable sync with power BI semantic model.
- Use Auto-Create Report in Power BI and save**.**

ARCHITECTURE DIAGRAM

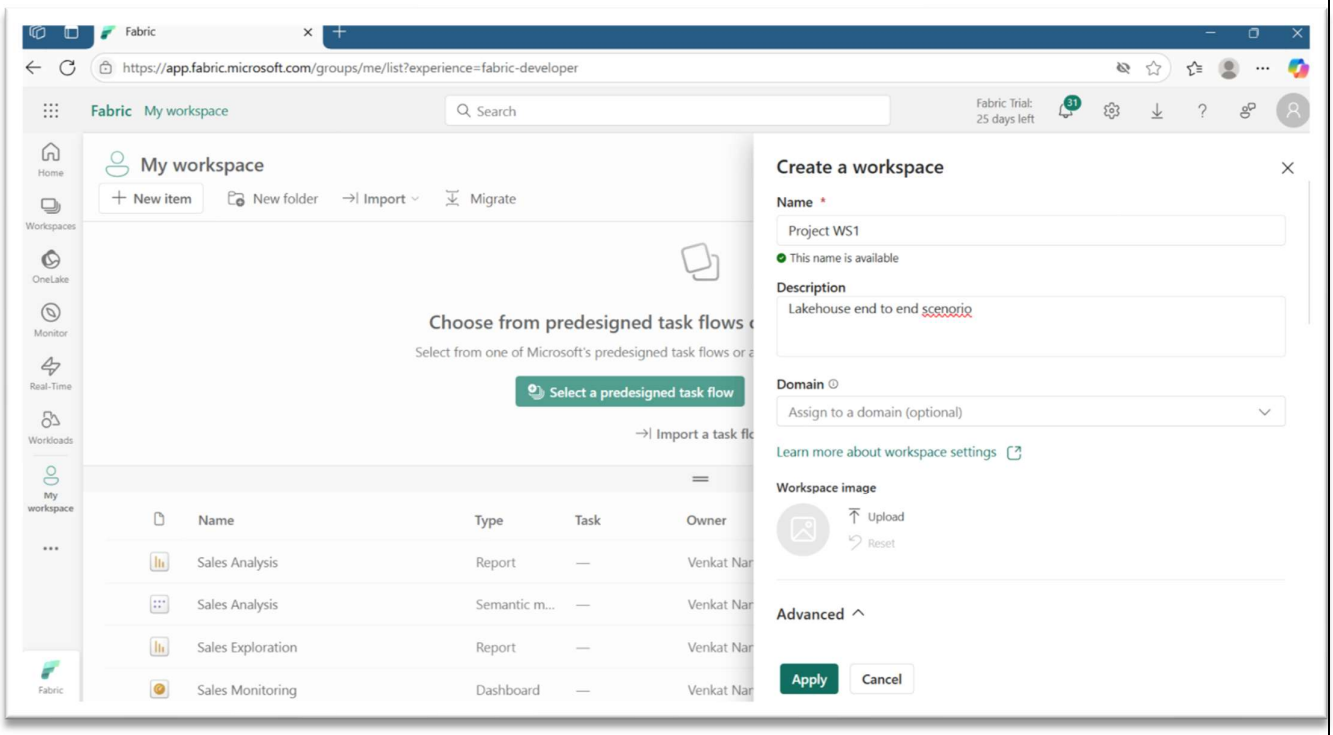


THE PROCCESS OF AN ENTIRE IMPLEMENTATION

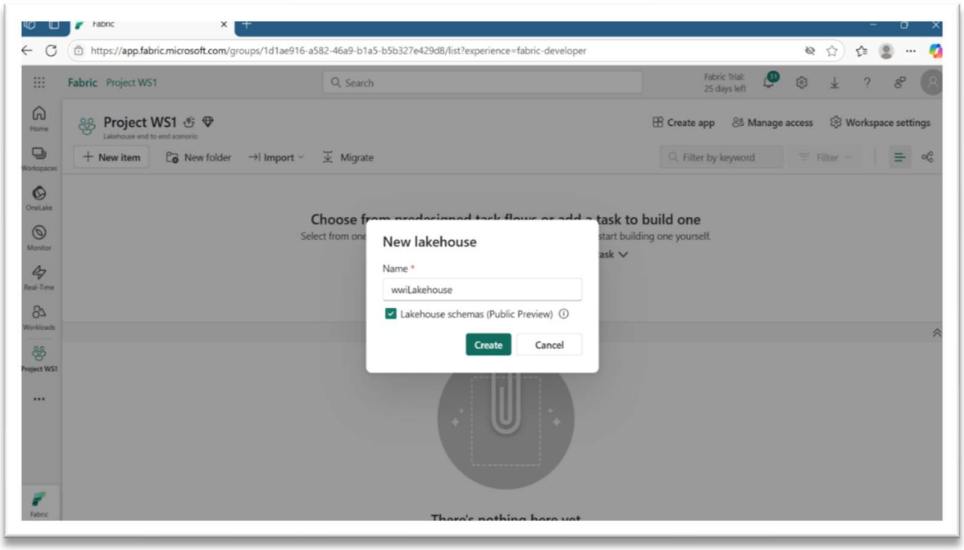
STEP – 1

✓ Create a workspace named **Project WS1.**

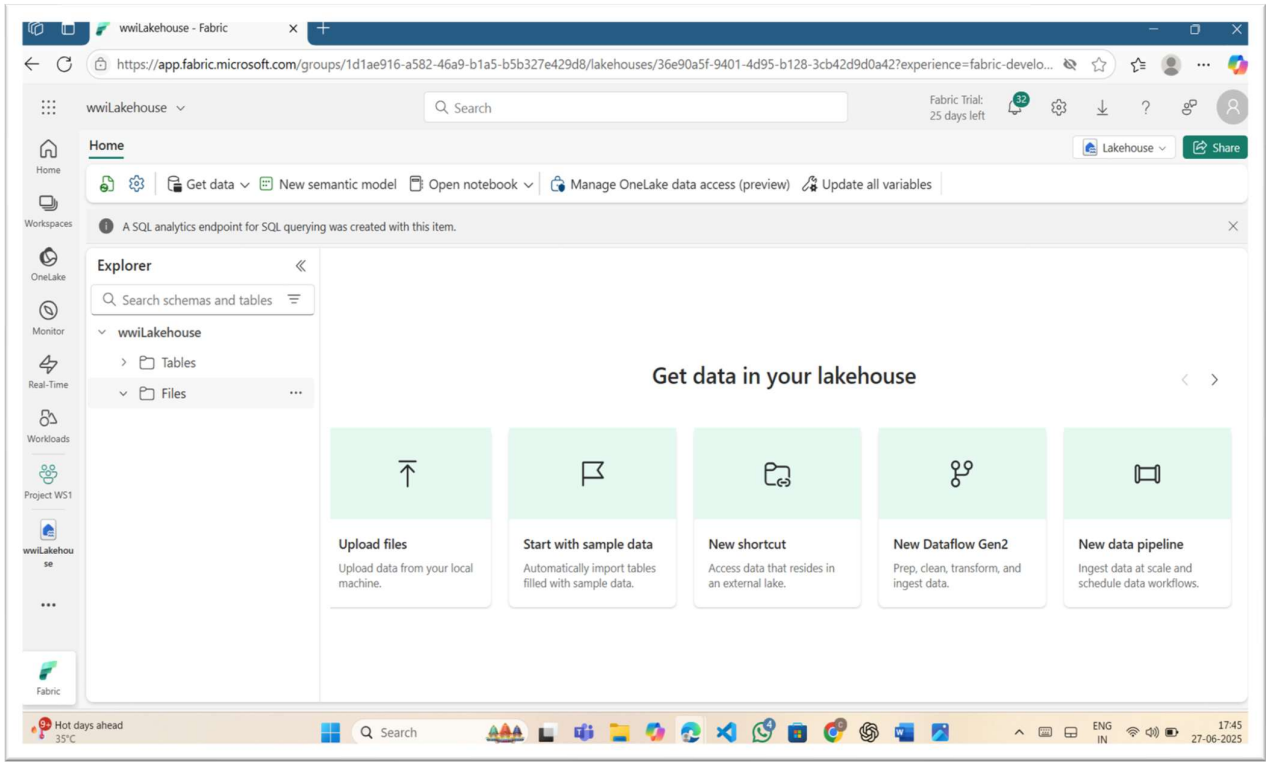
- ✓ Select License Mode – **Trail**
- ✓ Select Semantic Model Storage Format – **Small SMSF**
- ✓ Click **Apply.**



- ✓ Open the **Project WS1** workspace. From the workspace, select **New item,** then choose **Lakehouse.**
- ✓ In the **New lakehouse** dialog box, enter **wwiLakehouse** in the name field.
- ✓ Click **Create** to open the Lakehouse.

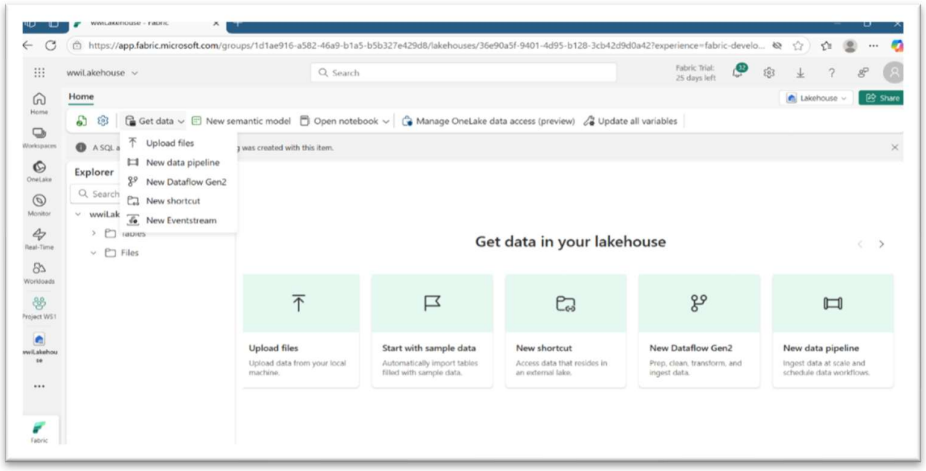


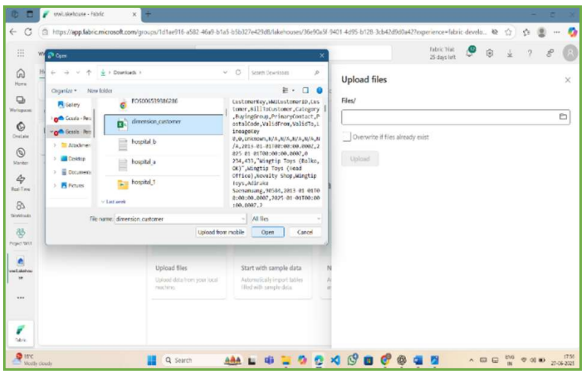
✓ **T**he Lakehouse has already been created, as shown in the figure below. It includes the setup to ingest sample data and build a report.

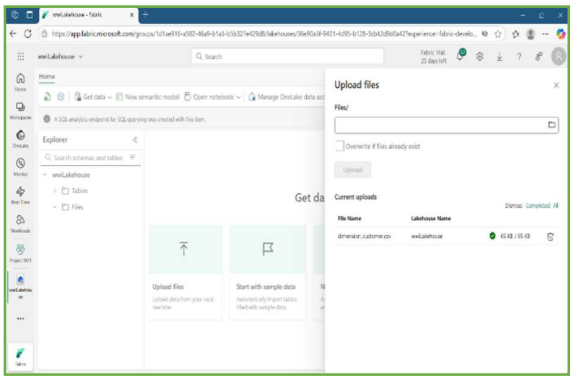


➢ INGEST SAMPLE DATA

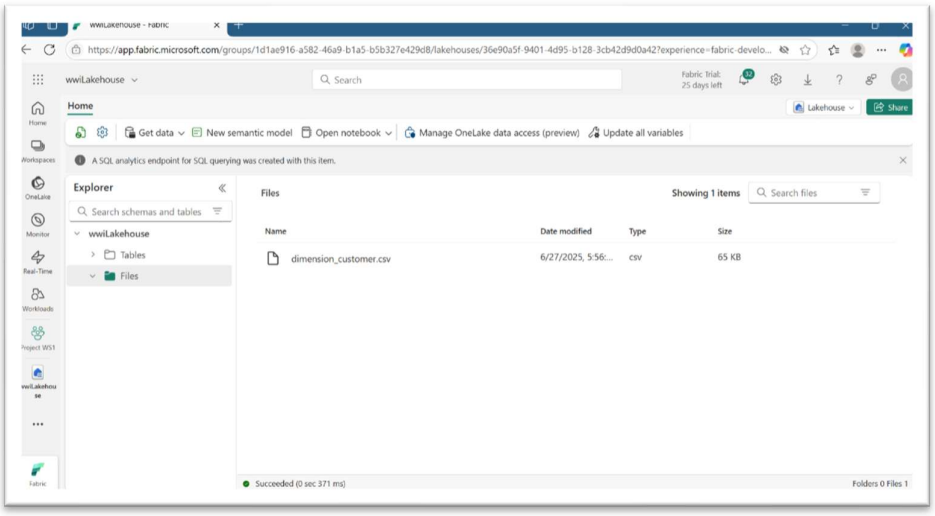
- Download the *dimension_customer.csv* file from the Fabric sample repository.
- In the Lakehouse Home panel Select the **GET DATA** and Explore pane. Click **Upload files** ,then browse and select the *dimension_customer.csv* file from the local system folder.



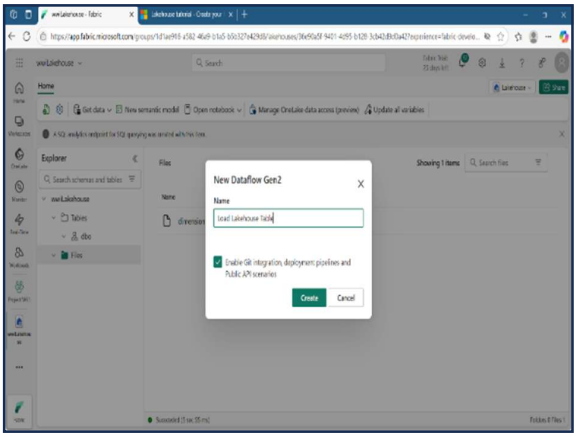


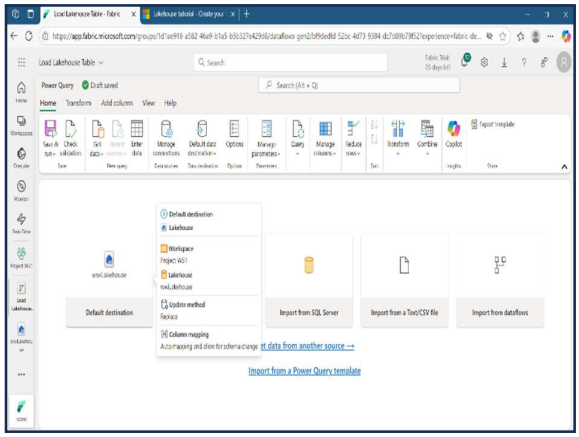


- After the file is successfully uploaded to the lakehouse, it will appear as shown below.

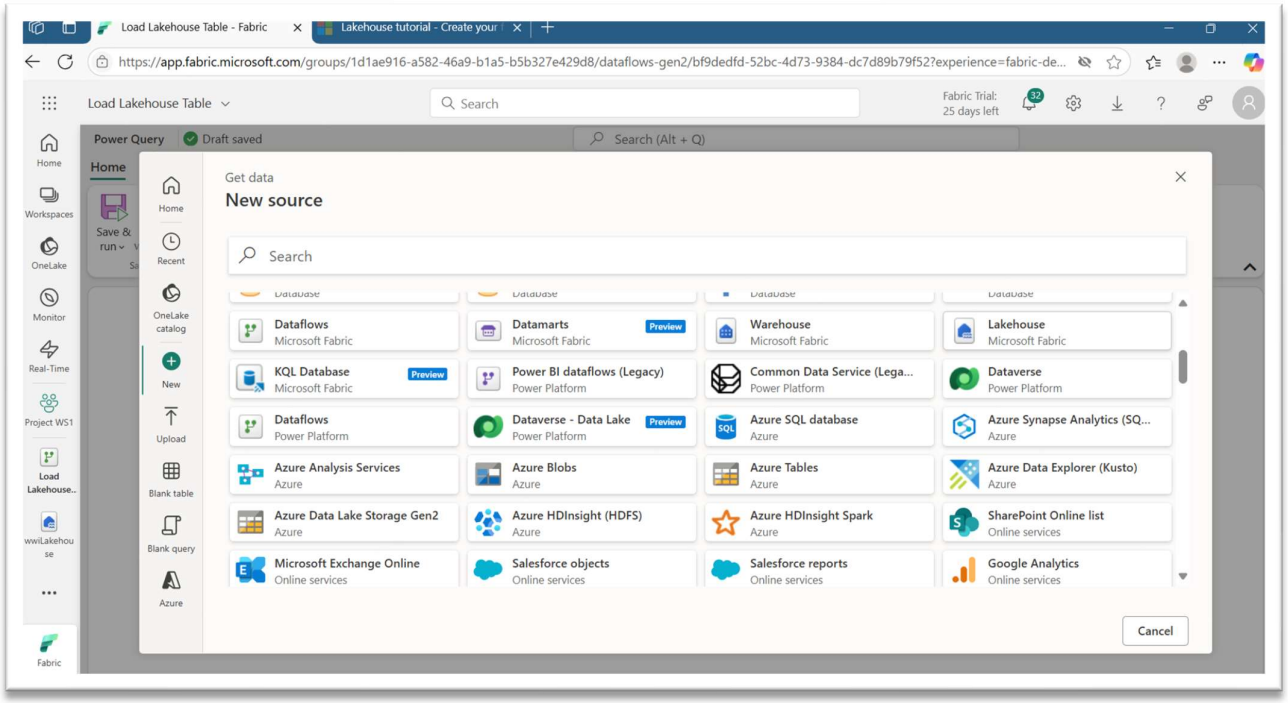


- Now Select **GET DATA** again and Explore the pane. Then select **the NEW DATAFLOW GEN2** and Create it with name **Load Lakehouse Table**.

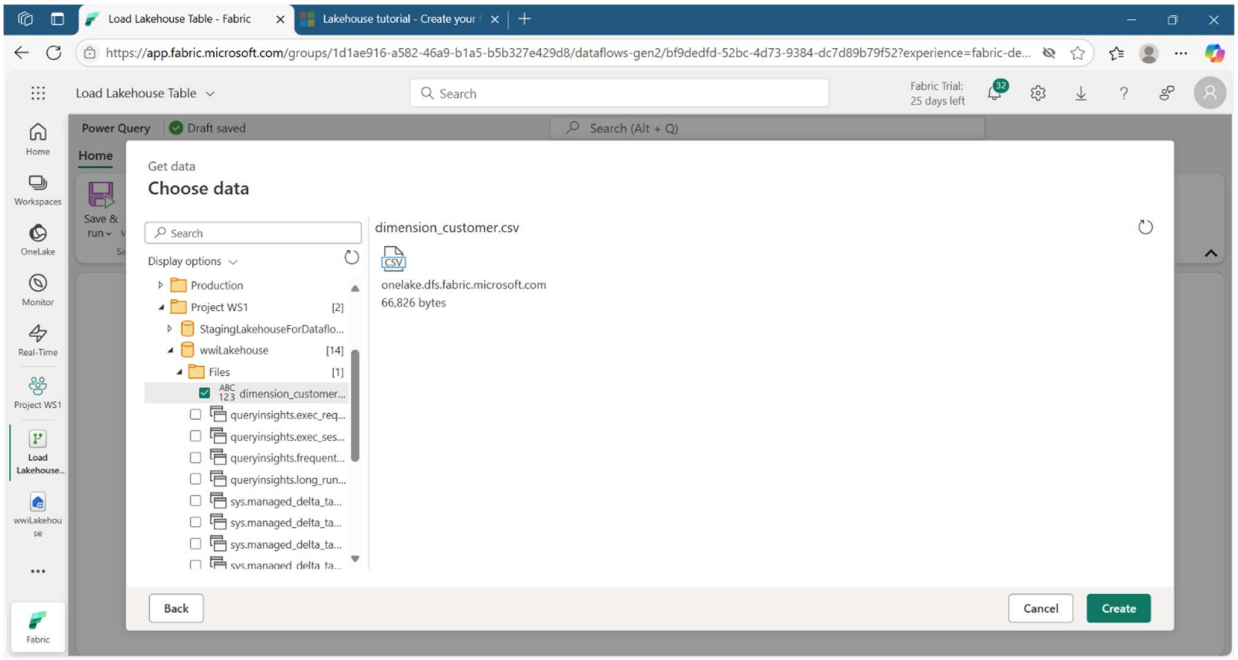




- The Dataflow Gen2 Opens with the Default Destination **(wwiLakehouse).**
- Now Select **GET DATA** from the pane , expand it , and click on View more. Then, from the **New source** options, select **Lakehouse**.

