BOOTCAMP PROJECT 2

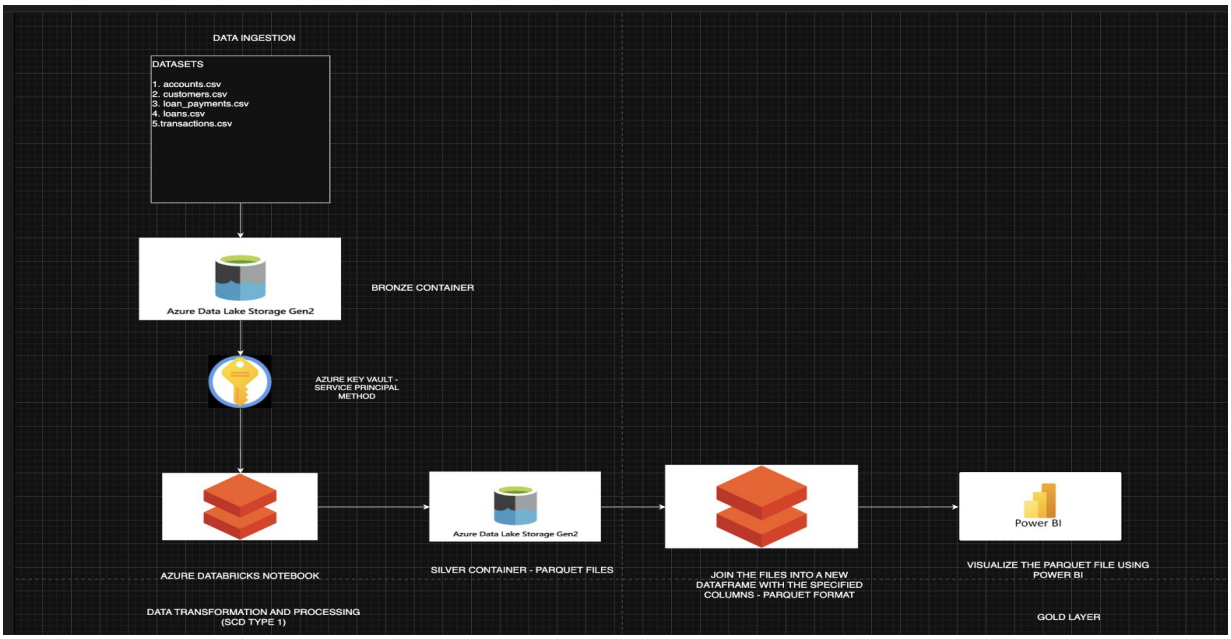
TRANSACTIONS AND LOAN DATA – DATABRICKS

SUBMITTED BY – DRUSYA SURESH

OBJECTIVE

This project aims in building a robust and efficient data processing of a financial data using Databricks notebook. This includes loading data into the ADLS storage container (Bronze layer) and transforming the data in the silver layer using Data bricks notebooks and Gold layer storage storage into SCD Type 1 Delta tables in ADLS gen2. The project also includes a visualization of the delta table using power BI and Fabric. The project aims to ensure an efficient, accurate and scalable data processing to support downstream analytics and reporting needs.

ARCHITECTURE DIAGRAM



STEPS INVOLVED:

Step 1: Load the data into the Bronze container in the ADLS Gen2 storage account (raw data).

Step2: Using Data bricks notebooks connect the ADLS storage account using service principal method.

Step 3: Read the data from the files, perform necessary transformations on the data (cleaning data), load them as Parquet files in the ADLS storage account Silver container. Also create a new Parquet file with specified columns using join condition and store it in ADLS.

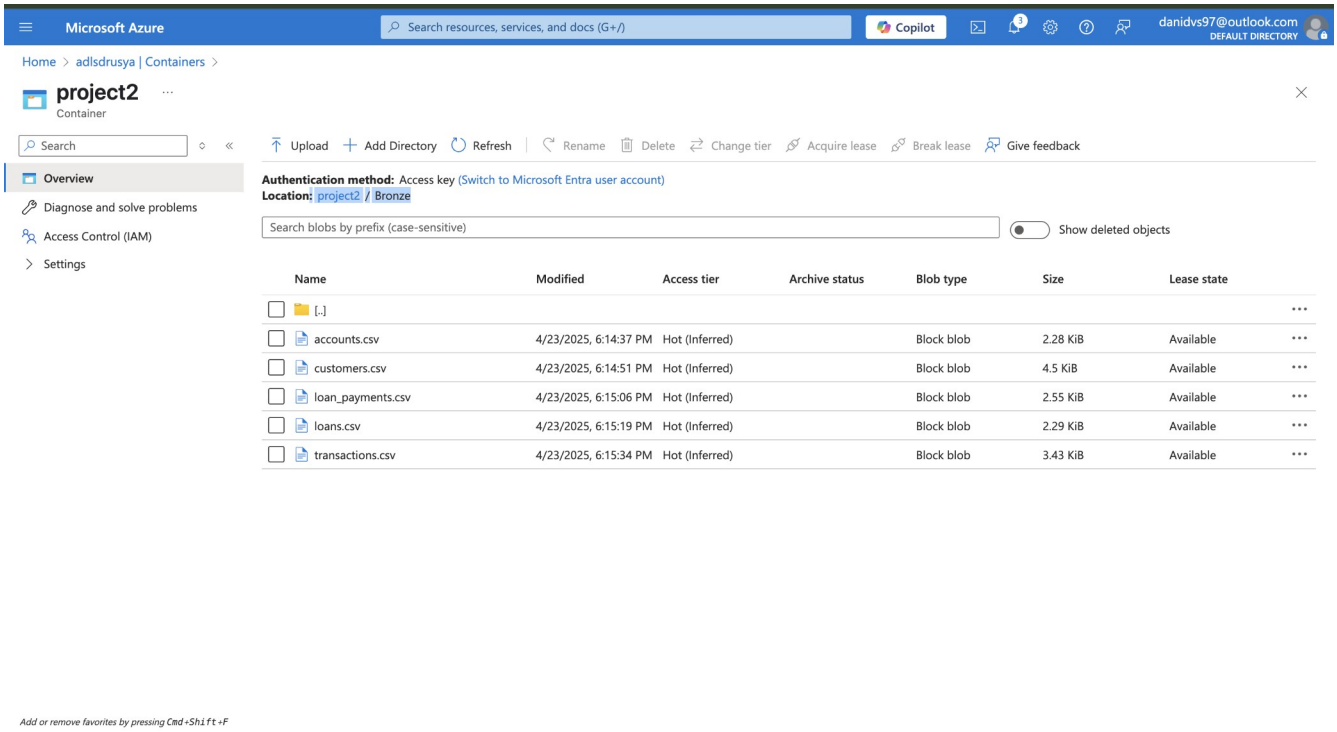
Step 4: Perform SCD Type 1 logic on the 5 Parquet files.

Step 5: Schedule the workspace in Data bricks to run on a specified time.

Step 5: Visualize the joined file using power BI and publish the report in Fabric.

STEP 1:

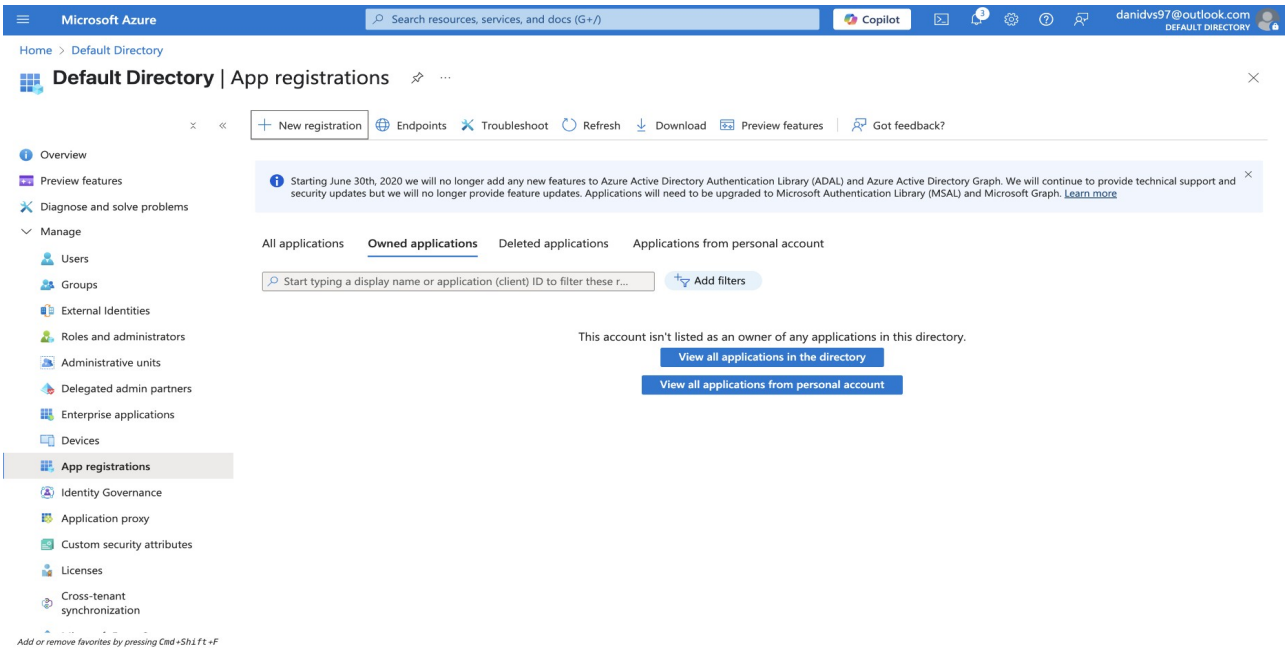
Bronze layer – Loaded the raw csv files into the ADLS storage account directly using the upload files option successfully under the folder project 2/Bronze.

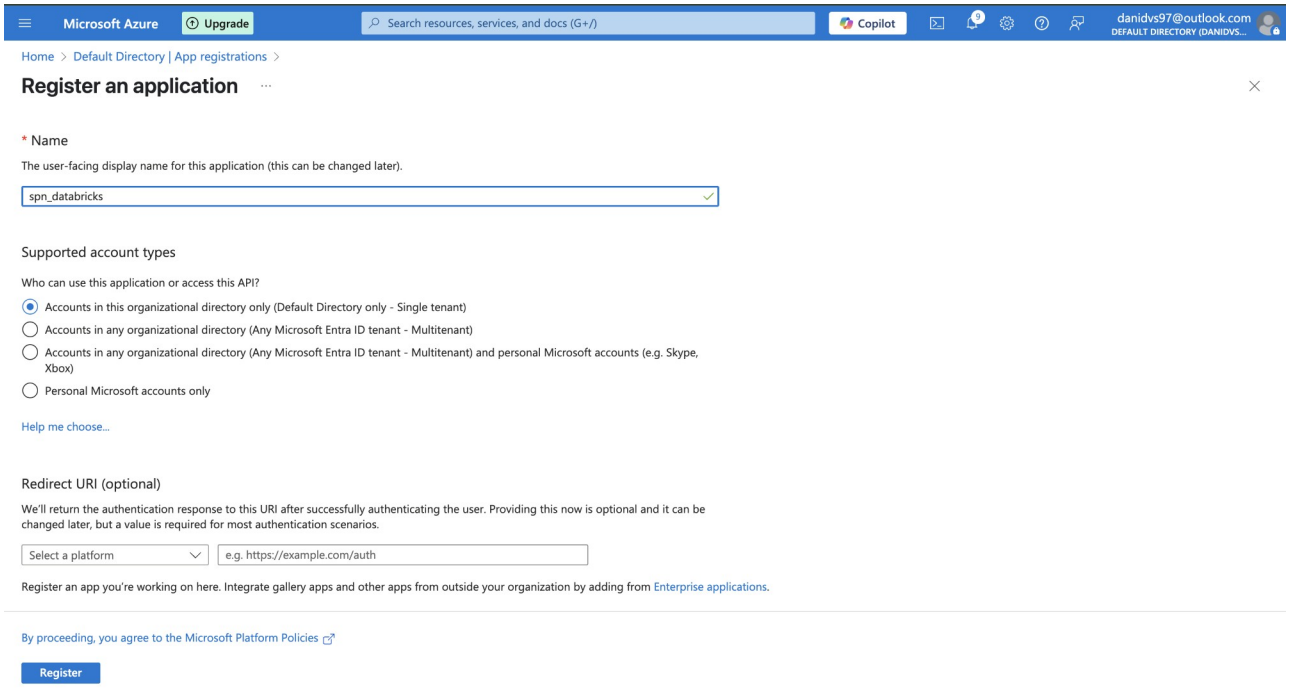


STEP 2:

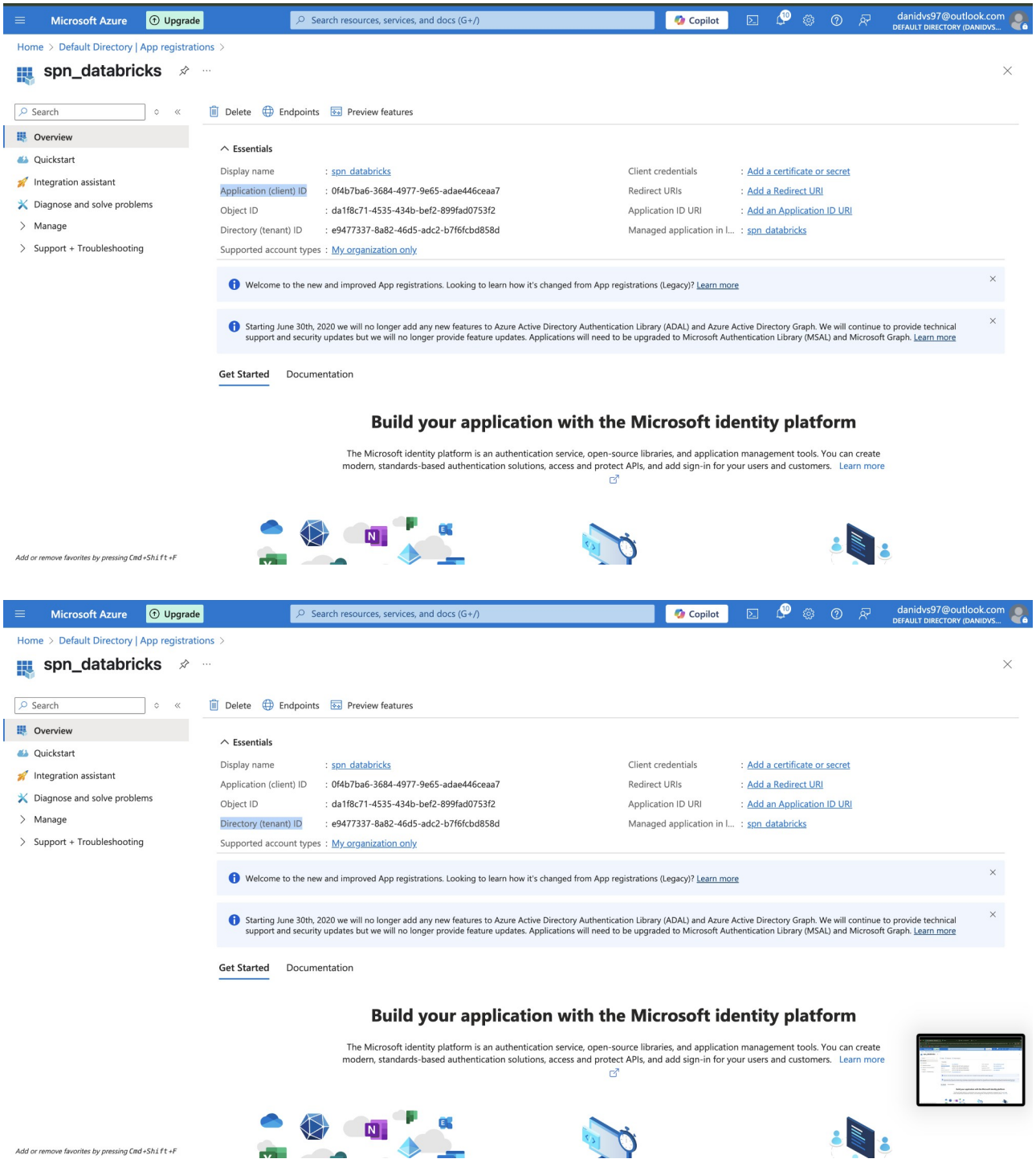
Connecting ADLS with notebook ising service principal method.

- Go to the microsoft entra ID and click on app registrations and create a new registration.

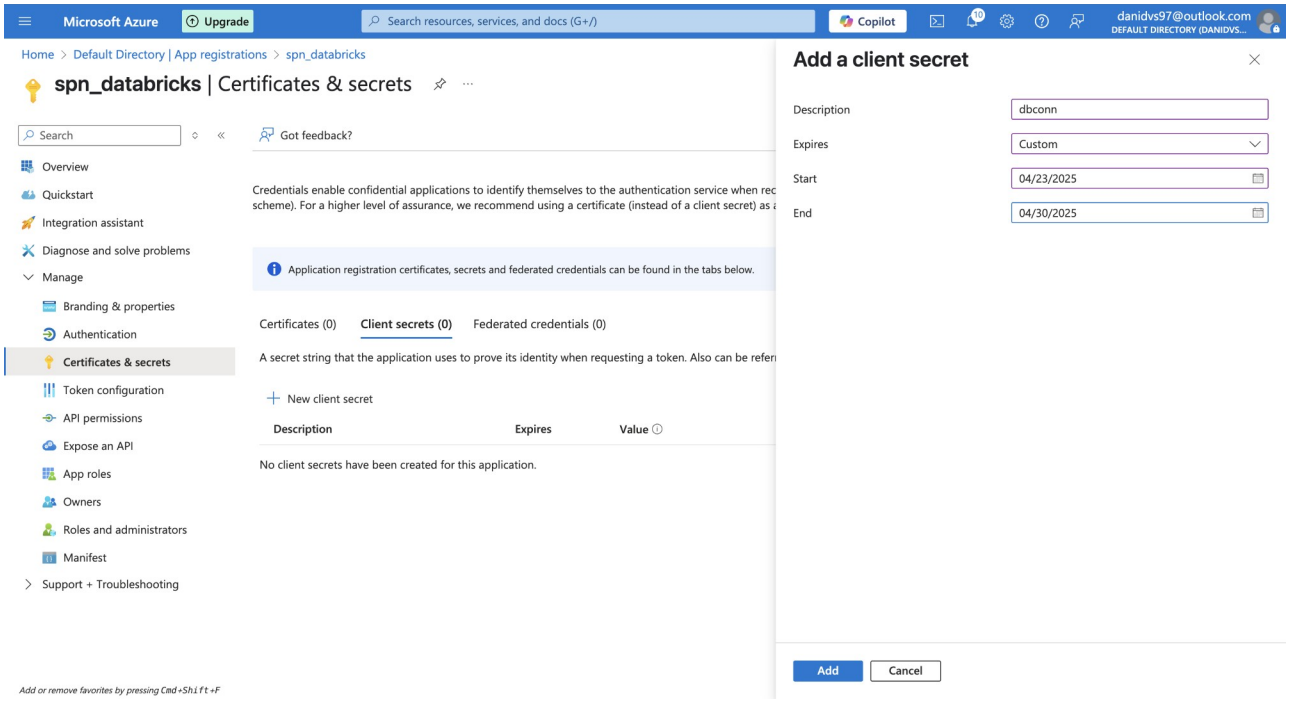


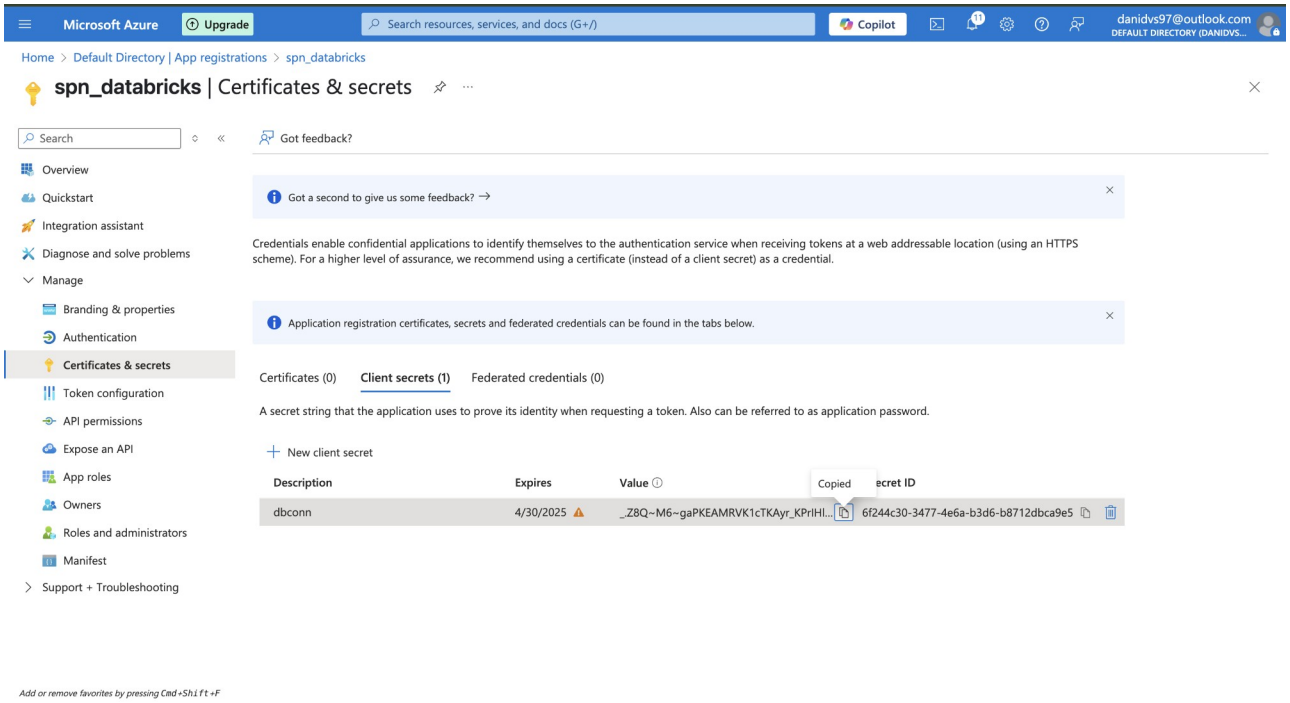


- Now click on the created app registration and in the following page copy the application ID and directory ID from there are paste it somewhere.

