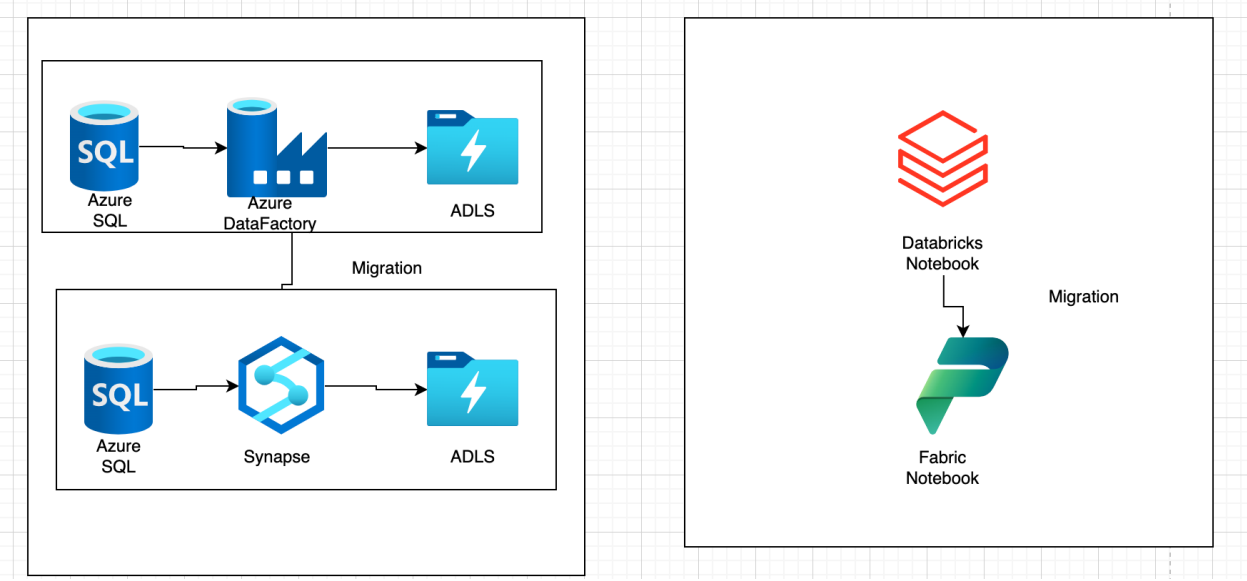
Bootcamp Project 5

Project Title: Migrating pipelines from ADF to Synapse

Problem Statement:

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

Architecture diagram:



Tools & Technologies:

- Azure Data factory
- Azure Synapse
- Azure SQL
- Azure data lake Gen2
- Databricks
- Fabric

Create a pipeline to bring multiple tables from Azure SQL to ADLS gen2.

Creating pipeline in ADF

Run Cancel Disconnect Change | Database: shubha | Estimated Plan Enable Actual Plan Parse Enabl

To Notebook

```
1   CREATE TABLE PEOPLE (
2       ID INT,
3       PERSON_NAME VARCHAR(50),
4       PERSON_CITY VARCHAR(20),
5       PERSON_SAL INT,
6       PERSON_PHNO BIGINT,
7       FAVORITE_COLOR VARCHAR(20) NULL  -- Allows NULL values
8   );
9   INSERT INTO PEOPLE (ID, PERSON_NAME, PERSON_CITY, PERSON_SAL, PERSON_PHNO, FAVORITE_COLOR)
10  VALUES
11  (1, 'SHUBHA', 'NEWYORK', 20000, 58585855858, NULL),
12  (2, 'RAMA', 'DENMARK', 80000, 98585855858, NULL),
13  (3, 'SHAMITHA', 'BALI', 70000, 18585855858, NULL),
14  (4, 'ALICE', 'LONDON', 65000, 75858585858, 'Green'),
15  (5, 'BOB', 'TORONTO', 72000, 65858585858, 'Yellow'),
16  (6, 'CHARLIE', 'SYDNEY', 80000, 55858585858, 'Orange');
17  select * from PEOPLE
18
```

Results Messages

	ID	schema_name	Table_name	LPV	Delta_col
1	1	dbo	PEOPLE	0	ID
2	2	dbo	EMPLOYEE	0	EMP_ID
3	3	dbo	CUST	2025-03-01 10:00:00	UPDATED_DATE
4	4	dbo	COST	2025-03-01 10:00:00	UPDATED_DATE
5	5	dbo	ORDERS	2025-03-01 14:00:00	ORDER_DATE

1.

```
19  CREATE TABLE EMPLOYEE
20  (
21  EMP_ID INT, EMP_NAME VARCHAR(50), EMP_CITY VARCHAR(20),
22  EMP_SAL INT,EMP_PHNO BIGINT
23  )
24
25  INSERT INTO EMPLOYEE VALUES ( 1, 'SHUBHA', 'NEWYORK', 20000, 58585855858)
26  INSERT INTO EMPLOYEE VALUES ( 2, 'RAMA', 'DENMARK', 80000, 98585855858)
27   INSERT INTO EMPLOYEE VALUES ( 3, 'SHAMITHA', 'BALI', 70000, 18585855858)
28  select * from EMPLOYEE
29
```

esults Messages

	EMP_ID	EMP_NAME	EMP_CITY	EMP_SAL	EMP_PHNO
	1	SHUBHA	NEWYORK	20000	58585855858
	2	RAMA	DENMARK	80000	98585855858
	3	SHAMITHA	BALI	70000	18585855858

Results grid

```sql
CREATE table CUST(
CID INT, CNAME VARCHAR(100), CCITY VARCHAR(100), UPDATED_DATE DATETIME)


INSERT INTO CUST (CID, CNAME, CCITY, UPDATED_DATE)
VALUES (1, 'John Doe', 'New York', '2025-03-01 10:30:00');

INSERT INTO CUST (CID, CNAME, CCITY, UPDATED_DATE)
VALUES (2, 'Jane Smith', 'Los Angeles', '2025-03-01 11:00:00');

INSERT INTO CUST (CID, CNAME, CCITY, UPDATED_DATE)
VALUES (3, 'Robert Brown', 'Chicago', '2025-03-01 12:00:00');
select * from CUST

CREATE TABLE COST (
```

esults Messages

CID	CNAME	CCITY	UPDATED_DATE
1	John Doe	New York	2025-03-01 10:30:00.000
2	Jane Smith	Los Angeles	2025-03-01 11:00:00.000
3	Robert Brown	Chicago	2025-03-01 12:00:00.000

```sql
CREATE TABLE COST (
    COST_ID INT PRIMARY KEY,
    COST_NAME VARCHAR(100) NOT NULL,
    COST_AMOUNT DECIMAL(10,2),
    UPDATED_DATE DATETIME
);
INSERT INTO COST (COST_ID, COST_NAME, COST_AMOUNT, UPDATED_DATE)
VALUES (1, 'Labor Cost', 5000.00, '2025-03-01 10:00:00');

INSERT INTO COST (COST_ID, COST_NAME, COST_AMOUNT, UPDATED_DATE)
VALUES (2, 'Material Cost', 12000.50, '2025-03-01 11:00:00');

INSERT INTO COST (COST_ID, COST_NAME, COST_AMOUNT, UPDATED_DATE)
VALUES (3, 'Transportation Cost', 3000.75, '2025-03-01 12:00:00');
```

Results Messages

	COST_ID	COST_NAME	COST_AMOUNT	UPDATED_DATE
	1	Labor Cost	5000.00	2025-03-01 10:00:00.000
	2	Material Cost	12000.50	2025-03-01 11:00:00.000
	3	Transportation Cost	3000.75	2025-03-01 12:00:00.000

```sql
CREATE TABLE ORDERS (
    ORDER_ID INT,
    CUSTOMER_ID INT,
    ORDER_AMOUNT DECIMAL(10,2),
    ORDER_DATE DATETIME
);
INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, ORDER_AMOUNT, ORDER_DATE)
VALUES (1, 1, 150.00, '2025-03-01 14:00:00');

INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, ORDER_AMOUNT, ORDER_DATE)
VALUES (2, 2, 250.75, '2025-03-01 15:00:00');

INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, ORDER_AMOUNT, ORDER_DATE)
VALUES (3, 3, 99.99, '2025-03-01 16:00:00');
```

Results Messages

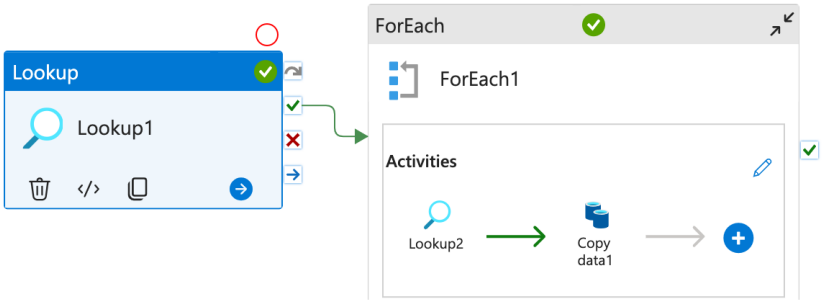
	ORDER_ID	CUSTOMER_ID	ORDER_AMOUNT	ORDER_DATE
1	1	1	150.00	2025-03-01 14:00:00.000
2	2	2	250.75	2025-03-01 15:00:00.000
3	3	3	99.99	2025-03-01 16:00:00.000

```sql
CREATE TABLE WatermarkTable (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    schema_name VARCHAR(50),
    Table_name VARCHAR(50),
    LPV VARCHAR(100),
    Delta_col VARCHAR(50)
);

Insert into WatermarkTable values ('dbo', 'PEOPLE' , '0', 'ID')
Insert into WatermarkTable values ('dbo', 'EMPLOYEE', '0', 'EMP_ID')
Insert into WatermarkTable values ('dbo', 'CUST' , '2025-03-01 10:00:00', 'UPDATED_DATE')
Insert into WatermarkTable values ('dbo', 'COST' , '2025-03-01 10:00:00', 'UPDATED_DATE')
Insert into WatermarkTable values ('dbo', 'ORDERS' , '2025-03-01 14:00:00', 'ORDER_DATE')
```

lts Messages

ID	schema_name	Table_name	LPV	Delta_col
1	dbo	PEOPLE	0	ID
2	dbo	EMPLOYEE	0	EMP_ID
3	dbo	CUST	2025-03-01 10:00:00	UPDATED_DATE
4	dbo	COST	2025-03-01 10:00:00	UPDATED_DATE
5	dbo	ORDERS	2025-03-01 14:00:00	ORDER_DATE



