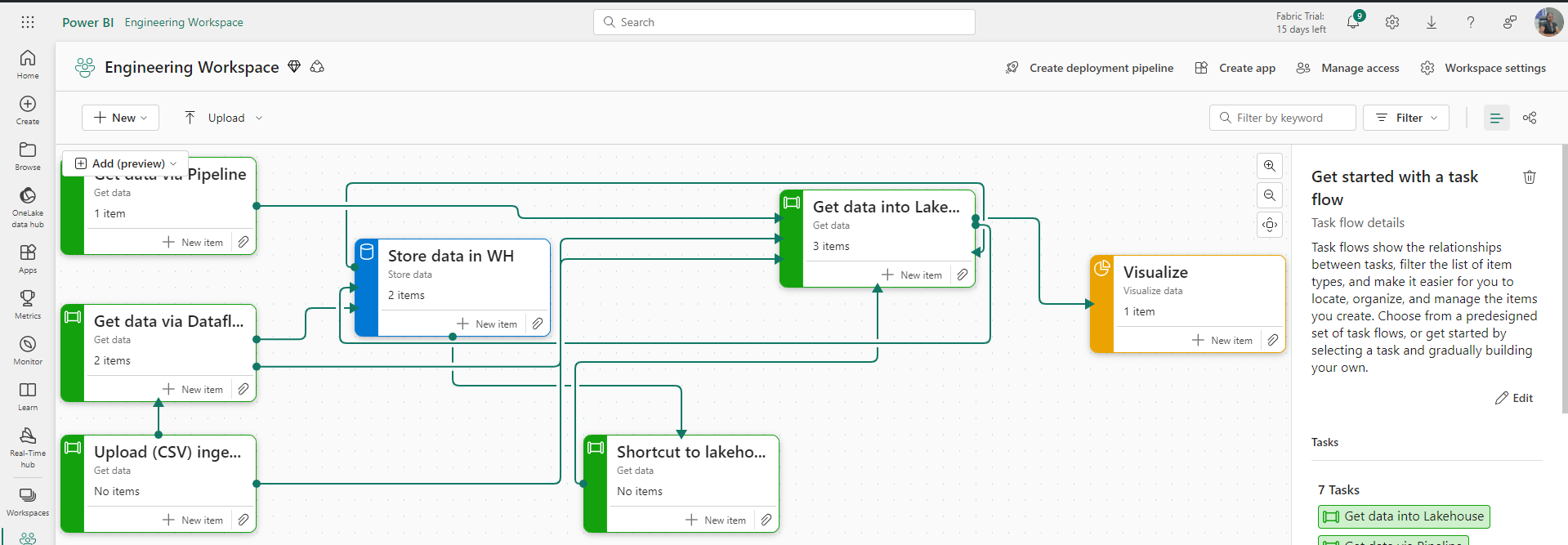
**  
  
Wide World Importer Microsoft Fabric End -End Project**

This project outlines the end-to-end analysis of the wide world imports of various products to different countries. It encompasses the different aspects of Microsoft fabric including data engineering, data factory and data analytics.

**Four things were taken into consideration in this project which are;**

1. Data Ingestion
2. Data Transformation
3. Data modelling, and
4. Data visualization

**Purpose of the project**

To ingest data into a newly created Lakehouse (a storage that hold both structured and unstructured data) using various methods, transforming the data using dataflow Gen2 and finally creating a compelling story via data visualization.