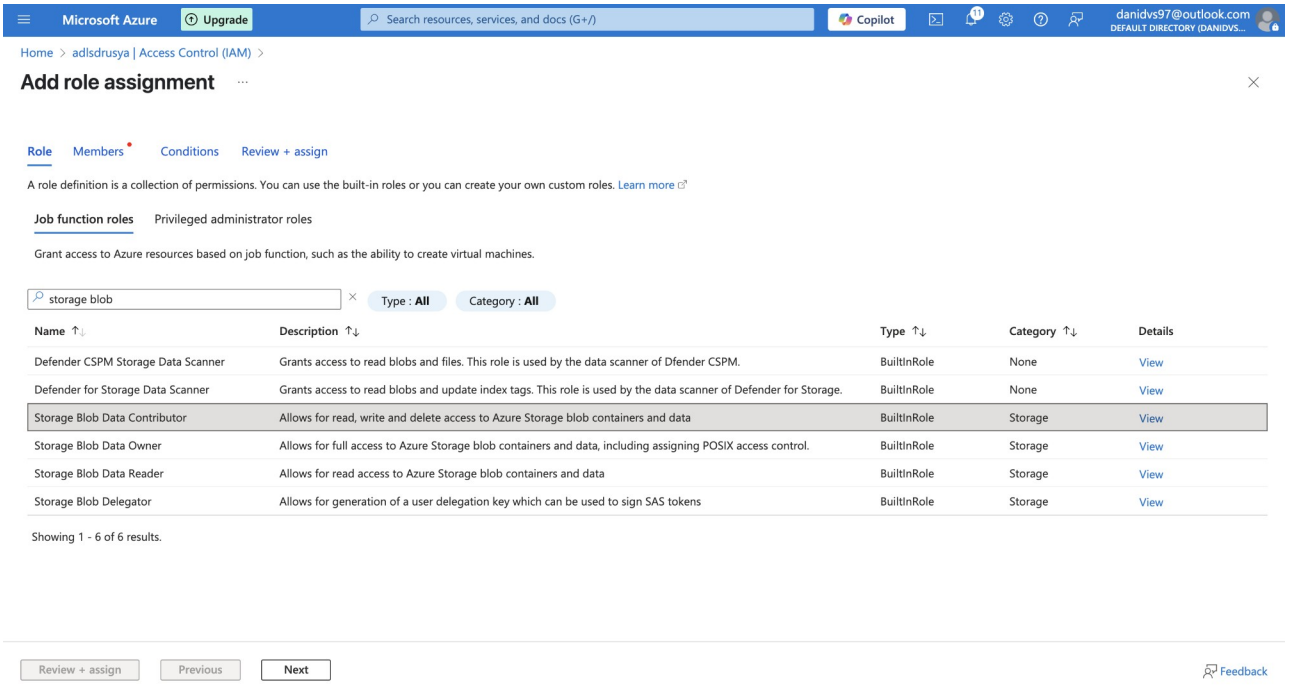


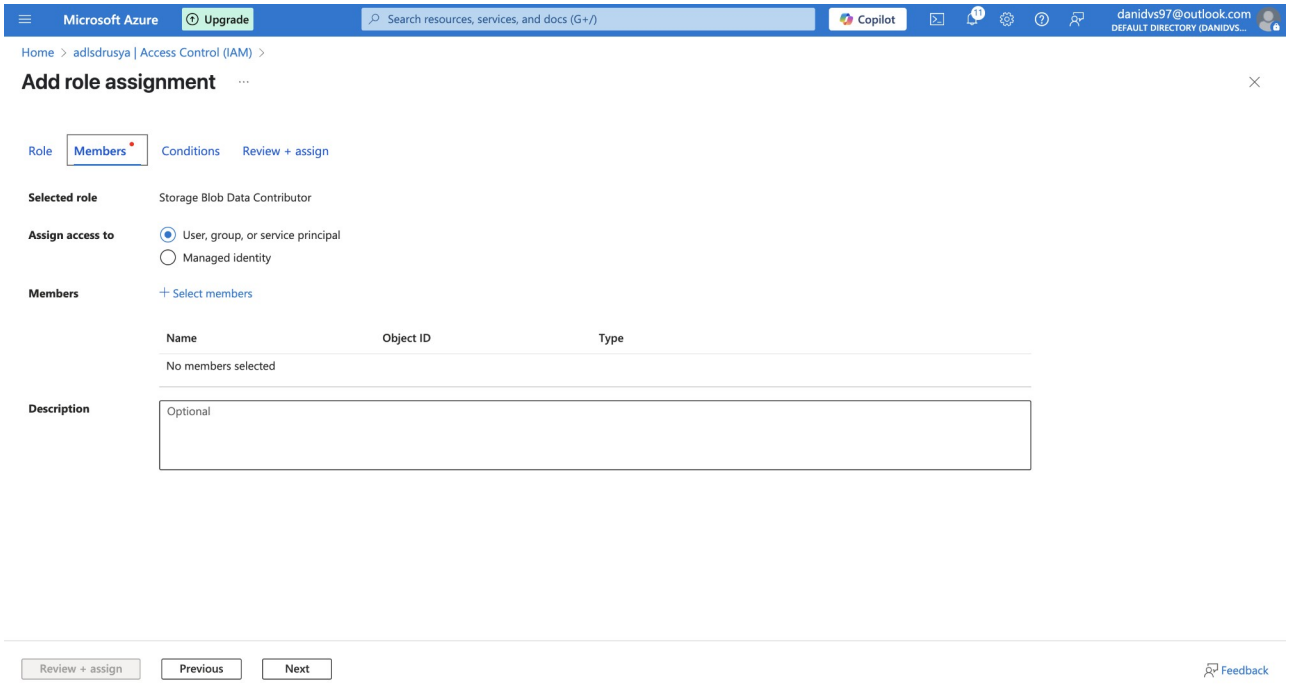
- Click on the manage tab in the app registered and click on certificates and secrets tab, now create a new secret and copy paste the secret value somewhere.

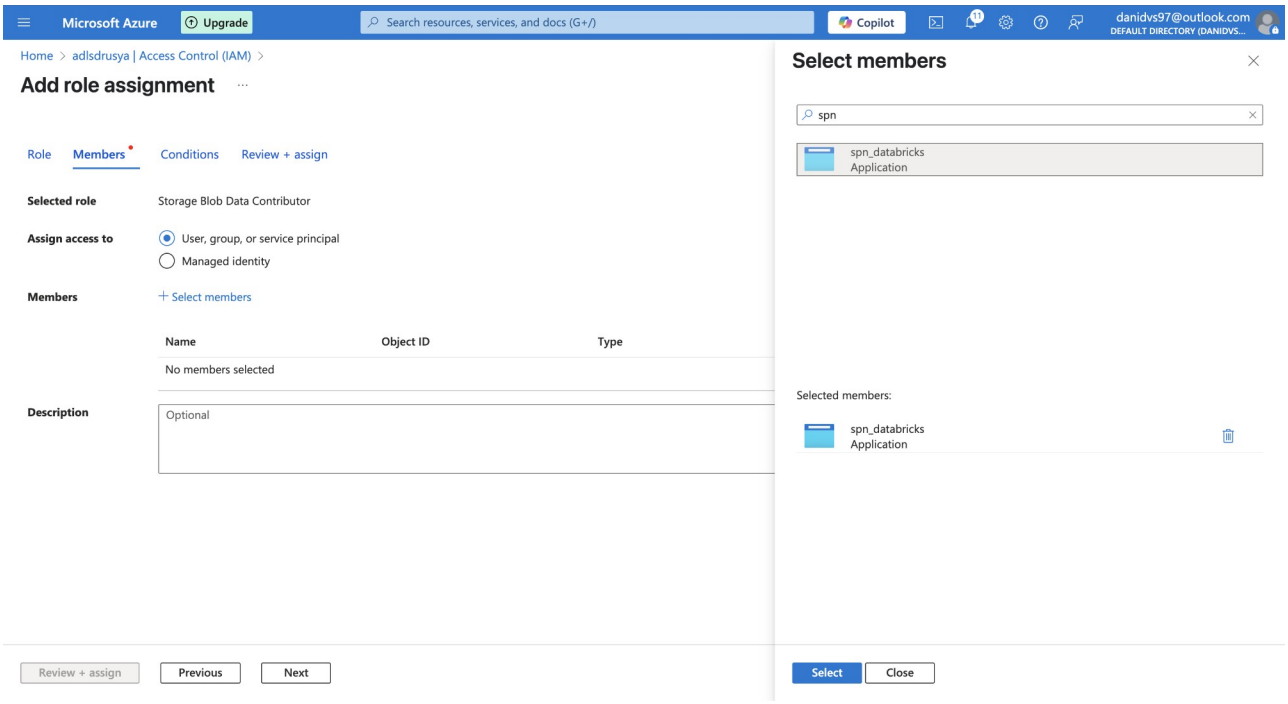


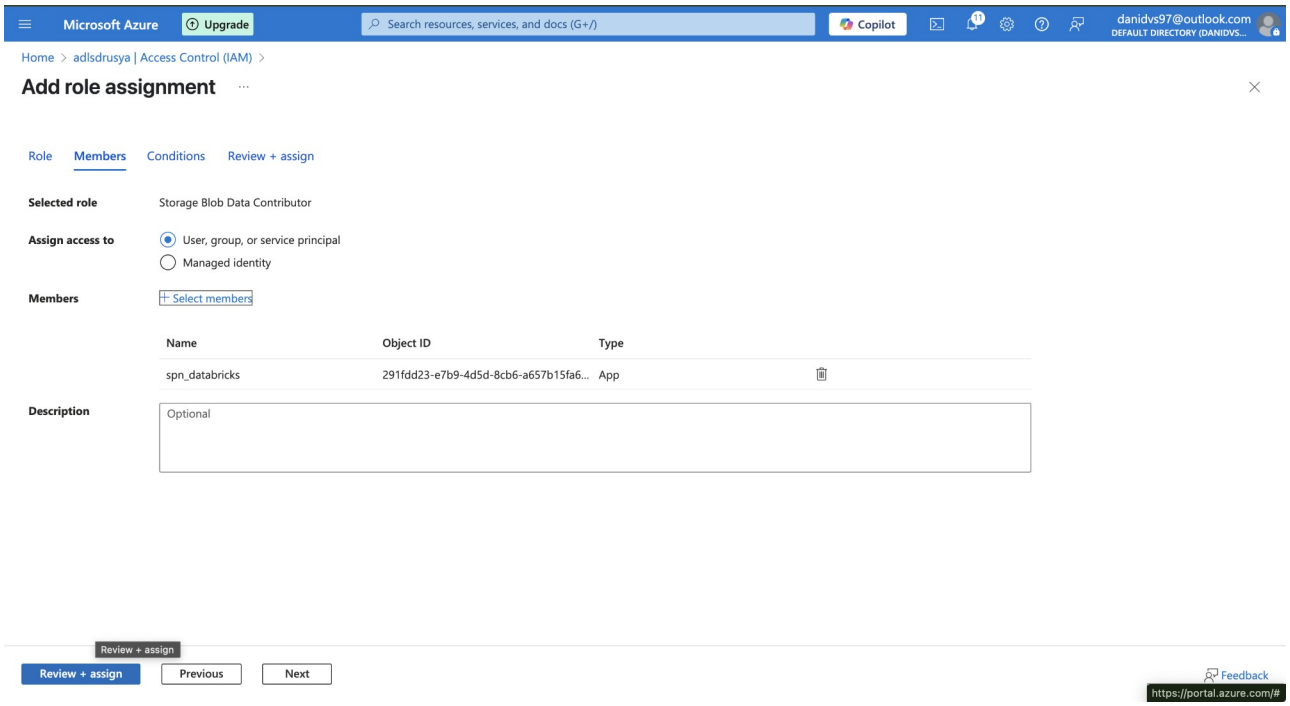


- Now go to the ADLS storage account and under the access control add a new role assignment as storage blob data contributor rolet o the app registration we created using the select members option.

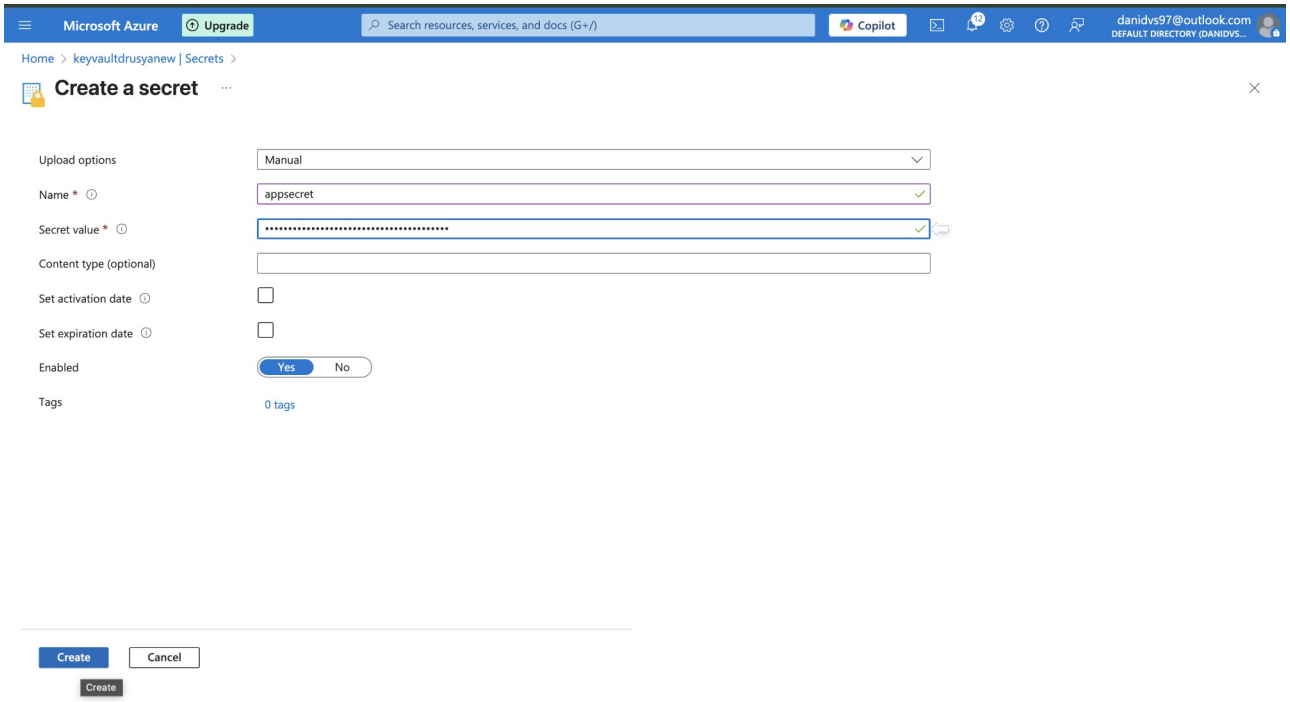

Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Copilot
danidvs97@outlook.com
DEFAULT DIRECTORY (DANIDVS...
Home > adlsdrusya | Access Control (IAM) >
Add role assignment
Role
Members
Conditions
Review + assign
A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. Learn more
Job function roles
Privileged administrator roles
Grant access to Azure resources based on job function, such as the ability to create virtual machines.
storage blob
Type : All
Category : All
Name
Description
Type
Category
Details
Defender CSPM Storage Data Scanner
Grants access to read blobs and files. This role is used by the data scanner of Dfender CSPM.
BuiltInRole
None
View
Defender for Storage Data Scanner
Grants access to read blobs and update index tags. This role is used by the data scanner of Defender for Storage.
BuiltInRole
None
View
Storage Blob Data Contributor
Allows for read, write and delete access to Azure Storage blob containers and data
BuiltInRole
Storage
View
Storage Blob Data Owner
Allows for full access to Azure Storage blob containers and data, including assigning POSIX access control.
BuiltInRole
Storage
View
Storage Blob Data Reader
Allows for read access to Azure Storage blob containers and data
BuiltInRole
Storage
View
Storage Blob Delegator
Allows for generation of a user delegation key which can be used to sign SAS tokens
BuiltInRole
Storage
View
Showing 1 - 6 of 6 results.
Review + assign
Previous
Next
Feedback

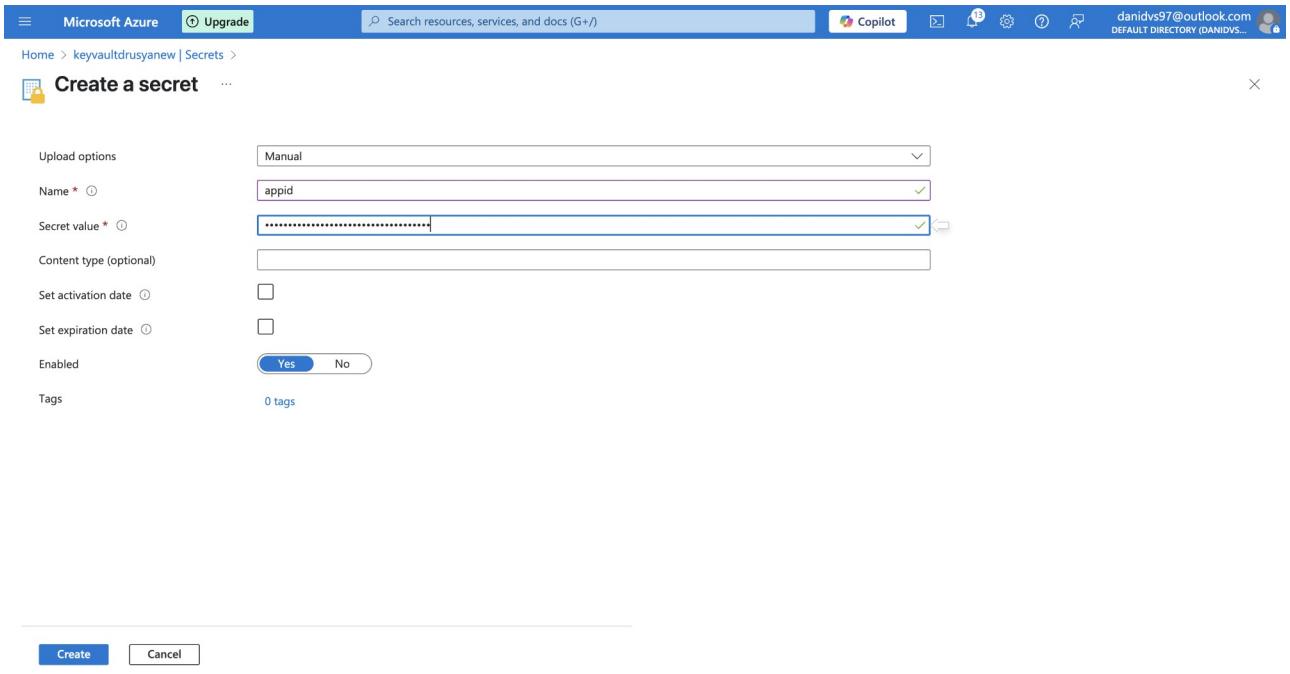

Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Copilot
danidvs97@outlook.com
DEFAULT DIRECTORY (DANIDVS...
Home > adlsdrusya | Access Control (IAM) >
Add role assignment
Role
Members
Conditions
Review + assign
Selected role
Storage Blob Data Contributor
Assign access to
User, group, or service principal
Managed identity
Members
+ Select members
Name
Object ID
Type
No members selected
Description
Optional
Review + assign
Previous
Next
Feedback

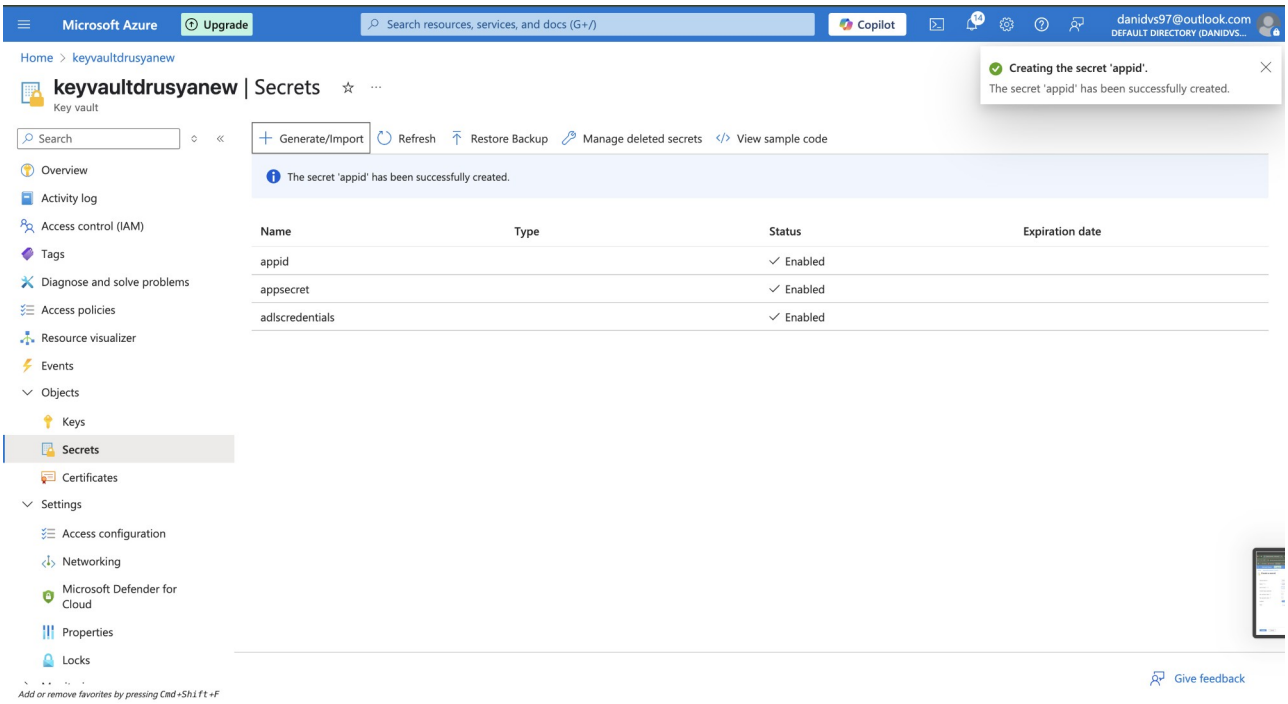

Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Copilot
danidvs97@outlook.com
DEFAULT DIRECTORY (DANIDVS...
Home > adlsdrusya | Access Control (IAM) >
Add role assignment
Role
Members
Conditions
Review + assign
Selected role
Storage Blob Data Contributor
Assign access to
User, group, or service principal
Managed identity
Members
+ Select members
Name
Object ID
Type
No members selected
Description
Optional
Review + assign
Previous
Next
Select members
spn
spn_databricks
Application
Selected members:
spn_databricks
Application
Select
Close


Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Copilot
danidvs97@outlook.com
DEFAULT DIRECTORY (DANIDVS...
Home > adlsdrusya | Access Control (IAM) >
Add role assignment
Role
Members
Conditions
Review + assign
Selected role
Storage Blob Data Contributor
Assign access to
User, group, or service principal
Managed identity
Members
+ Select members
Name
Object ID
Type
spn_databricks
291fdd23-e7b9-4d5d-8cb6-a657b15fa6...
App
Description
Optional
Review + assign
Review + assign
Previous
Next
Feedback
https://portal.azure.com/#

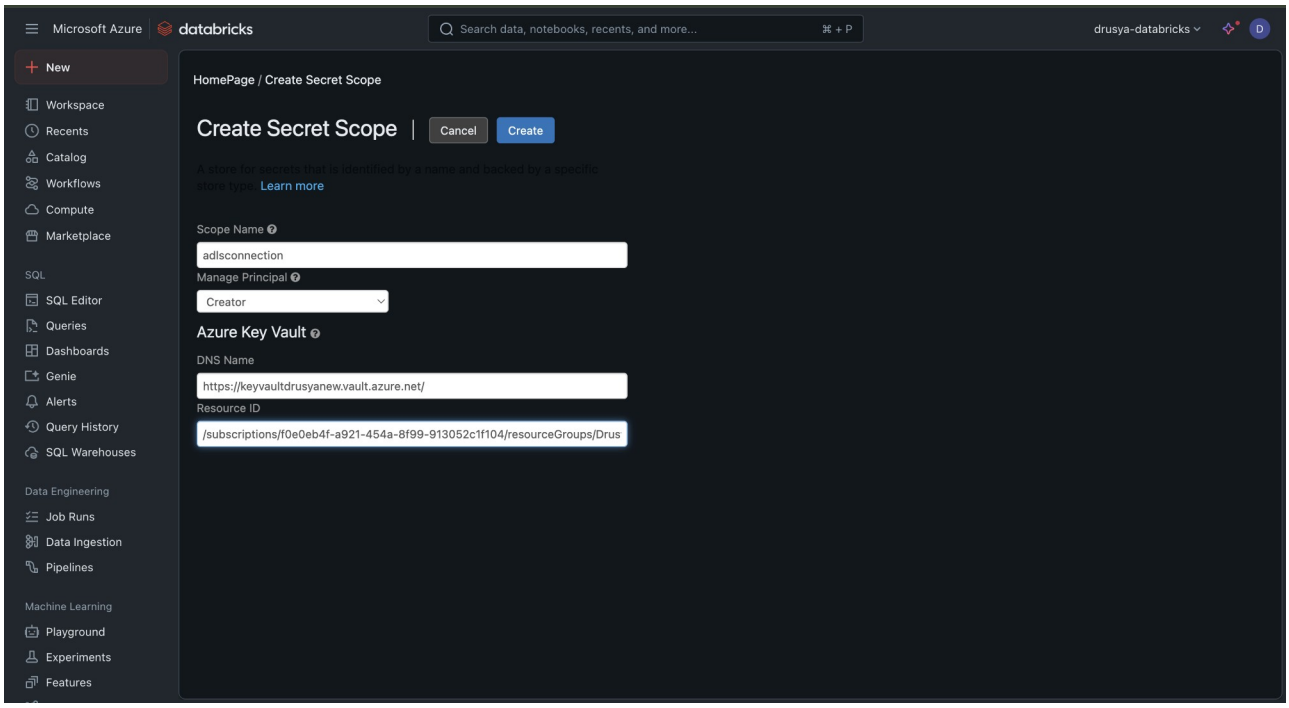
- Now go to the key vault and add two secrets for the app id and secret value.



