Microsoft Fabric in a Day Lab Manual – **Lab 5**

Author: Will Crayger
Email: wcrayger@lucidbi.co
LinkedIn: LinkedIn.com/in/willcrayger/
Course Material: GitHub.com/Lucid-Will/FabCon-EU-Zero-To-Hero-with-Fabric

Data Warehousing – Designing Your Data Warehouse Strategy in Fabric

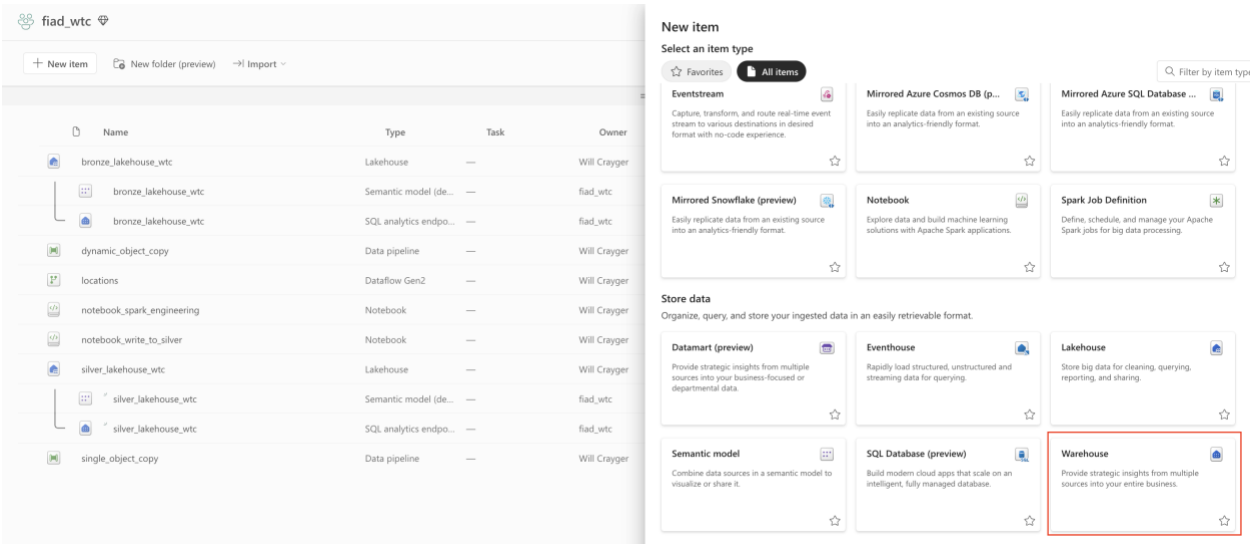
Introduction:

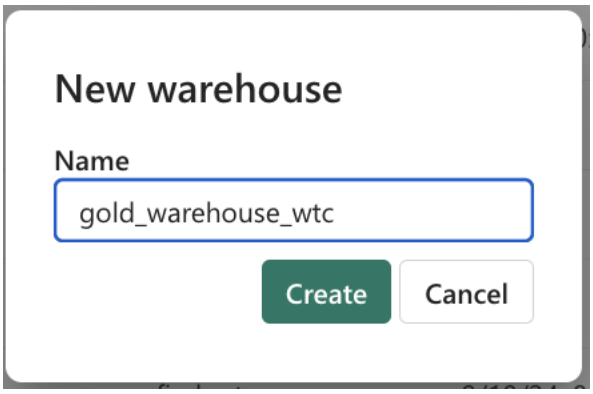
In this lab, we will be creating a new **Fabric Warehouse** to serve as the **Gold** layer in our medallion pattern. We will explore several techniques for loading data into the warehouse, including views, cross-database joins, and **CTAS** (Create Table As Select) statements.

Part 1: Creating the Warehouse

Creating the Warehouse: To begin, navigate to the **Warehouse** experience within your workspace. Click on **Warehouse** from the tiles across the top. Name your new warehouse **gold_warehouse_{your_initials}** and click **Create**.

You now have an empty **Fabric Warehouse** and have successfully completed **Part 1** of this lab.