General Settings User properties

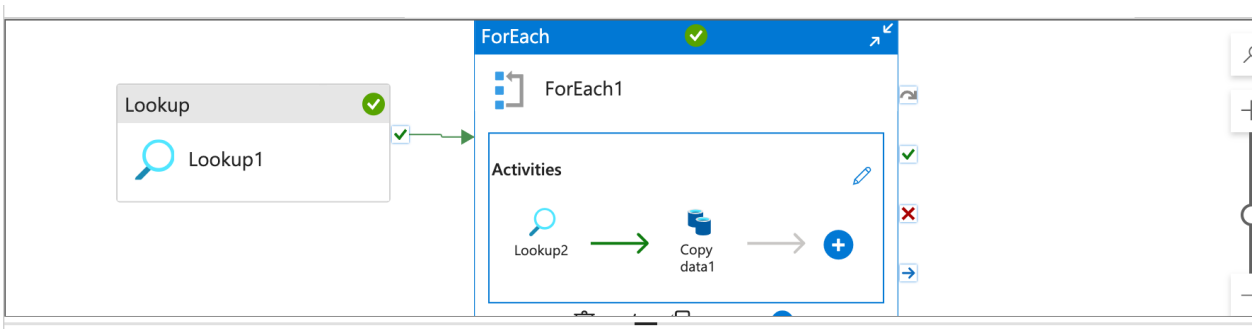
Source dataset *

AzureSqlTable1 Open New Preview data Learn more

Dataset properties

Name	Value	Type
schema_name	dbo	string
table_name	WatermarkTable	string

First row only

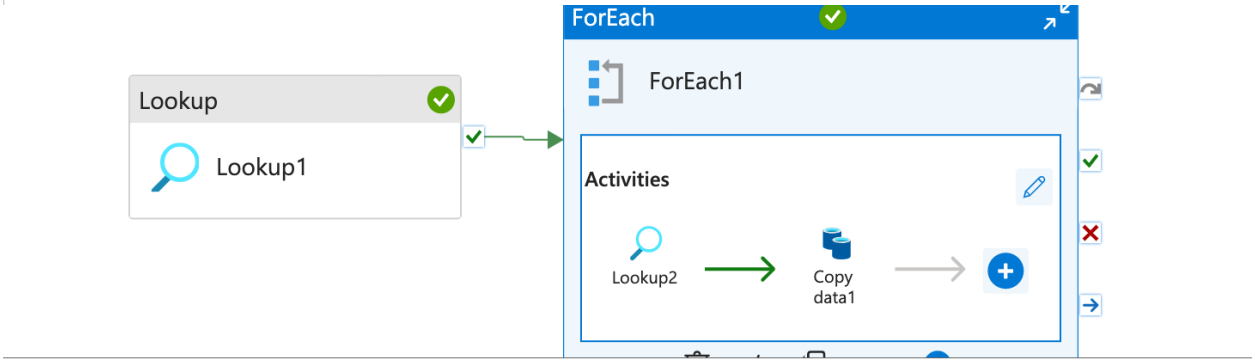


General | Settings | Activities (2) | User properties

Sequential

Batch count

Items: @activity('Lookup1').output.value



General | Settings | User properties

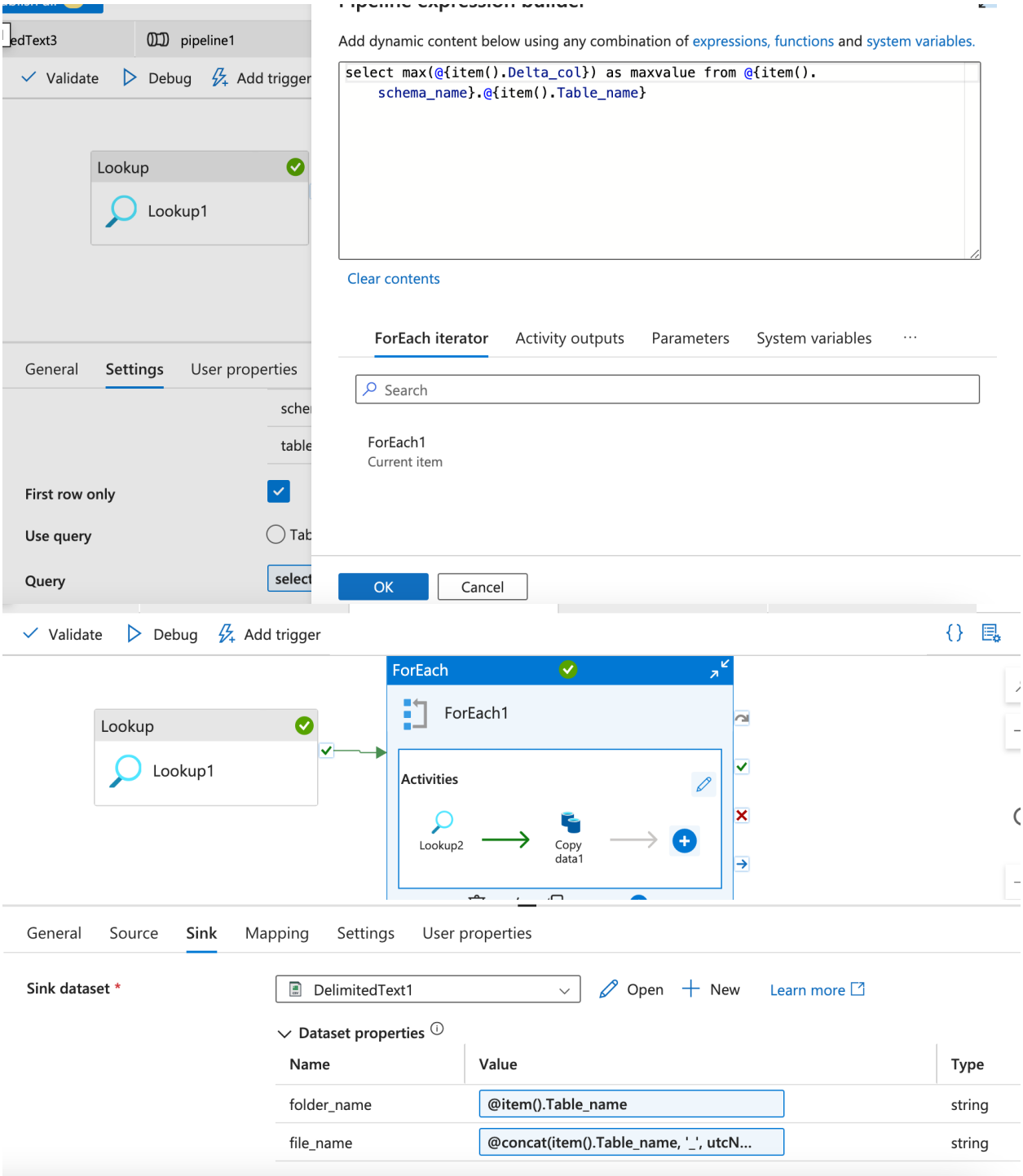
schema_name: ''

table_name: ''

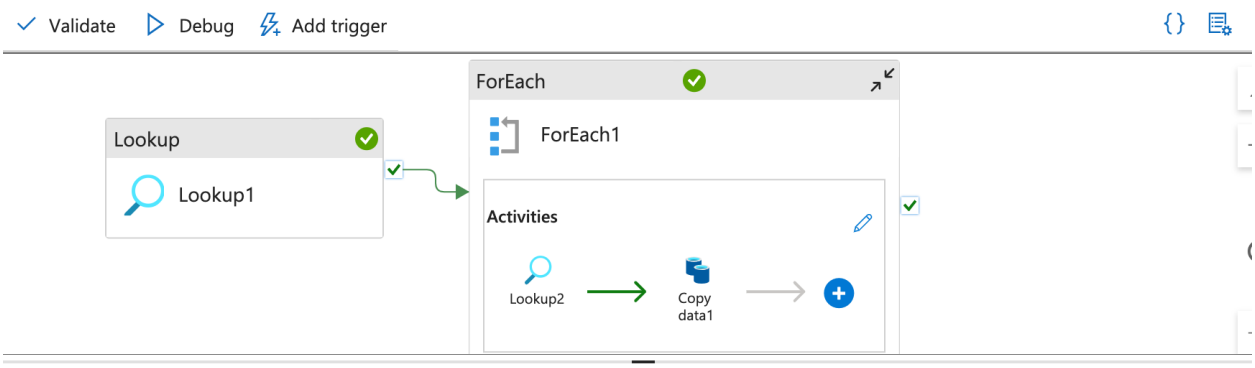
First row only: ☑

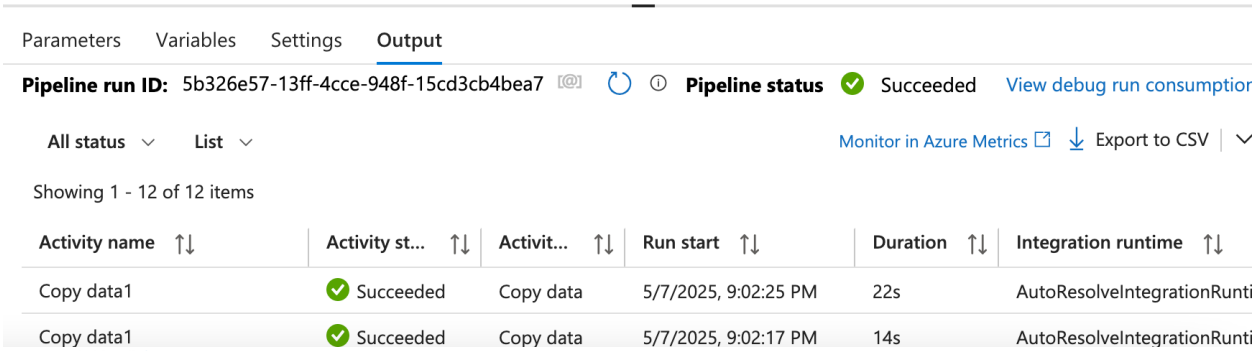
Use query: ○ Table ◉ Query ○ Stored procedure

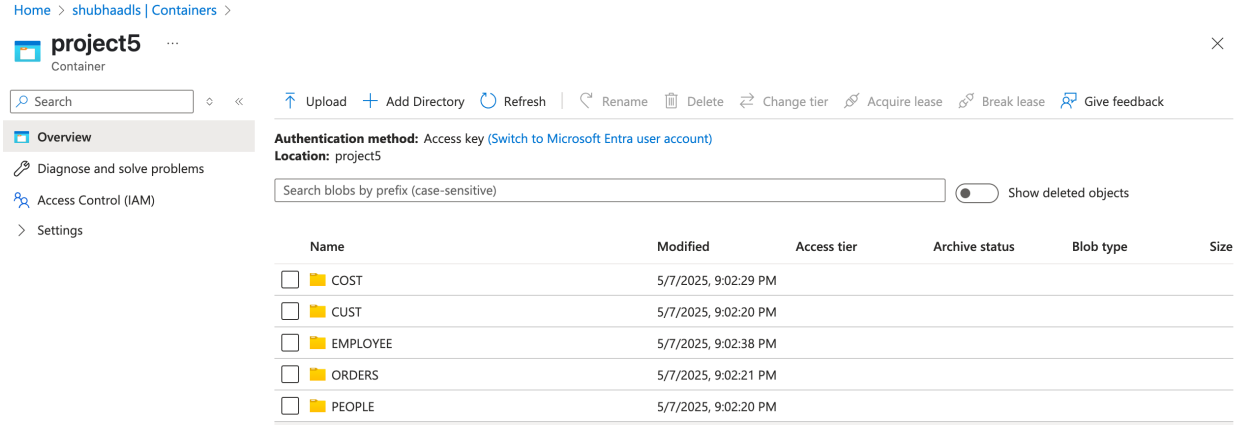
Query: select max(@{item().Delta_col}) as m...



