Project Assignment: Data Pipeline for Customer Account Analysis

This document outlines a data pipeline for processing customer account data, including copying data from a backend team's storage account, performing transformations in Databricks, and exposing data for analysis in Azure Synapse Analytics.

Step 1: Data Ingestion (Backend Storage to Raw(Bronze) Container)

Data ingestion pipeline is a crucial component of modern data architecture, enabling businesses to efficiently manage and utilize their data. It's the process of importing, transferring, loading, and processing data for later use or storage in a database.

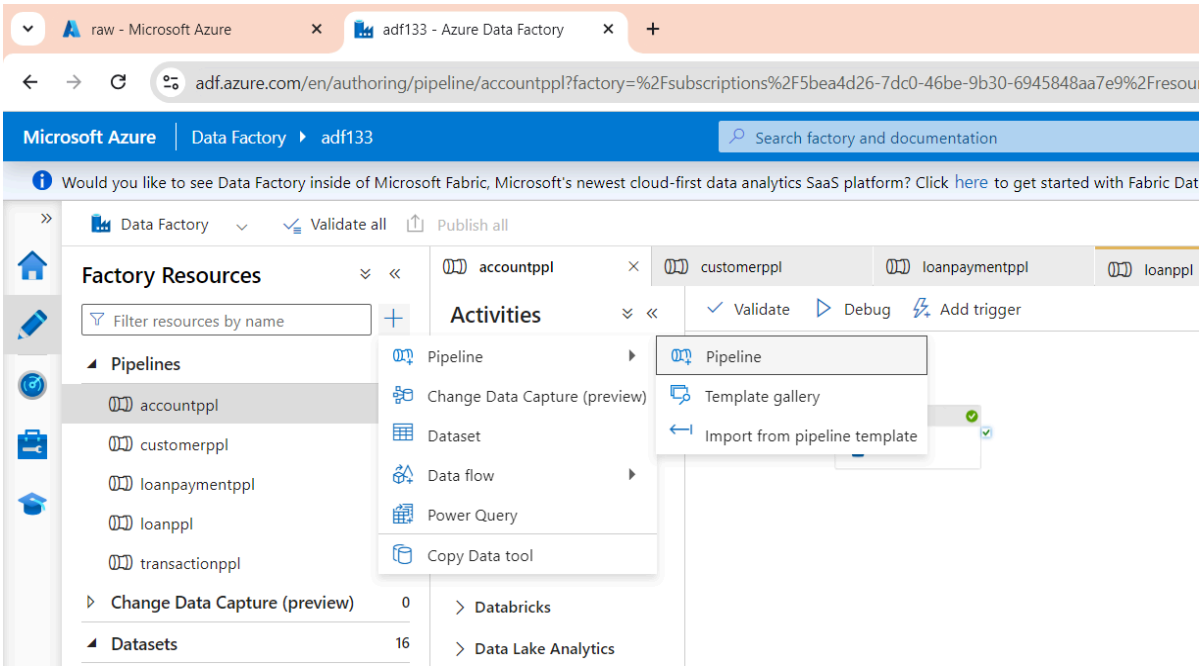
1.1. Configure Azure Data Factory (ADF) for Data Copy

1. **Sign in to the Azure Portal**:
    - Navigate to the Azure Portal.
2. **Create an Azure Data Factory Instance**:
    - In the Azure Portal, search for **"Data Factory"** and click **"Create"**.

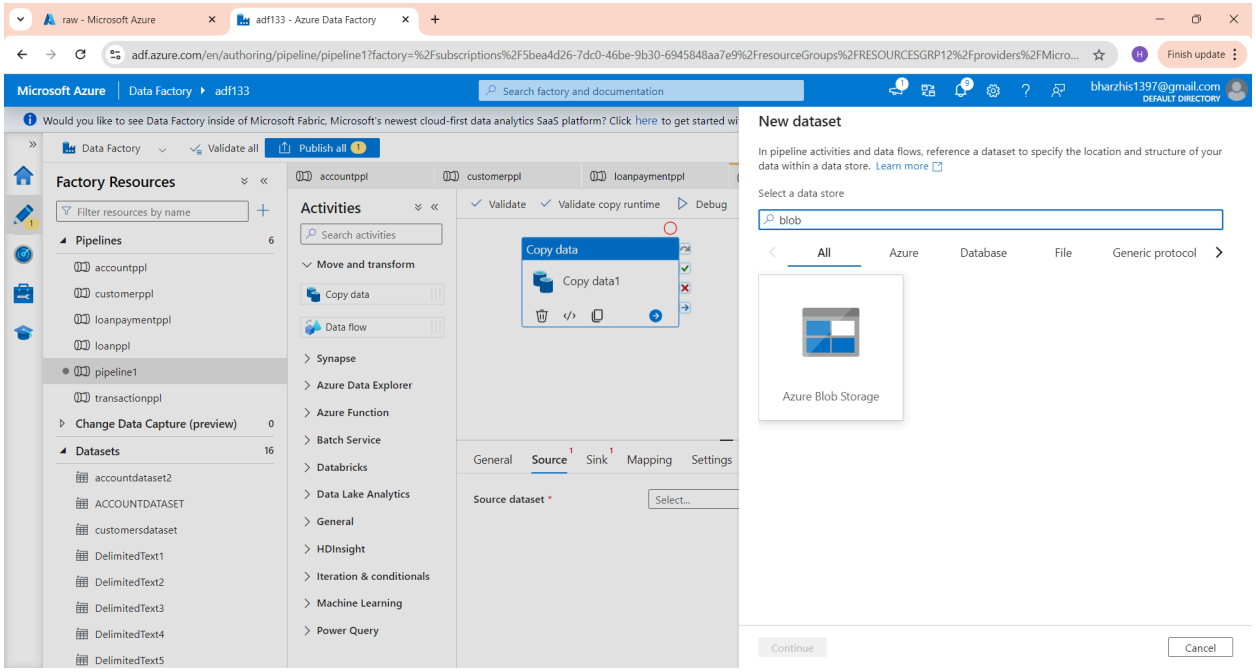
    - Fill in the necessary details, such as:
        - **Resource Group**: Select an existing group or create a new one.
        - **Name**: Provide a unique name for your Data Factory instance.
        - **Region**: Choose the region where you want to deploy.
    - Click **"Review + Create"** and then **"Create"**.
3. **Set Up Linked Services**:
    - In the Data Factory, go to **Manage** > **Linked Services** > **New**.
    - **Create a Linked Service for Backend Storage**:
        - Choose **Azure Blob Storage** as the data store.
        - Enter the storage account details of the backend team's storage account.
        - Use either **Account Key** or **SAS token** for authentication.
    - **Create a Linked Service for Your Data Lake Storage**:
        - Repeat the steps above for your own Data Lake Storage account.

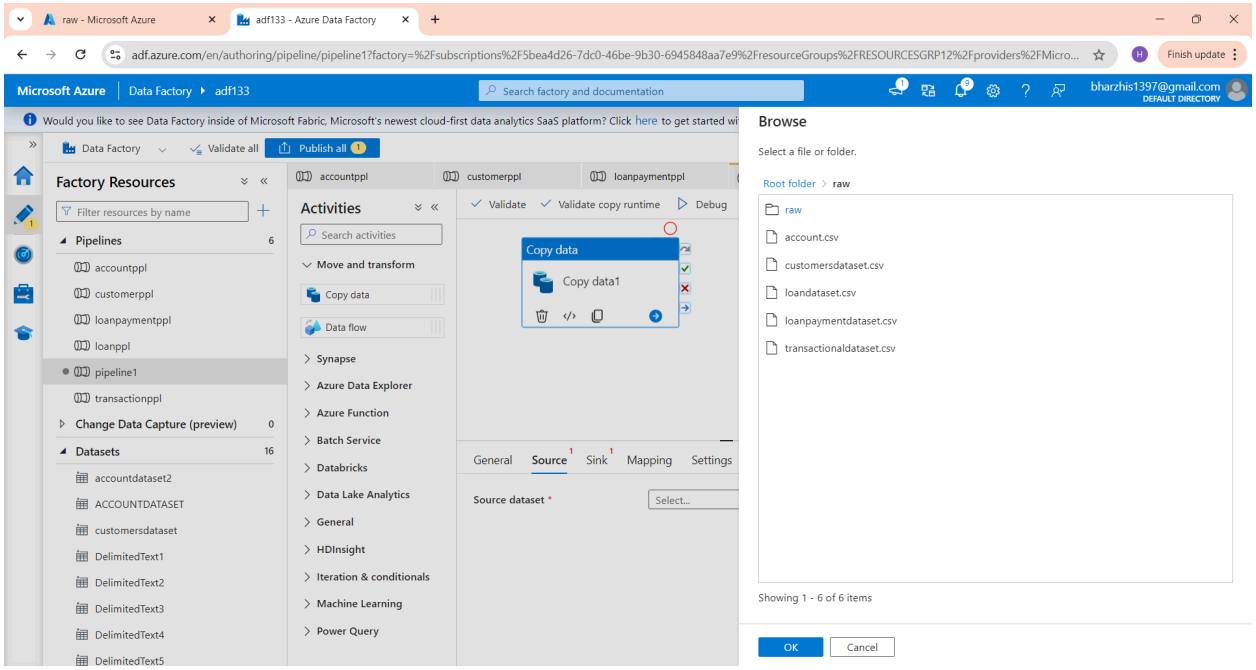
1.2. Create a Data Factory Pipeline

1. **Navigate to the Author Tab**:
    - Click **"Author"** > **"+"** > **"Pipeline"**.

