Lab 2 - Implement the Medallion Architecture using Microsoft Fabric

Lab Guide

October 2024

Contents

Document Purpose

This document describes the IP named "**Lab 2 - Implement the Medallion Architecture using Microsoft Fabric(Bronze, Silver and Gold layers)**".

Lab Description

Implement the Medallion Architecture using Microsoft Fabric (Bronze, Silver and Gold layers)

This exercise revolves around implementing the Medallion Architecture utilizing Microsoft Fabric, with a particular emphasis on its Bronze, Silver, and Gold layers.
The Medallion Architecture is a data processing paradigm where raw data is ingested into the Bronze layer, transformed and curated in the Silver layer, and then aggregated and analyzed in the Gold layer.
In this context, Microsoft Fabric serves as the platform for executing data transformations, leveraging its powerful analytics capabilities to process data from the Bronze layer, convert it into a delta format, and subsequently store it in the Silver layer for further refinement and analysis.

Lab Tasks

The lab tasks for "Lab 2 - Implement the Medallion Architecture using Microsoft Fabric (Bronze, Silver, and Gold layers)" involve a series of steps designed to enhance data processing and analytics capabilities.

- The first task focuses on reading data from the Bronze layer, converting it into delta format, and writing it to the Silver layer to improve data quality.

- The second task involves verifying that the delta files have been successfully written to the Silver layer.

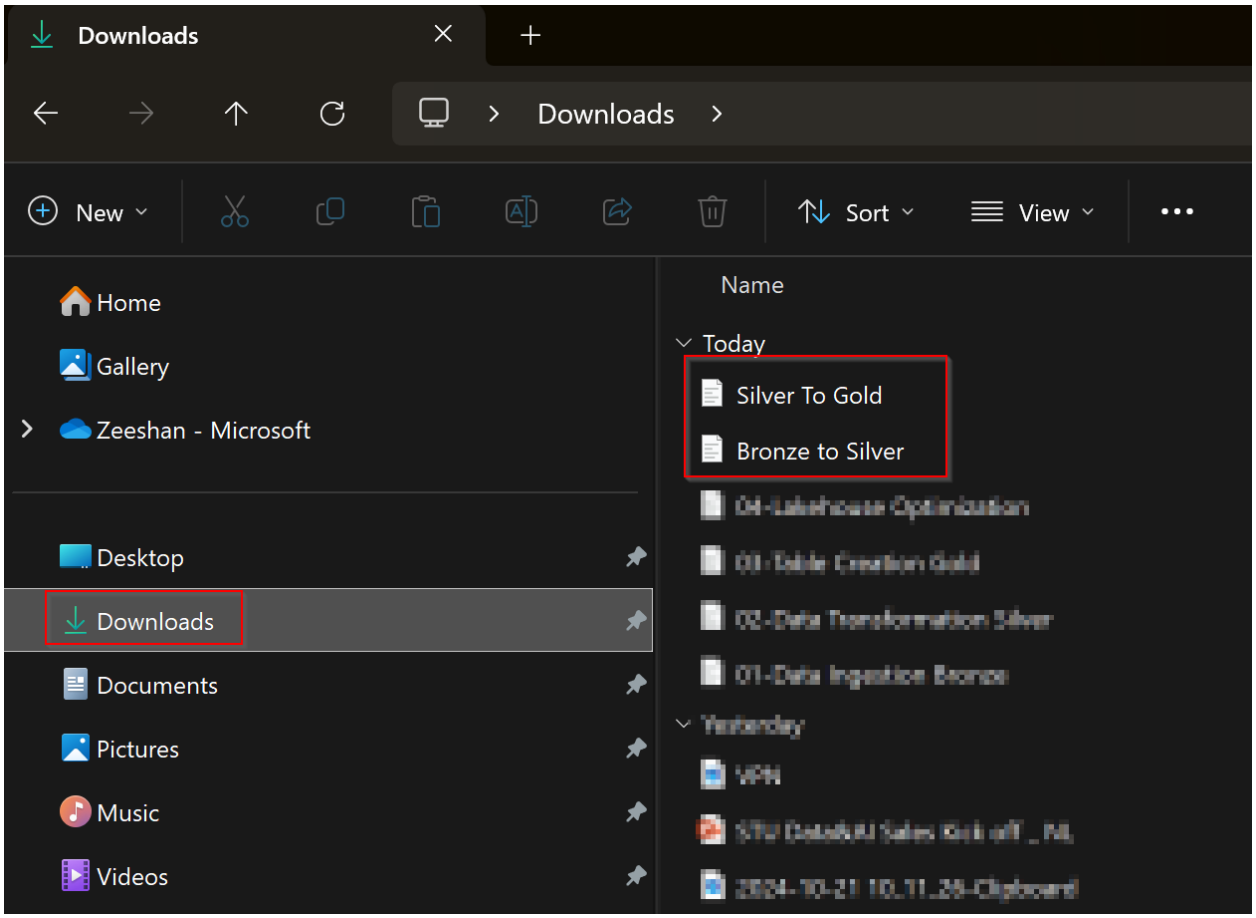
- In the third task, participants read data from the Silver layer, create a data model, and write the refined data into the Gold layer as tables. Finally, the fourth task requires verifying that the delta files have been successfully written to the Gold layer. These tasks collectively help implement the Medallion Architecture, leveraging Microsoft