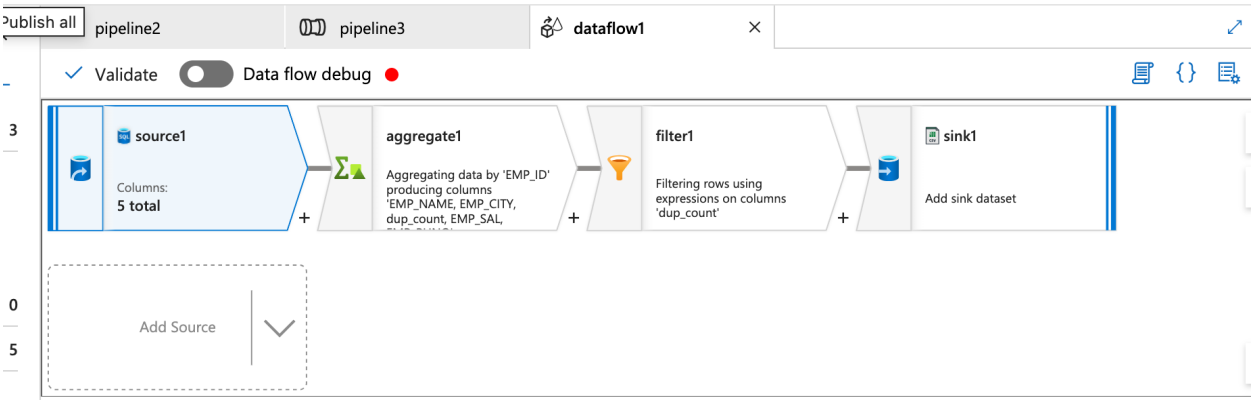
pipeline ran successfully

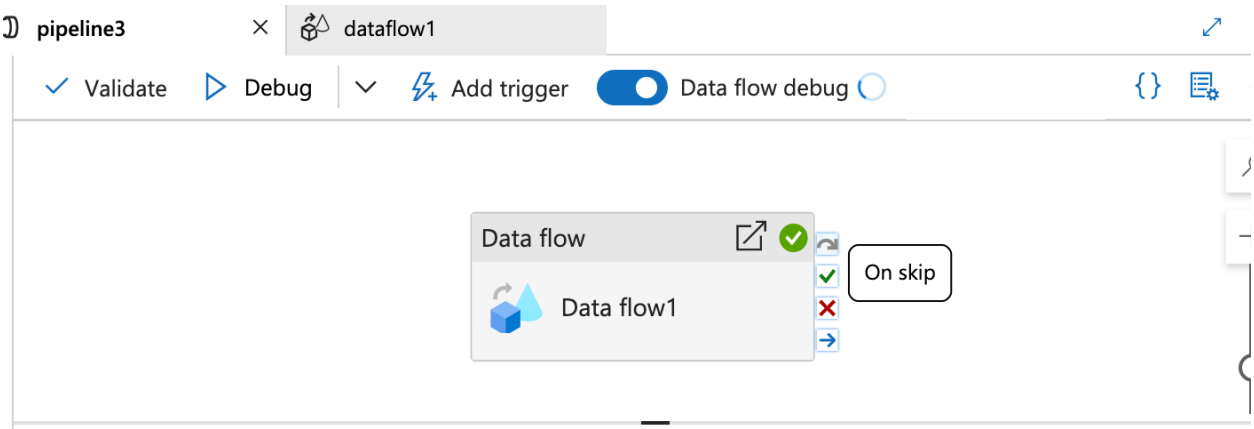






Create a pipeline to clean the data using Dataflows.





Parameters Variables Settings **Output**

Pipeline run ID: 80ae0921-bd5f-40cb-8a7a-0ddf2f907def

Pipeline status Succeeded

View debug run consumption

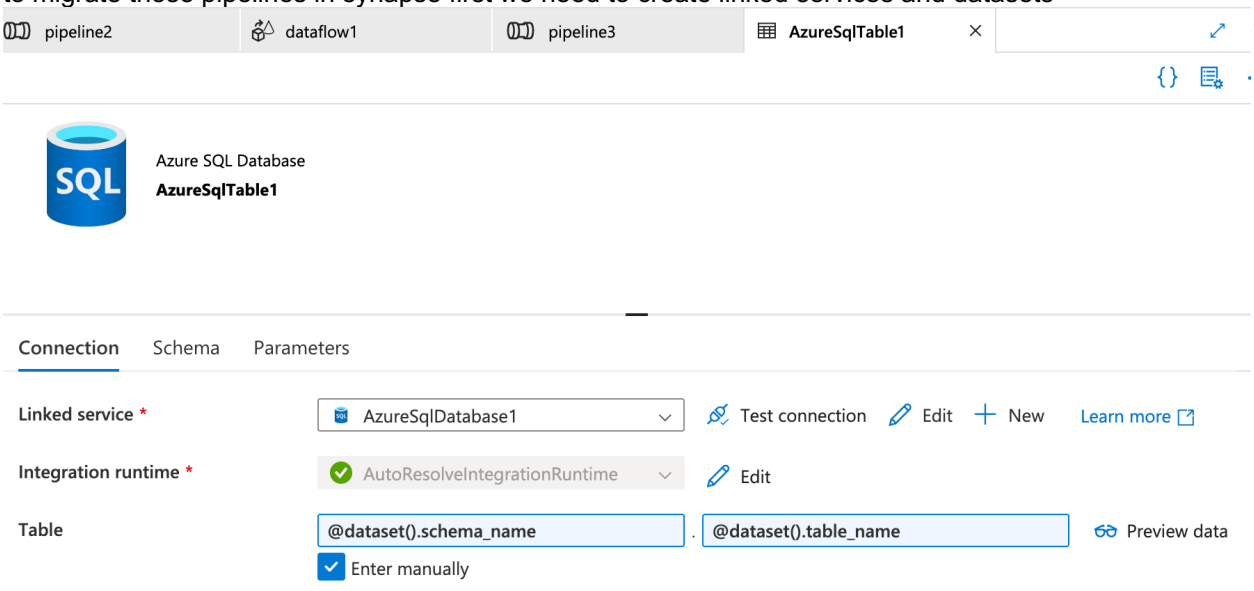
All status

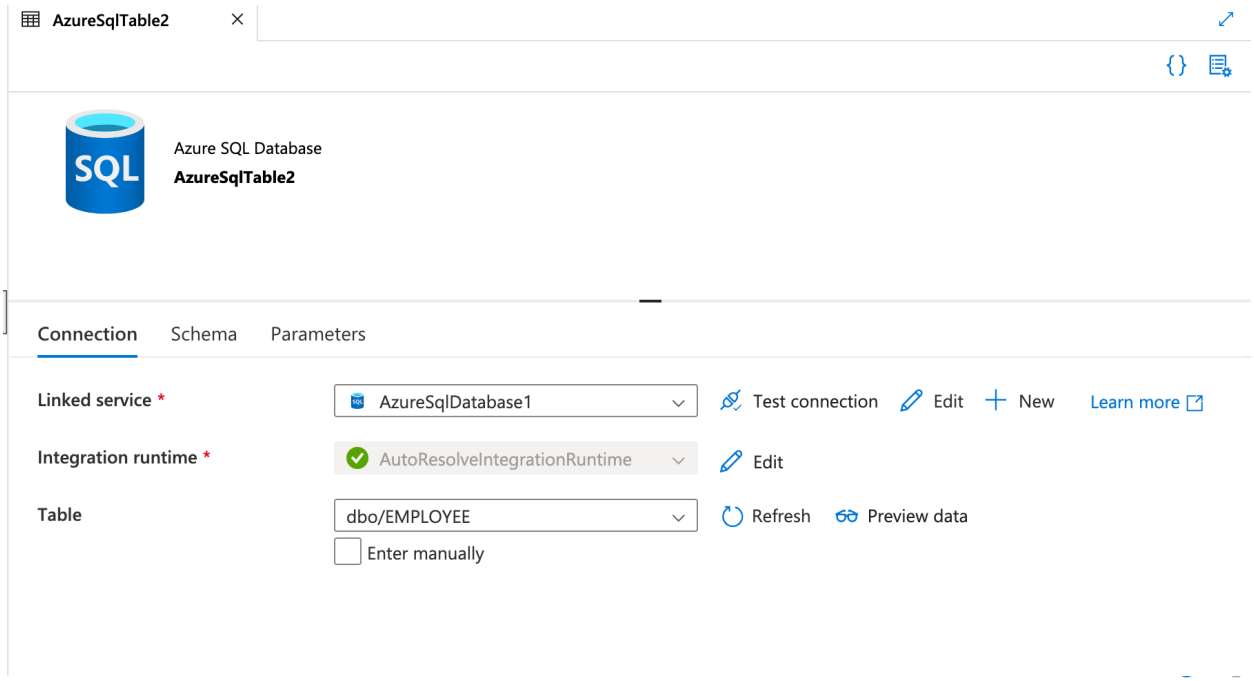
Monitor in Azure Metrics | Export to CSV

