

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

**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 Engineering – Data Transformation and Engineering in Fabric

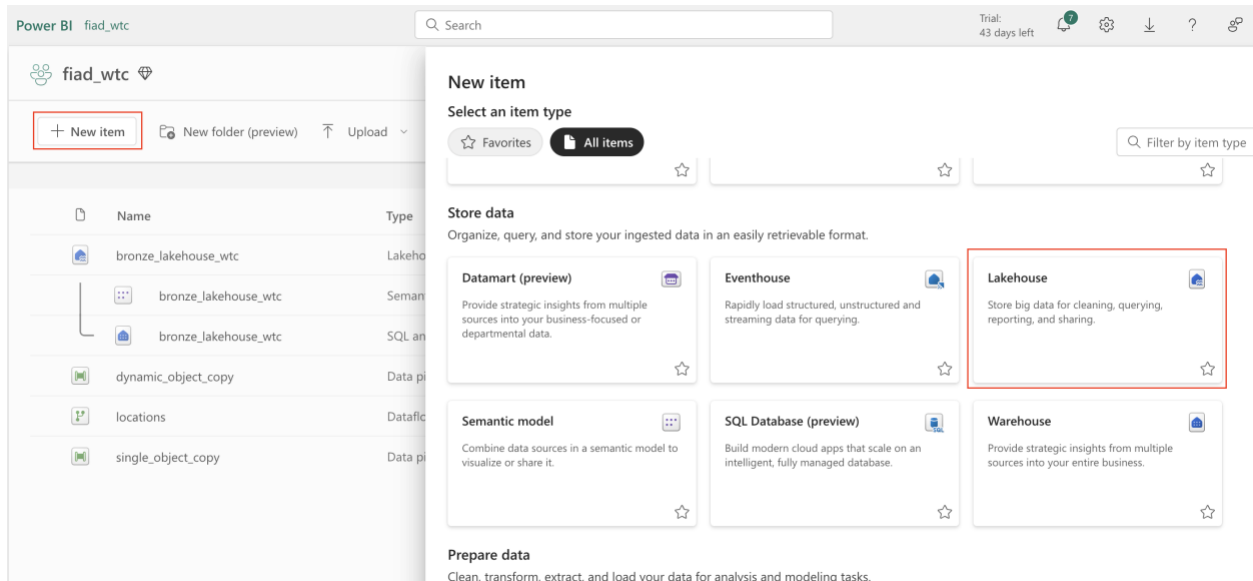
## Introduction

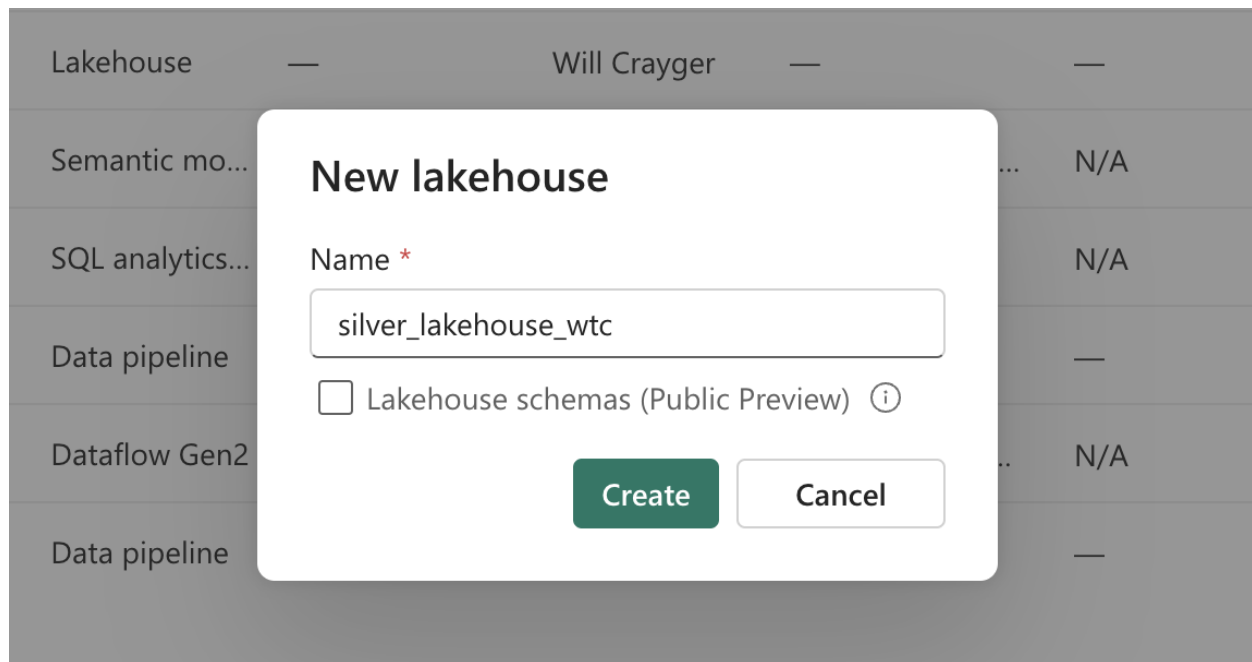
In this lab, you're going to create a **Silver** Lakehouse and begin loading data from the **Bronze** Lakehouse that was created in **Lab 1**. As you proceed through the lab, please note that you're working in a case-sensitive environment.