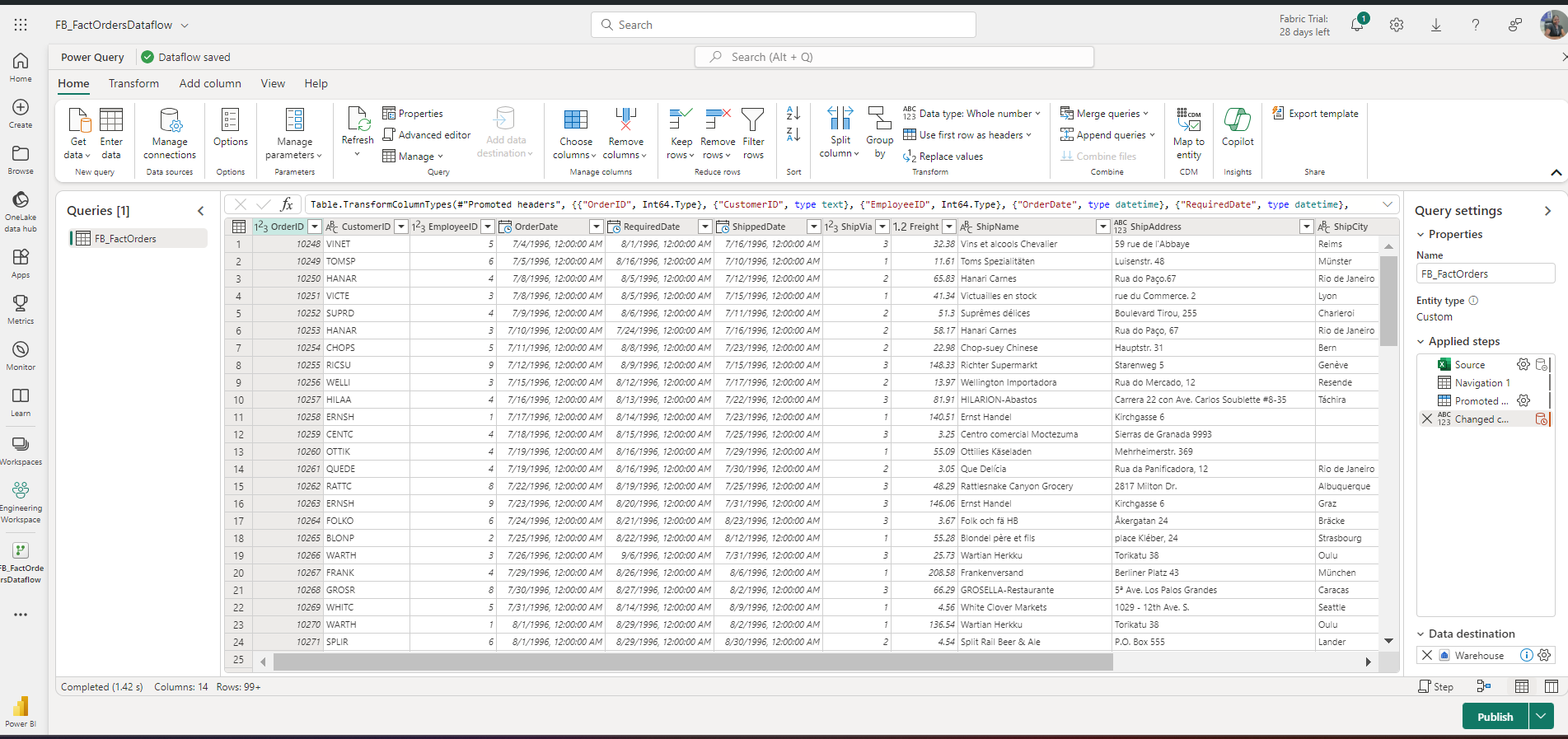
**Data Sources**

**Seven data tables were used for this project analysis;**

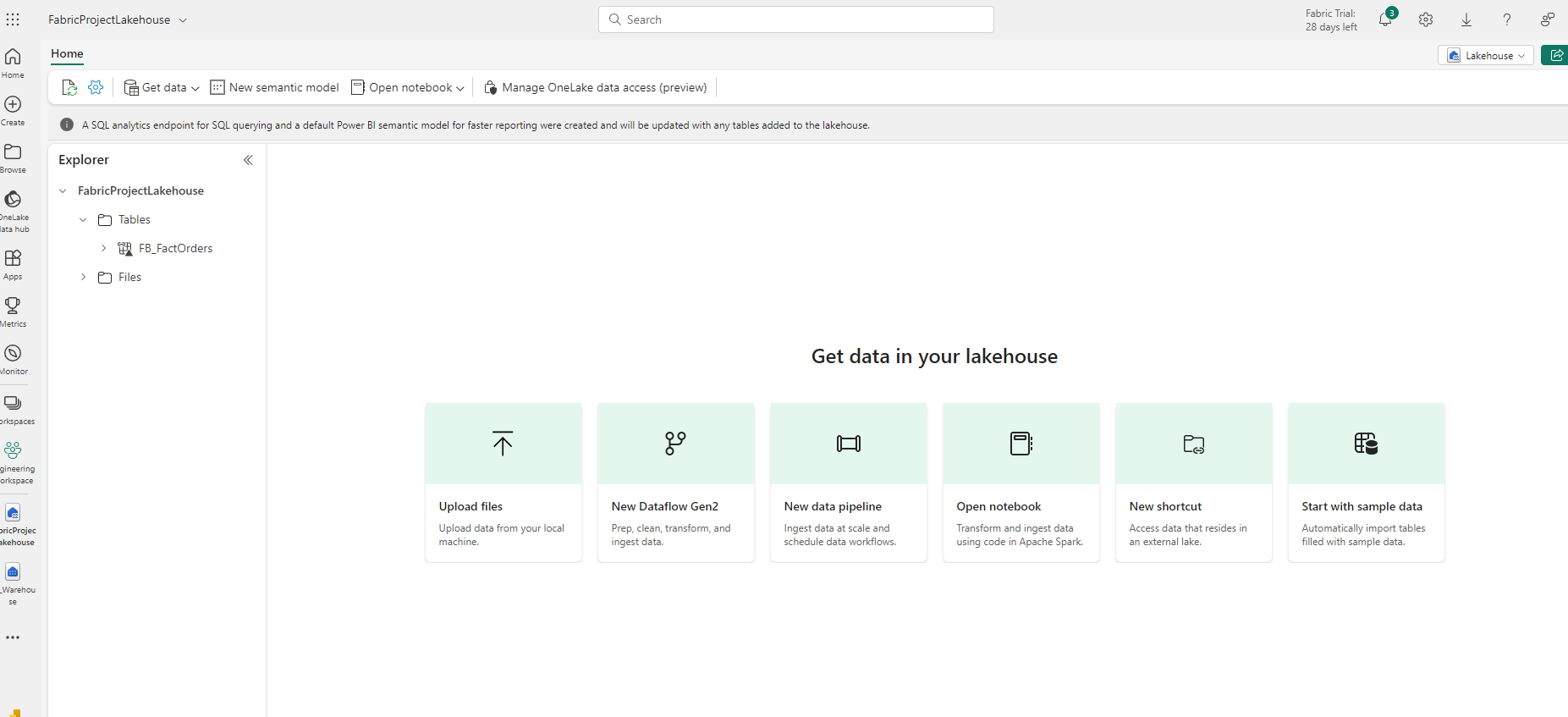
1. FactOrders data – CSV file
2. OrderDetails data – CSV file
3. Shippers data -CSV file
4. Product data – XLSX file
5. Categories data -CSV file
6. Date table – CSV file
7. Customers table – XLSX file
8. **Data Ingestion to Lakehouse**