- 

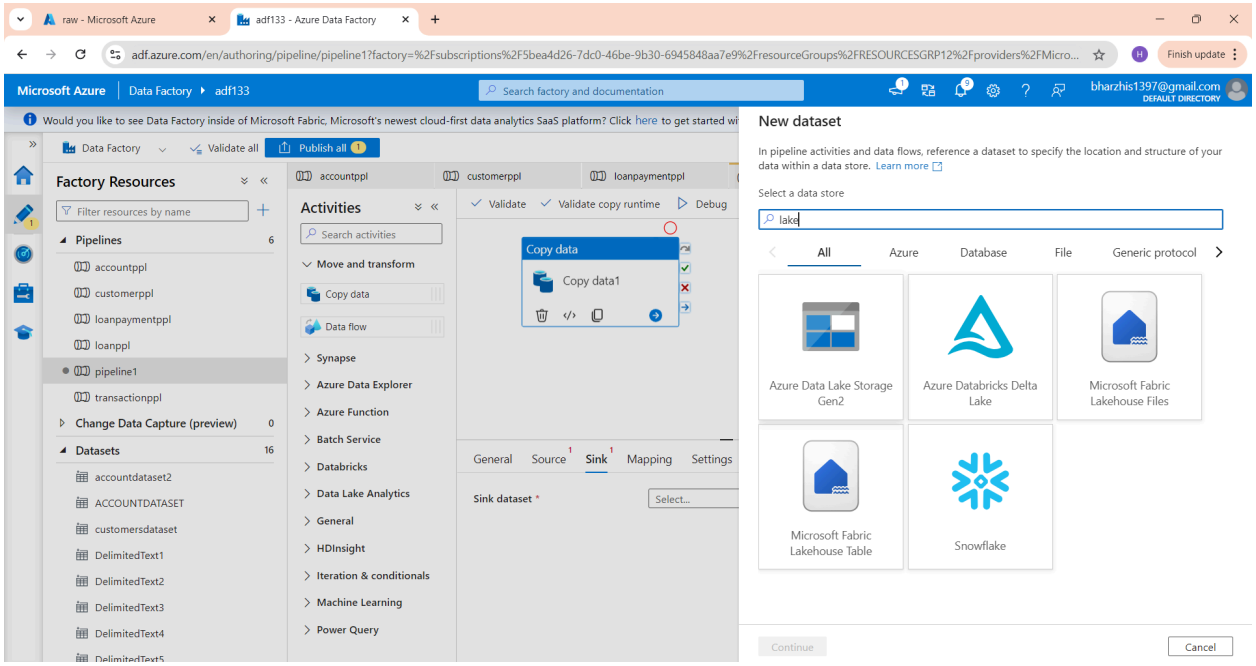
2. **Add a Copy Data Activity**:
   - Drag and drop the **Copy Data** activity onto the canvas.
   - **Configure the Source**:
     - Select **Source** in the Copy Data activity settings.
     - Click **New** to add a dataset pointing to the backend storage account.
     - Choose **DelimitedText** for CSV files.
     - Specify the path for each file:
       - `accounts.csv`
       - `customers.csv`
       - `loan_payments.csv`
       - `loans.csv`
       - `transactions.csv`



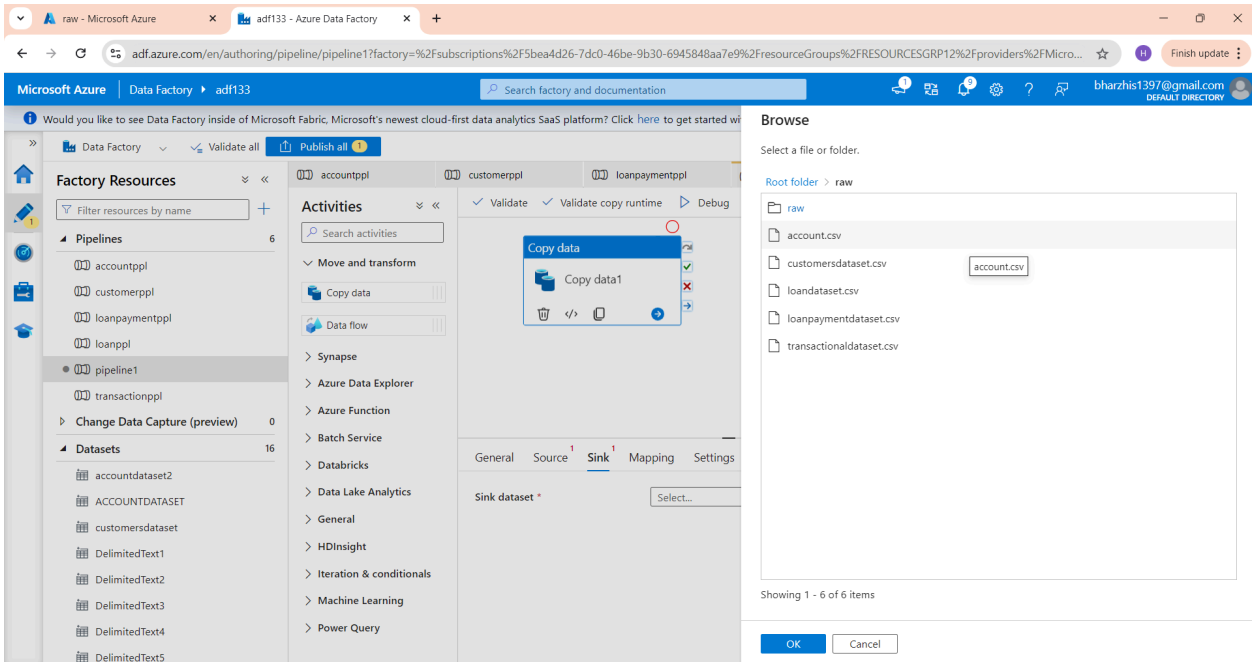


- **Configure the Sink**:

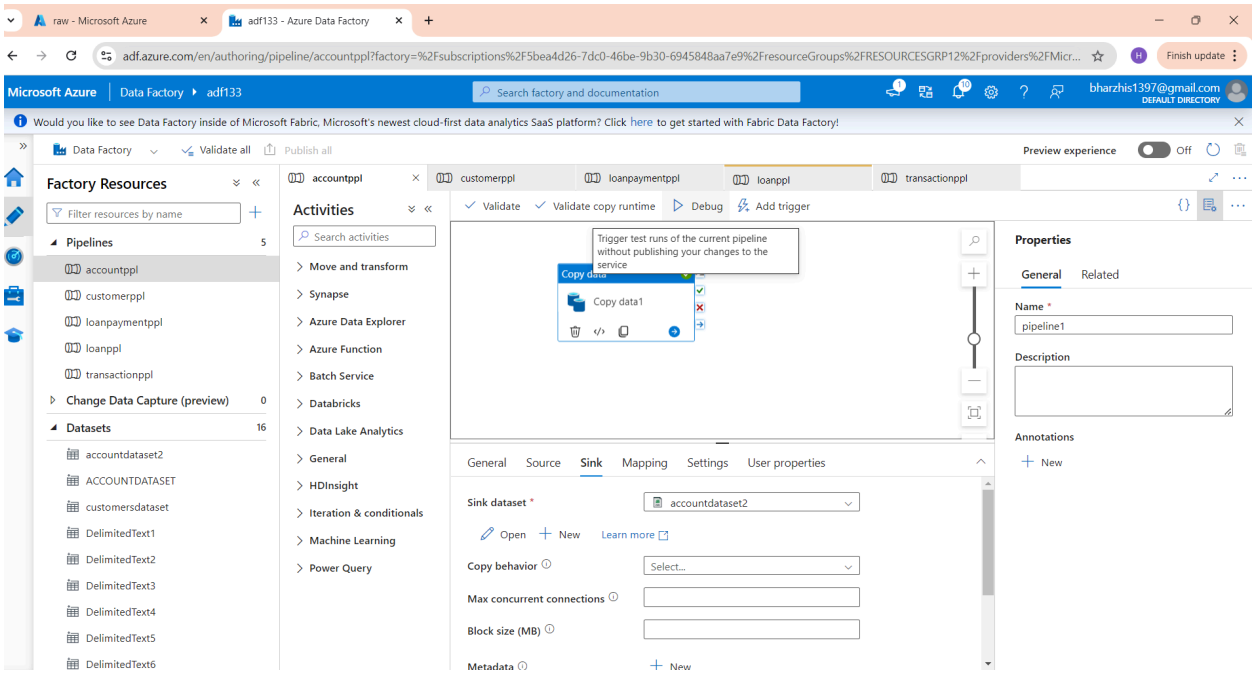
- Select **Sink** in the Copy Data activity settings.
- Click **New** to add a dataset pointing to your Data Lake Storage Raw (Bronze) container.