Fabric's powerful analytics capabilities to process and refine data across different layers.

Task 01: Read from Bronze layer, convert the data into delta format and write to the Silver layer

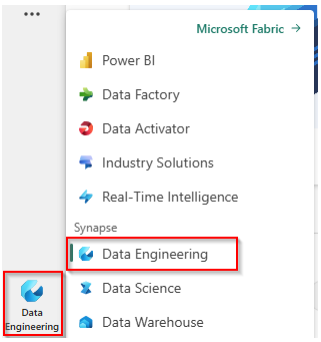
With this move we aim to clean and refine data within the Bronze layer, improving data quality and preparing it for further processing, aligning with Contoso's focus on reliable data foundations.

1. Download the Bronze to Silver and Silver to Gold notebooks under Assets/Day2/Notebooks/ on your local machine.
2. Open **File Explorer** and go to the **Downloads** folder.

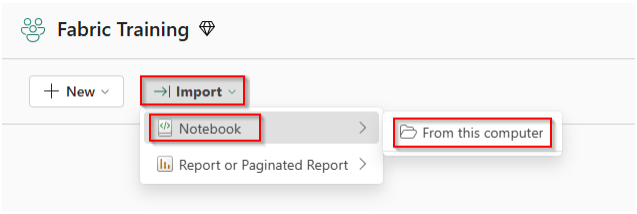


1. The notebooks have been downloaded in the Downloads folder.

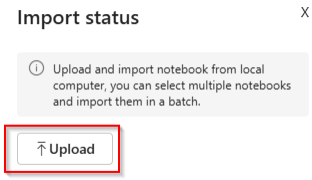
2. Return to the **Fabric Workspace** browser tab.
3. In the bottom navigation pane, select **Data Engineering Experience.**



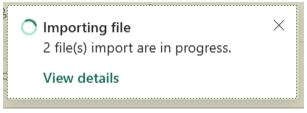
4. In the left navigation pane, select **Workspaces**.
5. In the Workspace pane, select **Workspaces** and then select **Fabric Training**.
6. In the upper-left pane, select **Import**.



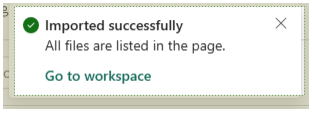
7. In the Import window, select **Notebook** and then select **From this computer**.
8. In the right pane, click on **Upload.**



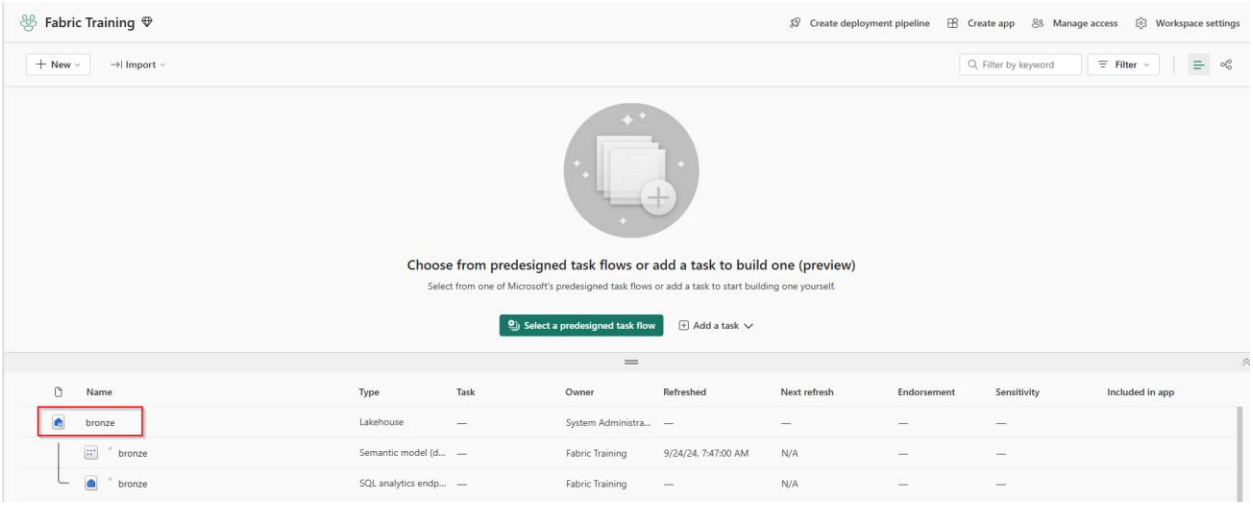
9. In **File Explorer**, go to your **Downloads folder**, then open the **BronzeToSilver.ipynb and SilverToGold.ipynb** files.
10. Wait for the Notebooks to imported.



