

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

Author: Will Crayger

Email: [wcrayger@lucidbi.co](mailto:wcrayger@lucidbi.co)

LinkedIn: [LinkedIn.com/in/willcrayger/](https://www.linkedin.com/in/willcrayger/)

Course Material: [GitHub.com/Lucid-Will/FabCon-EU-Zero-To-Hero-with-Fabric](https://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.

The screenshot displays the Microsoft Fabric workspace interface. On the left, a table lists existing items:

Name	Type	Task	Owner
bronze_lakehouse_wtc	Lakehouse	—	Will Crayger
bronze_lakehouse_wtc	Semantic model (de...)	—	fiad_wtc
bronze_lakehouse_wtc	SQL analytics endpo...	—	fiad_wtc
dynamic_object_copy	Data pipeline	—	Will Crayger
locations	Dataflow Gen2	—	Will Crayger
notebook_spark_engineering	Notebook	—	Will Crayger
notebook_write_to_silver	Notebook	—	Will Crayger
silver_lakehouse_wtc	Lakehouse	—	Will Crayger
silver_lakehouse_wtc	Semantic model (de...)	—	fiad_wtc
silver_lakehouse_wtc	SQL analytics endpo...	—	fiad_wtc
single_object_copy	Data pipeline	—	Will Crayger

On the right, the 'New item' dialog is open, showing a grid of item types. The 'Warehouse' item is highlighted with a red border. The 'Warehouse' item description is: 'Provide strategic insights from multiple sources into your entire business.'