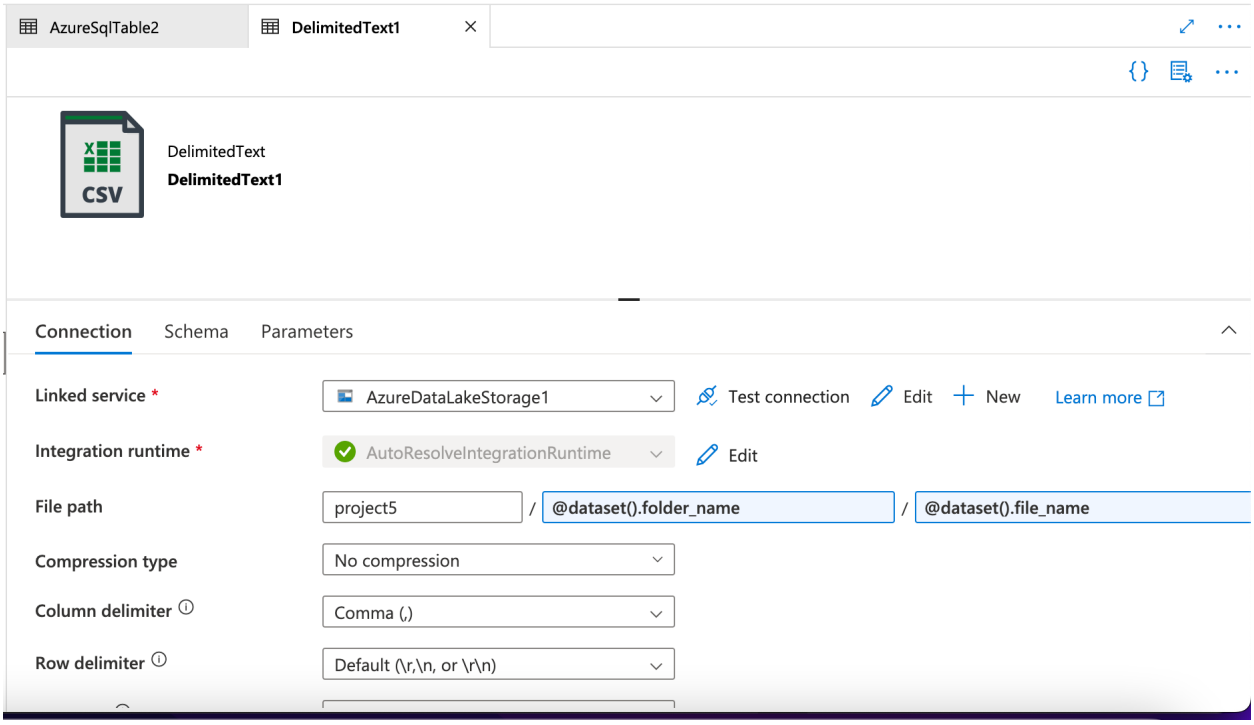
Showing 1 - 1 of 1 items

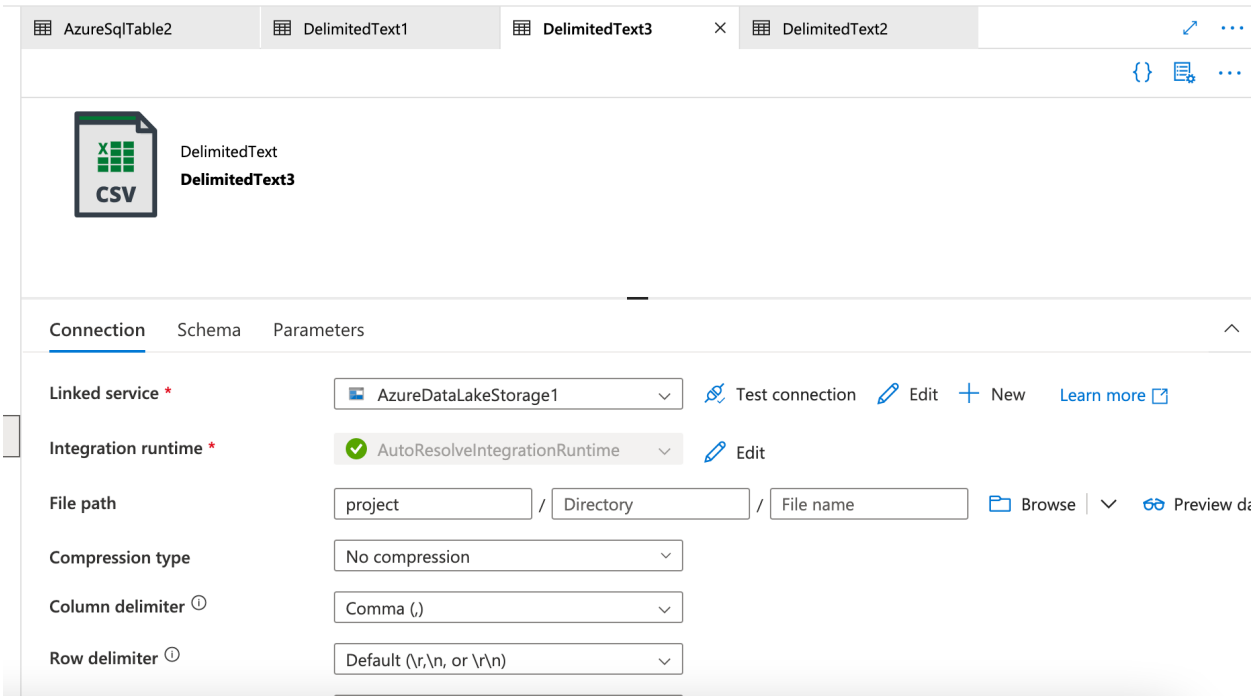
Activity name	Activity st...	Activit...	Run start	Duration
Data flow1	Succeeded	Data flow	5/7/2025, 9:20:16 PM	53s

to migrate these pipelines in synapse first we need to create linked services and datasets