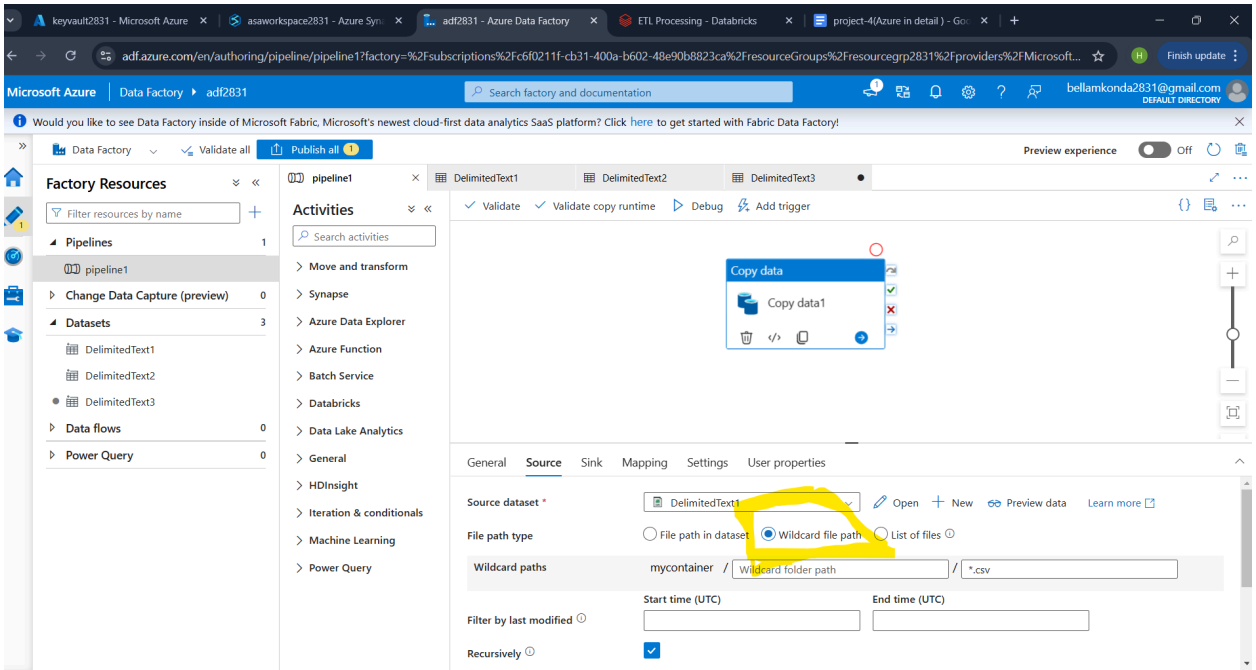
- Specify the destination paths (e.g., `raw/accounts.csv`, `raw/customers.csv`).



3. **Set Up Parameters (Optional for Dynamic Configurations)**:
   - Use **parameters** to define file paths, making your pipeline flexible and easily configurable.
4. **Debug and Publish**:
   - Click **Debug** to test the pipeline.
   - If successful, click **Publish All** to save your pipeline.



Note: If you would like to push N number of files into the pipeline, you can achieve it by the option called ”wildcard file path” in the source tab as shown in the below screenshot.



Step 2: Databricks Activity (Incremental/Delta Processing)

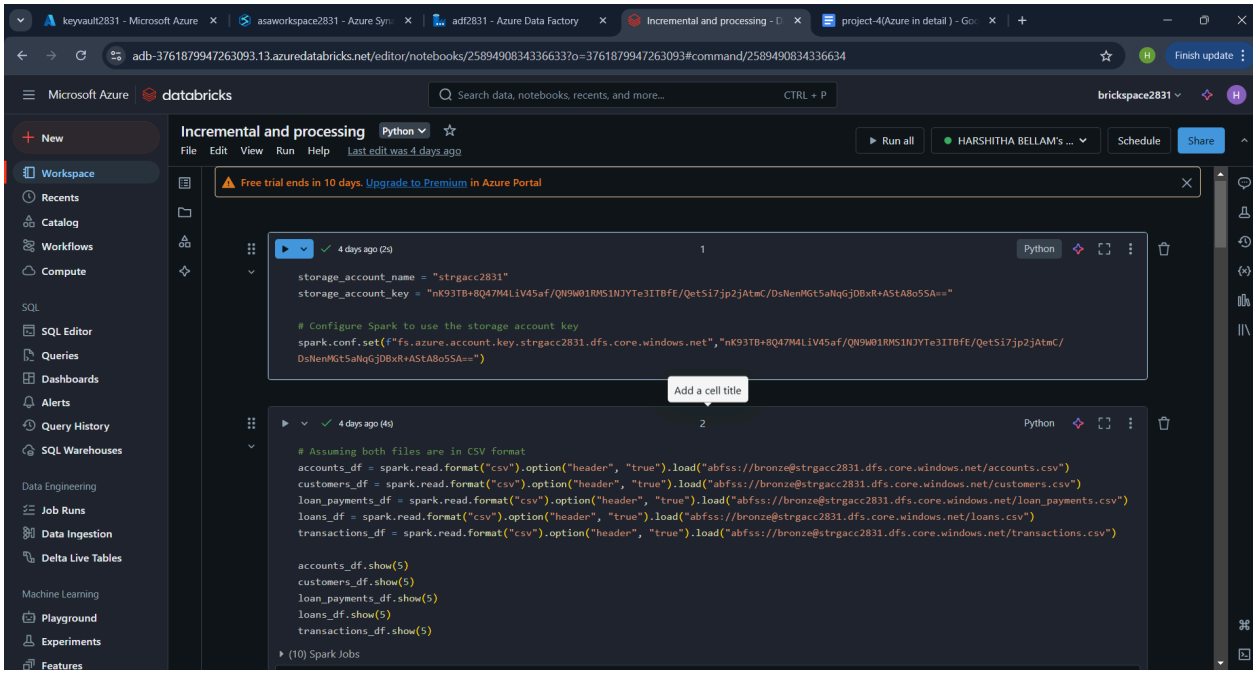
2.1. Set Up Databricks

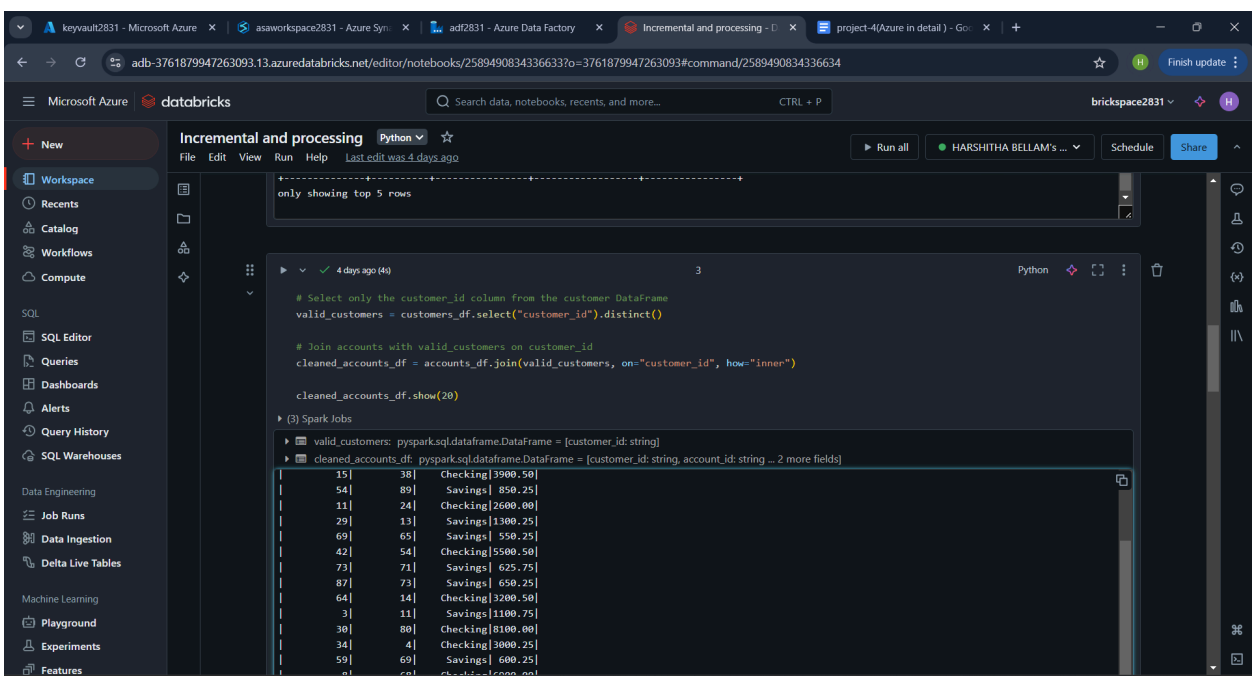
Incremental and delta processing in Databricks allows for the processing of data in a way that is more efficient and cost-effective than repeated batch jobs.

