

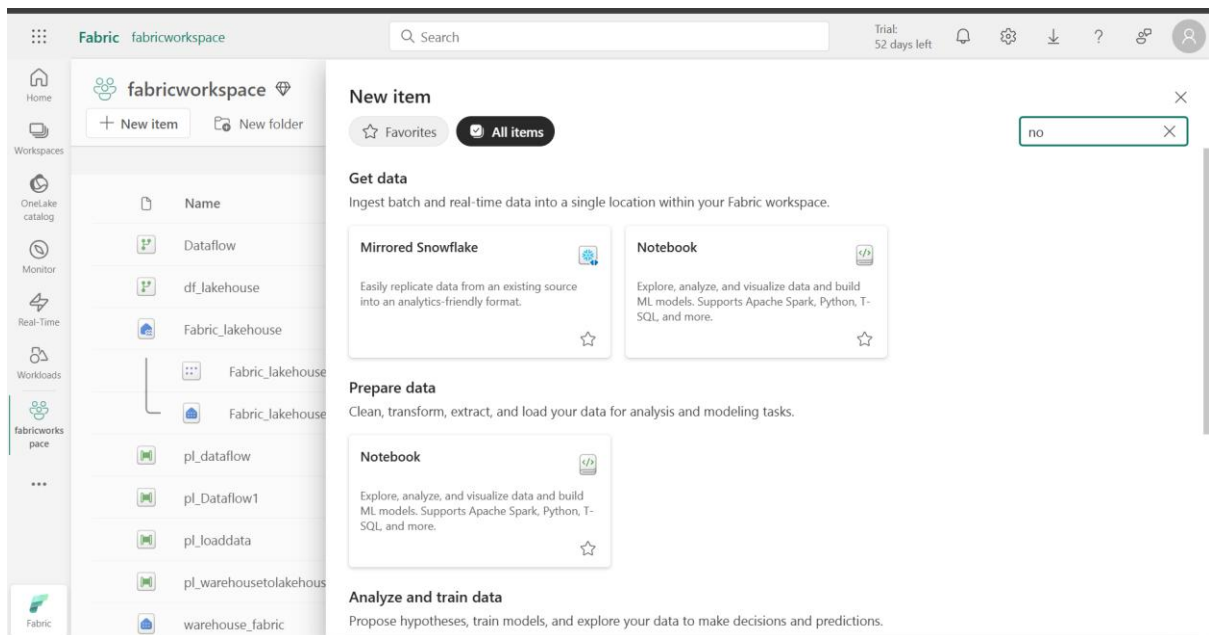
# Fabric Notebook

## Explore the fabric Notebooks

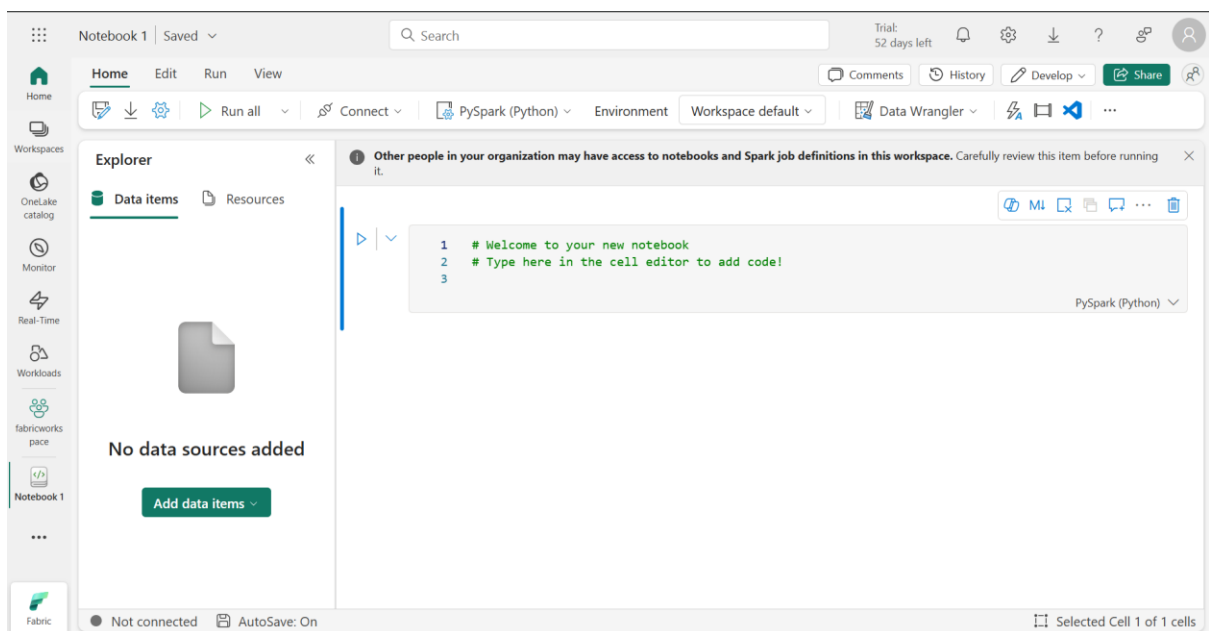
### Notebook:

A **Fabric Notebook** is a feature within Microsoft Fabric that enables users to perform data exploration, transformation, and analysis using interactive code cells. It supports multiple programming languages, including **PySpark**, **SQL**, and **SparkR**, and runs on **Apache Spark** clusters provided by Fabric.