## New warehouse

Name

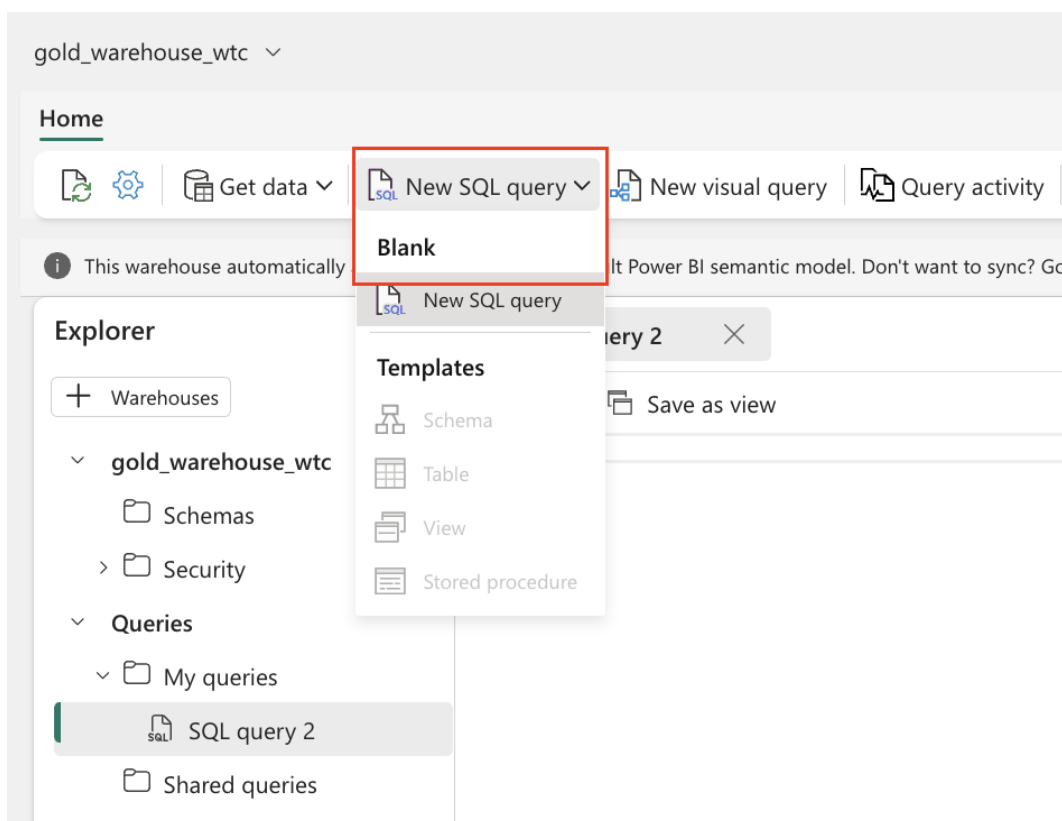
gold\_warehouse\_wtc

CreateCancel

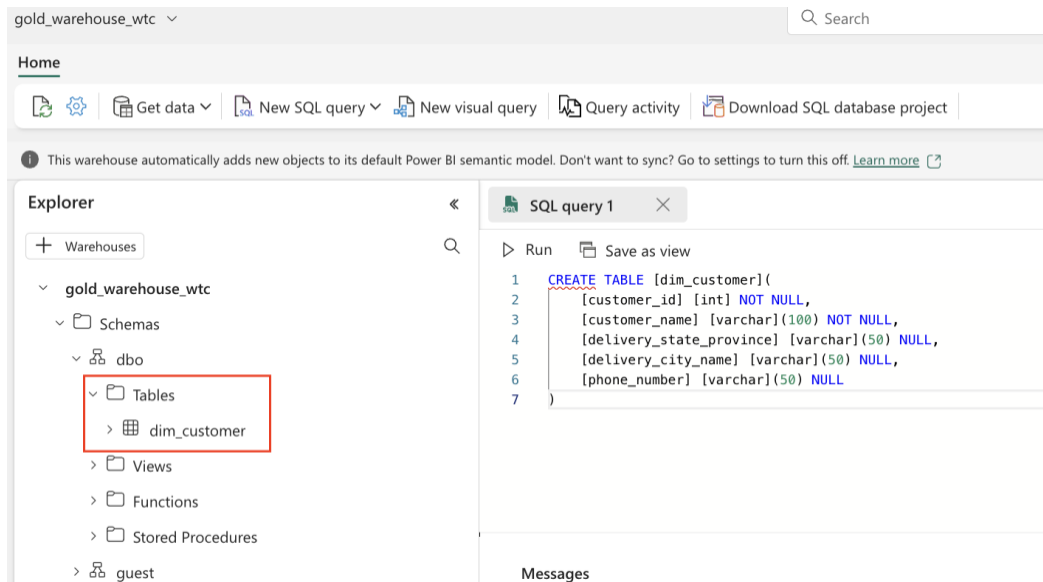
## Part 2: Creating and Loading a Table Using CREATE TABLE INSERT INTO

