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

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 Engineering – Data Transformation and Engineering in Fabric

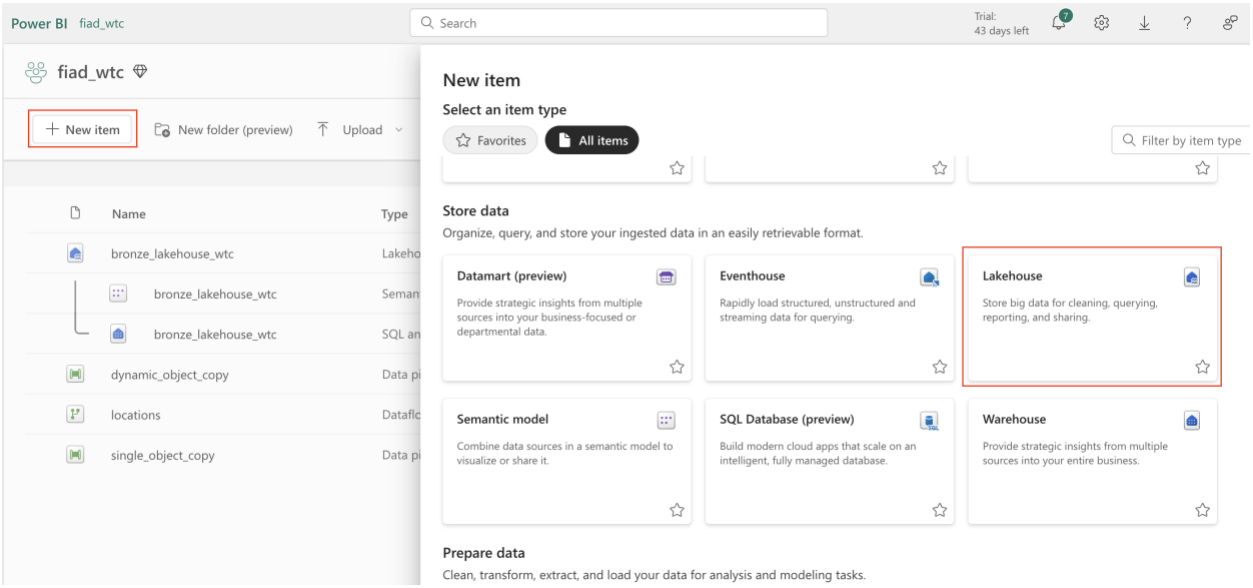
Introduction

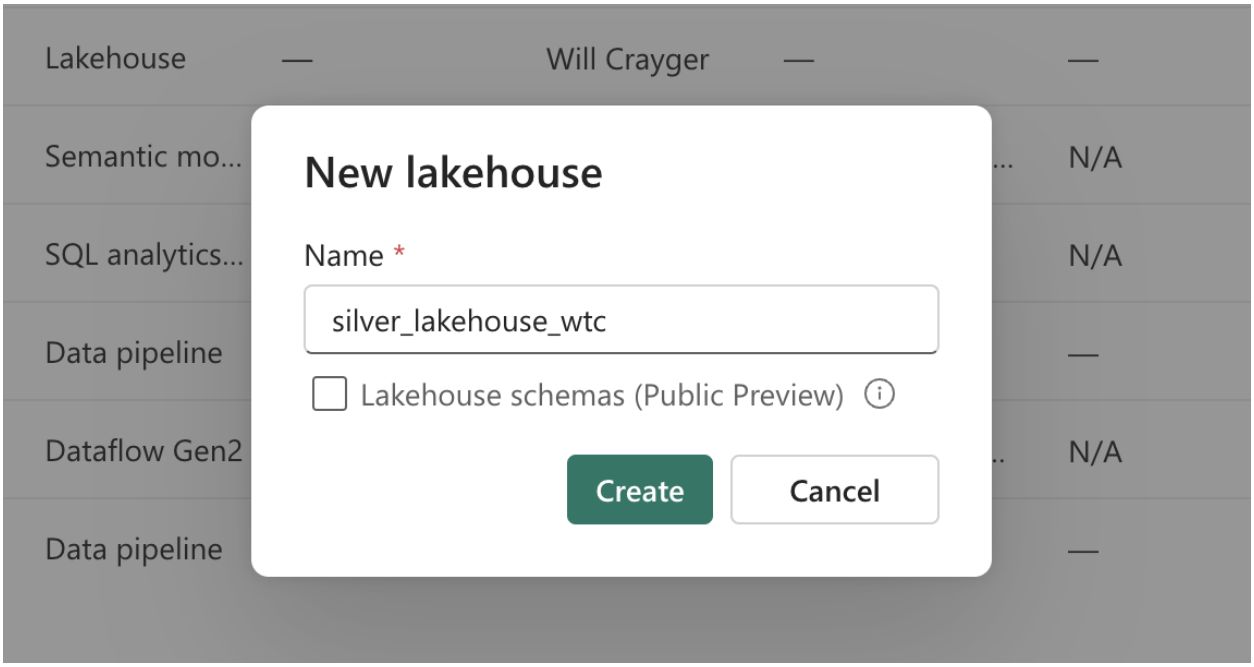
In this lab, you're going to create a **Silver** Lakehouse and begin loading data from the **Bronze** Lakehouse that was created in **Lab 1**. As you proceed through the lab, please note that you're working in a case-sensitive environment.

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.

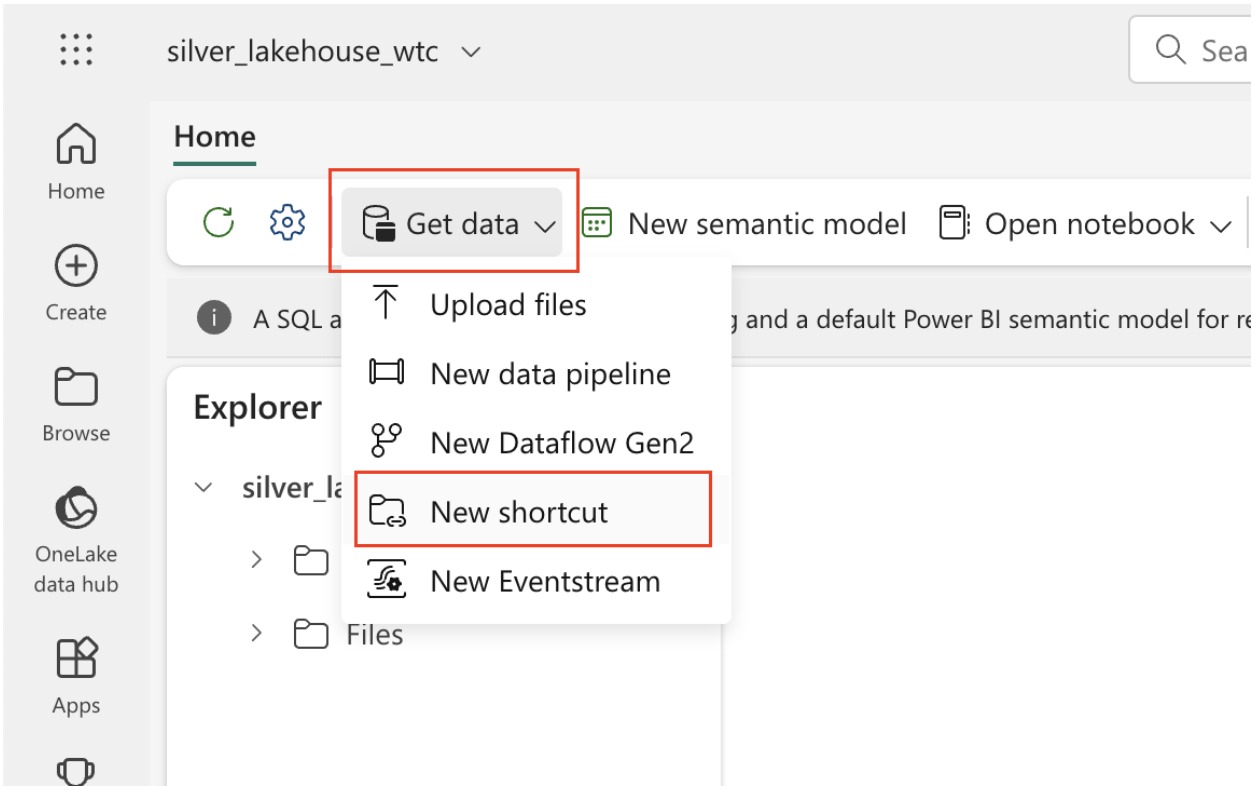
To begin, select **New item** from your Workspace home page. Scroll down and select **Lakehouse** from the item list. From the tiles at the top of the page, select **Lakehouse**. Name your lakehouse **silver_lakehouse_<your_initials>** and click **Create**.