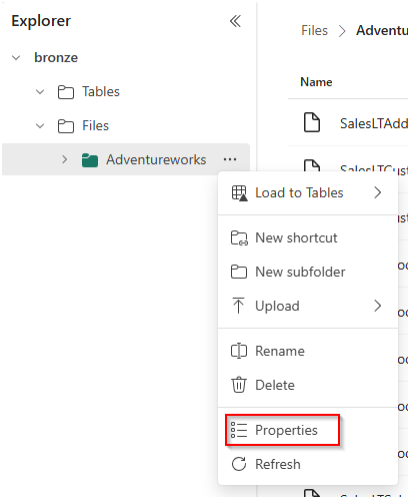
11. Once the notebooks have been imported successfully. You will see this notification.



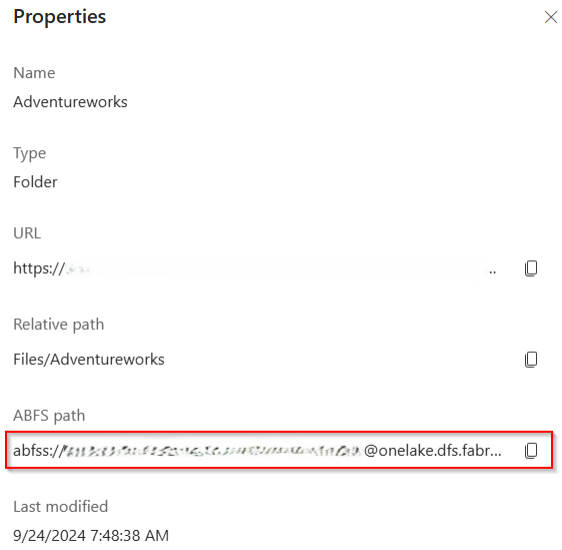
12. Now open the bronze lakehouse.



13. In the lakehouse explorer, on the left navigation panel right click on Adventureworks folder under Files and select Properties.



14. In the properties section copy the ABFS path of Adventureworks folder in the bronze lakehouse.



15. Now the notebooks are imported, select the **Bronze to Silver** notebook. We're now getting ready to transform and store processed data in the Silver layer in delta format, enhancing Contoso's data storage efficiency and accessibility for analytical processes.
16. In the second code cell,

```
from notebookutils import mssparkutils
adventureWorksPath = "<ENTER HERE>"

file_list = mssparkutils.fs.ls(adventureWorksPath)

# Read each file and create a DataFrame
for file_path in file_list:
    print(file_path)
    df = spark.read.format("csv").options(inferSchema="true", header="true").load(path=f"{file_path.path}*")
    # You can process the DataFrame or register it as a table here
    # For example, to create a temporary table:
    df.createOrReplaceTempView(file_path.name.removesuffix('.csv'))
```

[2]

update the ENTERHERE plcaeholders with the following information:

Name	**Value**	**Comment**
adventureWorksPath	**<ENTERHERE>**	Paste the path you copied in step 14

17. In the third code cell,

```
%%sql

<ENTER HERE>
```

replace the **<ENTERHERE>** placeholder with the following command.

```
SHOW VIEWS
```

This command displays available views.

18. In the fourth code cell,

```
%%sql

<ENTER HERE>
```

replace the **<ENTERHERE>** placeholder with the following command.

```
SELECT * FROM salesltaddress LIMIT 100
```

This command displays the first 100 rows from salesltaddress.