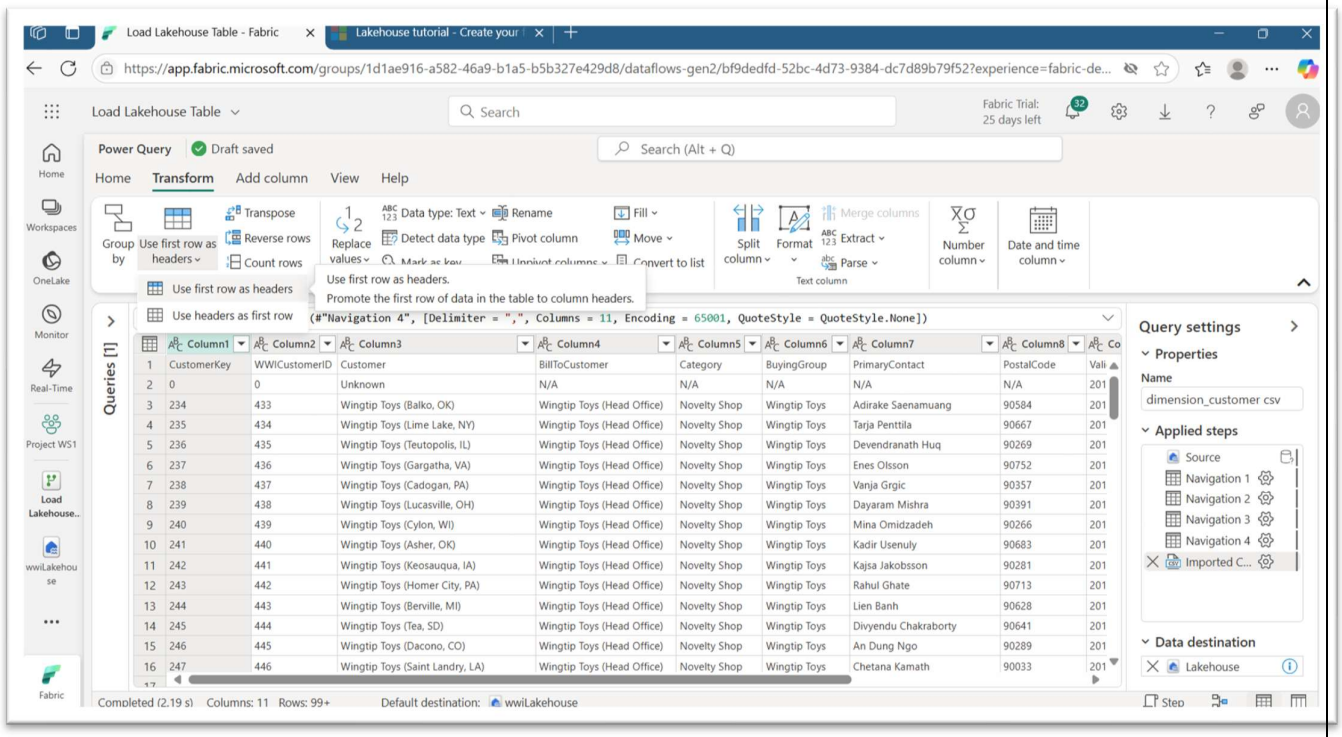


- After selecting the Lakehouse and setting the connection credentials, click Next to proceed.
- After that, we need choose the data from the display options , Select **Project WS1 > wwiLakehouse > Files > *dimension_customer.csv*** , then click **Create**.

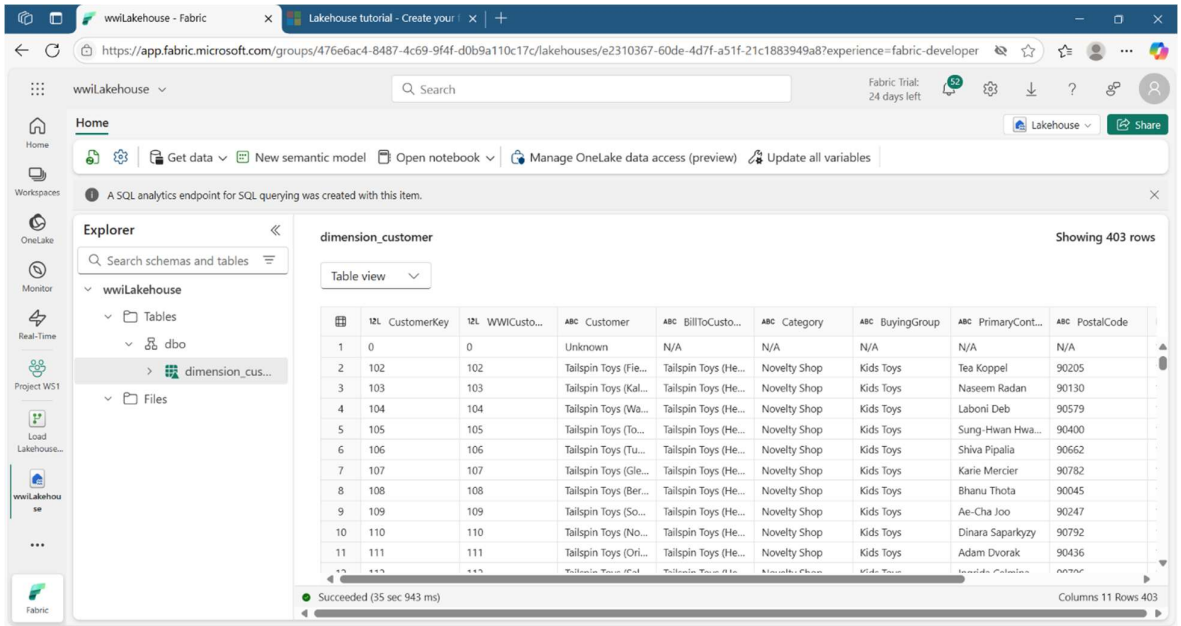


- After Creating it, open the preview of the table.. On the left side of table, check the **Query Settings** and **Properties,** and ensure the name is set to *dimension_customer.*

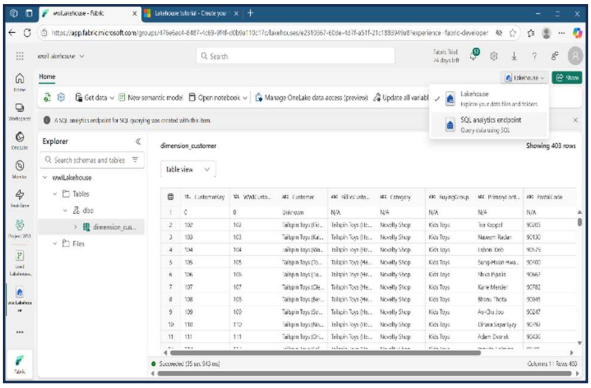
- Now select the Transform from the Home Pane. Choose **Use first row as headers**. This is Transforms the dataset into a proper table with correct headers. Then, save and run it.

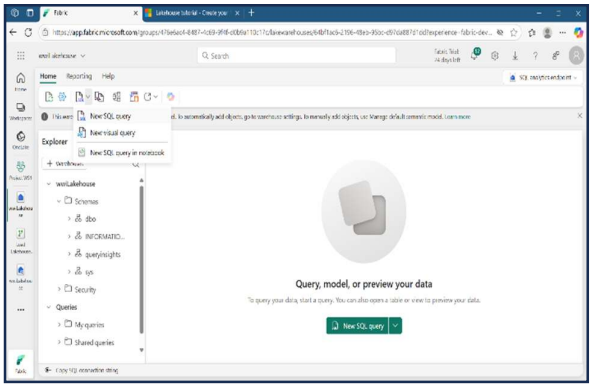


- Open the Lakehouse (wwiLakehouse) and refresh it. Then, select and expand the *dimension_customer* table to view its structure.

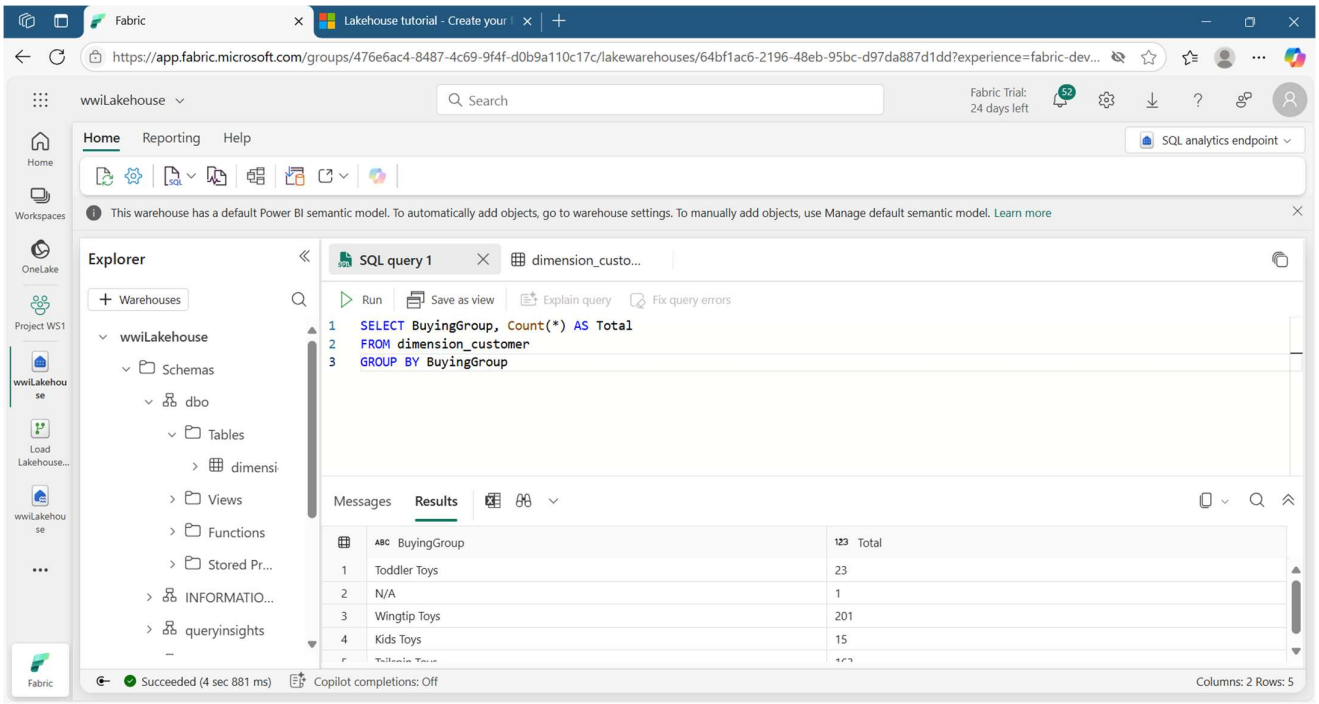


- Select the **SQL analytics endpoint** from the lakehouse, then click **New SQL Query** from the Home pane.

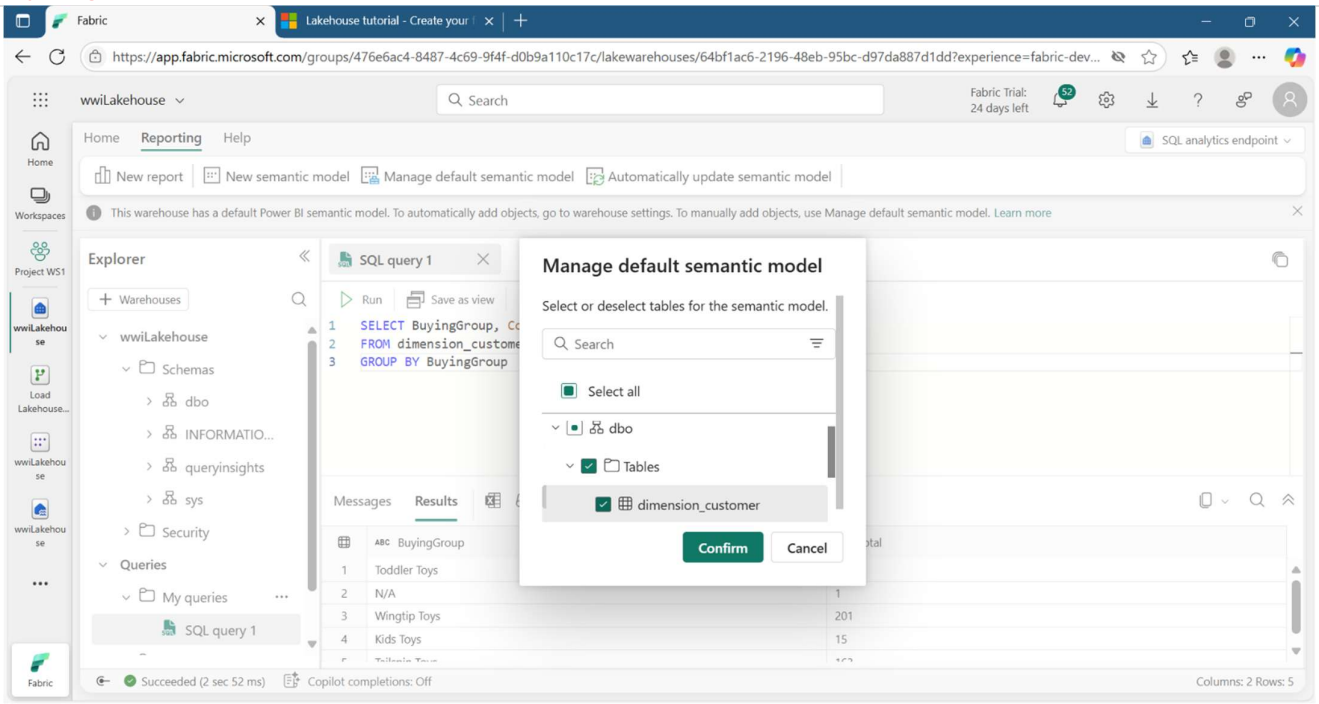




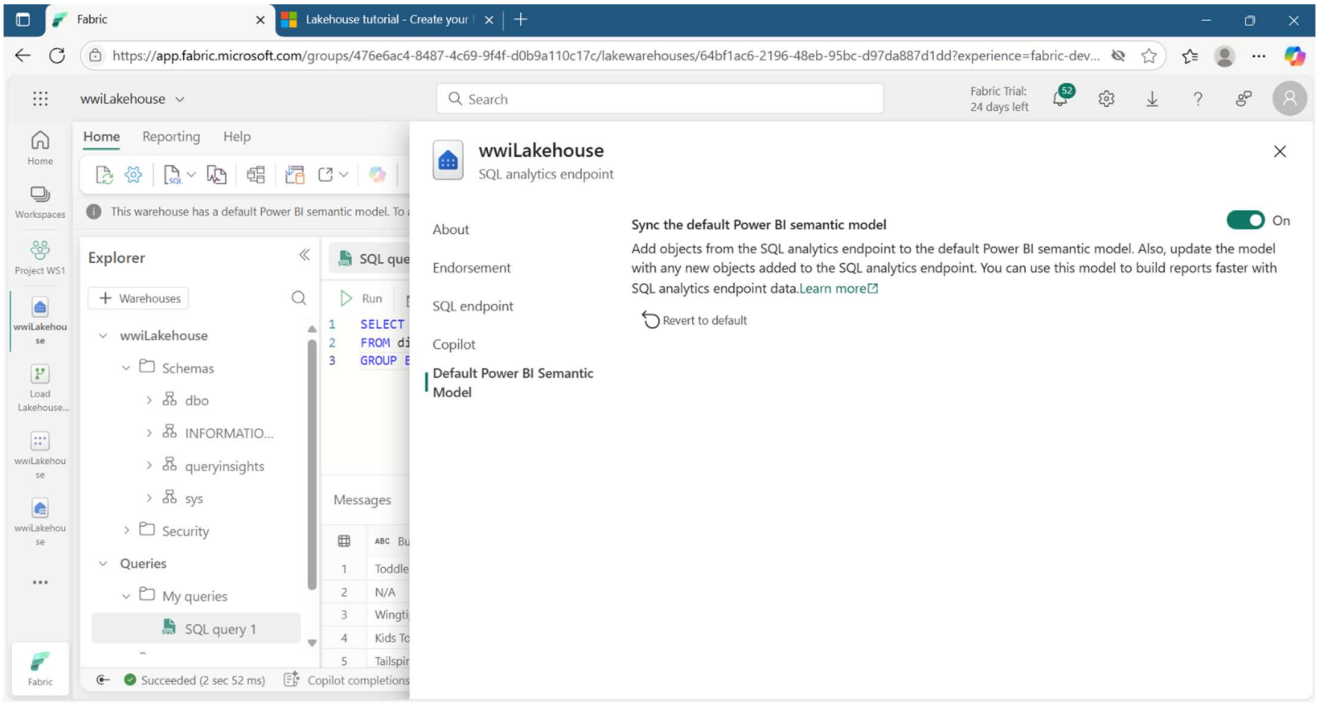
- Run the SQL query to validate the table and confirm successful ingestion.



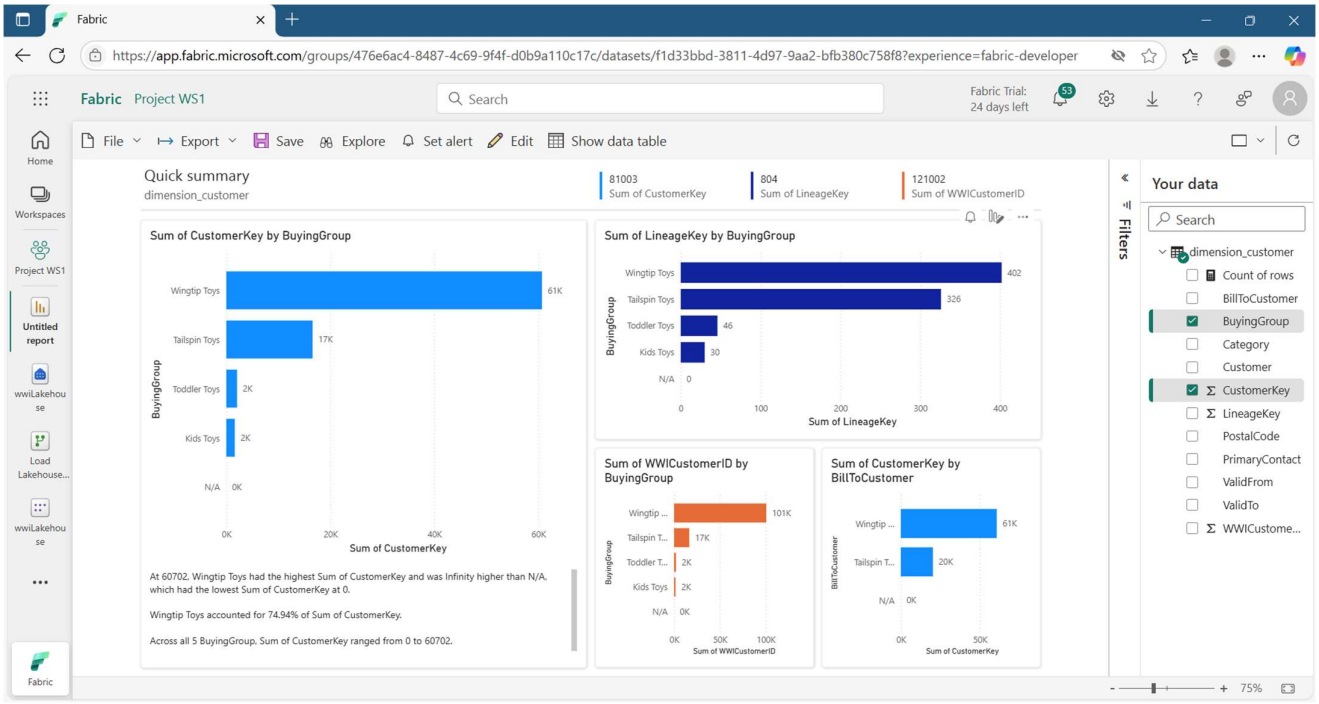
- Open the Lakehouse and switch to the **SQL analytics endpoint** view. From the **Reporting** tab, select **Manage default semantic model** and choose the tables that wants to include. In this case, select the ***dimension_customer*** table.



- To ensure the tables in the semantic model stay in sync, switch to the **SQL analytics endpoint** view and open the Lakehouse **settings** pane.
  Select **Default Power BI semantic model**, then enable the **Sync the default Power BI semantic model** option.

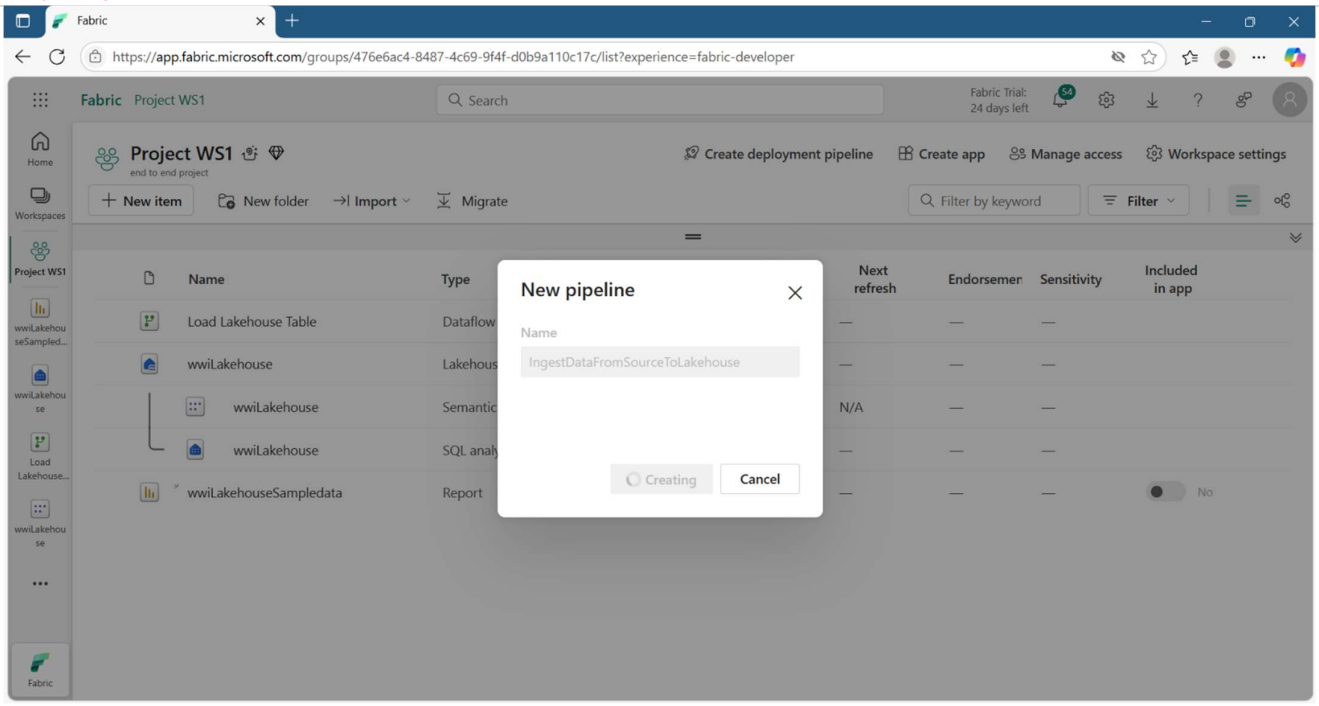


- After the table is added, Fabric creates a semantic model with the same name as the lakehouse.
- Let Power BI automatically create a report based on your data. Under **Explore this data**, select **Auto-create a report**.
- Because the table is a dimension and contains no measures, Power BI creates a row count measure. It then uses this generate various charts, as shown in the figure below. **Save** the Report.