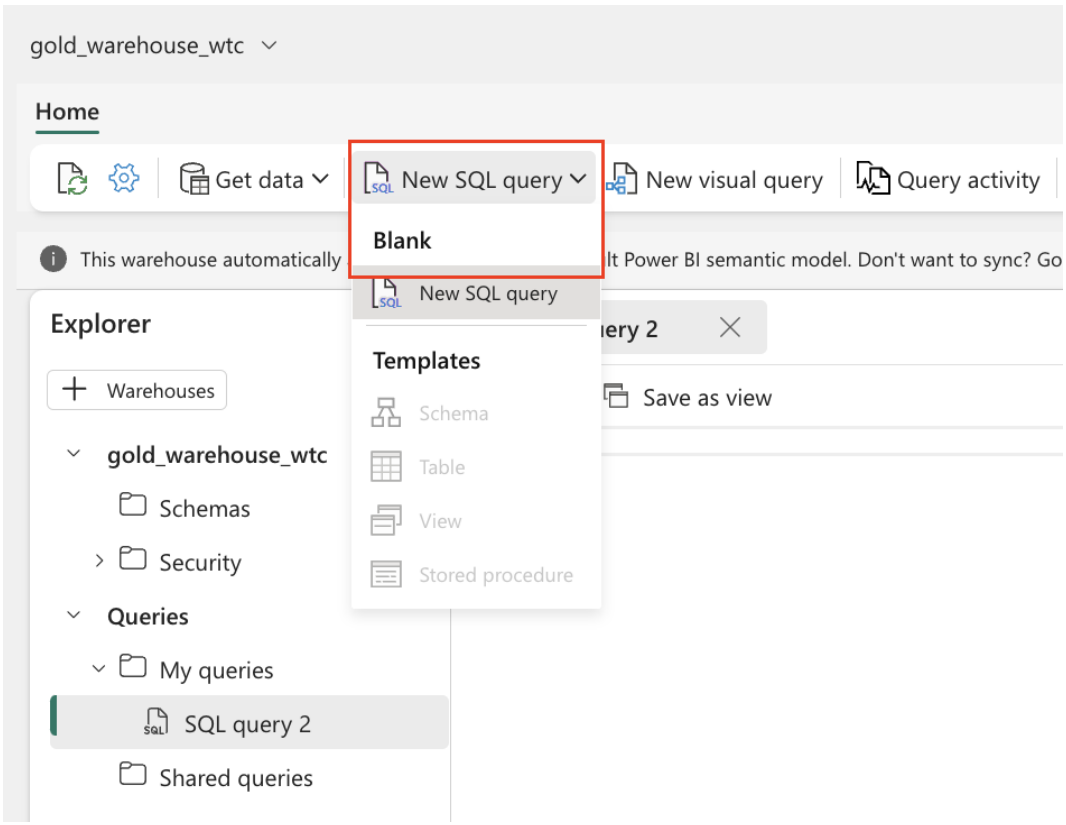




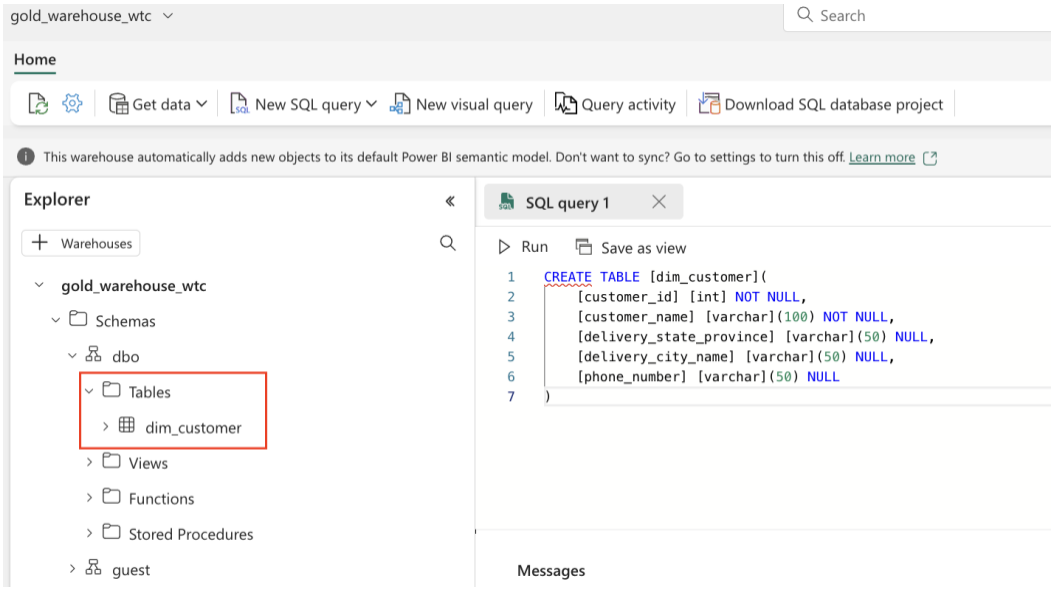
Part 2: Creating and Loading a Table Using CREATE TABLE INSERT INTO

