Bootcamp Project

Project 5 - Migrating pipelines from ADF to Synapse

Madhumitha Vijayakumar

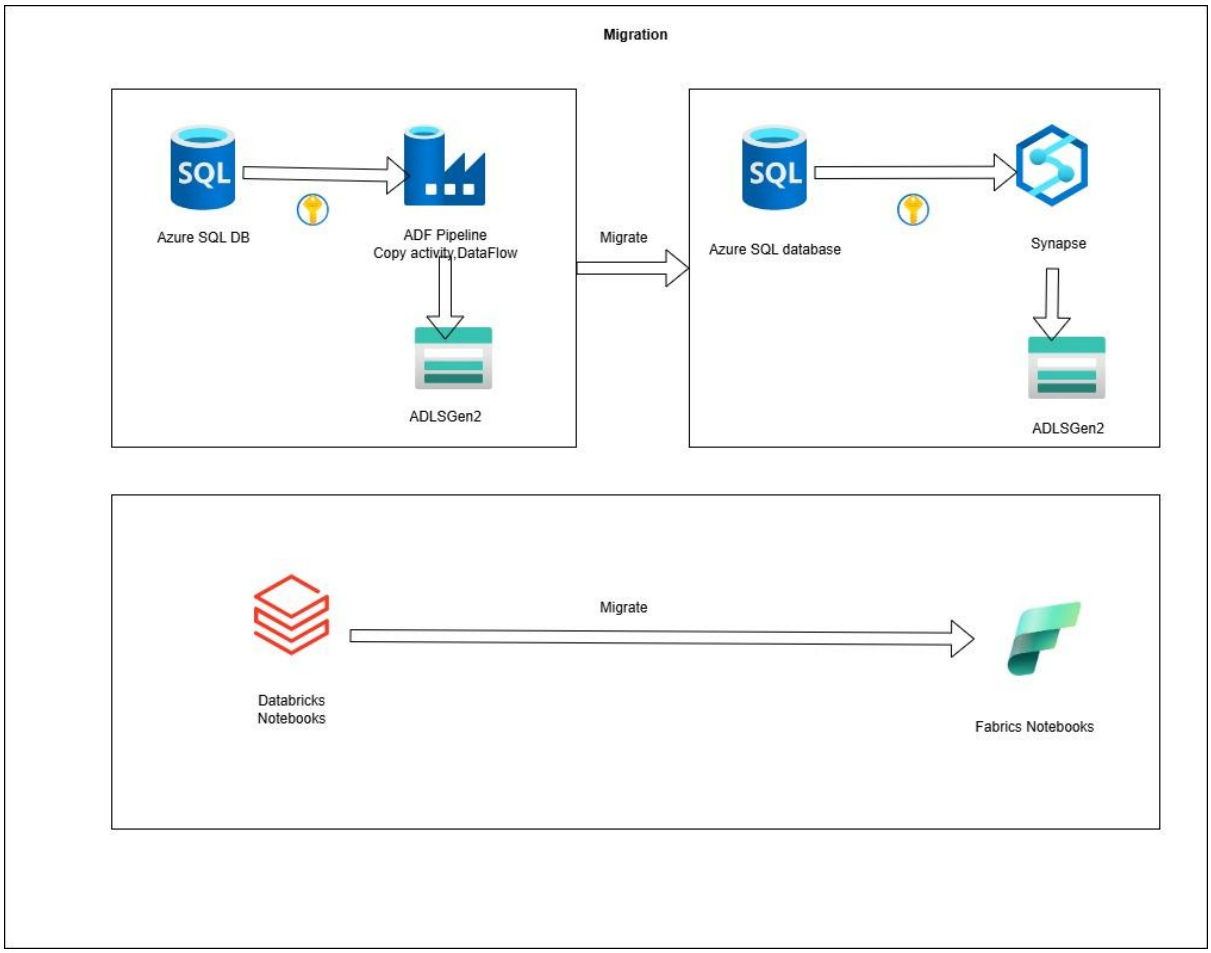
11-05-2025

Project Overview

Develop ADF pipelines (copy activity, foreach loop, look up) and migrate the pipelines, datasets, linked services to Azure Synapse.

Github link –https://github.com/DeepthiChethi/Azure_DE/tree/Madhumita/Project5.

Architecture



Tools and Technologies

- Azure Data Factory
- Azure Synapse
- Azure SQL database
- ADLSGen2
- Azure Key vault
- Databricks
- Fabrics
- Draw.io

Task -1: Migrating pipelines from ADF to Synapse

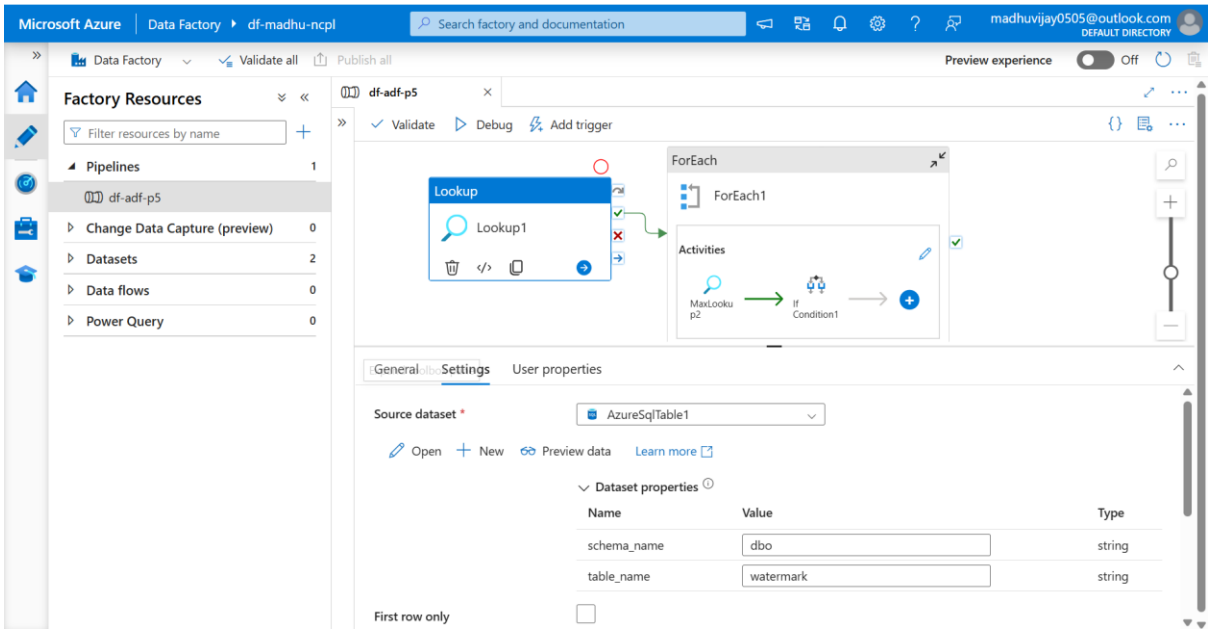
Create a watermark table in SSMS

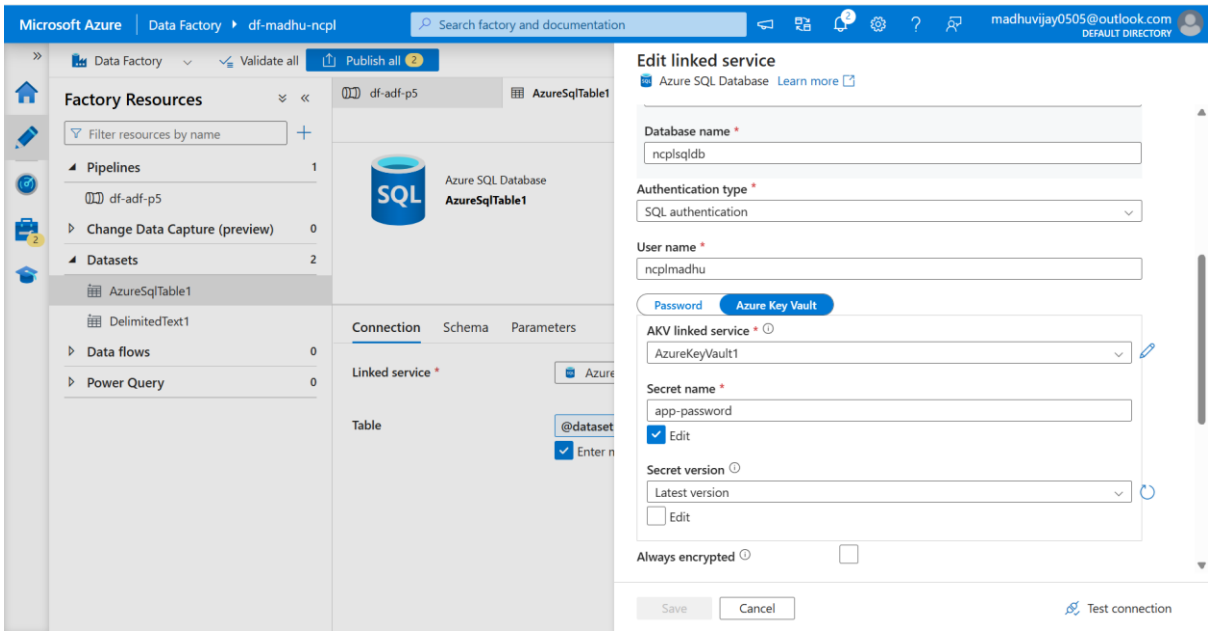
Insert the values in the watermark table

Create stored procedure for updating values in lpv column

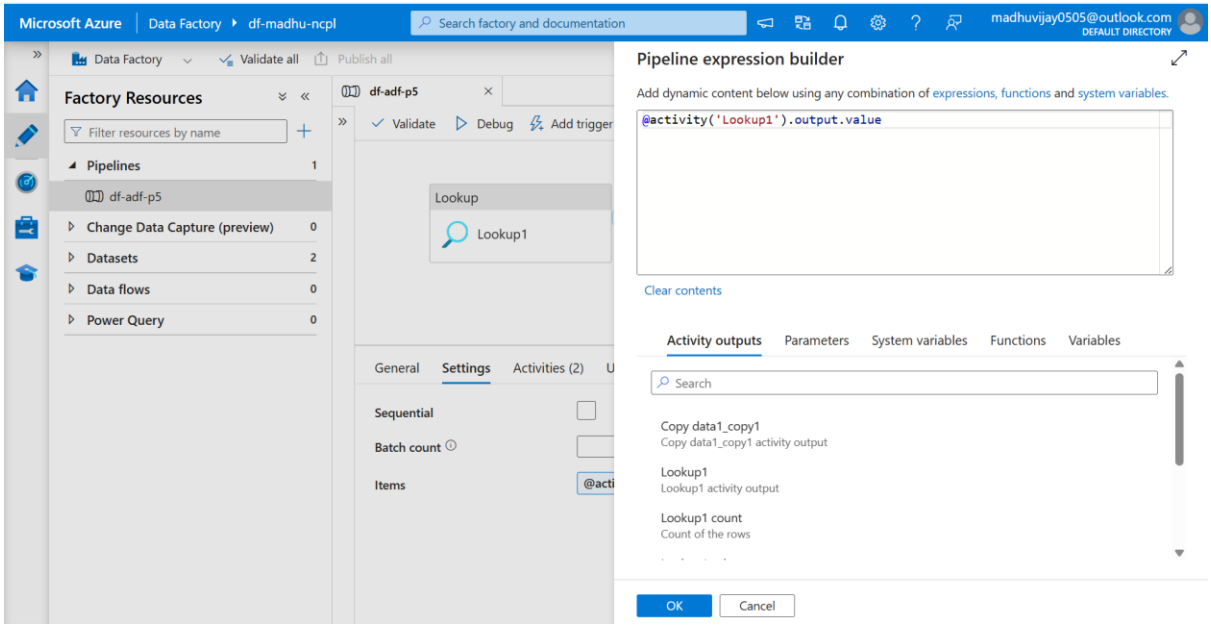
Navigate to ADF workspace

Create linked services for SQL database using key vault and connect lookup activity to foreach loop. Create parameters schema_name and table_name in lookup activity.