## 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.

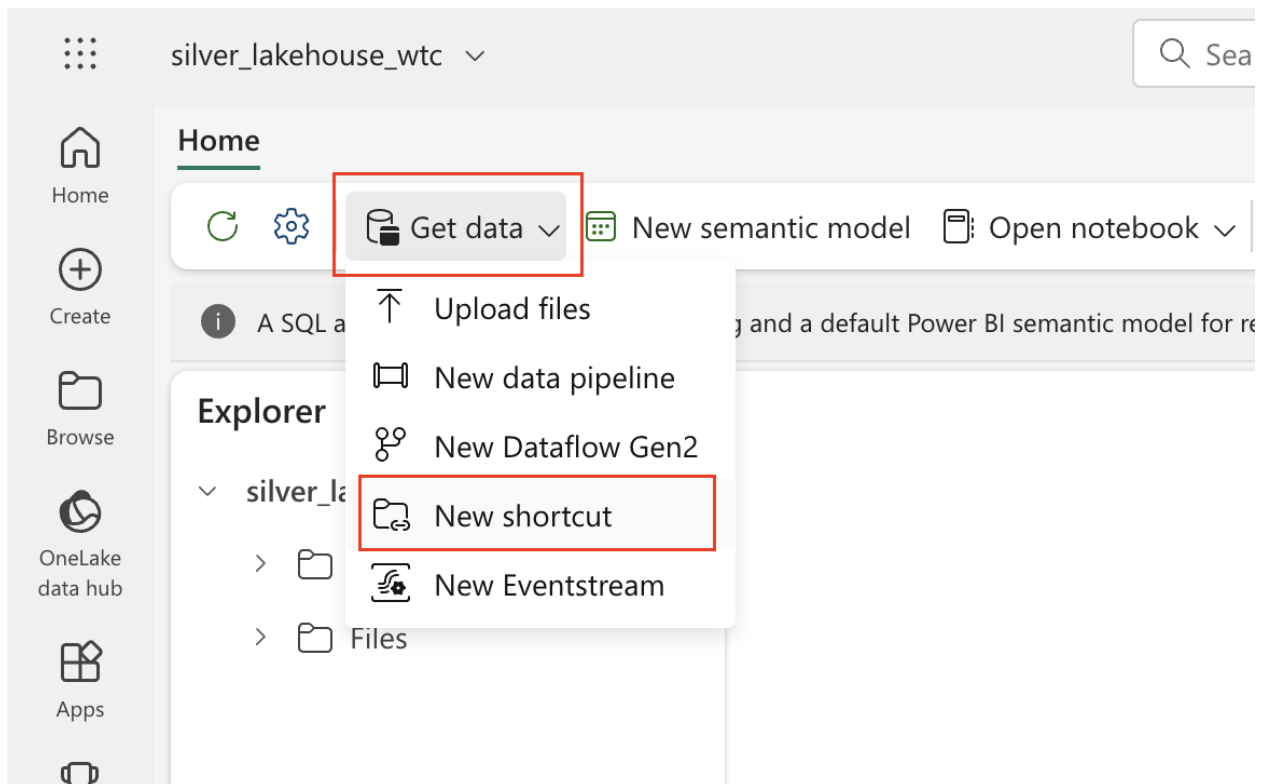
To begin, select **New item** from your Workspace home page. Scroll down and select **Lakehouse** from the item list. From the tiles at the top of the page, select **Lakehouse**. Name your lakehouse **silver\_lakehouse\_<your\_initials>** and click **Create**.