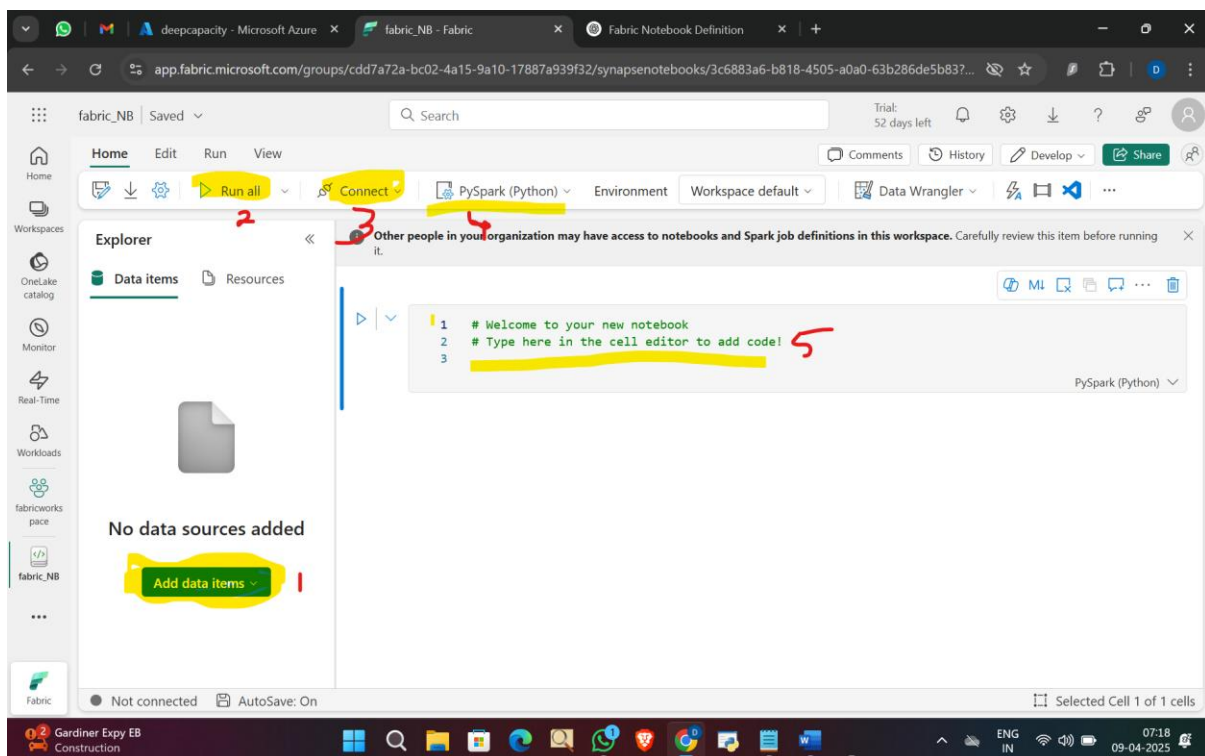
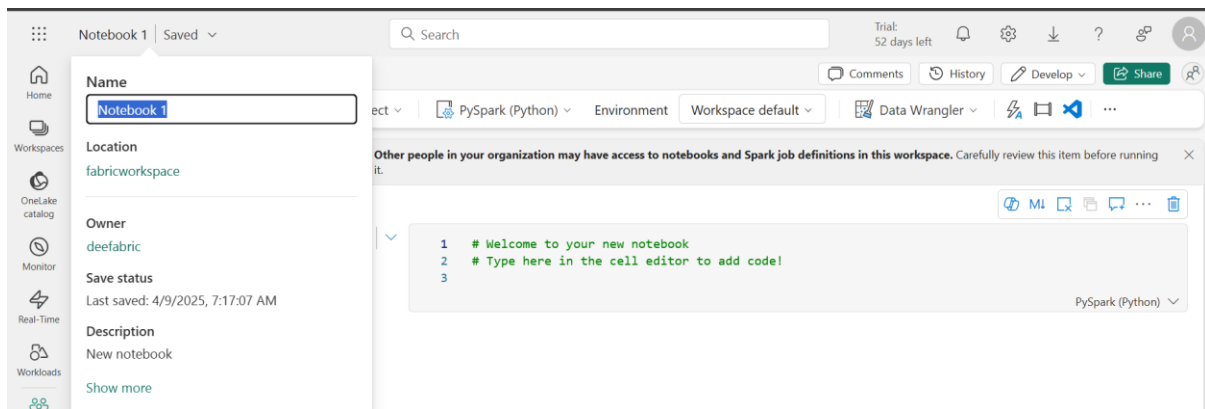
To create a notebook in fabric we need to go to workspace-> new item -> notebook



Notebook is created.