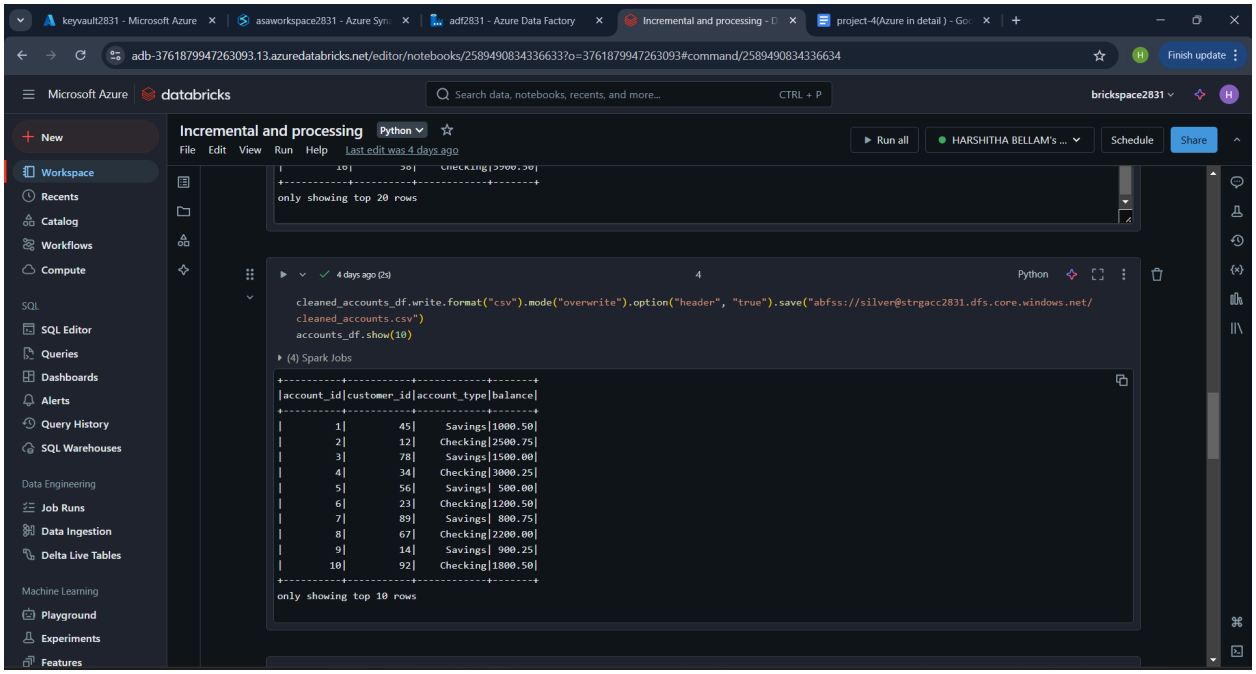
1. **Create an Azure Databricks Workspace**:
    - In the Azure Portal, search for **"Azure Databricks"** and click **"Create"**.
    - Provide the required details and click **"Review + Create"**, then **"Create"**.
    - Once created, launch the **Databricks workspace**.
2. **Create a Databricks Cluster**:
    - In the Databricks workspace, go to **Clusters** > **Create Cluster**.
    - Configure the cluster settings (e.g., name, node types) and create

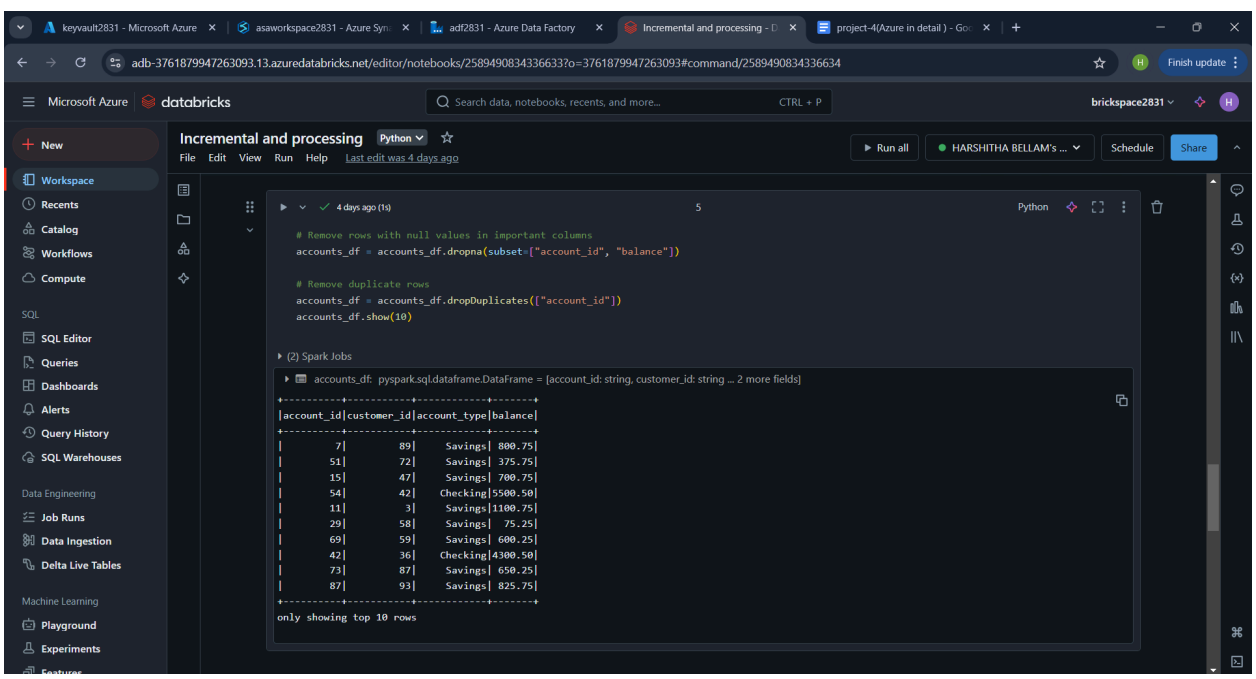
2.2. Create a Databricks Notebook for Incremental Processing

1. **Set Up a Notebook**:
    - In Databricks, click **"Create"** > **"Notebook"**.
    - Name the notebook (e.g., `Incremental_Processing`).
    - Choose **Language** as PySpark.
2. **Read Data from Raw (Bronze) Container**:


keyvault2831 - Microsoft Azure
asaworkspace2831 - Azure Syn
adf2831 - Azure Data Factory
Incremental and processing - D
project-4(Azure in detail) - Goo
adb-3761879947263093.13.azuredatabricks.net/editor/notebooks/2589490834336633?o=3761879947263093#command/2589490834336634
Finish update
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
brickspace2831
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Playground
Experiments
Features
Incremental and processing
Python
File Edit View Run Help
Last edit was 4 days ago
Run all
HARSHITHA BELLAM's ...
Schedule
Share
Free trial ends in 10 days. Upgrade to Premium in Azure Portal
4 days ago (2s)
1
storage_account_name = "strgacc2831"
storage_account_key = "nK93TB+8Q47M4LiV45af/QN9W01RMS1NJYTe3ITBfE/QetSi7jp2jAtmC/DsNenMGt5aNqGjDBxR+AStA8o5SA==="
# Configure Spark to use the storage account key
spark.conf.set(f"fs.azure.account.key.strgacc2831.dfs.core.windows.net","nK93TB+8Q47M4LiV45af/QN9W01RMS1NJYTe3ITBfE/QetSi7jp2jAtmC/
DsNenMGt5aNqGjDBxR+AStA8o5SA==")
Add a cell title
4 days ago (4s)
2
# Assuming both files are in CSV format
accounts_df = spark.read.format("csv").option("header", "true").load("abfss://bronze@strgacc2831.dfs.core.windows.net/accounts.csv")
customers_df = spark.read.format("csv").option("header", "true").load("abfss://bronze@strgacc2831.dfs.core.windows.net/customers.csv")
loan_payments_df = spark.read.format("csv").option("header", "true").load("abfss://bronze@strgacc2831.dfs.core.windows.net/loan_payments.csv")
loans_df = spark.read.format("csv").option("header", "true").load("abfss://bronze@strgacc2831.dfs.core.windows.net/loans.csv")
transactions_df = spark.read.format("csv").option("header", "true").load("abfss://bronze@strgacc2831.dfs.core.windows.net/transactions.csv")
accounts_df.show(5)
customers_df.show(5)
loan_payments_df.show(5)
loans_df.show(5)
transactions_df.show(5)
(10) Spark Jobs


keyvault2831 - Microsoft Azure
asaworkspace2831 - Azure Syn
adf2831 - Azure Data Factory
Incremental and processing - D
project-4(Azure in detail) - Goo
adb-3761879947263093.13.azuredatabricks.net/editor/notebooks/2589490834336633?o=3761879947263093#command/2589490834336634
Finish update
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
brickspace2831
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Playground
Experiments
Features
Incremental and processing
Python
File Edit View Run Help
Last edit was 4 days ago
Run all
HARSHITHA BELLAM's ...
Schedule
Share
only showing top 5 rows
4 days ago (4s)
3
# Select only the customer_id column from the customer DataFrame
valid_customers = customers_df.select("customer_id").distinct()
# Join accounts with valid_customers on customer_id
cleaned_accounts_df = accounts_df.join(valid_customers, on="customer_id", how="inner")
cleaned_accounts_df.show(20)
(3) Spark Jobs
valid_customers: pyspark.sql.dataframe.DataFrame = [customer_id: string]
cleaned_accounts_df: pyspark.sql.dataframe.DataFrame = [customer_id: string, account_id: string ... 2 more fields]
15	38	Checking	3900.50
54	89	Savings	850.25
11	24	Checking	2600.00
29	13	Savings	1300.25
69	65	Savings	550.25
42	54	Checking	5500.50
73	71	Savings	625.75
87	73	Savings	650.25
64	14	Checking	3200.50
3	11	Savings	1100.75
30	80	Checking	8100.00
34	4	Checking	3000.25
59	69	Savings	600.25