**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**.




## 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


 **silver\_lakehouse\_wtc** is located in the region **East US 2**. Any data sourced through this shortcut will be processed in the same region.

 **Azure Data Lake Storage**  
Gen2  
Azure  
[Learn more](#) 

### Connection settings

URL \* 

### Connection credentials

Connection  
 

Connection name

Authentication kind








SAS token

## New shortcut

 **silver\_lakehouse\_wtc** is located in the region **East US 2**. Any data sourced through this shortcut will be processed in the same region.

 **Azure Data Lake Storage Gen2**

Select a bucket or directory

- ✓ ☐  fabcon
- ✓ ☐  synapse
- ✓ ☐  workspaces
- ✓ ☐  synw-fabcon-eu
  - > ☐  sparkpools
- ✓ ☐  warehouse
  - > ☒  invoices

Explorer		invoices					
▼	silver_lakehouse_wtc						
▼	Tables						
>	invoices						
▼	Files						

	ABC	invoice_line_id	ABC	invoice_id	ABC	customer_id	ABC	bill_to_customer_id	ABC	delivery_meth...	ABC	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

	ABC	invoice_line_id	ABC	invoice_id	ABC	customer_id	ABC	bill_to_customer_id	ABC	delivery_meth...	ABC	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

PySpark (Python) Workspace default Data Wrangler Copilot

Name

notebook\_write\_to\_silver

Location

fiad\_wtc

Owner

Will Crayger

Description

New notebook

Show more

```
1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3
```

notebook\_write\_to\_silver

Home Edit Run View

Run all Connect PySpark (Python) Workspace default Data Wrangler

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 Edit Run View

Run all Connect PySpark (Python) Workspace default Data Wrangler

All sources

Lakehouses  
+ Lakehouse

silver\_lakehouse\_wtc

Tables

invoices

Files

Add Lakehouse

☐ New Lakehouse

☒ Existing Lakehouse

Add Cancel

```
1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3
```



OneLake data hub

Discover data from your org and beyond and use it to create reports

All My data Endorsed in your org Favorites

fiad\_ All domains

Name	Owner	Refreshed	Location	Endorsement	Sensitivity
silver_lakehouse_wtc	Will Crayger	—	fiad_wtc	—	—
bronze_lakehouse_wtc	Will Crayger	—	fiad_wtc	—	—
DataflowsStagingLakehouse	Will Crayger	—	fiad_wtc	—	—

Add Cancel

notebook\_write\_to\_silver | Saved

Search

Home Edit Run View

PySpark (Python) Workspace default Data Wrangler Copilot

All sources

Lakehouses

bronze\_lakehouse\_wtc

Tables

- application\_people
- locations
- package\_types
- purchasing\_suppliers
- sales\_customers
- warehouse\_stockitems

Files

```

1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3

```

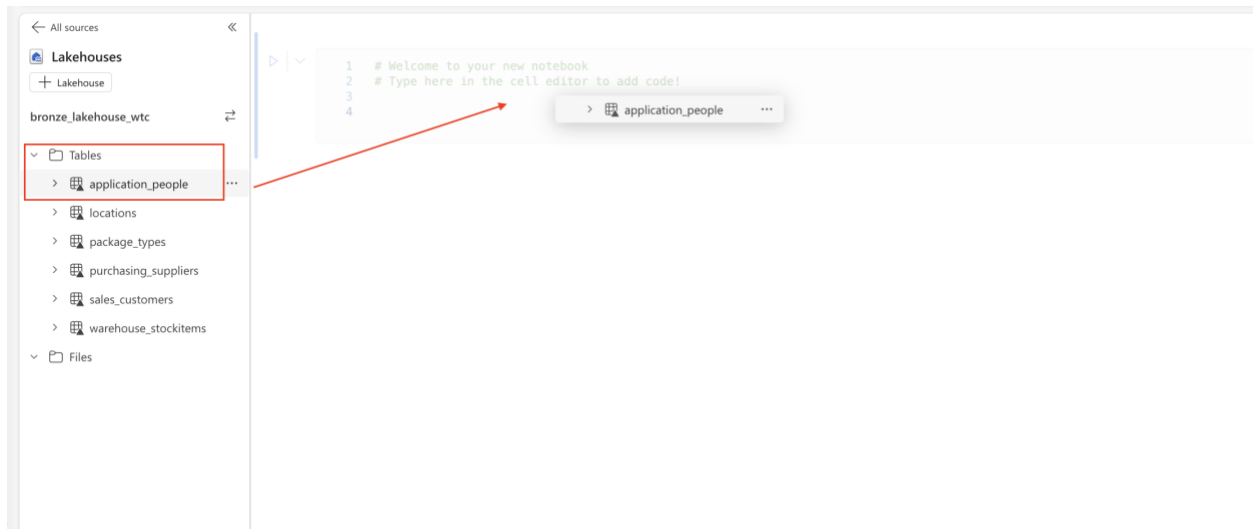
PySpark (Python)