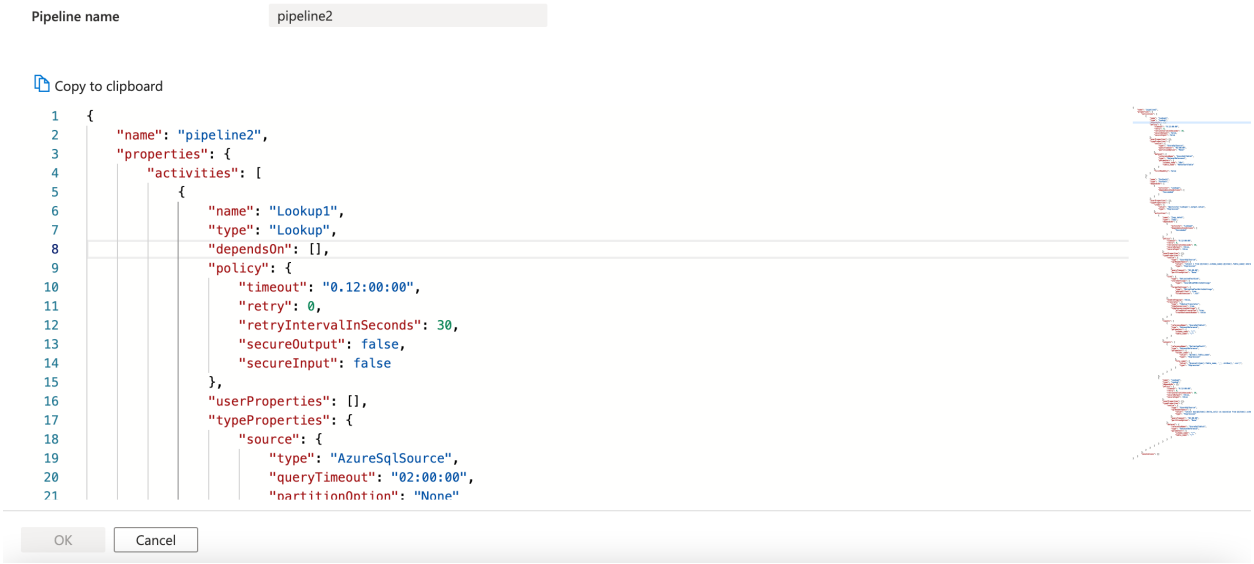


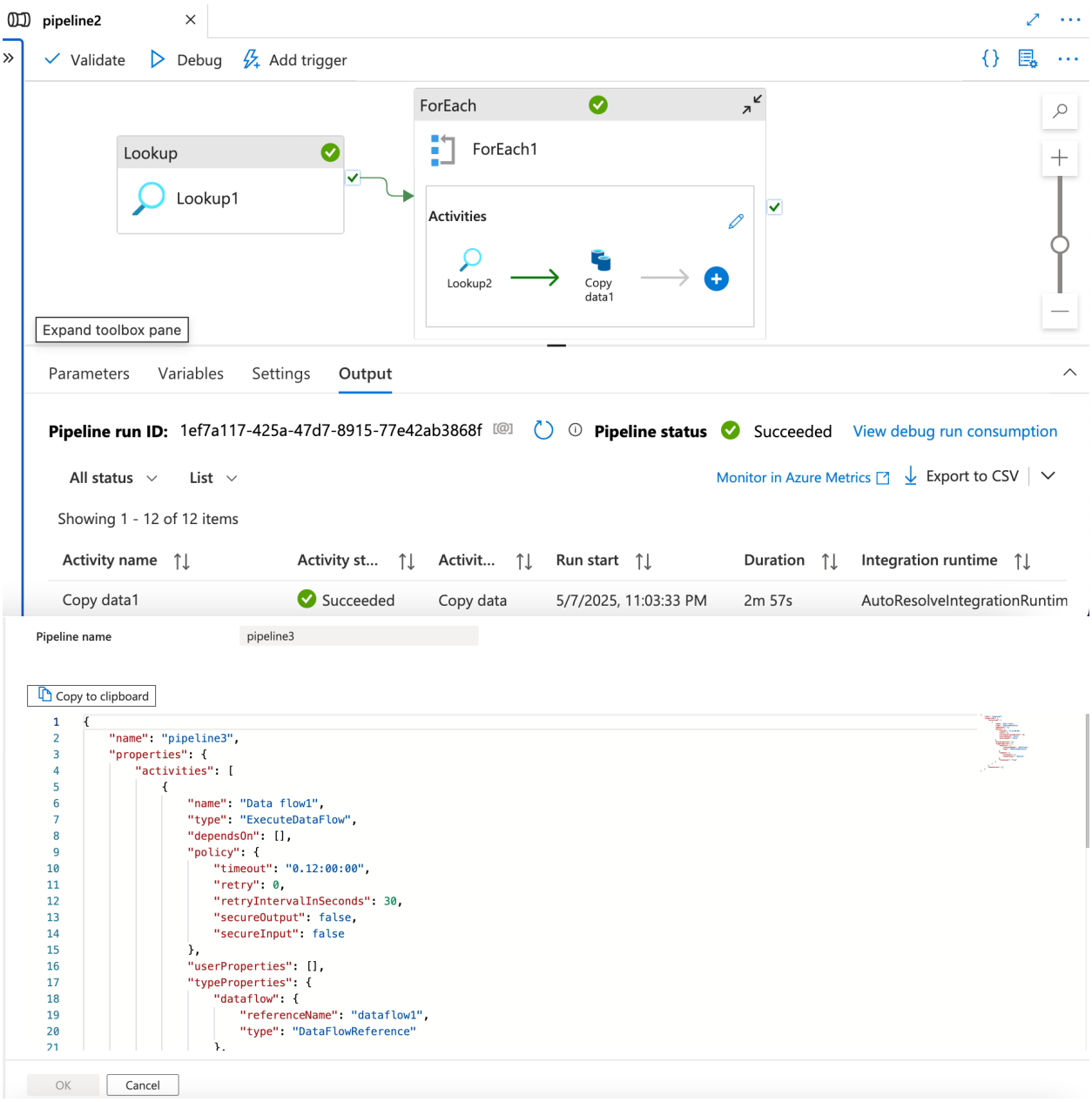

AzureSqlTable2
Azure SQL Database
AzureSqlTable2
Connection
Schema
Parameters
Linked service *
AzureSqlDatabase1
Test connection
Edit
New
Learn more
Integration runtime *
AutoResolveIntegrationRuntime
Edit
Table
dbo/EMPLOYEE
Refresh
Preview data
Enter manually

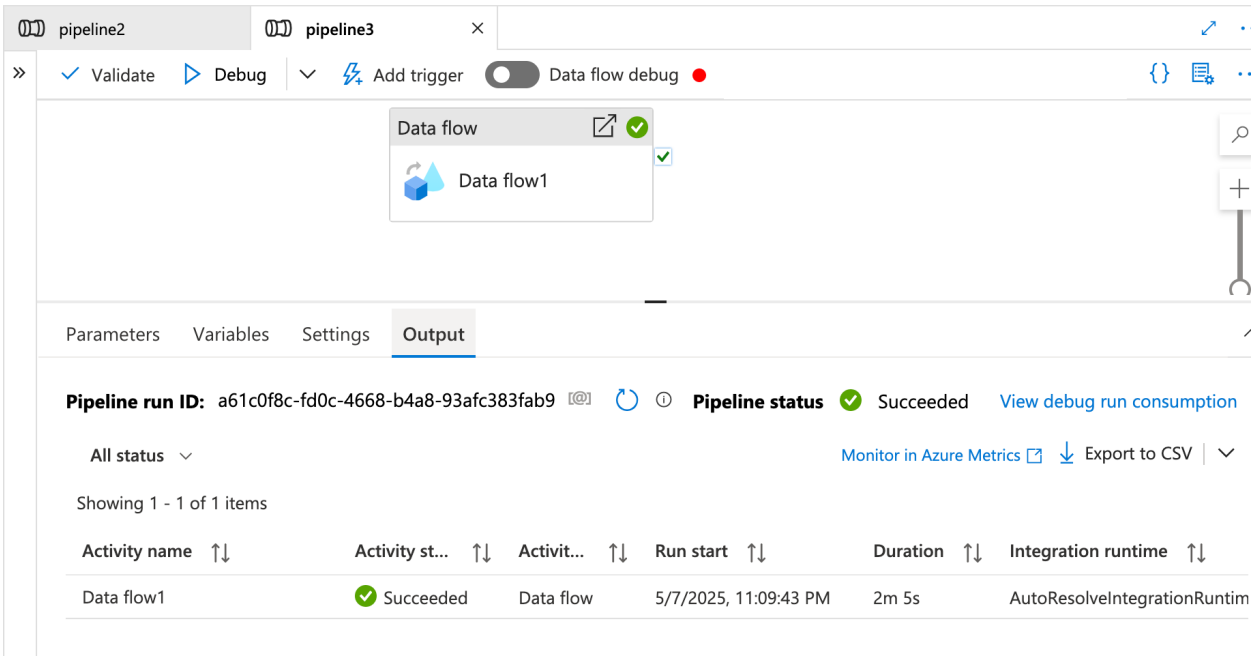

AzureSqlTable2
DelimitedText1
DelimitedText
DelimitedText1
Connection
Schema
Parameters
Linked service *
AzureDataLakeStorage1
Test connection
Edit
New
Learn more
Integration runtime *
AutoResolveIntegrationRuntime
Edit
File path
project5
@dataset().folder_name
@dataset().file_name
Compression type
No compression
Column delimiter
Comma (,)
Row delimiter
Default (\r,\n, or \r\n)



now copy the json from ADF and paste it in synapse



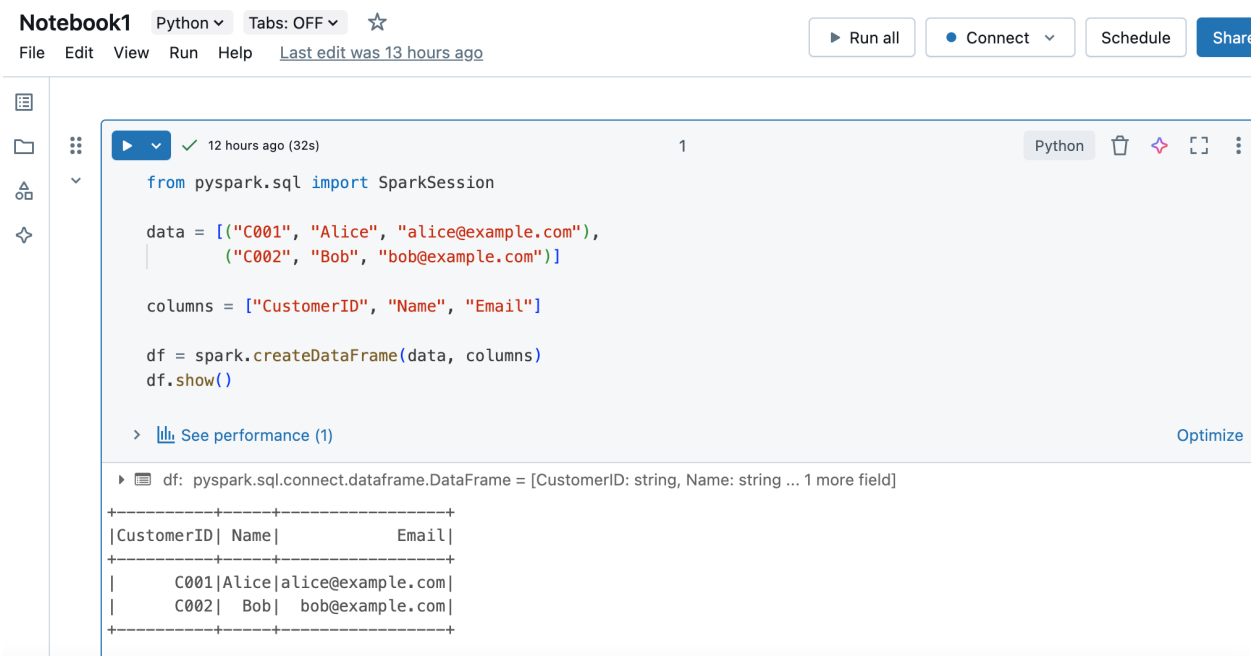

pipeline2
Validate
Debug
Add trigger
ForEach
ForEach1
Lookup
Lookup1
Activities
Lookup2
Copy data1
Expand toolbox pane
Parameters
Variables
Settings
Output
Pipeline run ID: 1ef7a117-425a-47d7-8915-77e42ab3868f
Pipeline status
Succeeded
View debug run consumption
All status
List
Monitor in Azure Metrics
Export to CSV
Showing 1 - 12 of 12 items
Activity name
Activity st...
Activit...
Run start
Duration
Integration runtime
Copy data1
Succeeded
Copy data
5/7/2025, 11:03:33 PM
2m 57s
AutoResolveIntegrationRuntim
Pipeline name
pipeline3
Copy to clipboard
{
"name": "pipeline3",
"properties": {
"activities": [
{
"name": "Data flow1",
"type": "ExecuteDataFlow",
"dependsOn": [],
"policy": {
"timeout": "0.12:00:00",
"retry": 0,
"retryIntervalInSeconds": 30,
"secureOutput": false,
"secureInput": false
},
"userProperties": [],
"typeProperties": {
"dataflow": {
"referenceName": "dataflow1",
"type": "DataFlowReference"
},
OK
Cancel