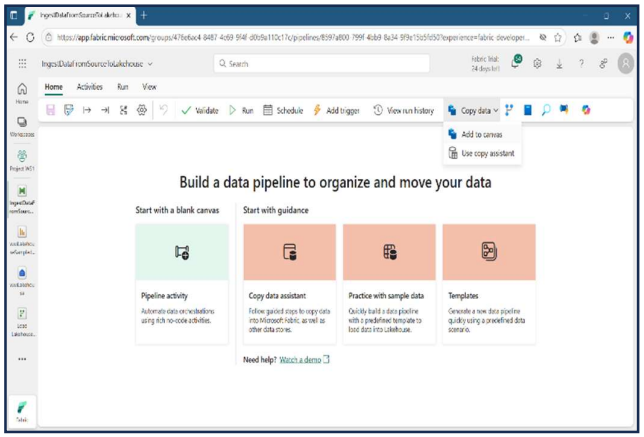


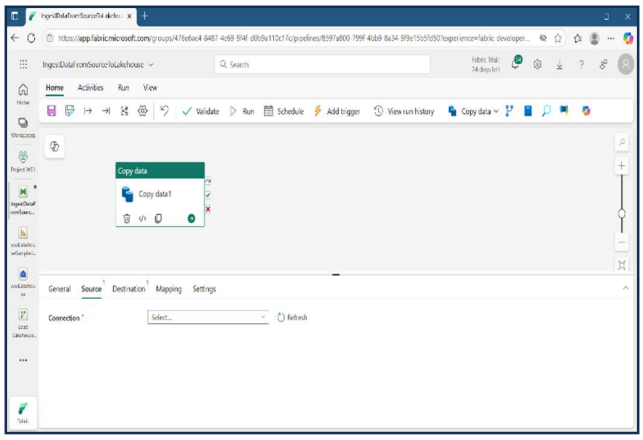
INGEST DATA INTO THE LAKEHOUSE:

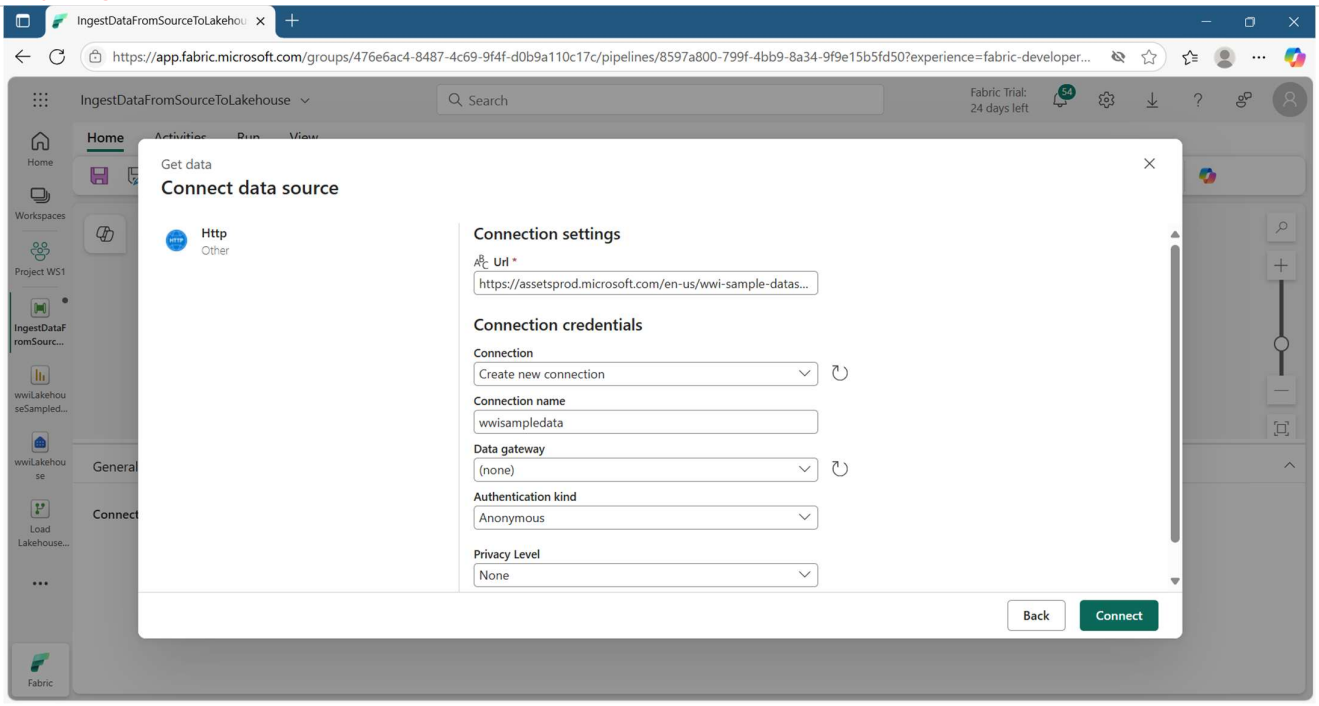
- Open the workspace (Project WS1), select the **DATA PIPELINE** from the **New item.**
- Specify the name as **IngestDataFromSourceToLakehouse** and select create. A new data factory pipeline is created and opened.



- Next, set up an HTTP connection to import the sample World Wide Importers data into the Lakehouse. From the **Copy data** list **,** select **Add to canvas**. Then open **Source and Destination**, choose **Source,** click **View more**, search for **Http** and select it.



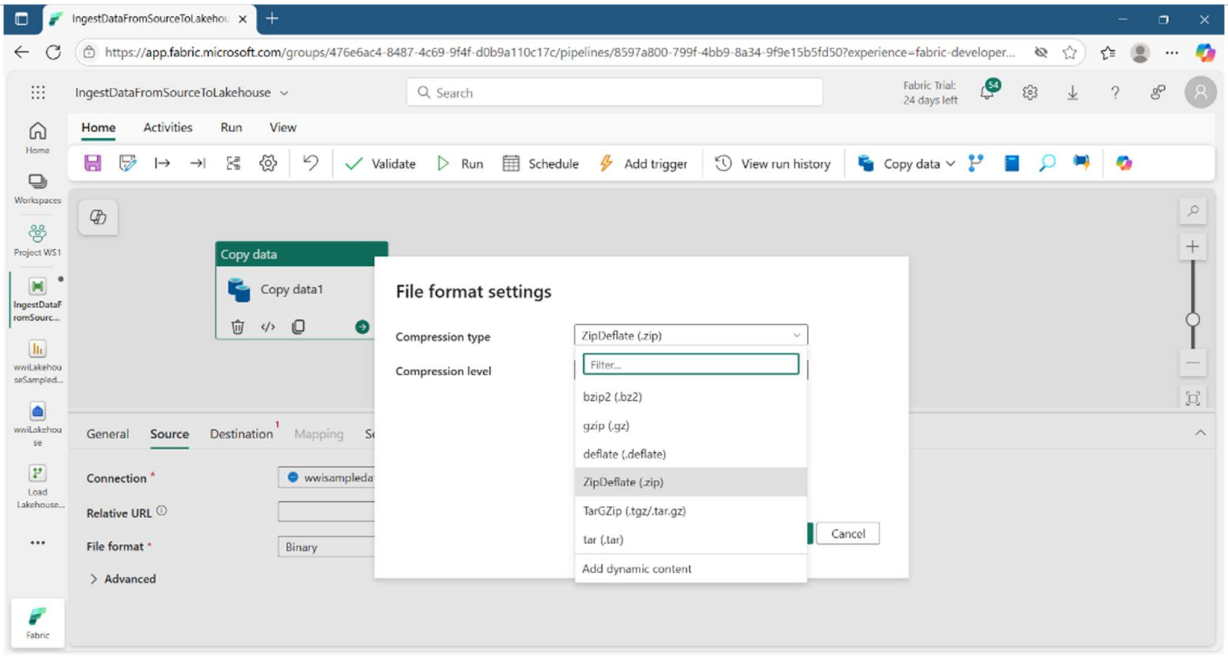


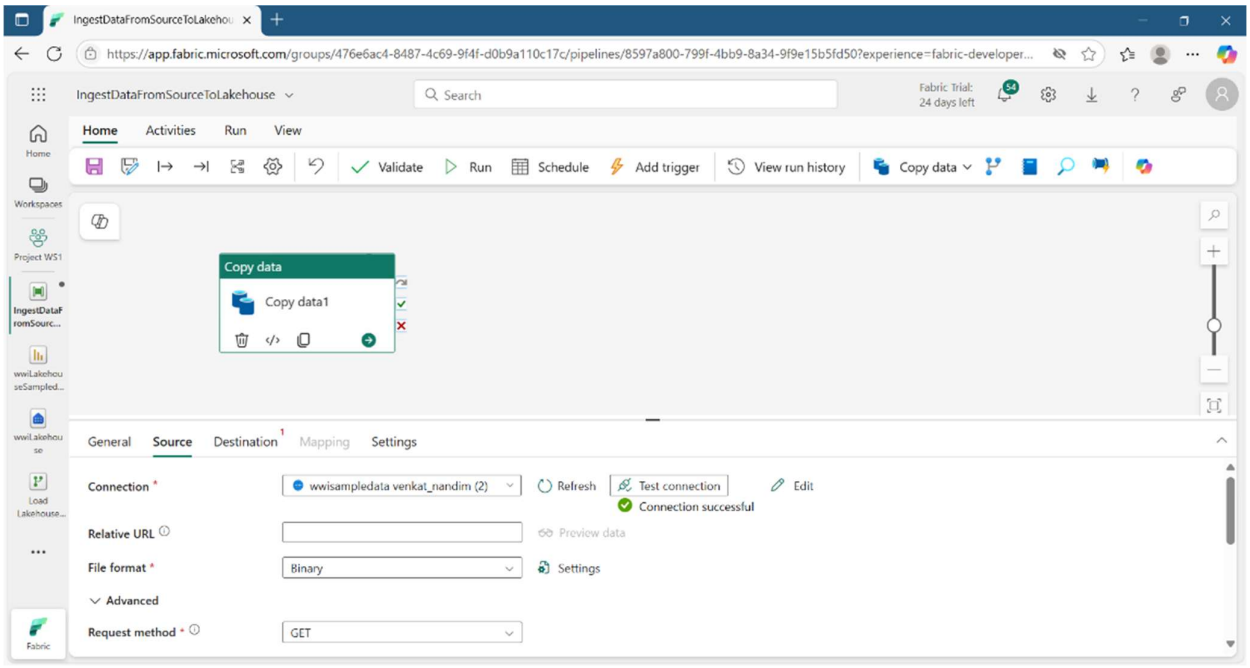


- In the **Connect to Data Source** window, enter the following details.

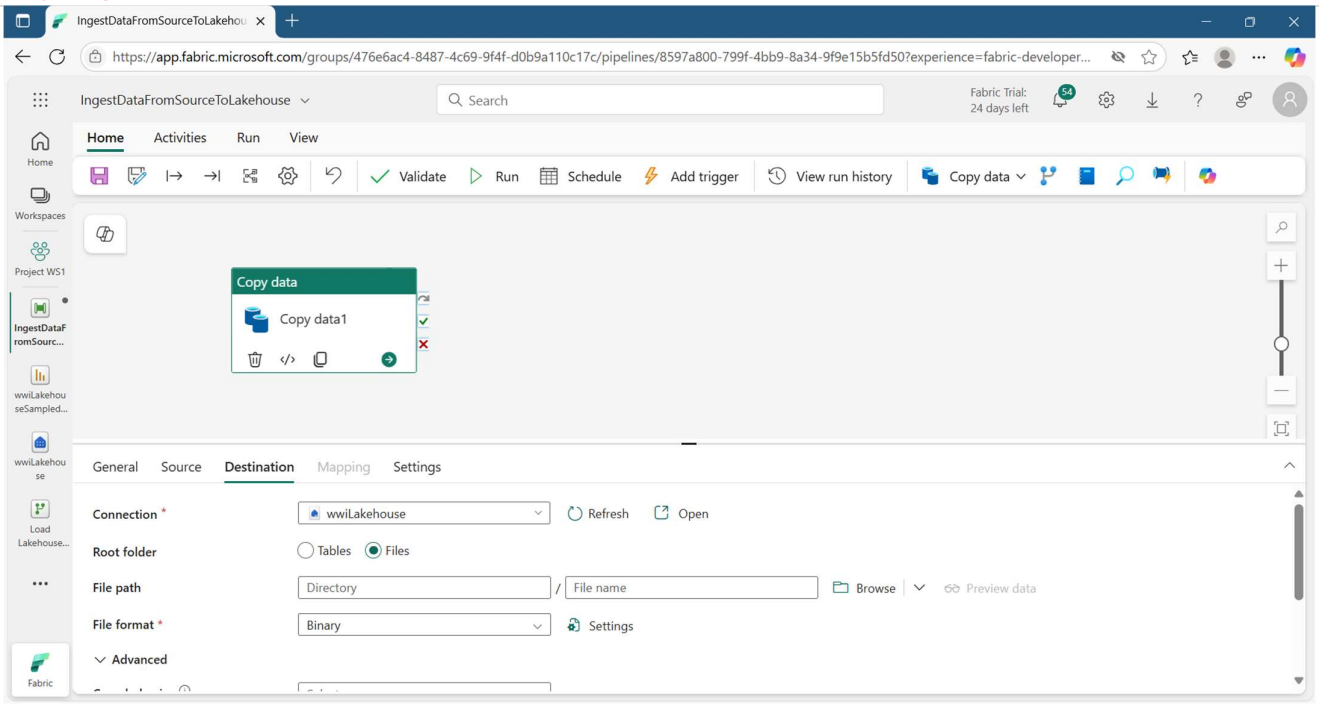
Property	Value
URL	`https://assetsprod.microsoft.com/en-us/wwi-sample-dataset.zip`
Connection	Create a new connection
Connection name	wwisampledata
Data gateway	None
Authentication kind	Anonymous

- In the next step, enable **Binary copy** and choose **ZipDeflate (.zip)** as the **Compression type**, since the source is a ZIP file. Keep the other fields at their default values, then click Next. Finally, click **Test Connection to ensure it is successful.**

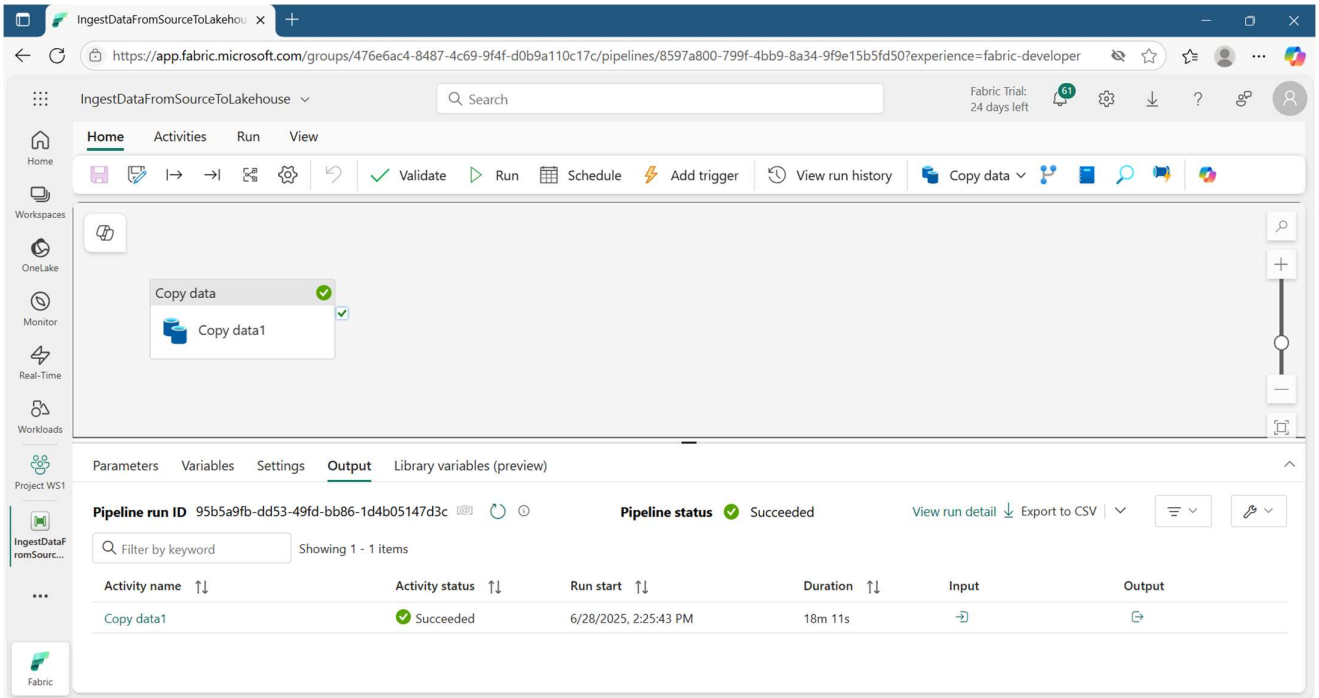




- In the **Connect to data destination** window, specify **Files** as the root folder and click Next. This will write the data to the Files section of the Lakehouse.

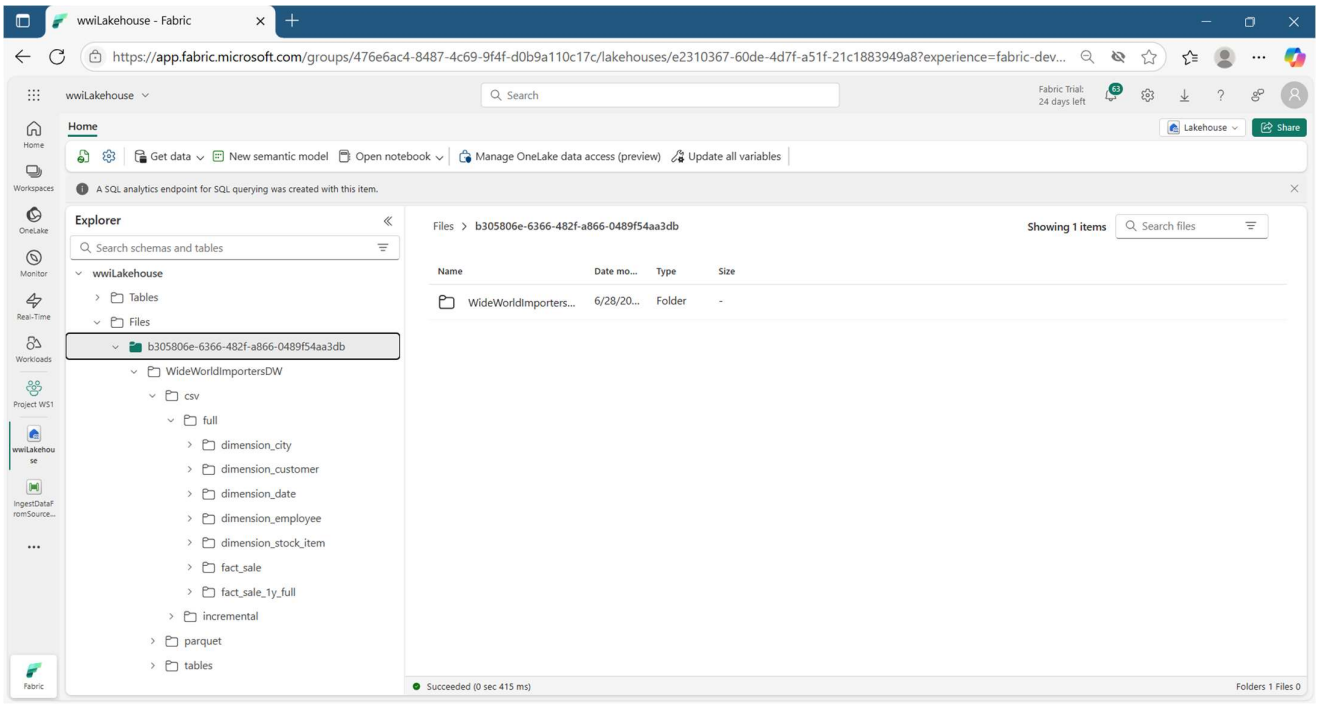


➢ Next, click **Save and Run** .After approximately 20 minutes, the output will be displayed.

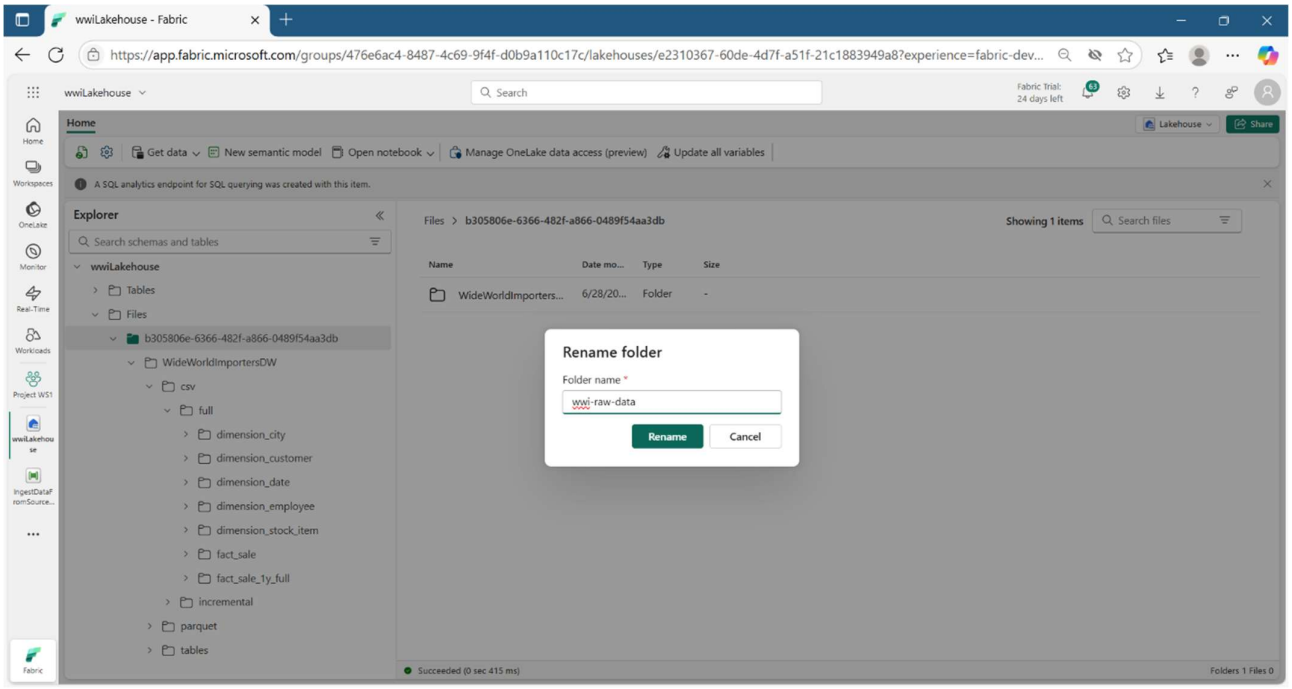


➢ After the successful execution of the pipeline, go to your Lakehouse (**wwiLakehouse**) and open the Explorer to view the imported data.