**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)
```



Home Edit Run View

Standard session PySpark (Python) Workspace default Data Wrangler Copilot

All sources

Lakehouses

+ Lakehouse

bronze\_lakehouse\_wtc

Tables

- application\_people
- locations
- package\_types
- purchasing\_suppliers
- sales\_customers
- warehouse\_stockitems

Files

```

1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3
4 df = spark.sql("SELECT * FROM bronze_lakehouse_wtc.application_people")
5 display(df)
6

```

3 min 3 sec - Session ready in 2 min 23 sec 181 ms. Command executed in 39 sec 957 ms by Will Crayger on 11:48:45 PM, 9/10/24

Spark jobs (6 of 6 succeeded) Resources Log

Table Chart Download Showing rows 1 - 1000

PersonID	ABC FullName	ABC PreferredName	ABC SearchName	0/1 IsPermittedToLogon	ABC LogonName	0/1 IsExternalLogonProvider	ANY HashedPassword	0/1 IsSystemUser
1	Hudson Onslow	Hudson	Hudson Hudson ...	true	hudsono@wide...	false	I2aMzFeQFeqRzbD...	true
2	Eva Muirden	Eva	Eva Eva Muirden	false	evam@wideworl...	false	5oLTbkO2o5QO1KtL...	true
3	Alica Fatnowna	Alica	Alica Alica Fatno...	true	alica@wideworl...	false	ftqwjprFdMWXpGd...	true
4	Stella Rosenhain	Stella	Stella Stella Rose...	true	stellar@wideworl...	false	GG51WlflvczGCHog4...	true
5	Ethan Onslow	Ethan	Ethan Ethan Onsl...	true	ethano@widewo...	false	1w839cAZ5ZiZ3fmH...	true
6	Archer Lamble	Archer	Archer Archer La...	false	archerl@widewo...	false	Bhh/aGMSlUEbAixs...	true
7	Piper Koch	Piper	Piper Piper Koch	true	piperk@widewor...	false	YdYQgo5V6ileWuL/pj...	true
8	Katie Darwin	Katie	Katie Katie Darwin	true	katiel@widewor...	false	R60ShnhF1b7Fk3pl...	true
9	Jack Potter	Jack	Jack Jack Potter	true	jackp@wideworl...	true	gCto8BwFW9mP5Gm...	true
10	Kayla Woodcock	Kayla	Kayla Kayla Woo...	true	kaylaw@widewo...	false	YW6bVY2U5i/FNeA6...	true
11	Isabella Rupp	Isabella	Isabella Isabella ...	true	isabellar@widew...	false	IF58TJDL6mioxYbnb...	true
12	Sophia Hinton	Sophia	Sophia Sophia HL...	true	sophiah@widew...	false	RRuxCfFBjMVQN5iS...	true
13	Amy Trefl	Amy	Amy Amy Trefl	true	amyt@wideworl...	false	epK76oMMXAn3M...	true
14	Anthony Grosse	Anthony	Anthony Anthon...	true	anthonyg@wide...	false	L94OKPHd3jKPRk8z...	true
15	Henry Forlonge	Henry	Henry Henry Forl...	true	henryh@widewor...	false	P35CQYDFT0JajlyjO2l...	true
16	Hudson Hollin...	Hudson	Hudson Hudson ...	true	hudsonh@wide...	false	SsCQYDFT0JajlyjO2l...	true
17	Lily Code	Lily	Lily Lily Code	true	lilyc@widewordi...	false	0AZyTts/ldwILPHHC...	true
18	Taj Shand	Taj	Taj Taj Shand	true	tajsh@widewordi...	false	mu/uwNuw6msl8Ou...	true

Home Edit Run View

Standard session PySpark (Python) Workspace default Data Wrangler Copilot

All sources

Lakehouses

+ Lakehouse

bronze\_lakehouse\_wtc

Tables

- application\_people
- locations
- package\_types
- purchasing\_suppliers
- sales\_customers
- warehouse\_stockitems

Files