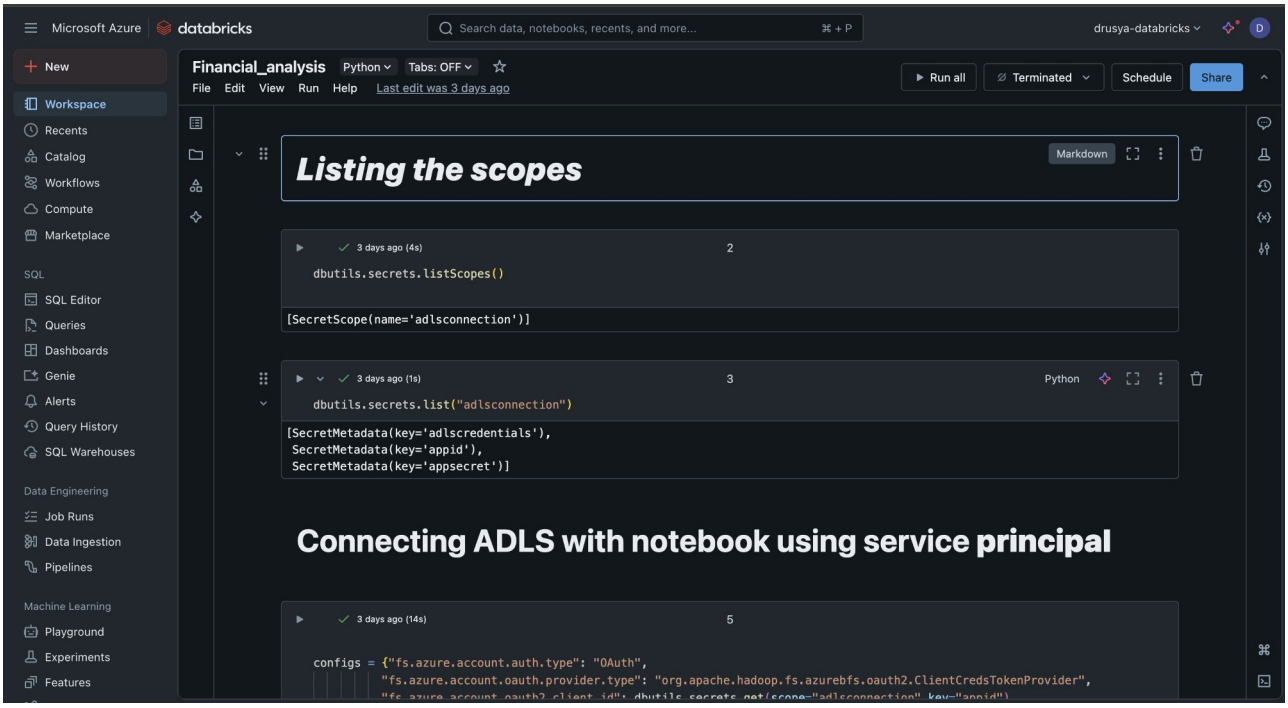


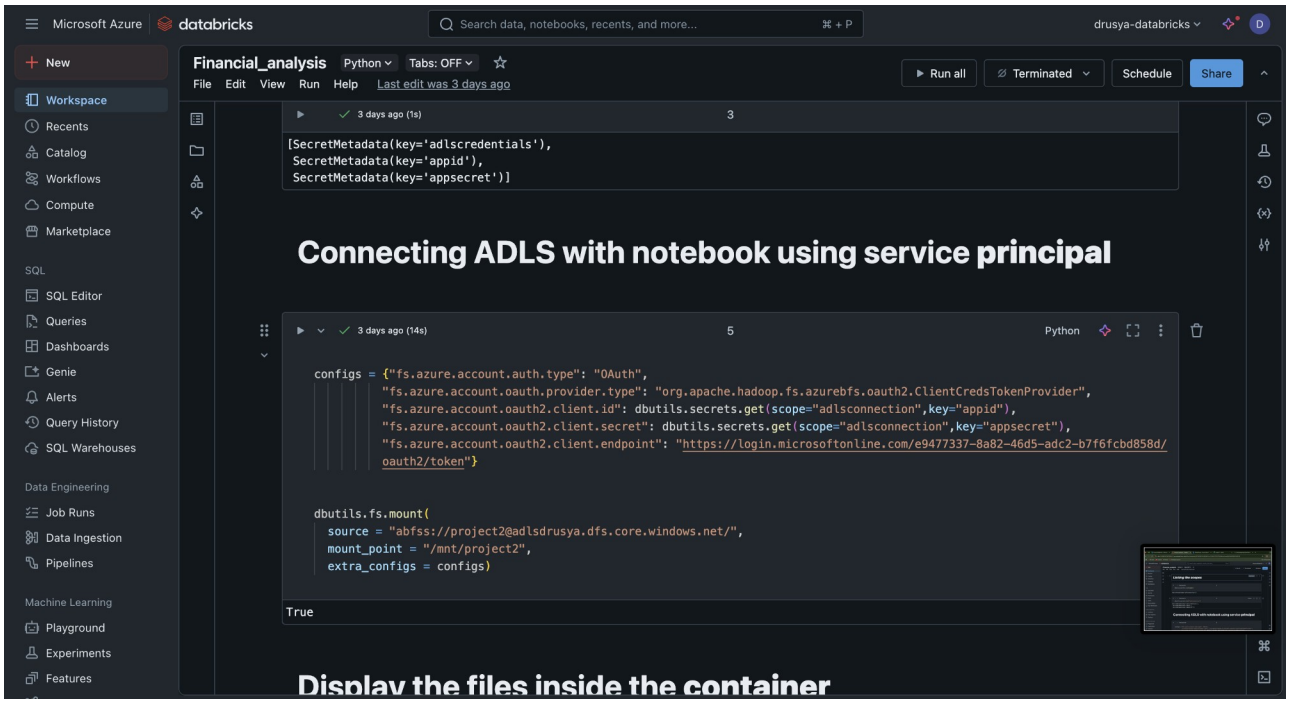


- Now open Data bricks note book and create a new scope for the key vault using key vault id and resource id.



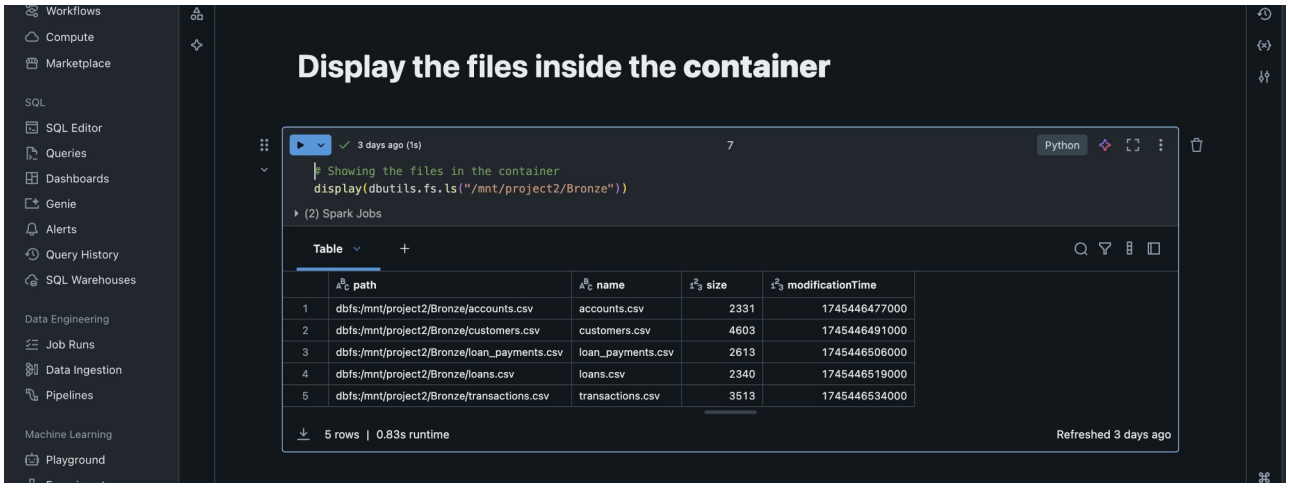
- Now connect the ADLS with note book using the code below and paste the values of the respective ids as shown in the screenshot and run the command it will connect ADLS with the notebook.

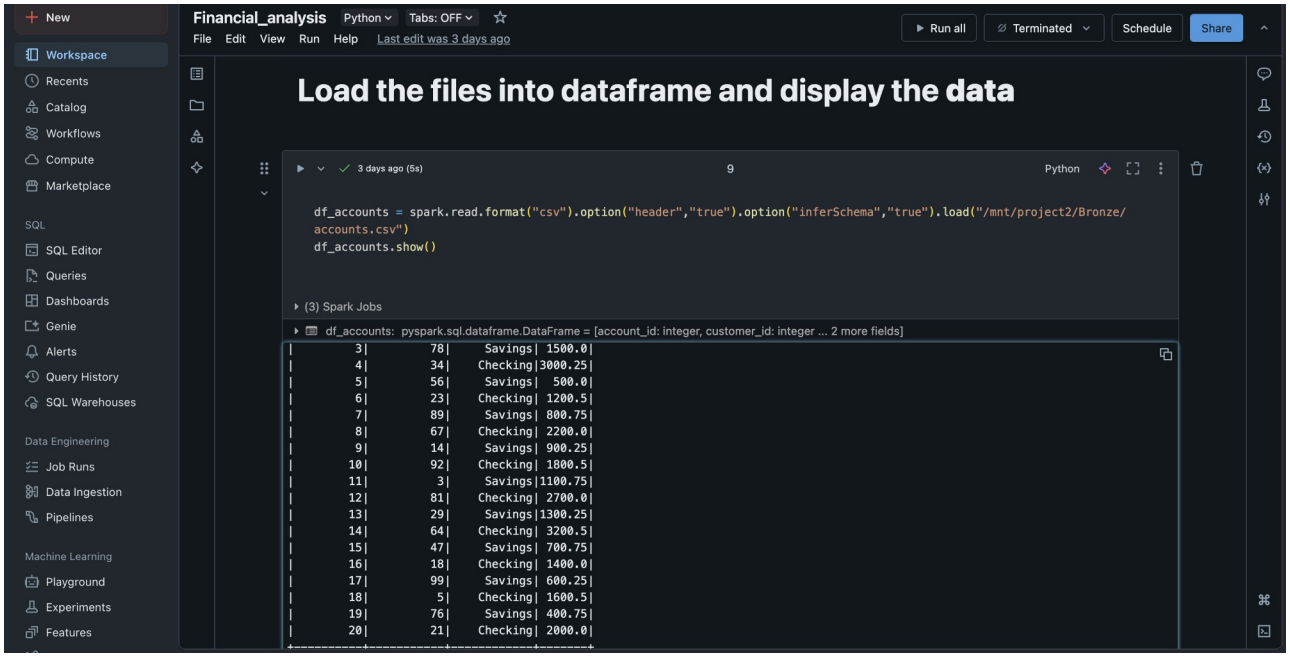


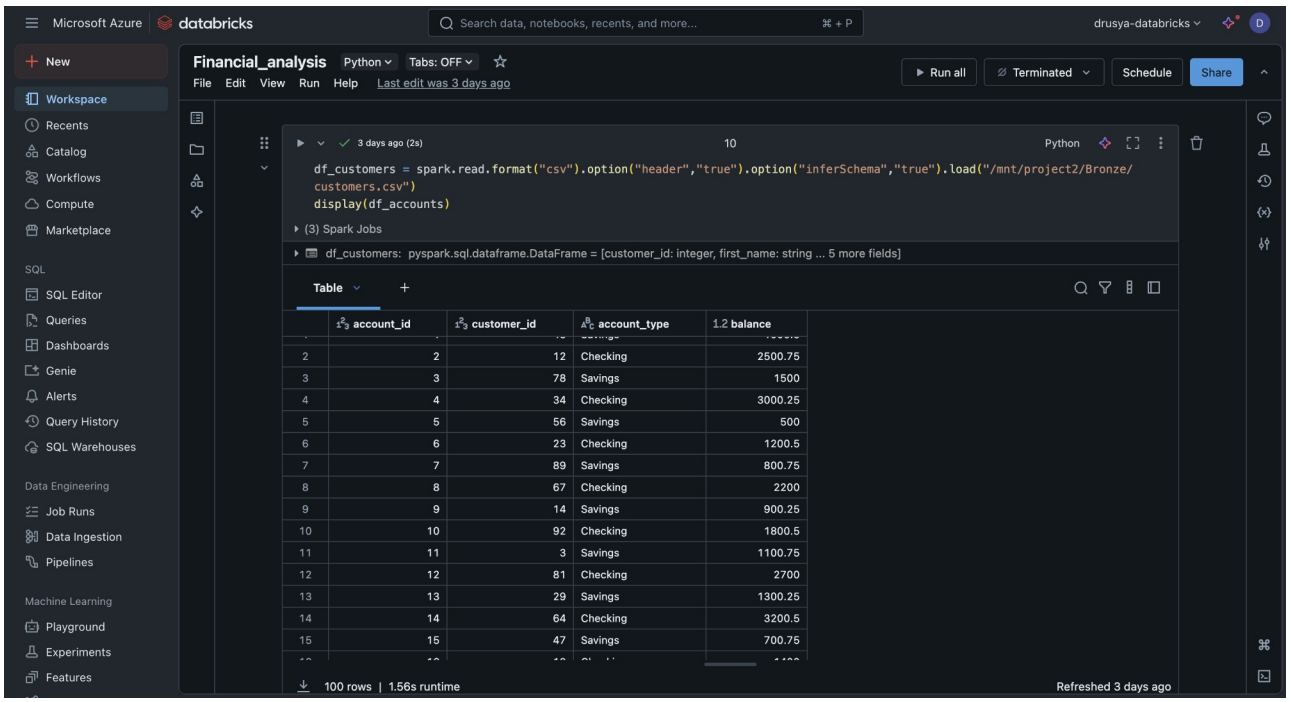


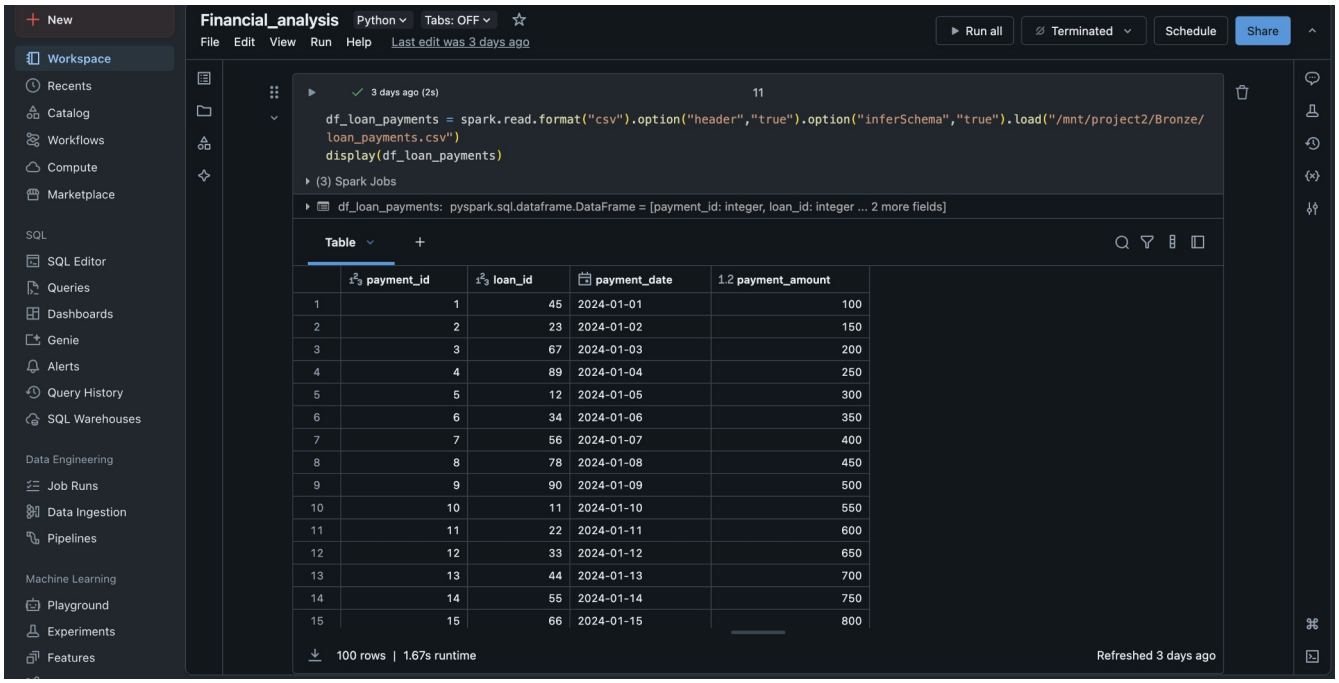
STEP 3: READING THE FILES AND CLEANING

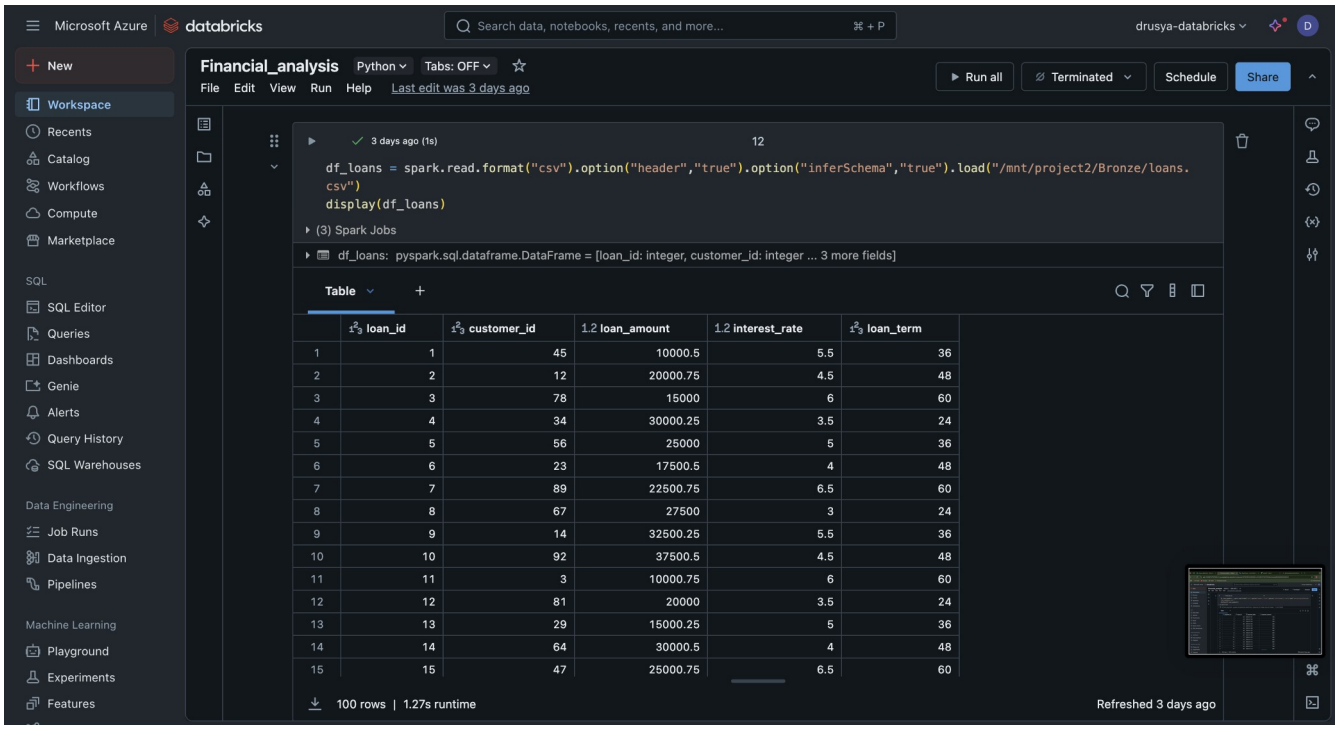
- Open the files from the notebooks and loading into a data frame using the read command.