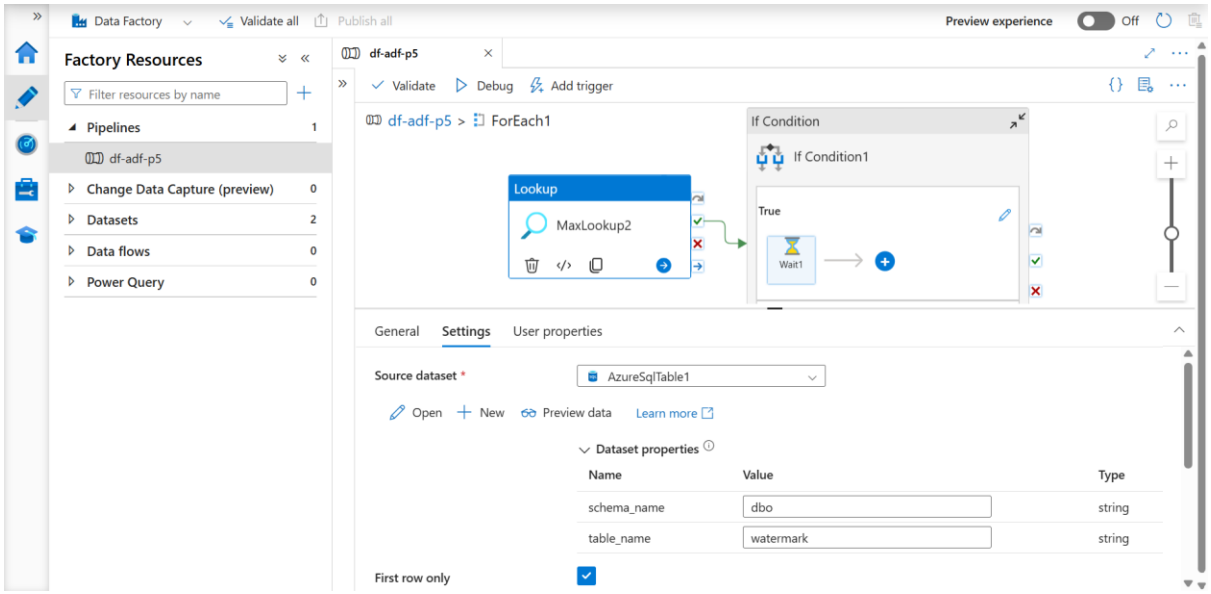




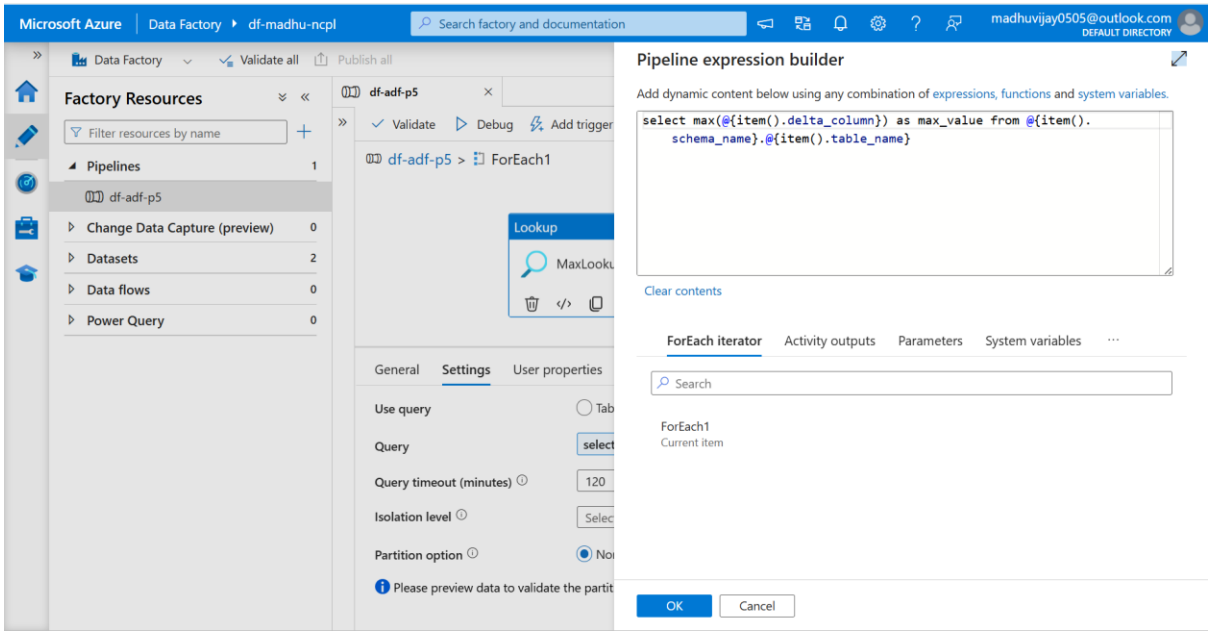
In foreach loop activity, click lookup value array for looping look up activity.



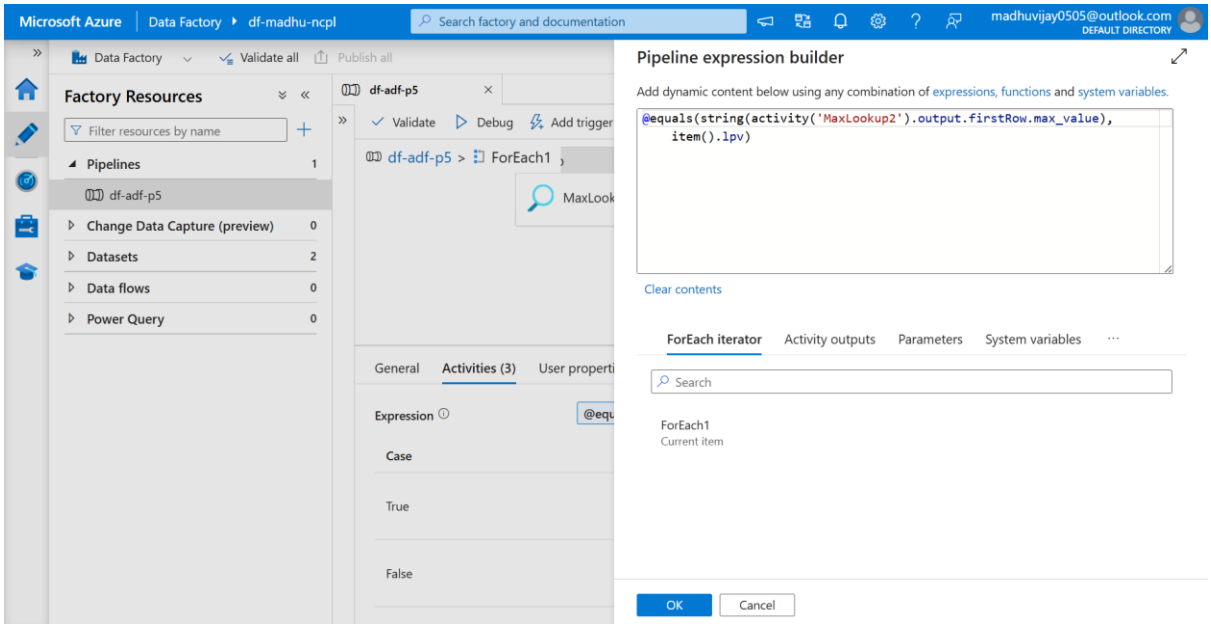
Drag and drop look up activity, enable first row only.



Use the query to select the maximum value of delta column.

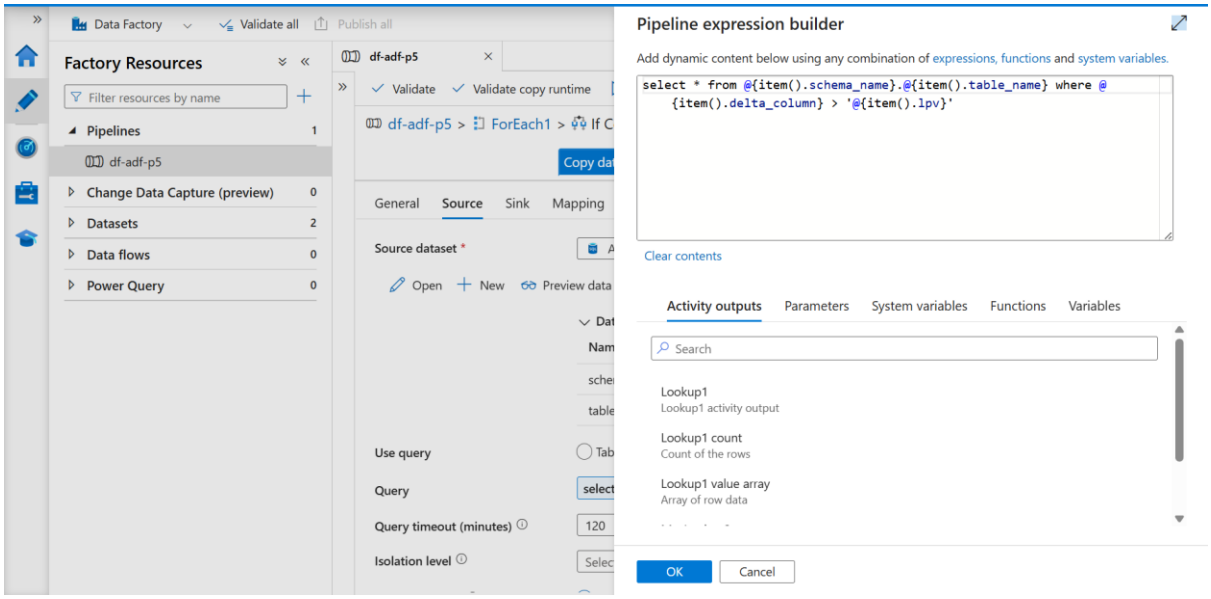


Drag and drop the if condition activity, if condition is true, wait activity will be executed and if condition is false, copy activity and Stored Procedure will be executed.