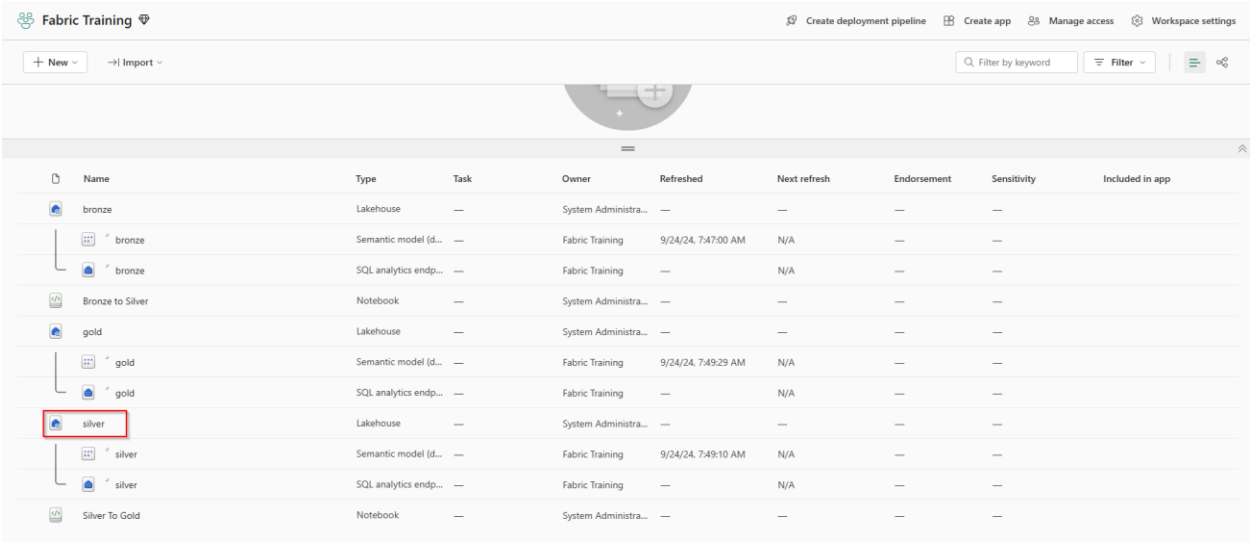
19. In the fifth code cell,

```
views = <ENTER HERE>
display(views)
```

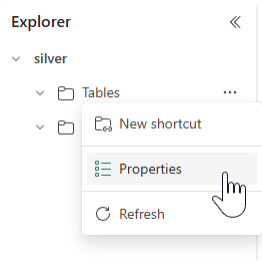
replace the **<ENTERHERE>** placeholder with the following command.

```
spark.sql("SHOW VIEWS")
```

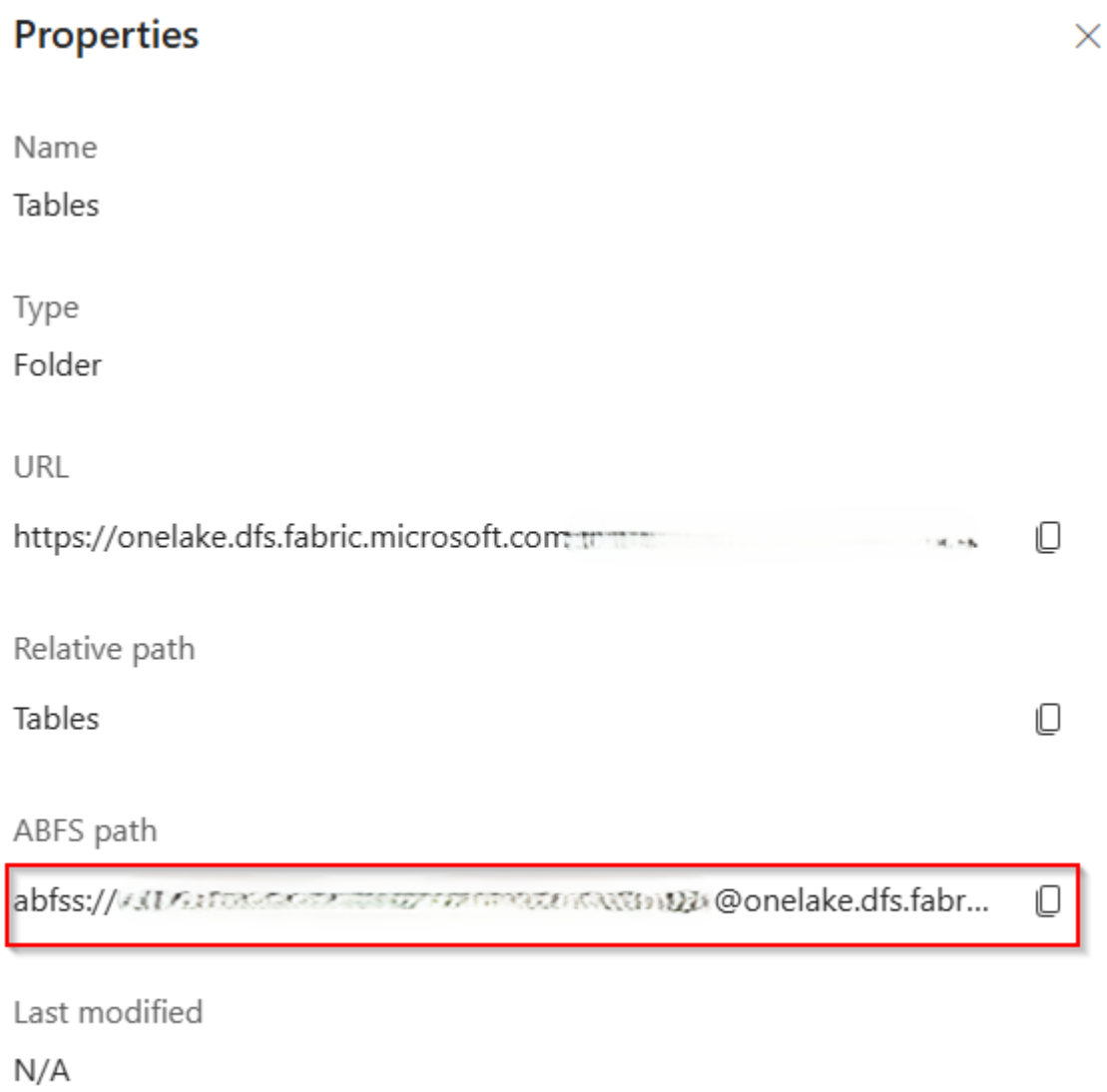
20. Now go to the Fabric Training workspace and open the silver lakehouse.