Rename the notebook from top left corner.



1. Add data items, here we will add our source
2. BY using Run all button we can run all code cell.
3. Connect, here we will connect our pool, it can be a starter pool(default) or customized pool.
4. Here we can select language as of now we are using Pyspark.
5. This is where we write code.

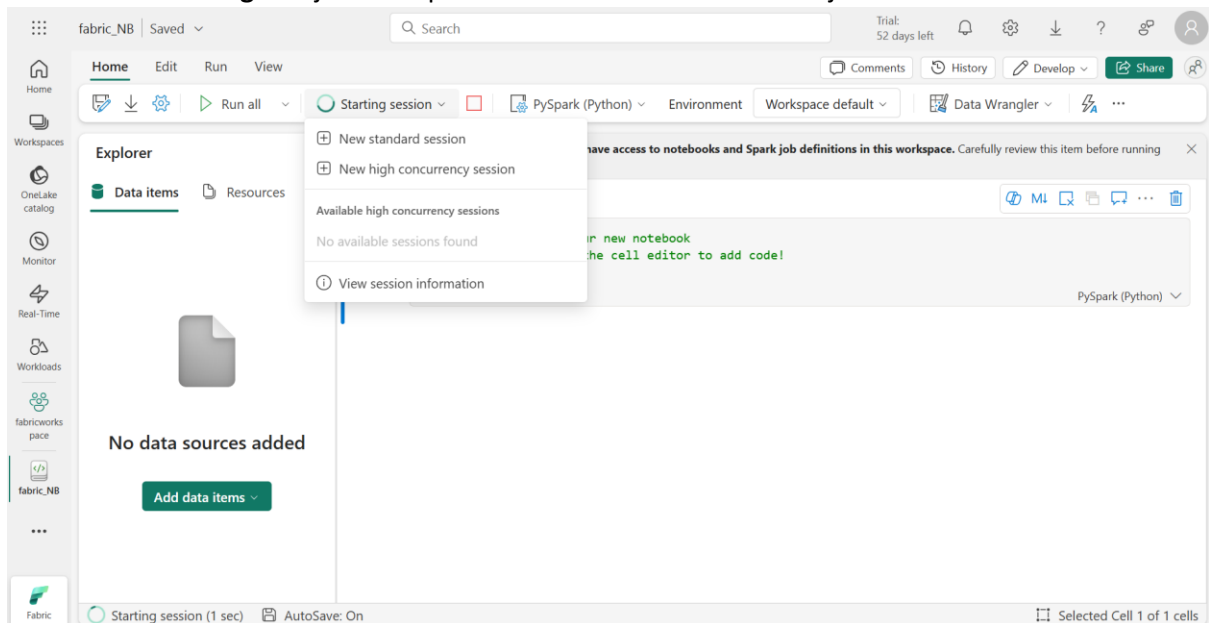
Starter pool:

A **Starter Pool** in Microsoft Fabric is a **pre-warmed Apache Spark compute cluster** that is **automatically available** to users for quick execution of notebooks without the need to create and manage custom Spark pools.

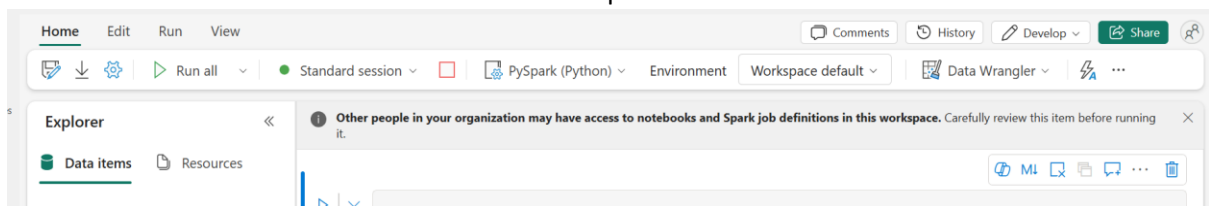
**Key Characteristics:**

- **No manual setup required** – It's ready to use by default in Microsoft Fabric.
- **Optimized for speed** – Because it's pre-warmed, notebooks start faster compared to cold Spark clusters.

- **Cost-efficient** – Good for small to medium workloads and exploratory tasks.
- **Auto-scaling** – Adjusts compute resources as needed for the job.