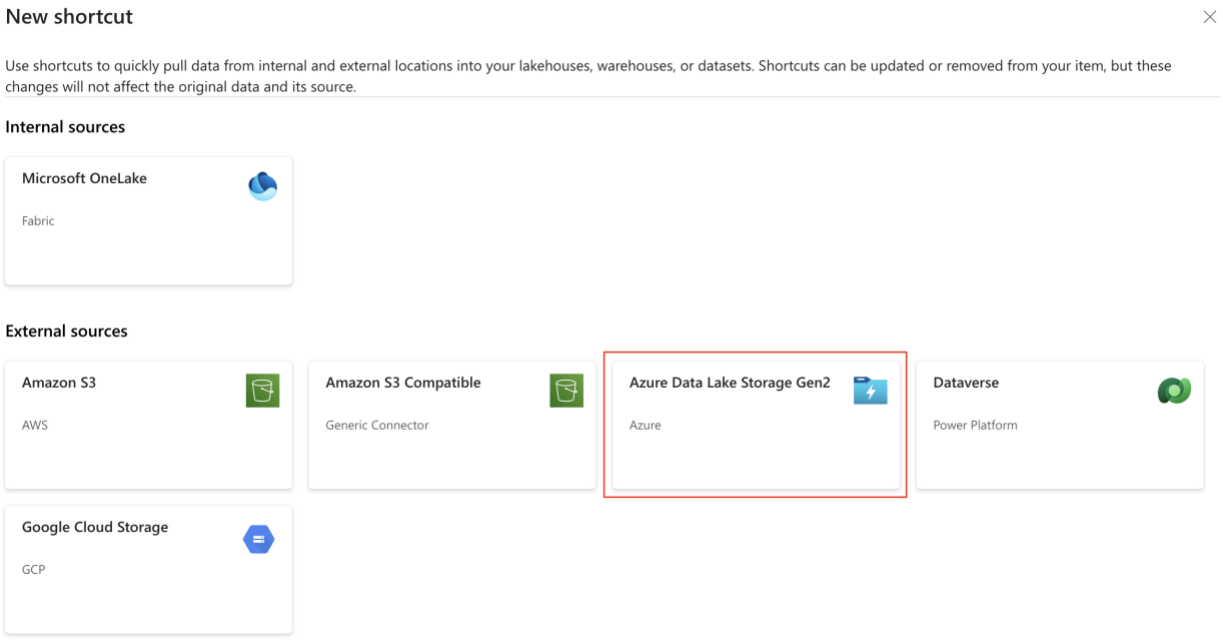




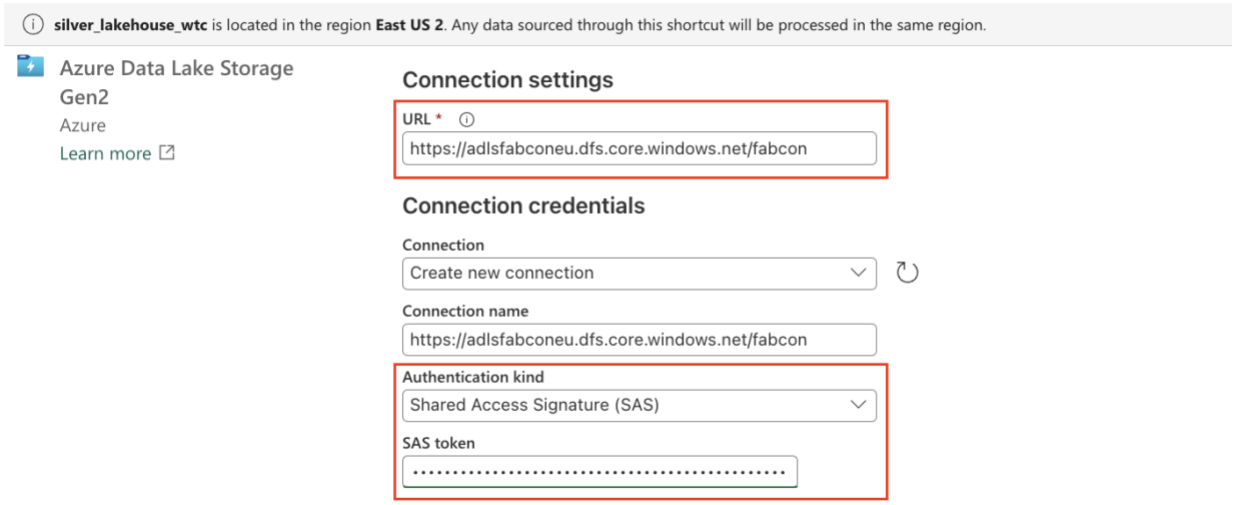
Creating a Shortcut to Azure Data Lake Gen 2: To access semi-curated invoice data, you'll need to create a shortcut to an Azure Data Lake Gen 2 resource provided as a part of this coursework.

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**. Navigate down the folder hierarchy and check the tickbox beside the **invoices** folder and click **Next**. Click **Create**.

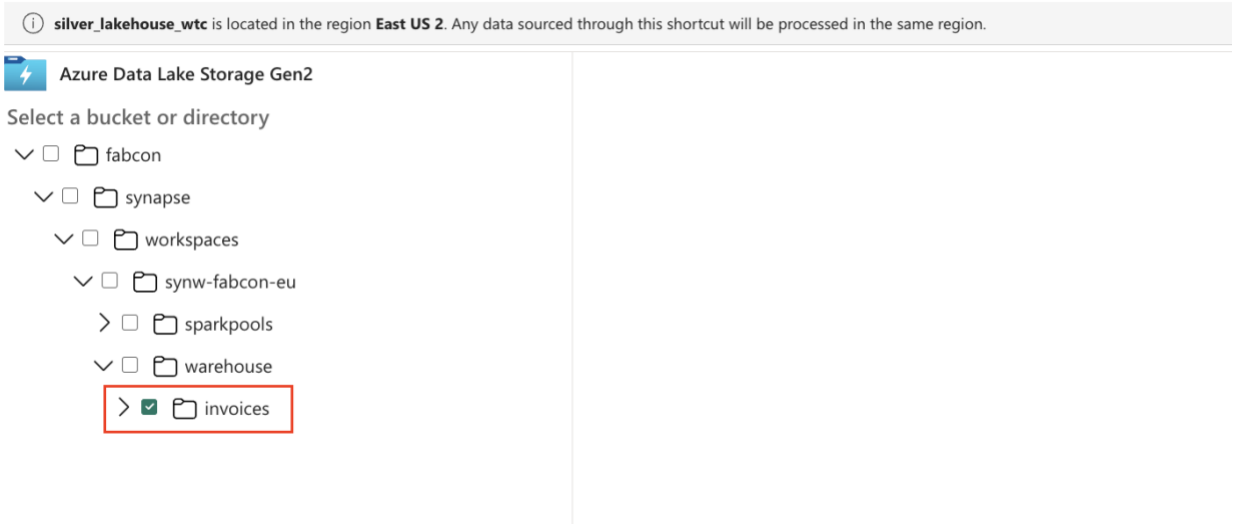

silver_lakehouse_wtc
Sea
Home
Home
Create
Browse
OneLake data hub
Apps
Get data
New semantic model
Open notebook
A SQL a
and a default Power BI semantic model for re
Upload files
New data pipeline
New Dataflow Gen2
New shortcut
New Eventstream
Explorer
silver_la
Files

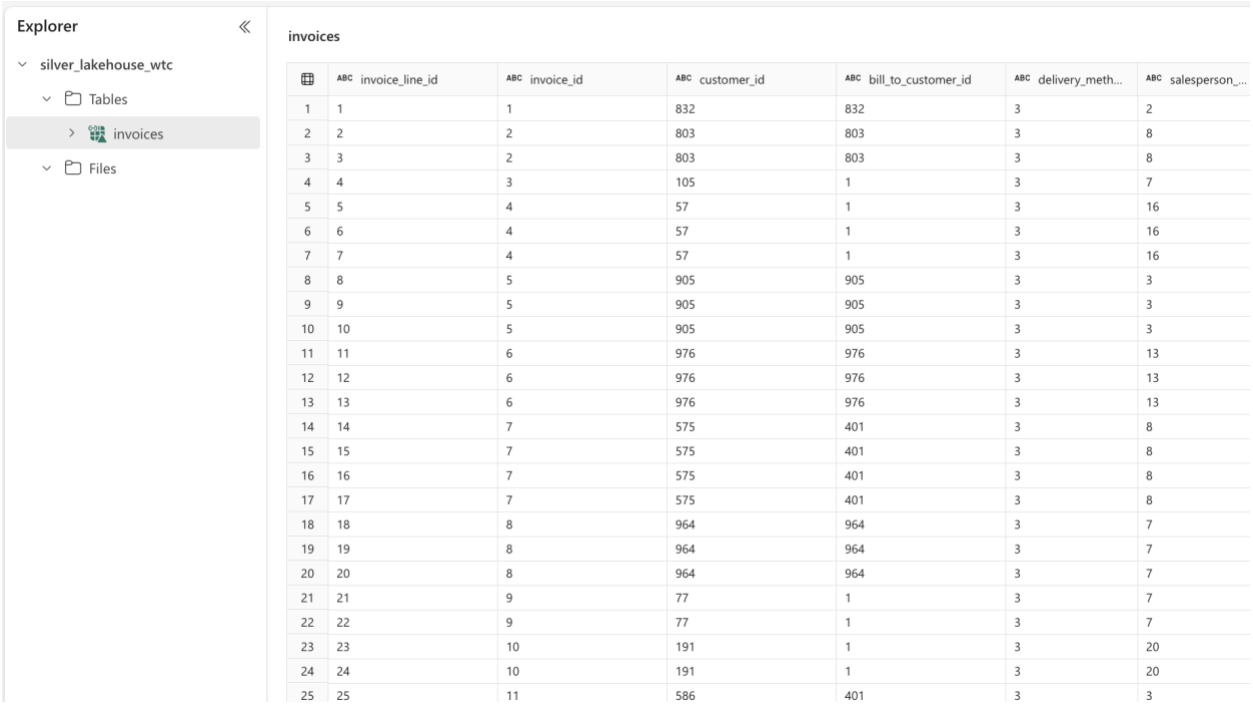

New shortcut
Use shortcuts to quickly pull data from internal and external locations into your lakehouses, warehouses, or datasets. Shortcuts can be updated or removed from your item, but these changes will not affect the original data and its source.
Internal sources
Microsoft OneLake
Fabric
External sources
Amazon S3
AWS
Amazon S3 Compatible
Generic Connector
Azure Data Lake Storage Gen2
Azure
Dataverse
Power Platform
Google Cloud Storage
GCP

New shortcut



New shortcut





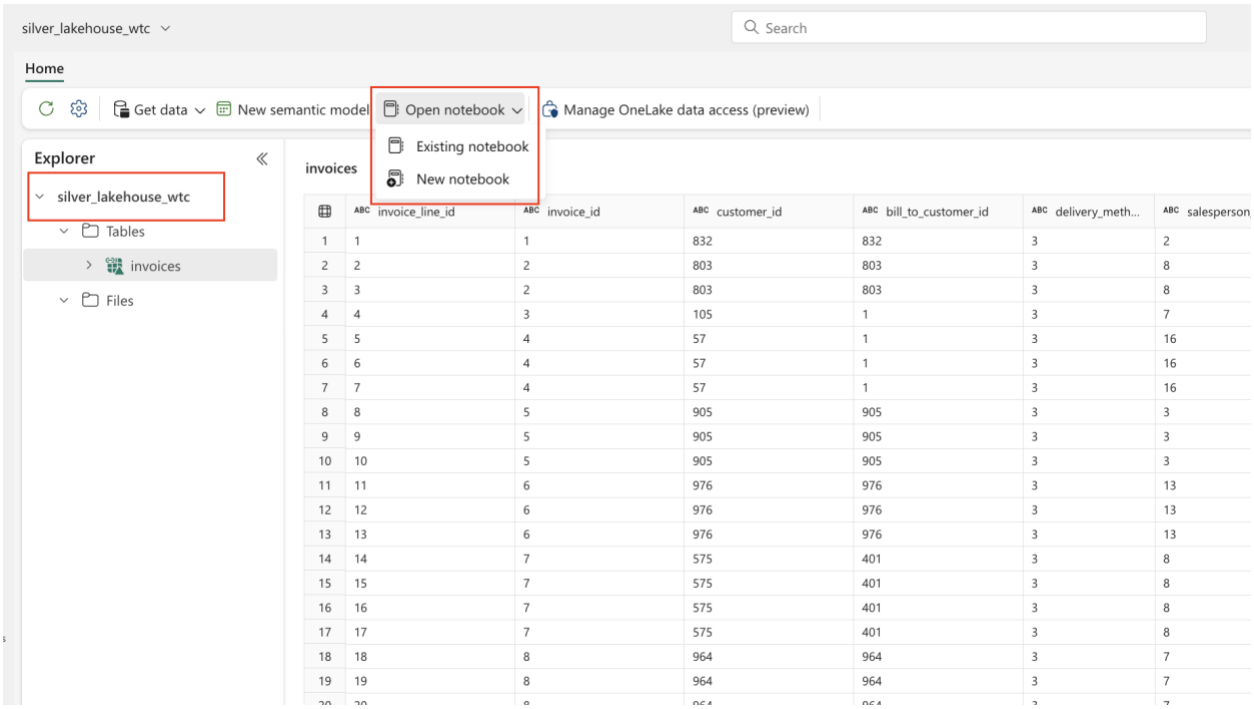
	invoice_line_id	invoice_id	customer_id	bill_to_customer_id	delivery_meth...	salesperson_...
1	1	1	832	832	3	2
2	2	2	803	803	3	8
3	3	2	803	803	3	8
4	4	3	105	1	3	7
5	5	4	57	1	3	16
6	6	4	57	1	3	16
7	7	4	57	1	3	16
8	8	5	905	905	3	3
9	9	5	905	905	3	3
10	10	5	905	905	3	3
11	11	6	976	976	3	13
12	12	6	976	976	3	13
13	13	6	976	976	3	13
14	14	7	575	401	3	8
15	15	7	575	401	3	8
16	16	7	575	401	3	8
17	17	7	575	401	3	8
18	18	8	964	964	3	7
19	19	8	964	964	3	7
20	20	8	964	964	3	7
21	21	9	77	1	3	7
22	22	9	77	1	3	7
23	23	10	191	1	3	20
24	24	10	191	1	3	20
25	25	11	586	401	3	3