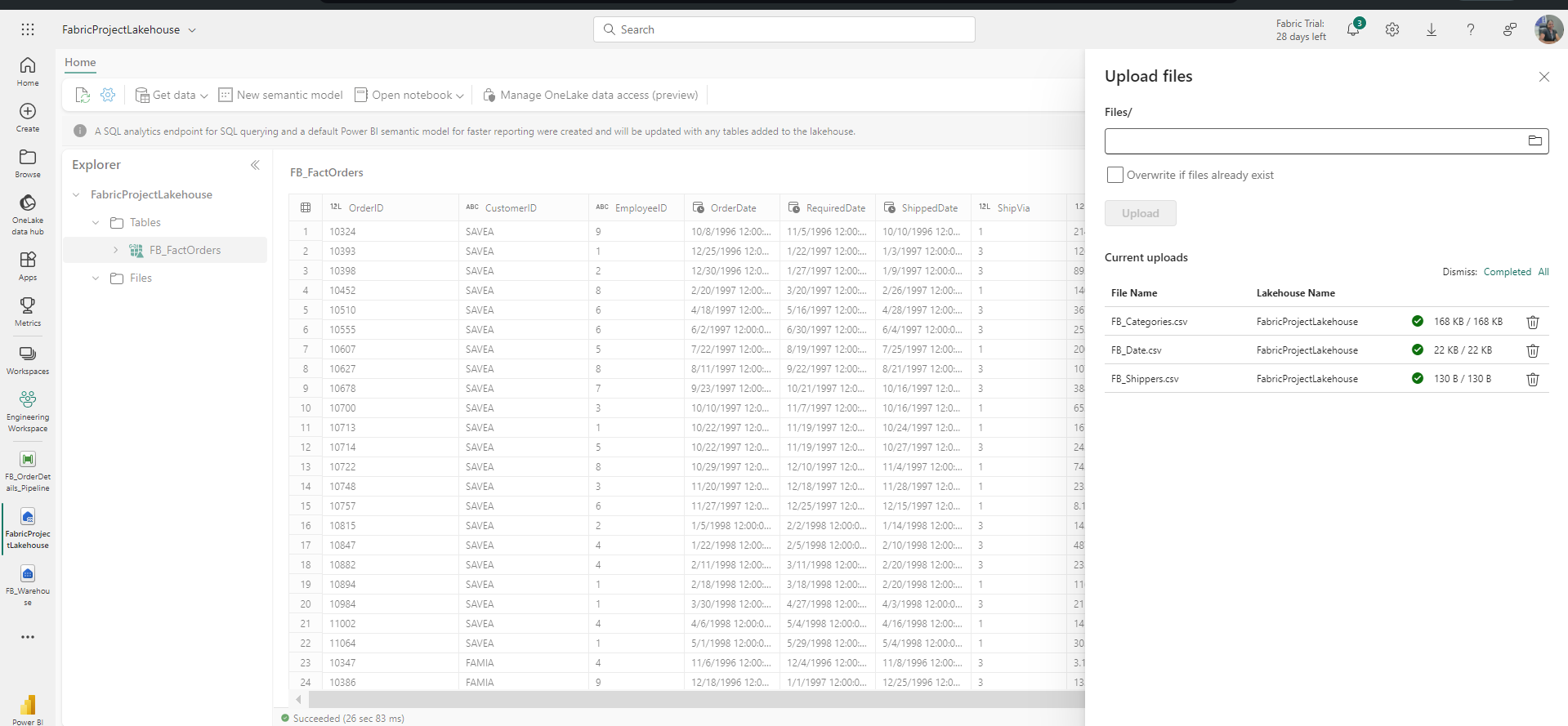
Four (4) different ways was used in getting the data to the created Lakehouse located in a workspace newly created on Microsoft fabric

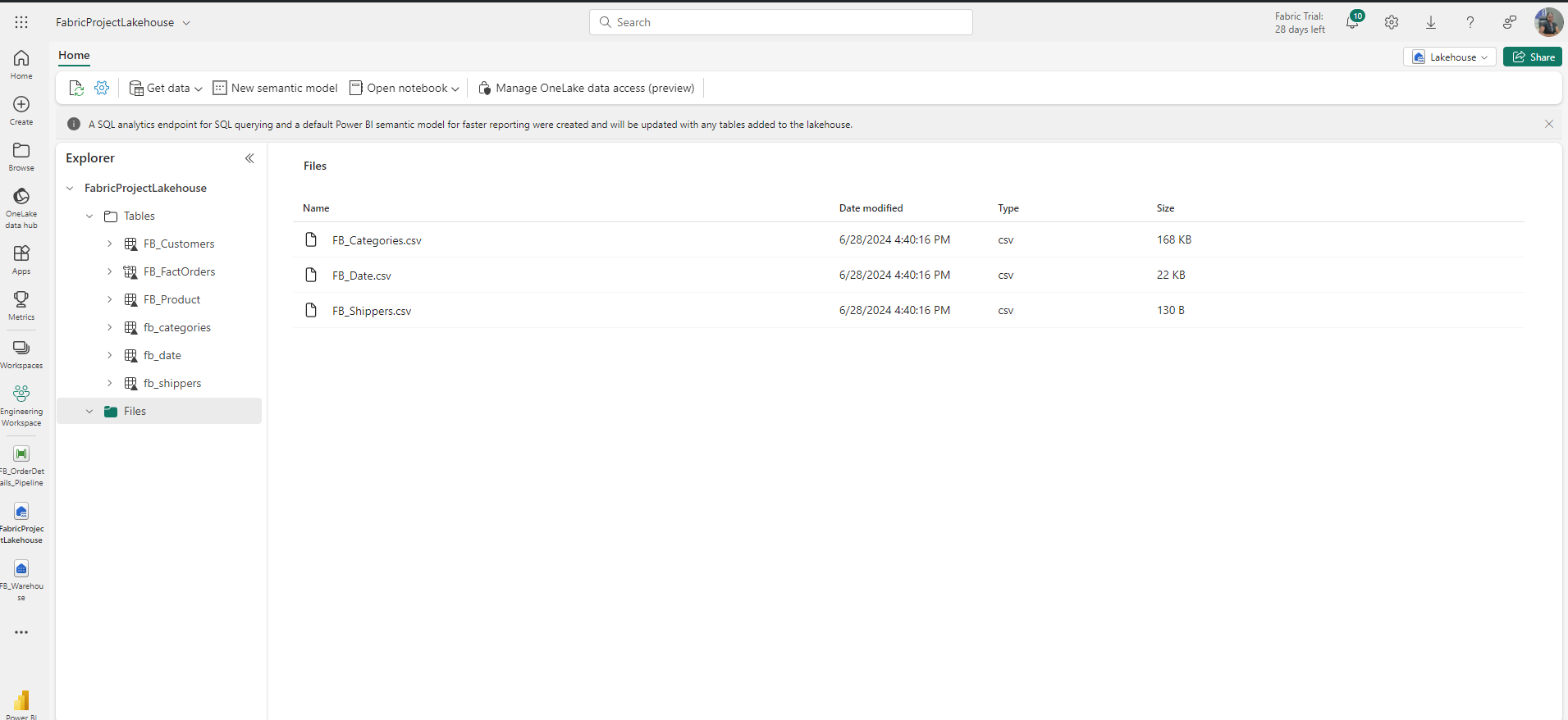
**Step 1: Shortcut from a warehouse –**

This method avoids the duplication of tables. To achieve this, a new data warehouse was created named “**FB\_Warehouse**” and dataflow Gen2 was used to ingest the “FactOrders” csv file into the warehouse using import from text/csv file button as seen below. The table was named **“FB\_FactOrders”**

