

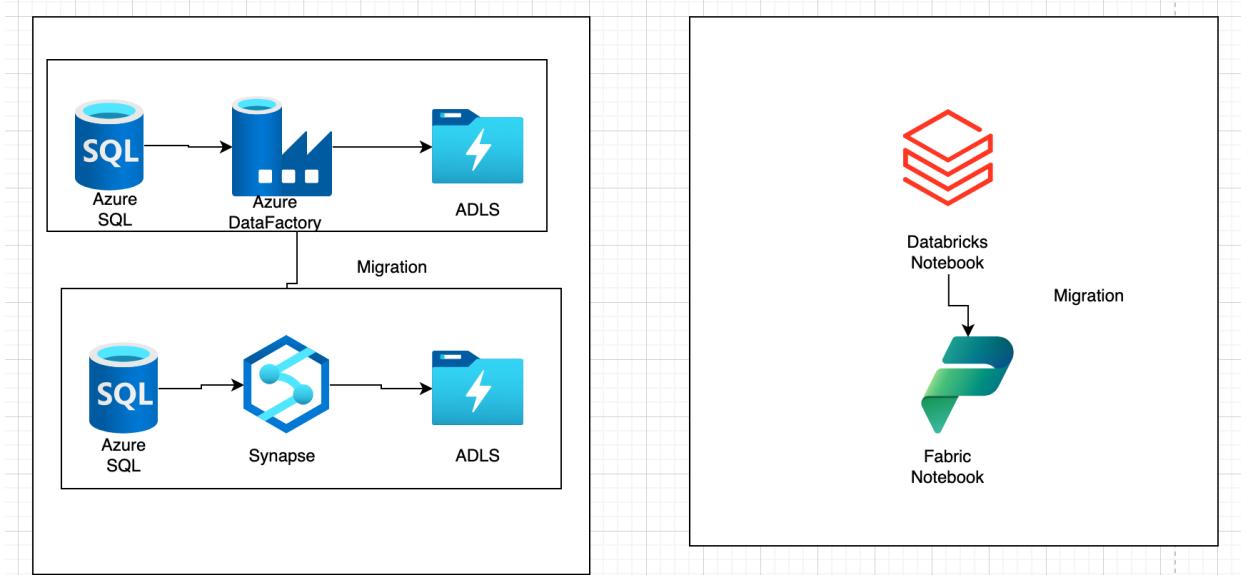
Bootcamp Project 5

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 Enable

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
19 CREATE TABLE EMPLOYEE
```

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
```

Results 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

```

30  CREATE table CUST(
31    CID INT, CNAME VARCHAR(100), CCITY VARCHAR(100), UPDATED_DATE DATETIME)
32
33
34  INSERT INTO CUST (CID, CNAME, CCITY, UPDATED_DATE)
35  VALUES (1, 'John Doe', 'New York', '2025-03-01 10:30:00');
36
37  INSERT INTO CUST (CID, CNAME, CCITY, UPDATED_DATE)
38  VALUES (2, 'Jane Smith', 'Los Angeles', '2025-03-01 11:00:00');
39
40  INSERT INTO CUST (CID, CNAME, CCITY, UPDATED_DATE)
41  VALUES (3, 'Robert Brown', 'Chicago', '2025-03-01 12:00:00');
42  select * from CUST
43
44  CRFATE TABLE COST (

```

results 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

```

44  CREATE TABLE COST (
45      COST_ID INT PRIMARY KEY,
46      COST_NAME VARCHAR(100) NOT NULL,
47      COST_AMOUNT DECIMAL(10,2),
48      UPDATED_DATE DATETIME
49  );
50  INSERT INTO COST (COST_ID, COST_NAME, COST_AMOUNT, UPDATED_DATE)
51  VALUES (1, 'Labor Cost', 5000.00, '2025-03-01 10:00:00');
52
53  INSERT INTO COST (COST_ID, COST_NAME, COST_AMOUNT, UPDATED_DATE)
54  VALUES (2, 'Material Cost', 12000.50, '2025-03-01 11:00:00');
55
56  INSERT INTO COST (COST_ID, COST_NAME, COST_AMOUNT, UPDATED_DATE)
57  VALUES (3, 'Transportation Cost', 3000.75, '2025-03-01 12:00:00');
58