**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.

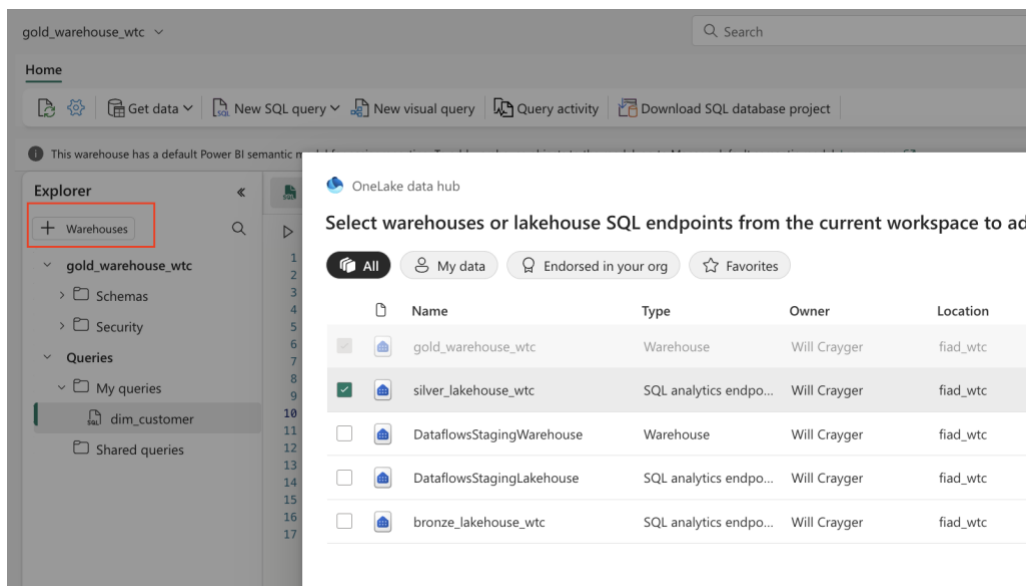


Write the **CREATE TABLE** statement shown in the image below, then click **Run**. You can also refer to the lab file **Create and Load Customer Dimension** for reference.



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**.

The screenshot shows the Microsoft Fabric interface. In the Explorer pane on the left, the 'silver\_lakehouse\_wtc' warehouse is selected, and the 'customers' table is highlighted. The SQL query window on the right contains the following SQL statement:

```
1 INSERT INTO [dim_customer]
2 SELECT
3     customer_id
4     ,customer_name
5     ,delivery_state_province
6     ,delivery_city_name
7     ,phone_number
8 FROM
9     [silver_lakehouse_wtc].[dbo].[customers]
```

The Results tab shows the following data:

	customer_id	customer_name
1	84	Tailspin Toys (Aceitunas, PR)
2	821	Kwanjai Saowaluk
3	454	Wingtip Toys (Corcovado, PR)
4	74	Tailspin Toys (Indios, PR)
5	488	Winatip Toys (Isabela, PR)

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

The screenshot shows the Microsoft Fabric interface. In the Explorer pane on the left, the 'gold\_warehouse\_wtc' warehouse is selected, and the 'dim\_customer' table is highlighted. The Data preview window on the right shows the following data:

	customer_id	customer_name
1	84	Tailspin Toys (Aceitunas, PR)
2	821	Kwanjai Saowaluk
3	454	Wingtip Toys (Corcovado, PR)
4	74	Tailspin Toys (Indios, PR)
5	488	Wingtip Toys (Isabela, PR)
6	3053	Kalpana Sen
7	560	Wingtip Toys (Palmas del Mar, PR)
8	116	Tailspin Toys (Rafael Capó, PR)
9	431	Wingtip Toys (Rosa Sánchez, PR)
10	1043	Raj Verma
11	596	Wingtip Toys (Cos Cob, CT)
12	165	Tailspin Toys (Topstone, CT)
13	75	Tailspin Toys (Windsor Locks, CT)
14	91	Tailspin Toys (Alstead, NH)
15	1030	Chompoo Atitarn
16	514	Wingtip Toys (Gifford, NH)

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

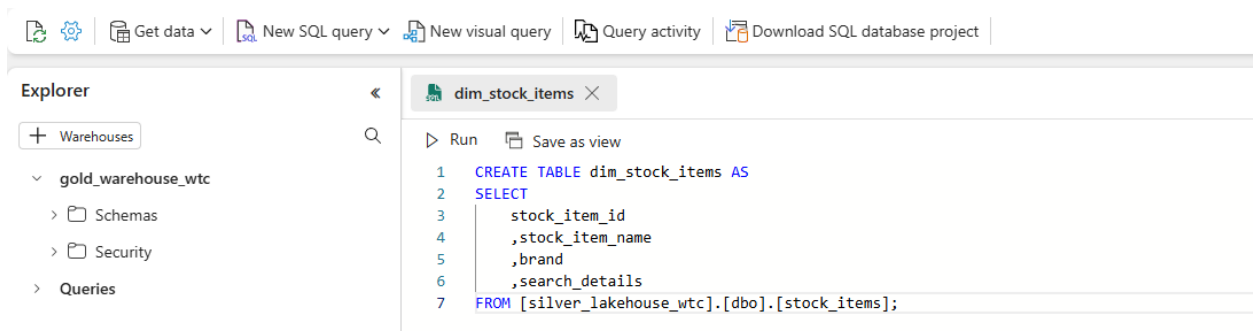
### Part 3: Creating and Loading a Table Using CTAS (CREATE TABLE AS SELECT)