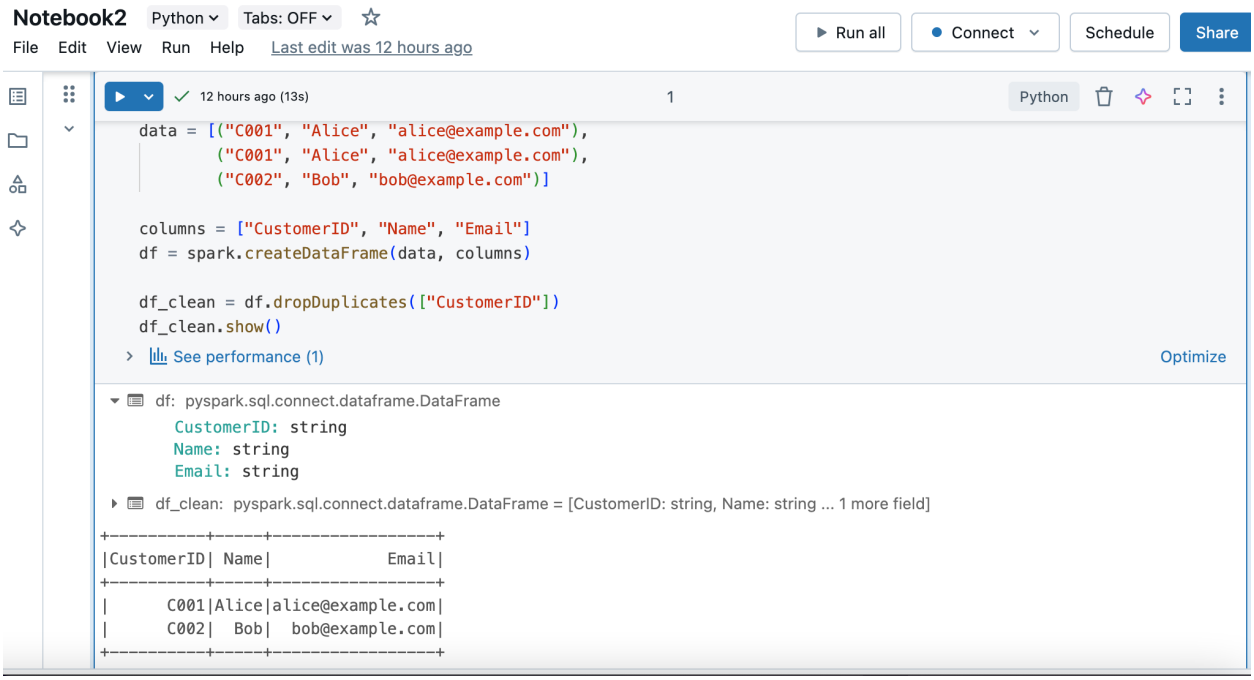


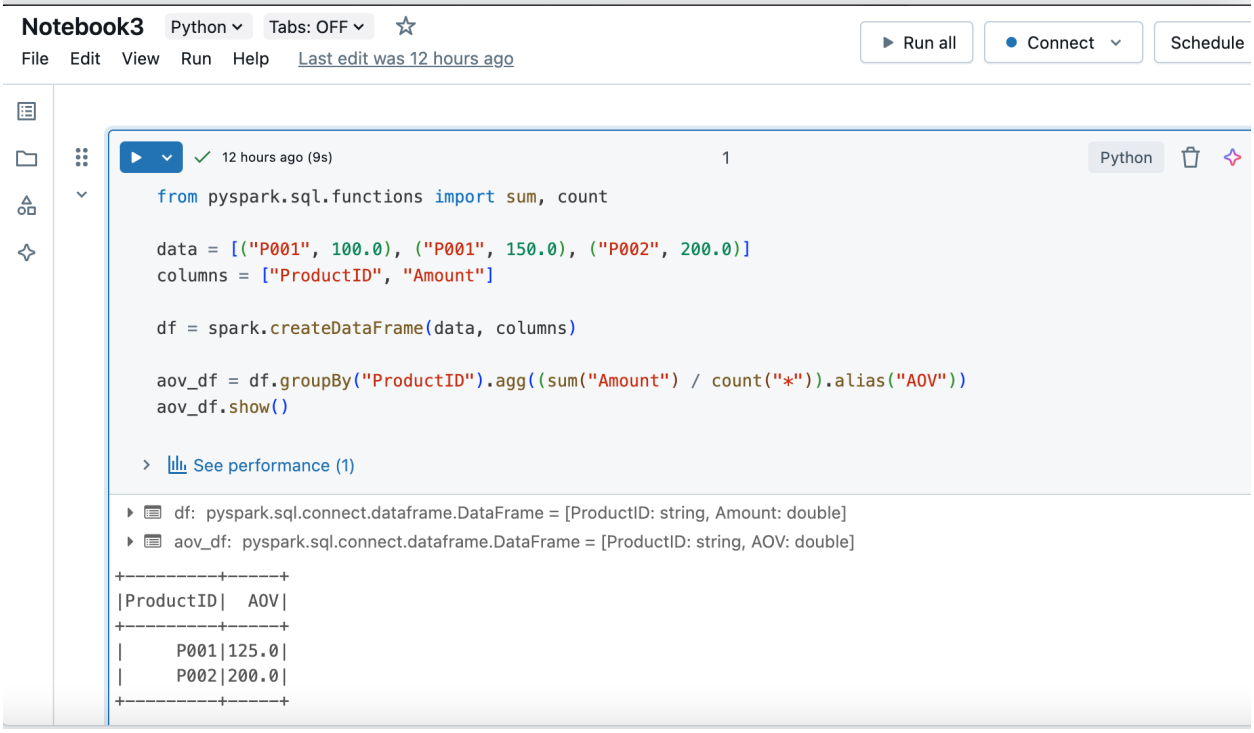
pipelines are migrated from ADF to Synapse and they have ran successfully

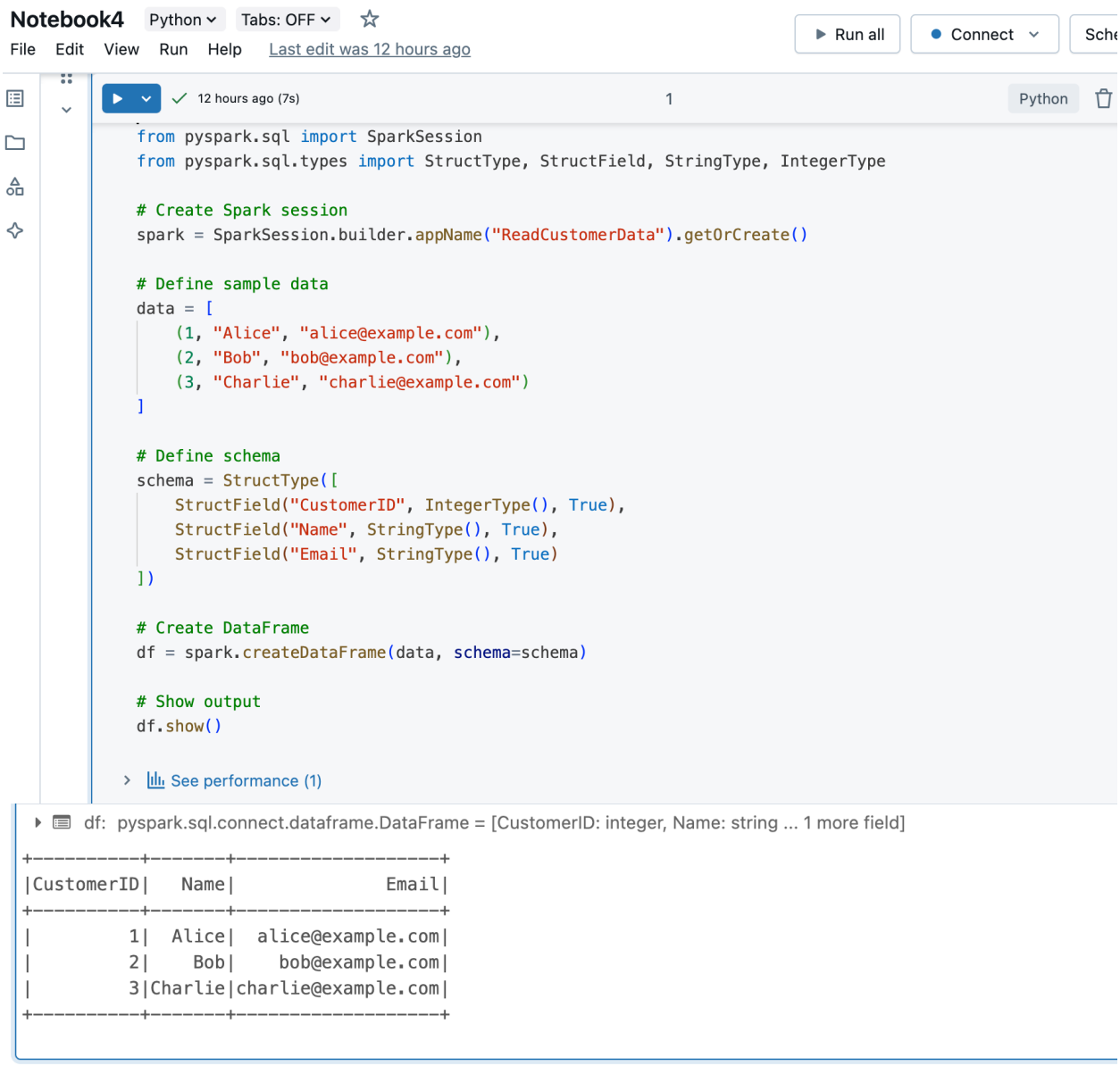
Task2

Migrating Databricks notebook to fabric

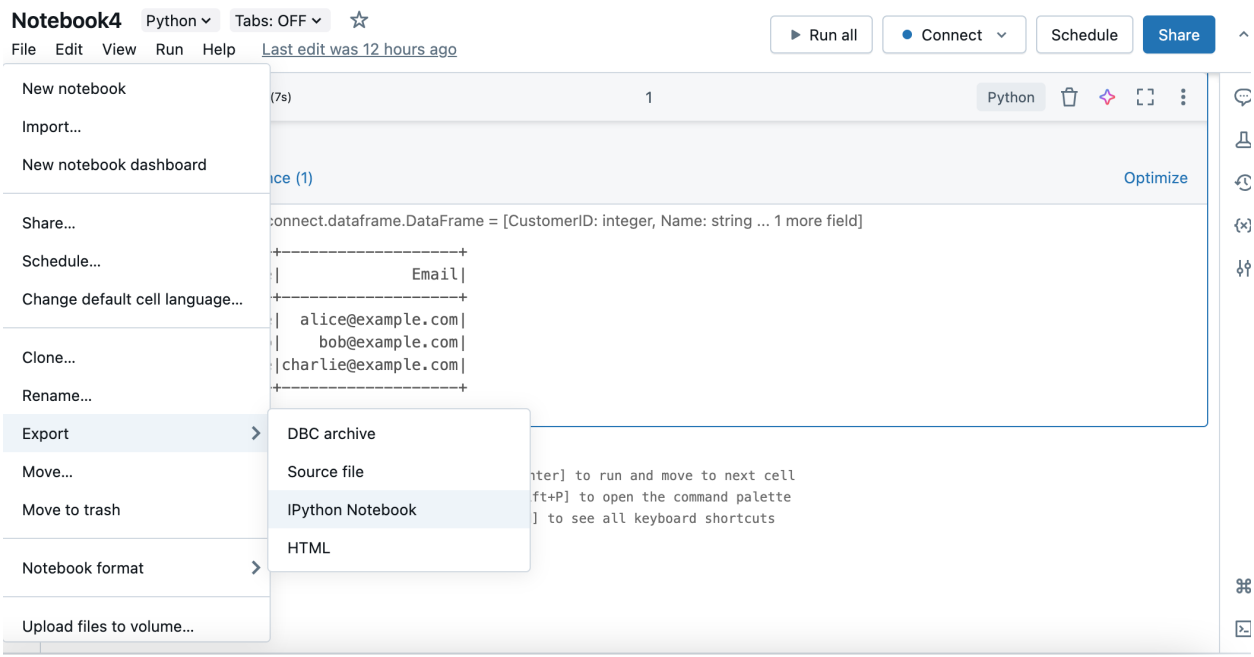



Notebook2
Python
Tabs: OFF
File Edit View Run Help
Last edit was 12 hours ago
Run all
Connect
Schedule
Share
12 hours ago (13s)
1
Python
data = [("C001", "Alice", "alice@example.com"),
("C001", "Alice", "alice@example.com"),
("C002", "Bob", "bob@example.com")]
columns = ["CustomerID", "Name", "Email"]
df = spark.createDataFrame(data, columns)
df_clean = df.dropDuplicates(["CustomerID"])
df_clean.show()
See performance (1)
Optimize
df: pyspark.sql.connect.dataframe.DataFrame
CustomerID: string
Name: string
Email: string
df_clean: pyspark.sql.connect.dataframe.DataFrame = [CustomerID: string, Name: string ... 1 more field]
+----------+-----+-----------------+
|CustomerID| Name| Email|
+----------+-----+-----------------+
| C001|Alice|alice@example.com|
| C002| Bob| bob@example.com|
+----------+-----+-----------------+