```

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

```

61  CREATE TABLE ORDERS (
62      ORDER_ID INT,
63      CUSTOMER_ID INT,
64      ORDER_AMOUNT DECIMAL(10,2),
65      ORDER_DATE DATETIME
66  );
67  INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, ORDER_AMOUNT, ORDER_DATE)
68  VALUES (1, 1, 150.00, '2025-03-01 14:00:00');
69
70  INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, ORDER_AMOUNT, ORDER_DATE)
71  VALUES (2, 2, 250.75, '2025-03-01 15:00:00');
72
73  INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, ORDER_AMOUNT, ORDER_DATE)
74  VALUES (3, 3, 99.99, '2025-03-01 16:00:00');
75

```

Results **Messages**

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

```

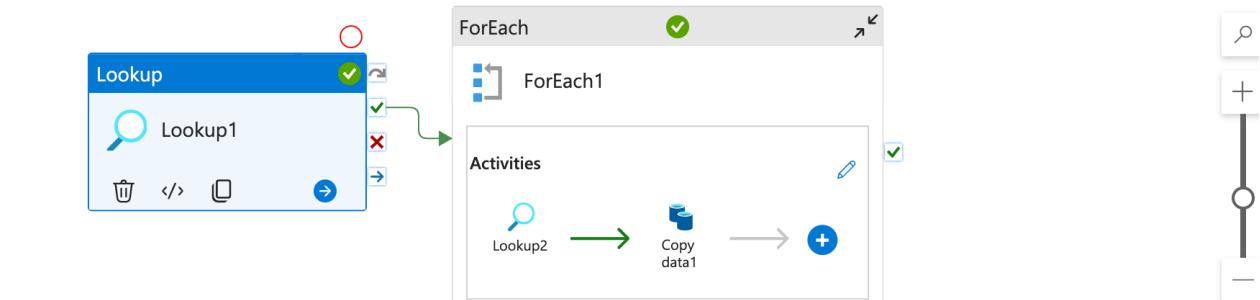
CREATE TABLE WatermarkTable (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    schema_name VARCHAR(50),
    Table_name VARCHAR(50),
    LPV VARCHAR(100),
    Delta_col VARCHAR(50)
);

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')

```

Its 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

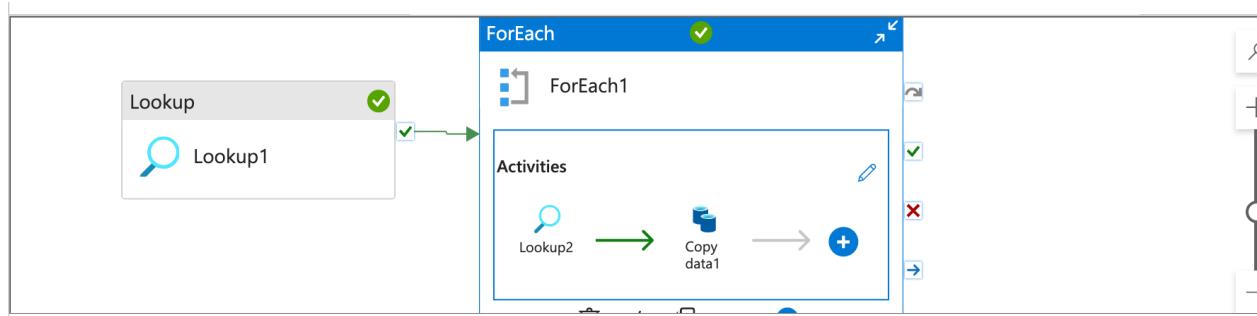


General Settings User properties

source dataset *

first row only

Dataset properties		
Name	Value	Type
schema_name	dbo	string
table_name	WatermarkTable	string