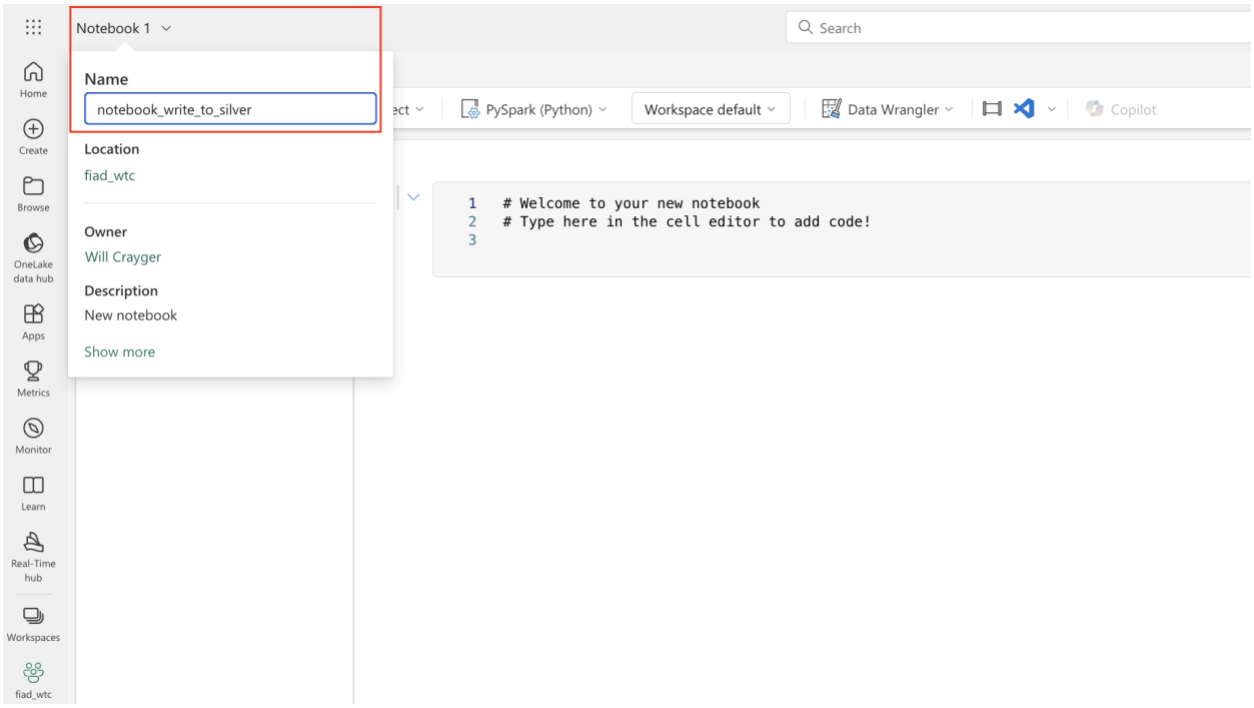
Part 2: Creating a New Notebook

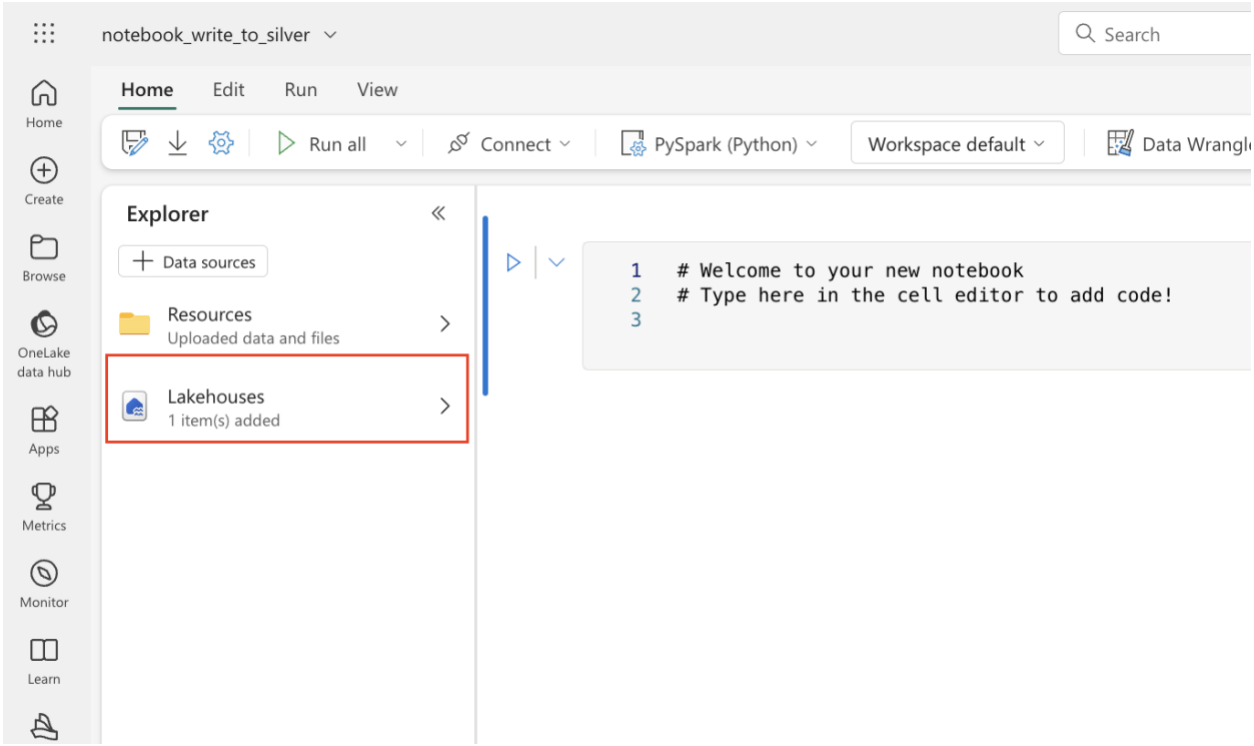
Mounting the Notebook: From your **Silver Lakehouse**, click **Open Notebook** and select **New Notebook**. At the top-left of your browser, click the name to rename the notebook to **notebook_write_to_silver**.

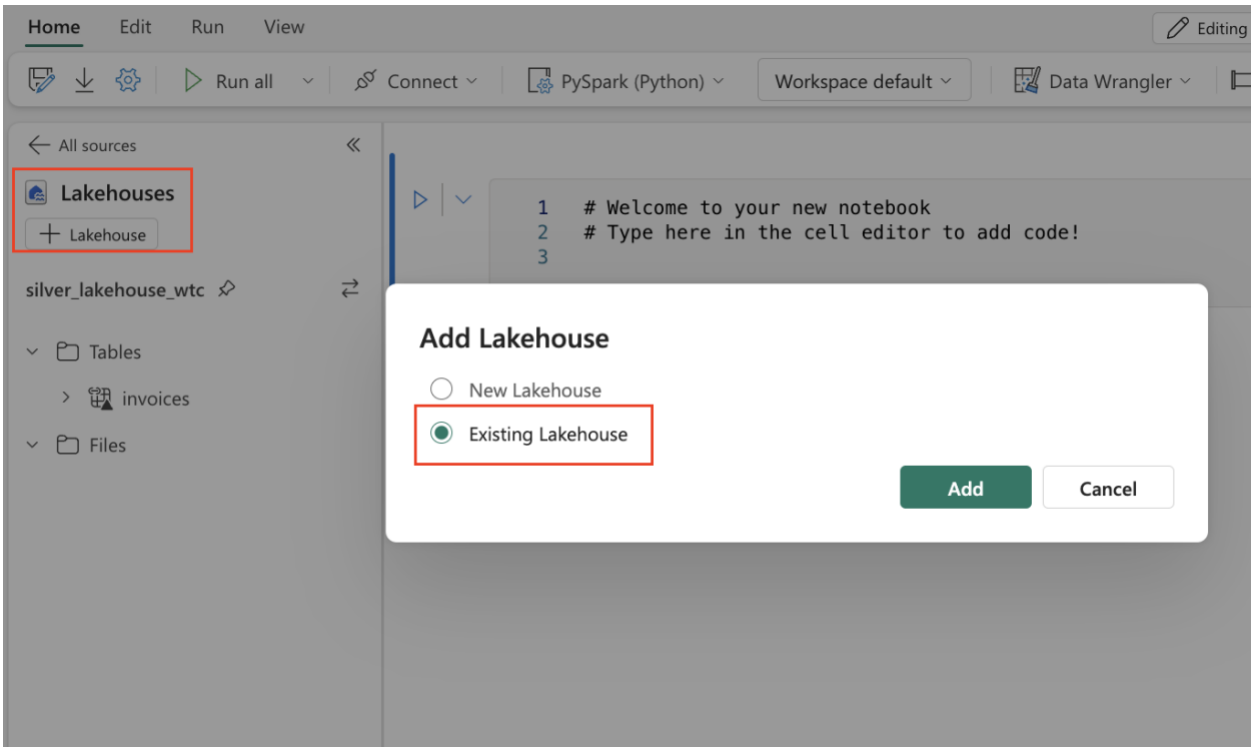
Next, click **Lakehouses** from the **Explorer** blade. Click the **+ Lakehouse** button, check the **Existing Lakehouse** radial button, and click **Add**. Select the **Bronze Lakehouse** and click **Add** again.