keyvault2831 - Microsoft Azure
asaworkspace2831 - Azure Syn
adf2831 - Azure Data Factory
Incremental and processing - D
project-4(Azure in detail) - Goo
adb-3761879947263093.13.azuredatabricks.net/editor/notebooks/2589490834336633?o=3761879947263093#command/2589490834336634
Finish update
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
brickspace2831
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Playground
Experiments
Features
Incremental and processing
Python
File
Edit
View
Run
Help
Last edit was 4 days ago
Run all
HARSHITHA BELLAM's ...
Schedule
Share
only showing top 20 rows
4 days ago (2s)
4
Python
cleaned_accounts_df.write.format("csv").mode("overwrite").option("header", "true").save("abfss://silver@strgacc2831.dfs.core.windows.net/cleaned_accounts.csv")
accounts_df.show(10)
(4) Spark Jobs
+----------+-----------+------------+-------+
|account_id|customer_id|account_type|balance|
+----------+-----------+------------+-------+
1	45	Savings	1000.50
2	12	Checking	2500.75
3	78	Savings	1500.00
4	34	Checking	3000.25
5	56	Savings	500.00
6	23	Checking	1200.50
7	89	Savings	800.75
8	67	Checking	2200.00
9	14	Savings	900.25
10	92	Checking	1800.50
+----------+-----------+------------+-------+
only showing top 10 rows


keyvault2831 - Microsoft Azure
asaworkspace2831 - Azure Syn
adf2831 - Azure Data Factory
Incremental and processing - D
project-4(Azure in detail) - Goo
adb-3761879947263093.13.azuredatabricks.net/editor/notebooks/2589490834336633?o=3761879947263093#command/2589490834336634
Finish update
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
brickspace2831
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Playground
Experiments
Incremental and processing
Python
File
Edit
View
Run
Help
Last edit was 4 days ago
Run all
HARSHITHA BELLAM's ...
Schedule
Share
4 days ago (1s)
5
Python
# Remove rows with null values in important columns
accounts_df = accounts_df.dropna(subset=["account_id", "balance"])
# Remove duplicate rows
accounts_df = accounts_df.dropDuplicates(["account_id"])
accounts_df.show(10)
(2) Spark Jobs
accounts_df: pyspark.sql.dataframe.DataFrame = [account_id: string, customer_id: string ... 2 more fields]
+----------+-----------+------------+-------+
|account_id|customer_id|account_type|balance|
+----------+-----------+------------+-------+
7	89	Savings	800.75
51	72	Savings	375.75
15	47	Savings	700.75
54	42	Checking	5500.50
11	3	Savings	1100.75
29	58	Savings	75.25
69	59	Savings	600.25
42	36	Checking	4300.50
73	87	Savings	650.25
87	93	Savings	825.75
+----------+-----------+------------+-------+
only showing top 10 rows

keyvault2831 - Microsoft Azure | asaworkspace2831 - Azure Syn | adf2831 - Azure Data Factory | Incremental and processing - D | project-4(Azure in detail) - Go

adb-3761879947263093.13.azuredatabricks.net/editor/notebooks/2589490834336633?o=3761879947263093#command/2589490834336634

Microsoft Azure | databricks | Search data, notebooks, recents, and more... CTRL + P | brickspace2831

New | Workspace | Recents | Catalog | Workflows | Compute | SQL | SQL Editor | Queries | Dashboards | Alerts | Query History | SQL Warehouses | Data Engineering | Job Runs | Data Ingestion | Delta Live Tables | Machine Learning | Playground | Experiments | Features

Incremental and processing | Python | File Edit View Run Help | Last edit was 4 days ago | Run all | HARSHITHA BELLAM's ... | Schedule | Share

4 days ago (1s) 6 Python

```python
# Filter out accounts with negative balances if not required
accounts_df = accounts_df.filter(accounts_df.balance >= 0)
accounts_df.show(10)
```

(2) Spark Jobs

accounts_df: pyspark.sql.dataframe.DataFrame = [account_id: string, customer_id: string ... 2 more fields]

```
+----------+-----------+------------+-------+
|account_id|customer_id|account_type|balance|
+----------+-----------+------------+-------+
|         7|         89|     Savings| 800.75|
|        51|         72|     Savings| 375.75|
|        15|         47|     Savings| 700.75|
|        54|         42|    Checking|5500.50|
|        11|          3|     Savings|1100.75|
|        29|         58|     Savings|  75.25|
|        69|         59|     Savings| 600.25|
|        42|         36|    Checking|4300.50|
|        73|         87|     Savings| 650.25|
|        87|         93|     Savings| 825.75|
+----------+-----------+------------+-------+
only showing top 10 rows
```

4 days ago (7s) 7

```python
# Save cleaned data as Delta format for efficient storage
```

keyvault2831 - Microsoft Azure | asaworkspace2831 - Azure Syn | adf2831 - Azure Data Factory | Incremental and processing - D | project-4(Azure in detail) - Go

adb-3761879947263093.13.azuredatabricks.net/editor/notebooks/2589490834336633?o=3761879947263093#command/2589490834336634

Microsoft Azure | databricks | Search data, notebooks, recents, and more... CTRL + P | brickspace2831

New | Workspace | Recents | Catalog | Workflows | Compute | SQL | SQL Editor | Queries | Dashboards | Alerts | Query History | SQL Warehouses | Data Engineering | Job Runs | Data Ingestion | Delta Live Tables | Machine Learning | Playground | Experiments | Features

Incremental and processing | Python | File Edit View Run Help | Last edit was 4 days ago | Run all | HARSHITHA BELLAM's ... | Schedule | Share

```
+----------+-----------+------------+-------+
|         7|         89|     Savings| 800.75|
|        51|         72|     Savings| 375.75|
|        15|         47|     Savings| 700.75|
|        54|         42|    Checking|5500.50|
|        11|          3|     Savings|1100.75|
|        29|         58|     Savings|  75.25|
|        69|         59|     Savings| 600.25|
|        42|         36|    Checking|4300.50|
|        73|         87|     Savings| 650.25|
|        87|         93|     Savings| 825.75|
+----------+-----------+------------+-------+
only showing top 10 rows
```

4 days ago (7s) 7

```python
# Save cleaned data as Delta format for efficient storage
accounts_df.write.format("delta").mode("overwrite").save("/mnt/datalake/silver/accounts_cleaned")
customers_df.write.format("delta").mode("overwrite").save("/mnt/datalake/silver/customers_cleaned")
```

(8) Spark Jobs

8

[Shift+Enter] to run and move to next cell

Step 3: Databricks Activity (ETL Processing)

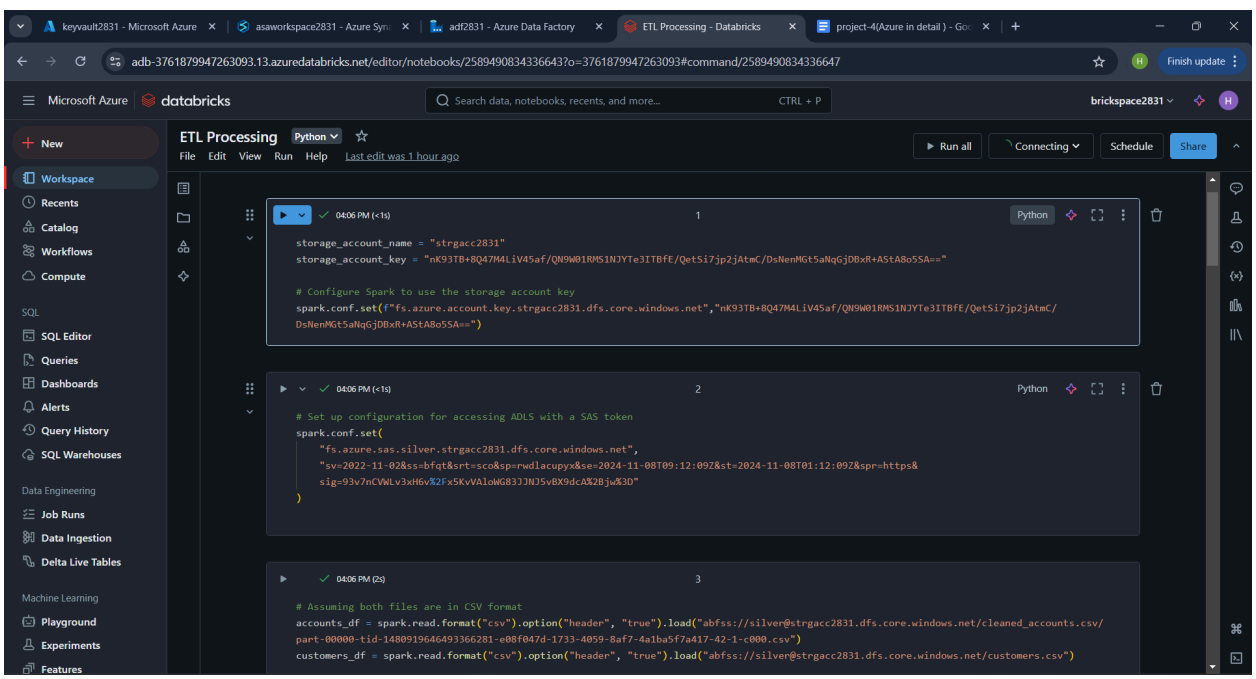
3.1. Create Another Databricks Notebook for ETL