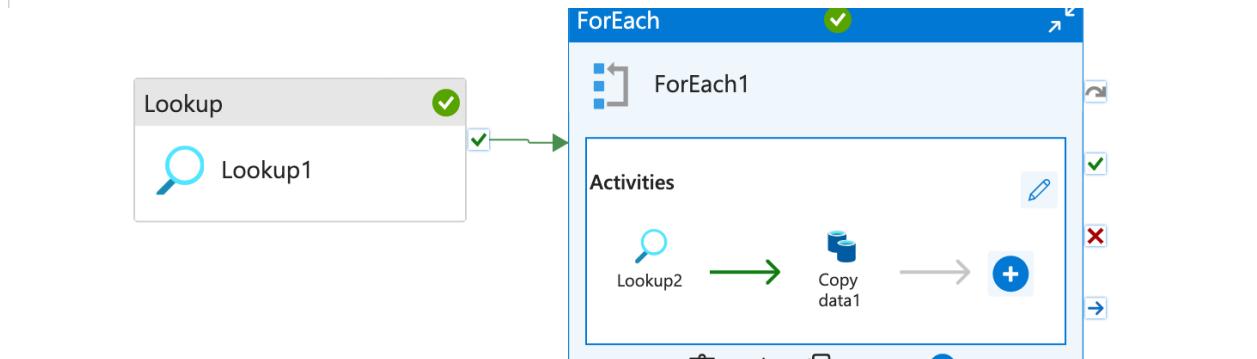


General **Settings** Activities (2) User properties

Sequential

Batch count

Items



General **Settings** User properties

schema_name

table_name

First row only

Table Query Stored procedure

Query

pipeline1

Add dynamic content below using any combination of **expressions**, **functions** and **system variables**.

```
select max(@{item().Delta_col}) as maxvalue from @{item().schema_name}.@{item().Table_name}
```

ForEach iterator Activity outputs Parameters System variables ...

General **Settings** User properties

First row only

Use query Tabular query SQL query

Query

OK **Cancel**

ForEach1
Current item

ForEach

Activities

Lookup2 → Copy data1

General Source **Sink** Mapping Settings User properties

Sink dataset *

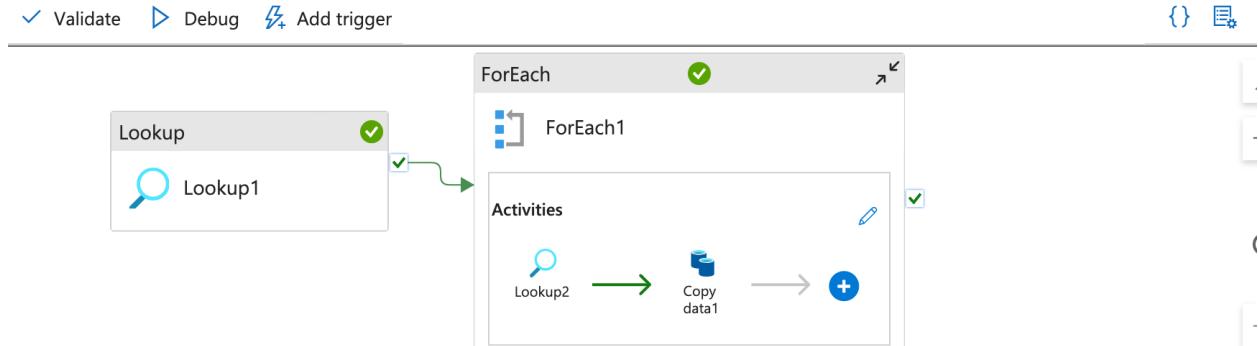
Dataset properties

Name	Type
folder_name	string
file_name	string

@item().Table_name

@concat(item().Table_name, '_', utcN...

pipeline ran successfully



Parameters Variables Settings **Output**

Pipeline run ID: 5b326e57-13ff-4cce-948f-15cd3cb4bea7 ⏪ ⏴ ⓘ 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, 9:02:25 PM	22s	AutoResolveIntegrationRunti
Copy data1	✓ Succeeded	Copy data	5/7/2025, 9:02:17 PM	14s	AutoResolveIntegrationRunti

Home > shubhaadis | Containers >

project5 ...