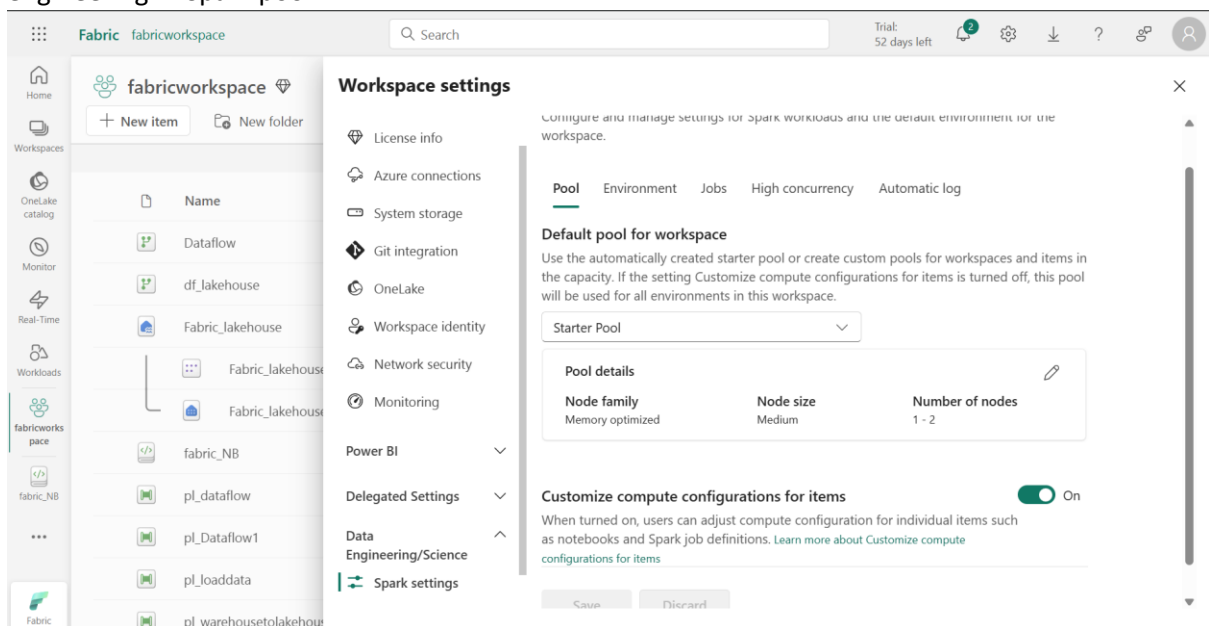


Click on New Standard Session to start the Starter pool.

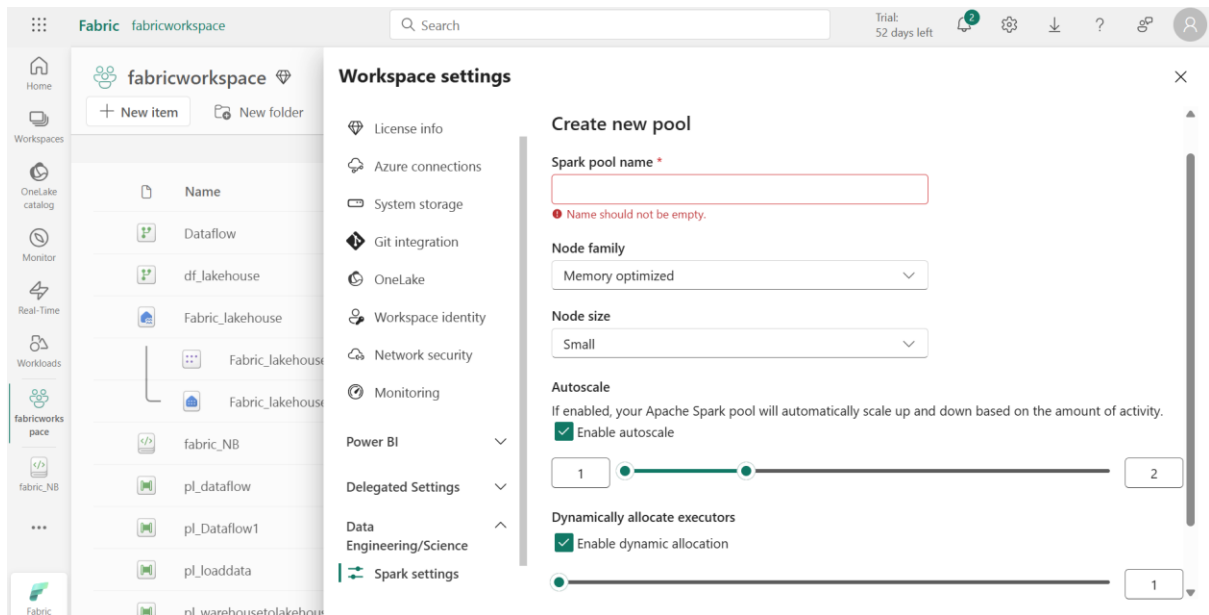


It is started, it takes only few seconds for starter pool to be up and running, and once it starts running billing will start.