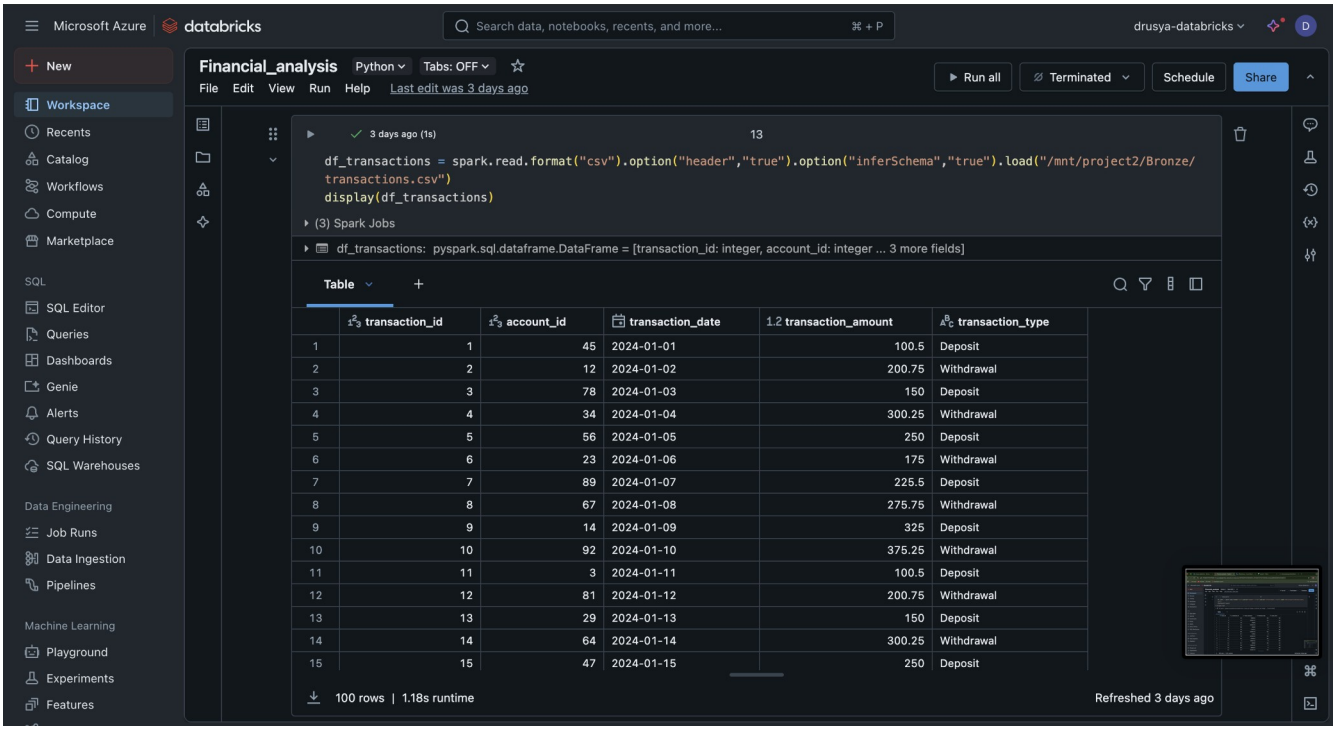





Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
New
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help
Last edit was 3 days ago
Run all
Terminated
Schedule
Share
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
3 days ago (2s)
10
Python
df_customers = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/project2/Bronze/customers.csv")
display(df_accounts)
(3) Spark Jobs
df_customers: pyspark.sql.dataframe.DataFrame = [customer_id: integer, first_name: string ... 5 more fields]
Table
account_id
customer_id
account_type
balance
2 2 12 Checking 2500.75
3 3 78 Savings 1500
4 4 34 Checking 3000.25
5 5 56 Savings 500
6 6 23 Checking 1200.5
7 7 89 Savings 800.75
8 8 67 Checking 2200
9 9 14 Savings 900.25
10 10 92 Checking 1800.5
11 11 3 Savings 1100.75
12 12 81 Checking 2700
13 13 29 Savings 1300.25
14 14 64 Checking 3200.5
15 15 47 Savings 700.75
100 rows | 1.56s runtime
Refreshed 3 days ago

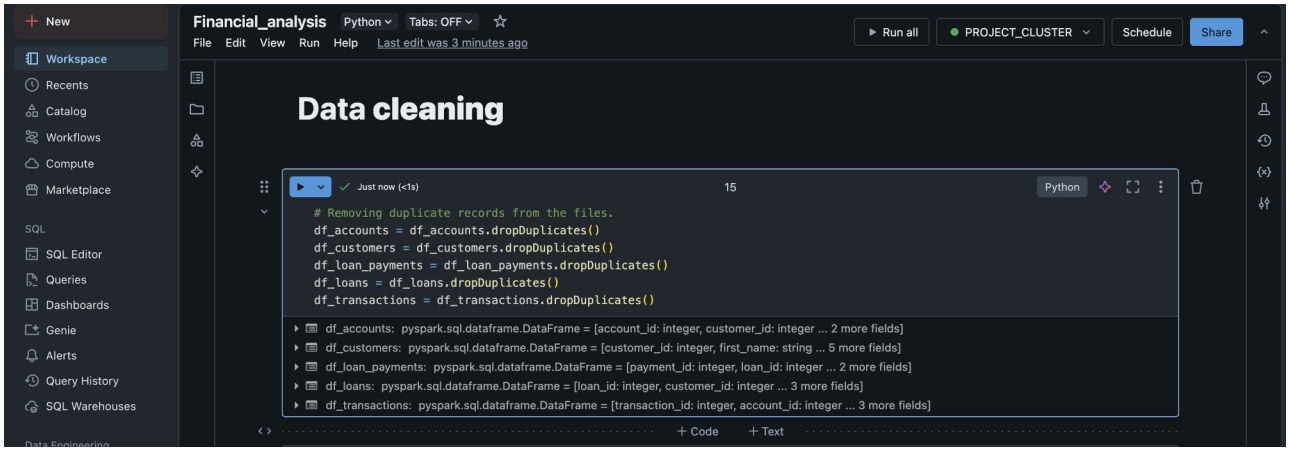

New
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help
Last edit was 3 days ago
Run all
Terminated
Schedule
Share
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
3 days ago (2s)
11
df_loan_payments = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/project2/Bronze/loan_payments.csv")
display(df_loan_payments)
(3) Spark Jobs
df_loan_payments: pyspark.sql.dataframe.DataFrame = [payment_id: integer, loan_id: integer ... 2 more fields]
Table
payment_id
loan_id
payment_date
payment_amount
1 1 45 2024-01-01 100
2 2 23 2024-01-02 150
3 3 67 2024-01-03 200
4 4 89 2024-01-04 250
5 5 12 2024-01-05 300
6 6 34 2024-01-06 350
7 7 56 2024-01-07 400
8 8 78 2024-01-08 450
9 9 90 2024-01-09 500
10 10 11 2024-01-10 550
11 11 22 2024-01-11 600
12 12 33 2024-01-12 650
13 13 44 2024-01-13 700
14 14 55 2024-01-14 750
15 15 66 2024-01-15 800
100 rows | 1.67s runtime
Refreshed 3 days ago


Microsoft Azure
databricks
Search data, notebooks, recents, and more... ⌘ + P
drusya-databricks
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
Machine Learning
Playground
Experiments
Features
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help
Last edit was 3 days ago
Run all
Terminated
Schedule
Share
3 days ago (1s)
12
df_loans = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/project2/Bronze/loans.csv")
display(df_loans)
(3) Spark Jobs
df_loans: pyspark.sql.dataframe.DataFrame = [loan_id: integer, customer_id: integer ... 3 more fields]
Table
loan_id
customer_id
loan_amount
interest_rate
loan_term
1 1 45 10000.5 5.5 36
2 2 12 20000.75 4.5 48
3 3 78 15000 6 60
4 4 34 30000.25 3.5 24
5 5 56 25000 5 36
6 6 23 17500.5 4 48
7 7 89 22500.75 6.5 60
8 8 67 27500 3 24
9 9 14 32500.25 5.5 36
10 10 92 37500.5 4.5 48
11 11 3 10000.75 6 60
12 12 81 20000 3.5 24
13 13 29 15000.25 5 36
14 14 64 30000.5 4 48
15 15 47 25000.75 6.5 60
100 rows | 1.27s runtime
Refreshed 3 days ago

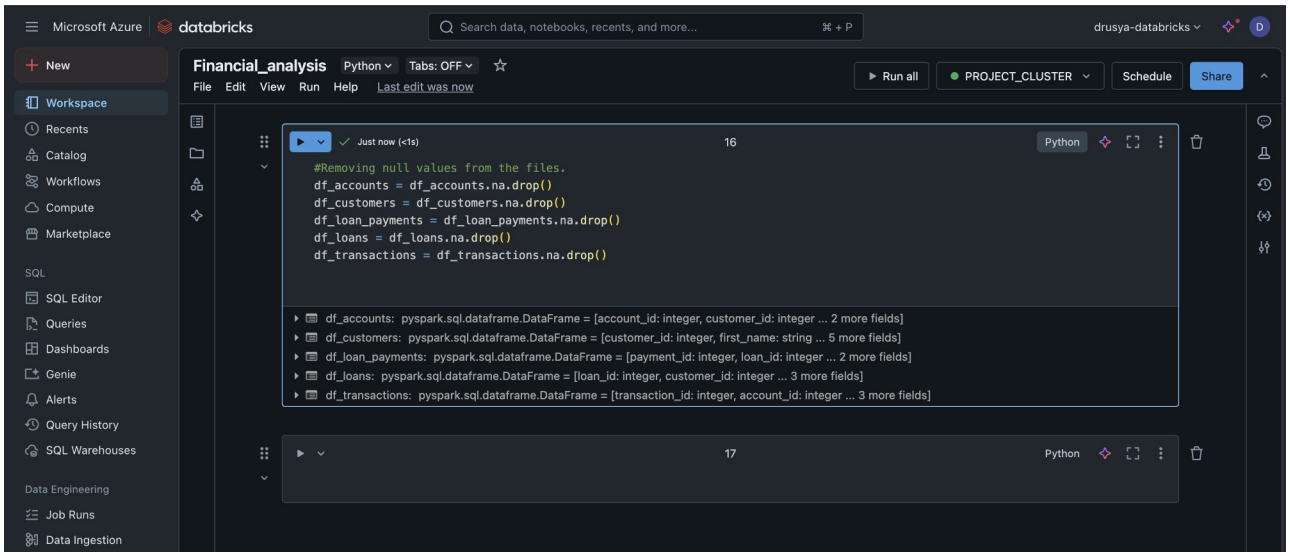

Microsoft Azure
databricks
Search data, notebooks, recents, and more... ⌘ + P
drusya-databricks
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help
Last edit was 3 days ago
Run all
Terminated
Schedule
Share
3 days ago (1s)
13
df_transactions = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/project2/Bronze/transactions.csv")
display(df_transactions)
(3) Spark Jobs
df_transactions: pyspark.sql.dataframe.DataFrame = [transaction_id: integer, account_id: integer ... 3 more fields]
Table
transaction_id
account_id
transaction_date
transaction_amount
transaction_type
1 1 45 2024-01-01 100.5 Deposit
2 2 12 2024-01-02 200.75 Withdrawal
3 3 78 2024-01-03 150 Deposit
4 4 34 2024-01-04 300.25 Withdrawal
5 5 56 2024-01-05 250 Deposit
6 6 23 2024-01-06 175 Withdrawal
7 7 89 2024-01-07 225.5 Deposit
8 8 67 2024-01-08 275.75 Withdrawal
9 9 14 2024-01-09 325 Deposit
10 10 92 2024-01-10 375.25 Withdrawal
11 11 3 2024-01-11 100.5 Deposit
12 12 81 2024-01-12 200.75 Withdrawal
13 13 29 2024-01-13 150 Deposit
14 14 64 2024-01-14 300.25 Withdrawal
15 15 47 2024-01-15 250 Deposit
100 rows | 1.18s runtime
Refreshed 3 days ago

- Data cleaning.

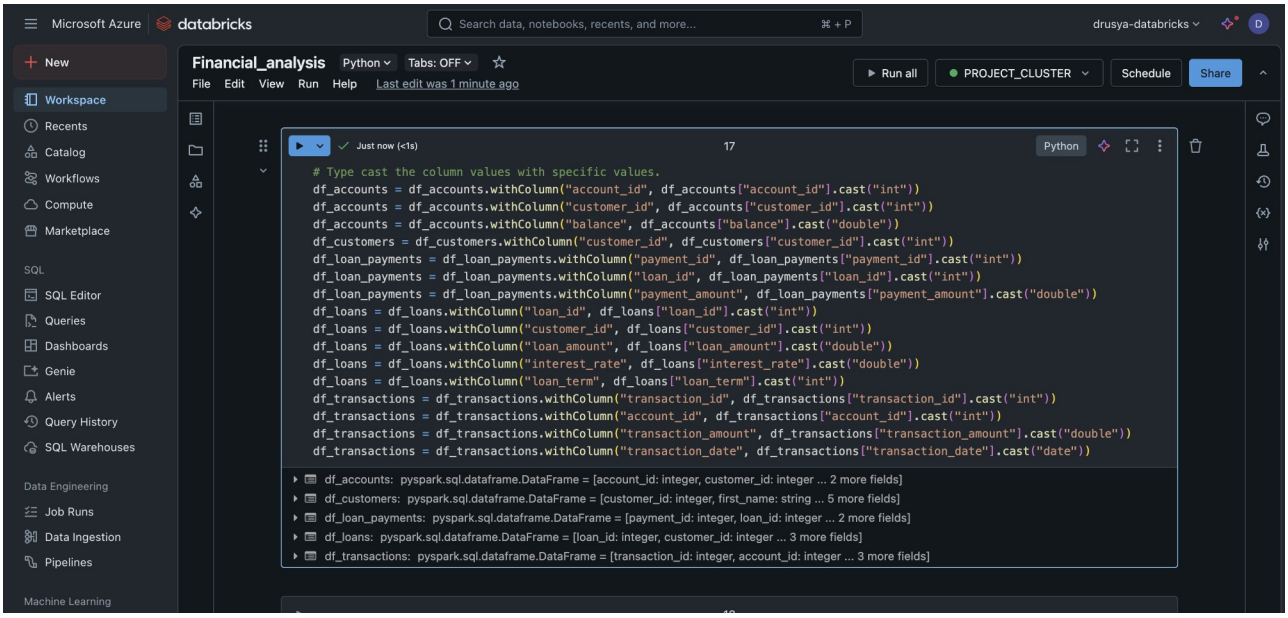
1. Removing duplicates from the files.



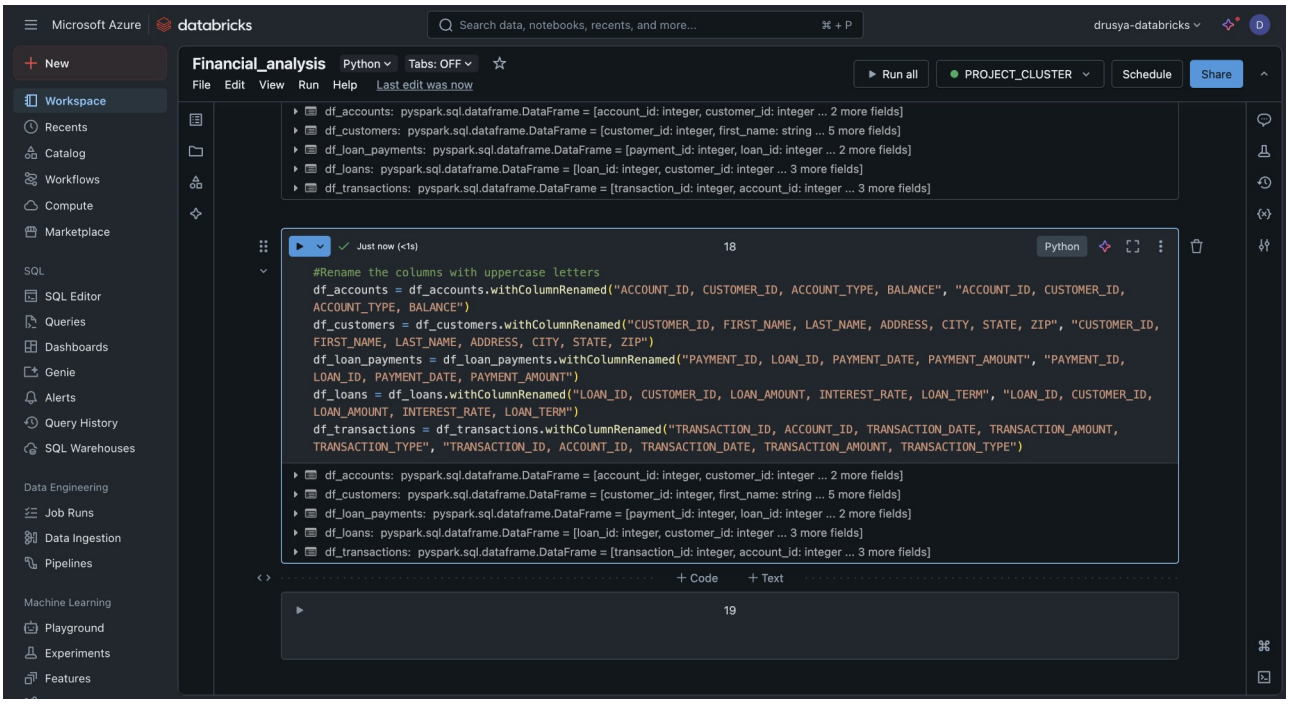
2. Removing null values if there are any null values.



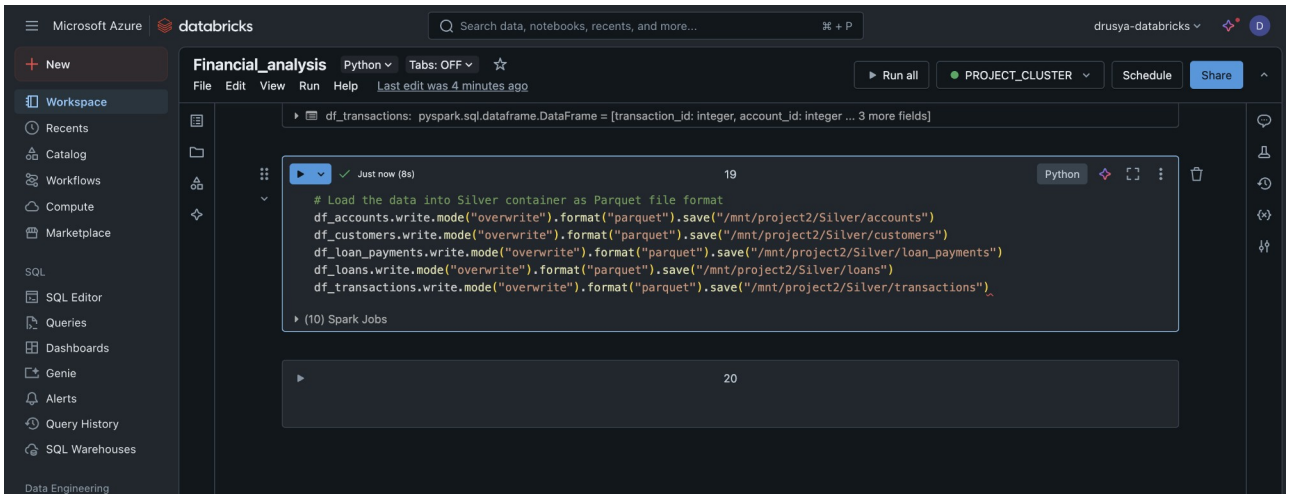
3. Type casting the columns with specific values.

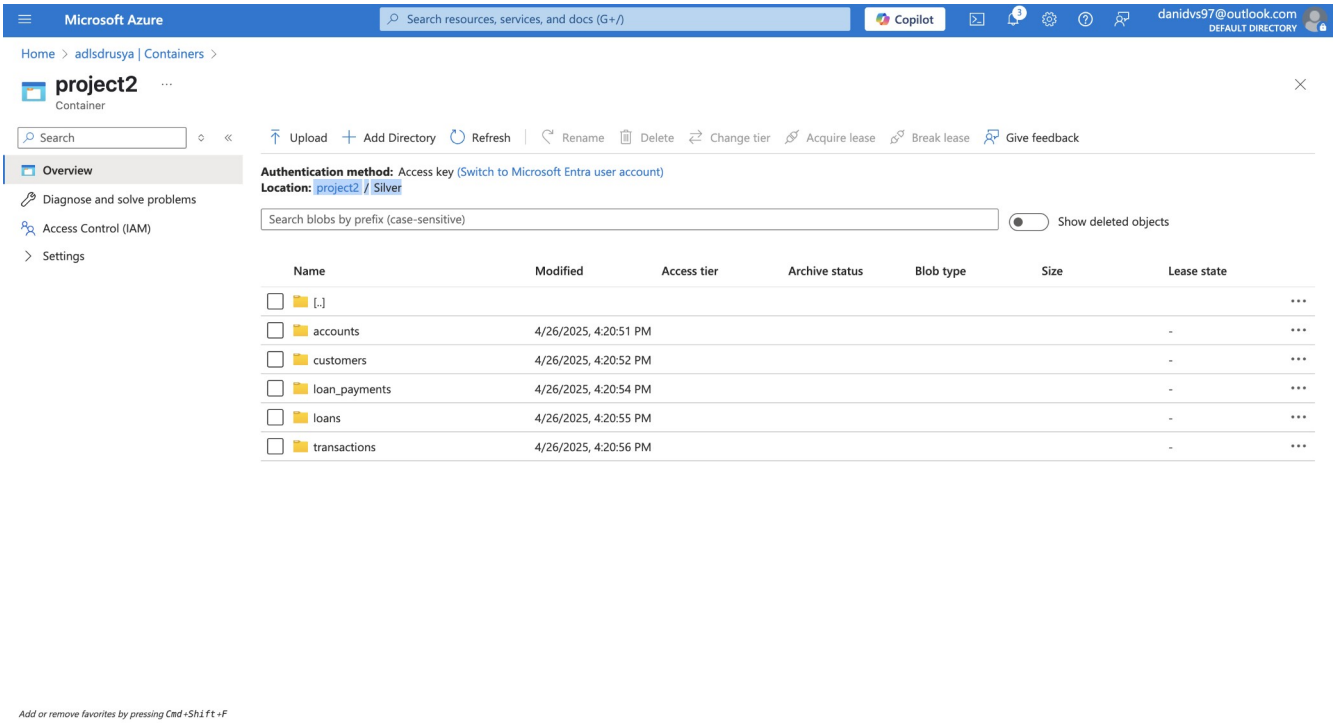


4. Renaming the columns with upper case letters.

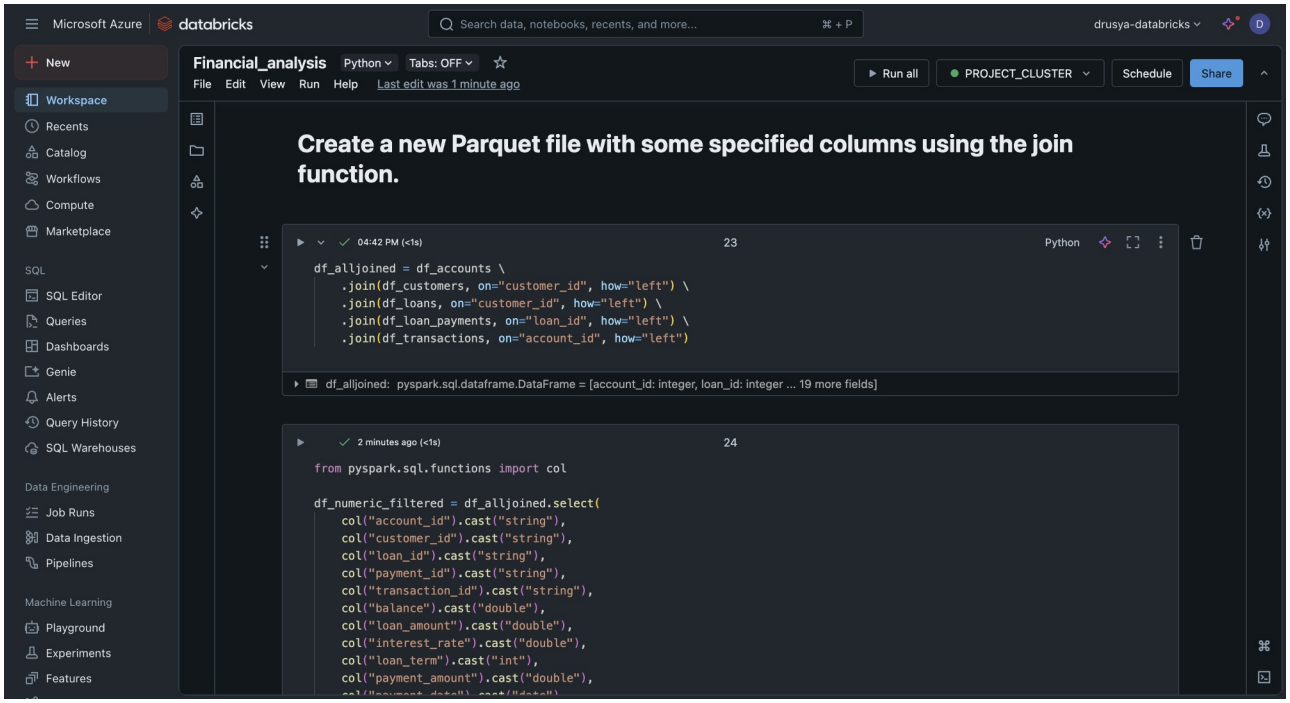


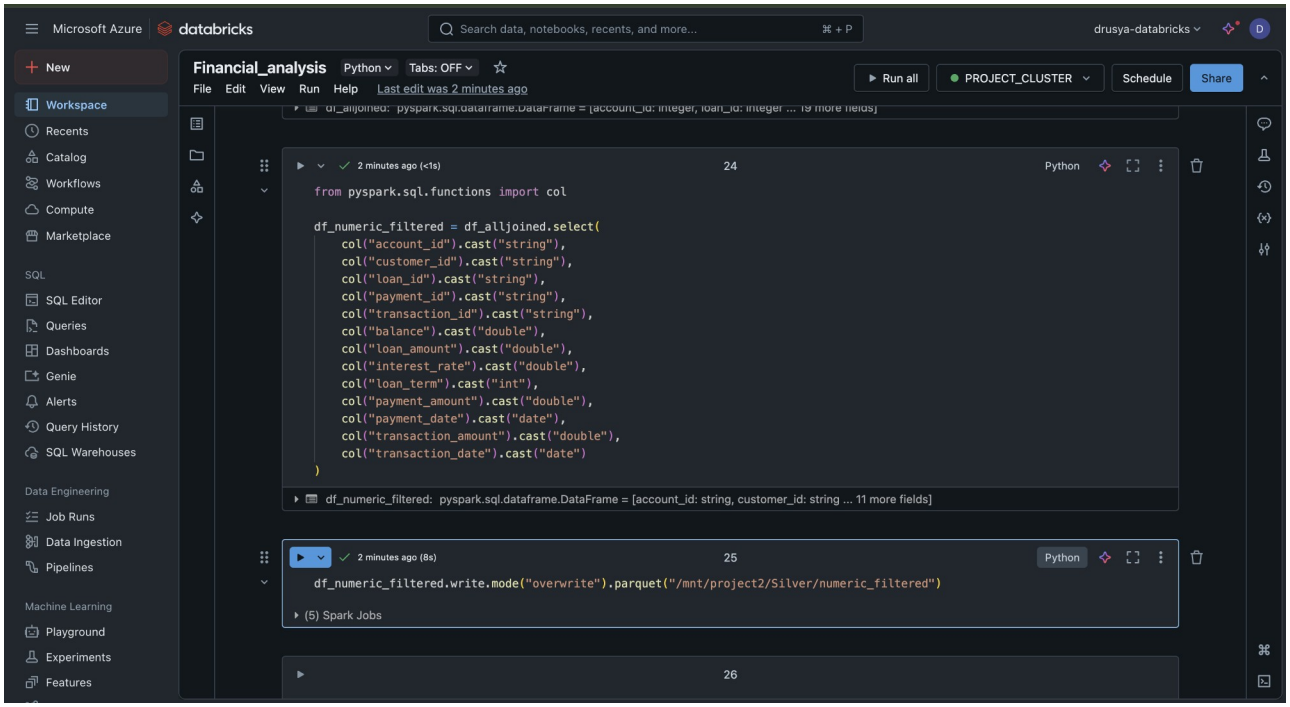
- Loading the data into ADLS in Silver container as Parquet format.