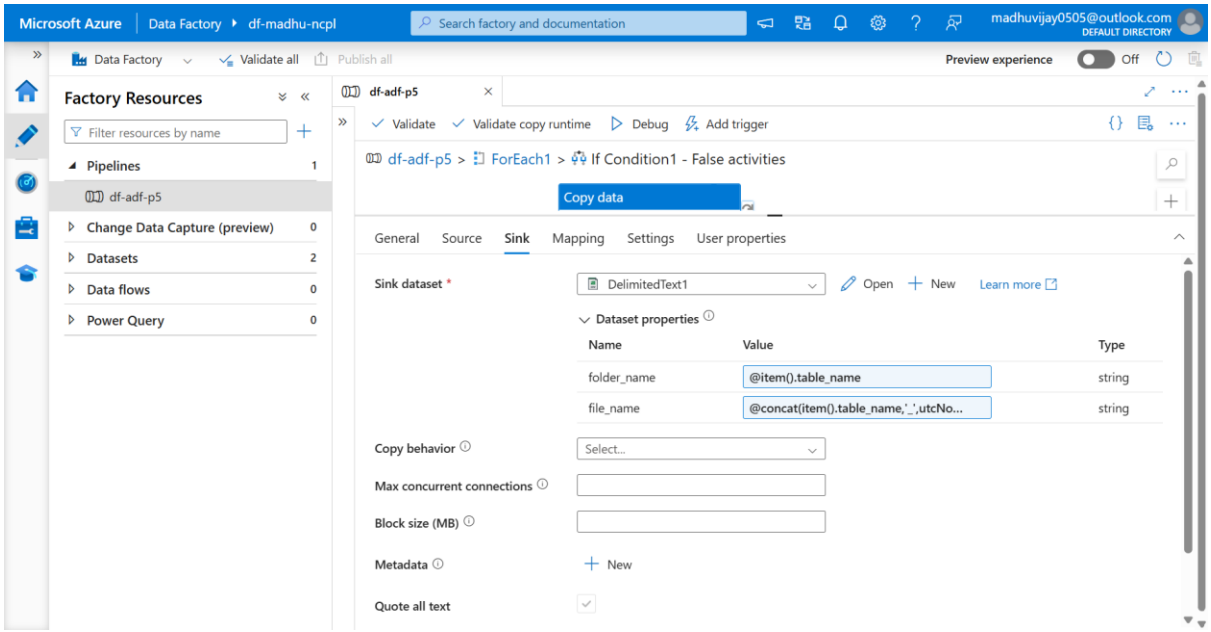
Copy activity-Source

Retrieve the records where the value in delta_column greater the value in LPV.

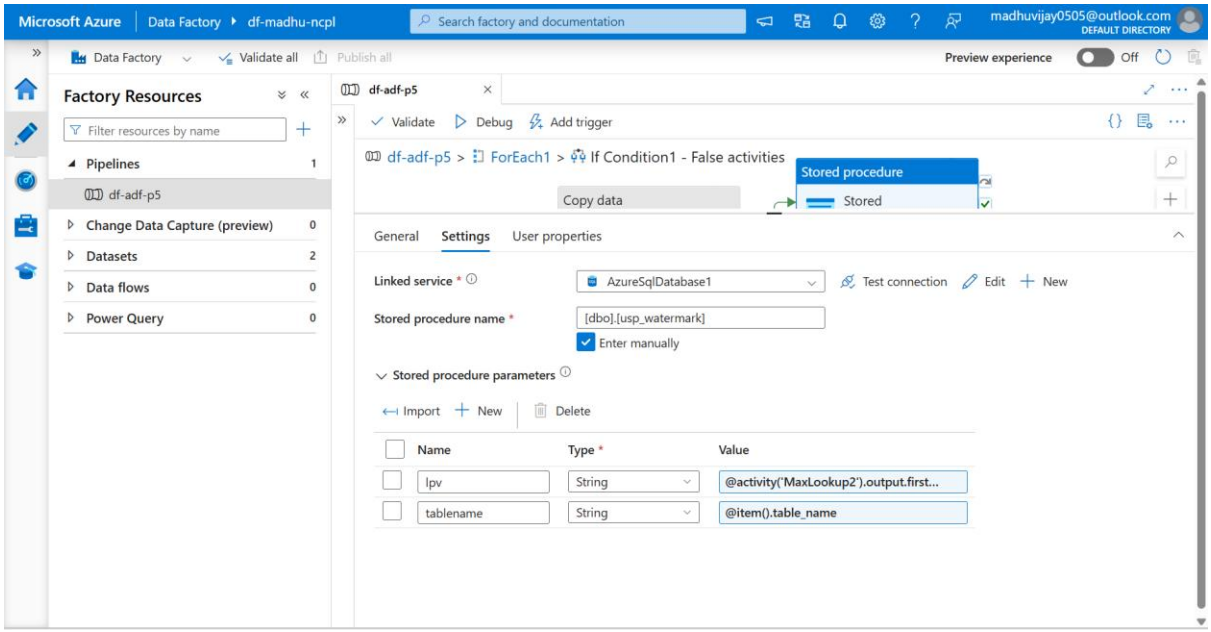


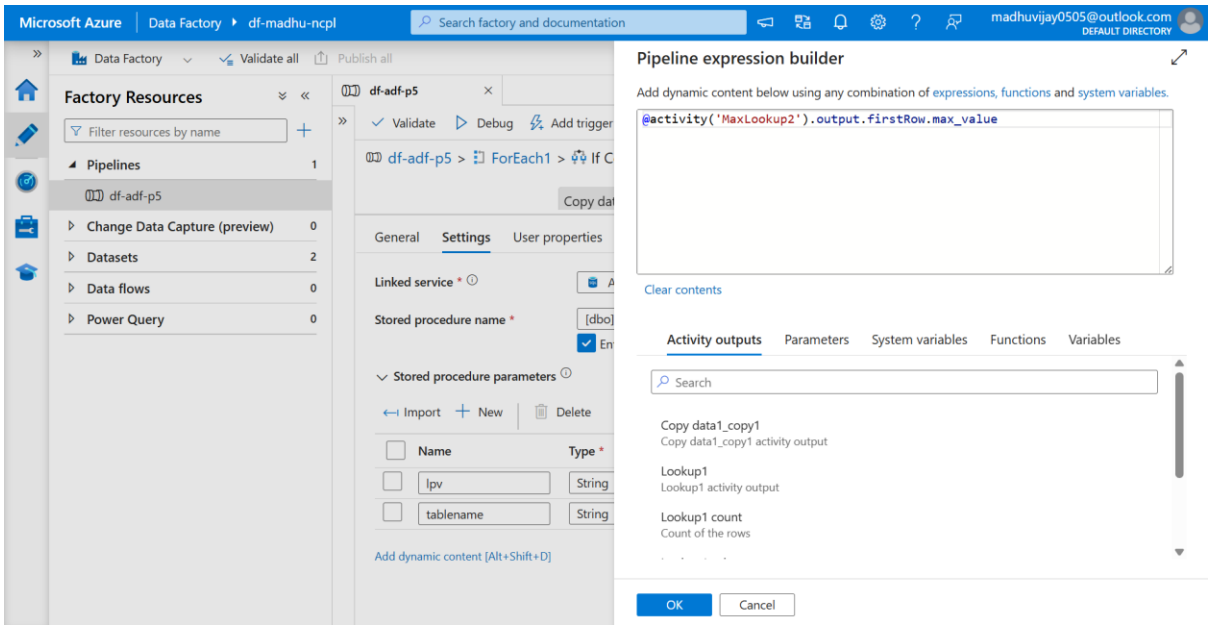
Sink

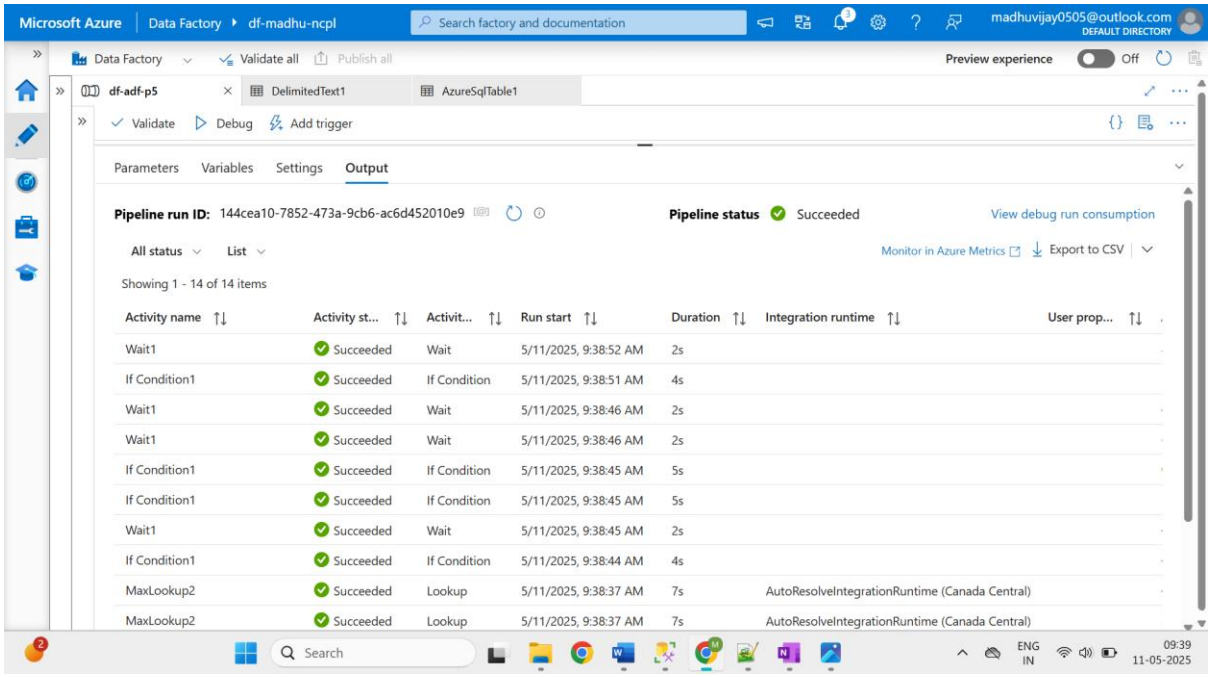
Select delimited text as sink dataset, create folder_name and file_name parameters and add @item().table_name and @concat(item().table_name,'_',utcNow(),'.csv') in the following parameters