```

1 df_salesperson = df.filter('IsSalesperson = true')
2 display(df_salesperson)

```

4 sec - Command executed in 3 sec 593 ms by Will Crayger on 11:59:03 PM, 9/10/24

Spark jobs (2 of 2 succeeded) Resources Log

Table Chart Download Showing rows 1 - 10

PersonID	ABC FullName	ABC PreferredName	ABC SearchName	0/1 IsPermittedToLogon	ABC LogonName	0/1 IsExternalLogonProvider	ANY HashedPassword	0/1 IsSystemUser
1	Hudson Onslow	Hudson	Hudson Hudson ...	true	hudsono@wide...	false	I2aMzFeQFeqRzbD...	true
2	Archer Lamble	Archer	Archer Archer La...	false	archerl@widewo...	false	Bhh/aGMSlUEbAixs...	true
3	Jack Potter	Jack	Jack Jack Potter	true	jackp@wideworl...	false	gCto8BwFW9mP5Gm...	true
4	Kayla Woodcock	Kayla	Kayla Kayla Woo...	true	kaylaw@widewo...	false	YW6bVY2U5i/FNeA6...	true
5	Sophia Hinton	Sophia	Sophia Sophia HL...	true	sophiah@widew...	false	RRuxCfFBjMVQN5iS...	true
6	Amy Trefl	Amy	Amy Amy Trefl	true	amyt@wideworl...	false	epK76oMMXAn3M...	true
7	Anthony Grosse	Anthony	Anthony Anthon...	true	anthonyg@wide...	false	L94OKPHd3jKPRk8z...	true
8	Hudson Hollin...	Hudson	Hudson Hudson ...	true	hudsonh@wide...	false	SsCQYDFT0JajlyjO2l...	true
9	Lily Code	Lily	Lily Lily Code	true	lilyc@widewordi...	false	0AZyTts/ldwILPHHC...	true
10	Taj Shand	Taj	Taj Taj Shand	true	tajsh@widewordi...	false	mu/uwNuw6msl8Ou...	true

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')
```

```
1 # Assign silver lakehouse variable value (e.g. silver_lakehouse = 'silver_lakehouse_wtc')
2 silver_lakehouse = '<your_silver_lakehouse_name>'
3
4 # Write dataframe to silver
5 df_salesperson.write.format('delta').mode('overwrite').saveAsTable(f'{silver_lakehouse}.salesperson')
```

[3] ✓ 14 sec - Command executed in 14 sec 374 ms by Will Crayger on 1:08:58 PM, 9/20/24