To create a custom pool, we need to go to fabric workspace -> workspace settings-> Data engineering -> Spark pool

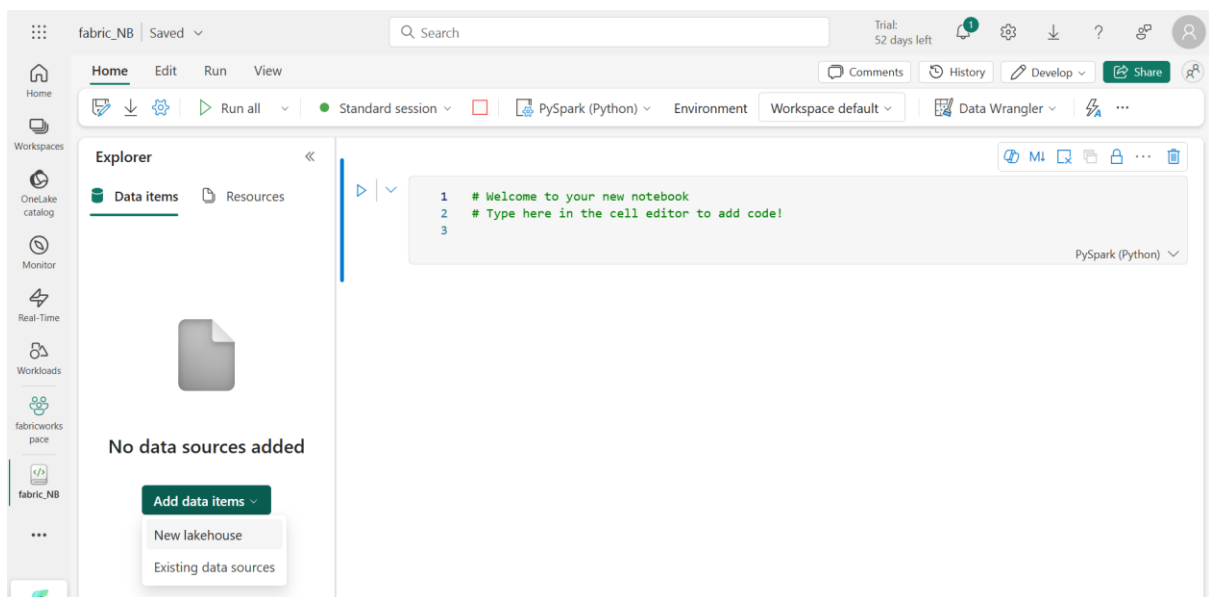


Here we can add our custom pool.



## Load data from lake house to warehouse

First, we will add source as lake house to notebook.



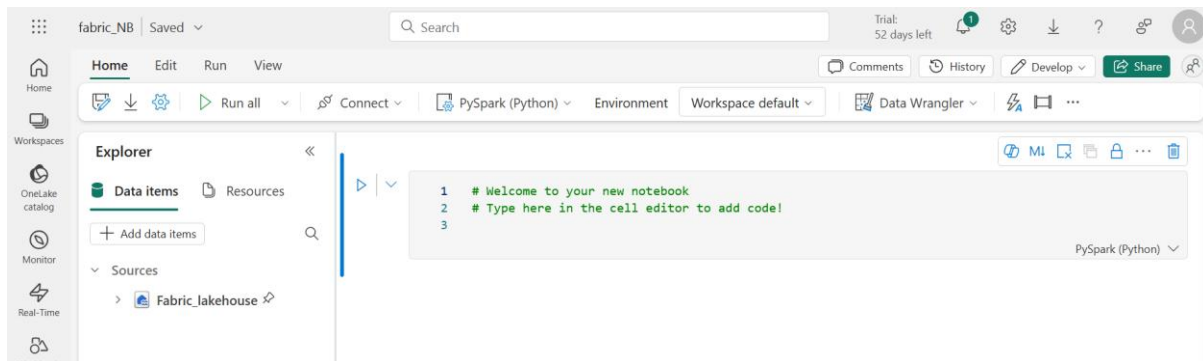
Go to existing data source as we already have lake house in our workspace.

OneLake catalog

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

[All](#)
[My data](#)
[Endorsed in your org](#)
[Favorites](#)

	Name	Owner	Refreshed	Location	Endorsement	Sensitivity
<input checked="" type="checkbox"/>	Fabric_lakehouse	deefabric	—	fabricworkspace	—	—
<input type="checkbox"/>	second_lh	deefabric	—	newws	—	—
<input type="checkbox"/>	DataflowsStagingLakehouse	deefabric	—	fabricworkspace	—	—

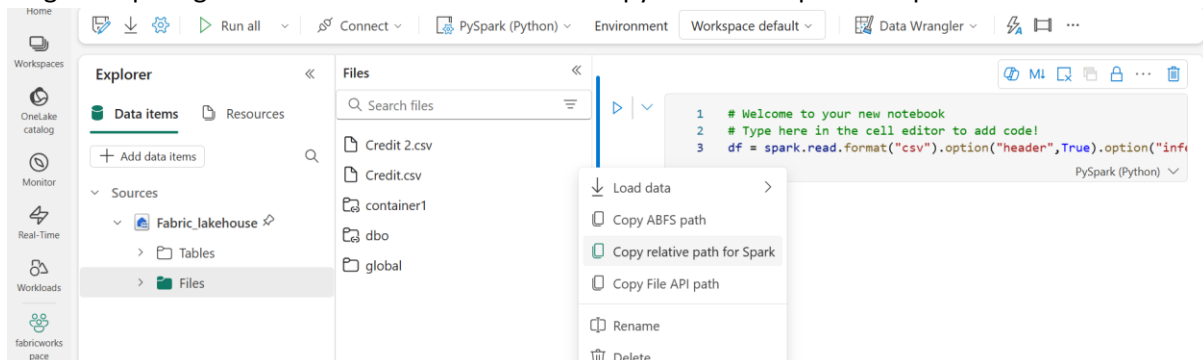


Source added.