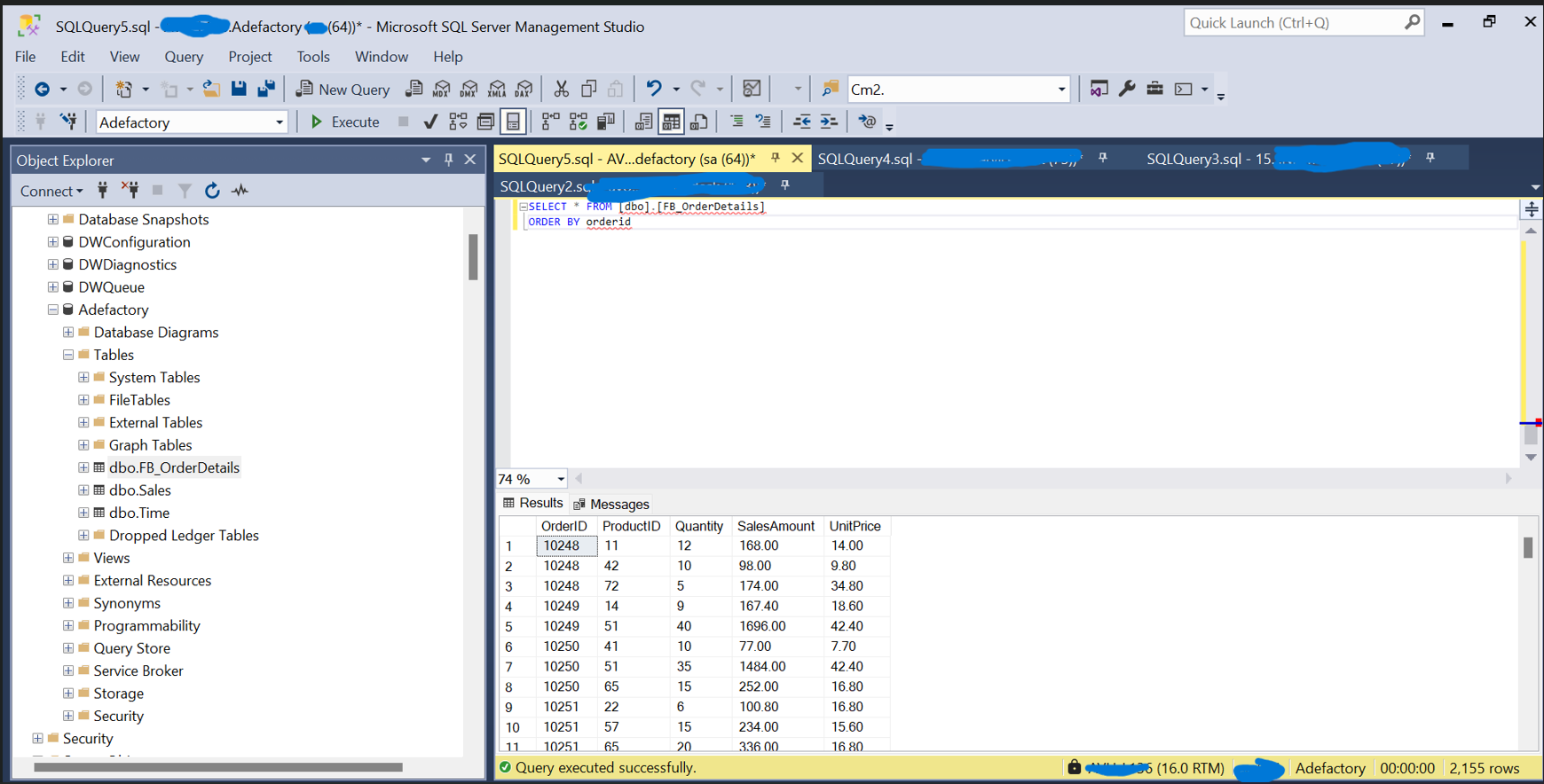
After this has been done, in my new Lakehouse, I clicked on new shortcut to get the data from the warehouse to the Lakehouse, selected the internal sources since the warehouse is located inside Fabric. I navigated my way to the warehouse destination to select the FB\_FactOrders for creation.

 **Step 2: Ingestion using upload files (Shippers, categories and Date CSV files)**

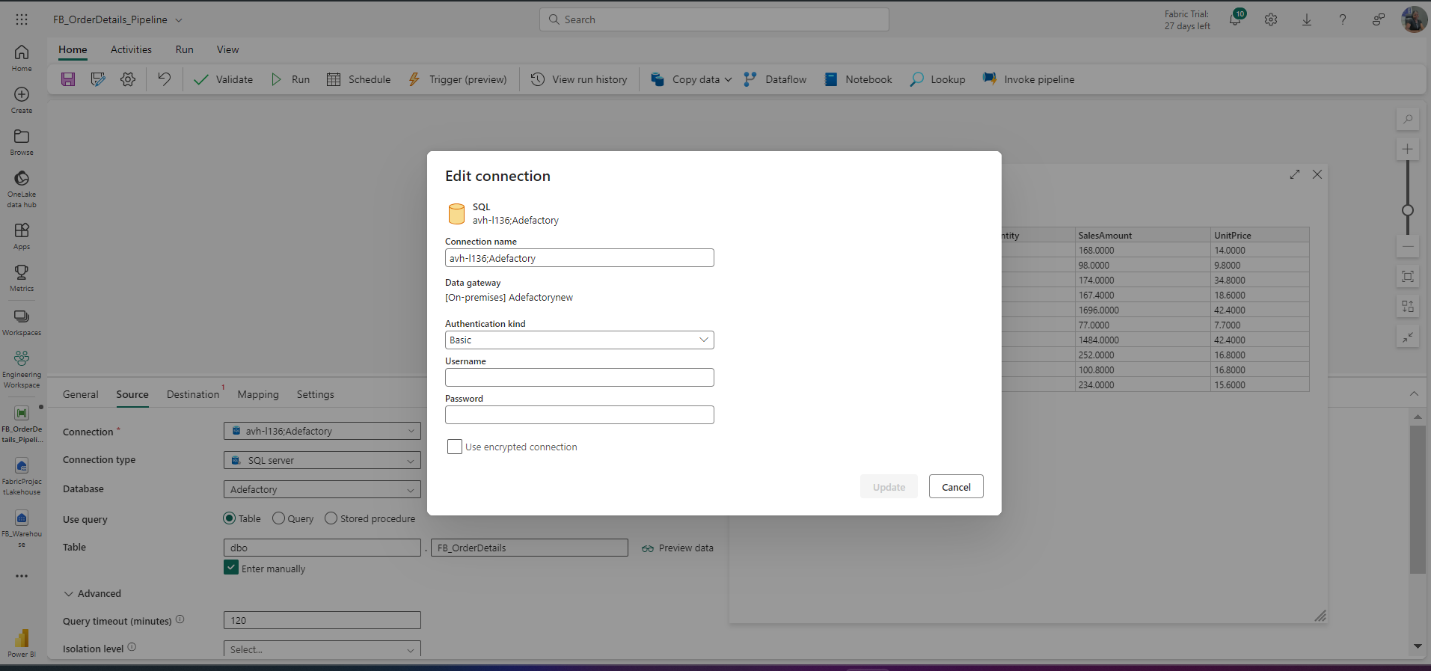
This method ensured all CSV dim files were uploaded from the local machine or rather can be uploaded on the OneDrive and selected from therein. However, note that it’s only CSV files that can be uploaded into the Lakehouse.   
  