Drag and drop stored procedure and on success connect copy activity with stored procedure

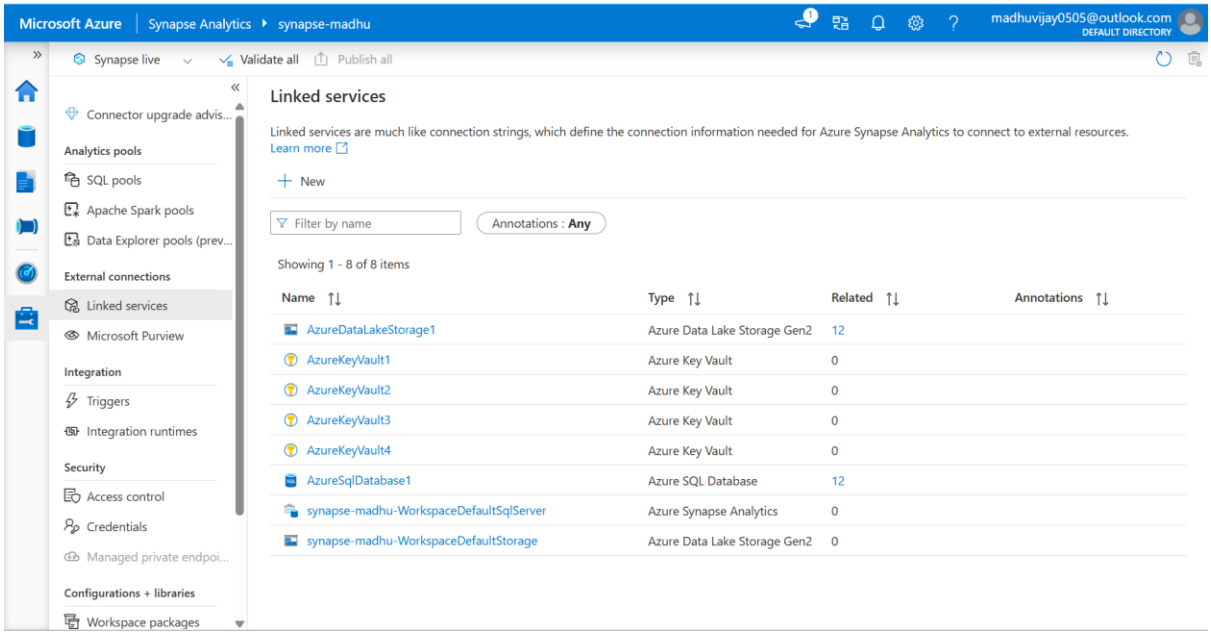

Microsoft Azure
Data Factory
df-madhu-ncpl
Search factory and documentation
madhuvijay0505@outlook.com
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 1
df-adf-p5
Change Data Capture (preview) 0
Datasets 2
Data flows 0
Power Query 0
df-adf-p5
Validate
Debug
Add trigger
df-adf-p5 > ForEach1 > If Condition1 - False activities
Stored procedure
Copy data
Stored
General
Settings
User properties
Linked service *
AzureSqlDatabase1
Test connection
Edit
New
Stored procedure name *
[dbo].[usp_watermark]
Enter manually
Stored procedure parameters
Import
New
Delete
Name
Type *
Value
lpv
String
@activity('MaxLookup2').output.first...
tablename
String
@item().table_name


Microsoft Azure
Data Factory
df-madhu-ncpl
Search factory and documentation
madhuvijay0505@outlook.com
DEFAULT DIRECTORY
Pipeline expression builder
Add dynamic content below using any combination of expressions, functions and system variables.
@activity('MaxLookup2').output.firstRow.max_value
Clear contents
Activity outputs
Parameters
System variables
Functions
Variables
Search
Copy data1_copy1
Copy data1_copy1 activity output
Lookup1
Lookup1 activity output
Lookup1 count
Count of the rows
OK
Cancel
Add dynamic content [Alt+Shift+D]



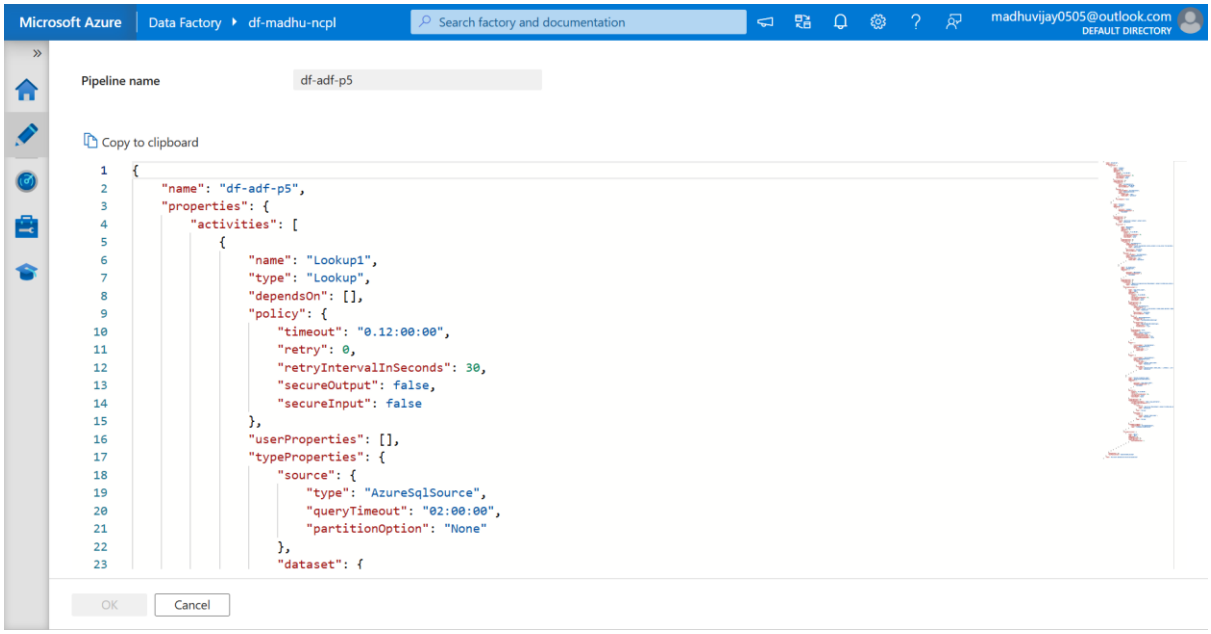
Synapse pipeline

At first, create linked service and datasets in synapse workspace.

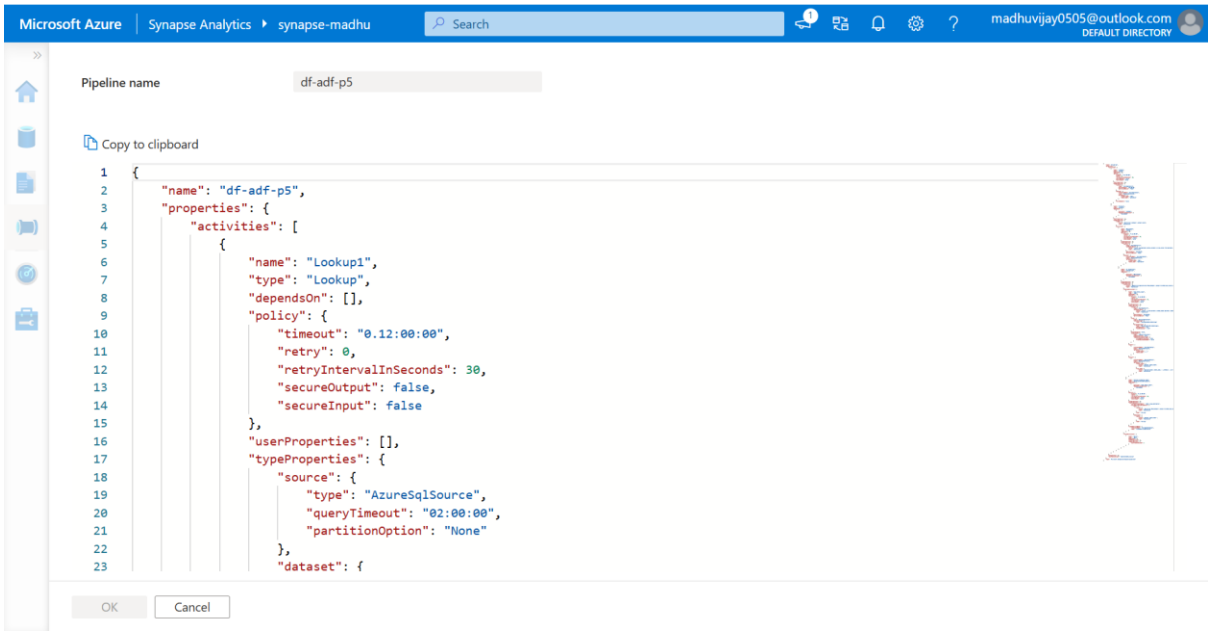


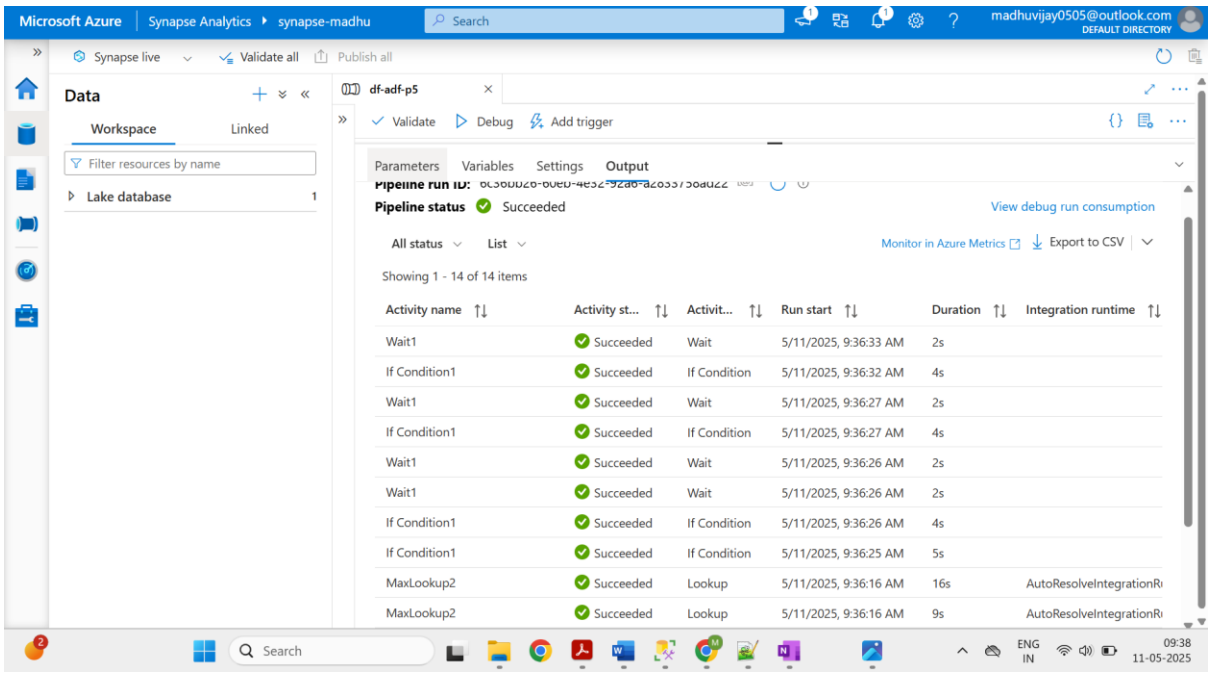
Create a pipeline in synapse workspace with same as ADF pipeline