21. In the lakehouse explorer, on the left navigation panel right click on Tables folder and select Properties.



22. In the properties section copy the ABFS path of Tables folder in the silver lakehouse.



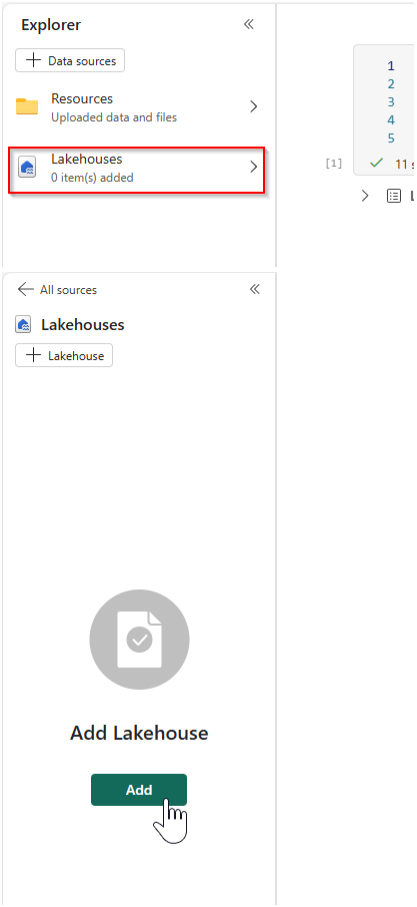
23. In the thirteenth code cell,

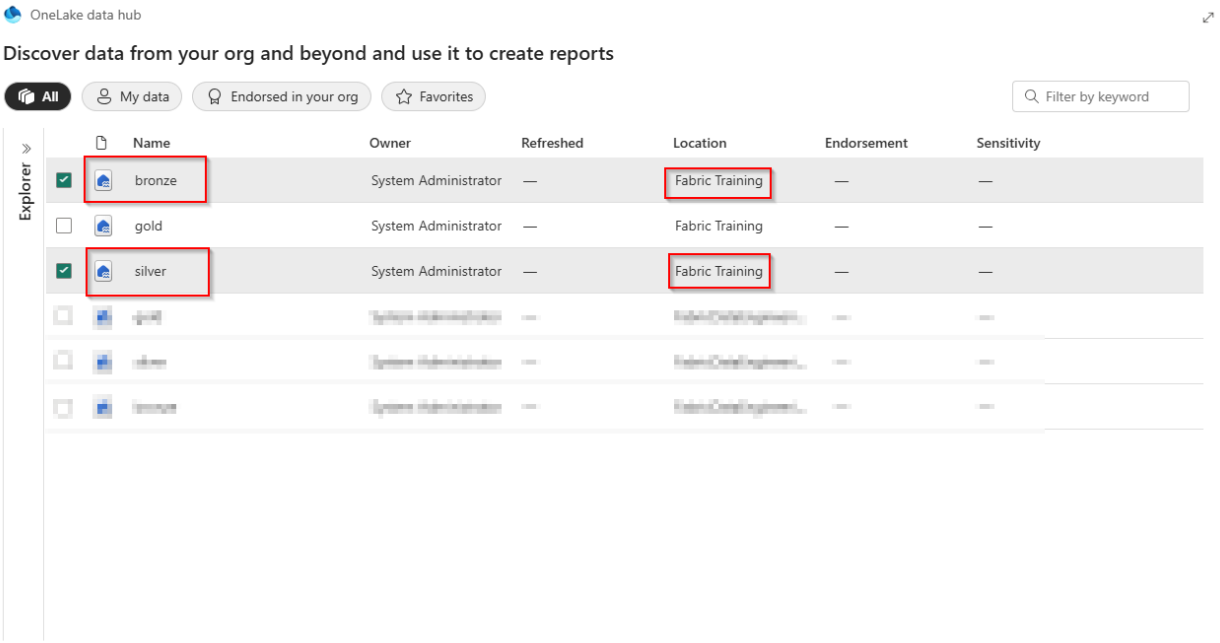
```
[13]
1  basePathSilverLakeHouse = "<ENTER HERE>"
2  tableName="salesOrderHeader"
3  df_salesltsalesorderheader.write.mode("overwrite").format("delta").save(basePathSilverLakeHouse + '///' + tableName)
```