1. **Set Up a New Notebook**:
    - Create another notebook named `ETL_Processing`.
2. **Read Data from Curated (Silver) Container**:

3.2 Transformation Logic

1. **Calculate Total Balance for Each Customer:**

3.3 Save Data to Refined (Gold) Container



keyvault2831 - Microsoft Azure | asaworkspace2831 - Azure Syn | adf2831 - Azure Data Factory | ETL Processing - Databricks | project-4(Azure in detail) - Go

adb-3761879947263093.13.azuredatabricks.net/editor/notebooks/2589490834336643?o=3761879947263093#command/2589490834336647

Microsoft Azure | databricks | Search data, notebooks, recents, and more... CTRL + P | brickspace2831

New | Workspace | Recents | Catalog | Workflows | Compute

SQL: SQL Editor | Queries | Dashboards | Alerts | Query History | SQL Warehouses

Data Engineering: Job Runs | Data Ingestion | Delta Live Tables

Machine Learning: Playground | Experiments | Features

ETL Processing | Python | File Edit View Run Help | Last edit was 1 hour ago | Run all | Connecting | Schedule | Share

04:06 PM (3s) 4 Python

```python
from pyspark.sql import functions as F

# Convert balance column to float if necessary
accounts_df = accounts_df.withColumn("balance", F.col("balance").cast("float"))

# Join customers and accounts DataFrames on customer_id and select specific columns to avoid ambiguity
joined_df = customers_df.alias("cust").join(accounts_df.alias("acct"), F.col("cust.customer_id") == F.col("acct.customer_id"), "inner") \
    .select(
        F.col("cust.customer_id").alias("customer_id"),
        "first_name",
        "last_name",
        "address",
        "city",
        "state",
        "zip",
        "account_type",
        "balance"
    )

# Aggregate to calculate the total balance for each customer
result_df = joined_df.groupBy(
    "customer_id",
    "first_name",
    "last_name",
    "address",
    "city",
    "state",
    "zip",
```

keyvault2831 - Microsoft Azure | asaworkspace2831 - Azure Syn | adf2831 - Azure Data Factory | ETL Processing - Databricks | project-4(Azure in detail) - Go

adb-3761879947263093.13.azuredatabricks.net/editor/notebooks/2589490834336643?o=3761879947263093#command/3863111482200947

Microsoft Azure | databricks | Search data, notebooks, recents, and more... CTRL + P | brickspace2831

New | Workspace | Recents | Catalog | Workflows | Compute

SQL: SQL Editor | Queries | Dashboards | Alerts | Query History | SQL Warehouses

Data Engineering: Job Runs | Data Ingestion | Delta Live Tables

Machine Learning: Playground | Experiments | Features

ETL Processing | Python | File Edit View Run Help | Last edit was 1 hour ago | Run all | Connecting | Schedule | Share

04:06 PM (3s) 4 Python

```python
        "account_type",
        "balance"
    )

# Aggregate to calculate the total balance for each customer
result_df = joined_df.groupBy(
    "customer_id",
    "first_name",
    "last_name",
    "address",
    "city",
    "state",
    "zip",
    "account_type"
).agg(
    F.sum("balance").alias("total_balance")
)
# Display the result
result_df.show()
```

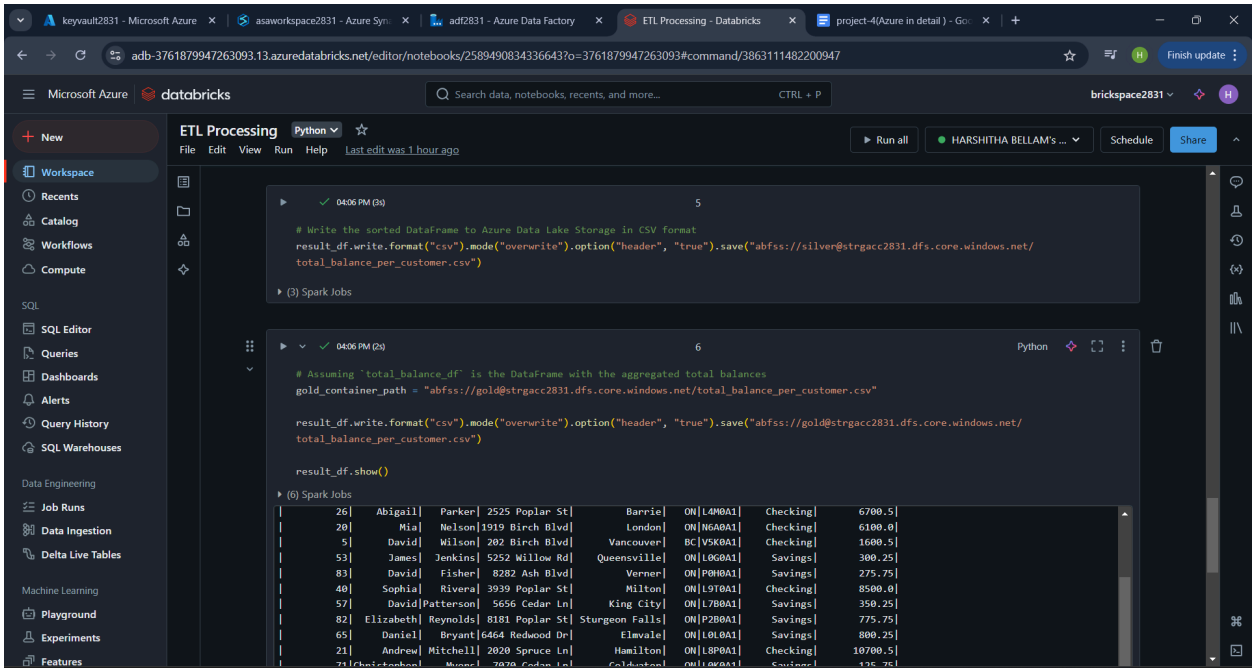
(3) Spark Jobs

accounts_df: pyspark.sql.dataframe.DataFrame = [customer_id: string, account_id: string ... 2 more fields]

joined_df: pyspark.sql.dataframe.DataFrame = [customer_id: string, first_name: string ... 7 more fields]

result_df: pyspark.sql.dataframe.DataFrame = [customer_id: string, first_name: string ... 7 more fields]

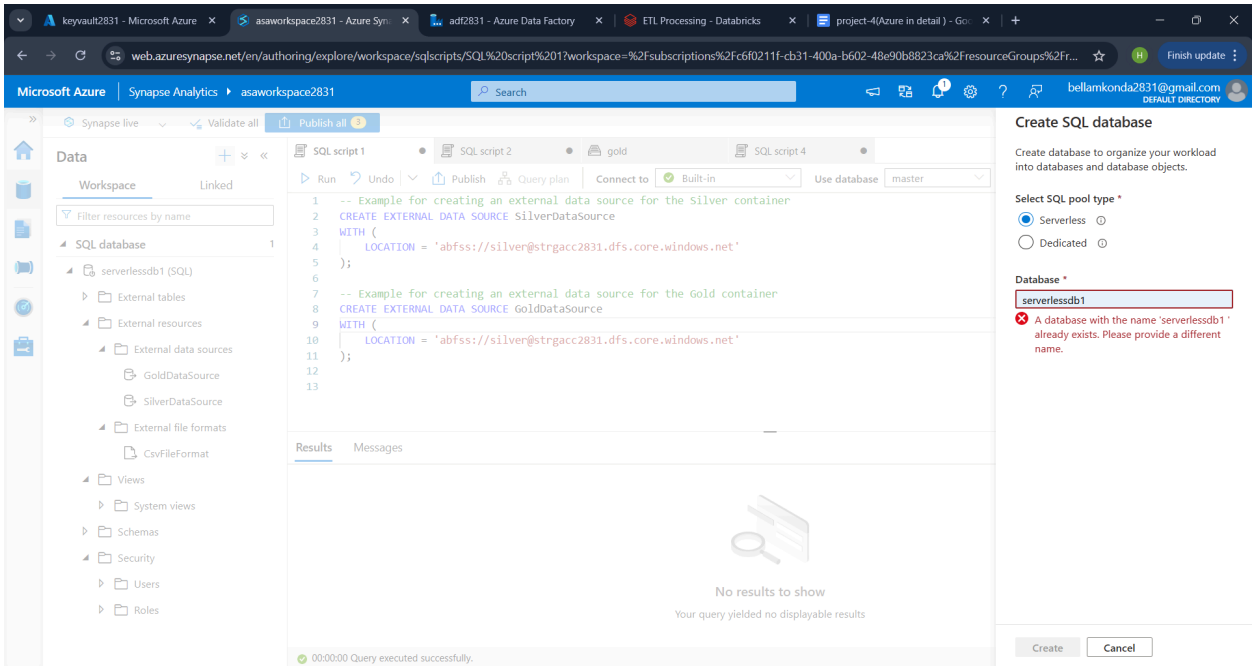
```
|        26|   Abigail|   Parker| 2525 Poplar St|        Barrie|   ON|L4M0A1|   Checking|      6700.5|
|        20|       Mia|   Nelson|1919 Birch Blvd|        London|   ON|N6A0A1|   Checking|      6100.0|
|         5|     David|   Wilson| 202 Birch Blvd|     Vancouver|   BC|V5K0A1|   Checking|      1600.5|
|        53|     James|  Jenkins| 5252 Willow Rd|   Queensville|   ON|L0G0A1|    Savings|      300.25|
|        83|     David|   Fisher|  8282 Ash Blvd|        Verner|   ON|P0H0A1|    Savings|      275.75|
```