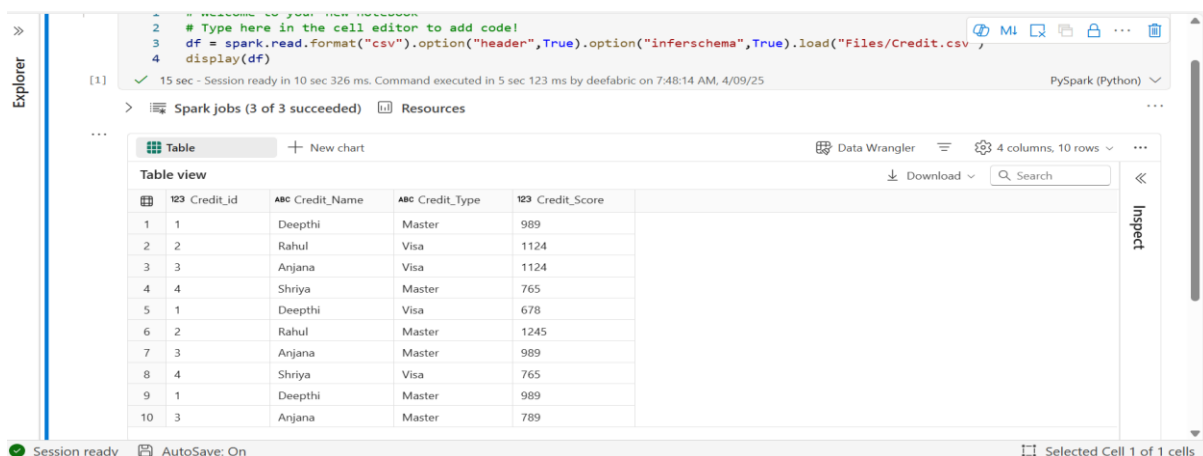
Now we have to read data from lake house to a data frame, using below code.

**Df = spark.read.format("File  
Format").options("Header",True).options("inferschema",True).load("Path where to load")**

To get file path go to file -> click on three dots -> Copy relative file path for spark.



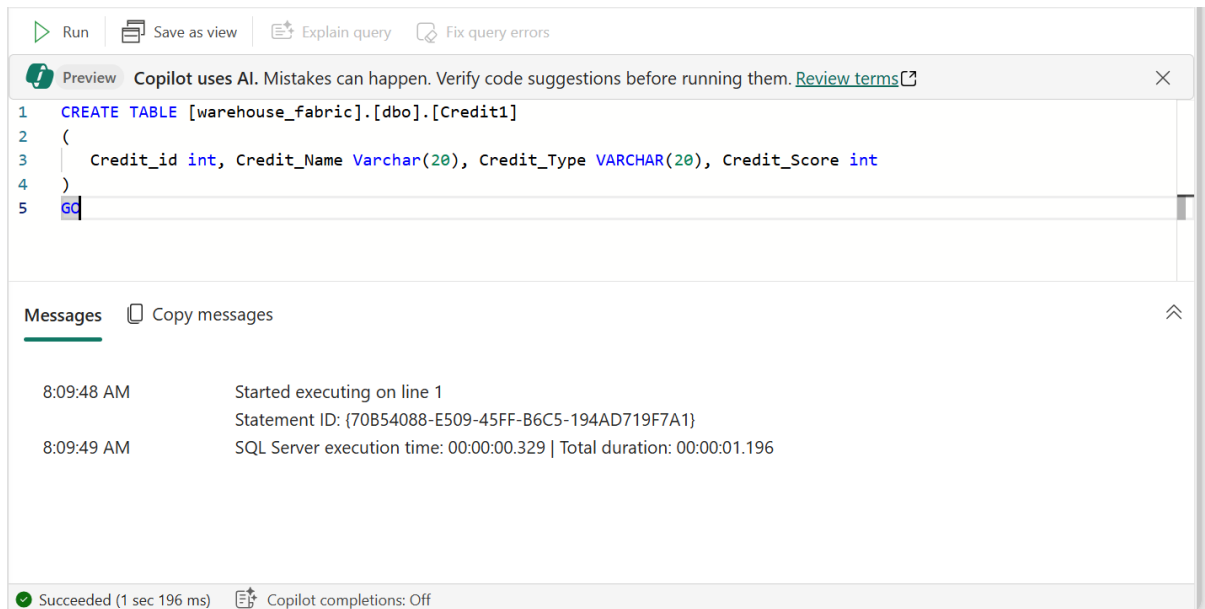
Add display(df) code and run this cell.



Code ran successfully.

Now write this data to datawarehouse.

