Microsoft Fabric in a Day Lab Manual – **Lab 4**

**Author**: Will Crayger

**Email**: [[wcrayger@lucidbi.co](mailto: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 Engineering – Data Transformation and Engineering in Fabric

## Introduction

In this lab, you’re going to create a **Silver** Lakehouse and begin loading data from the **Bronze** Lakehouse that was created in **Lab 1**. As you proceed through the lab, please note that you’re working in a case-sensitive environment.

## Part 1: Creating the Silver Lakehouse

**Creating a Silver Lakehouse:** The purpose of creating a Silver Lakehouse is to simulate the **medallion data storage pattern**. The Silver Lakehouse will provide a separation from the Bronze layer, allowing you to start cleansing and shaping the data.

To begin, select **New** **item** from your Workspace home page. Scroll down and select **Lakehouse** from the item list. From the tiles at the top of the page, select **Lakehouse**. Name your lakehouse **silver\_lakehouse\_<your\_initials>** and click **Create**.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Creating a Shortcut to Azure Data Lake Gen 2:** To access semi-curated invoice data, you’ll need to create a shortcut to an Azure Data Lake Gen 2 resource provided as a part of this coursework.

From your Lakehouse, click **Get data**, click **New shortcut**, and then select **Azure Data Lake Storage Gen2**. Click the **create new connection** radial button and set the Authentication kind to **Shared Access Signature (SAS)**. Using the connection details provided in the Shortcut Connection Details file located in the Lab 4 coursework folder, complete the **URL** and **SAS token** fields. Click **Next**. Check the tickbox beside the **invoices** folder and click **Next**. Click **Create**.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Part 2: Creating a New Notebook

**Mounting the Notebook:** From your **Silver Lakehouse**, click **Open Notebook** and select **New Notebook**. At the top-left of your browser, click the name to rename the notebook to **notebook\_write\_to\_silver**.

Next, click **Lakehouses** from the **Explorer** blade. Click the **+ Lakehouse** button, check the **Existing Lakehouse** radial button, and click **Add**. Select the **Bronze Lakehouse** and click **Add** again.

The tables from the **Bronze Lakehouse** will now be visible in the notebook, ready for transformation.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Writing Data to Silver:** Click and drag **application\_people** from the **Explorer** to the first cell in the notebook. Remove the **LIMIT 1000** clause from the cell and press **Shift + Enter** to execute the cell. This action will start a Spark session using the default **single\_node** Spark Pool that was created earlier and then render a preview of the results.

To filter the dataframe to individuals identified as a salesperson, copy and paste the following command into the blank cell at the bottom of your notebook. Press **Shift + Enter** to execute the cell

**df\_salesperson = df.filter('IsSalesperson = true')**

**display(df\_salesperson)**

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Next, to write the new dataframe to the **Silver Lakehouse**, copy and paste the following command into the blank cell at the bottom of your notebook.

Included in the code snippet is the creation and usage of a **silver\_lakehouse** variable. Update the value of the variable with your **silver\_lakehouse.**

After updating the variable, press **Shift + Enter** to execute the cell. To confirm the results were successfully written, switch to the **Silver Lakehouse** from the **Notebook Explorer**.

**# Assign silver lakehouse variable value (e.g. silver\_lakehouse = 'silver\_lakehouse\_wtc')**

**silver\_lakehouse = 'silver\_lakehouse\_wtc'**

**# Write dataframe to silver**

**df\_salesperson.write.format('delta').mode('overwrite').saveAsTable(f'{silver\_lakehouse}.salesperson')**

A screenshot of a computer code

Description automatically generated

## Part 3: Importing a Notebook

**Importing a Notebook:** In this section, the **Notebook** will serve as the primary documentation source, complete with comments for visibility and clarity, along with code examples that can be referenced to complete each exercise. The notebook used for these exercises is the **notebook\_spark\_engineering.ipynb** file, which was provided as part of the course material in the Lab 4 folder.

To set up the environment and launch the notebook, follow these steps:

Navigate to the **Data Engineering** landing page of **Fabric**. Click **Import Notebook** from the actions across the top. Then click **Upload**, navigate to the location where the course files were saved, select the **notebook\_spark\_engineering.ipynb**, and click **Open**. After this, return to the workspace home page, and you should see the notebook listed as one of the workspace artifacts.