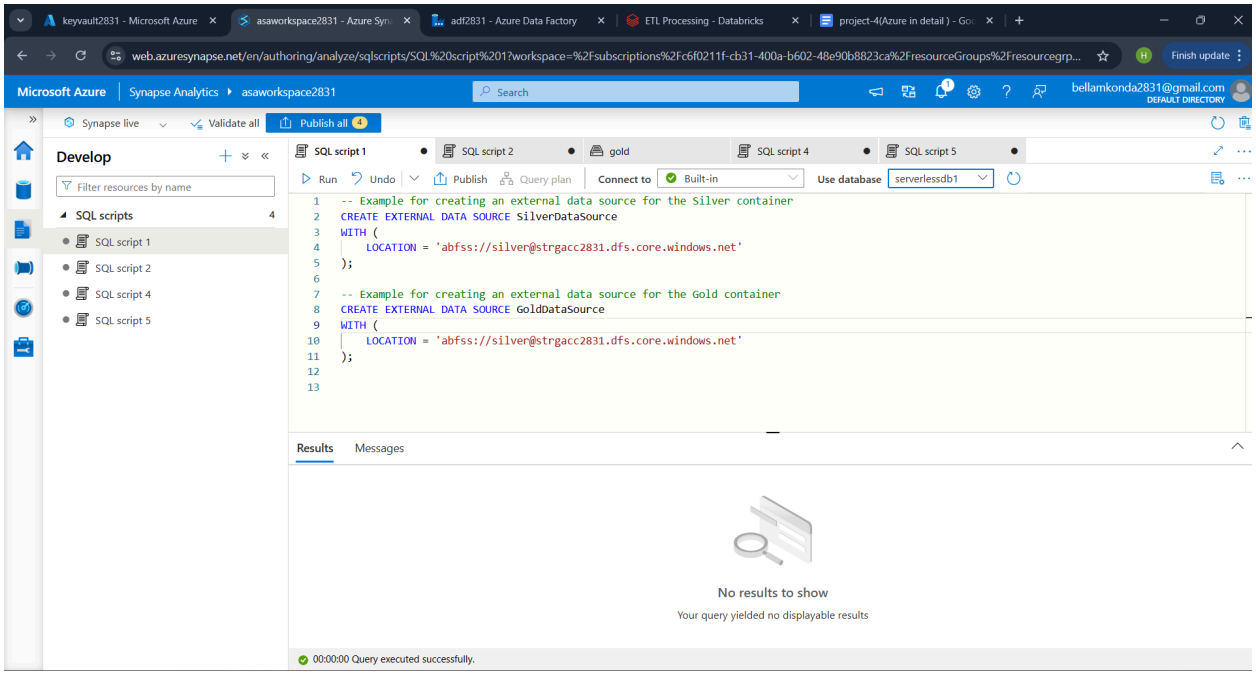
Step 4: Azure Synapse Analytics

Create External Tables in Synapse

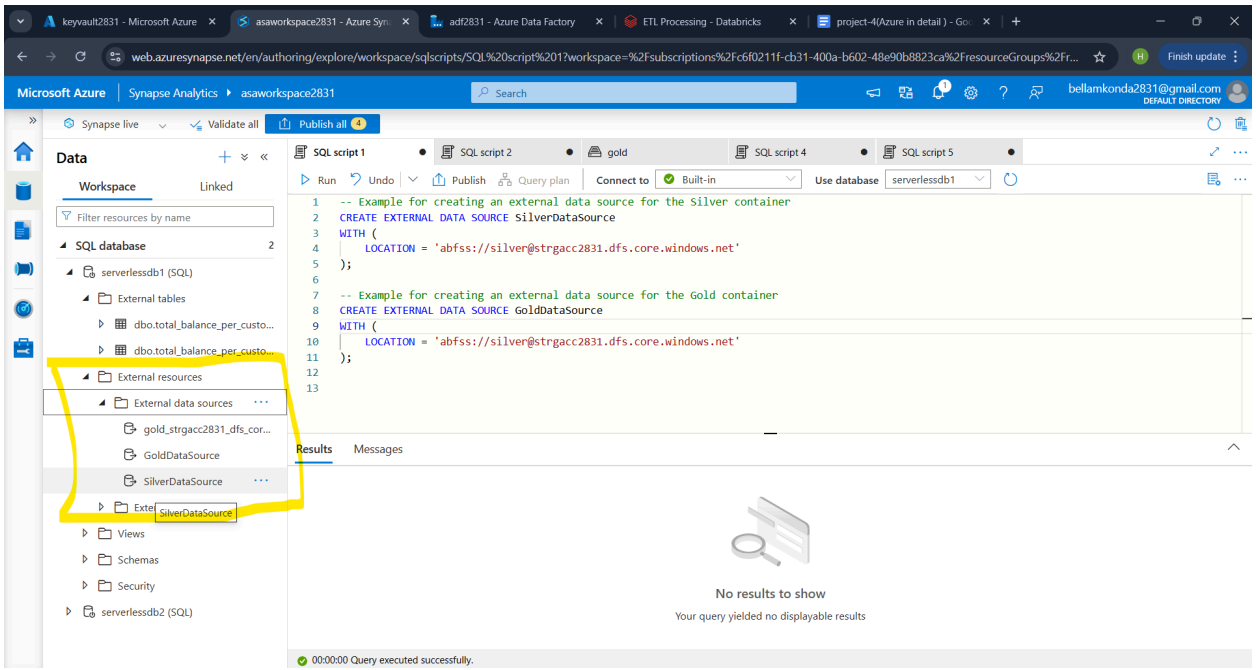
Connect to Synapse Studio and create a SQL Database in ‘Data’ tab



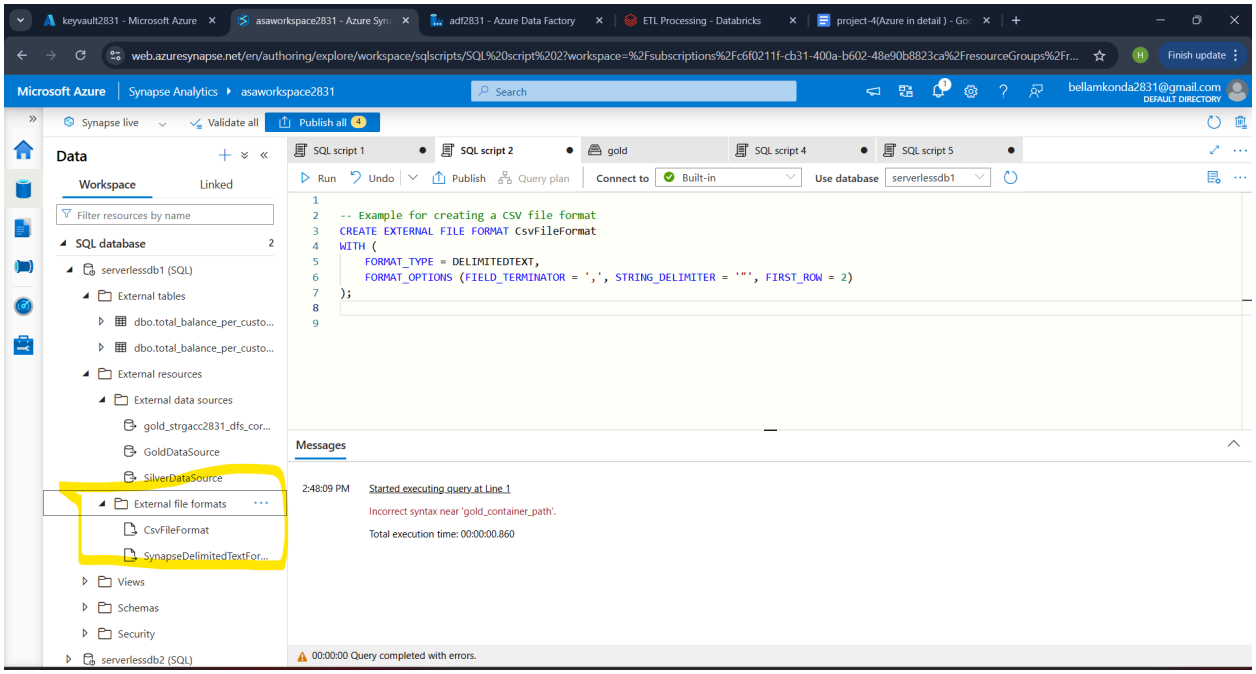
Define External Data Sources for Silver and Gold containers by selecting appropriate database in ‘use database’