- Verify that the folder **WideWorldImportersDW** is present on the Explorer view and contains data for all tables.

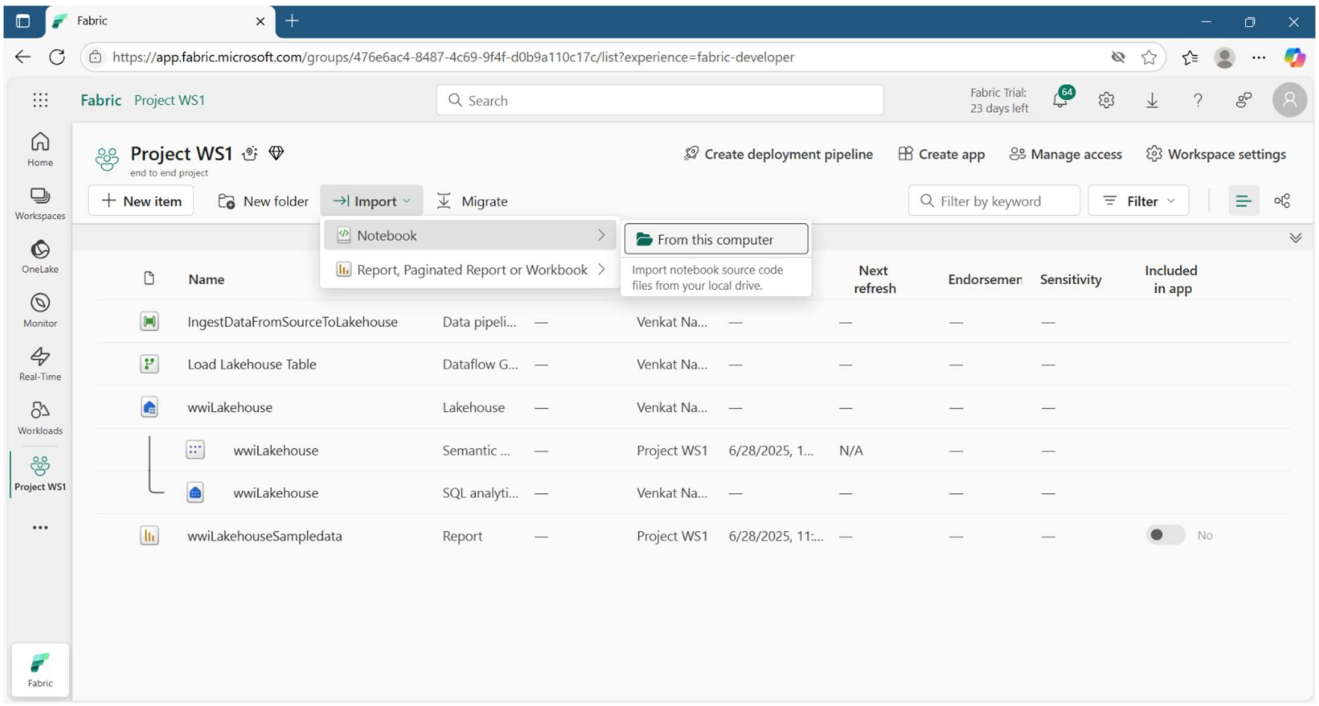


- The data is created under the **Files** section of the Lakehouse Explorer. A new folder with a **GUID** is generated, which contains all the required data. Rename this folder to **wwi-raw-data.**

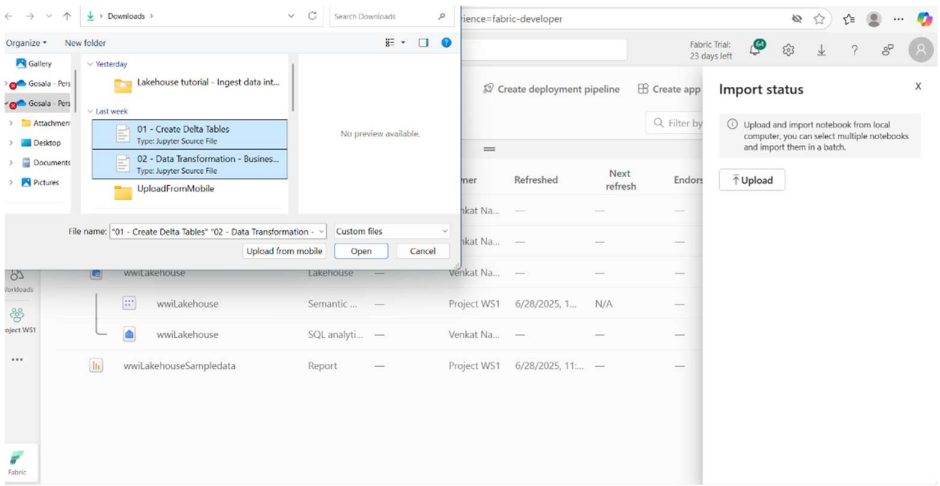


PREPARE AND TRANSFORM DATA IN THE LAKEHOUSE

- Download the notebooks from the Lakehouse Tutorial Source Code folder.
- From the workspace, select **Import** > **Notebook** > **From this computer.**

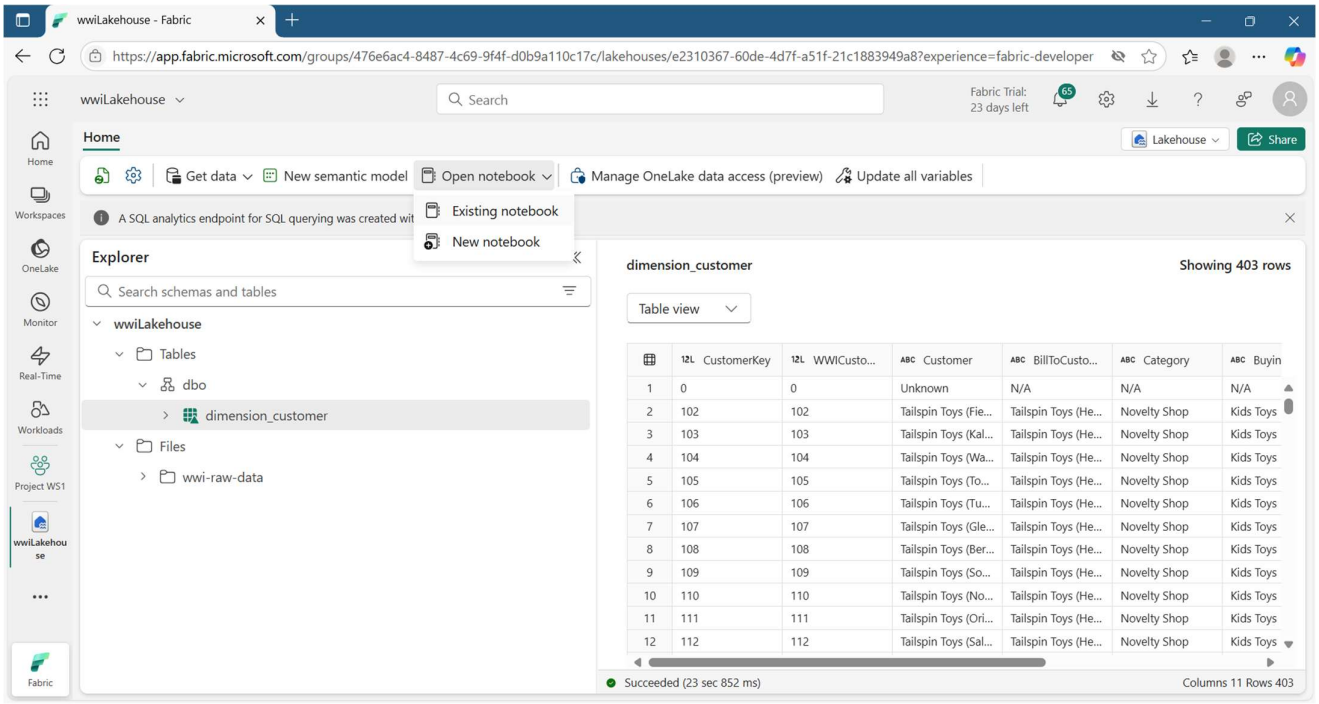


- Select **Import Notebook** from the **New** section at the top of the landing page.
- Select **Upload** from the **Import status** pane that opens on the right side of the screen.

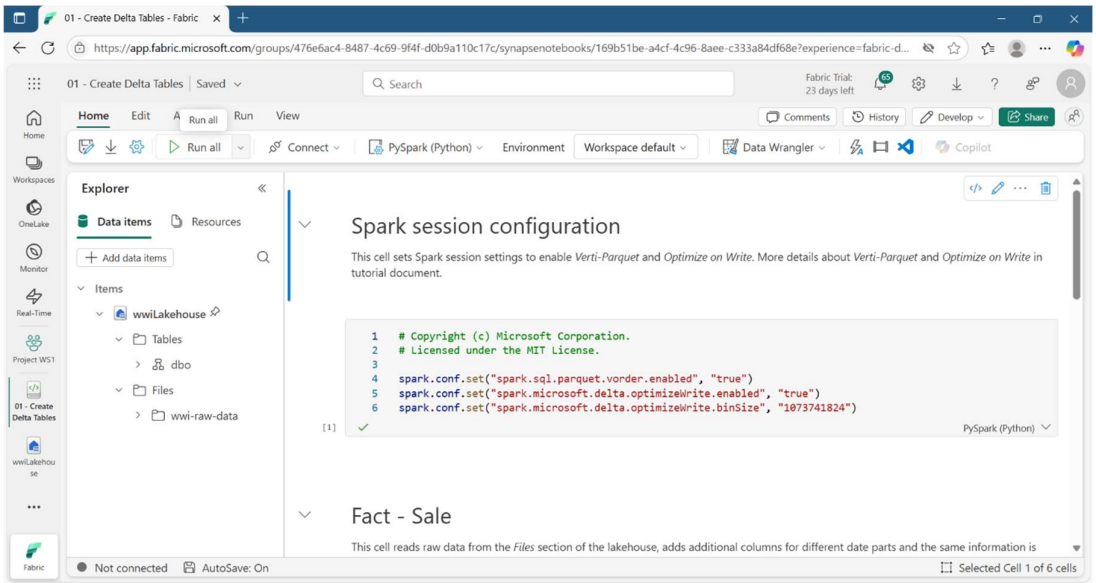


- Select both notebooks and click **Open.**

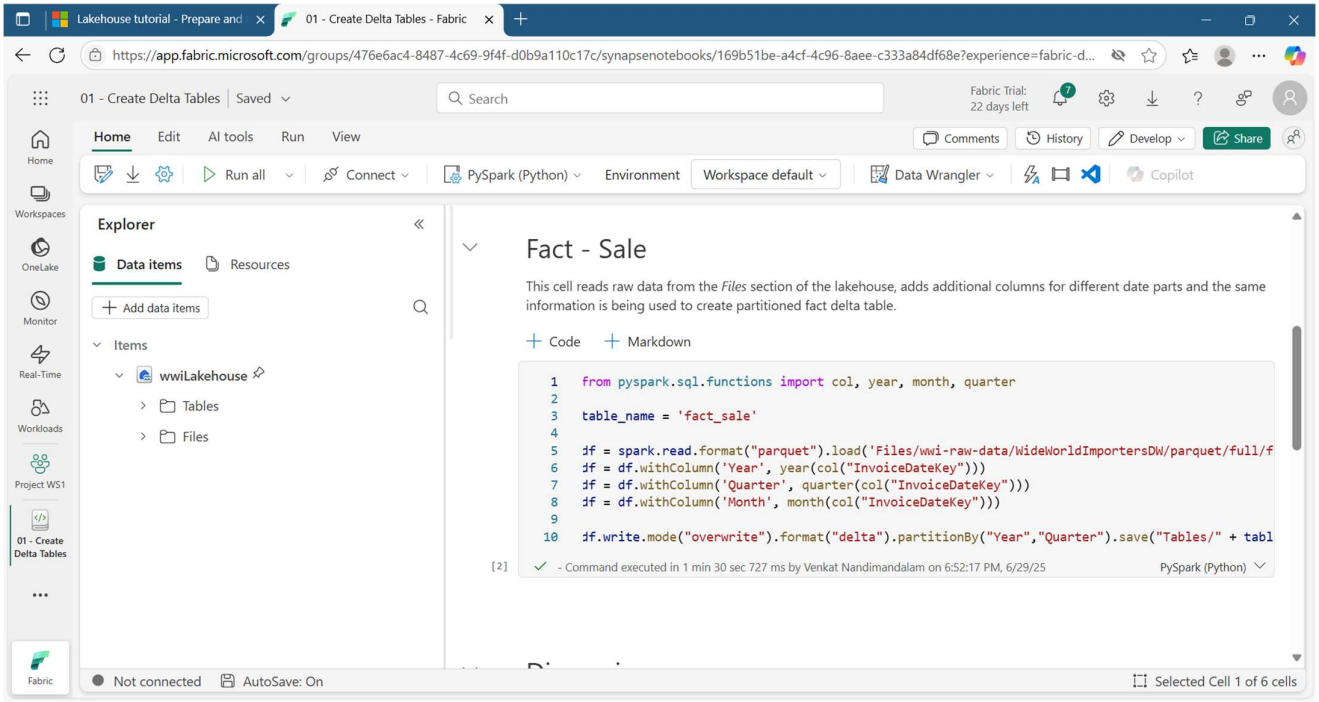
- After the import is successful, go to the **items** view of the workspace to see the newly imported notebooks. Then select **wwiLakehouse** to open it.
- Once the wwiLakehouse lakehouse Is opened, select **Open notebook** > **Existing notebook** from the top menu.



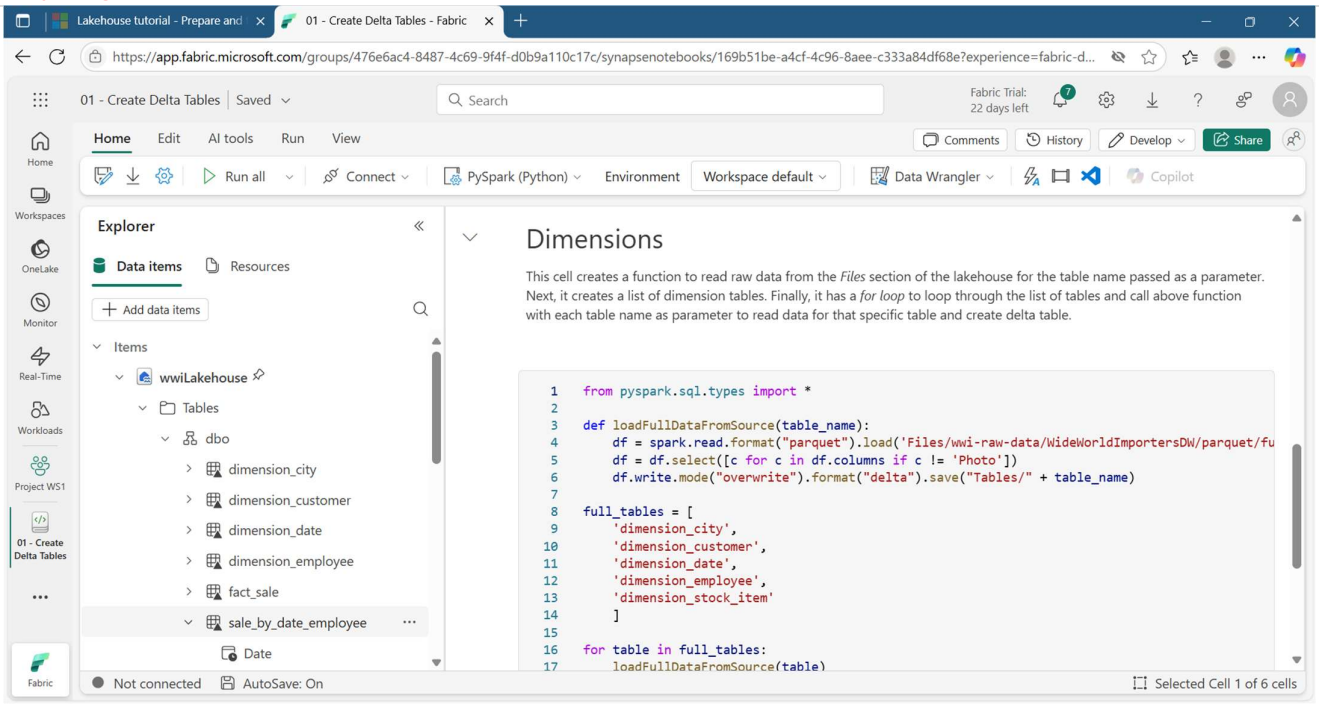
- From the **Existing Notebook** section, select the **01- Create Delta Tables,** then click **Open.**
- In the list of existing notebooks in the Lakehouse Explorer, you'll see the notebook is already linked to the opened Lakehouse.



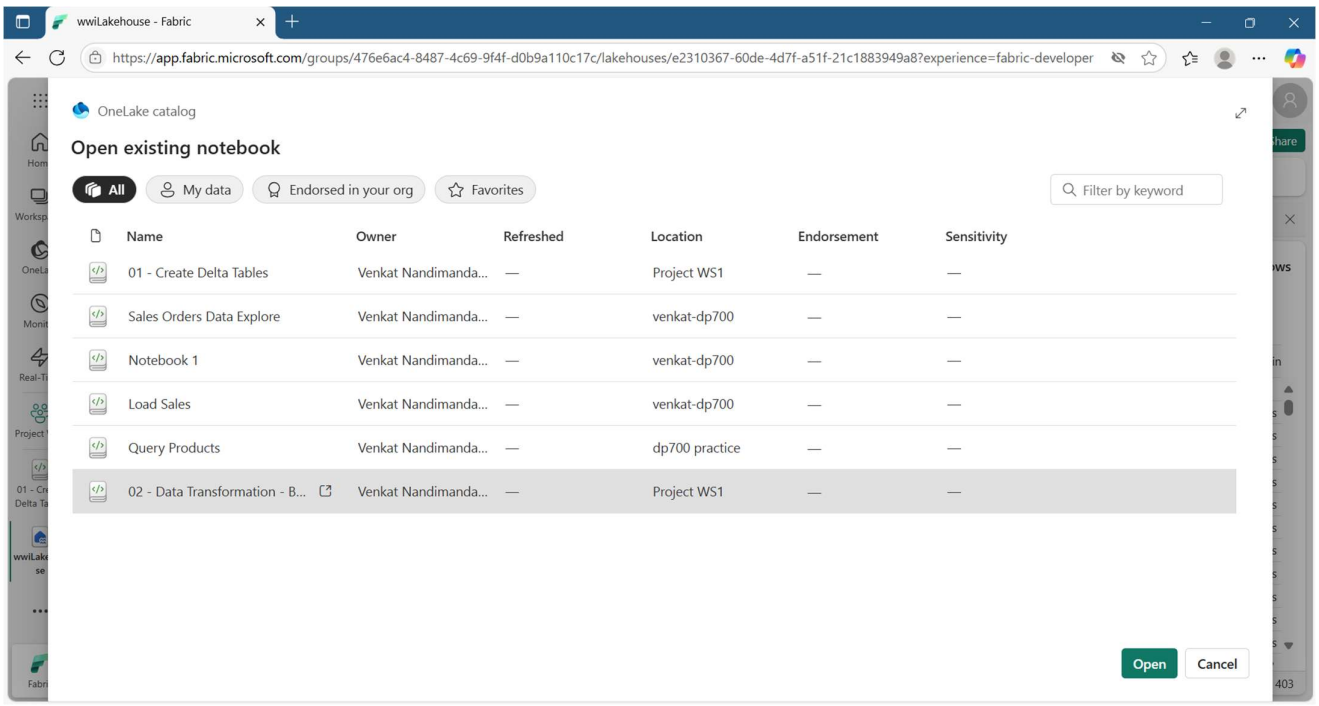
- Before writing data as Delta Lake tables in the **Tables s**ection of the Lakehouse, we enable two Fabric features : **V-order** and **Optimize Write**.
- To start the notebook and execute all the cells in sequence, click **Run All** from the **Home** ribbon.
- Next, read raw data from the **Files** section of the Lakehouse, and add additional columns for various date parts as part of the transformation. Finally, use Spark API to partition the data before writing it in Delta table format, based on the newly created data columns (Year and Quarter). **Fact_Sale Table.**



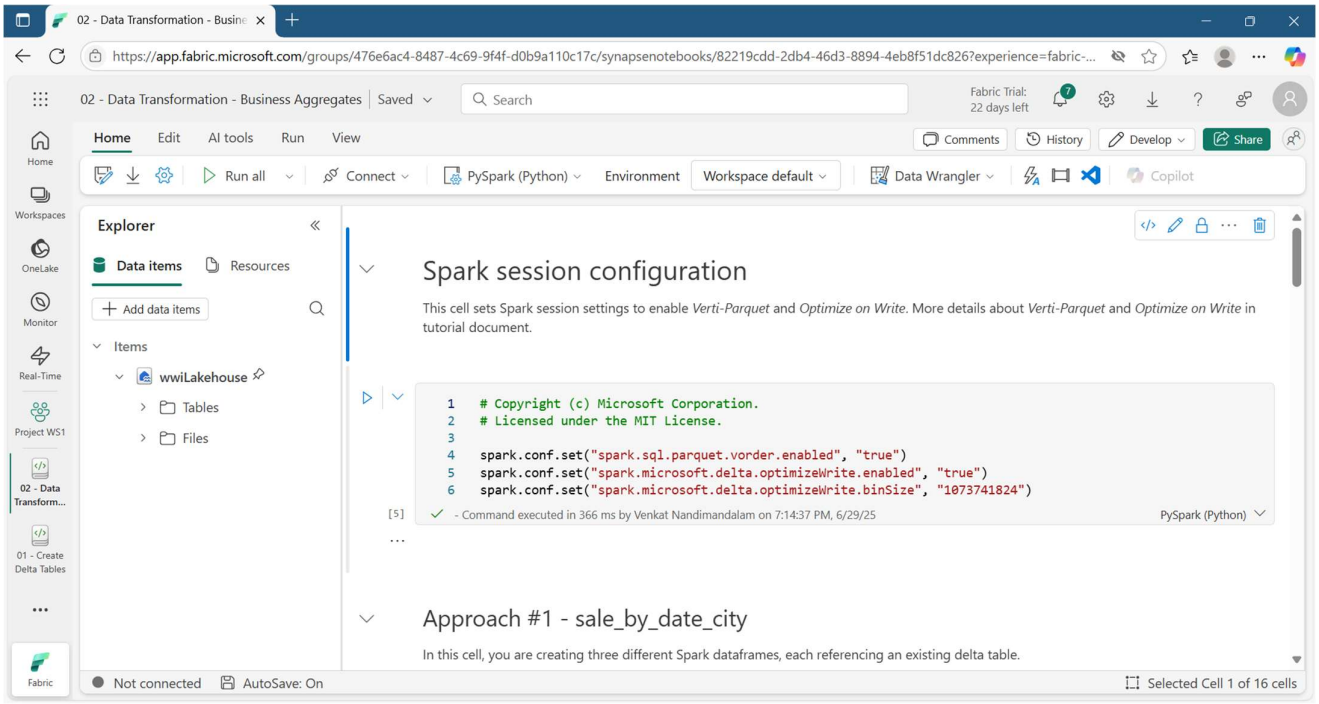
- After **the fact table loaded**, you can proceed to load data for the rest of the dimensions. The following cell creates a function that reads raw data from the Files section of the Lakehouse for each table names passed as a parameter.