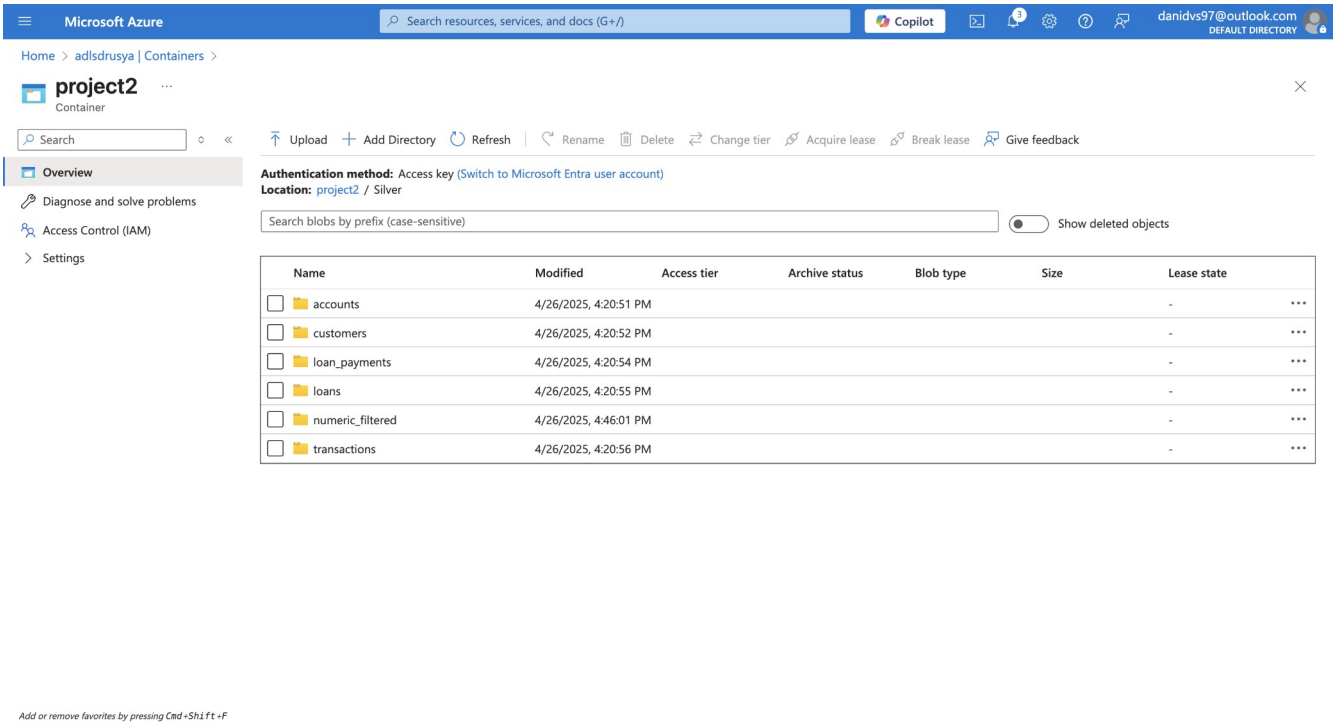




- Join the data frames together and create a new data frame with data including numeric values and dates only from the joined data frame and save it as a Parquet file in the silver container.

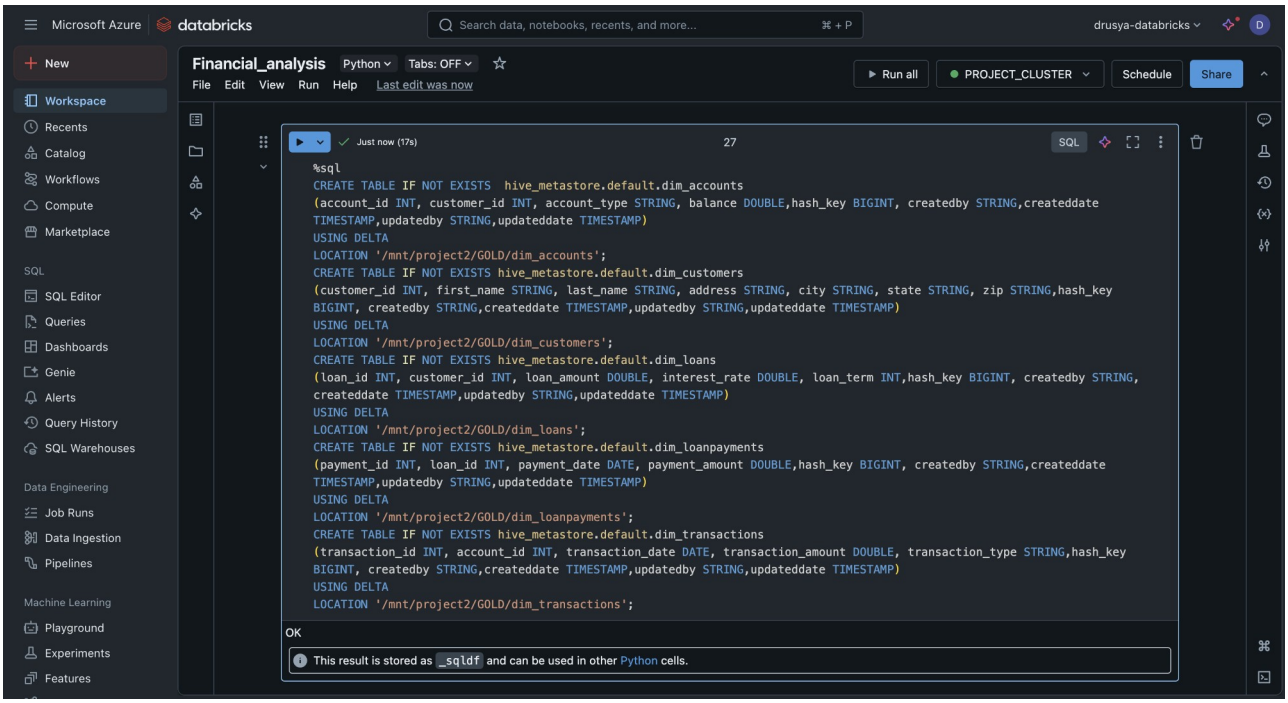




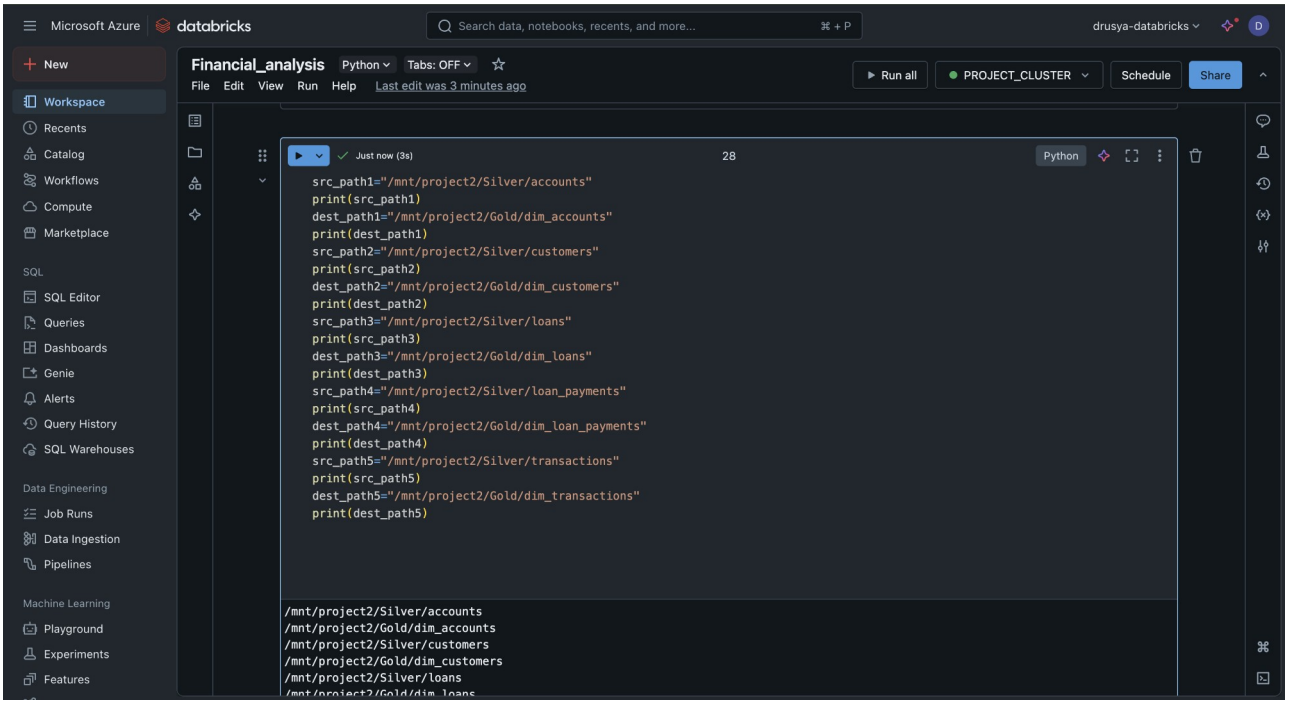


STEP 4: PERFORM SCD TYPE 1 LOGIC ON 5 PARQUET FILES

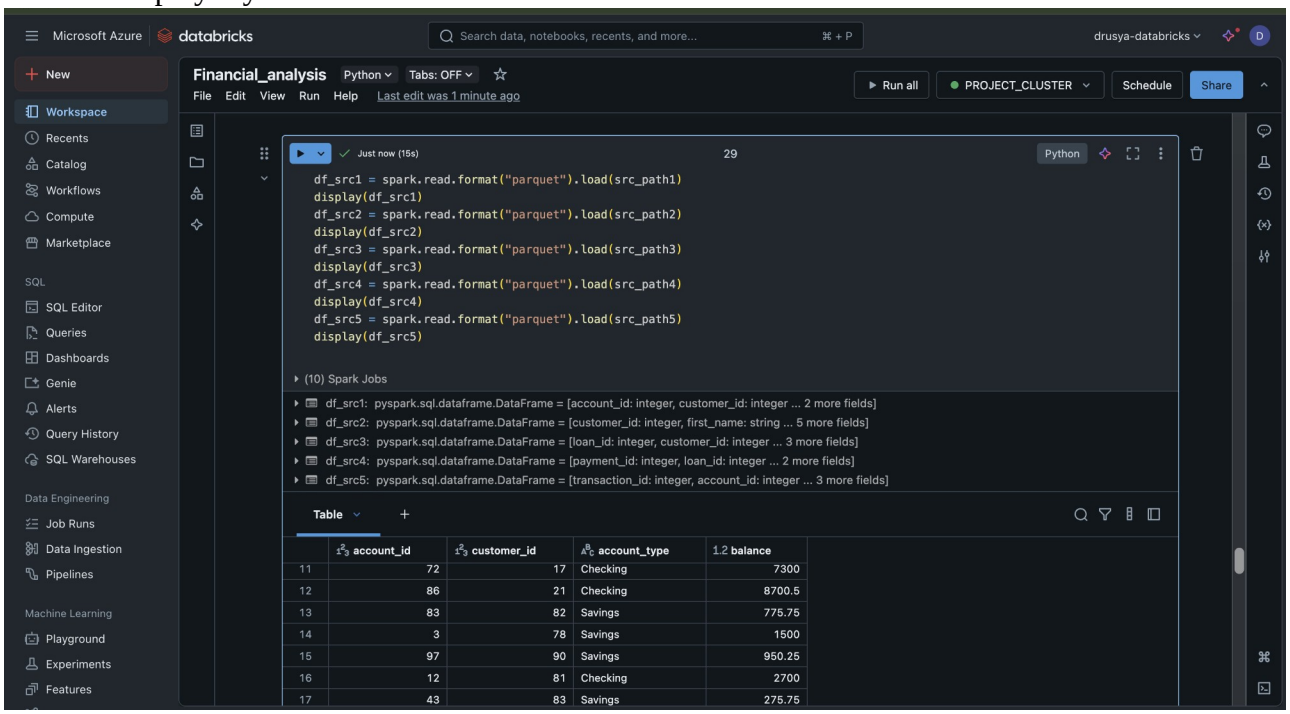
- Create a delta table in hive metastore as default storage location.



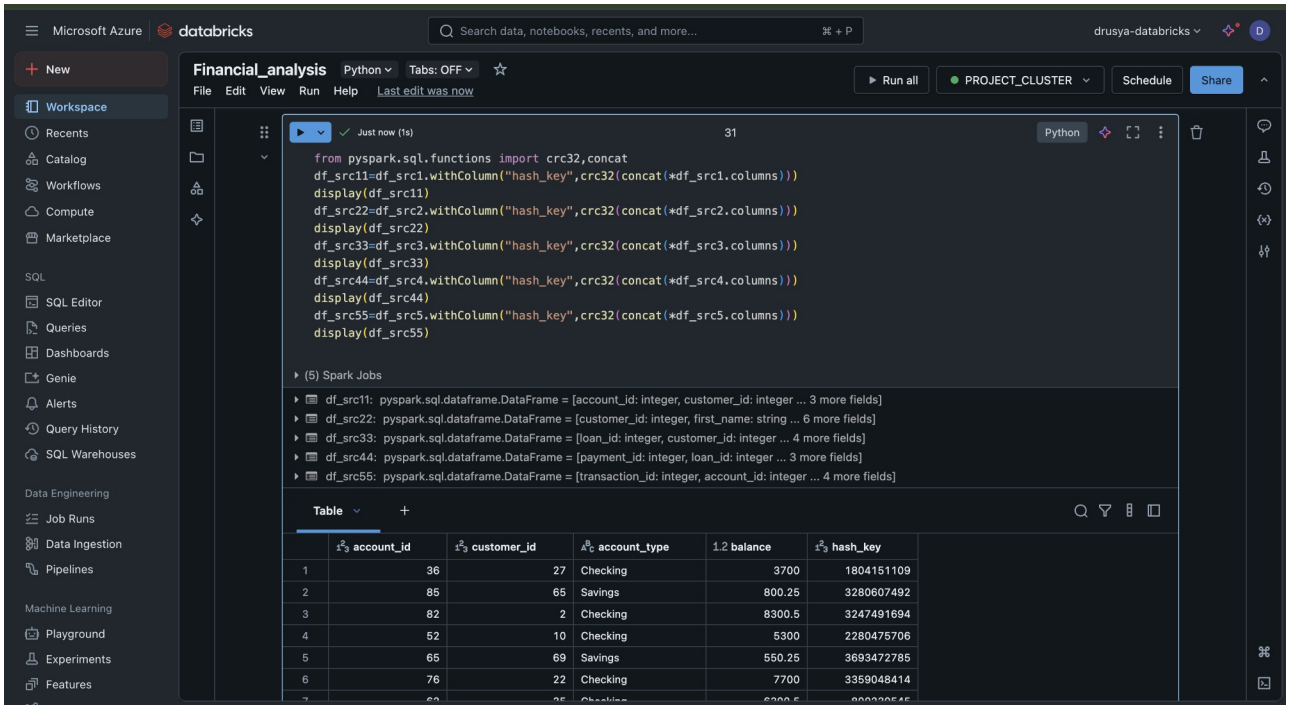
- Now enter a source path and target path



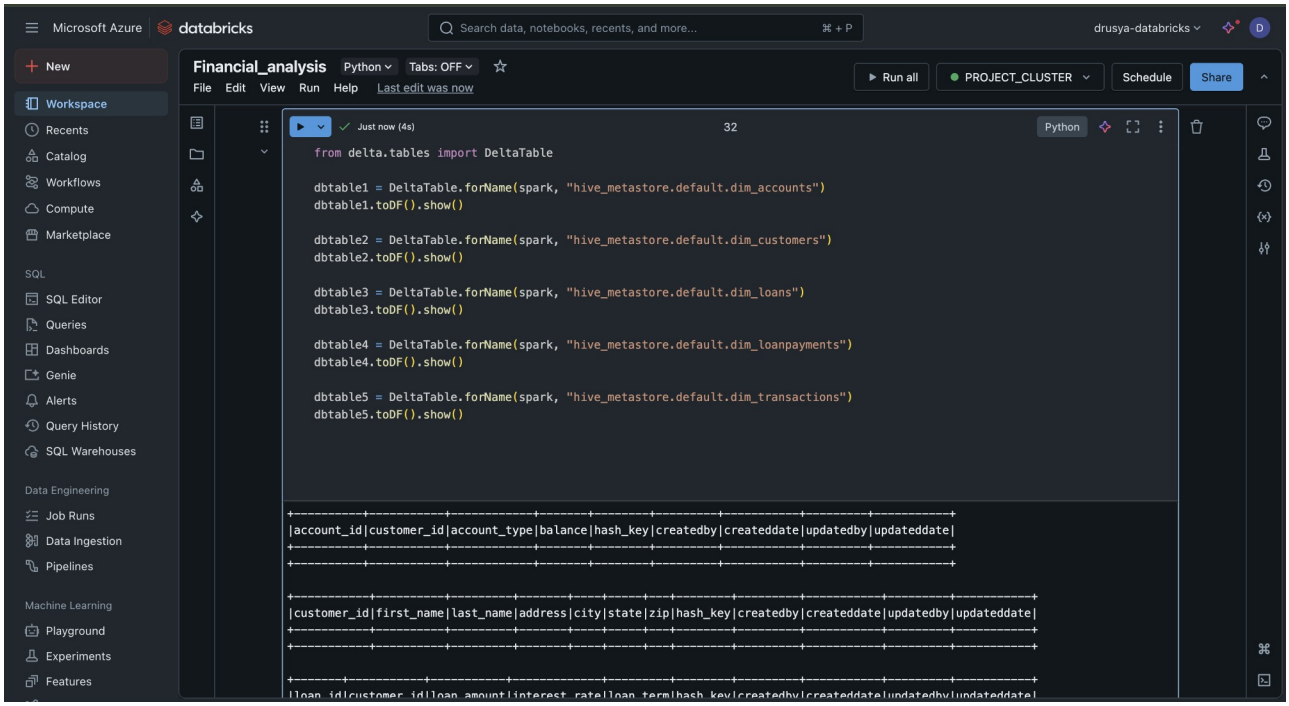
- To display my source file I use the read command.



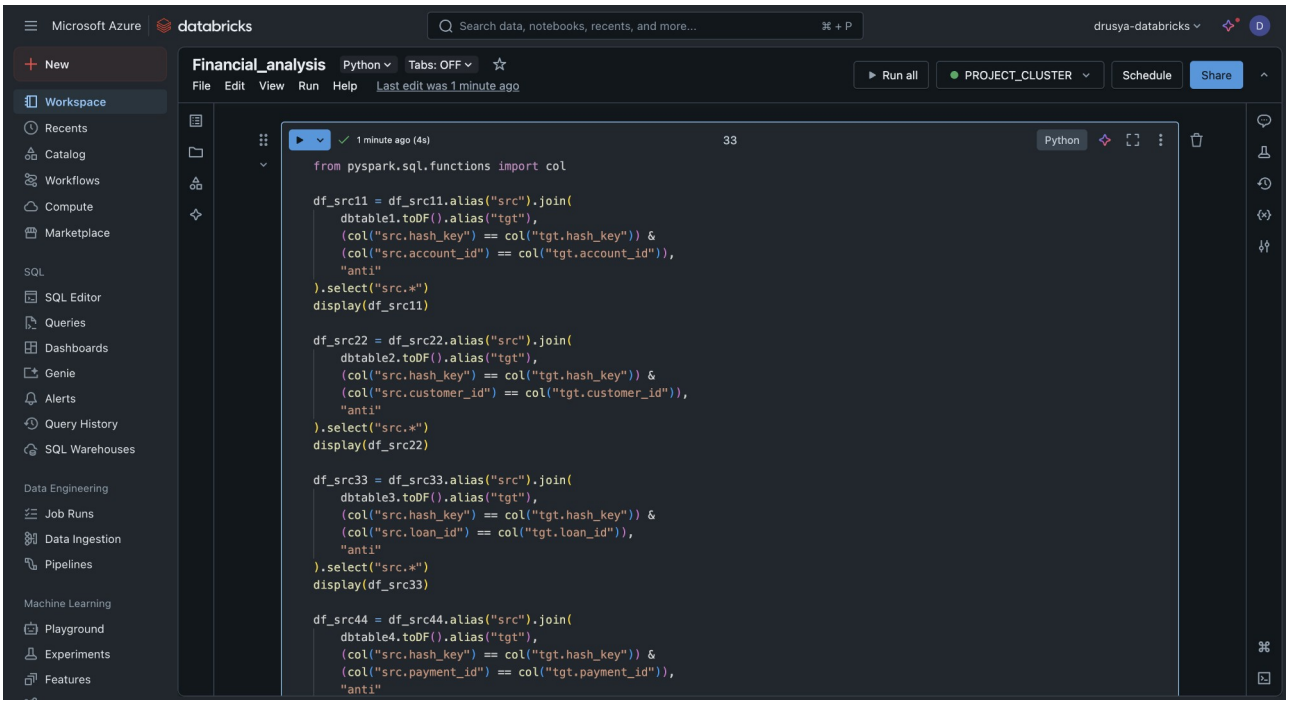
- Now added a new column hashkey so that we can compare the data with other data files to check it is a new or updated data. For that use crc32 command.



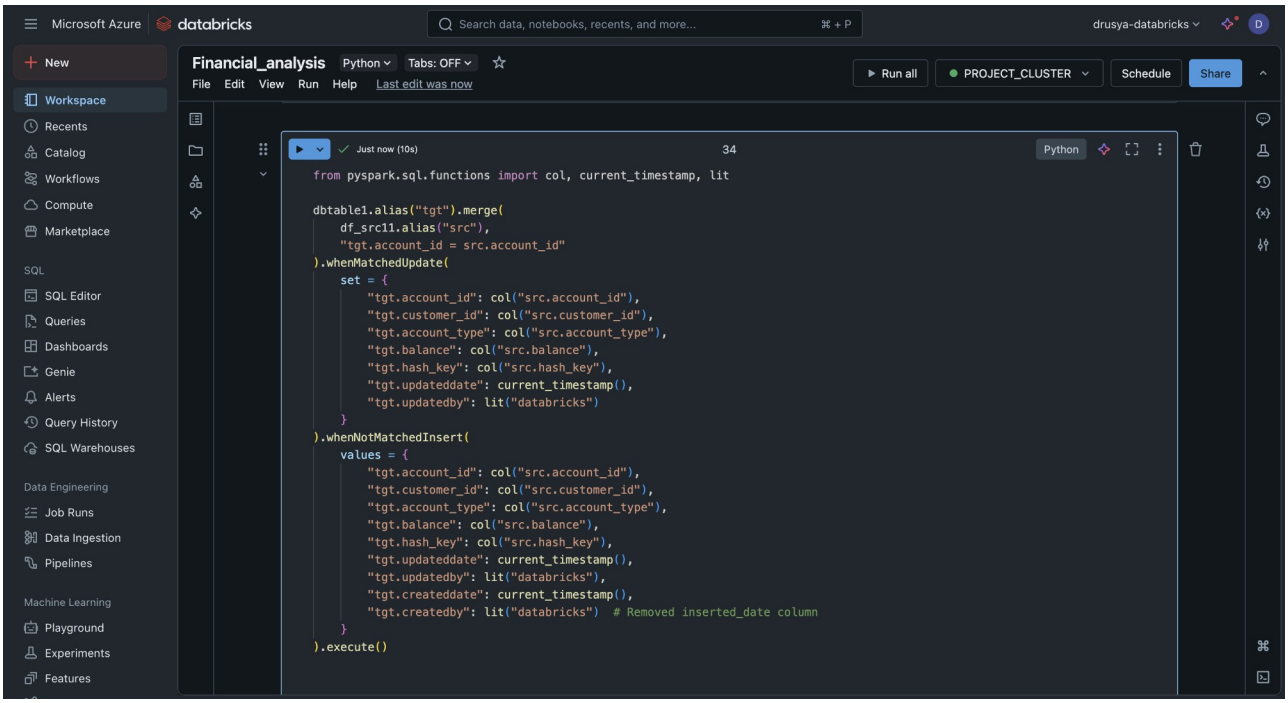
- Now we are converting our delta target table into a data frame and is displaying it.