Creating and Loading a Table: In this exercise, you'll use **TSQL** to create a table in the **Warehouse** and then load data into it.

To begin, navigate to the **Fabric Warehouse** from the workspace landing page. Click **New SQL Query** from the action ribbon at the top.

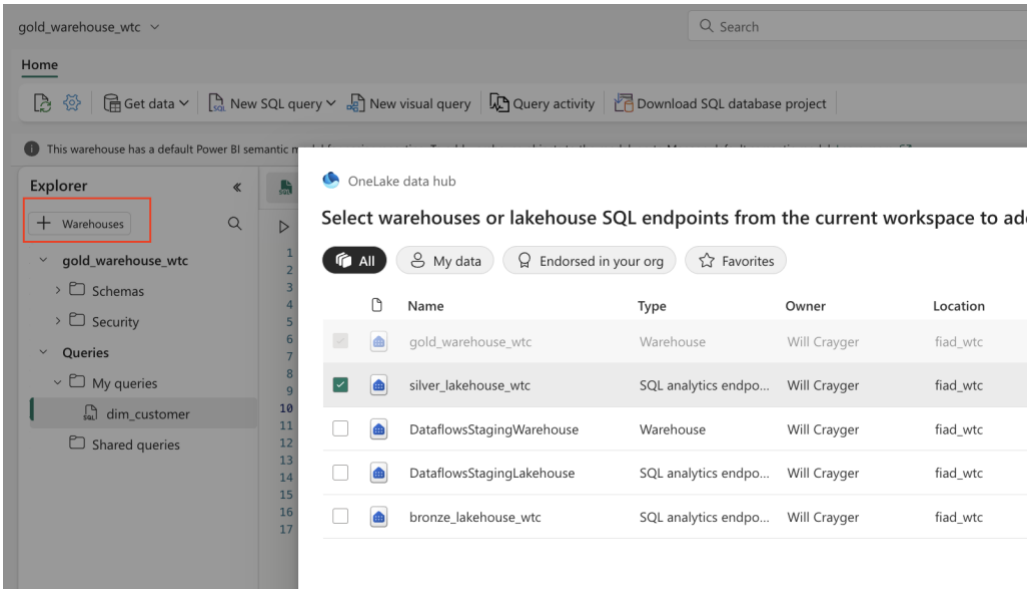


Write the **CREATE TABLE** statement shown in the image below, then click **Run**. You can also refer to the lab file **Create and Load Customer Dimension** for reference.