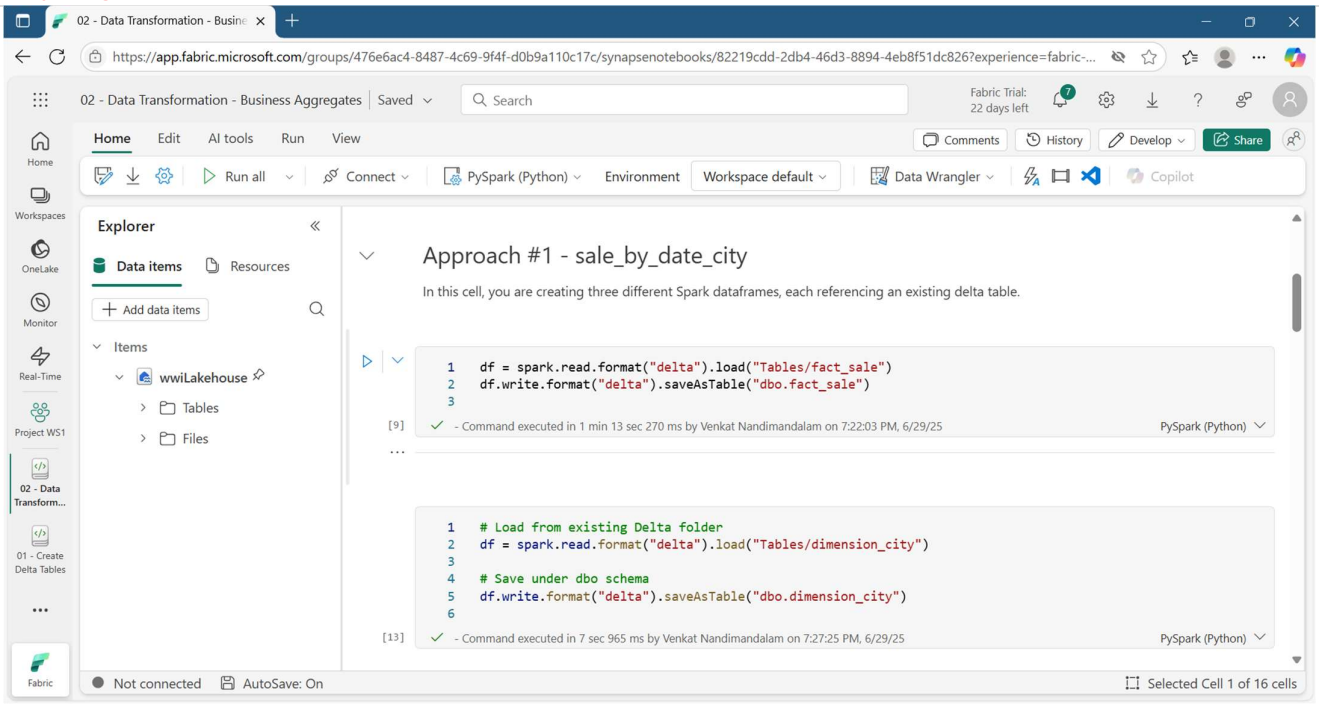
- To validate the created tables, right-click on **wwiLakehouse** and select **Refresh**. The tables appear in the above figure.
- Open the Lakehouse again, then launch the second notebook. In the Lakehouse view, select **Open notebook** > **Existing notebook** from the **Home** menu.
- Select the **02 – Data Transformation** – **Business** notebook to open it.



- There are two different approaches to transforming and generating business aggregates:
  - Approach #1 – Use **PySpark** to join and aggregates data for generating business – level summaries. This approach is ideal if you have a background in programming, such as Python or PySpark.
  - Approach #2 – Use **Spark SQL** to join and aggregates data using SQL syntax. This method is useful for those more comfortable with SQL-based development.
  - First, set the Spark session configuration before proceeding with either method.



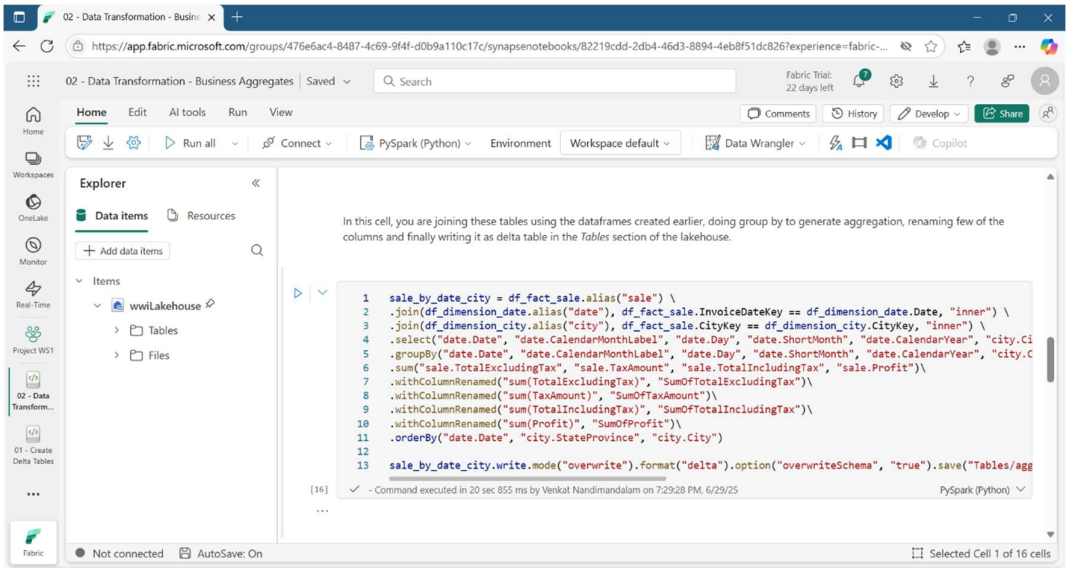
- **Approach #1 (sale_by_data_city)** – Use **PySpark** to join and aggregate data for generating business aggregates. Using the following code, three different Spark DataFrames are created, each referencing an existing Delta table.A few columns are renamed, and the results are written as a Delta table to the **Tables** section of the Lakehouse to persist the data.
  - In this step, you create three Spark DataFrames, each referencing an existing Delta table.



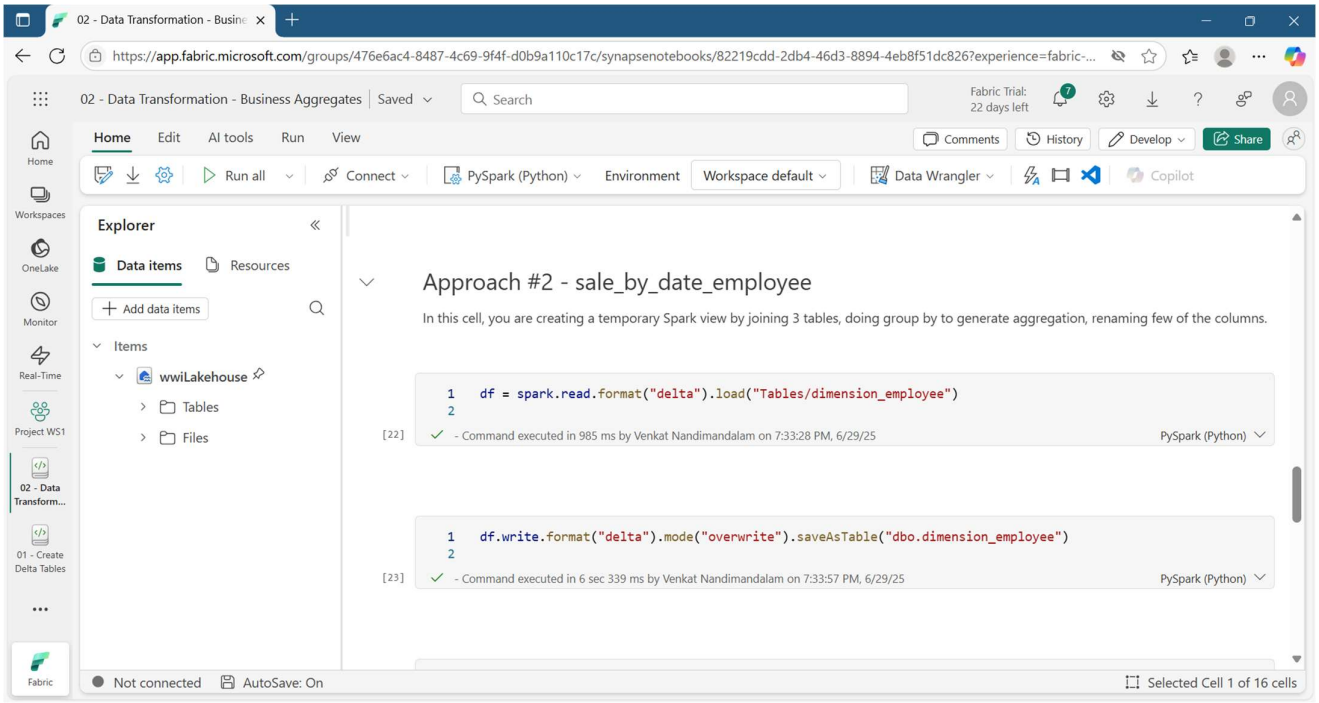

```
df_fact_sale = spark.read.table("dbo.fact_sale")
df_dimension_date = spark.read.table("dbo.dimension_date")
df_dimension_city = spark.read.table("dbo.dimension_city")

```

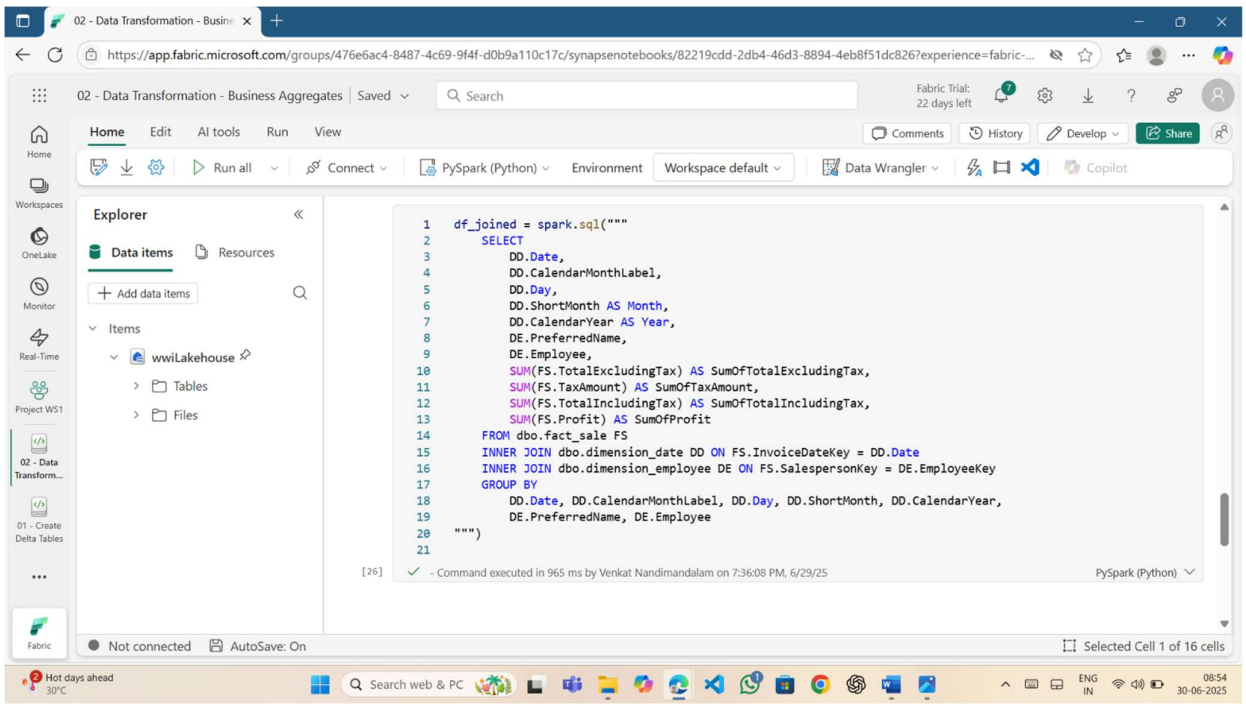
[15] ✓ - Command executed in 3 sec 146 ms by Venkat Nandimandalam on 7:28:51 PM, 6/29/25 PySpark (Python)

- Add the following code to the same cell to join the tables using the dataframes created earlier. **group by** generate aggregation, rename a few of the columns, and finally write the results as a Delta table in the Tables section of the Lakehouse.

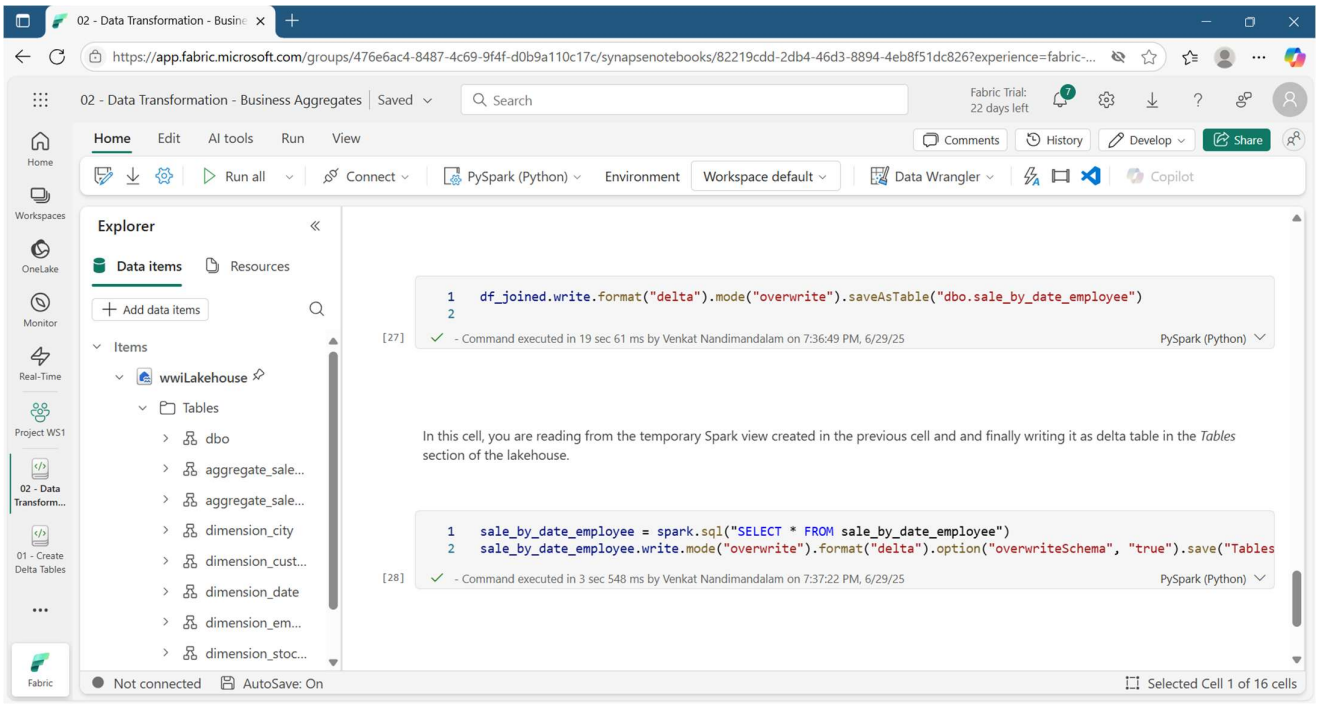


- **Approach #2 (sale_by_date_employee)** – Use Spark SQL to join and aggregate data for generating business aggreagtes. Using the following code, I create a temporary Spark view by joining three tables, perform a **group by** to generate aggregation, and rename a few of the columns. Finally, I read form the temporary Saprk view and write the results as a Delta table in the **Tables** section of the Lakehouse to persist the data.





- In this step, the data is read from the temporary Spark view created earlier and written as a Delta table to the **Tables** section of the lakehouse.

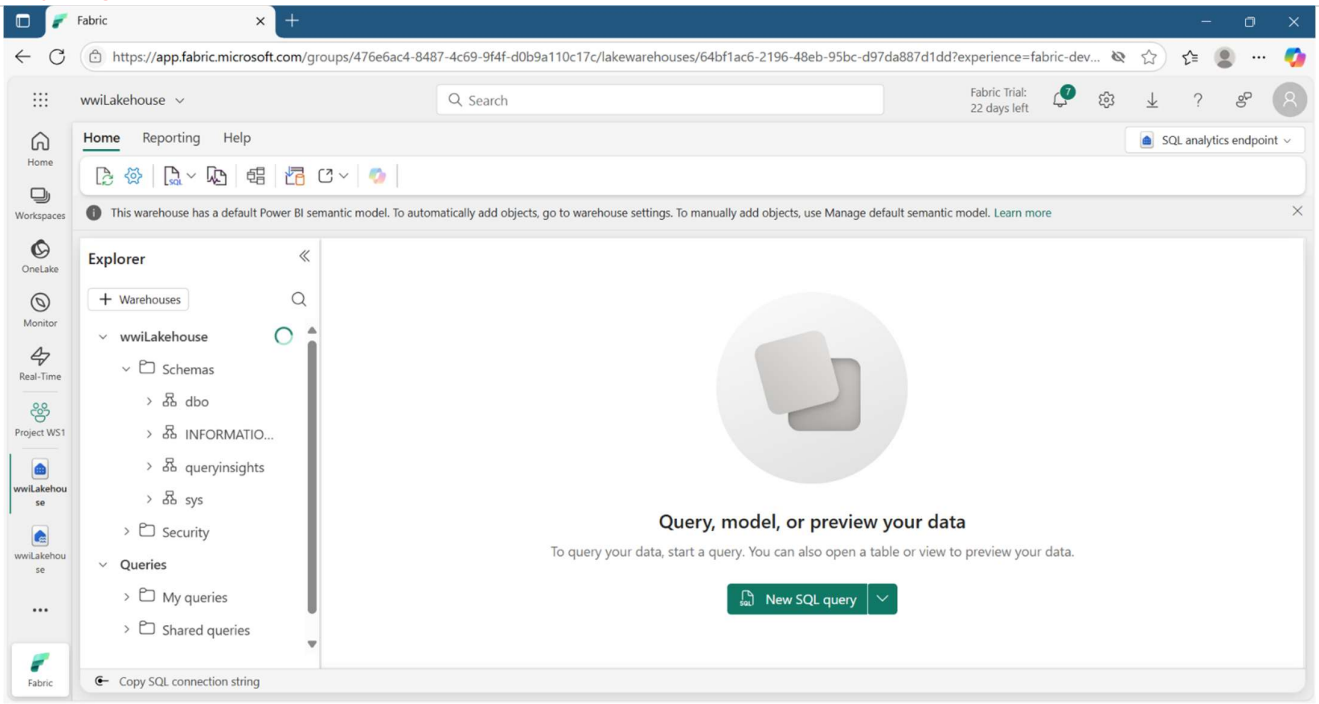


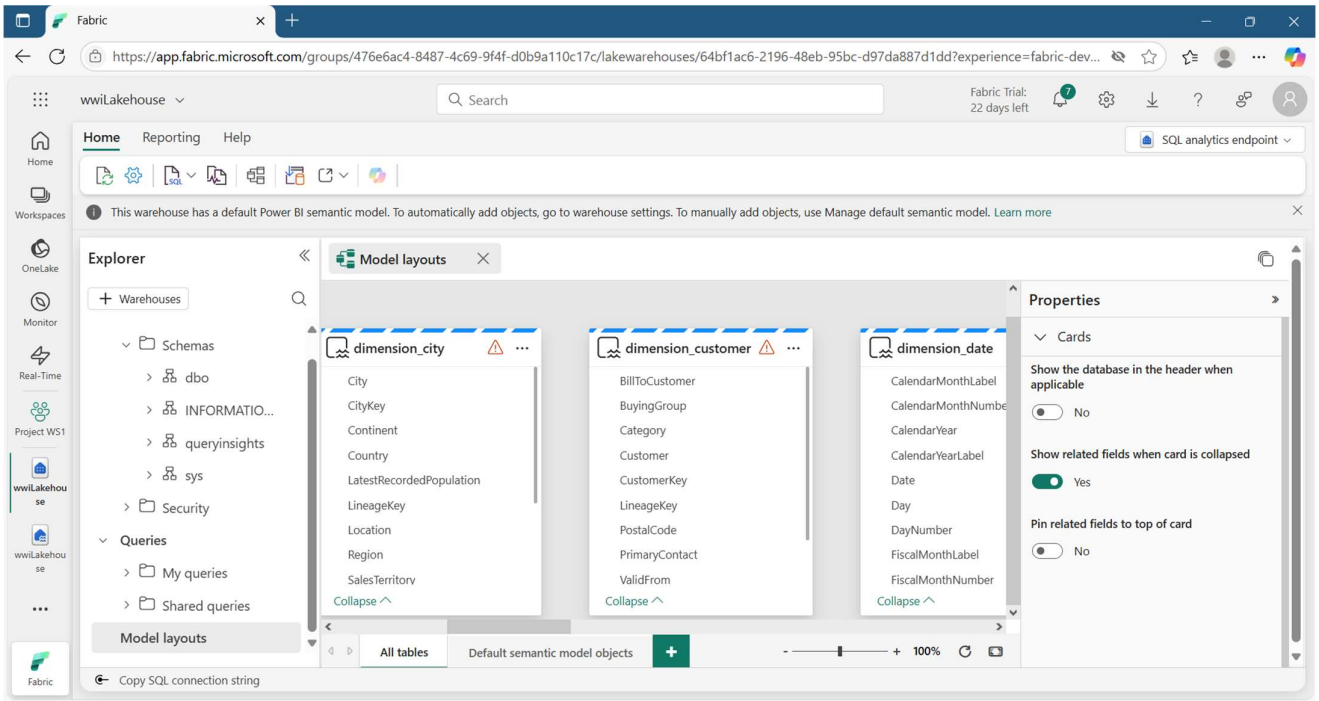
- The both approaches – using PySpark or Spark SQL produce a similar outcome. To avoid the need to learn a new technology or compromise on performance, choose the method that best suits for background and personal preferences.
- you may notice that the data writing as **Delta lake** files. Microsoft Fabric's automatic table discovery and registration feature detects and register these files in the metastore. Therefore you don't need to explicitly run *CREATE TABLE* statements to make them available for SQL queries.