After successful upload as files into the File section, they were loaded as delta tables into the Table section of the Lakehouse.

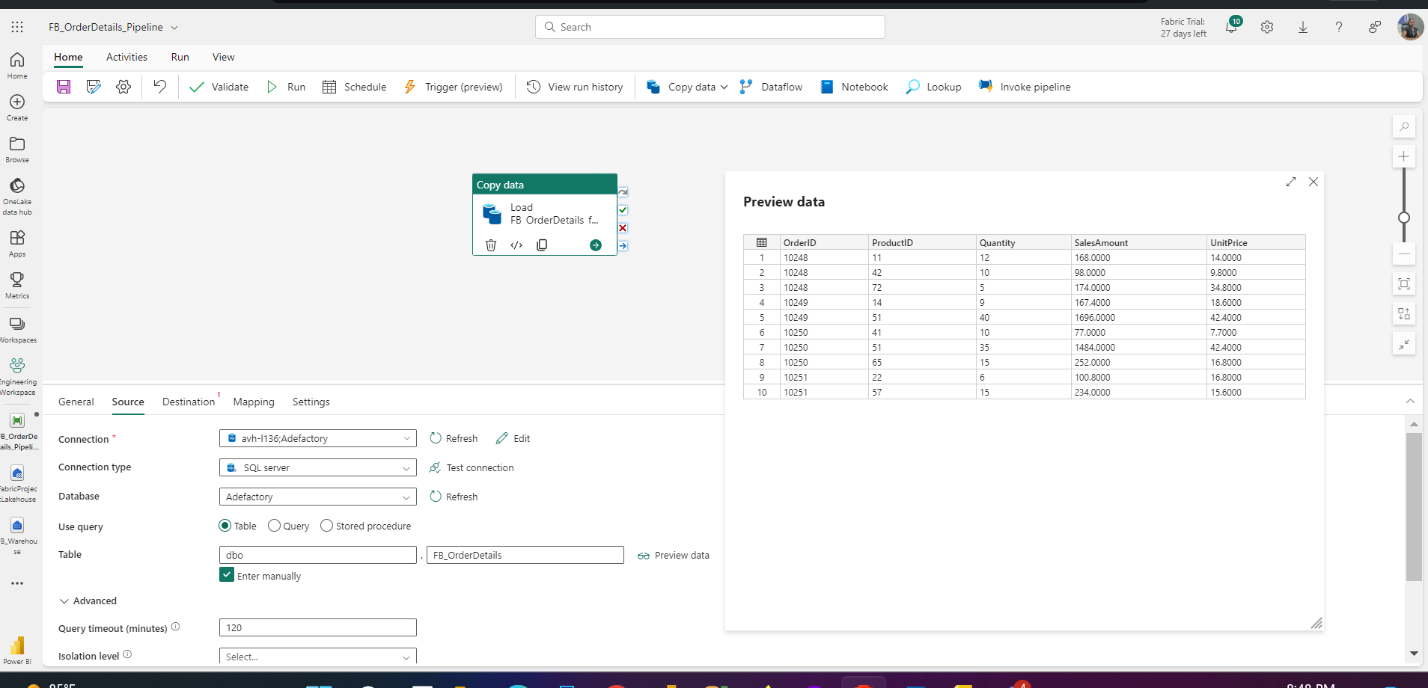
  
 **Step 3: Ingestion using data factory pipeline -**

This method was used to get the OrderDetails table from SQL database to the Lakehouse.

  
In the Lakehouse, I created a new pipeline and get data using copy data. I selected SQL database, and a new connection setting was configured using server name, database name and password as seen below.

After this has been done, I exited and selected the copy data assistant in pipeline, I ensured there is no encryption of data selected in the edit button pop up of the settings as shown below.

  
  
I selected the manual entry of table name as written in my SQL database after which I previewed the data. In the destination settings, A new table name was created as WW1\_OrderDetails In my Lakehouse and loaded which might take some minutes.



A screenshot of a computer

Description automatically generated

**Step 4: Ingestion using dataflow Gen2 for all XLSX file (Products and Customers)**

Since the xlsx file could not be uploaded and loaded directly to the Lakehouse, the get data option using dataflow Gen2 was used to ingest the customers and products table. This is also like the power query we have on Power BI desktop. I changed the data type of all columns after loading for transformation. I then ensured that the correct Lakehouse was selected as the destination of the tables. They were renamed as FB\_Products and WW1\_customers respectively.

**Transformations in Data Flow Gen2**

The FB\_OrderDetails, FB\_Product and FB\_Categories were loaded to dataflow for normalization. This was necessary as the categories table could only reference the product table using productid but was showing no relationship with the OrderDetails table. In other word, the categories table could not be used to filter the OrderDetails . Hence, the need for the normalizations.

**Steps taken**

1. The FB\_categories table was collapsed to the product table using merge queries option (Left join product to categories to add the categoryname, category description and picture url fields to the product table.