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Attach the Notebook to Bronze and Silver Lakehouses:** Navigate back to your workspace and click the notebook to open it. In the **Add Lakehouse** blade, click **Add**. Select **Existing Lakehouse** and click **Add** again. Tick both **Bronze** **Lakehouse** and **Silver Lakehouse**, then click **Add**.

The remainder of this section will be completed using the provided notebook. The notebook itself will serve as the documentation to complete part 4.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Part 4: Data Engineering via Lakehouse SQL Endpoint

**Using the Lakehouse SQL Endpoint:** We can leverage the **Lakehouse SQL Analytics Endpoint** to transform and prepare our data using T-SQL. In this section, you’ll use cross-database joins to query data from the **Bronze Lakehouse** and **create a view** in the **Silver Lakehouse**.

To begin, navigate to your workspace landing page and select the **SQL analytics endpoint** located under your **Silver Lakehouse**. Click the **+ Warehouses** button at the top of the **Explorer**. Then, check the box for your **Bronze Lakehouse** and click **Confirm**. You should now see the schema from the **Bronze Lakehouse** available in the Explorer.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Next, select **New SQL Query** from the action ribbon at the top. Paste in the below T-SQL to query the **warehouse\_stockitems** table from the **Bronze Lakehouse** or **Drag + Drop** the table to the query and begin writing the below query.

**SELECT**

**StockItemID AS stock\_item\_id**

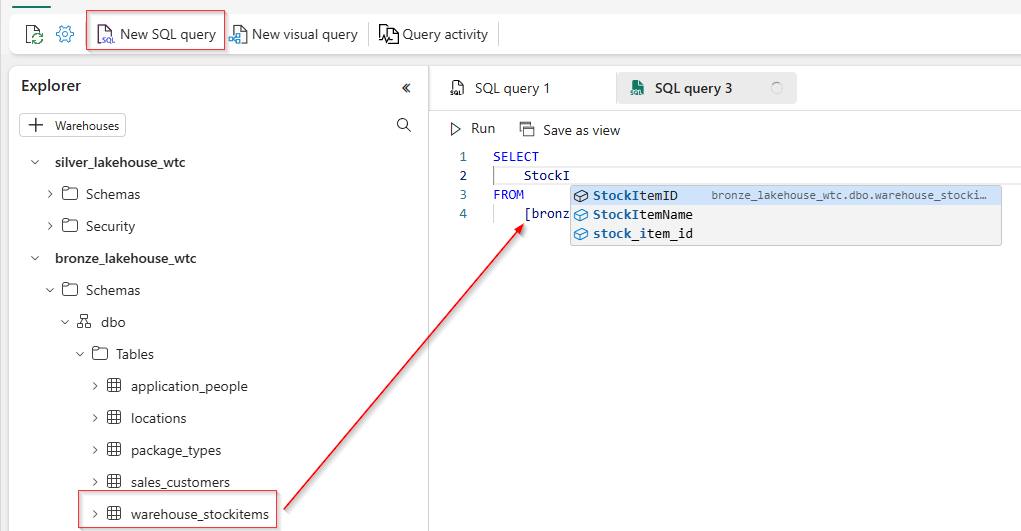
**,StockItemName AS stock\_item\_name**

**,Brand AS brand**

**,SearchDetails AS search\_details**

**FROM [bronze\_lakehouse\_wtc].[dbo].[warehouse\_stockitems]**

If pasting the above code you’ll need to replace the reference of the **bronze\_lakehouse** name.



Once your query is complete, it should look similar to the image provided below with aliasing. Execute the query by clicking **Run** and review the output.

With the query complete, highlight the query from the query window and click **Save as view**. Name the view **stock\_items** and click **OK**.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Create another **View** for **customers** using the following **CustomerID**, **CustomerName**, **CityName**, **StateProvinceCode**, and **PhoneNumber**.

**SELECT**

**CustomerID AS customer\_id**

**,CustomerName AS customer\_name**

**,CityName AS delivery\_city\_name**

**,StateProvinceCode AS delivery\_state\_province**

**,PhoneNumber AS phone\_number**

**FROM**

**[bronze\_lakehouse\_wtc].[dbo].[sales\_customers] sc**

**JOIN [bronze\_lakehouse\_wtc].[dbo].[locations] l**

**ON sc.DeliveryCityID = l.CityID**

Again, remember to replace the **bronze\_lakehouse** reference if pasting the above code.

A screenshot of a computer

Description automatically generated

You have successfully completed **Part 4** of this lab.