BUILDING REPORTS IN MICROSOFT FABRIC.

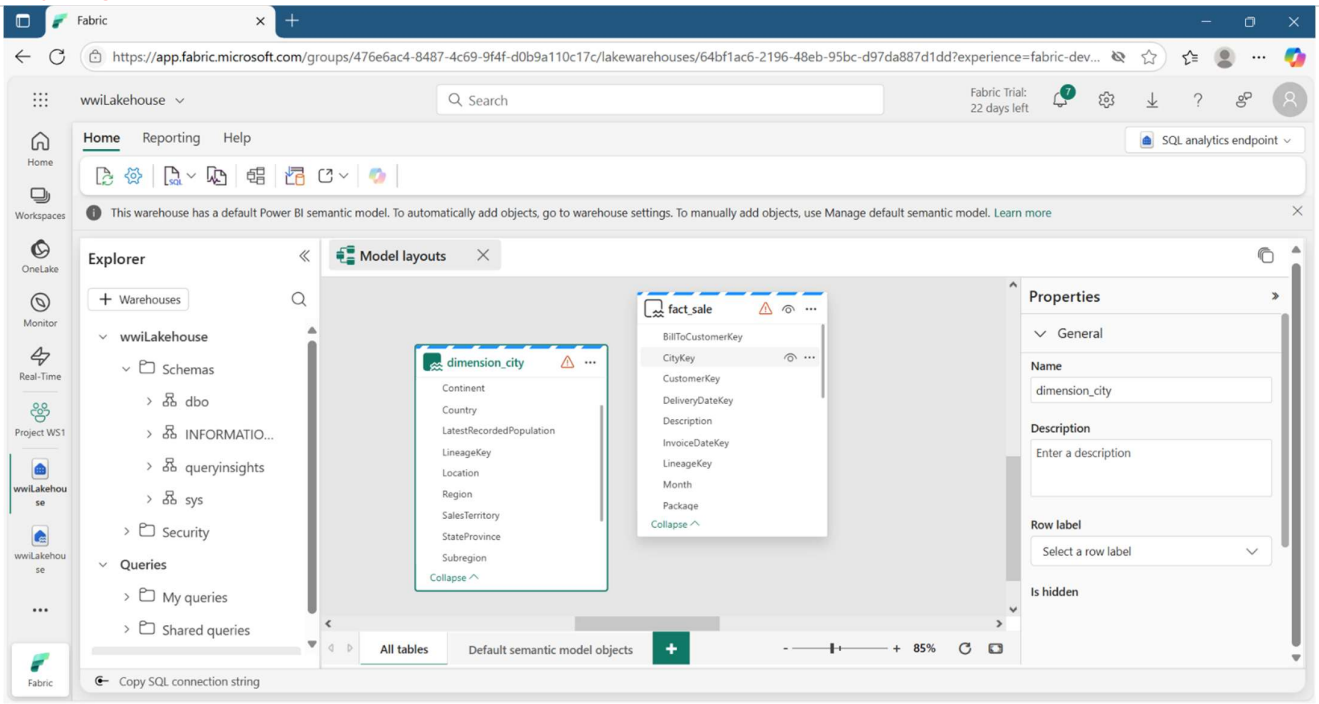
Build a report

- From the **wwiLakehouse** , select **SQL Analytics Endpoint** from the Lakehouse dropdown menu at the top.
- In the SQL Analytics Endpoint pane, you should be able to see all the created tables. Click **Refresh** at the top, then select the **Model** at the bottom to open the default Power BI semantic model.



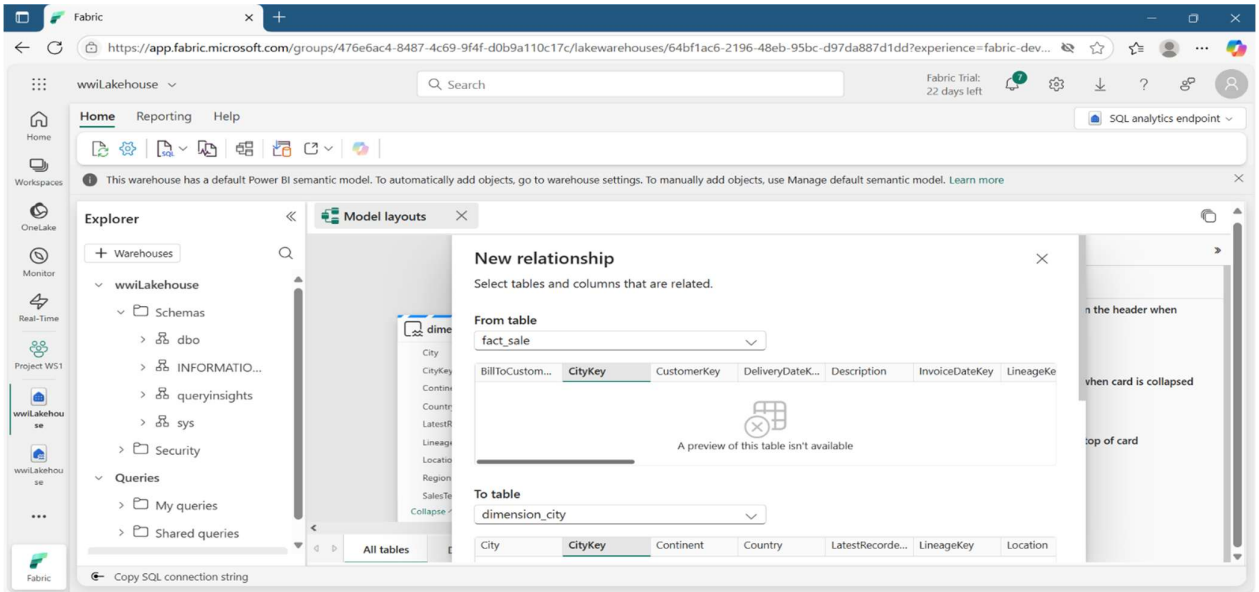


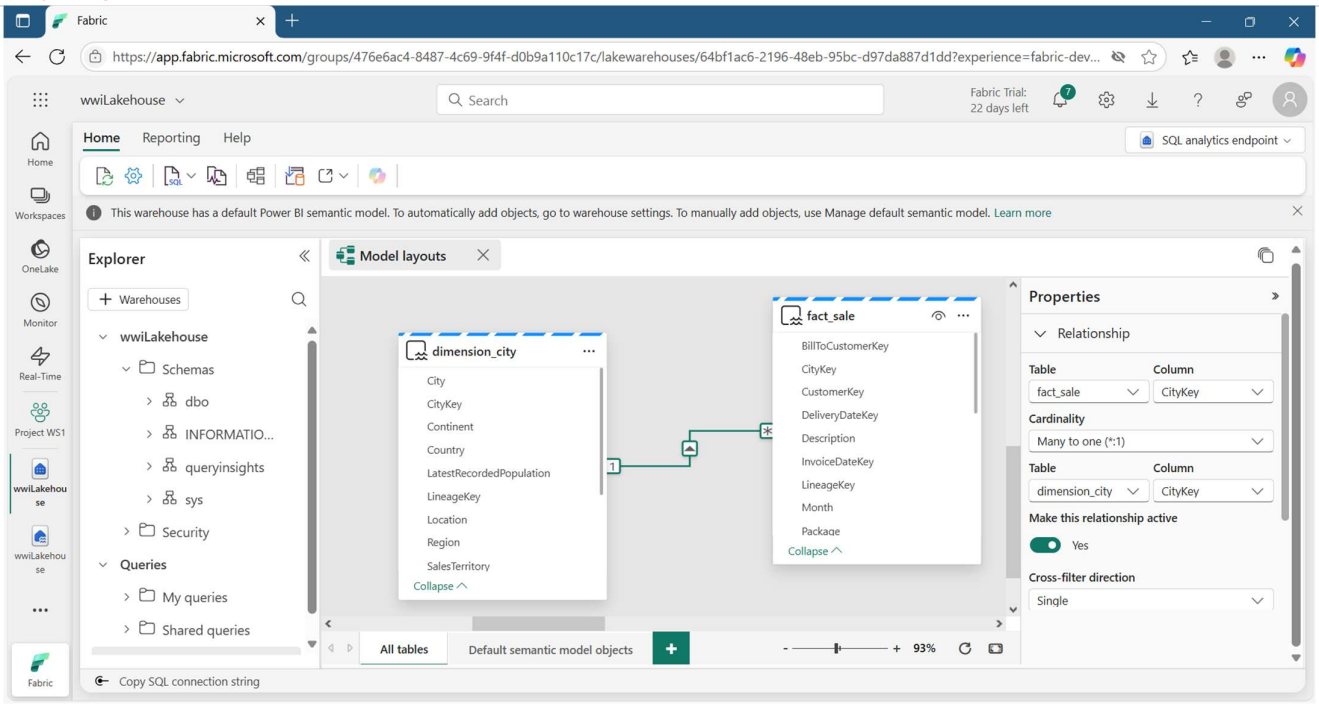
- To build meaningful reports, you need to define the relationship between different tables so that can reports and visualization based on the data coming across different tables. From the **fact_sale** table, drag the **CityKey** field and drop it on the **CityKey** in the **dimension_city** table .
- The **New relationship** dialog box appears.



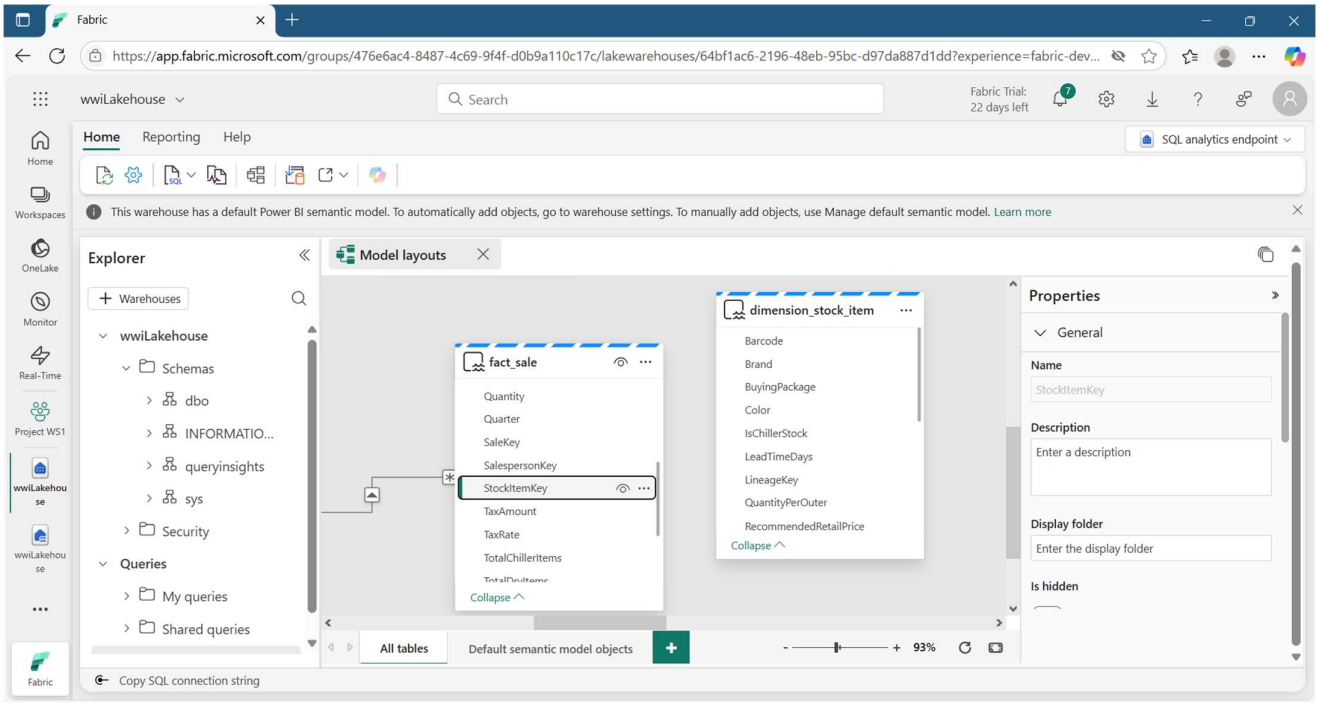
❖ In the New relationship dialog box:

- Table 1 is populated with **fact_sale** and column of CityKey.
- Table 2 is populated with **dimension_city** and the column of CityKey.
- Cardinality : **Many to one (*:1).**
- Cross filter direction : **Single.**
- Leave the box next to Make this relationship active selected.
- Select the box next to Assume referential integrity.
- Select **Save.**





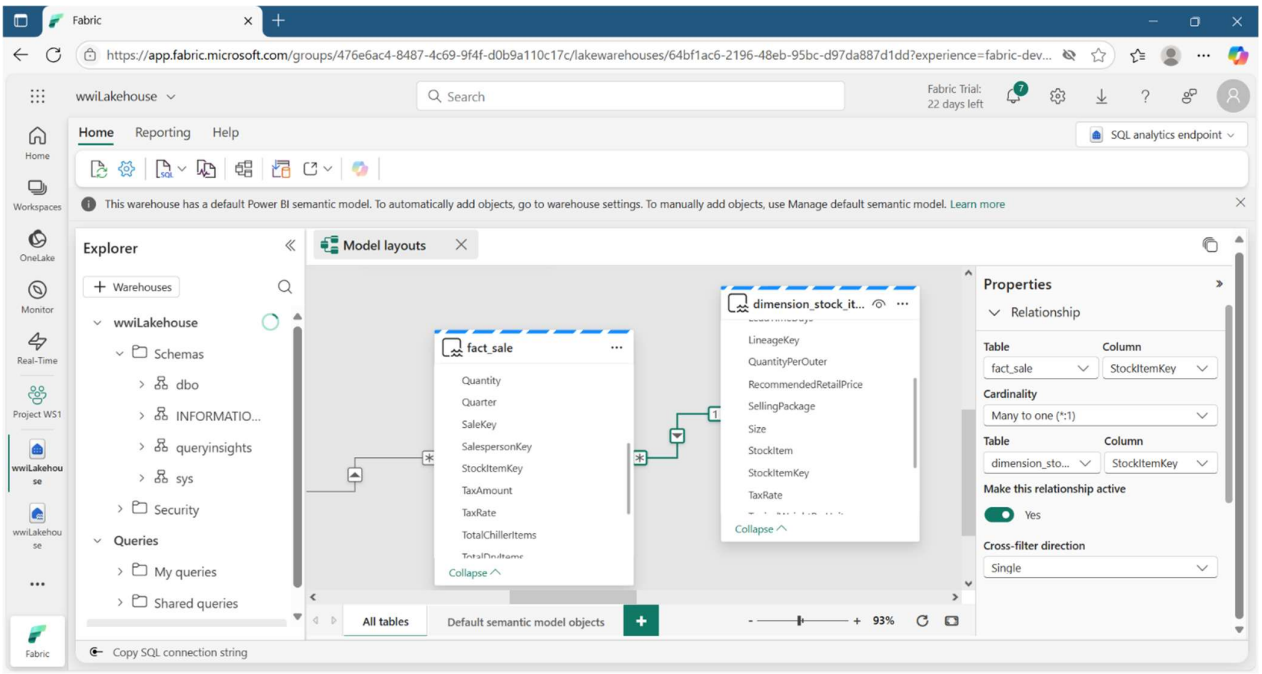
❖ **The relationship between these two tables (fact_sale)-(dimension_stock_item):**



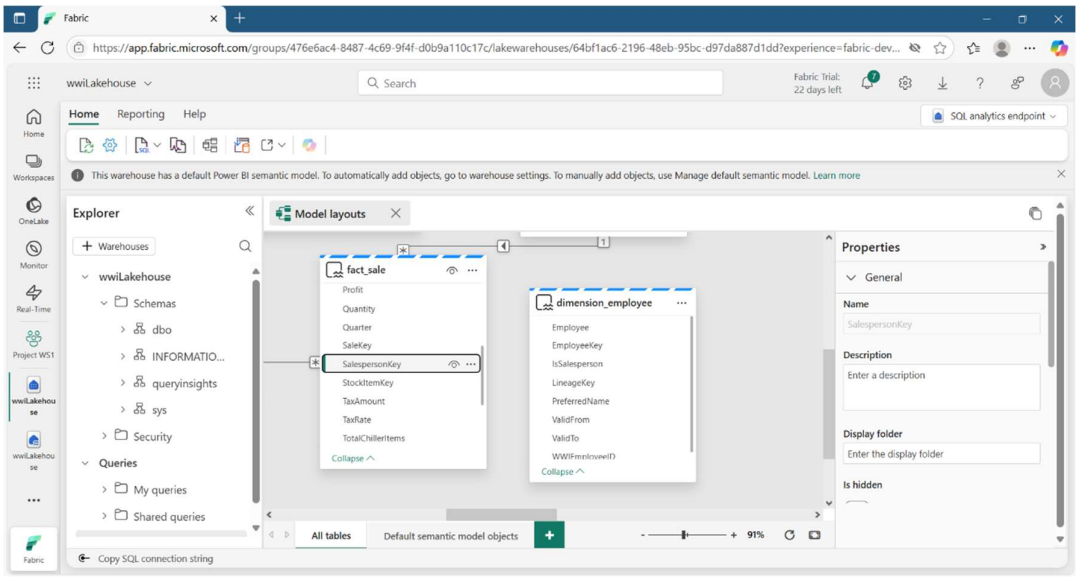
❖ In the New relationship dialog box:

- Table 1 is populated with **fact_sale** and column of **StockItemKey.**
- Table 2 is populated with **dimension_stock_city** and the column of **StockItemKey.**
- Cardinality : **Many to one (*:1).**

- Cross filter direction : **Single.**
- Leave the box next to Make this relationship active selected.
- Select the box next to Assume referential integrity.
- Select **Save.**



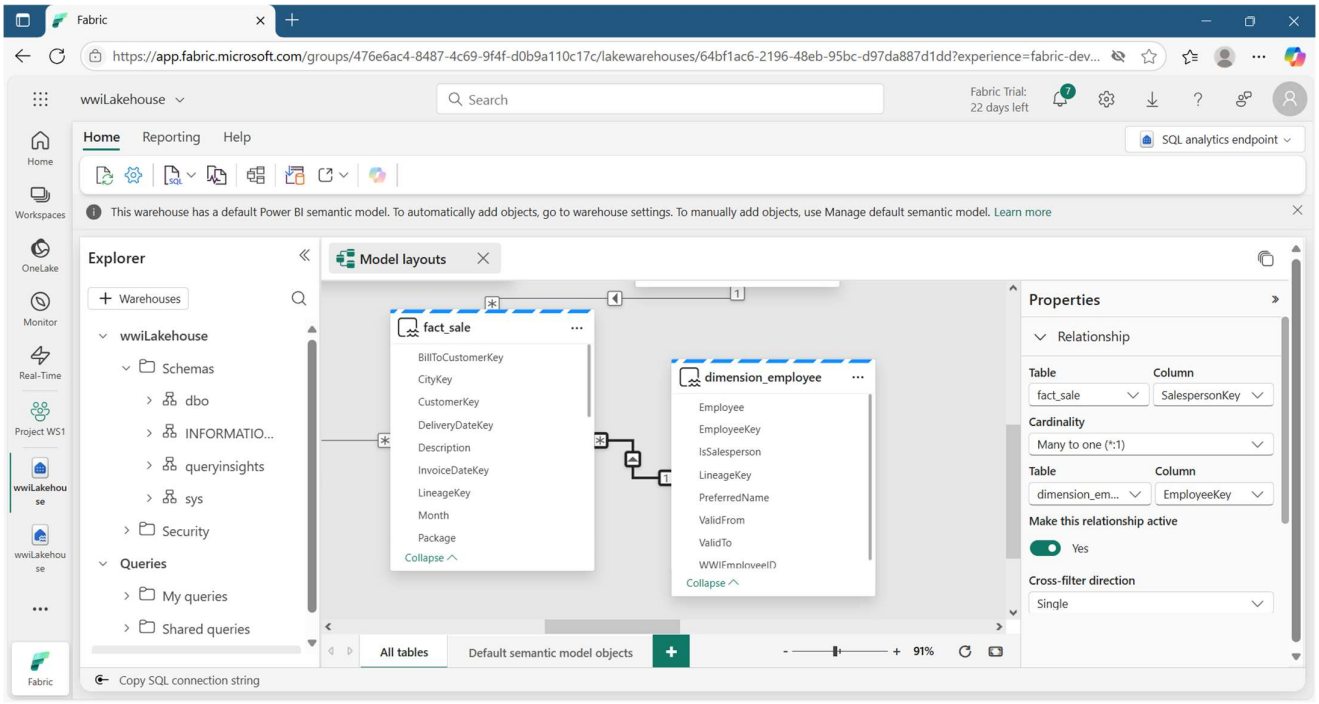
❖ **The relationship between these two tables (fact_sale)-(dimension_employee):**