Search Upload + Add Directory Refresh Rename Delete Change tier Acquire lease Break lease Give feedback

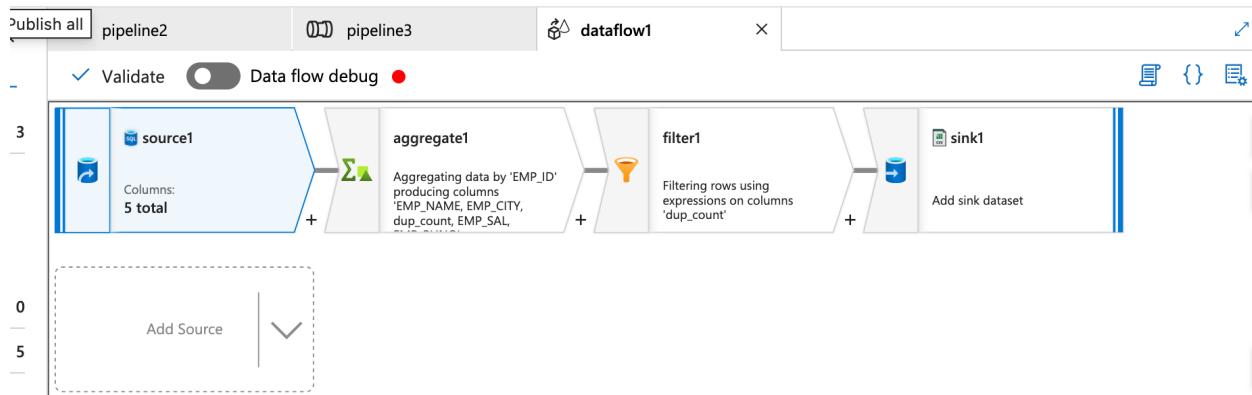
Overview

Authentication method: Access key (Switch to Microsoft Entra user account)
Location: project5

Search blobs by prefix (case-sensitive)

Name	Modified	Access tier	Archive status	Blob type	Size
COST	5/7/2025, 9:02:29 PM				
CUST	5/7/2025, 9:02:20 PM				
EMPLOYEE	5/7/2025, 9:02:38 PM				
ORDERS	5/7/2025, 9:02:21 PM				
PEOPLE	5/7/2025, 9:02:20 PM				

Create a pipeline to clean the data using Dataflows.



pipeline3 X dataflow1

Validate Debug Add trigger Data flow debug

Data flow On skip

Parameters Variables Settings Output

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

Pipeline status Succeeded

All status

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

pipeline2 dataflow1 pipeline3 AzureSqlTable1

Azure SQL Database
AzureSqlTable1

Connection Schema Parameters

Linked service * AzureSqlDatabase1 Test connection Edit New Learn more

Integration runtime * AutoResolveIntegrationRuntime Edit

Table @dataset().schema_name . @dataset().table_name Preview data

Enter manually

AzureSqlTable2

SQL 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 CSV 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)

AzureSqlTable2 DelimitedText1 DelimitedText3 DelimitedText2

 DelimitedText
DelimitedText3

Connection Schema Parameters

Linked service * AzureDataLakeStorage1 Test connection Edit New Learn more

Integration runtime * AutoResolveIntegrationRuntime Edit

File path project / Directory / File name Browse Preview

Compression type No compression

Column delimiter (,) Comma (,)

Row delimiter (Default (\r,\n, or \r\n)) Default (\r,\n, or \r\n)

now copy the json from ADF and paste it in synapse

Pipeline name pipeline2

 Copy to clipboard

```

1  {
2    "name": "pipeline2",
3    "properties": {
4      "activities": [
5        {
6          "name": "Lookup1",
7          "type": "Lookup",
8          "dependsOn": [],
9          "policy": {
10            "timeout": "0.12:00:00",
11            "retry": 0,
12            "retryIntervalInSeconds": 30,
13            "secureOutput": false,
14            "secureInput": false
15          },
16          "userProperties": [],
17          "typeProperties": {
18            "source": {
19              "type": "AzureSqlSource",
20              "queryTimeout": "02:00:00",
21              "partitionOption": "None"
22            }
23          }
24        }
25      ]
26    }
27  }