Copy JSON code from ADF to synapse



Paste it in synapse workspace



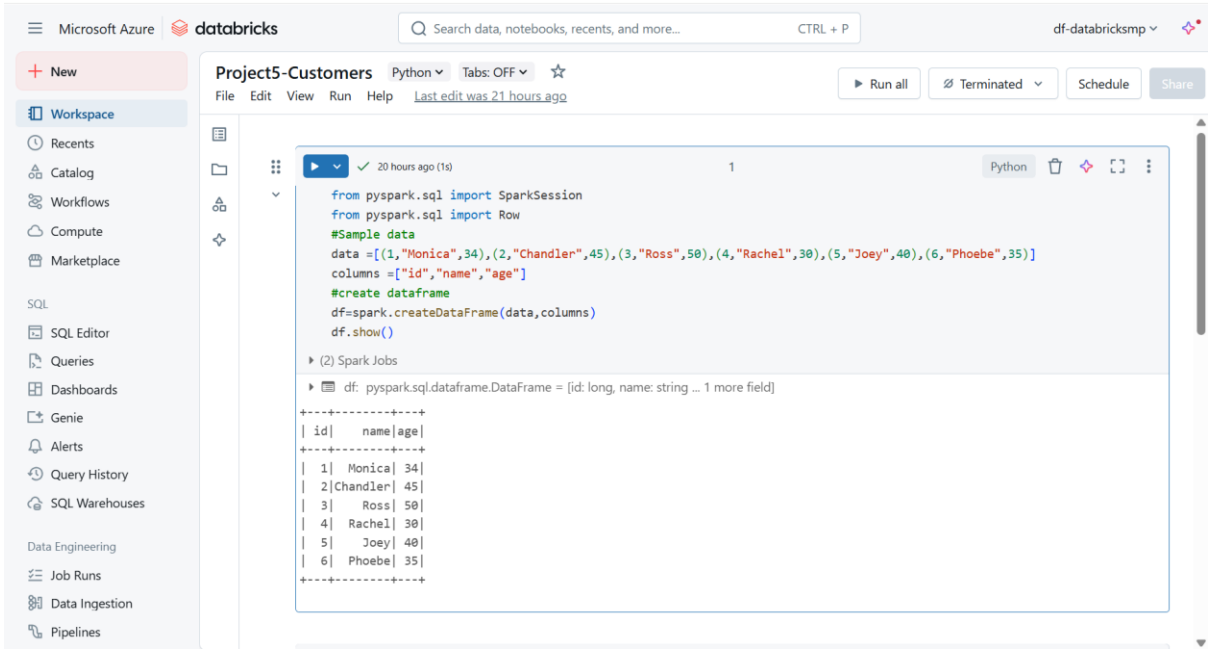


Run the pipeline and it successfully executed.

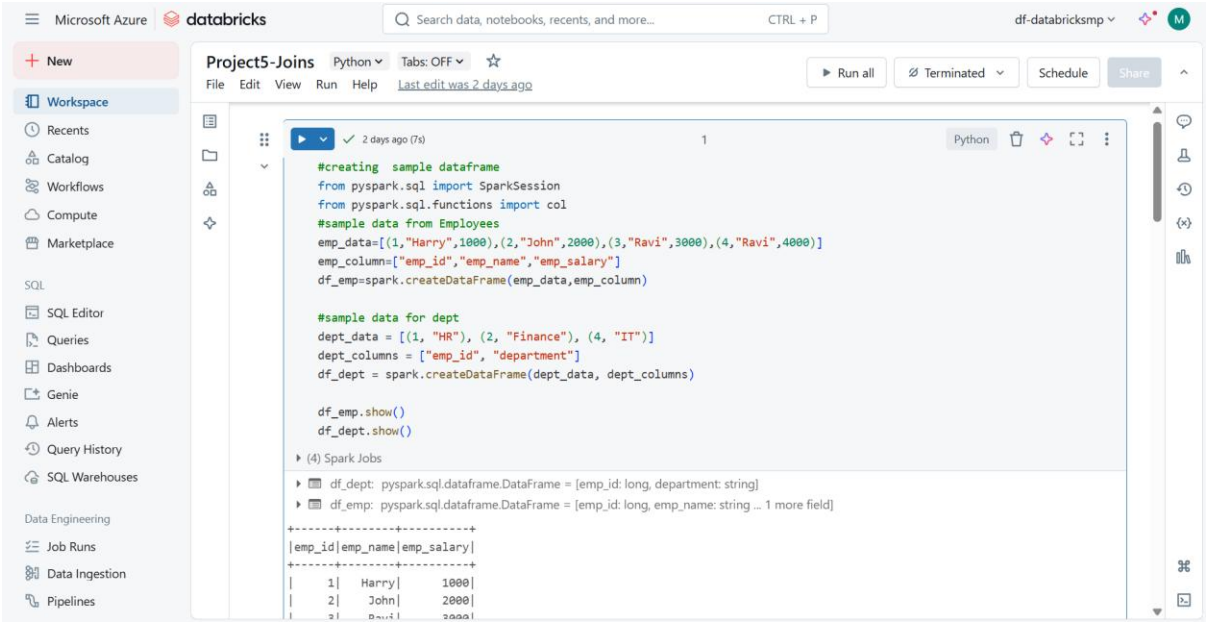
Task 2

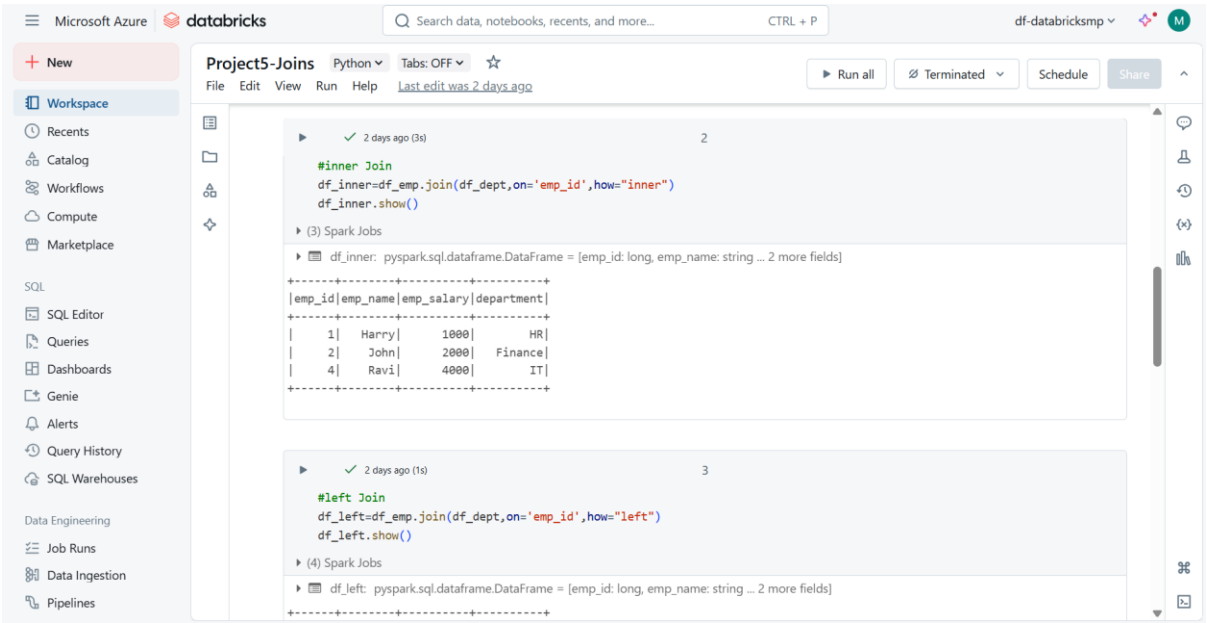
Migrate databricks notebook to fabric notebook.