A screenshot of a computer

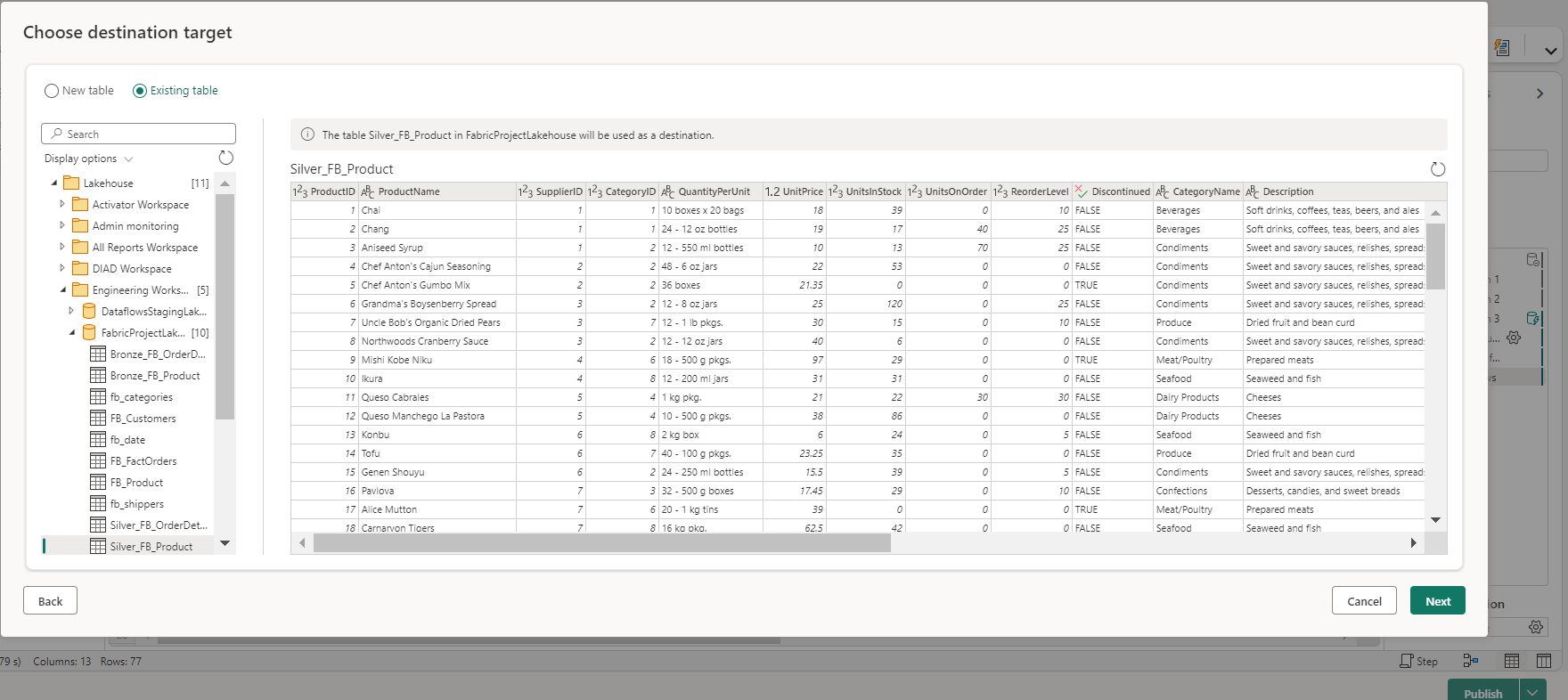
Description automatically generated

1. Merged the new FB\_Product to the FB\_OrderDetails table using Left join on productid. The categoryID was returned as a new field in the FB\_Orderdetails which was later reloaded to Lakehouse as Silver\_FB\_OrderDetails.

A screenshot of a computer

Description automatically generated

1. Ensure to select the data destination by selecting the lakehouse and click on existing table if already created, else write a new table name.