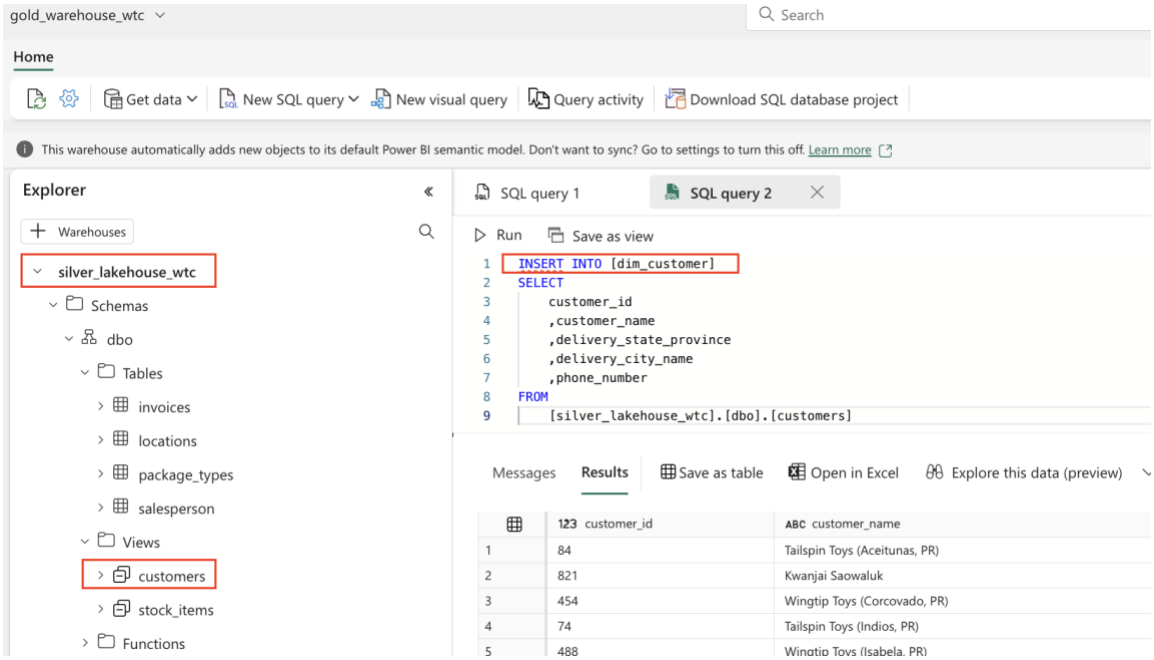


Once the query is executed, the schema of the warehouse will update to reflect the newly created empty table.

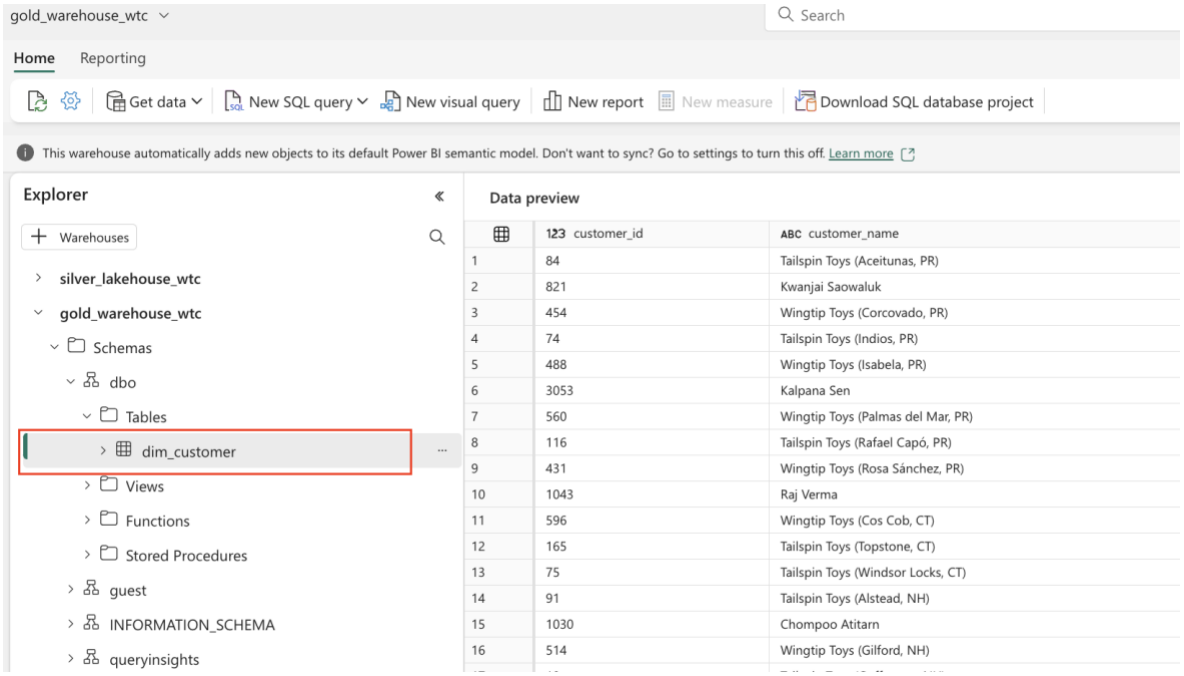
Next, click **New SQL Query** again from the top ribbon to open a blank query. Click the **+ Warehouses** button in the **Explorer**, check the box for the **silver_lakehouse** from Lab 4, and click **Connect**.