> 🔍 Spark jobs (9 of 9 succeeded) 📄 Resources

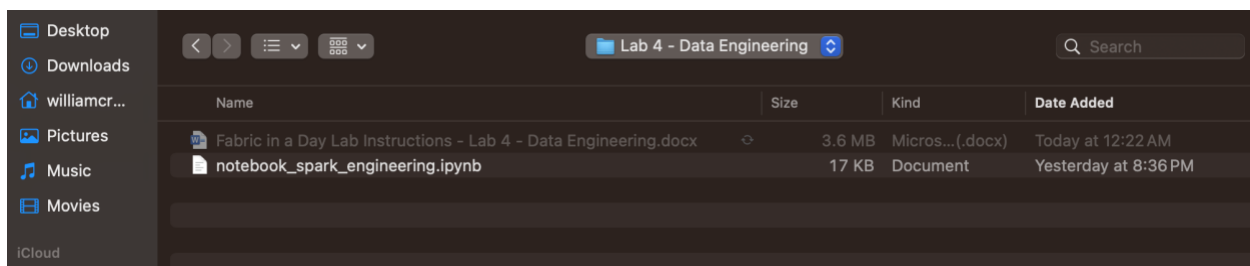
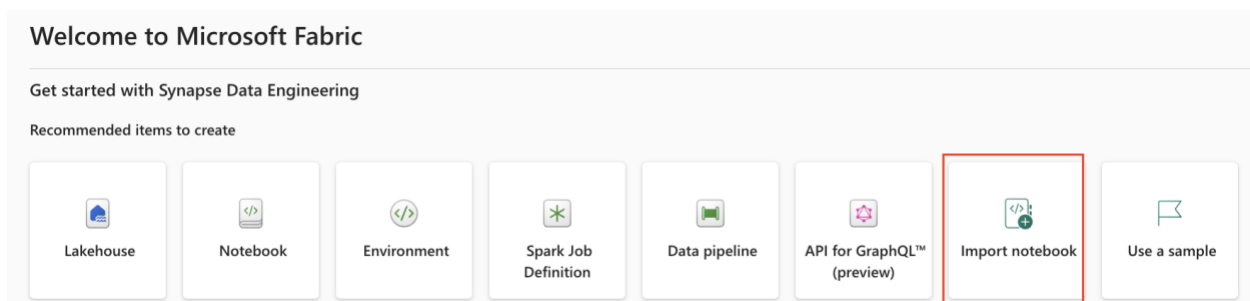
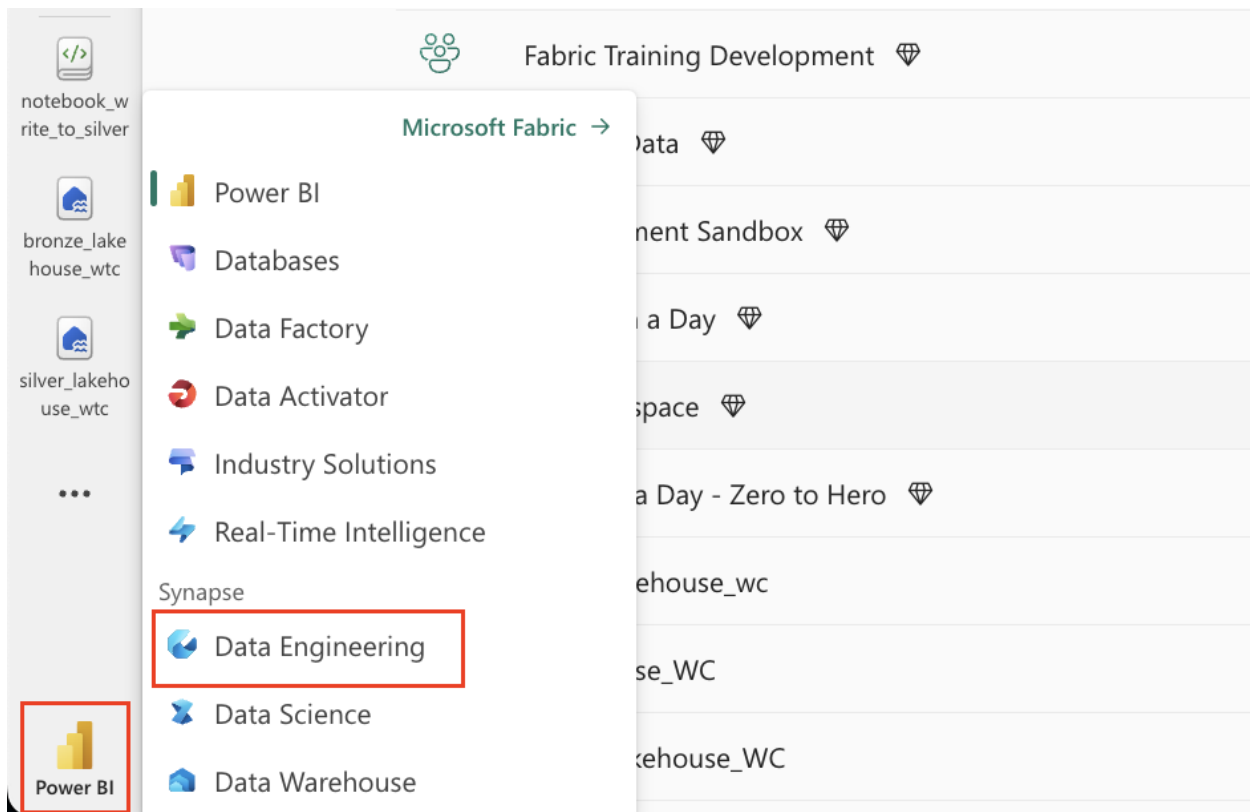
...