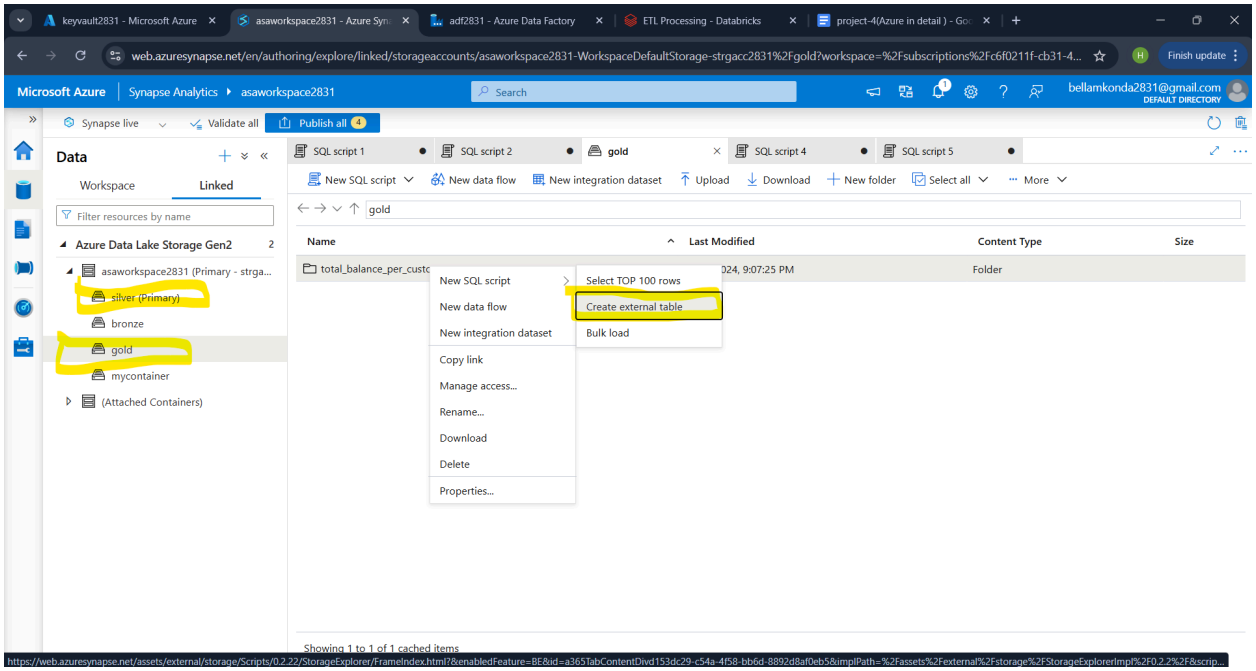
Verify whether the data sources are created for both the containers in the Data>workspace.



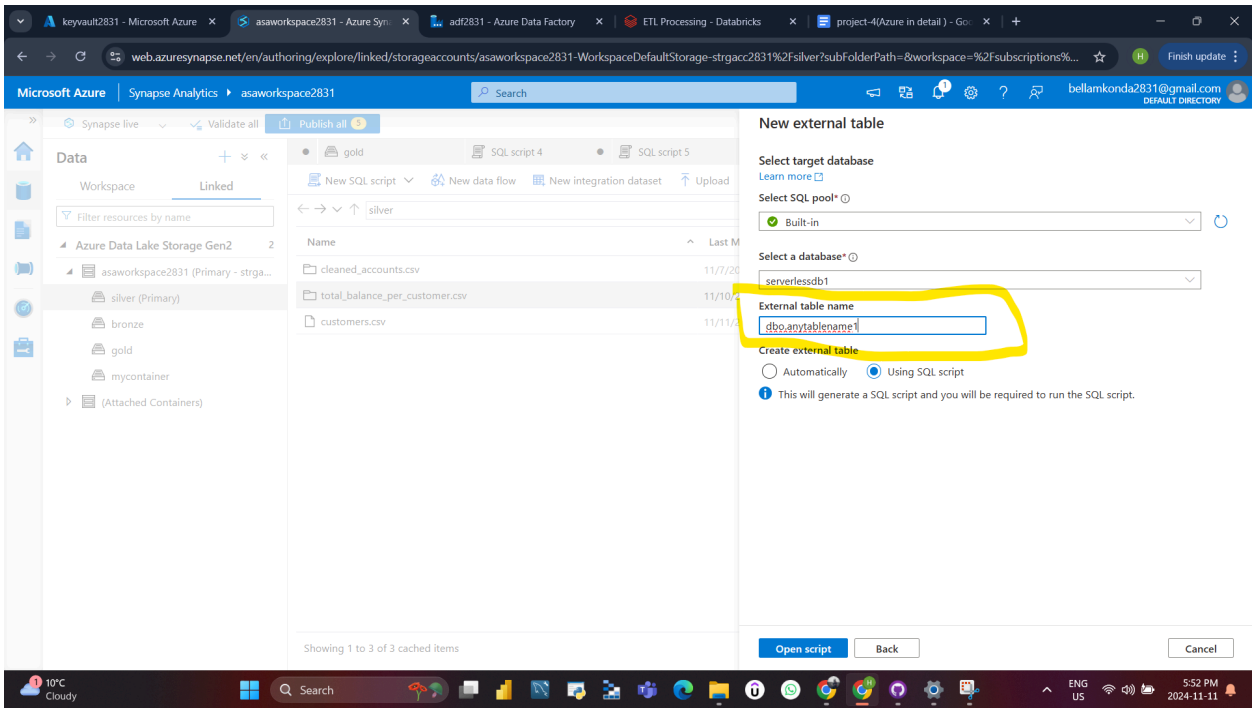
Create an External file format in which you want to share it with the Analyst or scientist teams.

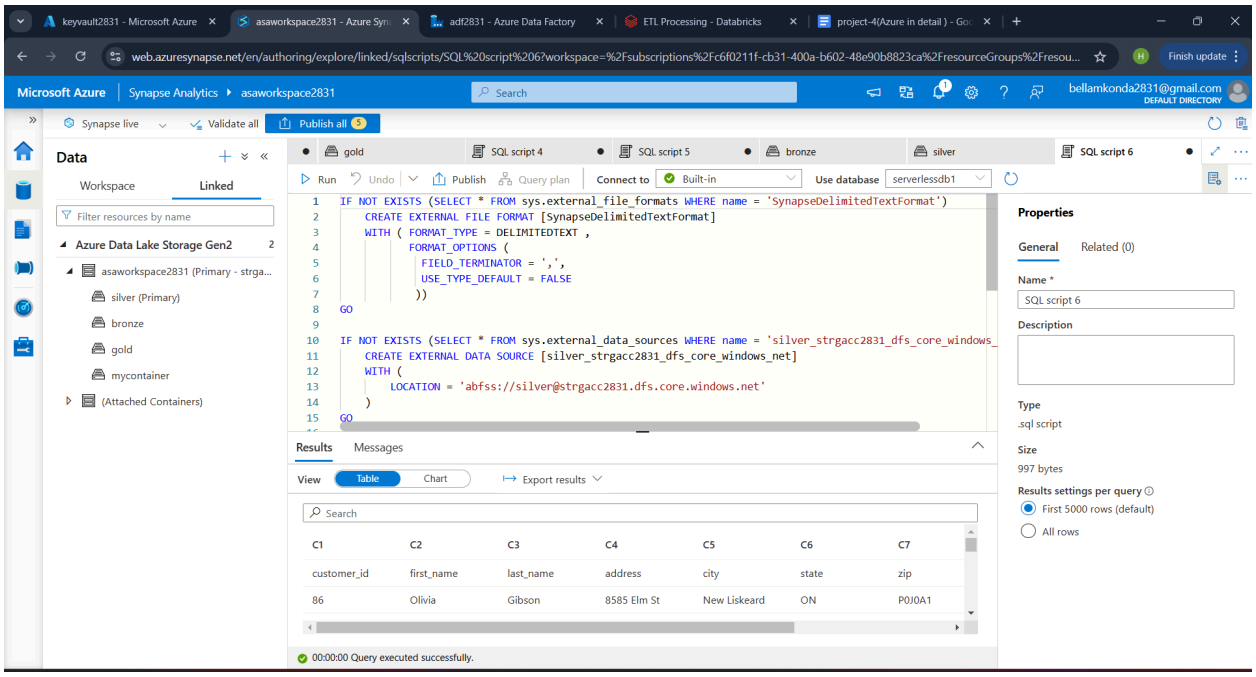


Create External Tables in Synapse for both the curated and refined data. Choose the container in Data>linked for which you wanted to create an external table.



Name the external table name a unique table name followed by schema dbo(default schema name).





This is how we can create an External table with an Automatic code generator.

This allows data analysts and business intelligence teams to access and query the data directly using tools like Synapse Studio or notebooks.