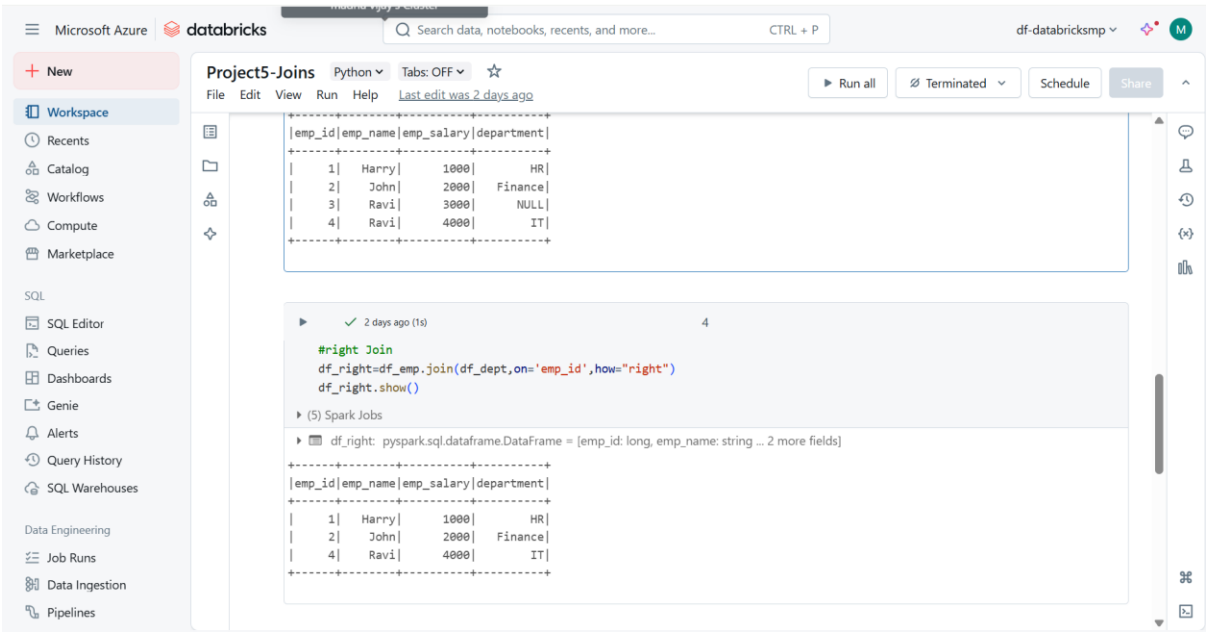
Notebook 1 -Customers

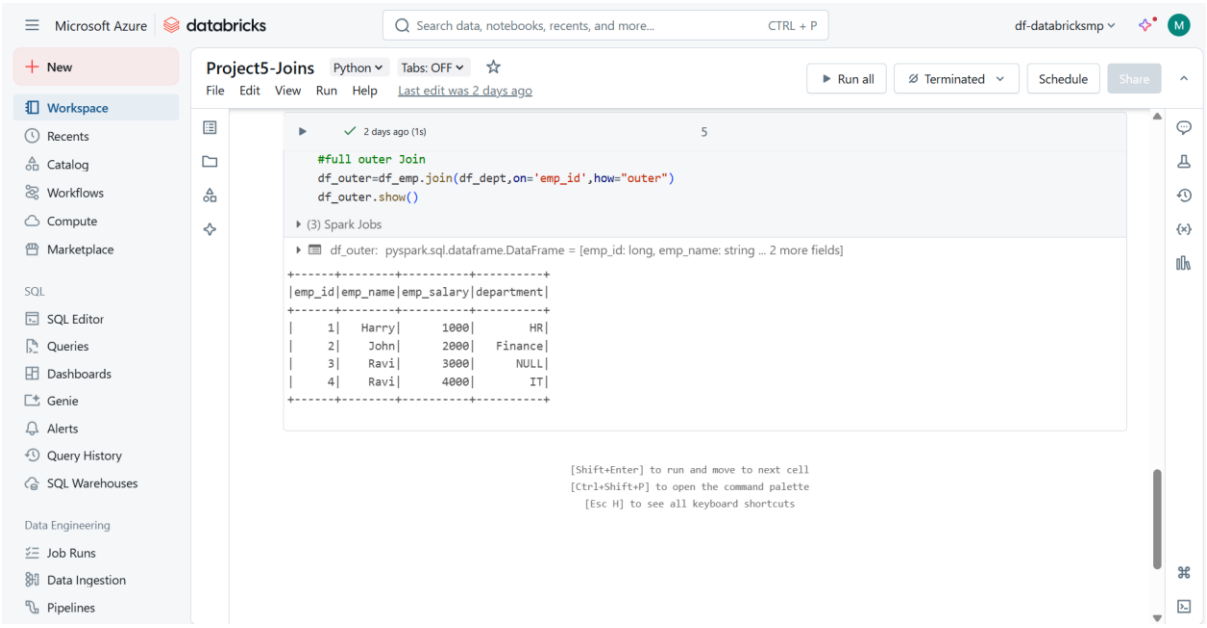


Notebook 2- Joins

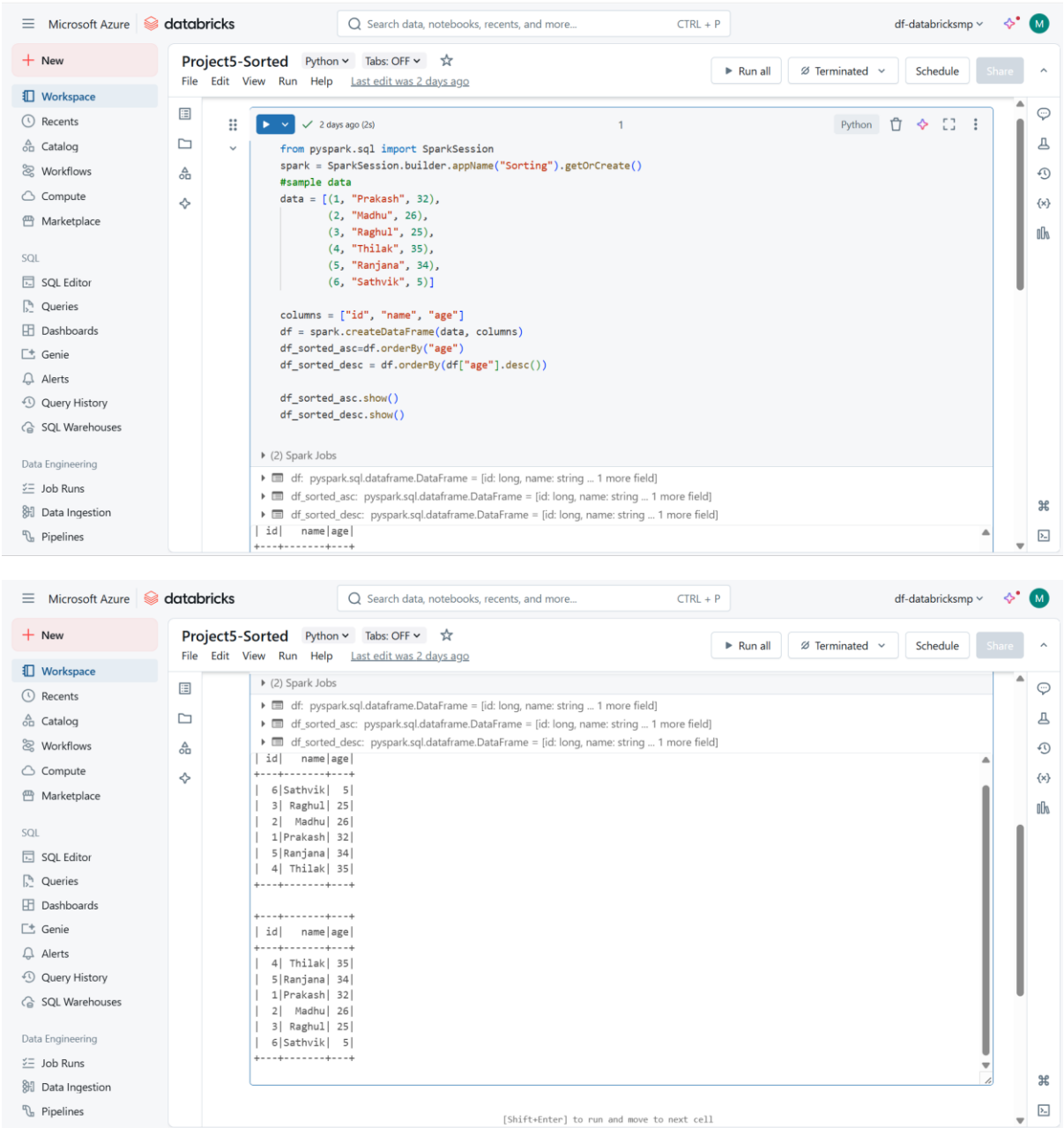

Microsoft Azure
databricks
Search data, notebooks, recents, and more... CTRL + P
df-databricksmp
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Project5-Joins
Python
Tabs: OFF
File Edit View Run Help
Last edit was 2 days ago
Run all
Terminated
Schedule
Share
2 days ago (7s)
1
Python
#creating sample dataframe
from pyspark.sql import SparkSession
from pyspark.sql.functions import col
#sample data from Employees
emp_data=[(1,"Harry",1000),(2,"John",2000),(3,"Ravi",3000),(4,"Ravi",4000)]
emp_column=["emp_id","emp_name","emp_salary"]
df_emp=spark.createDataFrame(emp_data,emp_column)
#sample data for dept
dept_data = [(1, "HR"), (2, "Finance"), (4, "IT")]
dept_columns = ["emp_id", "department"]
df_dept = spark.createDataFrame(dept_data, dept_columns)
df_emp.show()
df_dept.show()
(4) Spark Jobs
df_dept: pyspark.sql.dataframe.DataFrame = [emp_id: long, department: string]
df_emp: pyspark.sql.dataframe.DataFrame = [emp_id: long, emp_name: string ... 1 more field]
+------+--------+----------+
|emp_id|emp_name|emp_salary|
+------+--------+----------+
| 1| Harry| 1000|
| 2| John| 2000|


Microsoft Azure
databricks
Search data, notebooks, recents, and more... CTRL + P
df-databricksmp
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Project5-Joins
Python
Tabs: OFF
File Edit View Run Help
Last edit was 2 days ago
Run all
Terminated
Schedule
Share
2 days ago (3s)
2
#inner Join
df_inner=df_emp.join(df_dept,on='emp_id',how="inner")
df_inner.show()
(3) Spark Jobs
df_inner: pyspark.sql.dataframe.DataFrame = [emp_id: long, emp_name: string ... 2 more fields]
+------+--------+----------+----------+
|emp_id|emp_name|emp_salary|department|
+------+--------+----------+----------+
1	Harry	1000	HR
2	John	2000	Finance
4	Ravi	4000	IT
+------+--------+----------+----------+
2 days ago (1s)
3
#left Join
df_left=df_emp.join(df_dept,on='emp_id',how="left")
df_left.show()
(4) Spark Jobs
df_left: pyspark.sql.dataframe.DataFrame = [emp_id: long, emp_name: string ... 2 more fields]