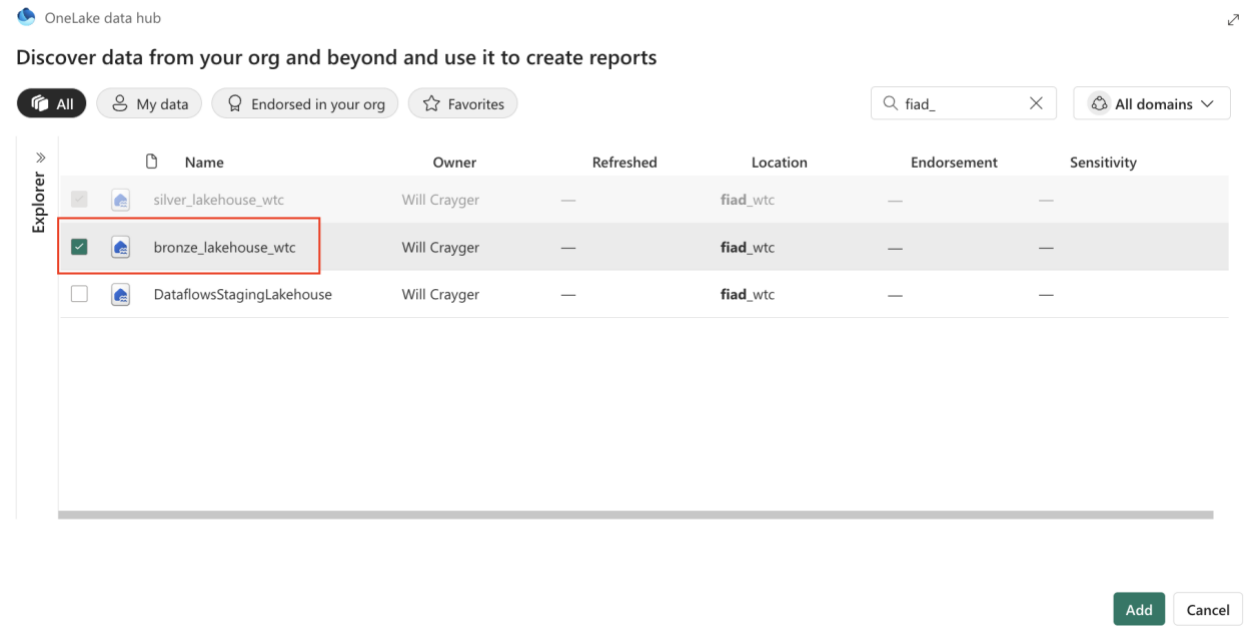
The tables from the **Bronze Lakehouse** will now be visible in the notebook, ready for transformation.

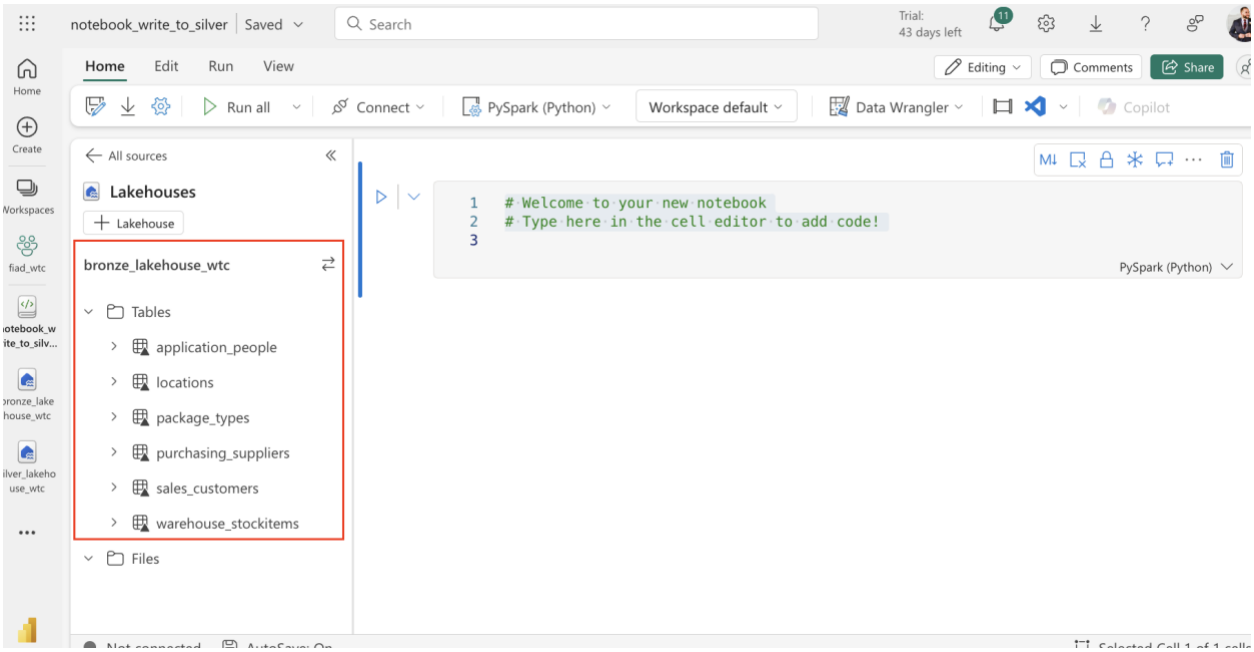

silver_lakehouse_wtc
Search
Home
Get data
New semantic model
Open notebook
Existing notebook
New notebook
Manage OneLake data access (preview)
Explorer
silver_lakehouse_wtc
Tables
invoices
Files
invoices
invoice_line_id
invoice_id
customer_id
bill_to_customer_id
delivery_meth...
salesperson
1 1 1 832 832 3 2
2 2 2 803 803 3 8
3 3 2 803 803 3 8
4 4 3 105 1 3 7
5 5 4 57 1 3 16
6 6 4 57 1 3 16
7 7 4 57 1 3 16
8 8 5 905 905 3 3
9 9 5 905 905 3 3
10 10 5 905 905 3 3
11 11 6 976 976 3 13
12 12 6 976 976 3 13
13 13 6 976 976 3 13
14 14 7 575 401 3 8
15 15 7 575 401 3 8
16 16 7 575 401 3 8
17 17 7 575 401 3 8
18 18 8 964 964 3 7
19 19 8 964 964 3 7


Notebook 1
Search
Name
notebook_write_to_silver
Location
fiad_wtc
Owner
Will Crayger
Description
New notebook
Show more
PySpark (Python)
Workspace default
Data Wrangler
Copilot
Home
Create
Browse
OneLake data hub
Apps
Metrics
Monitor
Learn
Real-Time hub
Workspaces
fiad_wtc
1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3


notebook_write_to_silver
Search
Home
Edit
Run
View
Run all
Connect
PySpark (Python)
Workspace default
Data Wrangl
Explorer
Data sources
Resources
Uploaded data and files
Lakehouses
1 item(s) added
1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3
Home
Create
Browse
OneLake data hub
Apps
Metrics
Monitor
Learn


Home
Edit
Run
View
Editing
Run all
Connect
PySpark (Python)
Workspace default
Data Wrangler
All sources
Lakehouses
Lakehouse
silver_lakehouse_wtc
Tables
invoices
Files
1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3
Add Lakehouse
New Lakehouse
Existing Lakehouse
Add
Cancel



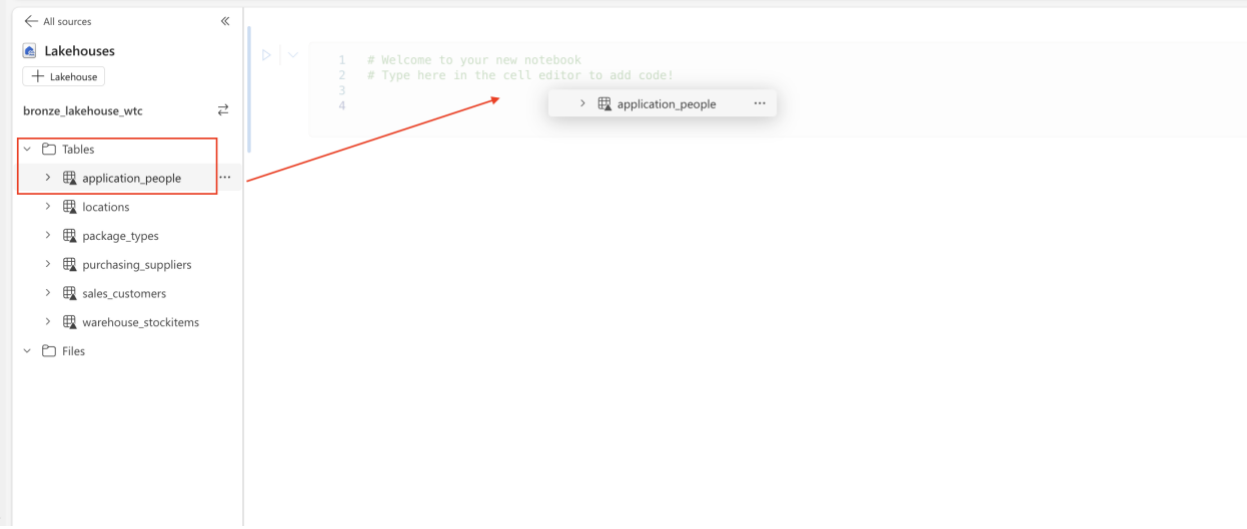


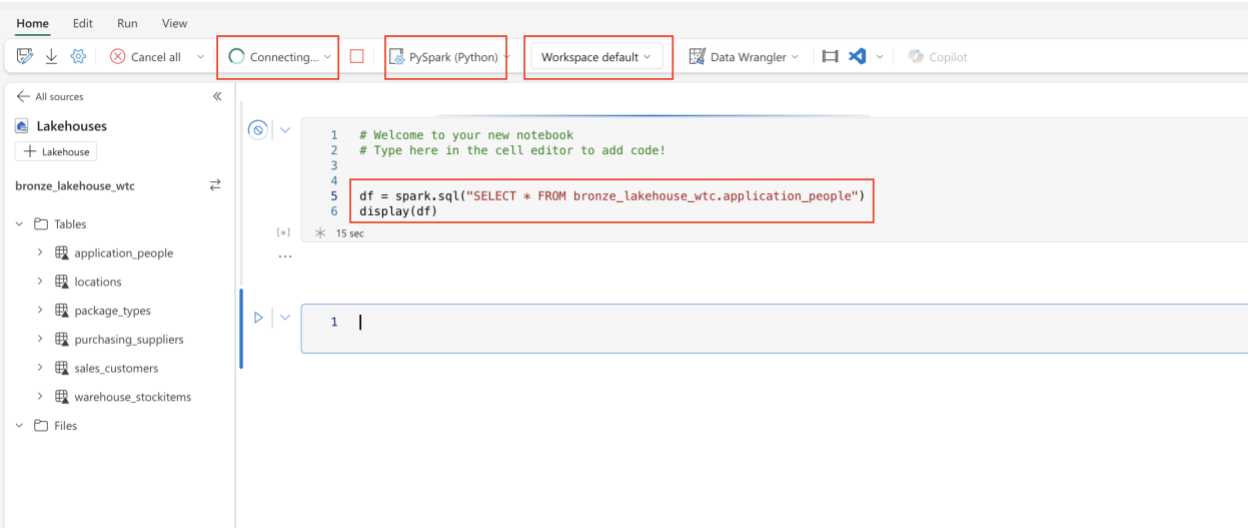
Writing Data to Silver: Click and drag **application_people** from the **Explorer** to the first cell in the notebook. Remove the **LIMIT 1000** clause from the cell and press **Shift + Enter** to execute the cell. This action will start a Spark session using the default **single_node** Spark Pool that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