```

OK Cancel

pipeline2

Validate Debug Add trigger

ForEach

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	AutoResolveIntegrationRuntime

Pipeline name pipeline3

```

1 {
2   "name": "pipeline3",
3   "properties": {
4     "activities": [
5       {
6         "name": "Data flow1",
7         "type": "ExecuteDataFlow",
8         "dependsOn": [],
9         "policy": {
10           "timeout": "0.12:00:00",
11           "retry": 0,
12           "retryIntervalInSeconds": 30,
13           "secureOutput": false,
14           "secureInput": false
15         },
16         "userProperties": [],
17         "typeProperties": {
18           "dataflow": {
19             "referenceName": "dataflow1",
20             "type": "DataFlowReference"
21           }
22         }
23       }
24     ]
25   }
26 }
```

OK Cancel

Pipeline run ID: a61c0f8c-fd0c-4668-b4a8-93afc383fab9

Pipeline status: Succeeded

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime
Data flow1	Succeeded	Data flow	5/7/2025, 11:09:43 PM	2m 5s	AutoResolveIntegrationRuntime

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

Task2

Migrating Databricks notebook to fabric

```

from pyspark.sql import SparkSession
data = [(“C001”, “Alice”, “alice@example.com”), (“C002”, “Bob”, “bob@example.com”)]
columns = [“CustomerID”, “Name”, “Email”]
df = spark.createDataFrame(data, columns)
df.show()

```

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

Notebook4 Python ▾ Tabs: OFF ☆

File Edit View Run Help Last edit was 12 hours ago

▶ Run all ⚙ Connect Schema

Python 🗑

```
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, StringType, IntegerType

# Create Spark session
spark = SparkSession.builder.appName("ReadCustomerData").getOrCreate()

# Define sample data
data = [
    (1, "Alice", "alice@example.com"),
    (2, "Bob", "bob@example.com"),
    (3, "Charlie", "charlie@example.com")
]

# Define schema
schema = StructType([
    StructField("CustomerID", IntegerType(), True),
    StructField("Name", StringType(), True),
    StructField("Email", StringType(), True)
])

# Create DataFrame
df = spark.createDataFrame(data, schema=schema)

# Show output
df.show()

> [See performance (1)]
```

▶ df: pyspark.sql.connect.DataFrame = [CustomerID: integer, Name: string ... 1 more field]

CustomerID	Name	Email
1	Alice	alice@example.com
2	Bob	bob@example.com
3	Charlie	charlie@example.com

now download all these 4 notebooks in python format

The screenshot shows a Jupyter Notebook interface with the title "Notebook4" and Python selected as the kernel. The notebook has one cell containing the following code:

```
connect.DataFrame = [CustomerID: integer, Name: string ... 1 more field]
+-----+
| Email |
+-----+
| alice@example.com |
| bob@example.com |
| charlie@example.com |
+-----+
```

A context menu is open over the second cell, with the "Export" option highlighted. The export options include "DBC archive", "Source file", "IPython Notebook", and "HTML".

now lets import these downloaded exported ython otebooks to fabricks

My workspace → import →import notebook

The screenshot shows the "Import" dialog in Jupyter Notebook. The "From this computer" tab is selected. The dialog lists four local files: "Notebook1.ipynb", "Notebook2.ipynb", "Notebook3.ipynb", and "Notebook4.ipynb". The left sidebar shows the user's workspace structure.

Notebook1 | Saved | Search Trial: 57 days left 69 Comments History Develop Share

Home Edit AI tools Run View 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 69 Comments History Develop

Home Edit AI tools Run View 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 Sh

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]

```
... +-----+-----+
|ProductID| AOV|
+-----+-----+
| P001|125.0|
| P002|200.0|
+-----+-----+
```

PySpark (Python)

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, 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
23 df = spark.createDataFrame(data, schema=schema)
24
25 # Show output
26 df.show()
```

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()
```

[0]

```
... +-----+-----+
|CustomerID| Name| Email|
+-----+-----+
| 1| Alice| alice@example.com|
| 2| Bob| bob@example.com|
| 3| Charlie| charlie@example.com|
+-----+-----+
```

PySpark (Python)