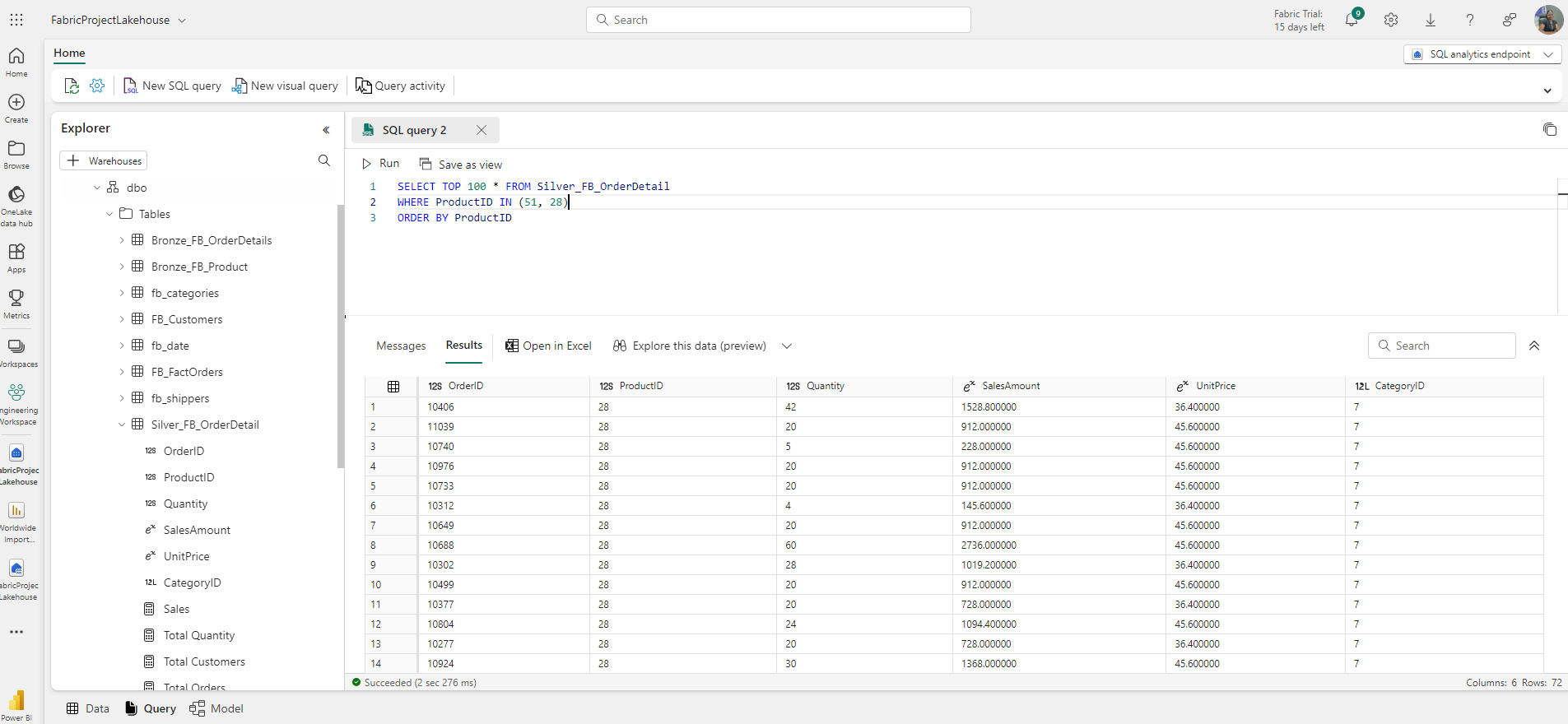


Likewise, click on replace existing table and save settings

A screenshot of a computer

Description automatically generated  
The new tables named Silver\_OrderDetail and Silver\_FB\_Product were loaded back to the Lakehouse

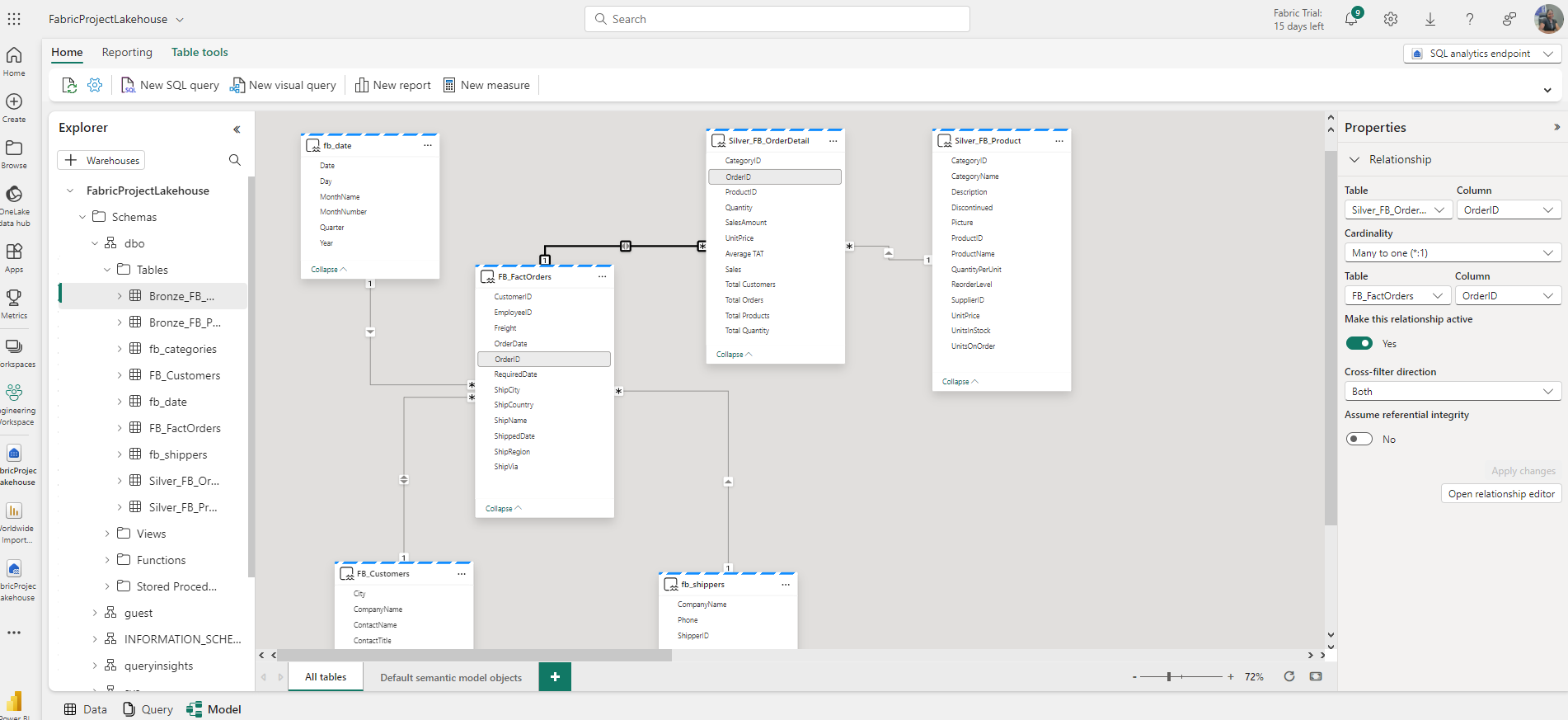
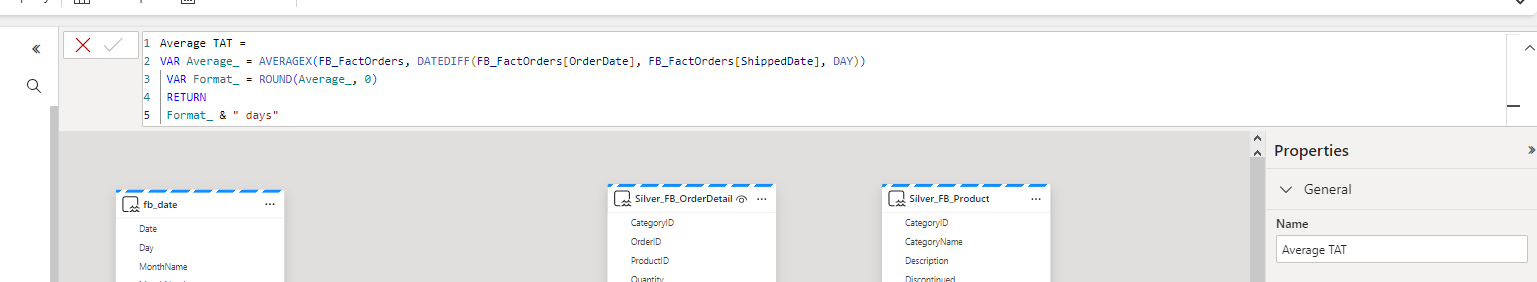
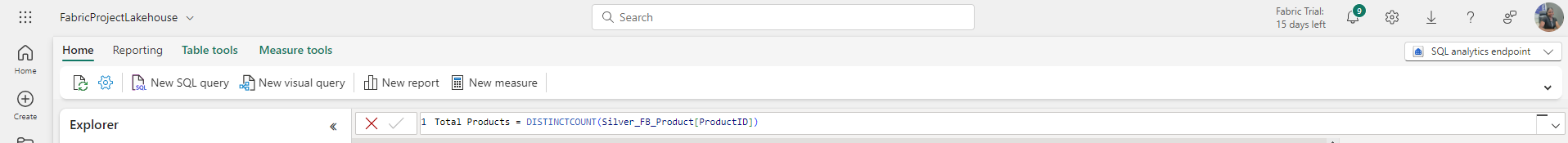
A screenshot of a computer

Description automatically generated  
  
You can choose to switch to query to run some queries from the table as seen below before modeling.  