In the query window, write the following **INSERT INTO** statement or refer to the lab file **Create and Load Customer Dimension**. Once the query is complete, click **Run**.



Finally, click on the **dim_customer** table in the **Fabric Warehouse Explorer** to preview the newly loaded data.



You have now successfully completed **Part 2** of this lab.

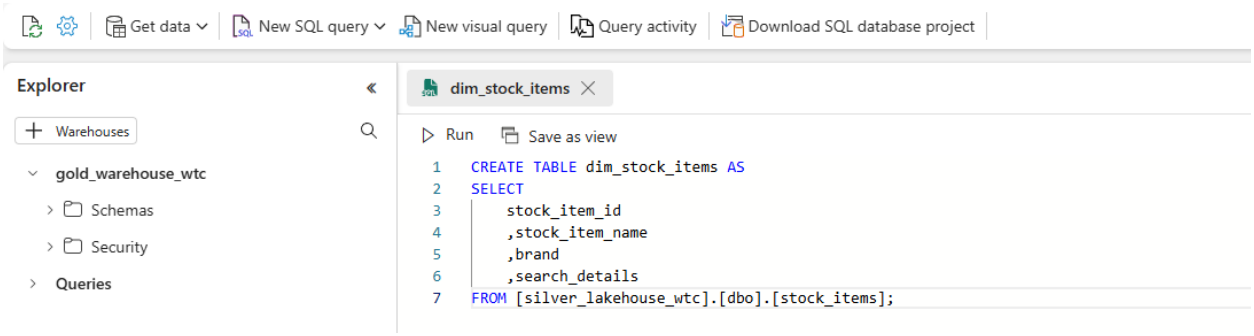
Part 3: Creating and Loading a Table Using CTAS (CREATE TABLE AS SELECT)

Combining Create and Load with CTAS: In your **Fabric Warehouse**, you can combine the table creation and data loading process using the **CREATE TABLE AS SELECT** (CTAS) functionality. If needed, refer to the supporting **Create and Load Stock Items Dimension** file from the lab coursework.

To start, navigate to the **Fabric Warehouse** from the workspace landing page. Click **New SQL Query** from the action ribbon at the top.

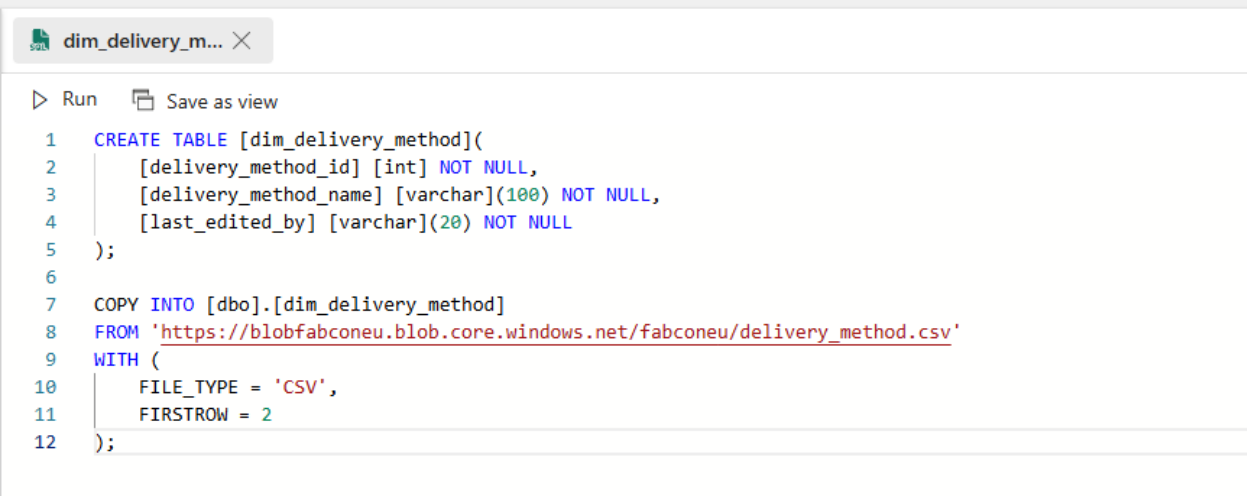
In the query window, write the following **SQL** statement. Use the **Click + Drag** functionality to populate the three-part table identifier from the previous lab, then click **Run**.