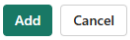
update the ENTERHERE placeholders with the following information:

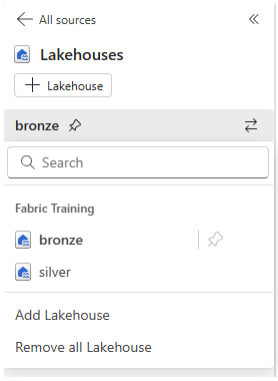
Name	Value	Comment
basePathSilverLakeHouse	**<ENTER HERE>**	Paste the path you copied in step 22

24. On the left navigation panel of the notebook. Click explorer, check that whether bronze and silver lakehouses are added or not. If bronze and silver lakehouses are not added, click on Lakehouses and add them.

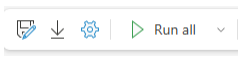








25. Select **Run all**.



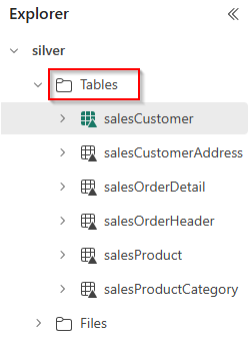
26. Review the output for each cell.

Task 02: Verify that delta files were written to the silver medallion lakehouse

1. Switch to the browser tab for the Fabric portal. In the left navigation panel, click on Fabric Training workspace.

2. From the center pane, select the silver Lakehouse.

3. Select the Tables. Verify that the delta tables are present in the Tables Folder.



Task 03: Read from the Silver layer, create a data model (Facts, Dimensions) on the onelake, and write data into Gold layer as tables.

The last step is to develop and implement a robust data model from the Silver layer, moving critical data into the Gold layer, thus bolstering Contoso's analytics and decision-making capabilities with structured and refined data sets.

This involves creating fact and dimension tables which are essential components of a data warehouse. Fact tables generally contain transactional data such as sales figures, while dimension tables hold descriptive information like product details or customer attributes.

By organizing data in this manner, it becomes easier to perform complex queries and generate insightful reports, significantly enhancing the company's ability to leverage its data for strategic purposes.

1. Switch back to the **Fabric** browser tab.
2. On the **Fabric** page, in the left navigation pane, select **Fabric Training Workspace**.
3. In the **Fabric Training** workspace, select the **Silver to Gold** notebook.