**Data Modeling**

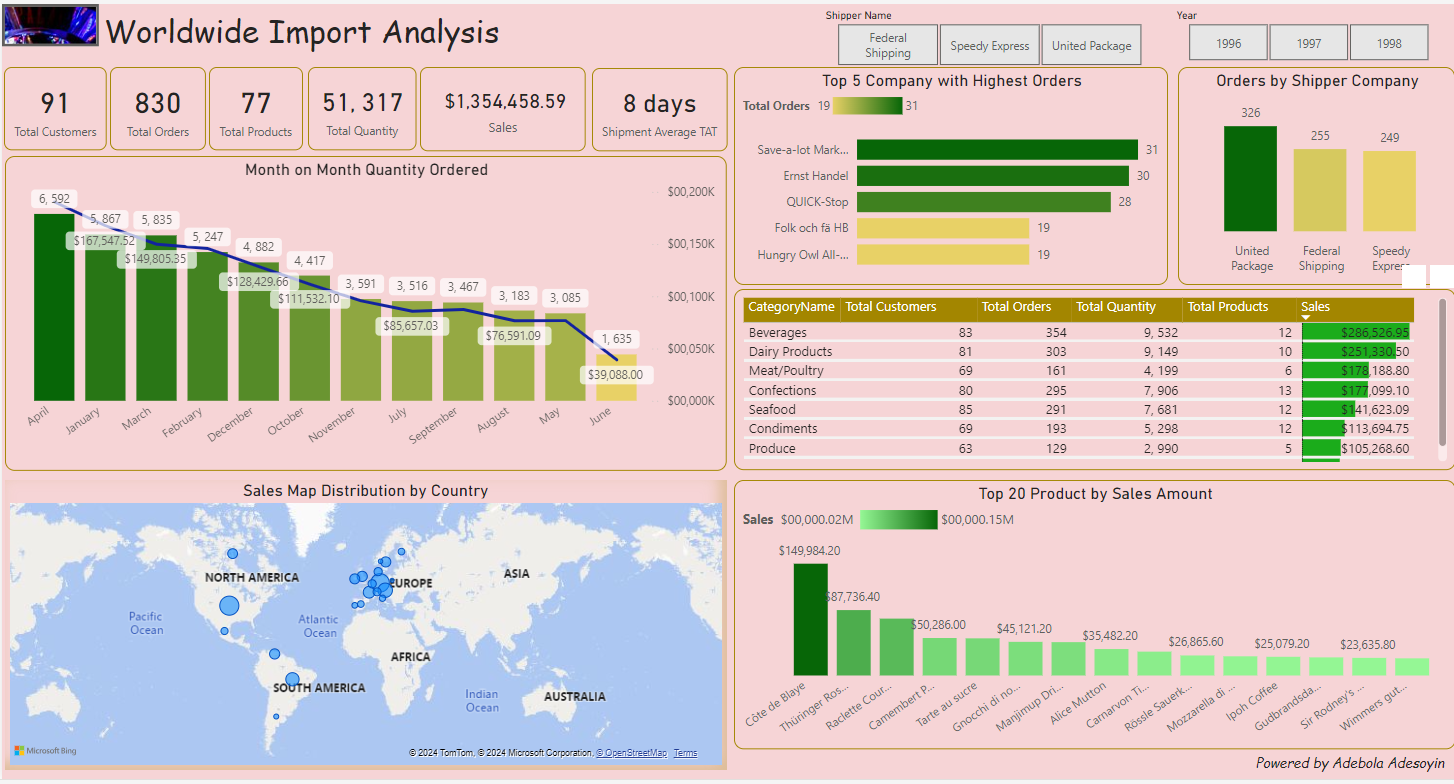
I switched to the SQL Analytics endpoint and clicked on the model tab where 5 relationships were built amongst the tables.

1. One to many from FB\_shippers to FB\_FactOrders on ShipperID key
2. One to many from FB\_Date to FB\_FactOrders on OrderDate key
3. One to many from FB\_FactOrders to Silver\_FB\_OrderDetails on OrderID key
4. One to many from Silver\_FB\_product to Silver\_FB\_OrderDetails on ProductID key
5. One to many from FB\_Customers to FB\_FactOrders on customerID key

  
With this relationship creation maintain a star schema structure. DAX (Data analysis expressions) Measures were created including Total sales, Total quantity, Total Orders, Turn around Time for delivery etc.  
  


Ensure that the measure button is not greyed out when trying to create one. If it is, then go to “**manage default semantic model”.** Ensure all tables to be used in your reporting are active and by selecting all and clicking okay. With this, new measure button shouldn’t be grey out any longer.

**Data Visualization**

Power BI was switched to start the creation of report for insights. I clicked on “New Report” and started with getting to know the summary of the data. The below report was created and can also be interacted with using this [link](https://app.powerbi.com/view?r=eyJrIjoiM2NlNDc5MTQtMzlhNS00NjEzLWEwMDAtMDQxNDY5NjkzYWVlIiwidCI6IjM3ZDc1MjFhLTUwNzktNDhhZi05MTMxLTRhYzJjYjZmMWUzYSIsImMiOjF9) .   
  


**Data Insights**

Always remember to schedule your dataflow refresh. This can be daily, weekly or depending on the time you would want it to refresh for daily report update.

**A screenshot of a computer

Description automatically generated  
  
  
*A screenshot of a computer

Description automatically generated***

***Thank you for reading.***