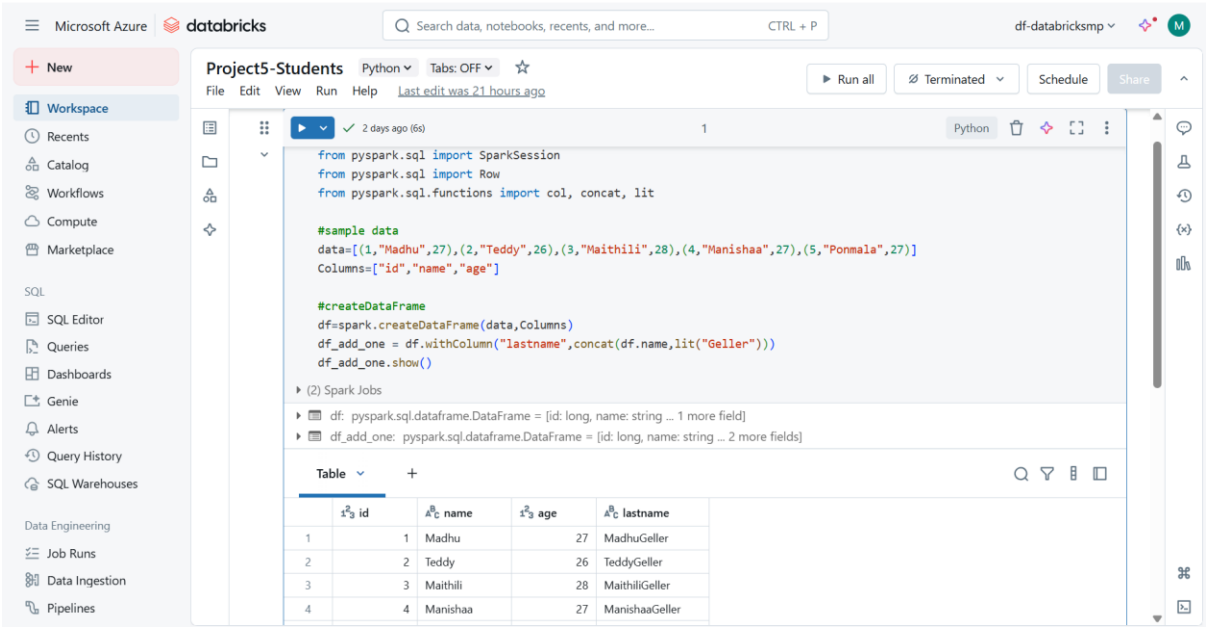


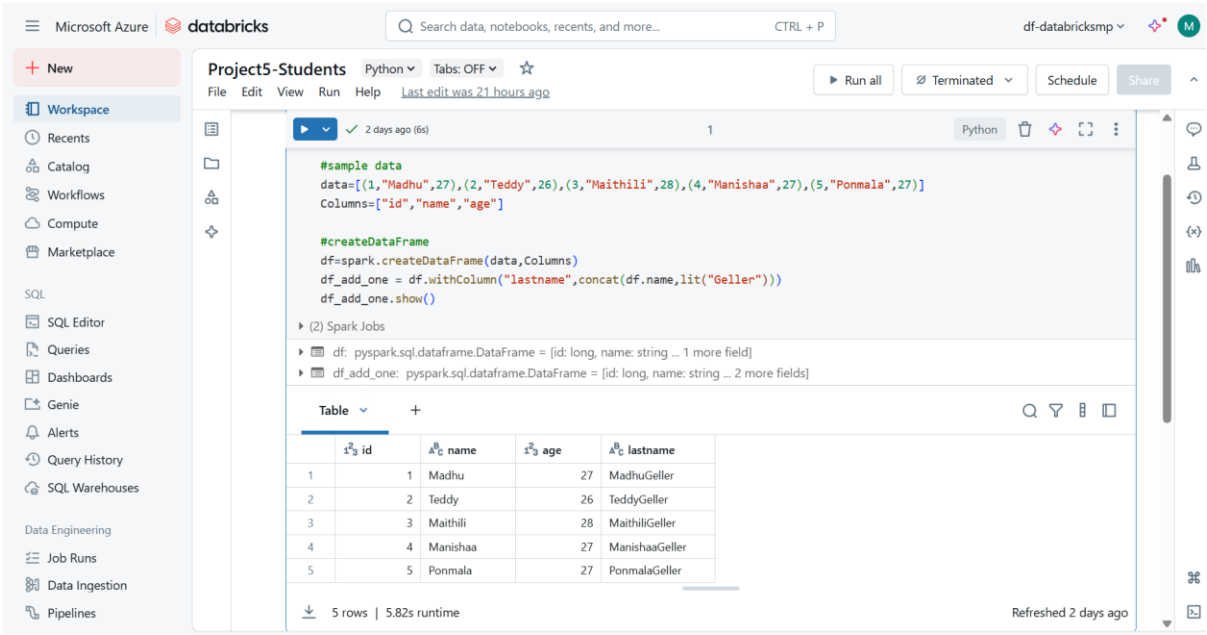


Notebook 3-Sorted



Notebook 4-Students

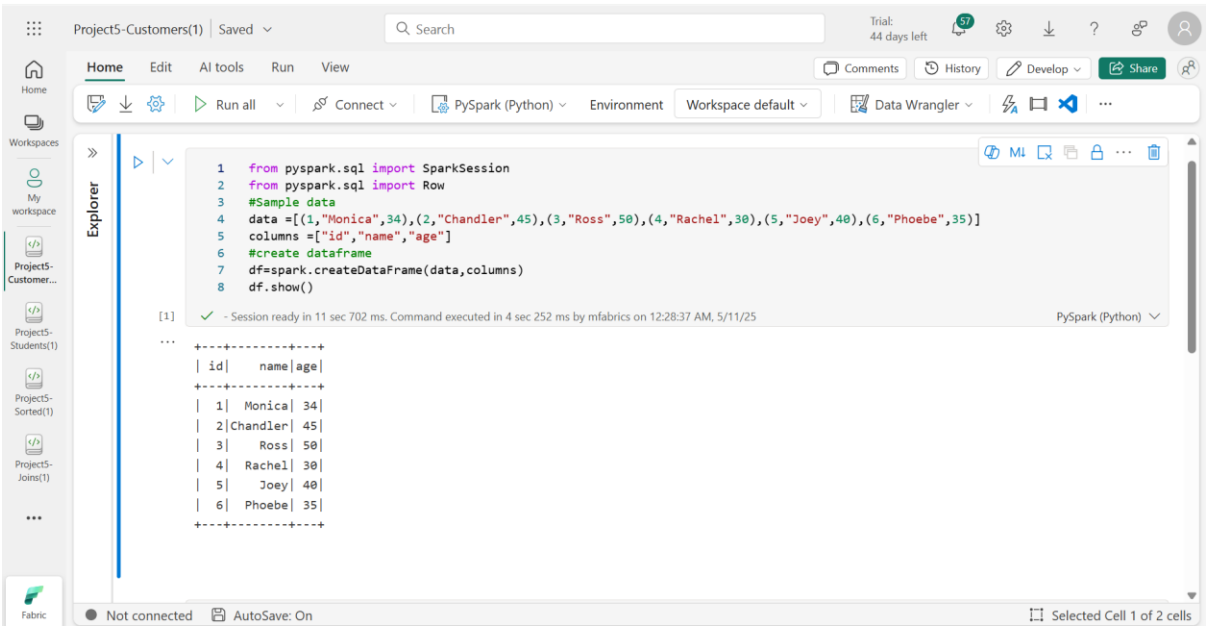


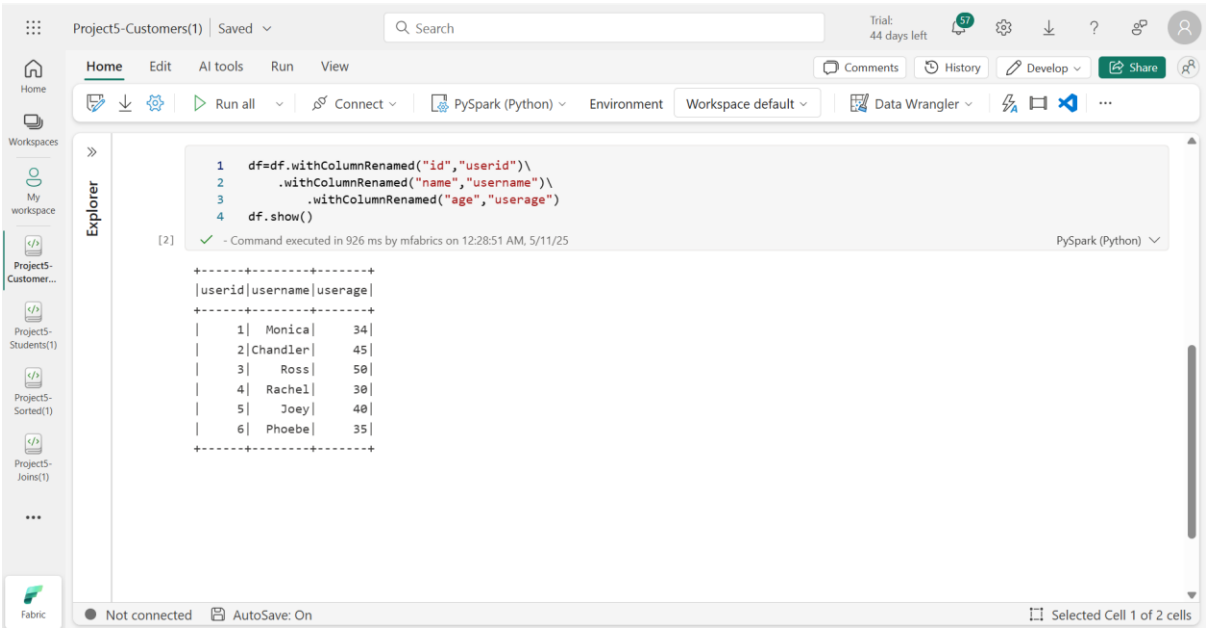


Export all the 4 notebooks from databricks in ipython notebook

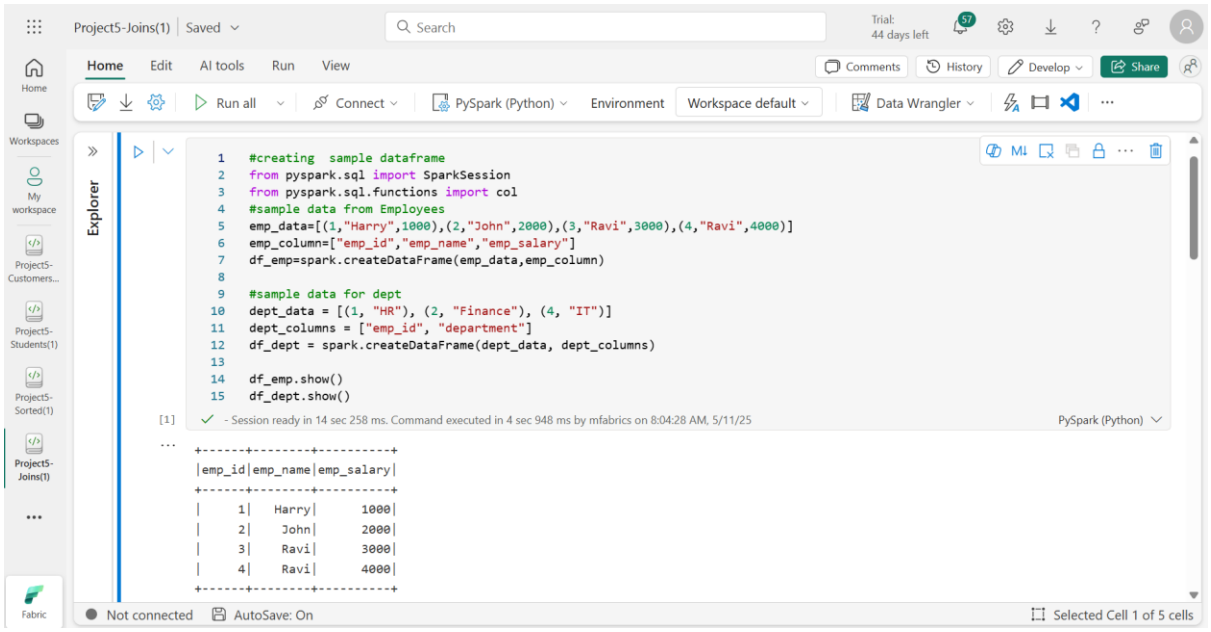
Import the all the ipython notebooks in fabrics. Go to workspace->import-> import notebook.