**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.



```
1 CREATE TABLE [dim_delivery_method](
2     [delivery_method_id] [int] NOT NULL,
3     [delivery_method_name] [varchar](100) NOT NULL,
4     [last_edited_by] [varchar](20) NOT NULL
5 );
6
7 COPY INTO [dbo].[dim_delivery_method]
8 FROM 'https://blobfabconeue.blob.core.windows.net/fabconeue/delivery_method.csv'
9 WITH (
10     FILE_TYPE = 'CSV',
11     FIRSTROW = 2
12 );
```

## Part 5: Creating Remaining Objects

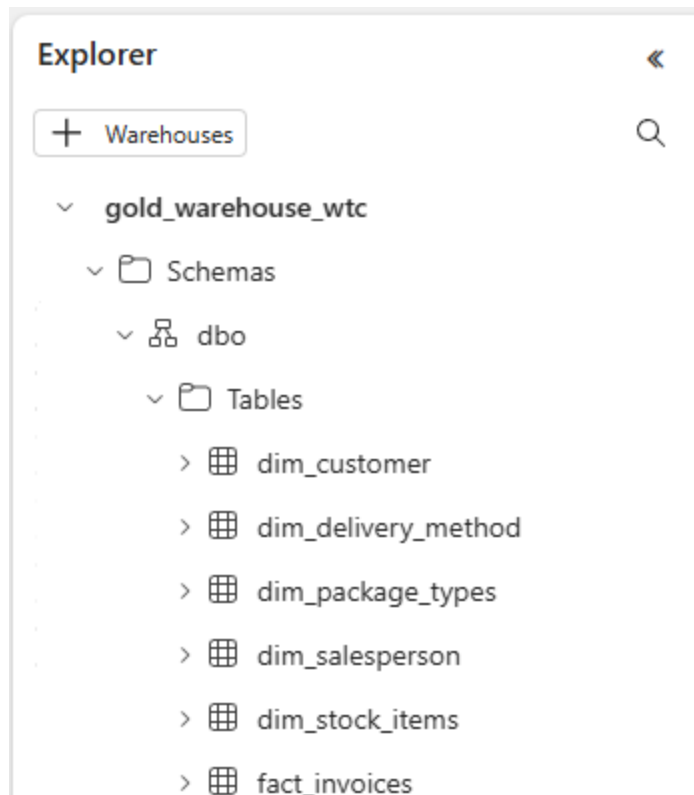
**Completing the Lab with Provided CTAS Code:** For the final part of this lab, we've provided the code to create the remaining tables using the **CREATE TABLE AS SELECT (CTAS)** method. You will need to use the supporting file to copy and paste the SQL statements into your **Fabric Warehouse** query 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 X
Run Save as view
1 CREATE TABLE dim_package_types AS
2 SELECT
3     package_type_id
4     ,package_type_name
5     ,last_edited_by
6     ,valid_from
7     ,valid_to
8 FROM [silver_lakehouse_wtc].[dbo].[package_types];
9
10
11 CREATE TABLE dim_salesperson AS
12 SELECT
13     PersonID          person_id
14     ,FullName          salesperson_name
15     ,PhoneNumber       phone_number
16 FROM [silver_lakehouse_wtc].[dbo].[salesperson];
17
18 CREATE TABLE fact_invoices AS
19 SELECT
20     invoice_line_id
21     ,invoice_id
22     ,customer_id
23     ,delivery_method_id
24     ,salesperson_id
25     ,stock_item_id
26     ,package_type_id
27     ,invoice_date
28     ,quantity
29     ,unit_price
30     ,tax_rate
31 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**.