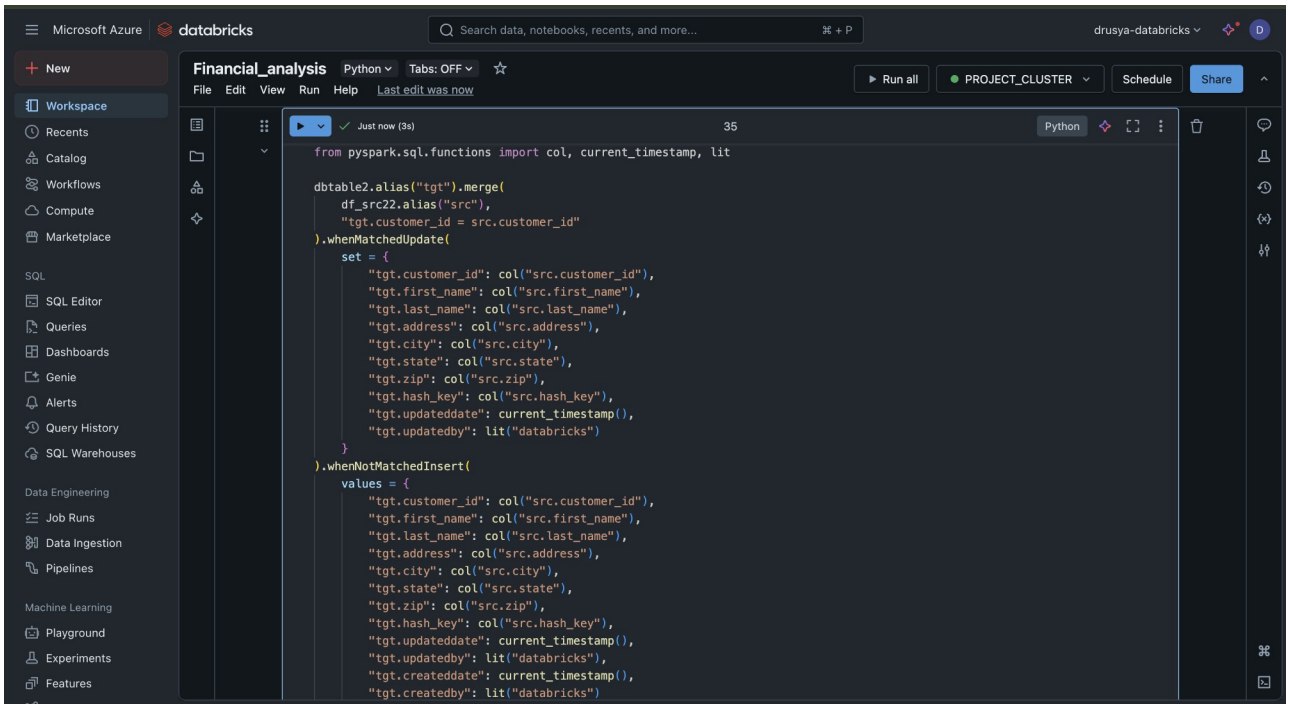


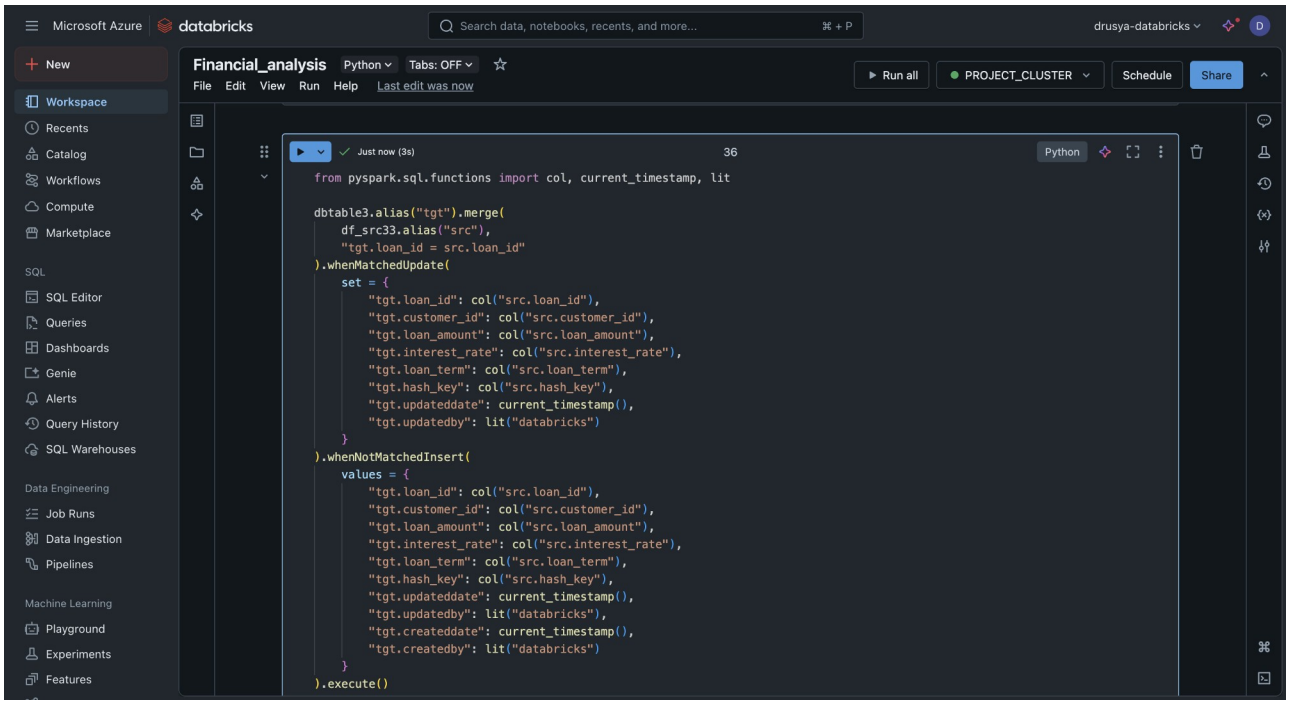
- Now join the source and target based on the ID so that in the next steps we can compare hash key and ID to determine is it an insert or update.

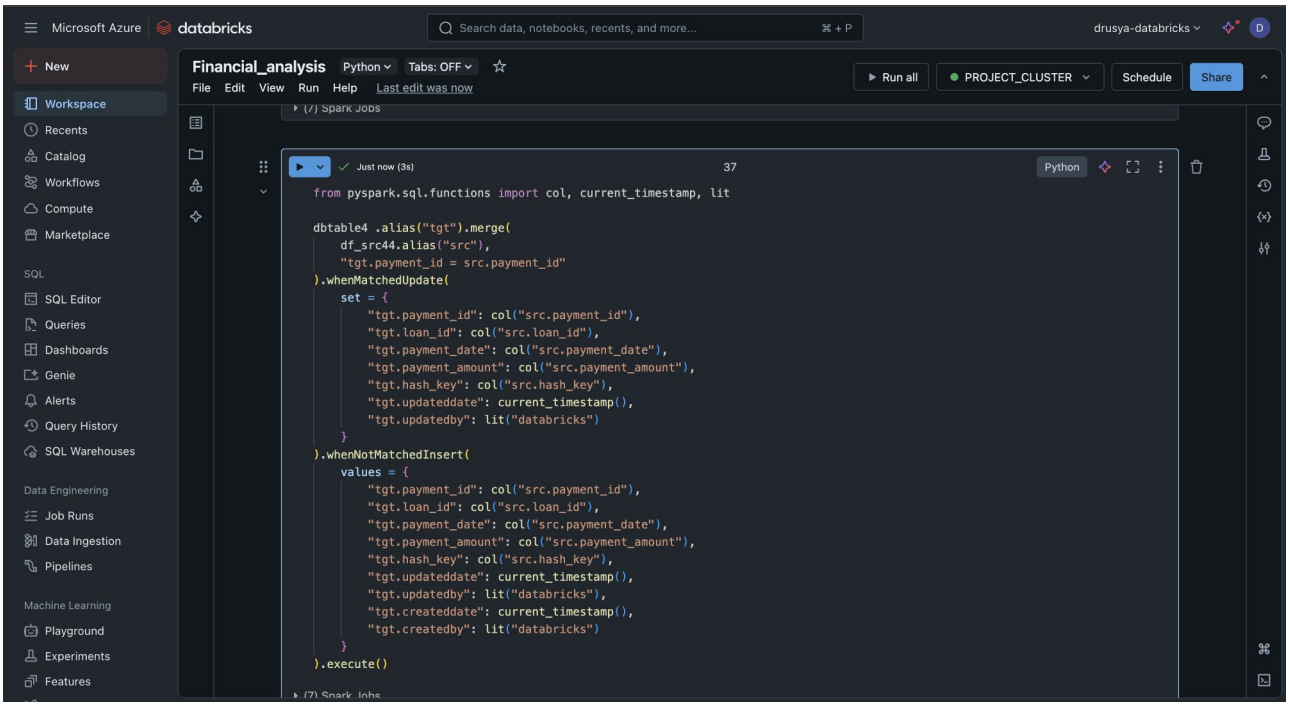


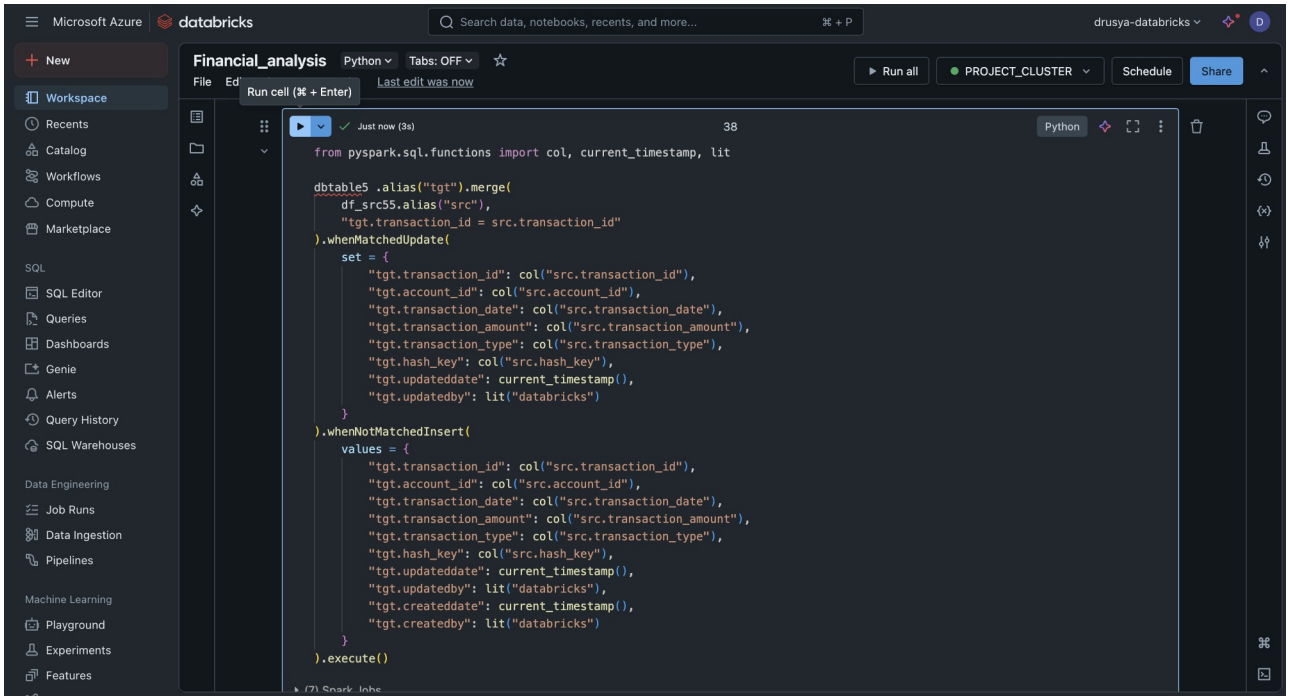
- Based on update or insert new records condition merge the files.


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
New
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help Last edit was now
Run all
PROJECT_CLUSTER
Schedule
Share
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Just now (10s)
34
Python
from pyspark.sql.functions import col, current_timestamp, lit
dbtable1.alias("tgt").merge(
df_src11.alias("src"),
"tgt.account_id = src.account_id"
).whenMatchedUpdate(
set = {
"tgt.account_id": col("src.account_id"),
"tgt.customer_id": col("src.customer_id"),
"tgt.account_type": col("src.account_type"),
"tgt.balance": col("src.balance"),
"tgt.hash_key": col("src.hash_key"),
"tgt.updateddate": current_timestamp(),
"tgt.updatedby": lit("databricks")
}
).whenNotMatchedInsert(
values = {
"tgt.account_id": col("src.account_id"),
"tgt.customer_id": col("src.customer_id"),
"tgt.account_type": col("src.account_type"),
"tgt.balance": col("src.balance"),
"tgt.hash_key": col("src.hash_key"),
"tgt.updateddate": current_timestamp(),
"tgt.updatedby": lit("databricks"),
"tgt.createddate": current_timestamp(),
"tgt.createdby": lit("databricks") # Removed inserted_date column
}
).execute()

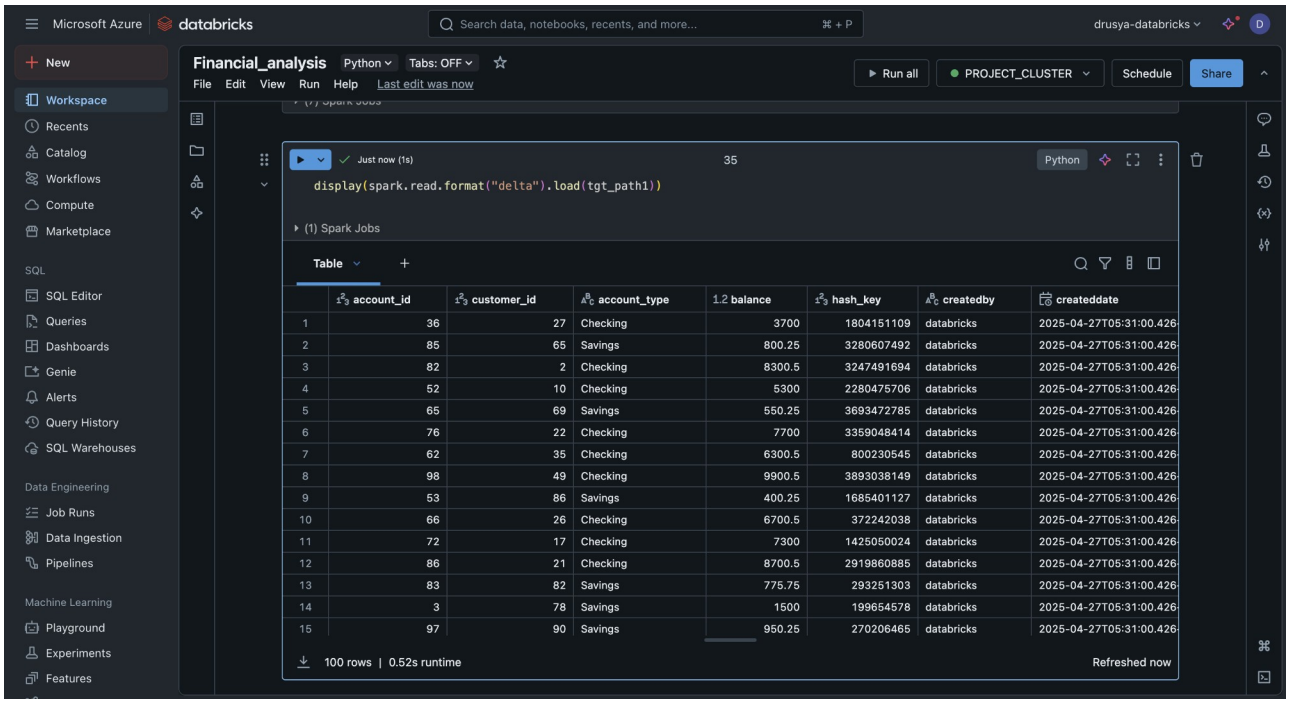

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
New
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help Last edit was now
Run all
PROJECT_CLUSTER
Schedule
Share
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Just now (3s)
35
Python
from pyspark.sql.functions import col, current_timestamp, lit
dbtable2.alias("tgt").merge(
df_src22.alias("src"),
"tgt.customer_id = src.customer_id"
).whenMatchedUpdate(
set = {
"tgt.customer_id": col("src.customer_id"),
"tgt.first_name": col("src.first_name"),
"tgt.last_name": col("src.last_name"),
"tgt.address": col("src.address"),
"tgt.city": col("src.city"),
"tgt.state": col("src.state"),
"tgt.zip": col("src.zip"),
"tgt.hash_key": col("src.hash_key"),
"tgt.updateddate": current_timestamp(),
"tgt.updatedby": lit("databricks")
}
).whenNotMatchedInsert(
values = {
"tgt.customer_id": col("src.customer_id"),
"tgt.first_name": col("src.first_name"),
"tgt.last_name": col("src.last_name"),
"tgt.address": col("src.address"),
"tgt.city": col("src.city"),
"tgt.state": col("src.state"),
"tgt.zip": col("src.zip"),
"tgt.hash_key": col("src.hash_key"),
"tgt.updateddate": current_timestamp(),
"tgt.updatedby": lit("databricks"),
"tgt.createddate": current_timestamp(),
"tgt.createdby": lit("databricks")

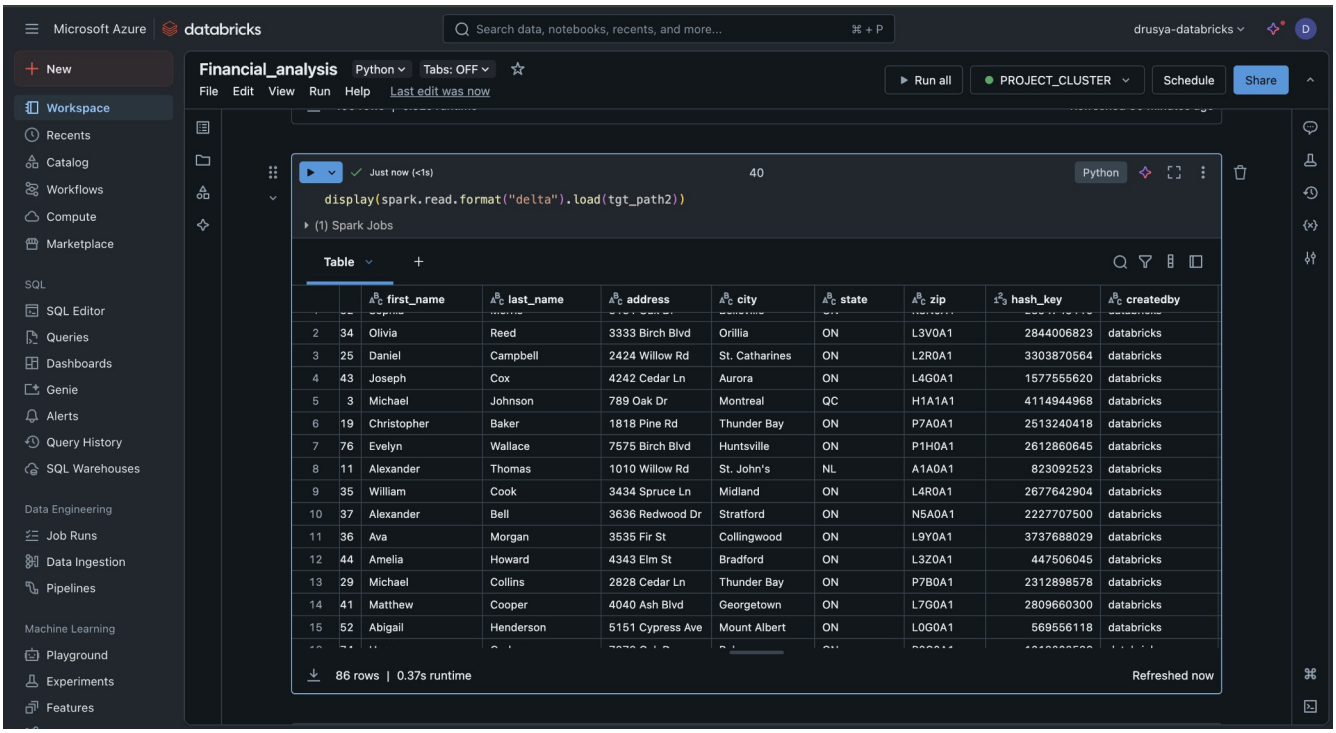

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
New
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help Last edit was now
Run all
PROJECT_CLUSTER
Schedule
Share
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Just now (3s)
36
Python
from pyspark.sql.functions import col, current_timestamp, lit
dbtable3.alias("tgt").merge(
df_src33.alias("src"),
"tgt.loan_id = src.loan_id"
).whenMatchedUpdate(
set = {
"tgt.loan_id": col("src.loan_id"),
"tgt.customer_id": col("src.customer_id"),
"tgt.loan_amount": col("src.loan_amount"),
"tgt.interest_rate": col("src.interest_rate"),
"tgt.loan_term": col("src.loan_term"),
"tgt.hash_key": col("src.hash_key"),
"tgt.updateddate": current_timestamp(),
"tgt.updatedby": lit("databricks")
}
).whenNotMatchedInsert(
values = {
"tgt.loan_id": col("src.loan_id"),
"tgt.customer_id": col("src.customer_id"),
"tgt.loan_amount": col("src.loan_amount"),
"tgt.interest_rate": col("src.interest_rate"),
"tgt.loan_term": col("src.loan_term"),
"tgt.hash_key": col("src.hash_key"),
"tgt.updateddate": current_timestamp(),
"tgt.updatedby": lit("databricks"),
"tgt.createddate": current_timestamp(),
"tgt.createdby": lit("databricks")
}
).execute()