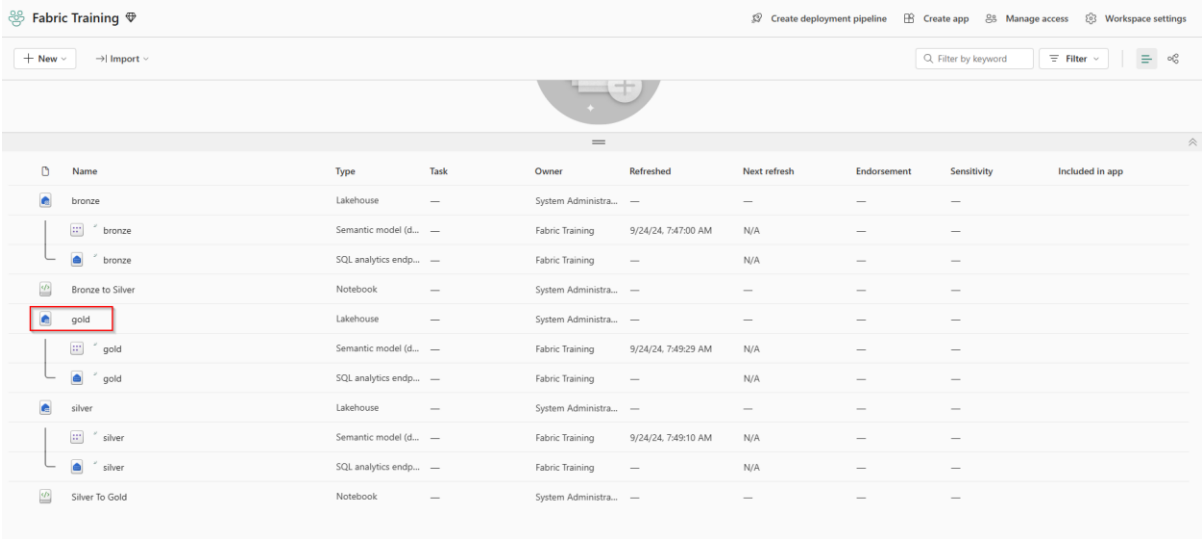
4. In the fourth code cell,

```
[4]
from pyspark.sql.functions import monotonically_increasing_id

#Create dimProduct
df_dimProduct = <ENTER HERE>



#Add surrogate key as the first column
df_dimProduct_with_surrogate_key = df_dimProduct.withColumn("ProductIDKey", monotonically_increasing_id())\
    .select(
        "ProductIDKey",  # Select the surrogate key column first
        *[column for column in df_dimProduct.columns if column != "ProductIDKey" and column!="spc.Name"]  # Select the remaining columns in their original order
    )
```

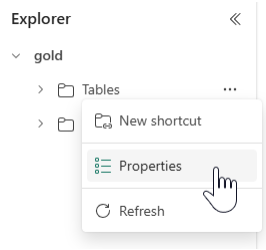
replace the **<ENTERHERE>** placeholder with the following code:

```
spark.sql("select sp.*, spc.ParentProductCategoryID, spc.Name as ProductCategoryName from silver.salesProduct sp join silver.salesProductCategory spc on sp.ProductCategoryID = spc.ProductCategoryID")
```

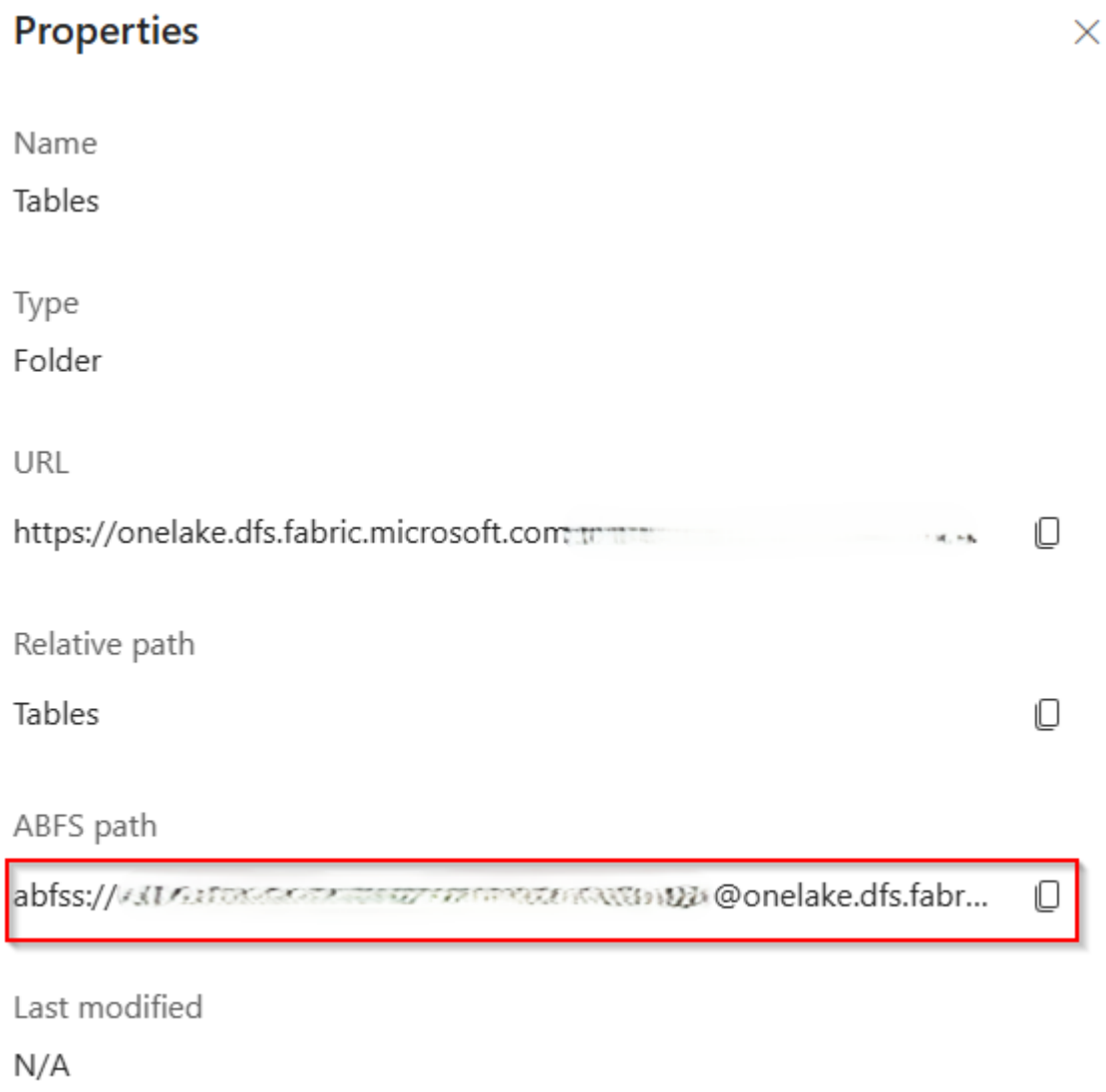
5. Now go to the Fabric Training workspace and open the gold lakehouse.