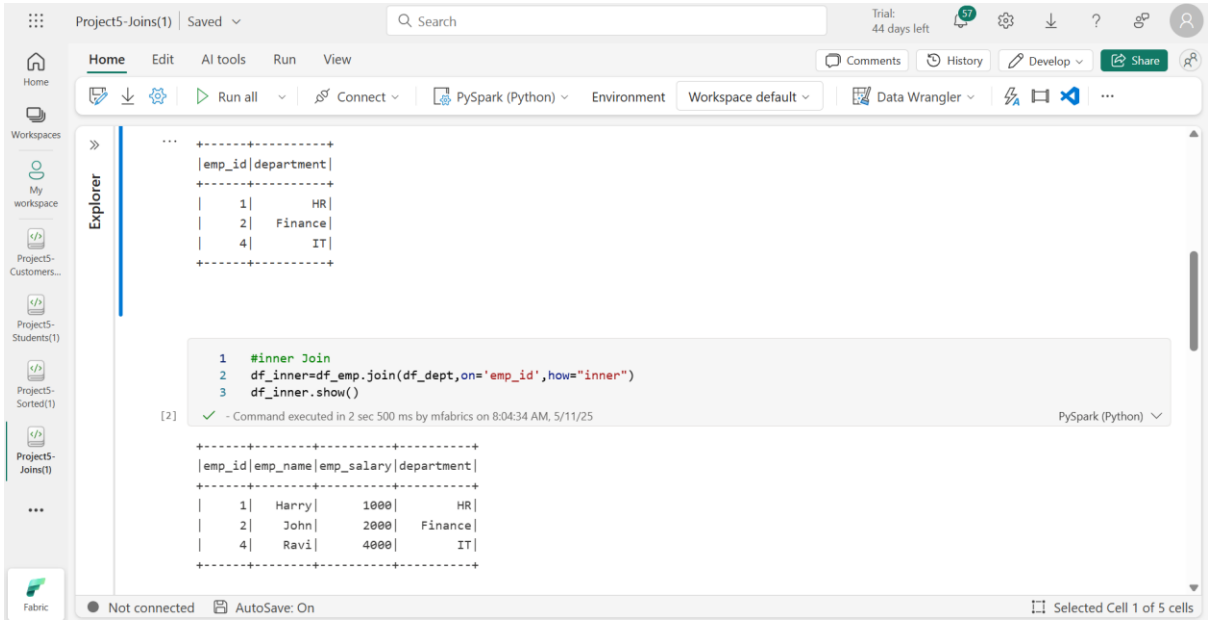
Notebook 1- Customers



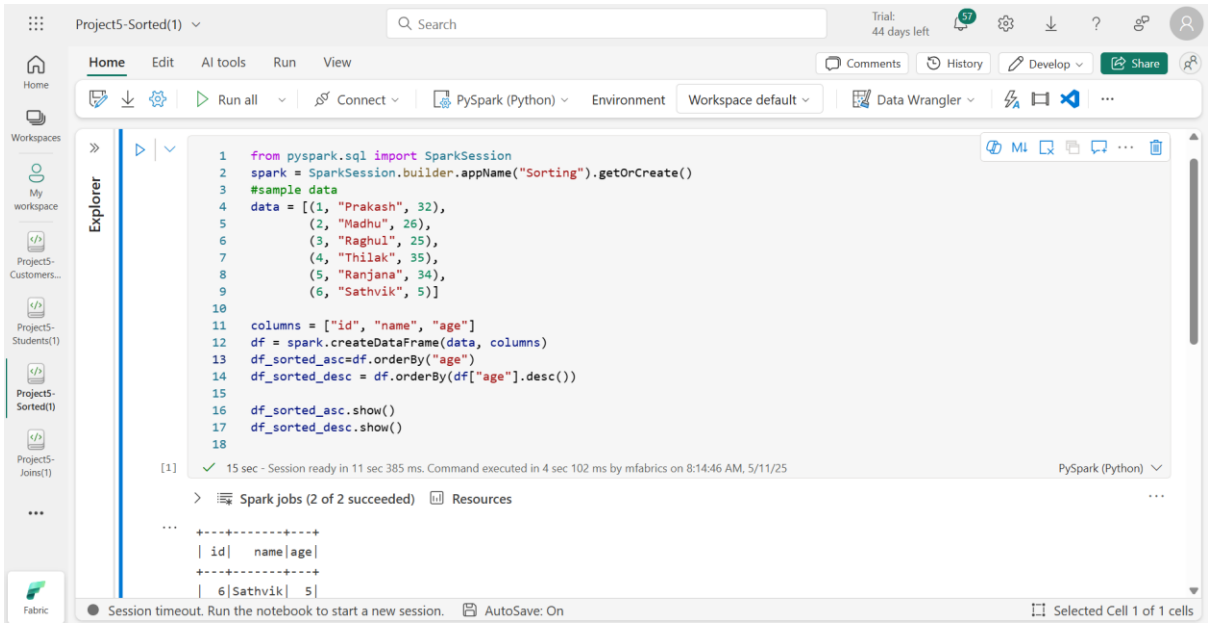


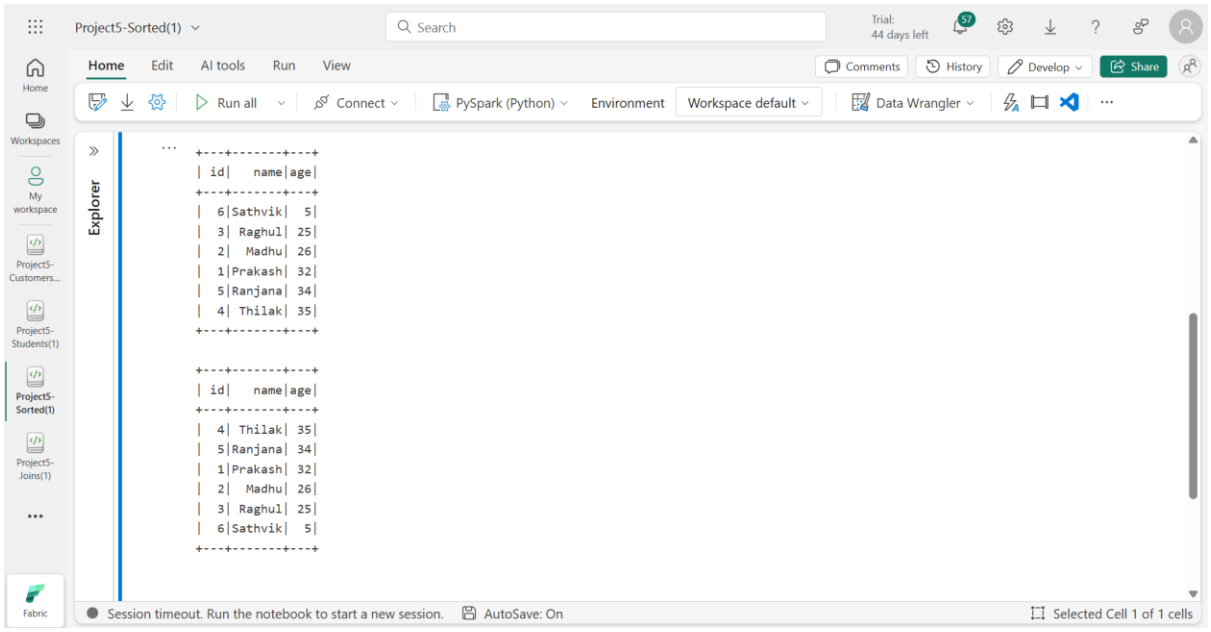
Notebook 2- joins



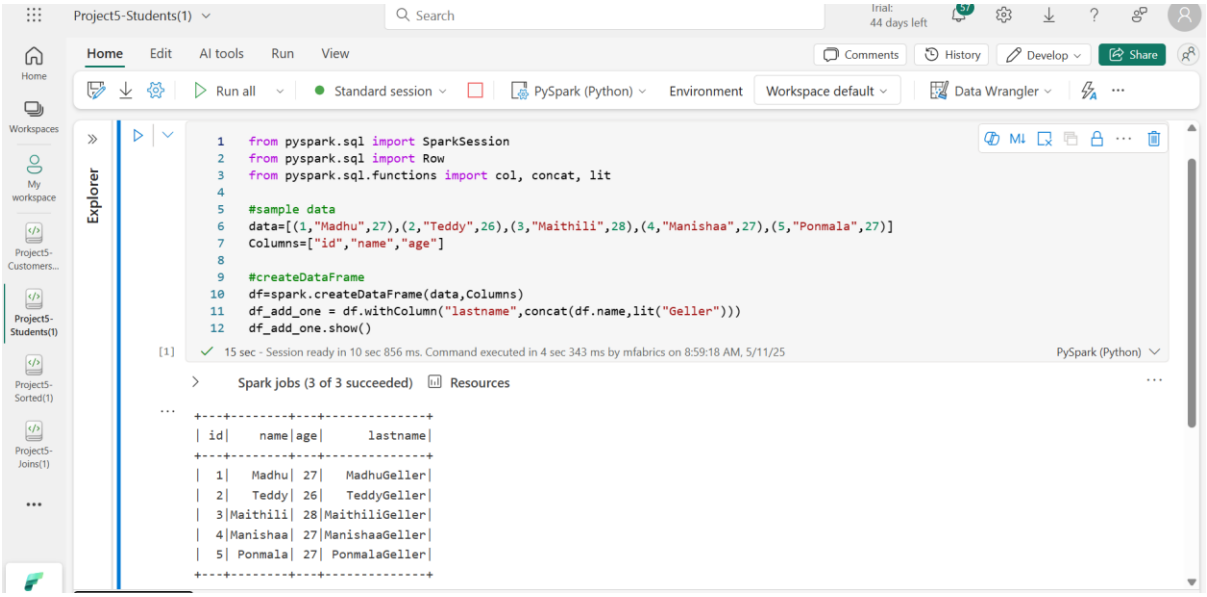


Notebook3- Sorted





Notebook 4-Students



All notebooks ran successfully