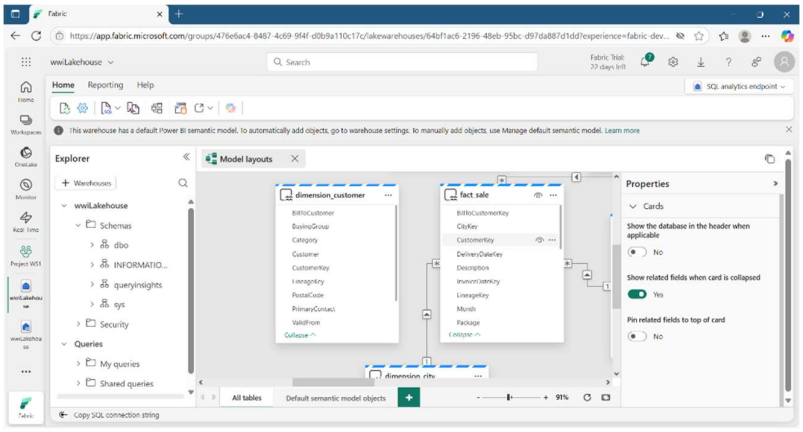
❖ In the New relationship dialog box:

- Table 1 is populated with **fact_sale** and column of **SalespersonKey.**
- Table 2 is populated with **dimension_employee** and the column of **EmployeeKey.**
- Cardinality : **Many to one (*:1).**

- Cross filter direction : **Single.**
- Leave the box next to Make this relationship active selected.
- Select the box next to Assume referential integrity.
- Select **Save.**



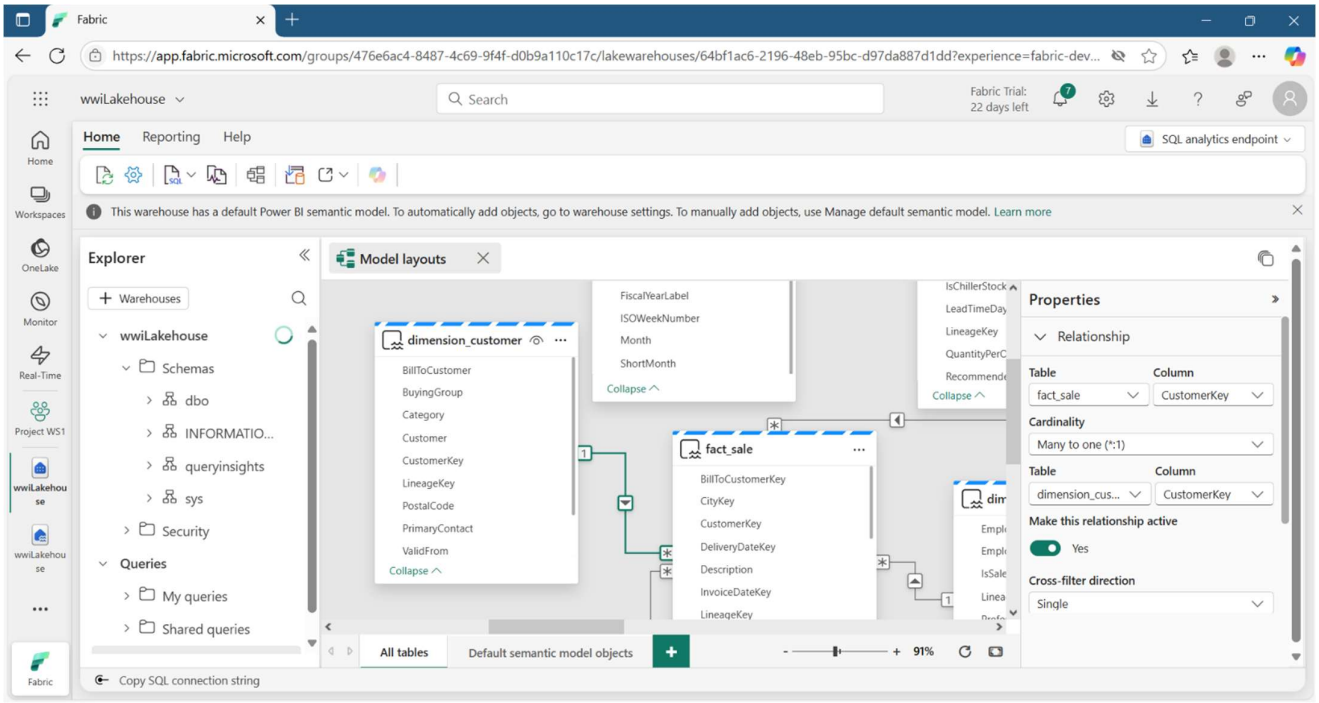
❖ The relationship between these two tables (fact_sale)-(dimension_customer):



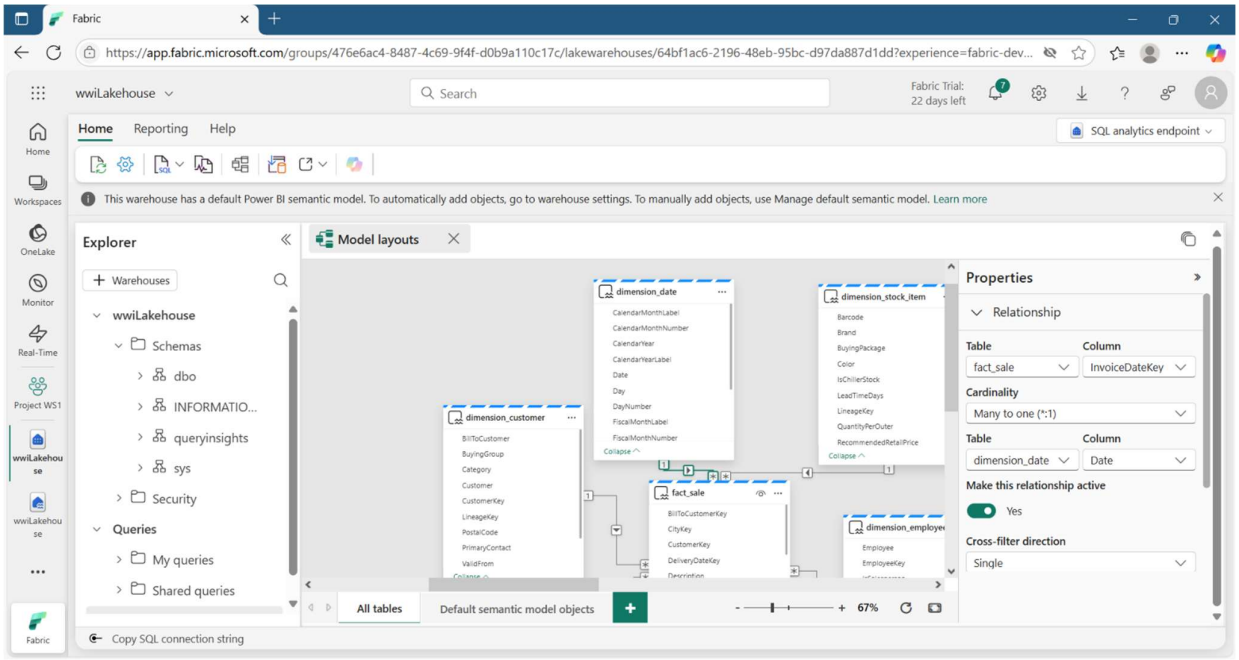
❖ In the New relationship dialog box:

- Table 1 is populated with **fact_sale** and column of **CustomerKey.**
- Table 2 is populated with **dimension_customer** and the column of **CustomerKey.**
- Cardinality : **Many to one (*:1).**
- Cross filter direction : **Single.**
- Leave the box next to Make this relationship active selected.

- Select the box next to Assume referential integrity.
- Select **Save.**



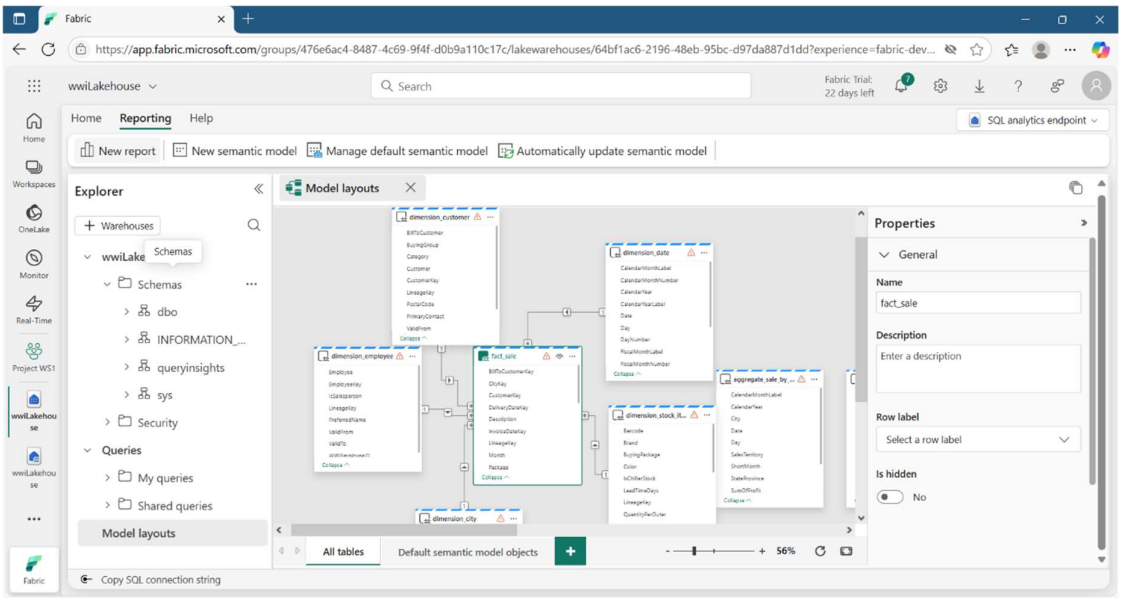
❖ The relationship between these two tables (fact_sale)-(dimension_date):



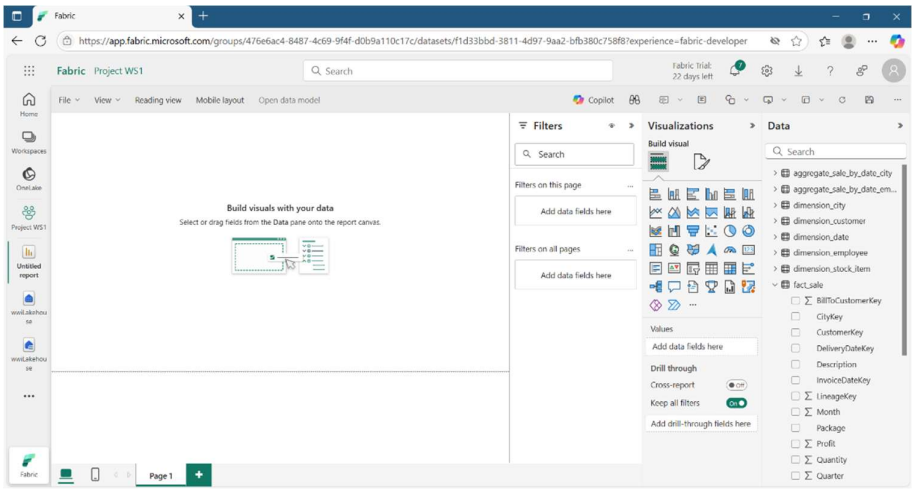
❖ In the New relationship dialog box:

- Table 1 is populated with **fact_sale** and column of **invoiceDateKey.**
- Table 2 is populated with **dimension_date** and the column of **Date.**

- Cardinality : **Many to one (*:1).**
- Cross filter direction : **Single.**
- Leave the box next to Make this relationship active selected.
- Select the box next to Assume referential integrity.
- Select **Save**



❖ From the Reporting tab , select New Report to start building reports/dashboards in Power BI. On the Power BI report canvas, you can create reports to meet business requirements by dragging required columns from the **Data** pane to the canvas and using one or more of available visualizations.

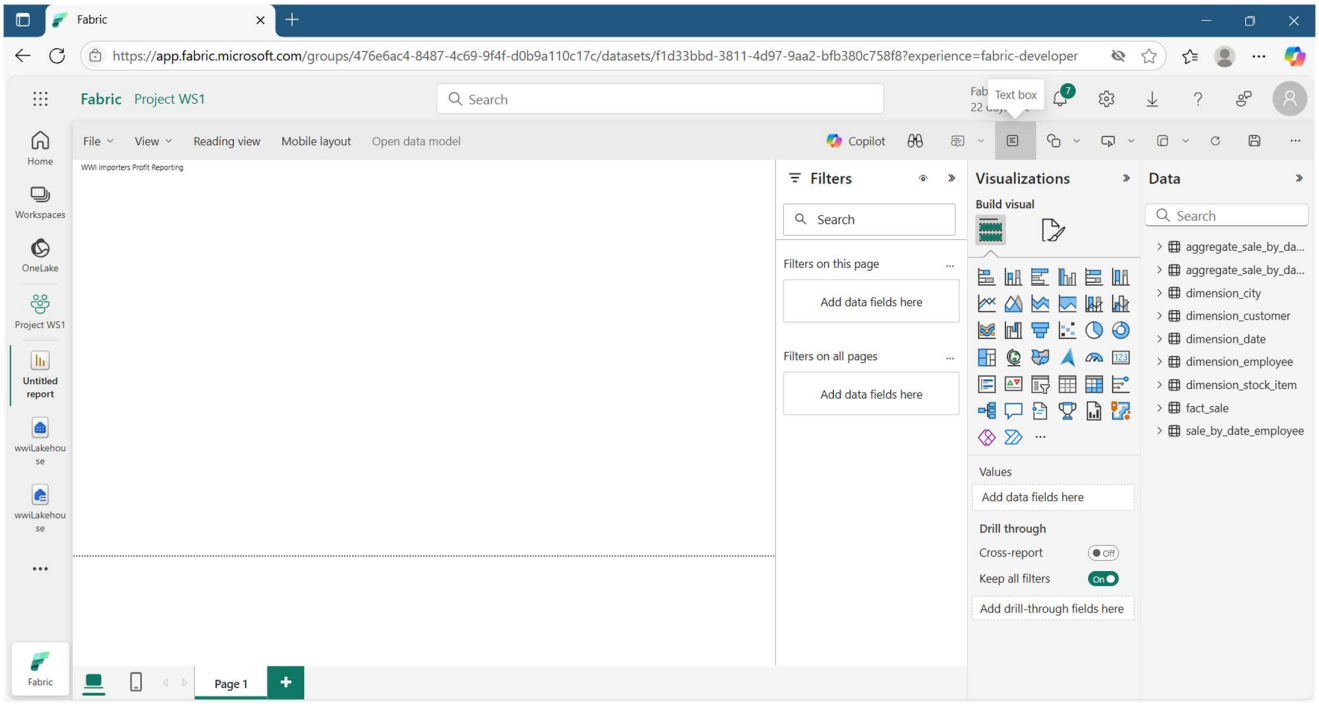


❖ ***Add a Title :***

i) In the right side of ribbon, select Text box.

ii) Type in **WW Importers Profit Reporting.**

iii) Highlight the text, increase the size to 20, and move it to the upper left of the report page.

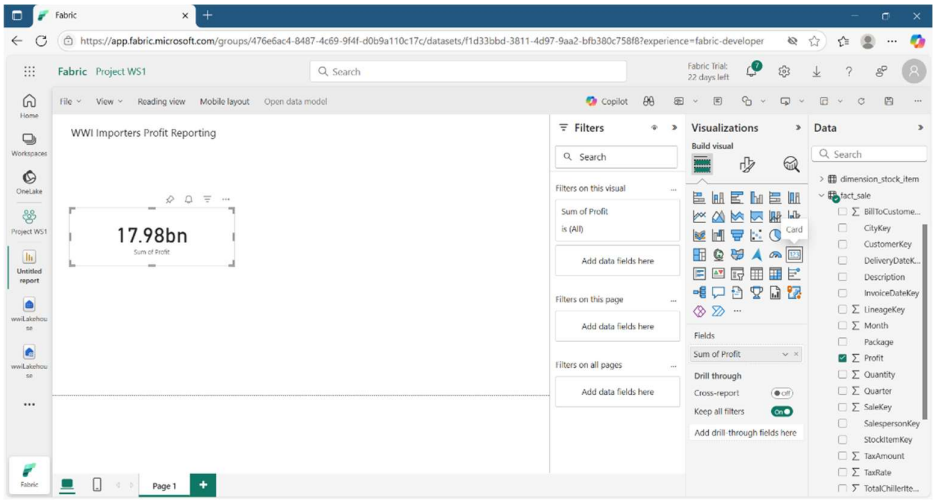


❖ ***Add a Card :***

i) On the Data pane, expand **fact_sale**, and check the box next to **Profit**. This selection creates a column chart and adds the field to the Y-axis.

ii) With the chart selected, select the **Card** visual in the visualization pane. This selection converts the visual to a card.

iii) Place the card under the Title like this.

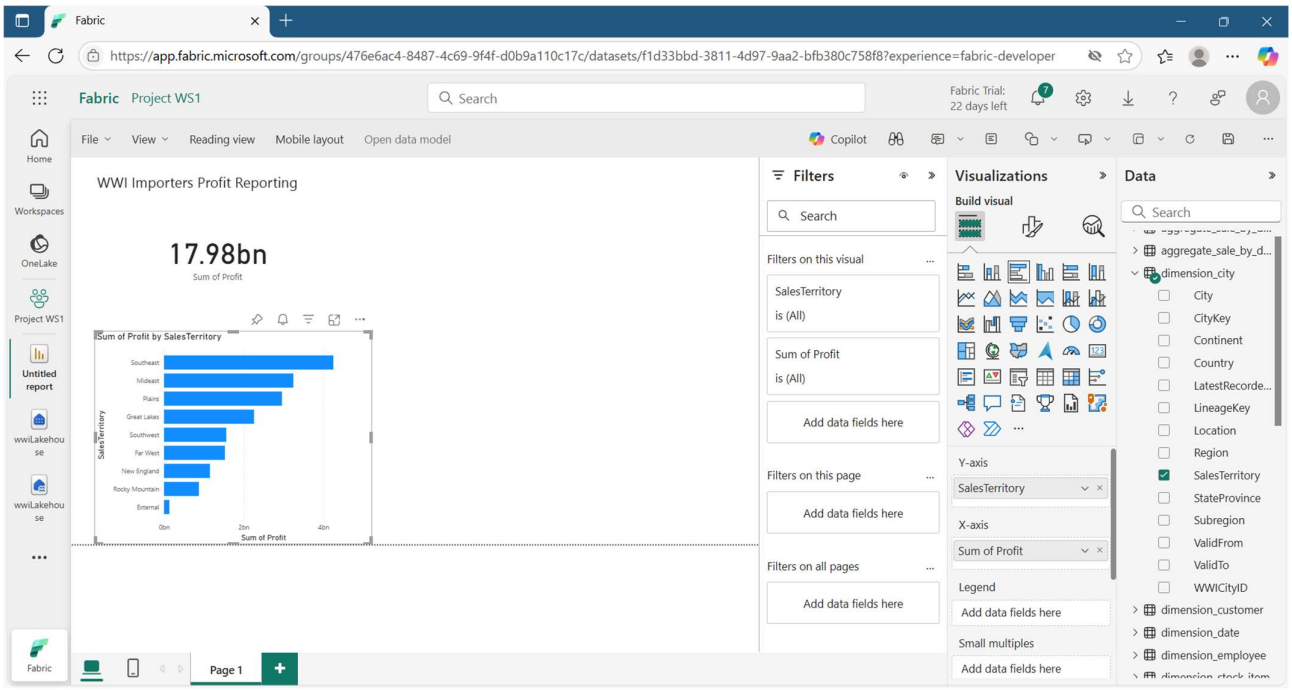


❖ ***Add a Bar chart :***

i) On the **Data** pane, expand **fact_sales** and check the box next to **Profit**. This selection creates a column chart and adds the field to the X-axis.