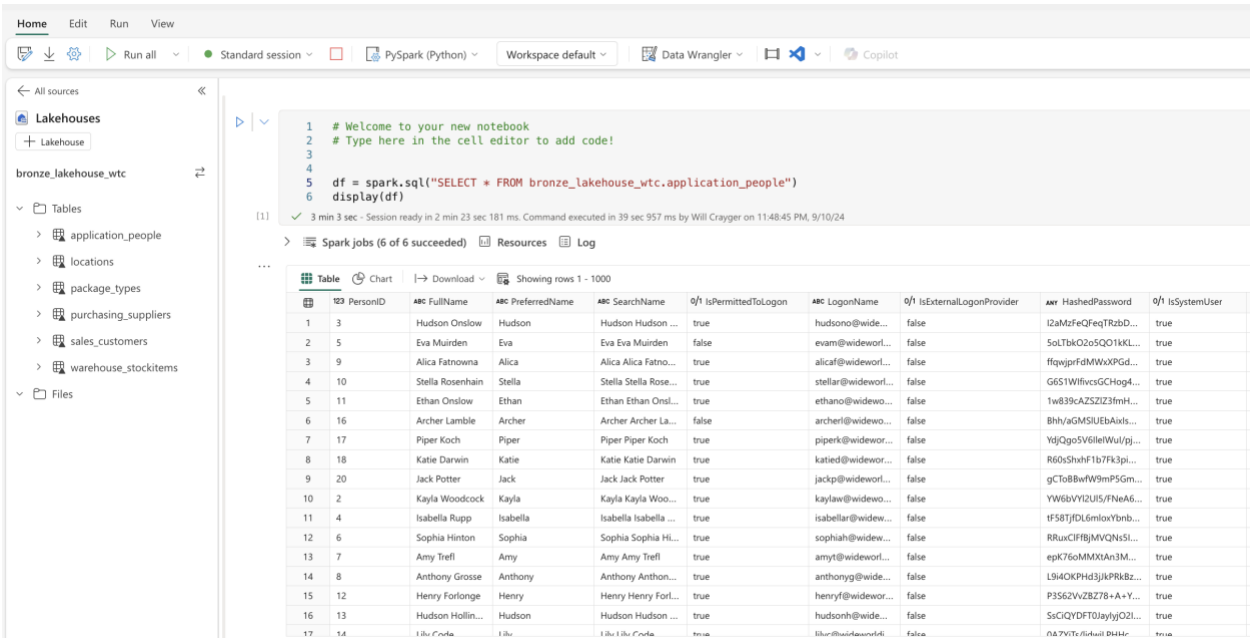
To filter the dataframe to individuals identified as a salesperson, copy and paste the following command into the blank cell at the bottom of your notebook. Press **Shift + Enter** to execute the cell

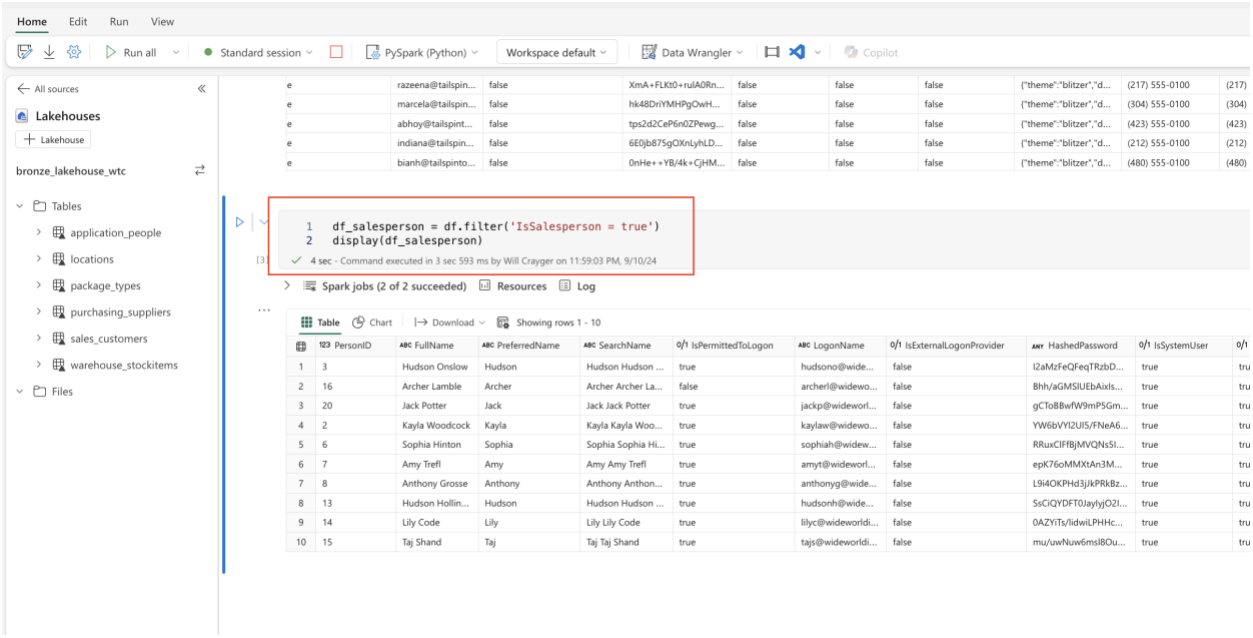
df_salesperson = df.filter('IsSalesperson = true')

display(df_salesperson)









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.**

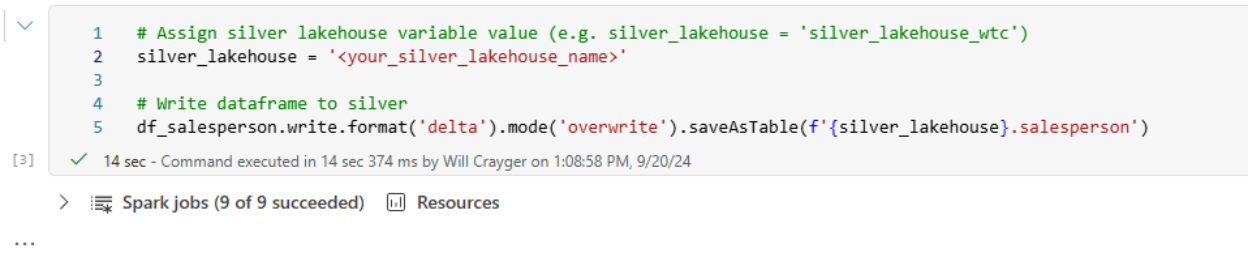
After updating the variable, press **Shift + Enter** to execute the cell. To confirm the results were successfully written, switch to the **Silver Lakehouse** from the **Notebook Explorer**.

 # 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')

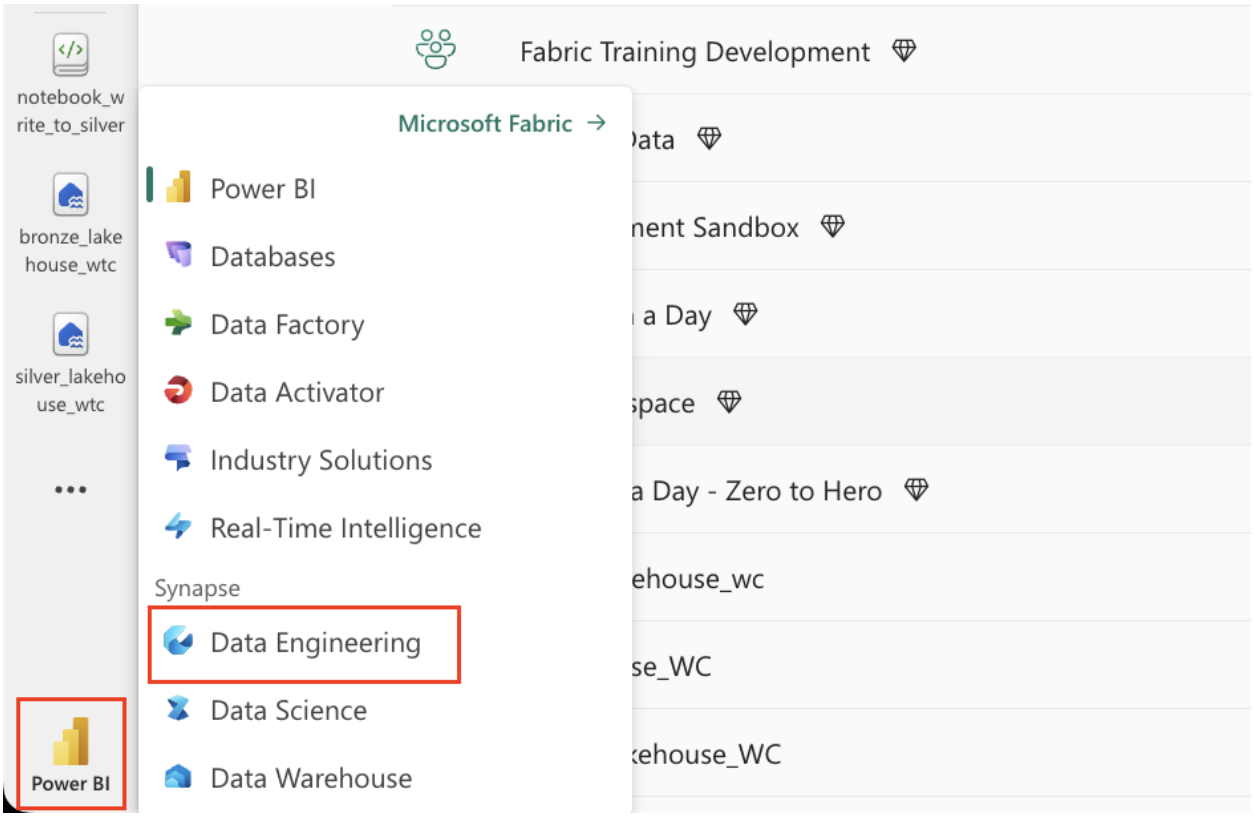


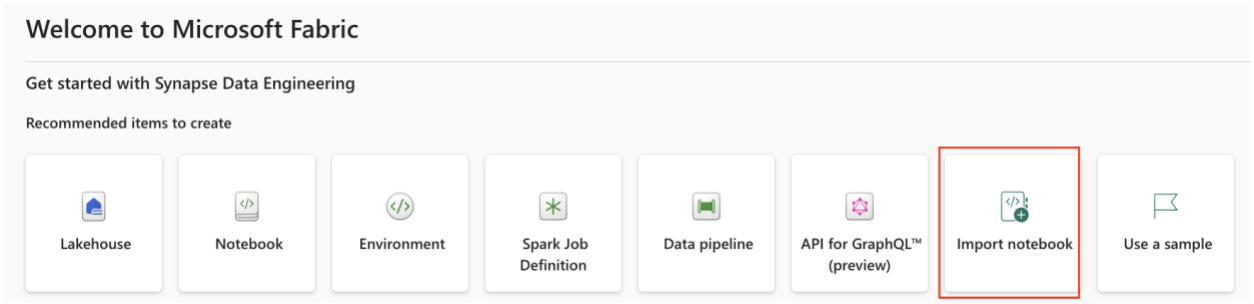
Part 3: Importing a Notebook

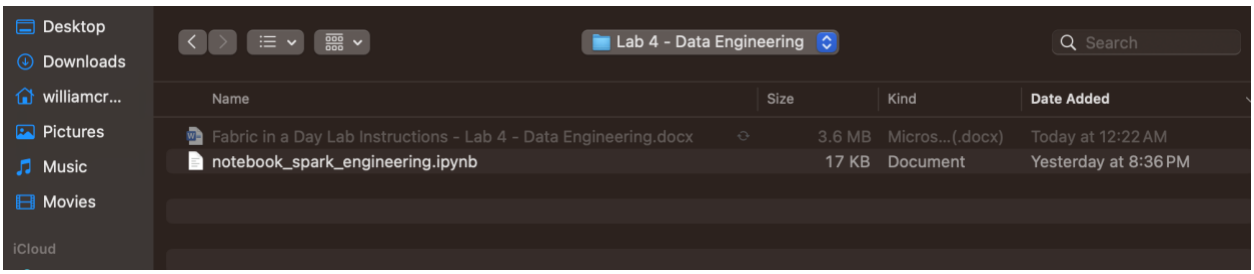
Importing a Notebook: In this section, the **Notebook** will serve as the primary documentation source, complete with comments for visibility and clarity, along with code examples that can be referenced to complete each exercise. The notebook used for these exercises is the **notebook_spark_engineering.ipynb** 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.