### 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.



**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.

fiad\_wtc

+ New item   New folder (preview)   → Import

	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

OneLake data hub

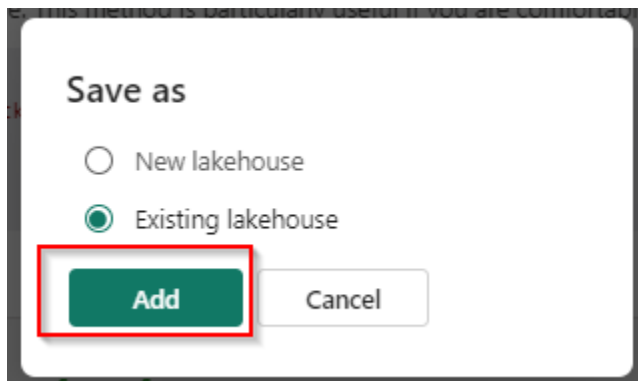
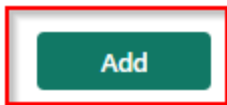
Discover data from your org and beyond and use it to create reports

All   My data   Endorsed in your org   Favorites   Filter

	Name	Owner	Refreshed	Location	Endorsemei
✓	silver_lakehouse_wtc	Will Crayger	—	fiad_wtc	—
✓	bronze_lakehouse_wtc	Will Crayger	—	fiad_wtc	—












## Add Lakehouse



### 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.

<div> <div>+ New item</div> <div>New folder (preview)</div> <div>→ Import</div> </div>				
	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	L
<input checked="" type="checkbox"/>	silver_lakehouse_wtc	SQL analytics endpo...	Will Crayger	fiad_w
<input type="checkbox"/>	DataflowsStagingWarehouse	Warehouse	Will Crayger	fiad_w
<input type="checkbox"/>	DataflowsStagingLakehouse	SQL analytics endpo...	Will Crayger	fiad_w
<input checked="" type="checkbox"/>	bronze_lakehouse_wtc	SQL analytics endpo...	Will Crayger	fiad_w



## 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

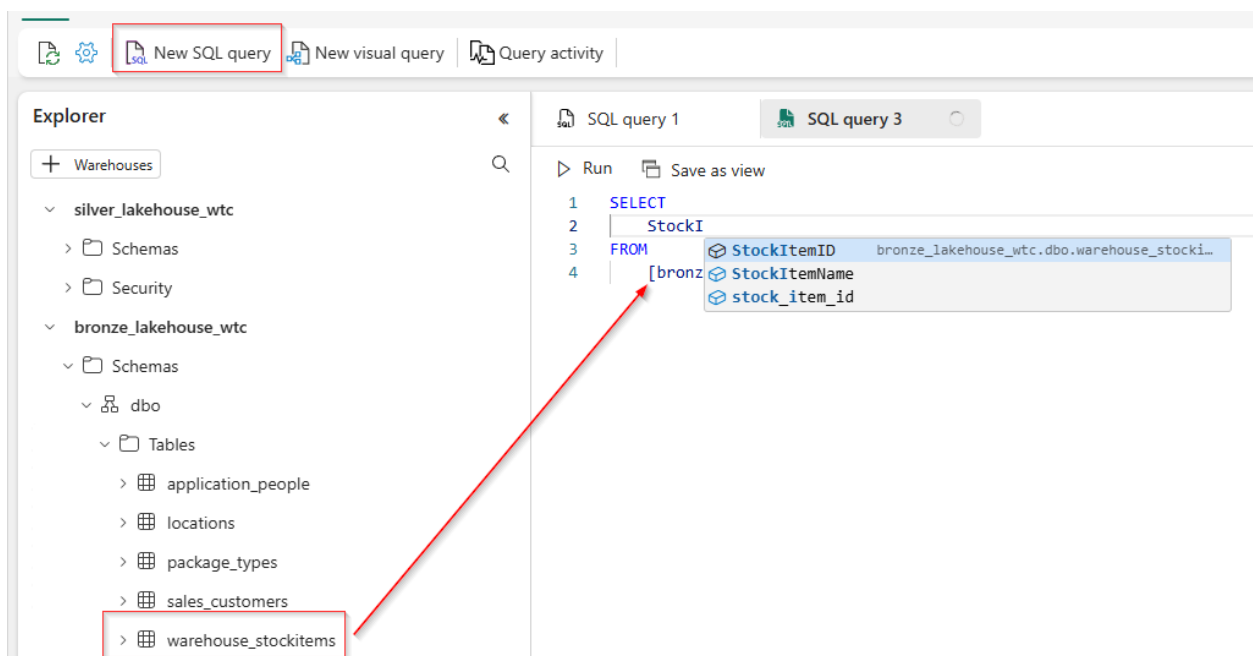
Next, select **New SQL Query** from the action ribbon at the top. Paste in the below T-SQL to query the **warehouse\_stockitems** table from the **Bronze Lakehouse** or **Drag + Drop** the table to the query and begin writing the below query.