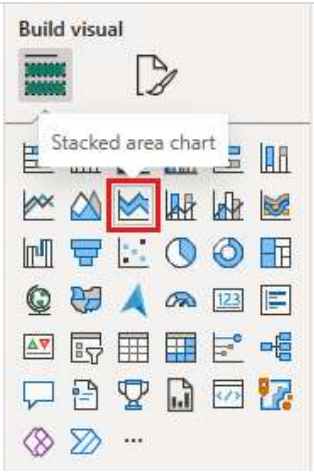
ii) On the **Data** pane, expand **dimension_city** and check the box for **SalesTerritory**. This selection adds the field to the Y-axis.

iii) With the bar chart selected, select the **Clustered bar chart** visual in the visualization pane. This selection converts the column chart into a bar chart. Resize the Bar chart and move it under the **Title and Card.**

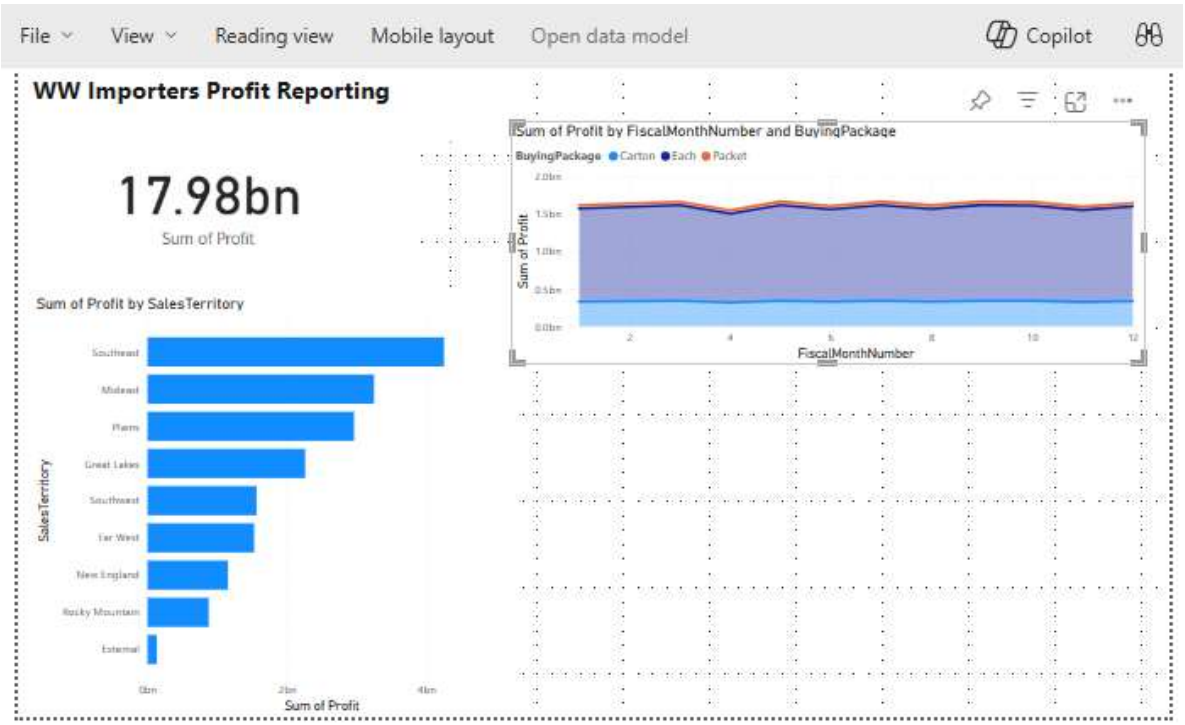


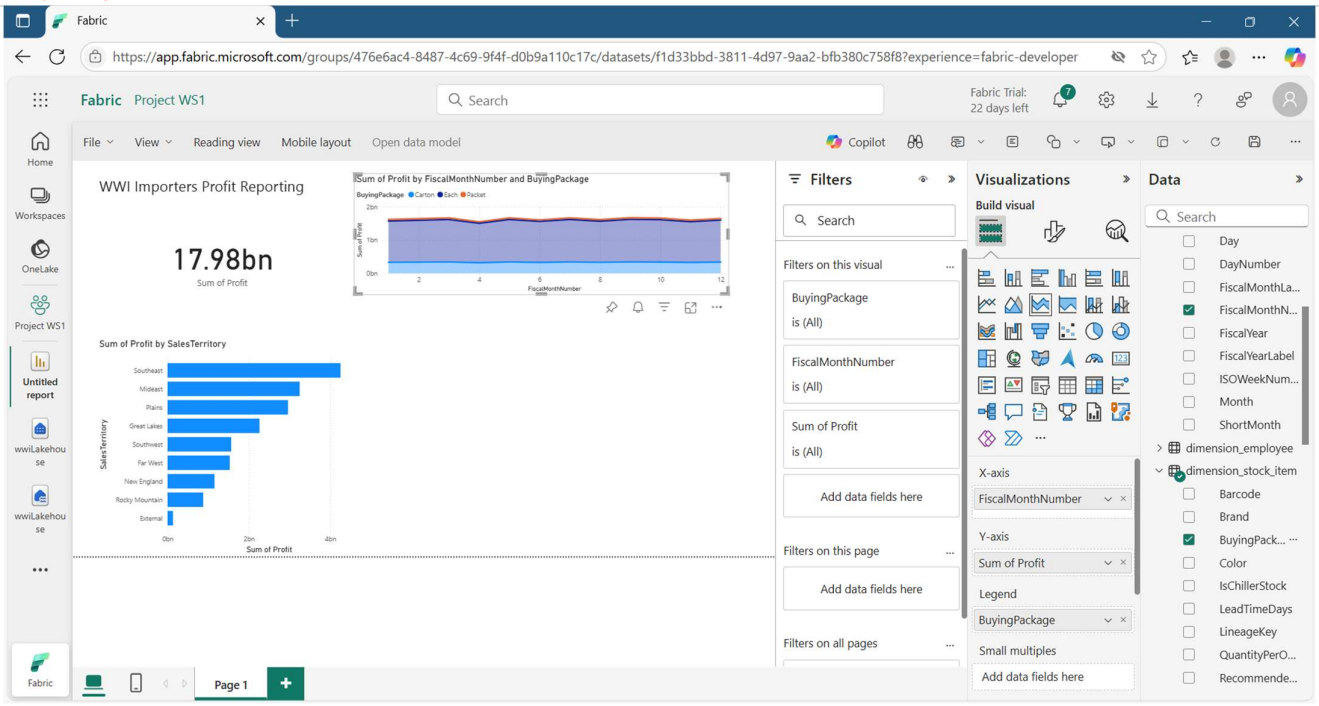
❖ ***Build a Stacked area chart visual :***

- On the **Visualizations** pane, select the **Stacked area chart visual.**



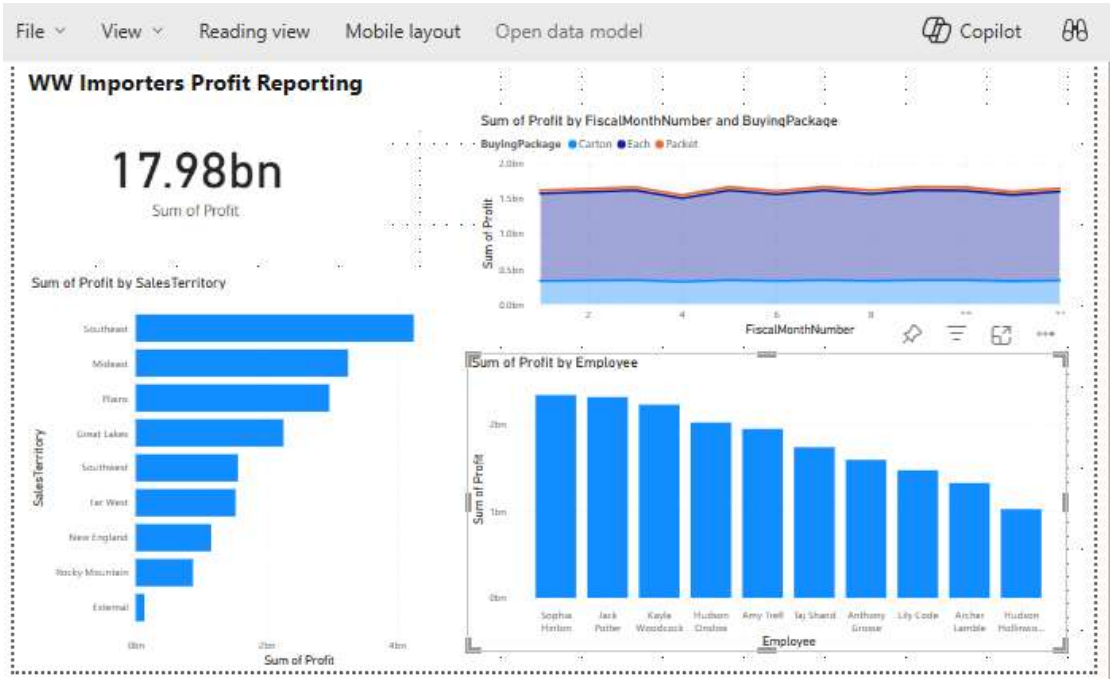
- Reposition and resize the stacked area chart to the right of the card and bar chart visuals created in the previous steps.
- On the **Data** pane, expand **fact_sales** and check the box next to **Profit**. Expand **dimension_date** and check the box next to **FiscalMonthNumber**. This selection creates a filled line chart showing profit by fiscal month.
- On the **Data** pane, expand **dimension_stock_item** and drag **BuyingPackage** into the Legend field well. This selection adds a line for each of the Buying Packages.

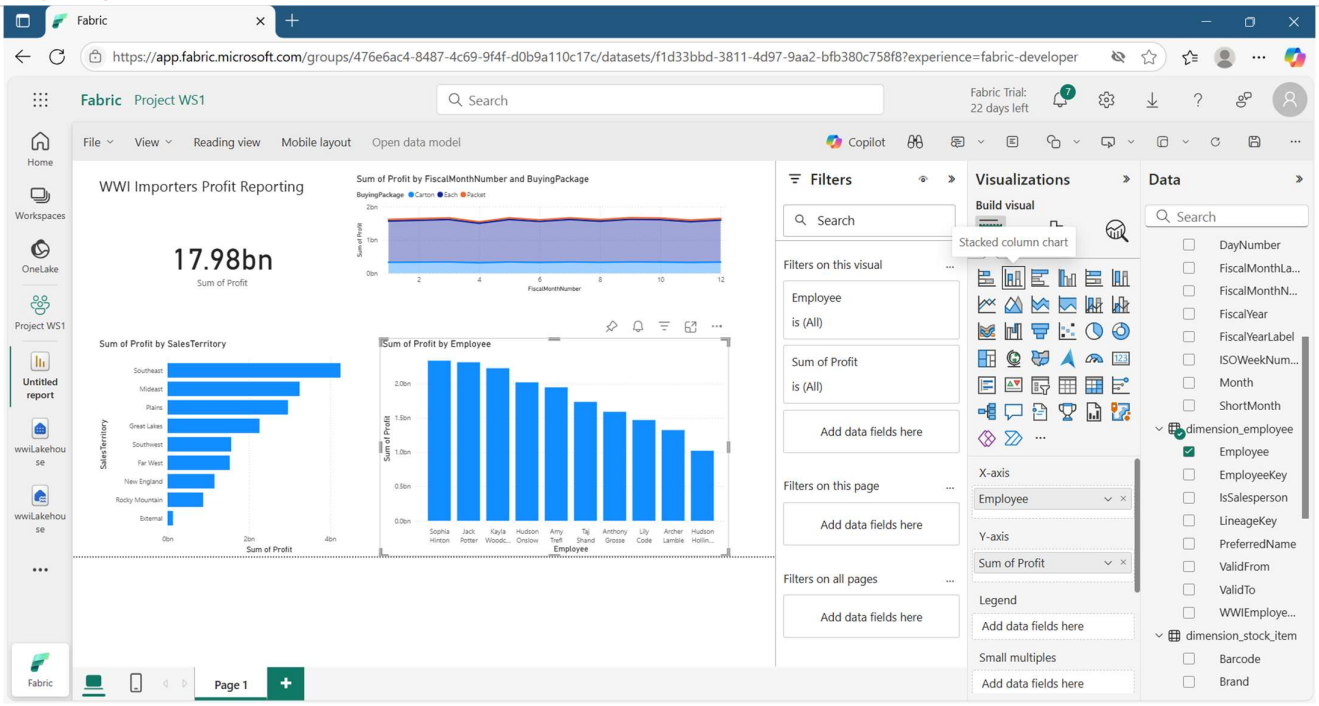




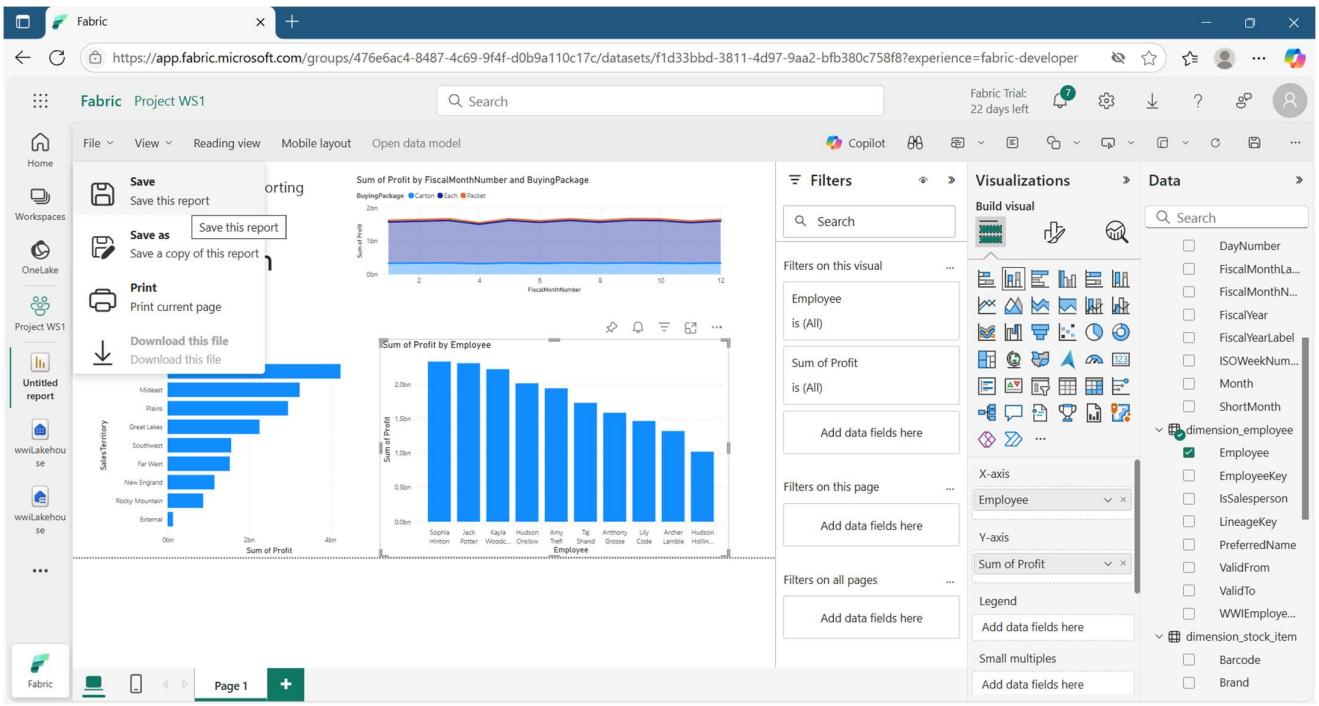
❖ Build a Column chart :

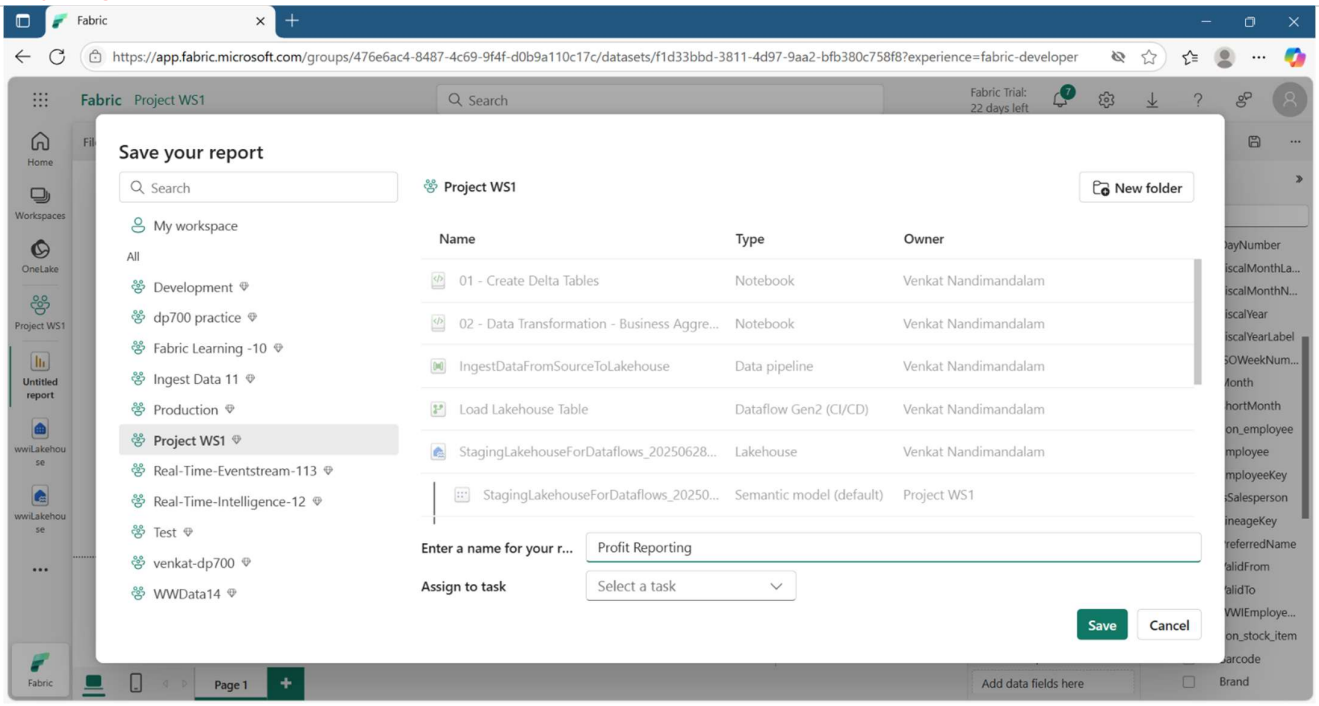
- On the **Data** pane, expand **fact_sales** and check the box next to **Profit**. This selection adds the field to the Y-axis.
- On the **Data** pane, expand **dimension_employee** and check the box next to **Employee**. This selection adds the field to the X-axis.

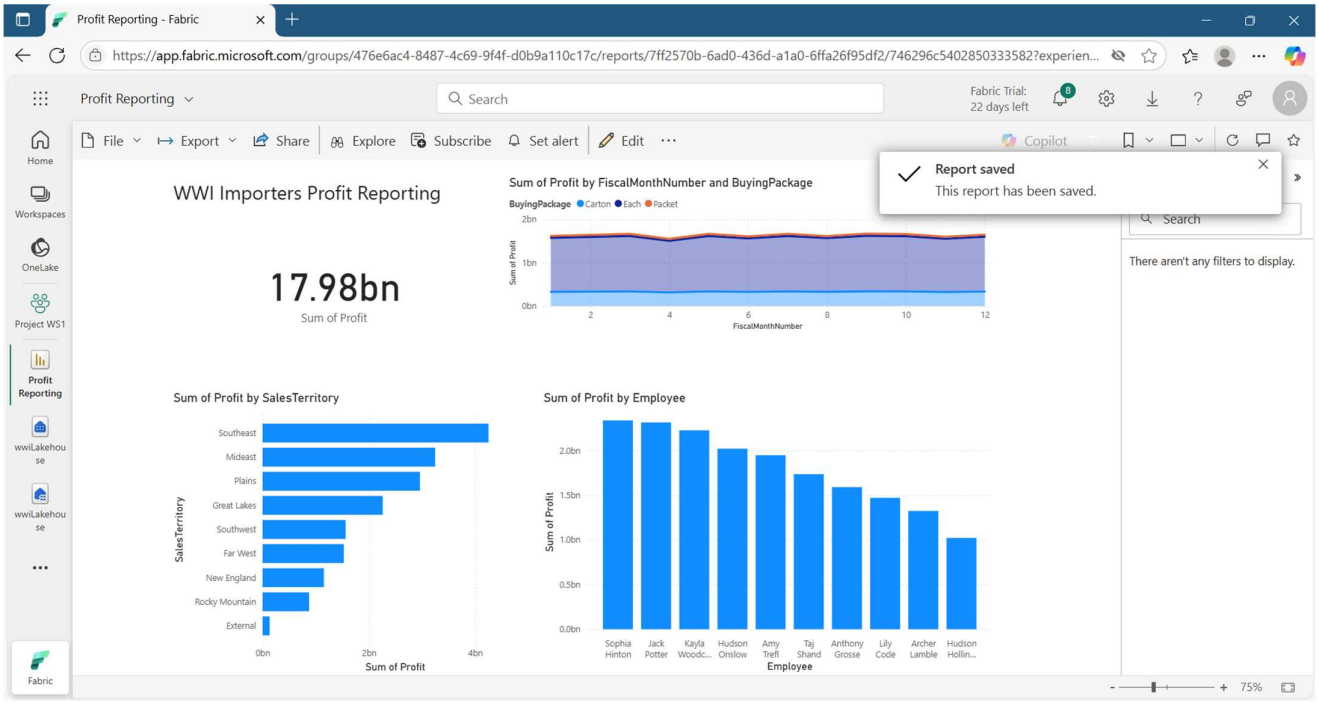




- From the ribbon, select **File > Save.**
- Enter the name of your report as **Profit Reporting.**
- Select **Save.**




Fabric
https://app.fabric.microsoft.com/groups/476e6ac4-8487-4c69-9f4f-d0b9a110c17c/datasets/f1d33bbd-3811-4d97-9aa2-bfb380c758f8?experience=fabric-developer
Fabric Project WS1
Search
Fabric Trial: 22 days left
Save your report
Search
My workspace
All
Development
dp700 practice
Fabric Learning -10
Ingest Data 11
Production
Project WS1
Real-Time-Eventstream-113
Real-Time-Intelligence-12
Test
venkat-dp700
WWData14
Project WS1
New folder
Name
Type
Owner
01 - Create Delta Tables
Notebook
Venkat Nandimandalam
02 - Data Transformation - Business Aggre...
Notebook
Venkat Nandimandalam
IngestDataFromSourceToLakehouse
Data pipeline
Venkat Nandimandalam
Load Lakehouse Table
Dataflow Gen2 (CI/CD)
Venkat Nandimandalam
StagingLakehouseForDataflows_20250628...
Lakehouse
Venkat Nandimandalam
StagingLakehouseForDataflows_20250...
Semantic model (default)
Project WS1
Enter a name for your r...
Profit Reporting
Assign to task
Select a task
Save
Cancel
Page 1
Add data fields here
Brand


Profit Reporting - Fabric
https://app.fabric.microsoft.com/groups/476e6ac4-8487-4c69-9f4f-d0b9a110c17c/reports/7ff2570b-6ad0-436d-a1a0-6ffa26f95df2/746296c5402850333582?experien...
Profit Reporting
Search
Fabric Trial: 22 days left
File
Export
Share
Explore
Subscribe
Set alert
Edit
Copilot
Report saved
This report has been saved.
WWI Importers Profit Reporting
17.98bn
Sum of Profit
Sum of Profit by FiscalMonthNumber and BuyingPackage
BuyingPackage Carton Each Packet
Sum of Profit
FiscalMonthNumber
Search
There aren't any filters to display.
Sum of Profit by SalesTerritory
Southeast
Mideast
Plains
Great Lakes
Southwest
Far West
New England
Rocky Mountain
External
SalesTerritory
Sum of Profit
Sum of Profit by Employee
Sum of Profit
Employee
Sophia Hinton
Jack Potter
Kayla Woodc...
Hudson Onslow
Amy Trefl
Taj Shand
Anthony Grosse
Lily Code
Archer Lamble
Hudson Hollin...
75%