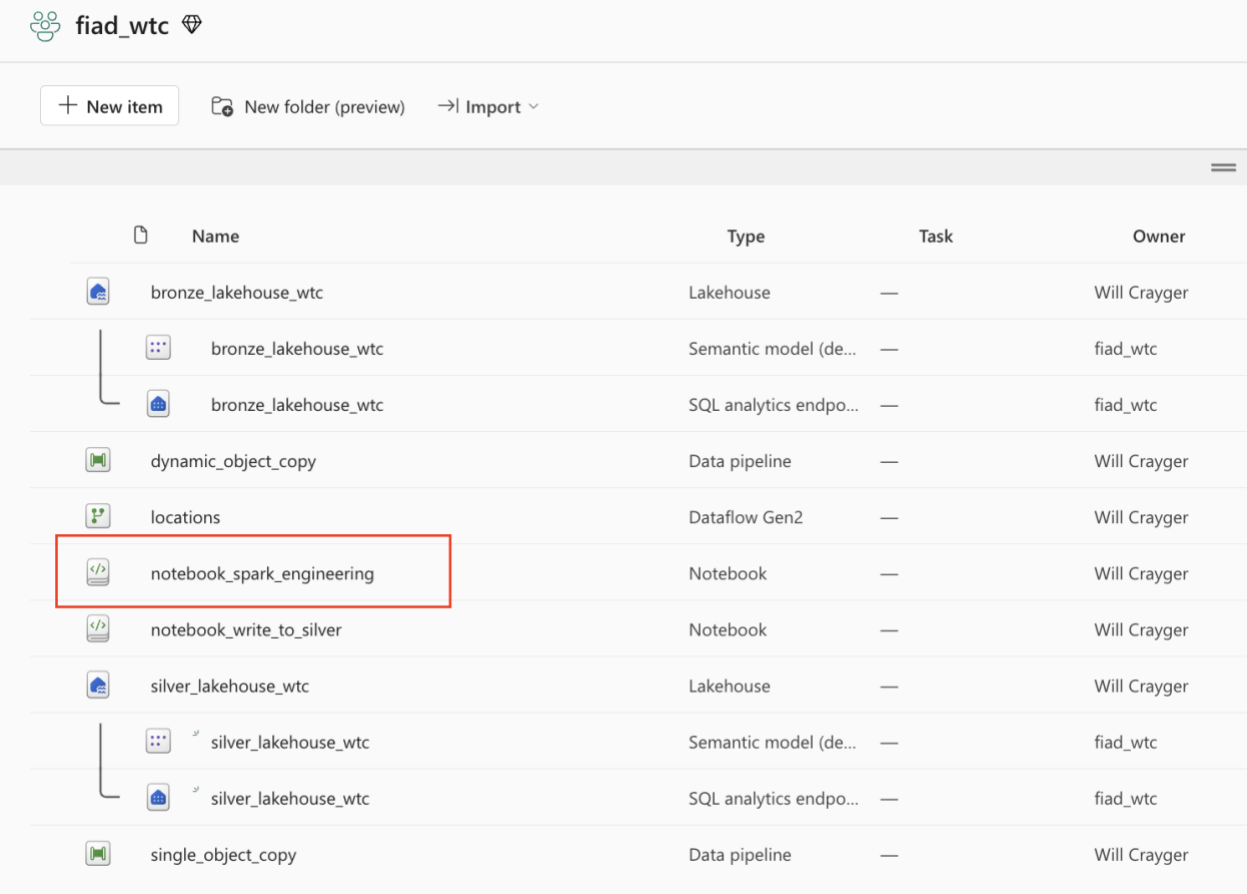

notebook_write_to_silver
bronze_lakehouse_wtc
silver_lakehouse_wtc
Power BI
Fabric Training Development
Microsoft Fabric
Power BI
Databases
Data Factory
Data Activator
Industry Solutions
Real-Time Intelligence
Synapse
Data Engineering
Data Science
Data Warehouse
ata
nent Sandbox
a Day
space
a Day - Zero to Hero
ehouse_wc
se_WC
kehouse_WC

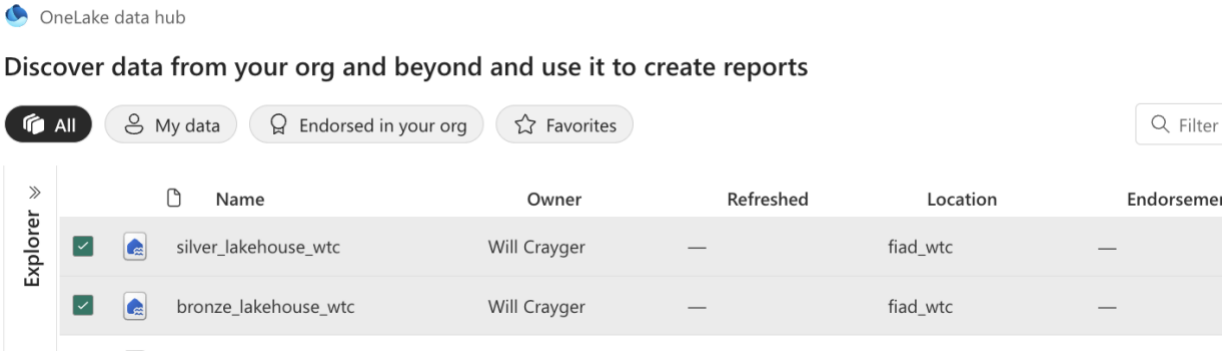

Welcome to Microsoft Fabric
Get started with Synapse Data Engineering
Recommended items to create
Lakehouse
Notebook
Environment
Spark Job Definition
Data pipeline
API for GraphQL™ (preview)
Import notebook
Use a sample

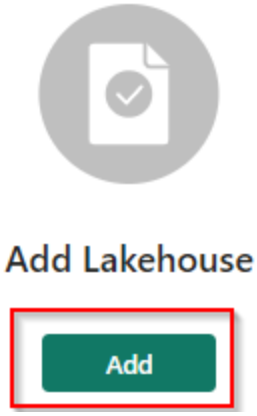

Desktop
Downloads
williamcr...
Pictures
Music
Movies
iCloud
Lab 4 - Data Engineering
Search
Name
Size
Kind
Date Added
Fabric in a Day Lab Instructions - Lab 4 - Data Engineering.docx
3.6 MB
Micros...(.docx)
Today at 12:22 AM
notebook_spark_engineering.ipynb
17 KB
Document
Yesterday at 8:36 PM

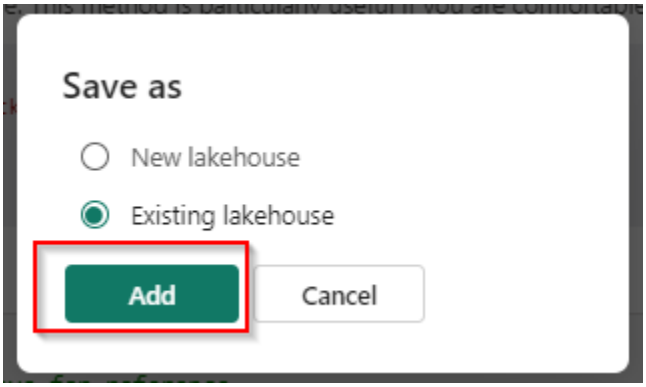
Attach the Notebook to Bronze and Silver Lakehouses: Navigate back to your workspace and click the notebook to open it. In the **Add Lakehouse** blade, click **Add**. Select **Existing Lakehouse** and click **Add** again. Tick both **Bronze Lakehouse** and **Silver Lakehouse**, then click **Add**.