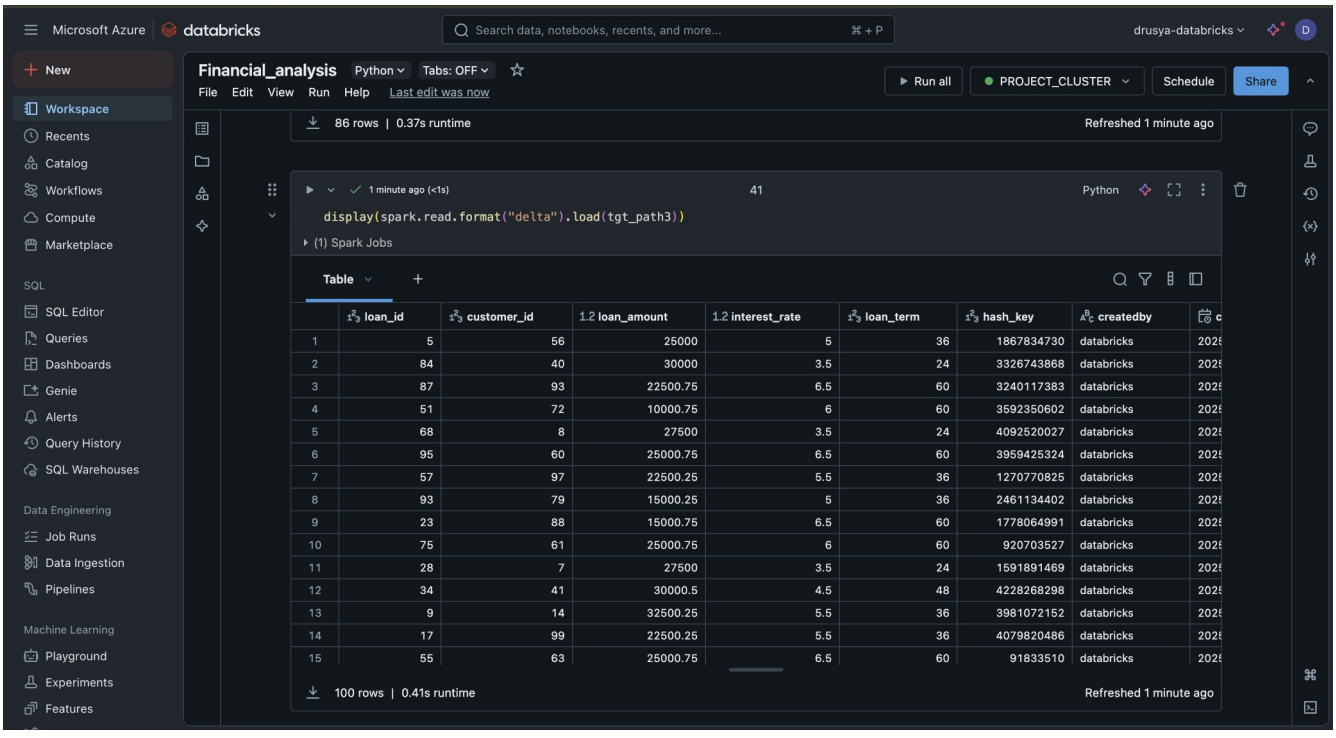

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
New
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help Last edit was now
Run all
PROJECT_CLUSTER
Schedule
Share
(7) Spark Jobs
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Just now (3s)
37
Python
from pyspark.sql.functions import col, current_timestamp, lit
dbtable4 .alias("tgt").merge(
df_src44.alias("src"),
"tgt.payment_id = src.payment_id"
).whenMatchedUpdate(
set = {
"tgt.payment_id": col("src.payment_id"),
"tgt.loan_id": col("src.loan_id"),
"tgt.payment_date": col("src.payment_date"),
"tgt.payment_amount": col("src.payment_amount"),
"tgt.hash_key": col("src.hash_key"),
"tgt.updateddate": current_timestamp(),
"tgt.updatedby": lit("databricks")
}
).whenNotMatchedInsert(
values = {
"tgt.payment_id": col("src.payment_id"),
"tgt.loan_id": col("src.loan_id"),
"tgt.payment_date": col("src.payment_date"),
"tgt.payment_amount": col("src.payment_amount"),
"tgt.hash_key": col("src.hash_key"),
"tgt.updateddate": current_timestamp(),
"tgt.updatedby": lit("databricks"),
"tgt.createddate": current_timestamp(),
"tgt.createdby": lit("databricks")
}
).execute()
(7) Spark Jobs

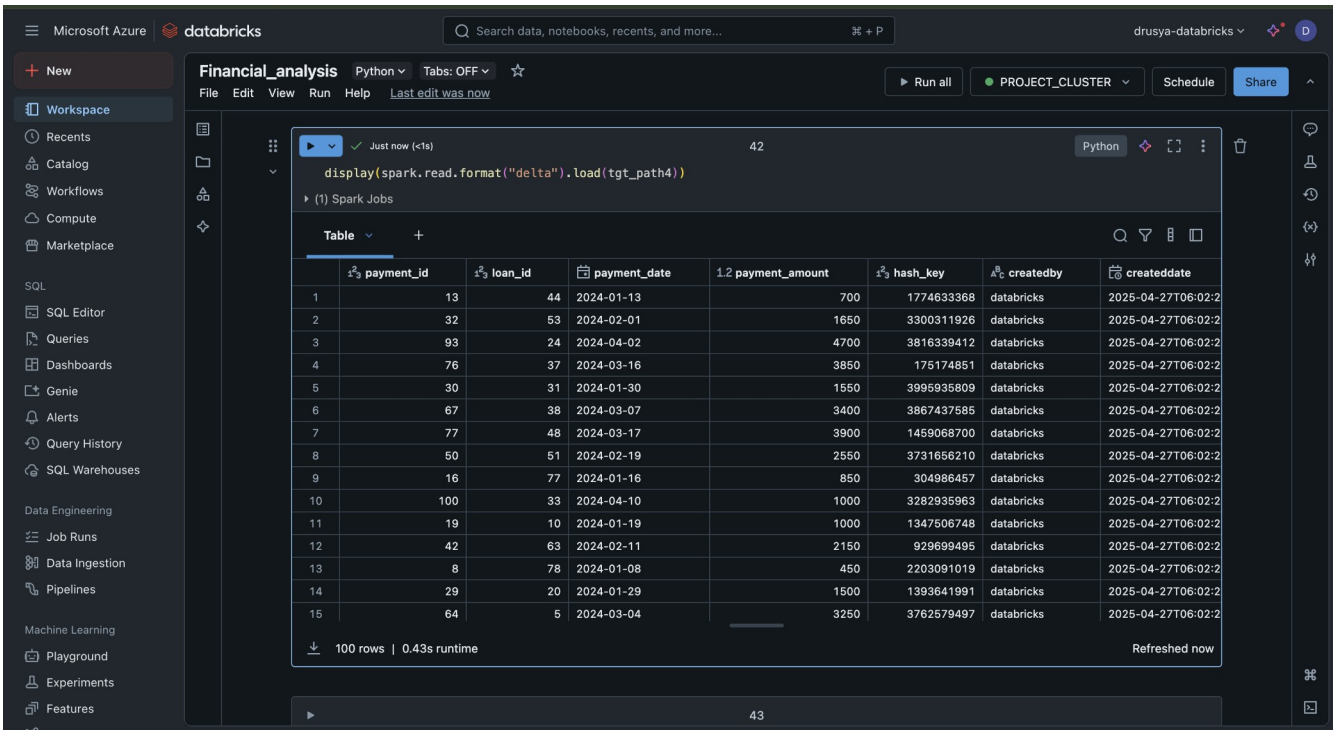


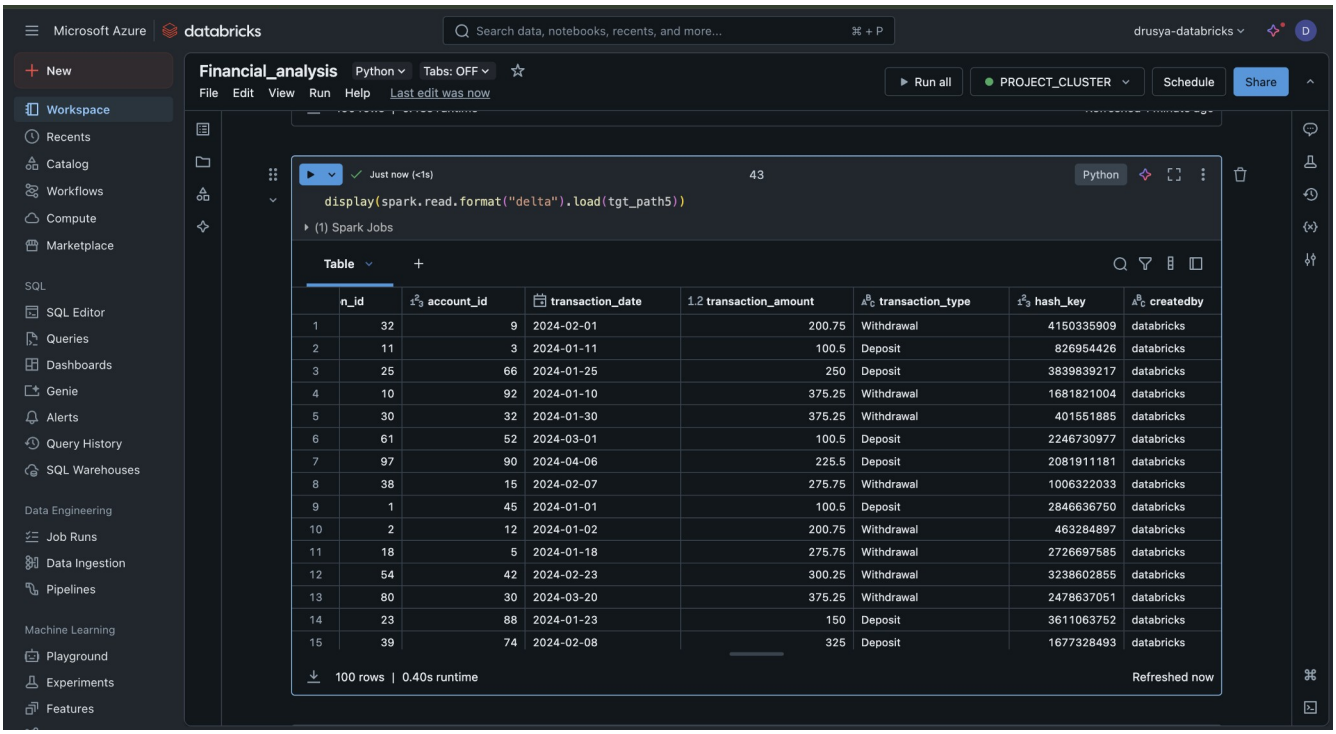
- Now read the target file.




Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
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
Machine Learning
Playground
Experiments
Features
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help
Last edit was now
Run all
PROJECT_CLUSTER
Schedule
Share
Just now (<1s)
40
Python
display(spark.read.format("delta").load(tgt_path2))
(1) Spark Jobs
Table
first_name
last_name
address
city
state
zip
hash_key
createdby
2 34 Olivia Reed 3333 Birch Blvd Orillia ON L3V0A1 2844006823 databricks
3 25 Daniel Campbell 2424 Willow Rd St. Catharines ON L2R0A1 3303870564 databricks
4 43 Joseph Cox 4242 Cedar Ln Aurora ON L4G0A1 1577555620 databricks
5 3 Michael Johnson 789 Oak Dr Montreal QC H1A1A1 4114944968 databricks
6 19 Christopher Baker 1818 Pine Rd Thunder Bay ON P7A0A1 2513240418 databricks
7 76 Evelyn Wallace 7575 Birch Blvd Huntsville ON P1H0A1 2612860645 databricks
8 11 Alexander Thomas 1010 Willow Rd St. John's NL A1A0A1 823092523 databricks
9 35 William Cook 3434 Spruce Ln Midland ON L4R0A1 2677642904 databricks
10 37 Alexander Bell 3636 Redwood Dr Stratford ON N5A0A1 2227707500 databricks
11 36 Ava Morgan 3535 Fir St Collingwood ON L9Y0A1 3737688029 databricks
12 44 Amelia Howard 4343 Elm St Bradford ON L3Z0A1 447506045 databricks
13 29 Michael Collins 2828 Cedar Ln Thunder Bay ON P7B0A1 2312898578 databricks
14 41 Matthew Cooper 4040 Ash Blvd Georgetown ON L7G0A1 2809660300 databricks
15 52 Abigail Henderson 5151 Cypress Ave Mount Albert ON L0G0A1 569556118 databricks
86 rows | 0.37s runtime
Refreshed now

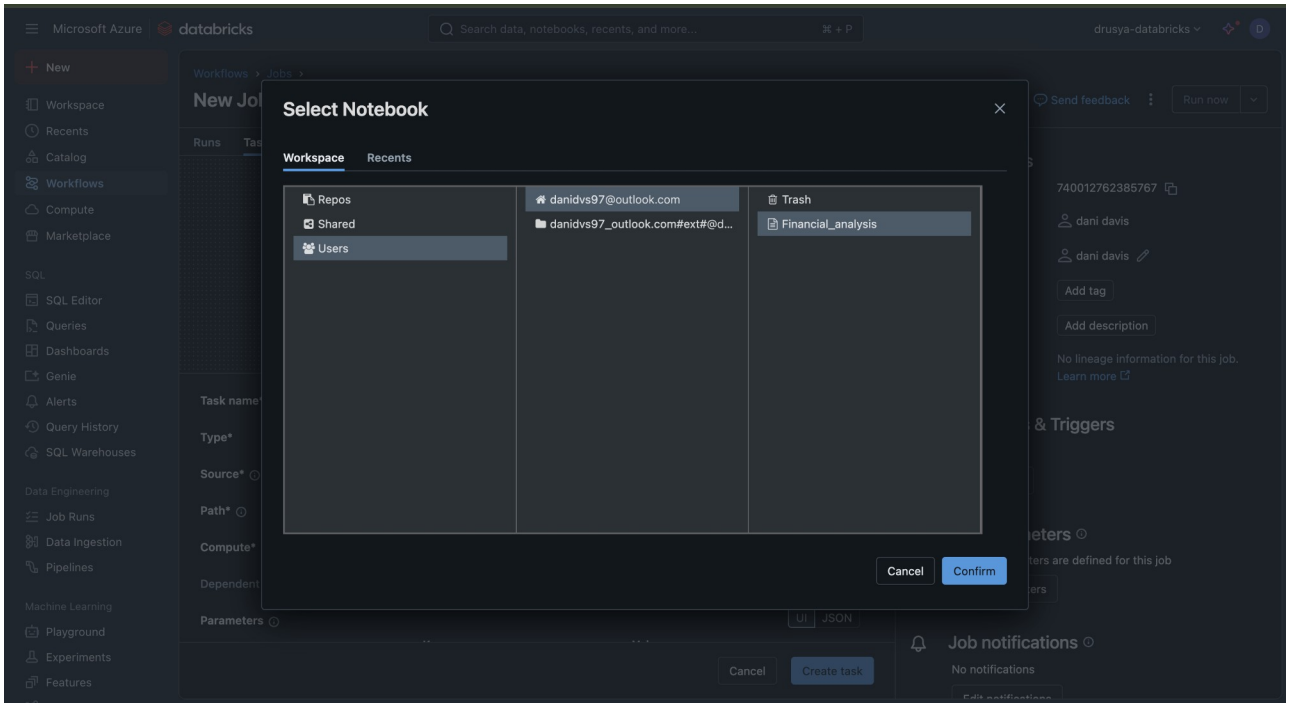

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help
Last edit was now
Run all
PROJECT_CLUSTER
Schedule
Share
86 rows | 0.37s runtime
Refreshed 1 minute ago
1 minute ago (<1s)
41
Python
display(spark.read.format("delta").load(tgt_path3))
(1) Spark Jobs
Table
loan_id
customer_id
loan_amount
interest_rate
loan_term
hash_key
createdby
1 5 56 25000 5 36 1867834730 databricks
2 84 40 30000 3.5 24 3326743868 databricks
3 87 93 22500.75 6.5 60 3240117383 databricks
4 51 72 10000.75 6 60 3592350602 databricks
5 68 8 27500 3.5 24 4092520027 databricks
6 95 60 25000.75 6.5 60 3959425324 databricks
7 57 97 22500.25 5.5 36 1270770825 databricks
8 93 79 15000.25 5 36 2461134402 databricks
9 23 88 15000.75 6.5 60 1778064991 databricks
10 75 61 25000.75 6 60 920703527 databricks
11 28 7 27500 3.5 24 1591891469 databricks
12 34 41 30000.5 4.5 48 4228268298 databricks
13 9 14 32500.25 5.5 36 3981072152 databricks
14 17 99 22500.25 5.5 36 4079820486 databricks
15 55 63 25000.75 6.5 60 91833510 databricks
100 rows | 0.41s runtime
Refreshed 1 minute ago

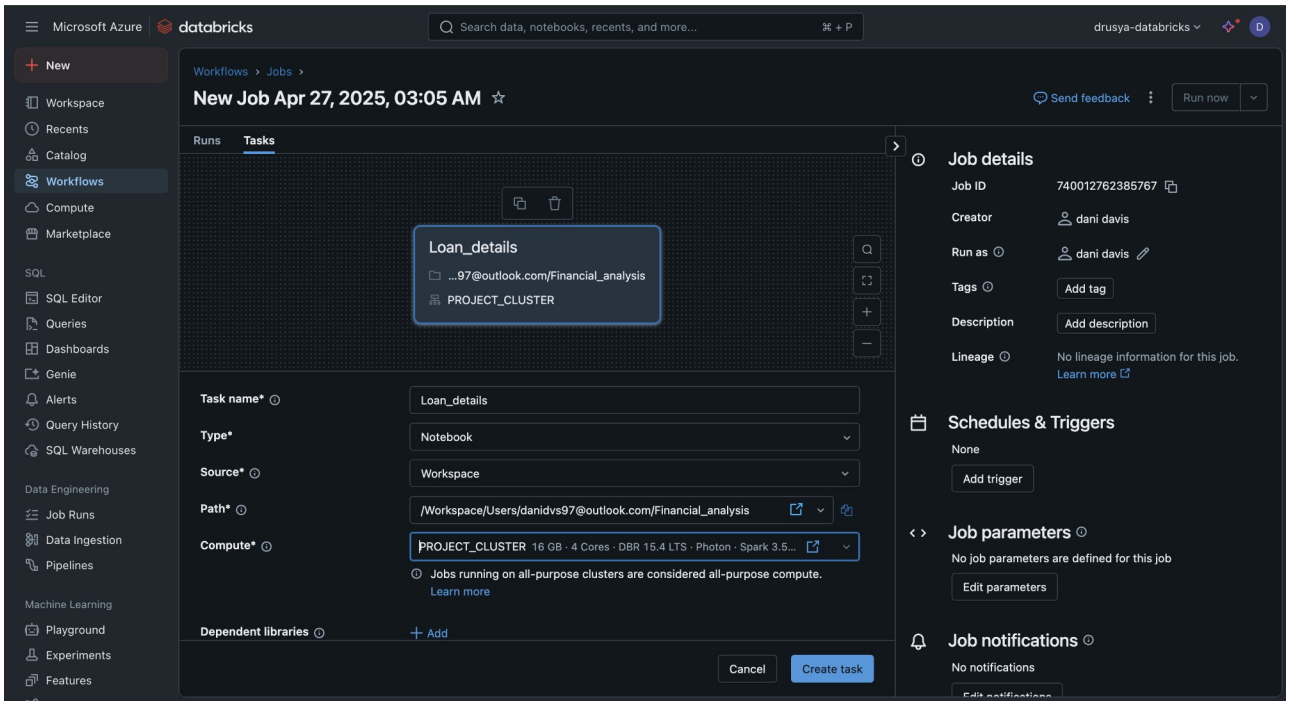

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help
Last edit was now
Run all
PROJECT_CLUSTER
Schedule
Share
Just now (<1s)
42
Python
display(spark.read.format("delta").load(tgt_path4))
(1) Spark Jobs
Table
payment_id
loan_id
payment_date
payment_amount
hash_key
createdby
createddate
1 13 44 2024-01-13 700 1774633368 databricks 2025-04-27T06:02:2
2 32 53 2024-02-01 1650 3300311926 databricks 2025-04-27T06:02:2
3 93 24 2024-04-02 4700 3816339412 databricks 2025-04-27T06:02:2
4 76 37 2024-03-16 3850 175174851 databricks 2025-04-27T06:02:2
5 30 31 2024-01-30 1550 3995935809 databricks 2025-04-27T06:02:2
6 67 38 2024-03-07 3400 3867437585 databricks 2025-04-27T06:02:2
7 77 48 2024-03-17 3900 1459068700 databricks 2025-04-27T06:02:2
8 50 51 2024-02-19 2550 3731656210 databricks 2025-04-27T06:02:2
9 16 77 2024-01-16 850 304986457 databricks 2025-04-27T06:02:2
10 100 33 2024-04-10 1000 3282935963 databricks 2025-04-27T06:02:2
11 19 10 2024-01-19 1000 1347506748 databricks 2025-04-27T06:02:2
12 42 63 2024-02-11 2150 929699495 databricks 2025-04-27T06:02:2
13 8 78 2024-01-08 450 2203091019 databricks 2025-04-27T06:02:2
14 29 20 2024-01-29 1500 1393641991 databricks 2025-04-27T06:02:2
15 64 5 2024-03-04 3250 3762579497 databricks 2025-04-27T06:02:2
100 rows | 0.43s runtime
Refreshed now
43

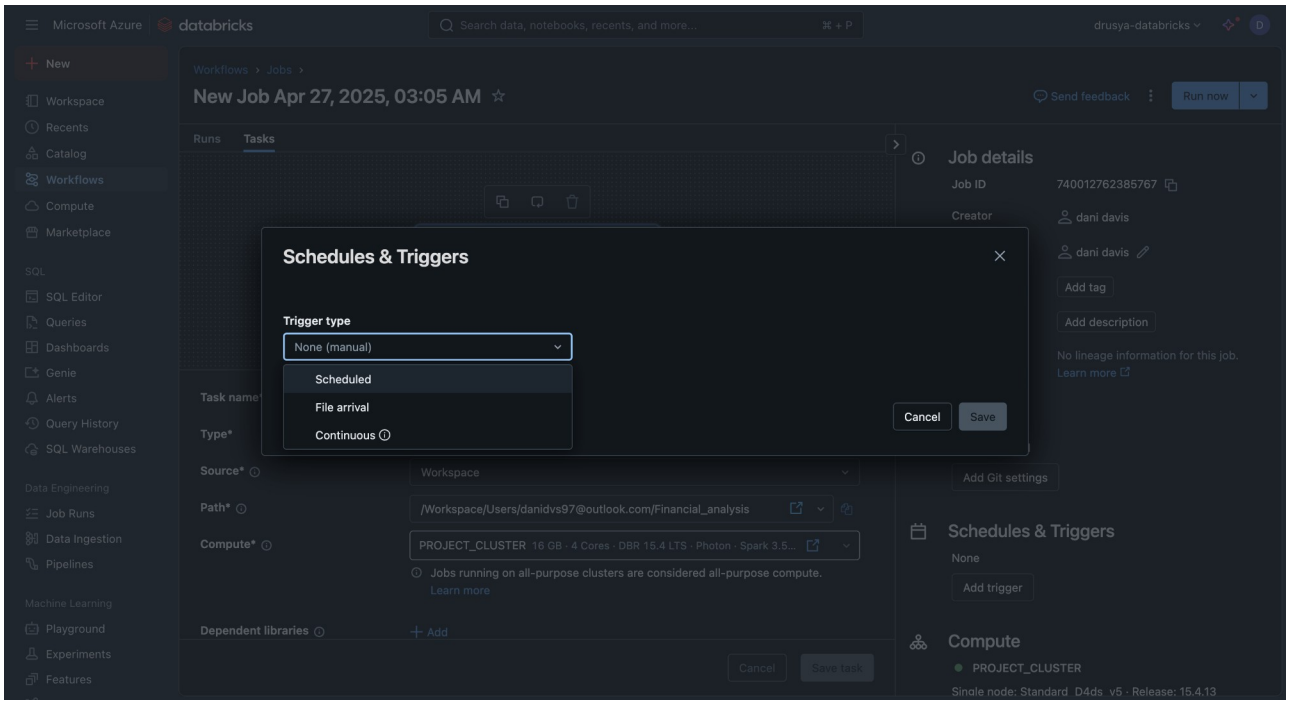

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Financial_analysis
Python
Tabs: OFF
File Edit View Run Help
Last edit was now
Run all
PROJECT_CLUSTER
Schedule
Share
Just now (<1s)
43
Python
display(spark.read.format("delta").load(tgt_path5))
(1) Spark Jobs
Table
n_id
account_id
transaction_date
transaction_amount
transaction_type
hash_key
createdby
1 32 9 2024-02-01 200.75 Withdrawal 4150335909 databricks
2 11 3 2024-01-11 100.5 Deposit 826954426 databricks
3 25 66 2024-01-25 250 Deposit 3839839217 databricks
4 10 92 2024-01-10 375.25 Withdrawal 1681821004 databricks
5 30 32 2024-01-30 375.25 Withdrawal 401551885 databricks
6 61 52 2024-03-01 100.5 Deposit 2246730977 databricks
7 97 90 2024-04-06 225.5 Deposit 2081911181 databricks
8 38 15 2024-02-07 275.75 Withdrawal 1006322033 databricks
9 1 45 2024-01-01 100.5 Deposit 2846636750 databricks
10 2 12 2024-01-02 200.75 Withdrawal 463284897 databricks
11 18 5 2024-01-18 275.75 Withdrawal 2726697585 databricks
12 54 42 2024-02-23 300.25 Withdrawal 3238602855 databricks
13 80 30 2024-03-20 375.25 Withdrawal 2478637051 databricks
14 23 88 2024-01-23 150 Deposit 3611063752 databricks
15 39 74 2024-02-08 325 Deposit 1677328493 databricks
100 rows | 0.40s runtime
Refreshed now