As we are loading data to warehouse we need to first create a table with same schema of lake house file.



The screenshot shows a SQL query editor with a toolbar at the top containing 'Run', 'Save as view', 'Explain query', and 'Fix query errors'. A notification banner at the top states: 'Copilot uses AI. Mistakes can happen. Verify code suggestions before running them. [Review terms](#)'. The query text is:

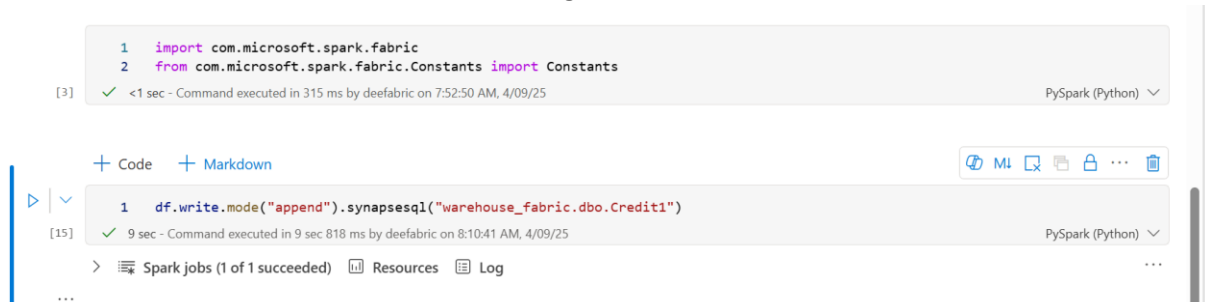
```
1 CREATE TABLE [warehouse_fabric].[dbo].[Credit1]
2 (
3     Credit_id int, Credit_Name Varchar(20), Credit_Type VARCHAR(20), Credit_Score int
4 )
5 go
```

Below the query editor is a 'Messages' panel with a 'Copy messages' button. It contains two messages:

- 8:09:48 AM: Started executing on line 1  
Statement ID: {70B54088-E509-45FF-B6C5-194AD719F7A1}
- 8:09:49 AM: SQL Server execution time: 00:00:00.329 | Total duration: 00:00:01.196

At the bottom, a status bar indicates 'Succeeded (1 sec 196 ms)' and 'Copilot completions: Off'.

Now we can write the data to warehouse, using below code



The screenshot shows a PySpark code editor with a toolbar at the top containing '+ Code', '+ Markdown', and icons for 'ML', 'Q', 'P', 'A', '...', and 'B'. The code text is:

```
1 import com.microsoft.spark.fabric
2 from com.microsoft.spark.fabric.Constants import Constants
```

Below the code editor is a message: '[3] ✓ <1 sec - Command executed in 315 ms by deefabric on 7:52:50 AM, 4/09/25'. The editor is set to 'PySpark (Python)'.

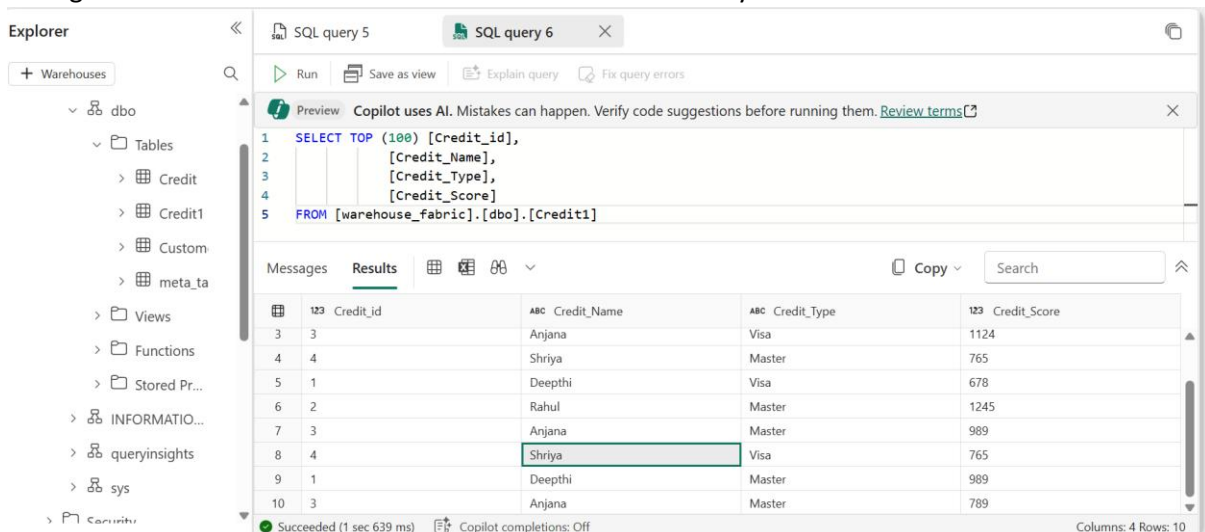
Below this is another code block:

```
1 df.write.mode("append").synapsesql("warehouse_fabric.dbo.Credit1")
```

Below the code is a message: '[15] ✓ 9 sec - Command executed in 9 sec 818 ms by deefabric on 8:10:41 AM, 4/09/25'. The editor is set to 'PySpark (Python)'.