Once executed, the **dim_stock_items** table will be added to the schema of your **Fabric Warehouse** and will be available for previewing.



Part 4: Creating and Loading a Table Using CREATE and COPY INTO

Using CREATE and COPY INTO: Access the **Create and Load Delivery Method Dimension** file from the coursework. The COPY INTO structure allows you to **COPY INTO FROM** using supported data formats from an external location such as a blob storage.



Part 5: Creating Remaining Objects

Completing the Lab with Provided CTAS Code: For the final part of this lab, we've provided the code to create the remaining tables using the **CREATE TABLE AS SELECT (CTAS)** method. You will need to use the supporting file to copy and paste the SQL statements into your **Fabric Warehouse** query editor.

To start, copy the SQL queries from the **Fabric Warehouse CTAS Statements** document and paste them into a new query in the **Fabric Warehouse** query editor. Replace the Lakehouse names with the correct name of your **Silver Lakehouse** artifacts, then click **Run**.

remaning_ctas

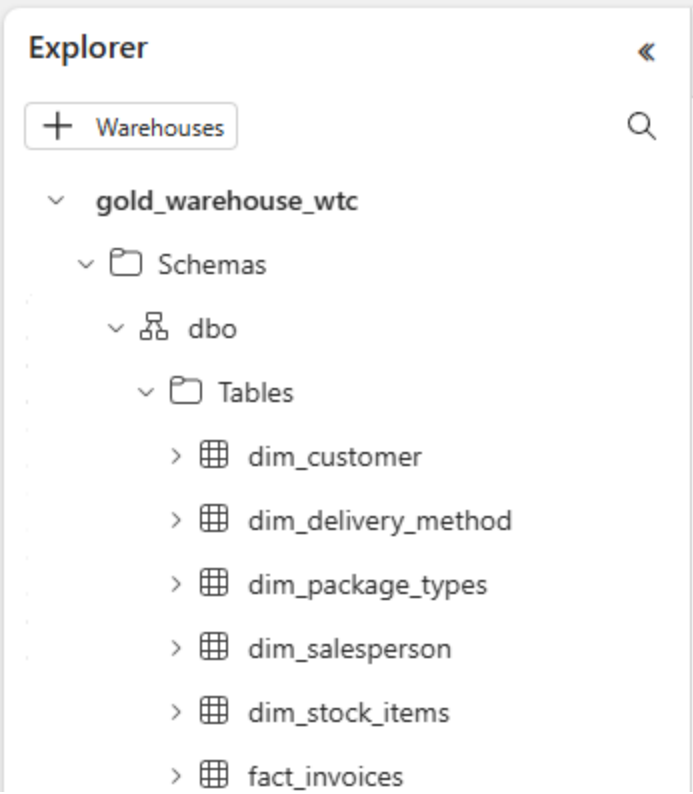
Run Save as view

```sql
CREATE TABLE dim_package_types AS
SELECT
    package_type_id
    ,package_type_name
    ,last_edited_by
    ,valid_from
    ,valid_to
FROM [silver_lakehouse_wtc].[dbo].[package_types];


CREATE TABLE dim_salesperson AS
SELECT
    PersonID                person_id
    ,FullName               salesperson_name
    ,PhoneNumber            phone_number
FROM [silver_lakehouse_wtc].[dbo].[salesperson];

CREATE TABLE fact_invoices AS
SELECT
    invoice_line_id
    ,invoice_id
    ,customer_id
    ,delivery_method_id
    ,salesperson_id
    ,stock_item_id
    ,package_type_id
    ,invoice_date
    ,quantity
    ,unit_price
    ,tax_rate
FROM [silver_lakehouse_wtc].[dbo].[invoices];
```

Once the queries have executed, you should have a total of 6 tables in your model.



You have now successfully completed **Lab 5**.