Notebook3
Python
Tabs: OFF
File Edit View Run Help
Last edit was 12 hours ago
Run all
Connect
Schedule
12 hours ago (9s)
1
Python
from pyspark.sql.functions import sum, count
data = [("P001", 100.0), ("P001", 150.0), ("P002", 200.0)]
columns = ["ProductID", "Amount"]
df = spark.createDataFrame(data, columns)
aov_df = df.groupBy("ProductID").agg((sum("Amount") / count("*")).alias("AOV"))
aov_df.show()
See performance (1)
df: pyspark.sql.connect.dataframe.DataFrame = [ProductID: string, Amount: double]
aov_df: pyspark.sql.connect.dataframe.DataFrame = [ProductID: string, AOV: double]
+---------+-----+
|ProductID| AOV|
+---------+-----+
| P001|125.0|
| P002|200.0|
+---------+-----+

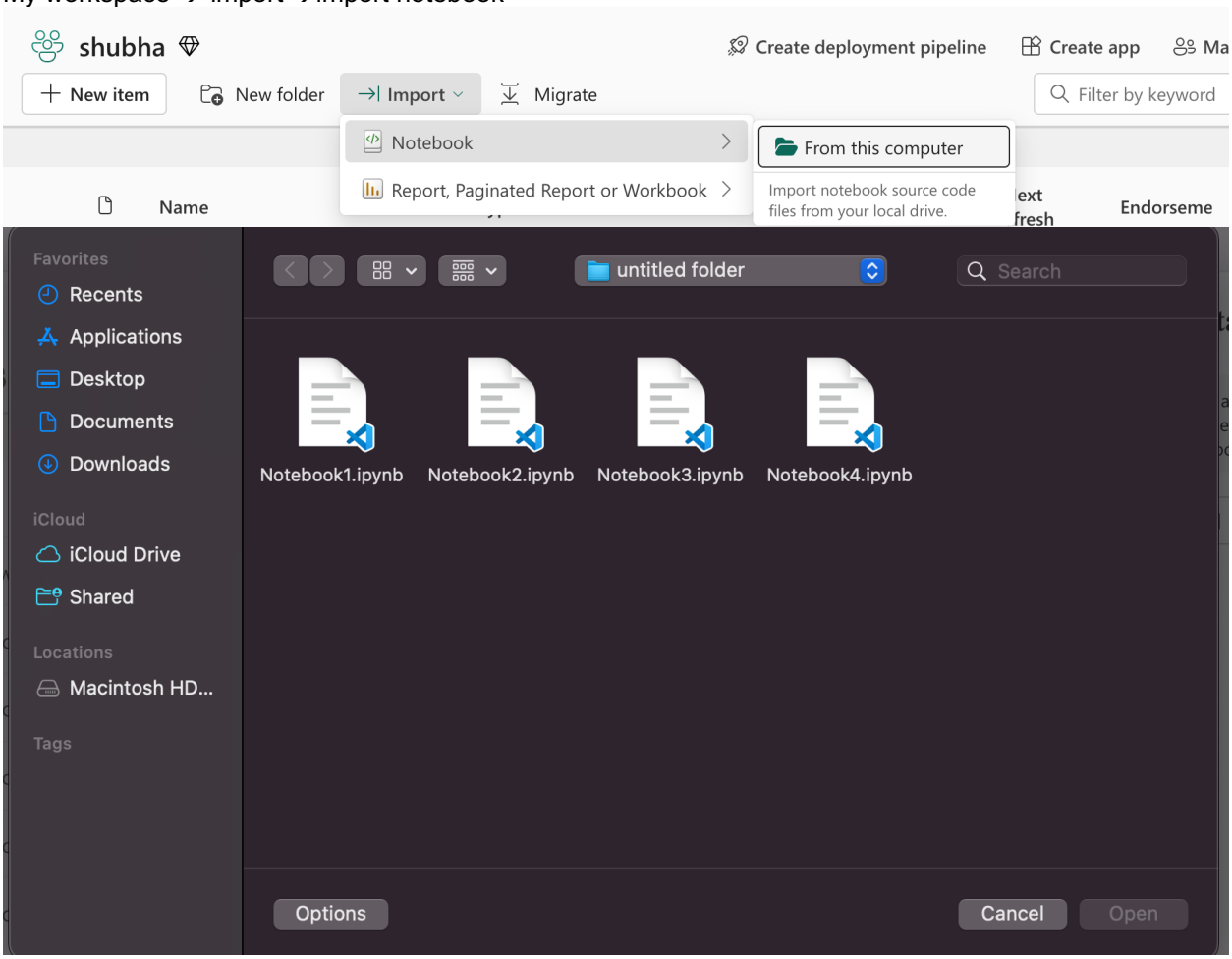


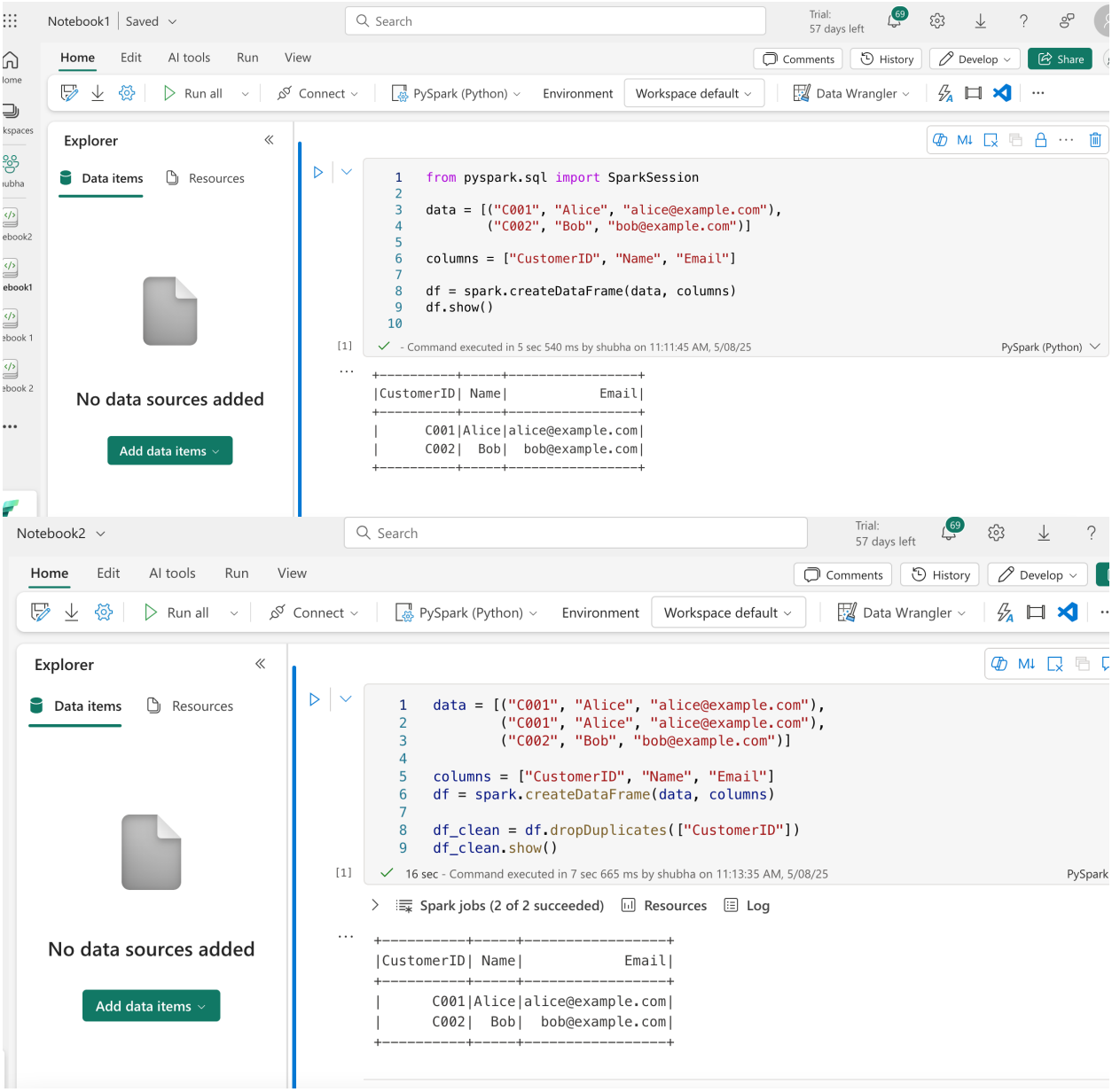
now download all these 4 notebooks in python format