Below the code blocks is a section for 'Spark jobs (1 of 1 succeeded)', 'Resources', and 'Log'.

Now go to warehouse and check if the data is loaded correctly or not.



The screenshot shows a SQL query editor with a toolbar at the top containing 'Run', 'Save as view', 'Explain query', and 'Fix query errors'. A notification banner at the top states: 'Copilot uses AI. Mistakes can happen. Verify code suggestions before running them. [Review terms](#)'. The query text is:

```
1 SELECT TOP (100) [Credit_id],
2     [Credit_Name],
3     [Credit_Type],
4     [Credit_Score]
5 FROM [warehouse_fabric].[dbo].[Credit1]
```

Below the query editor is a 'Messages' panel with a 'Copy' button and a search bar. The 'Results' tab is active, showing a table with 4 columns: 'Credit\_id', 'Credit\_Name', 'Credit\_Type', and 'Credit\_Score'. The table has 10 rows of data.

Credit_id	Credit_Name	Credit_Type	Credit_Score
3	Anjana	Visa	1124
4	Shriya	Master	765
5	Deepthi	Visa	678
6	Rahul	Master	1245
7	Anjana	Master	989
8	Shriya	Visa	765
9	Deepthi	Master	989
10	Anjana	Master	789

At the bottom, a status bar indicates 'Succeeded (1 sec 639 ms)' and 'Copilot completions: Off'. The bottom right corner shows 'Columns: 4 Rows: 10'.

We see that data is loaded correctly to warehouse from Lake house.

## Read data from warehouse and load data to lake house

Now we have to read a table from warehouse, using below code.

Here I am reading table **warehouse\_fabric.dbo.Customer**

Read data from warehouse and load data to lakehouse

```
1 df_warehouse = spark.read.synapsesql("warehouse_fabric.dbo.Customer")
2 display(df_warehouse)
```

[16] ✓ 4 sec - Command executed in 4 sec 750 ms by deefabric on 8:15:44 AM, 4/09/25

Spark jobs (1 of 1 succeeded) Resources Log

Table view

	12L Phoneno	ABC Name	123 ID	ABC City
1	4567824567	Sam	5	Brampton
2	4567824567	leo	4	Sudbury
3	4567824567	Naho	3	Paris
4	4567824567	Robert	2	Toronto
5	4567824567	John	1	New york

Data read successfully.

Now write this data to lakehouse, we need to use below code

```
df_warehouse.write.format("csv").option("header",True).mode("append").save("Files")
```

df\_warehouse is a data frame which contains warehouse data

we are trying to save as csv file and saving this file to location file.

```
1 df_warehouse.write.format("csv").option("header",True).mode("append").save("Files")
```

[18] ✓ 7 sec - Command executed in 7 sec 882 ms by deefabric on 8:21:18 AM, 4/09/25

Spark jobs (1 of 1 succeeded) Resources Log

Code ran successfully.

Now check in lakehouse if file loaded correctly or not.

Fabric\_lakehouse

Home

Get data New semantic model Open notebook Manage OneLake data access (preview)

Workspaces

OneLake catalog

fabricsworks pace

warehouse\_fabric

Fabric\_lakehouse

fabrics\_NB

Fabric

Explorer

Search tables

Fabric\_lakehouse

Tables

credit\_table

customer\_table

Merge\_Op

Files

container1

dbo

global

Files

Name	Date modified	Type	Size
Credit 2.csv	4/6/2025, 12:0...	csv	122 B
Credit.csv	4/2/2025, 10:2...	csv	252 B
_SUCCESS	4/9/2025, 8:21:...		0 B
container1	4/7/2025, 10:1...	Folder	-
dbo	4/7/2025, 9:51:...	Folder	-
global	4/7/2025, 9:26:...	Folder	-
part-00000-d80750b6-0b62-4645-90e9-f85b4fc2e...	4/9/2025, 8:21:...	csv	151 B

New file is loaded to lakehouse

Files > part-00000-d80750b6-0b62-4645-90e9-f85b4fc2e6ff-c000.csv (preview)

```
1 Phoneno,Name,ID,City
2 4567824567,Sam,5,Brampton
3 4567824567,Naho,3,Paris
4 4567824567,leo,4,Sudbury
5 4567824567,Robert,2,Toronto
6 4567824567,John,1,New york
7
```





File data.

We can also specify the filename in path.

```
1 df_warehouse.write.format("csv").option("header",True).mode("append").save("Files/new_data.csv")
```

✓ 6 sec - Command executed in 6 sec 171 ms by deefabric on 8:23:23 AM, 4/09/25

PySpark (Python)

	container1	4/7/2025, 10:1...	Folder	-
	dbo	4/7/2025, 9:51:...	Folder	-
	global	4/7/2025, 9:26:...	Folder	-
	new_data.csv	4/9/2025, 8:23:...	Folder	-

It is created in lake house.