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





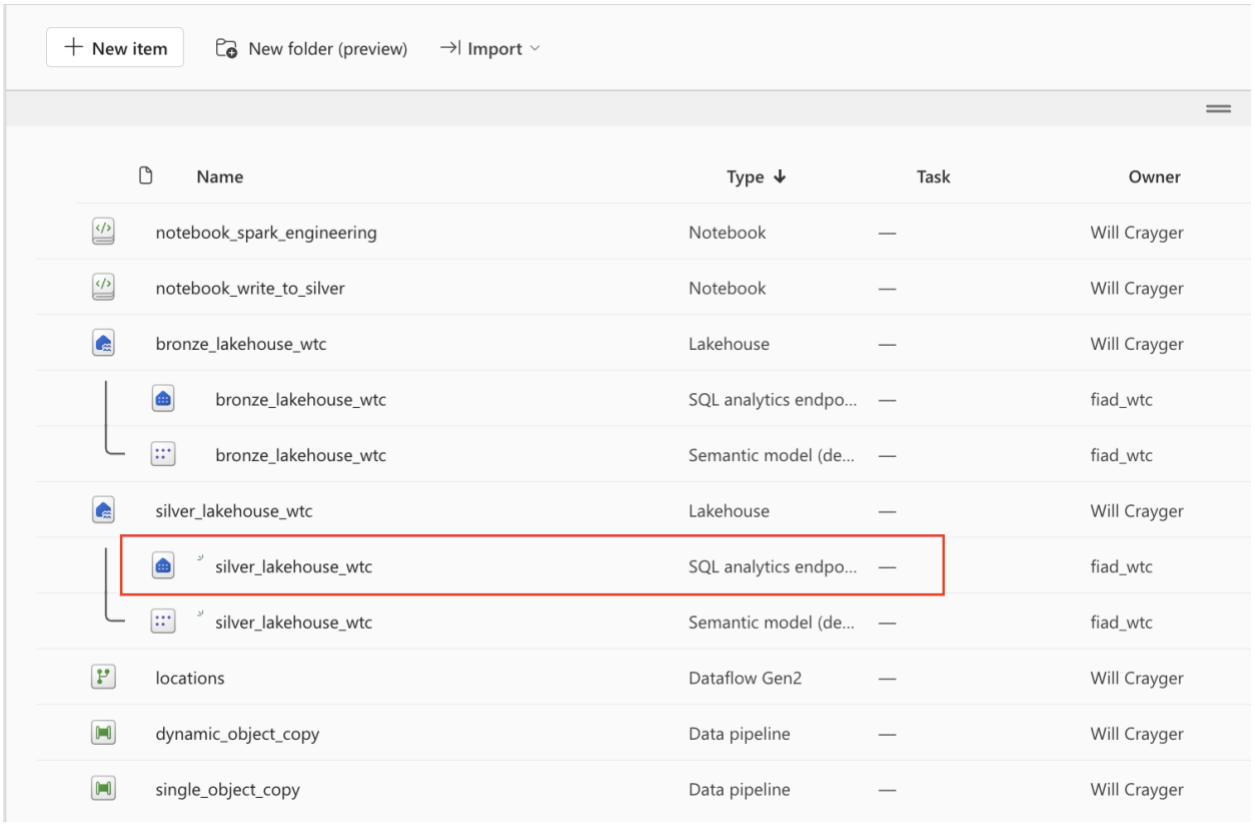


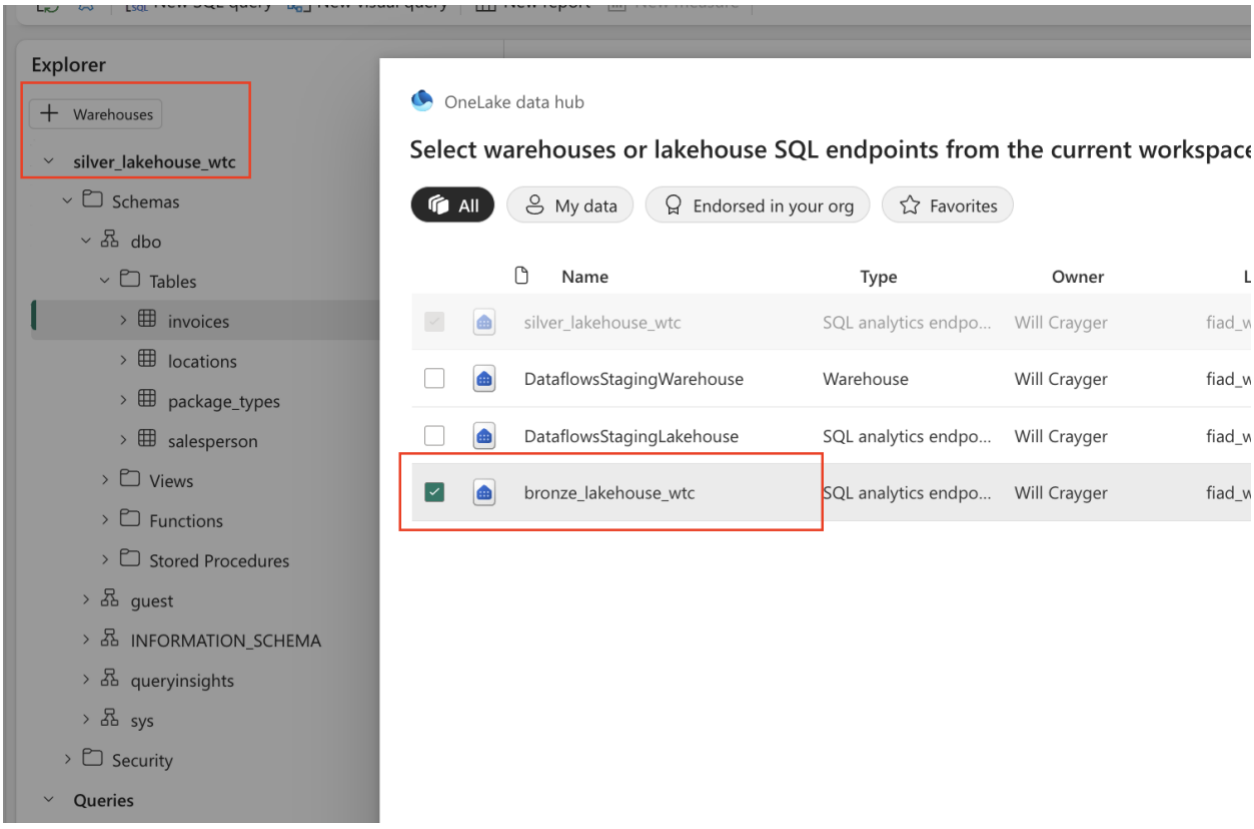


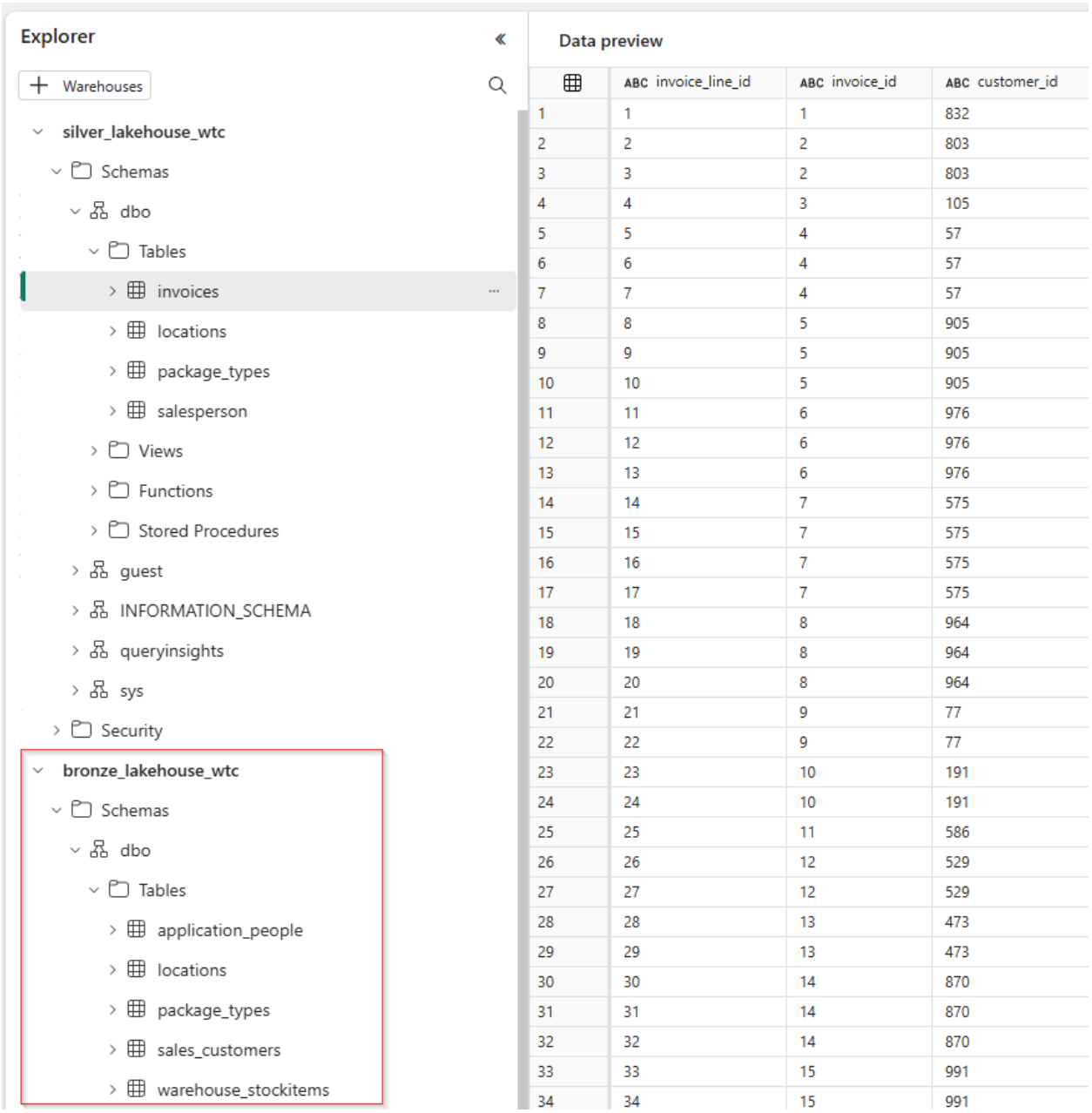
Part 4: Data Engineering via Lakehouse SQL Endpoint

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.


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

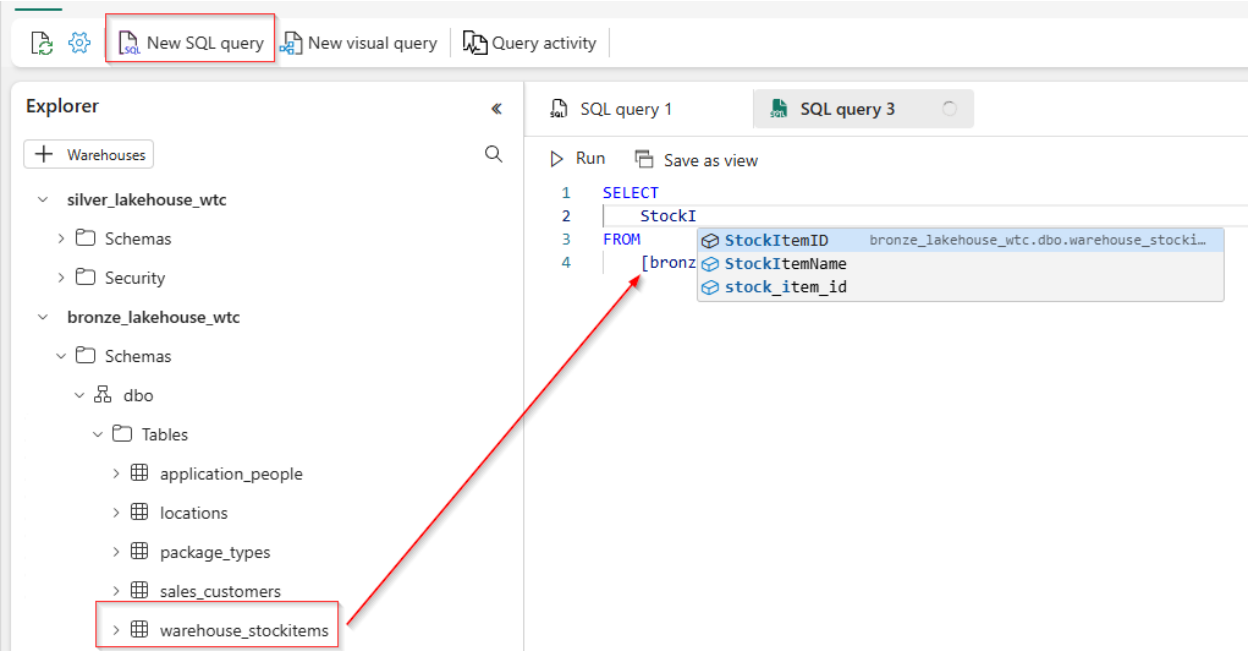

Explorer
Warehouses
silver_lakehouse_wtc
Schemas
dbo
Tables
invoices
locations
package_types
salesperson
Views
Functions
Stored Procedures
guest
INFORMATION_SCHEMA
queryinsights
sys
Security
Queries
OneLake data hub
Select warehouses or lakehouse SQL endpoints from the current workspace
All
My data
Endorsed in your org
Favorites
Name
Type
Owner
silver_lakehouse_wtc
SQL analytics endpo...
Will Crayger
DataflowsStagingWarehouse
Warehouse
Will Crayger
DataflowsStagingLakehouse
SQL analytics endpo...
Will Crayger
bronze_lakehouse_wtc
SQL analytics endpo...
Will Crayger


Explorer
Warehouses
silver_lakehouse_wtc
Schemas
dbo
Tables
invoices
locations
package_types
salesperson
Views
Functions
Stored Procedures
guest
INFORMATION_SCHEMA
queryinsights
sys
Security
bronze_lakehouse_wtc
Schemas
dbo
Tables
application_people
locations
package_types
sales_customers
warehouse_stockitems
Data preview
ABC invoice_line_id
ABC invoice_id
ABC customer_id
1 1 1 832
2 2 2 803
3 3 2 803
4 4 3 105
5 5 4 57
6 6 4 57
7 7 4 57
8 8 5 905
9 9 5 905
10 10 5 905
11 11 6 976
12 12 6 976
13 13 6 976
14 14 7 575
15 15 7 575
16 16 7 575
17 17 7 575
18 18 8 964
19 19 8 964
20 20 8 964
21 21 9 77
22 22 9 77
23 23 10 191
24 24 10 191
25 25 11 586
26 26 12 529
27 27 12 529
28 28 13 473
29 29 13 473
30 30 14 870
31 31 14 870
32 32 14 870
33 33 15 991
34 34 15 991

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.