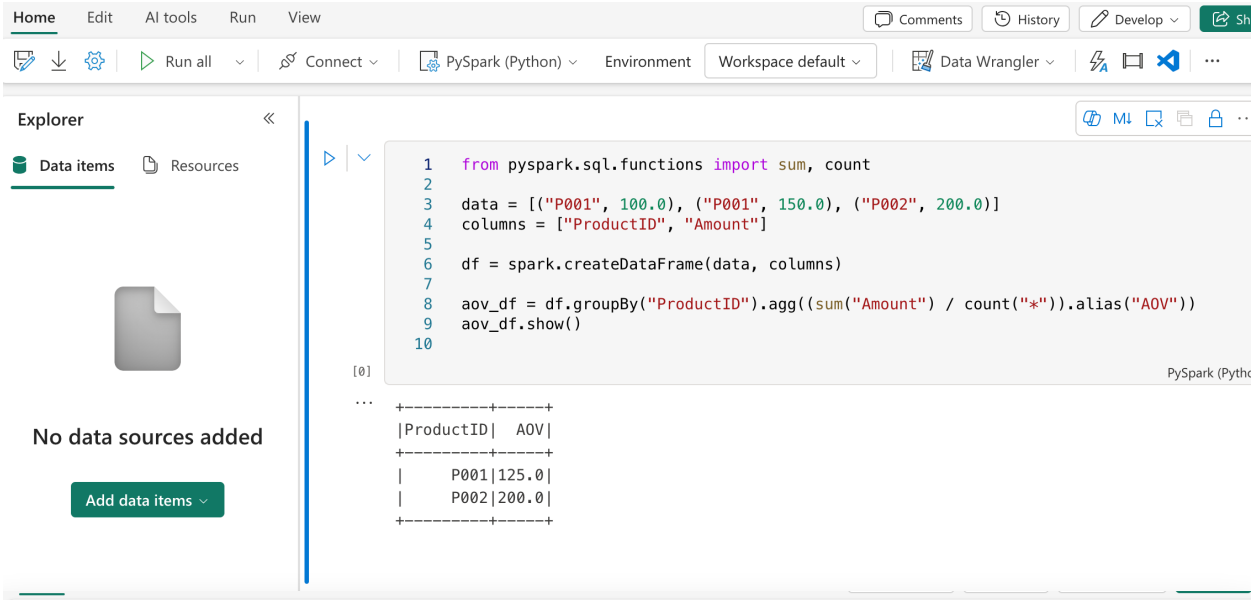
now lets import these downloaded exported ython otebooks to fabricks

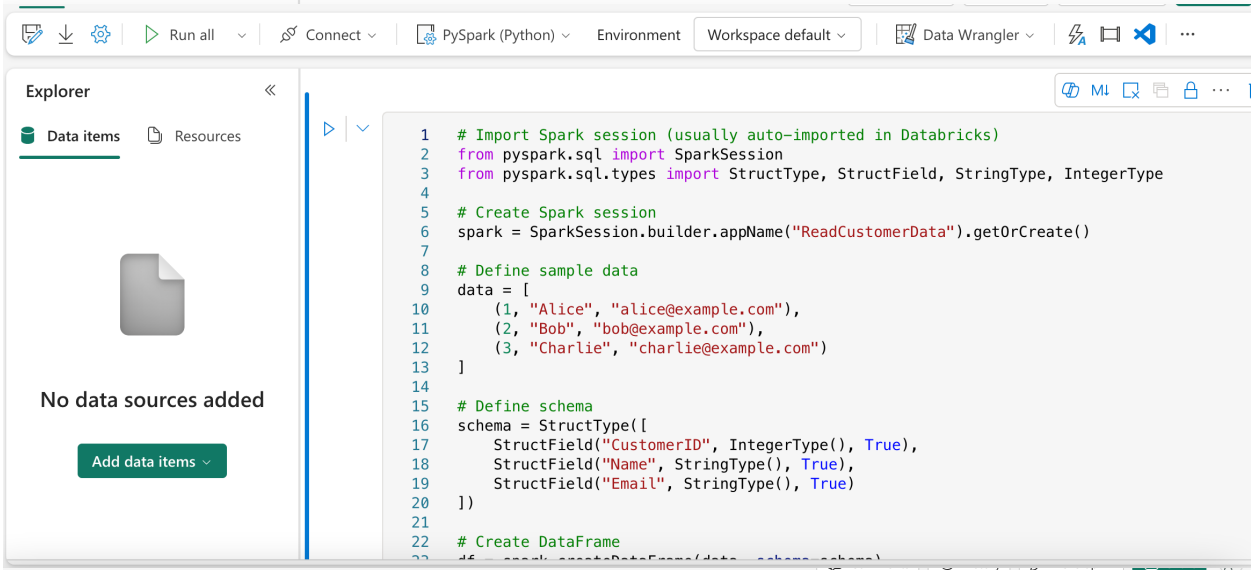
My workspace → import →import notebook

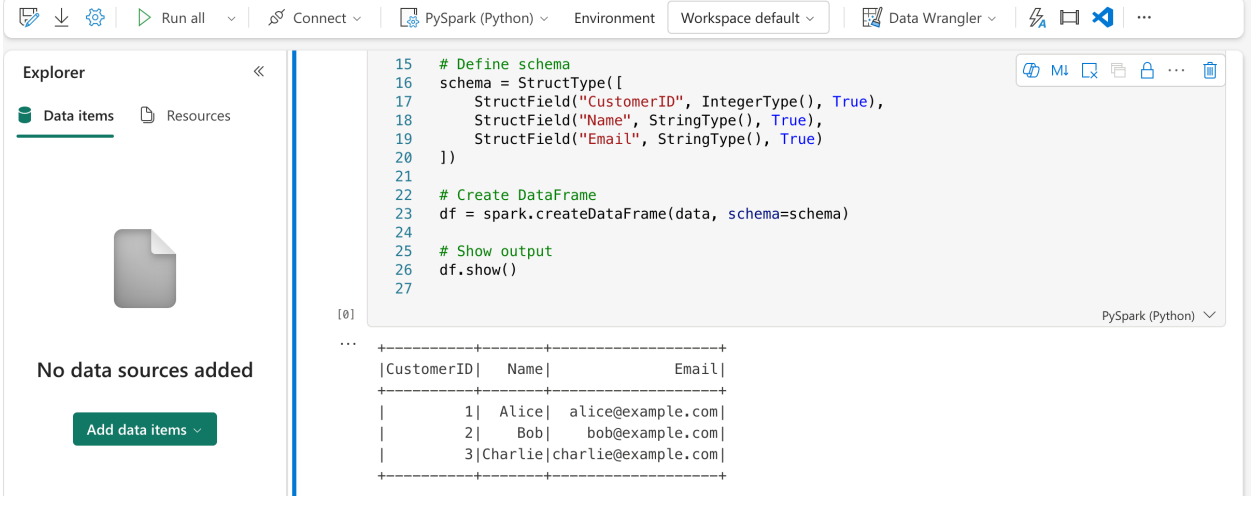



Notebook1 | Saved
Search
Trial: 57 days left
Home
Edit
AI tools
Run
View
Comments
History
Develop
Share
Run all
Connect
PySpark (Python)
Environment
Workspace default
Data Wrangler
Explorer
Data items
Resources
No data sources added
Add data items
1 from pyspark.sql import SparkSession
2
3 data = [("C001", "Alice", "alice@example.com"),
4 ("C002", "Bob", "bob@example.com")]
5
6 columns = ["CustomerID", "Name", "Email"]
7
8 df = spark.createDataFrame(data, columns)
9 df.show()
10
[1]
- Command executed in 5 sec 540 ms by shubha on 11:11:45 AM, 5/08/25
PySpark (Python)
+----------+-----+-----------------+
|CustomerID| Name| Email|
+----------+-----+-----------------+
| C001|Alice|alice@example.com|
| C002| Bob| bob@example.com|
+----------+-----+-----------------+

Notebook2
Search
Trial: 57 days left
Home
Edit
AI tools
Run
View
Comments
History
Develop
Run all
Connect
PySpark (Python)
Environment
Workspace default
Data Wrangler
Explorer
Data items
Resources
No data sources added
Add data items
1 data = [("C001", "Alice", "alice@example.com"),
2 ("C001", "Alice", "alice@example.com"),
3 ("C002", "Bob", "bob@example.com")]
4
5 columns = ["CustomerID", "Name", "Email"]
6 df = spark.createDataFrame(data, columns)
7
8 df_clean = df.dropDuplicates(["CustomerID"])
9 df_clean.show()
[1]
16 sec - Command executed in 7 sec 665 ms by shubha on 11:13:35 AM, 5/08/25
PySpark
Spark jobs (2 of 2 succeeded)
Resources
Log
+----------+-----+-----------------+
|CustomerID| Name| Email|
+----------+-----+-----------------+
| C001|Alice|alice@example.com|
| C002| Bob| bob@example.com|
+----------+-----+-----------------+


Home
Edit
AI tools
Run
View
Comments
History
Develop
Run all
Connect
PySpark (Python)
Environment
Workspace default
Data Wrangler
Explorer
Data items
Resources
No data sources added
Add data items
1 from pyspark.sql.functions import sum, count
2
3 data = [("P001", 100.0), ("P001", 150.0), ("P002", 200.0)]
4 columns = ["ProductID", "Amount"]
5
6 df = spark.createDataFrame(data, columns)
7
8 aov_df = df.groupBy("ProductID").agg((sum("Amount") / count("*")).alias("AOV"))
9 aov_df.show()
10
[0]
PySpark (Pyth
+---------+-----+
|ProductID| AOV|
+---------+-----+
| P001|125.0|
| P002|200.0|
+---------+-----+


Run all
Connect
PySpark (Python)
Environment
Workspace default
Data Wrangler
Explorer
Data items
Resources
No data sources added
Add data items
1 # Import Spark session (usually auto-imported in Databricks)
2 from pyspark.sql import SparkSession
3 from pyspark.sql.types import StructType, StructField, StringType, IntegerType
4
5 # Create Spark session
6 spark = SparkSession.builder.appName("ReadCustomerData").getOrCreate()
7
8 # Define sample data
9 data = [
10 (1, "Alice", "alice@example.com"),
11 (2, "Bob", "bob@example.com"),
12 (3, "Charlie", "charlie@example.com")
13]
14
15 # Define schema
16 schema = StructType([
17 StructField("CustomerID", IntegerType(), True),
18 StructField("Name", StringType(), True),
19 StructField("Email", StringType(), True)
20])
21
22 # Create DataFrame


Run all
Connect
PySpark (Python)
Environment
Workspace default
Data Wrangler
Explorer
Data items
Resources
No data sources added
Add data items
15 # Define schema
16 schema = StructType([
17 StructField("CustomerID", IntegerType(), True),
18 StructField("Name", StringType(), True),
19 StructField("Email", StringType(), True)
20])
21
22 # Create DataFrame
23 df = spark.createDataFrame(data, schema=schema)
24
25 # Show output
26 df.show()
27
[0]
PySpark (Python)
+----------+-------+-------------------+
|CustomerID| Name| Email|
+----------+-------+-------------------+
1	Alice	alice@example.com
2	Bob	bob@example.com
3	Charlie	charlie@example.com
+----------+-------+-------------------+