**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

```
1 SELECT
2     StockItemID stock_item_id
3     , StockItemName stock_item_name
4     , Brand brand
5     , SearchDetails search_details
6 FROM
7     [bronze_lakehouse_wtc].[dbo].[warehouse_stockitems] ws
```

## Save as view



**i** This will save the text of your SQL query as a view. Make sure the SQL syntax for the view is correct below.

### Warehouse

silver\_lakehouse\_wtc

### Schema

dbo

### View name \*

stock\_items

### SQL for view

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

Copy to Clipboard

OK

Cancel

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
```

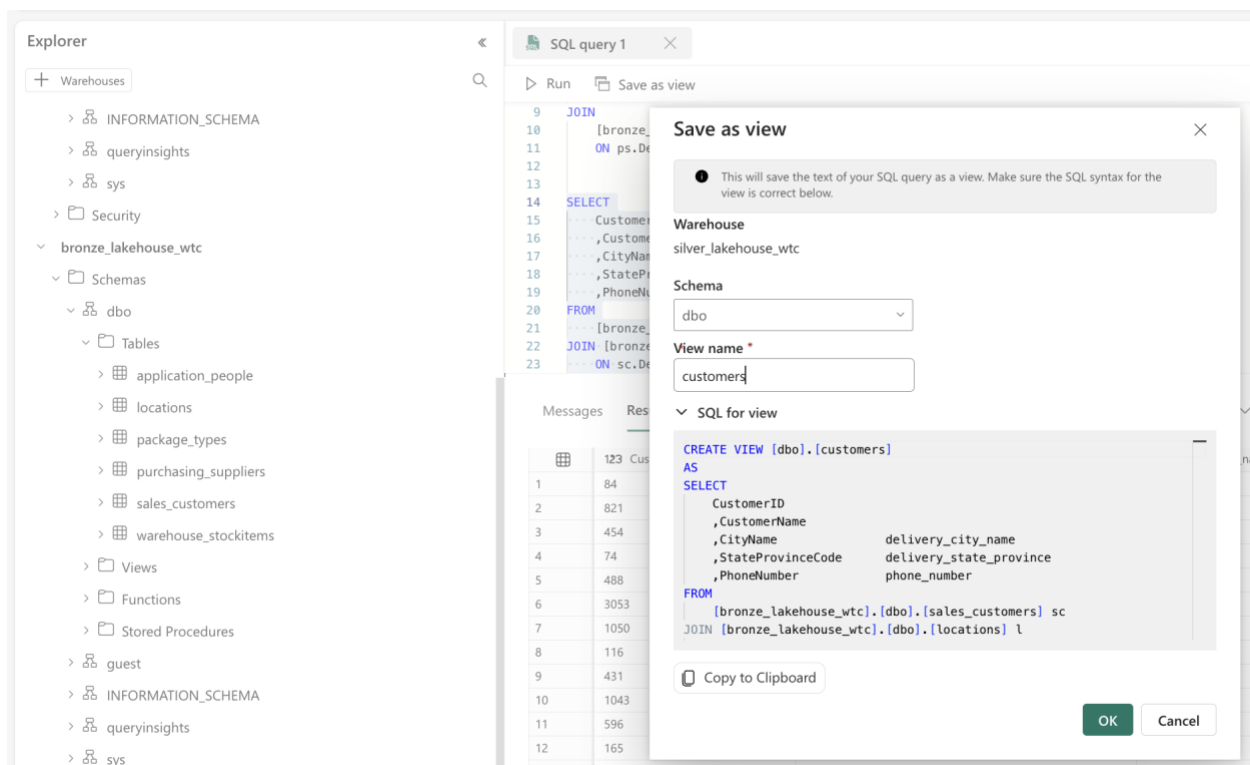
**FROM**

```
[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.