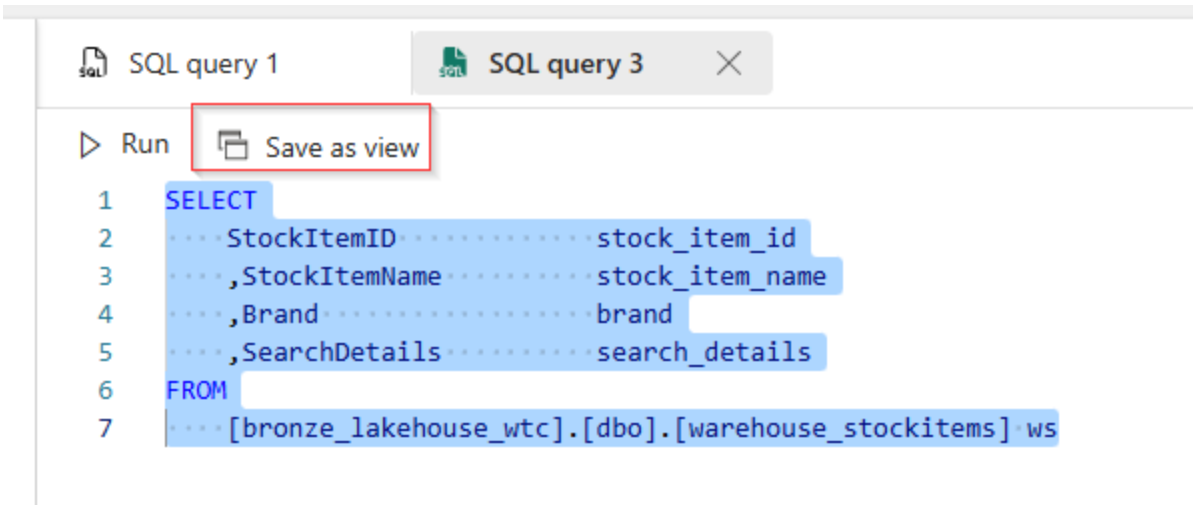
```
SELECT
    StockItemID         AS stock_item_id
    ,StockItemName      AS stock_item_name
    ,Brand              AS brand
    ,SearchDetails      AS search_details
FROM [bronze_lakehouse_wtc].[dbo].[warehouse_stockitems]
```

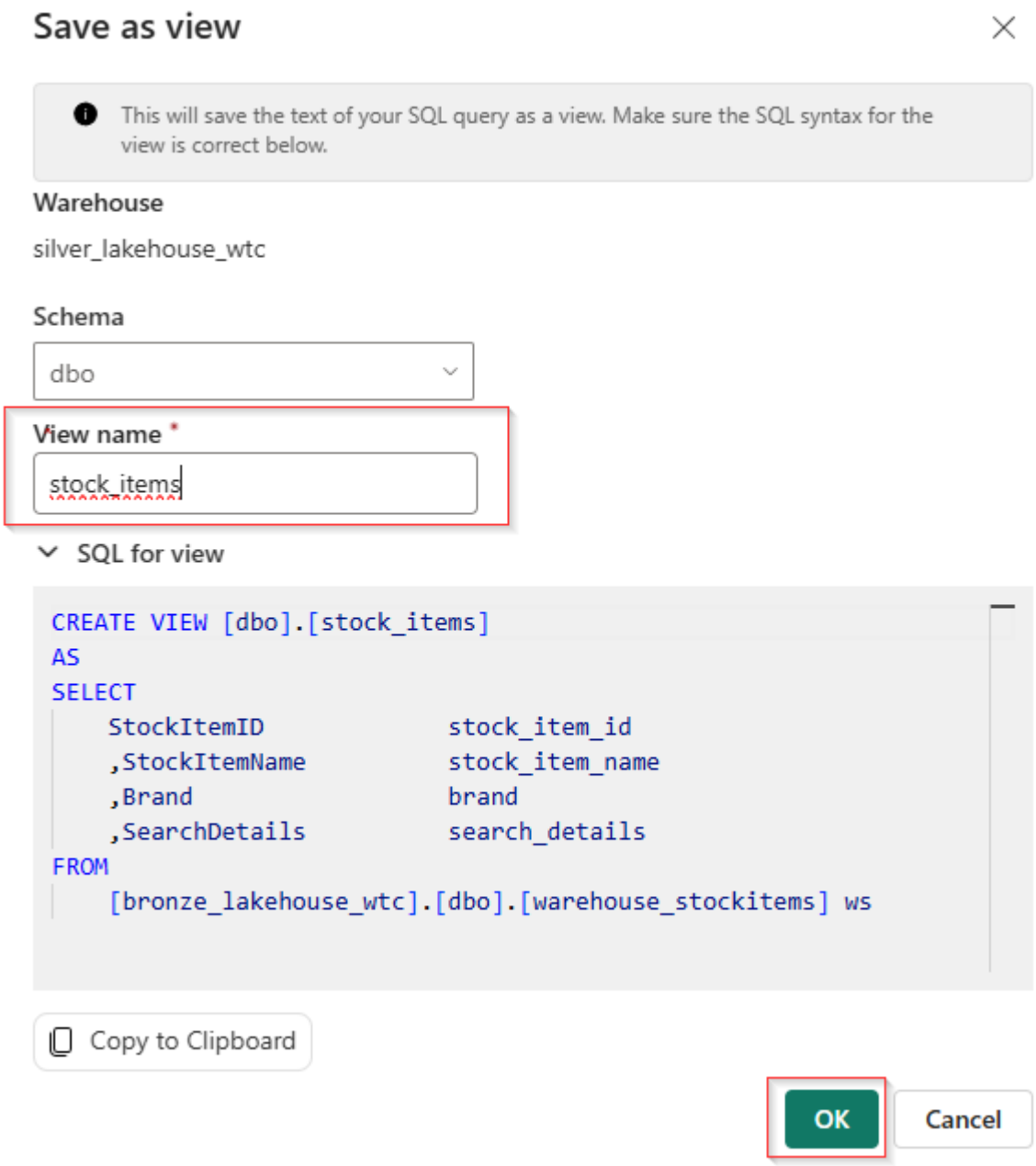
If pasting the above code you'll need to replace the reference of the **bronze_lakehouse** name.



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**.

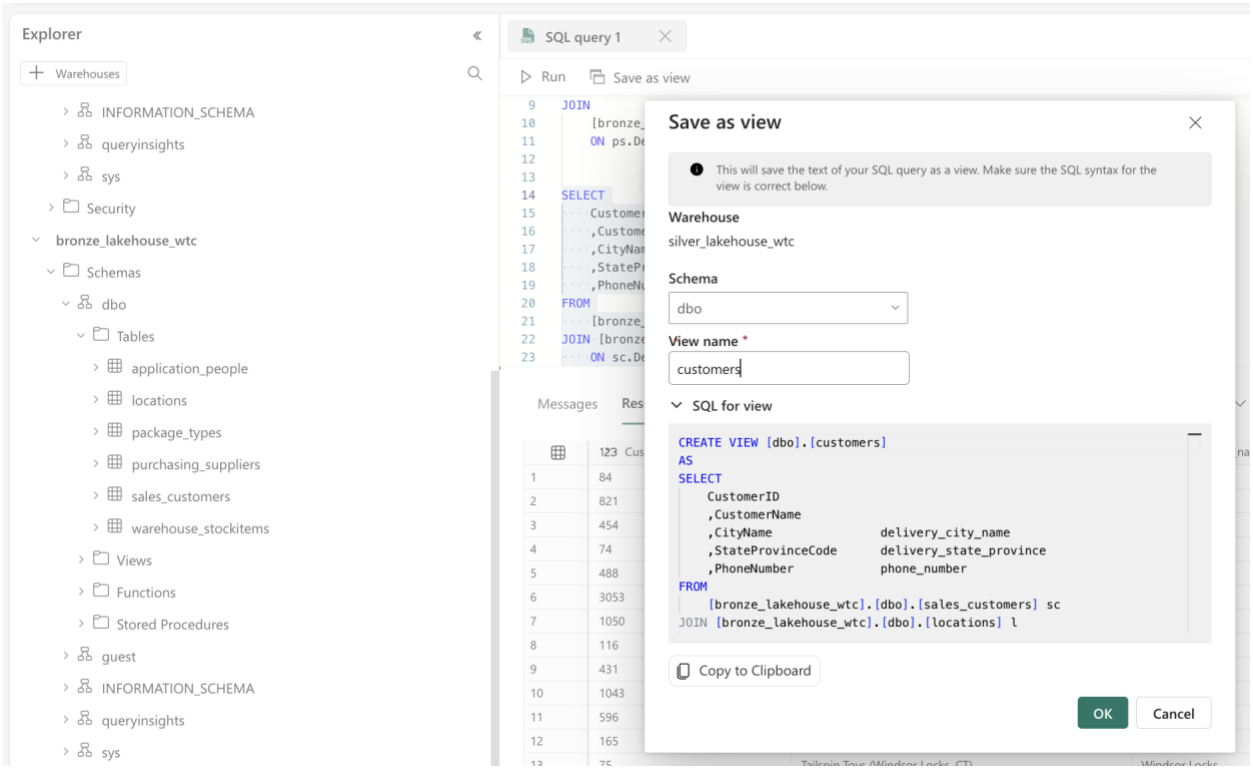

SQL query 1
SQL query 3
Run
Save as view
SELECT
StockItemID stock_item_id
,StockItemName stock_item_name
,Brand brand
,SearchDetails search_details
FROM
[bronze_lakehouse_wtc].[dbo].[warehouse_stockitems] ws



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.



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