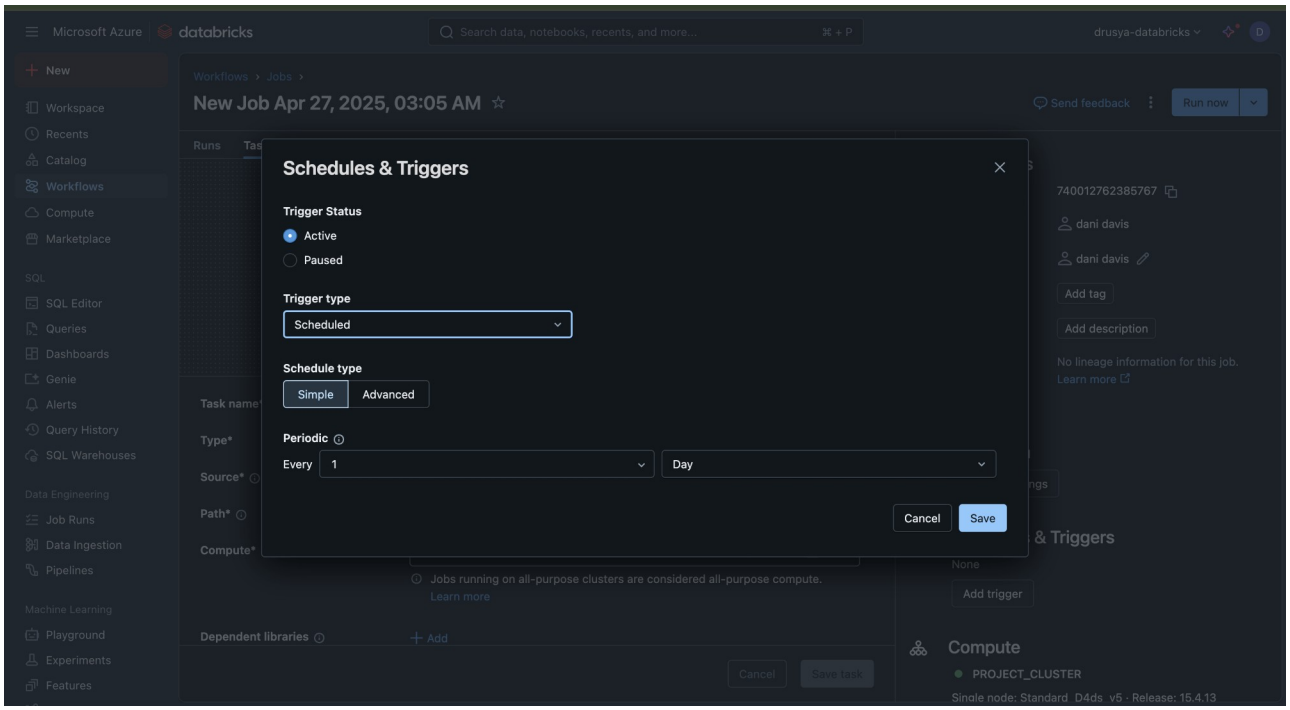
STEP 4: SCHEDULING THE NOTEBOOK USING WORKFLOWS

- Go to the workflows →Create jobs



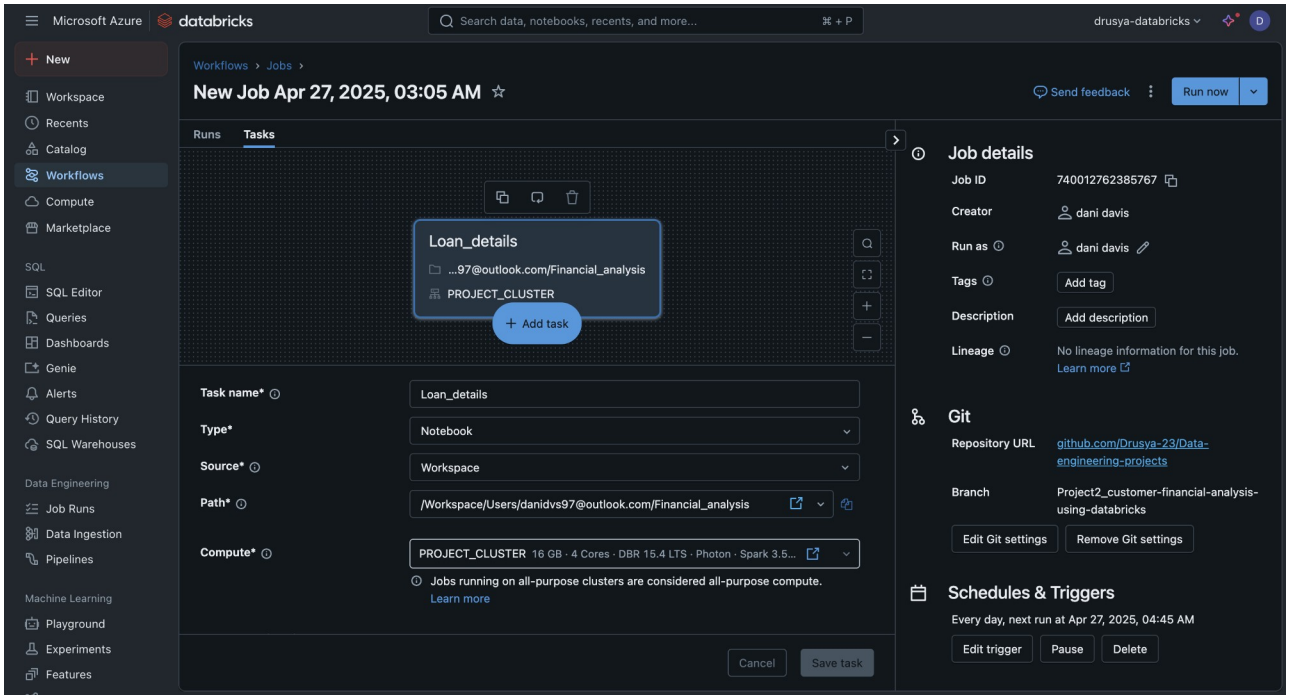


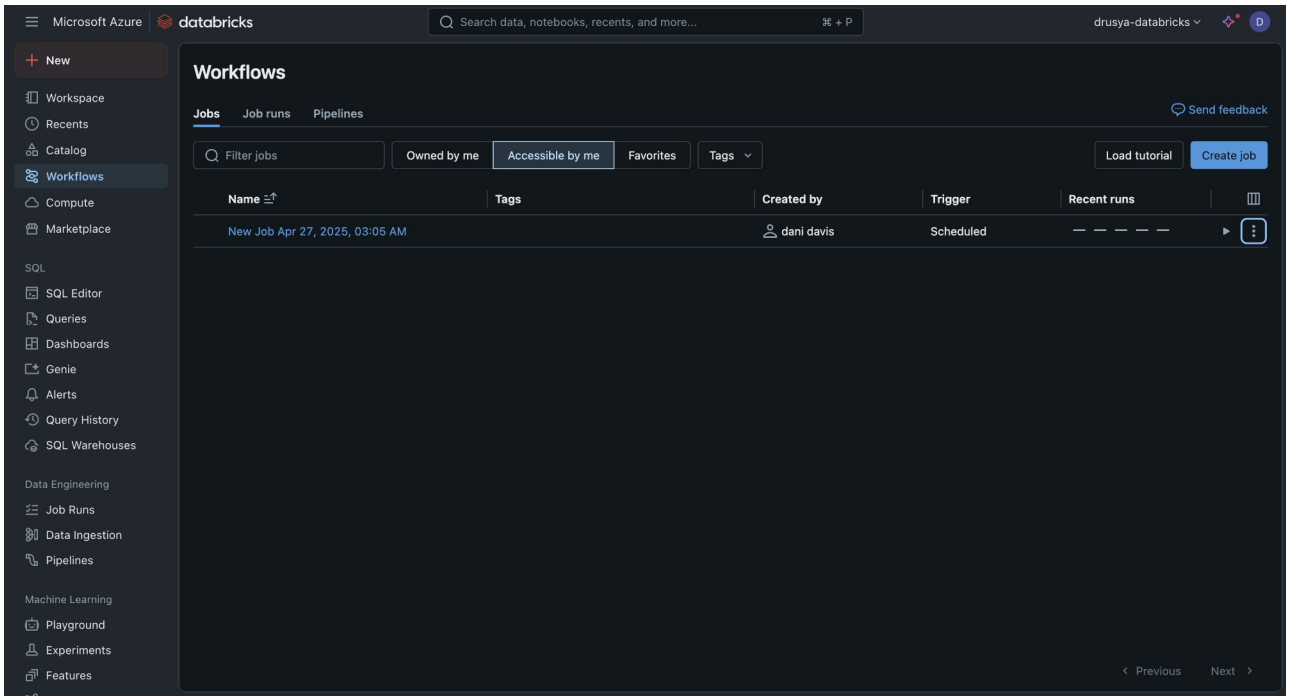

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Workflows › Jobs ›
New Job Apr 27, 2025, 03:05 AM
Send feedback
Run now
Runs
Tasks
Job details
Job ID
740012762385767
Creator
dani davis
dani davis
Add tag
Add description
No lineage information for this job.
Learn more
Schedules & Triggers
Trigger type
None (manual)
Scheduled
File arrival
Continuous
Cancel
Save
Task name
Type*
Source*
Workspace
Path*
/Workspace/Users/danidvs97@outlook.com/Financial_analysis
Compute*
PROJECT_CLUSTER 16 GB · 4 Cores · DBR 15.4 LTS · Photon · Spark 3.5...
Jobs running on all-purpose clusters are considered all-purpose compute.
Learn more
Dependent libraries
Add
Cancel
Save task
Add Git settings
Schedules & Triggers
None
Add trigger
Compute
PROJECT_CLUSTER
Single node: Standard_D4ds_v5 · Release: 15.4.13


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
drusya-databricks
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
Machine Learning
Playground
Experiments
Features
Workflows › Jobs ›
New Job Apr 27, 2025, 03:05 AM
Send feedback
Run now
Schedules & Triggers
Trigger Status
Active
Paused
Trigger type
Scheduled
Schedule type
Simple
Advanced
Periodic
Every
1
Day
Cancel
Save
740012762385767
dani davis
dani davis
Add tag
Add description
No lineage information for this job.
Learn more
Task name
Type*
Source*
Path*
Compute*
Jobs running on all-purpose clusters are considered all-purpose compute.
Learn more
Dependent libraries
Add
Cancel
Save task
& Triggers
None
Add trigger
Compute
PROJECT_CLUSTER
Single node: Standard_D4ds_v5 · Release: 15.4.13

•

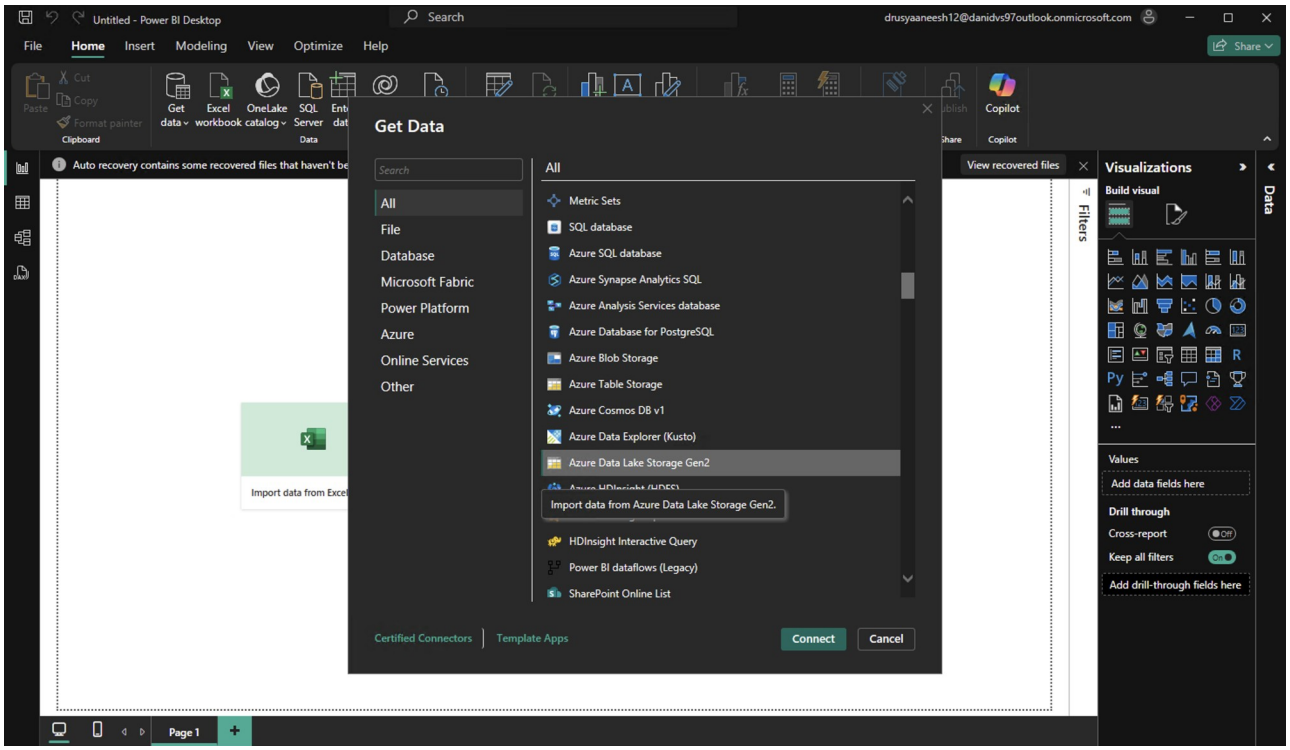
- Added the git hub repository details also.

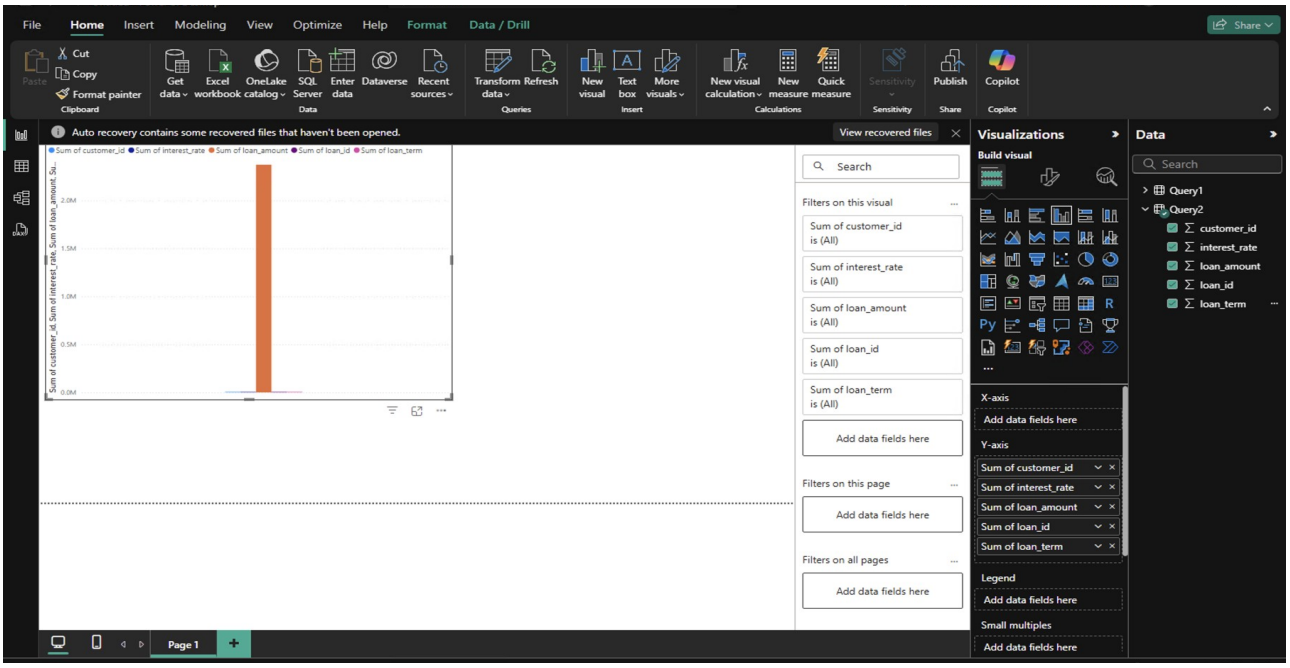


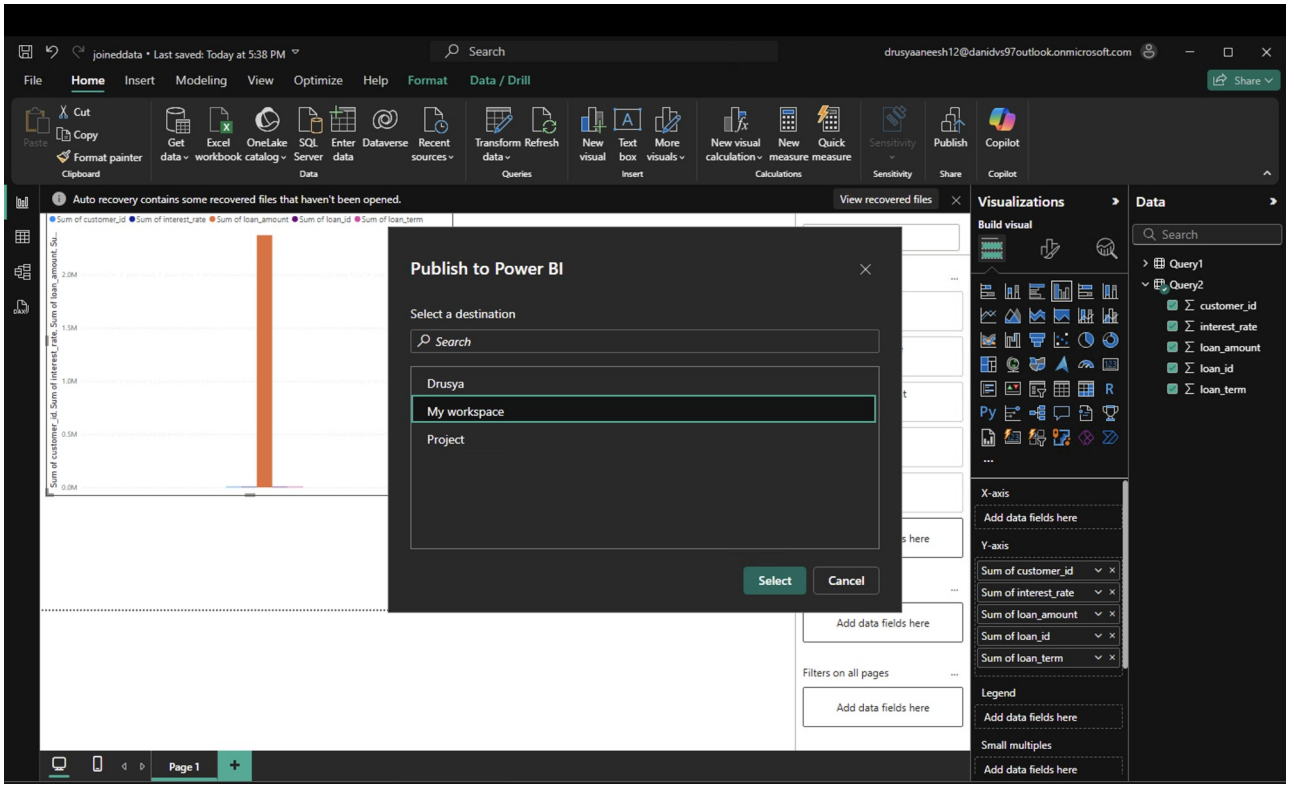


STEP 5: CREATE A REPORT OF THE JOINED FILE IN POWER BI AND PUBLISH IT IN FABRIC.

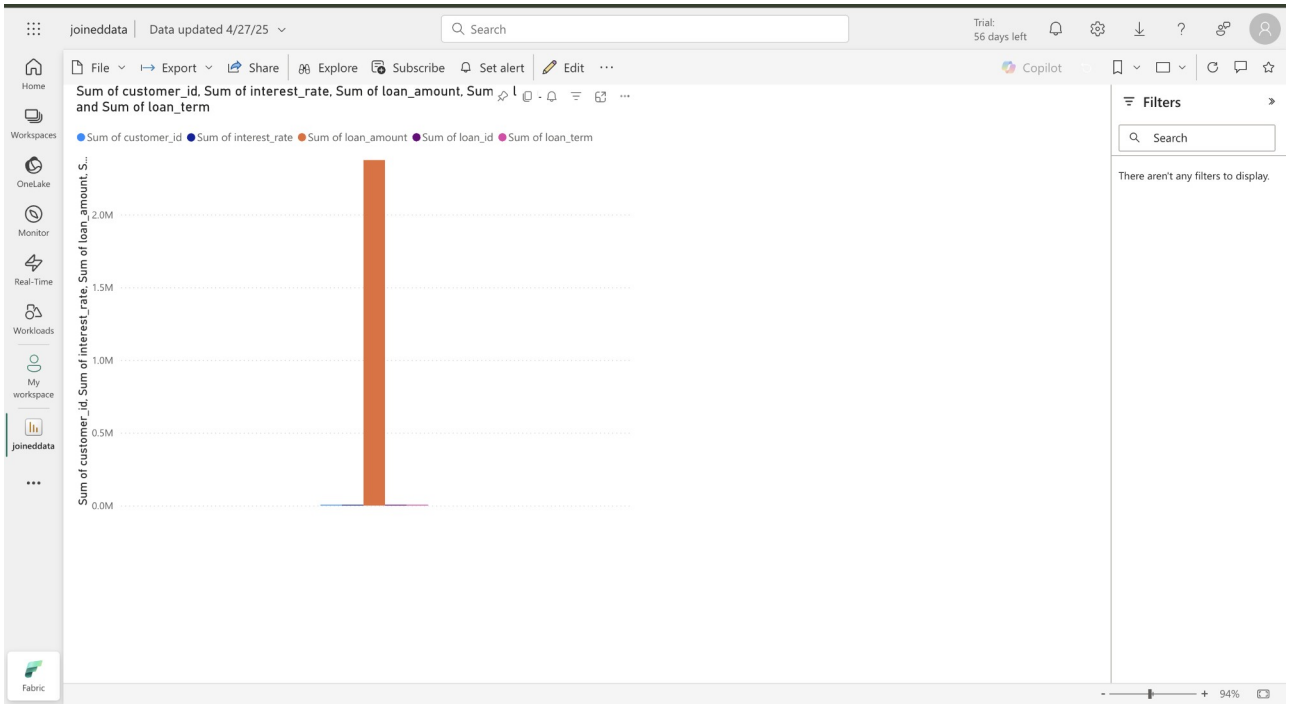
- Go to the power BI desktop and connect the storage account with power BI. Select the table we want to visualize and publish it to the fabric workspace.







- Successfully loaded the visualization in Fabric workspace.



CONCLUSION

Build an efficient and robust transformation of financial data of customers with the help of Databricks notebooks. Visualization of the files were done using Power BI and Fabric. Also done SCD Type 1 transformations on the cleaned data successfully and triggered the work to run on particular time using the workflows in Data bricks.