6. In the lakehouse explorer, on the left navigation panel right click on Tables folder and select Properties.



7. In the properties section copy the ABFS path of Tables folder in the gold lakehouse.



8. In the seventh code cell, update the ENTERHERE plcaeholders with the following information:

Name	**Value**	**Comment**
basePathGoldLakeHouse	**<ENTER HERE>**	Paste the path you copied in step 7

9. In the ninth code cell,

```
#Create factSales
df_factSales = <ENTER HERE>
display(df_factSales)
```

[9] ✓

populate the **ENTERHERE** placeholder to create the df_factSales dataframe:

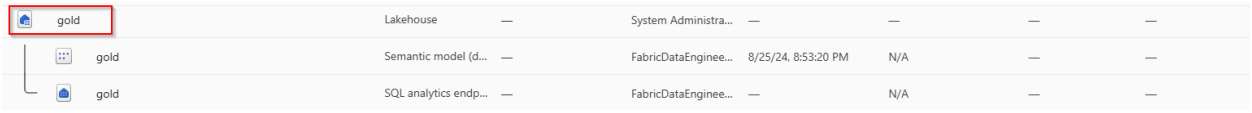
```
spark.sql("select dp.ProductIDKey, ds.CustomerIDKey, soh.*, sod.OrderQty, sod.ProductID, sod.UnitPrice, sod.UnitPriceDiscount, sod.LineTotal from silver.salesOrderHeader soh join silver.salesOrderDetail sod on soh.SalesOrderID = sod.SalesOrderID LEFT JOIN gold.dimProduct dp ON sod.ProductID = dp.ProductID LEFT JOIN gold.dimCustomer ds ON soh.CustomerID = ds.CustomerID")
```

10. Select **Run all**.
11. Review the output for each cell.

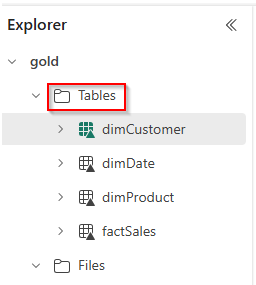
Task 04: Verify that delta files were written to the gold medallion container

1. Switch to the browser tab for the Fabric portal. In the left navigation panel, click on **Fabric Training** workspace.

2. From the center pane, select the gold Lakehouse.



3. Select the Tables. Verify that the delta tables are present in the Tables Folder.



Summary:

The lab tasks for "Lab 2 - Implement the Medallion Architecture using Microsoft Fabric (Bronze, Silver, and Gold layers)" are designed to enhance data processing and analytics capabilities through a series of steps. Participants start by reading data from the Bronze layer, converting it into delta format, and writing it to the Silver layer to improve data quality.

They then verify the successful writing of delta files to the Silver layer. The next step involves reading data from the Silver layer, creating a data model, and writing the refined data into the Gold layer as tables. Finally, participants verify that the delta files have been successfully written to the Gold layer. These tasks collectively help implement the Medallion Architecture, leveraging Microsoft Fabric's powerful analytics capabilities to process and refine data across different layers.