**TRANSACTIONS**

**AND**

**LOAN DATA**

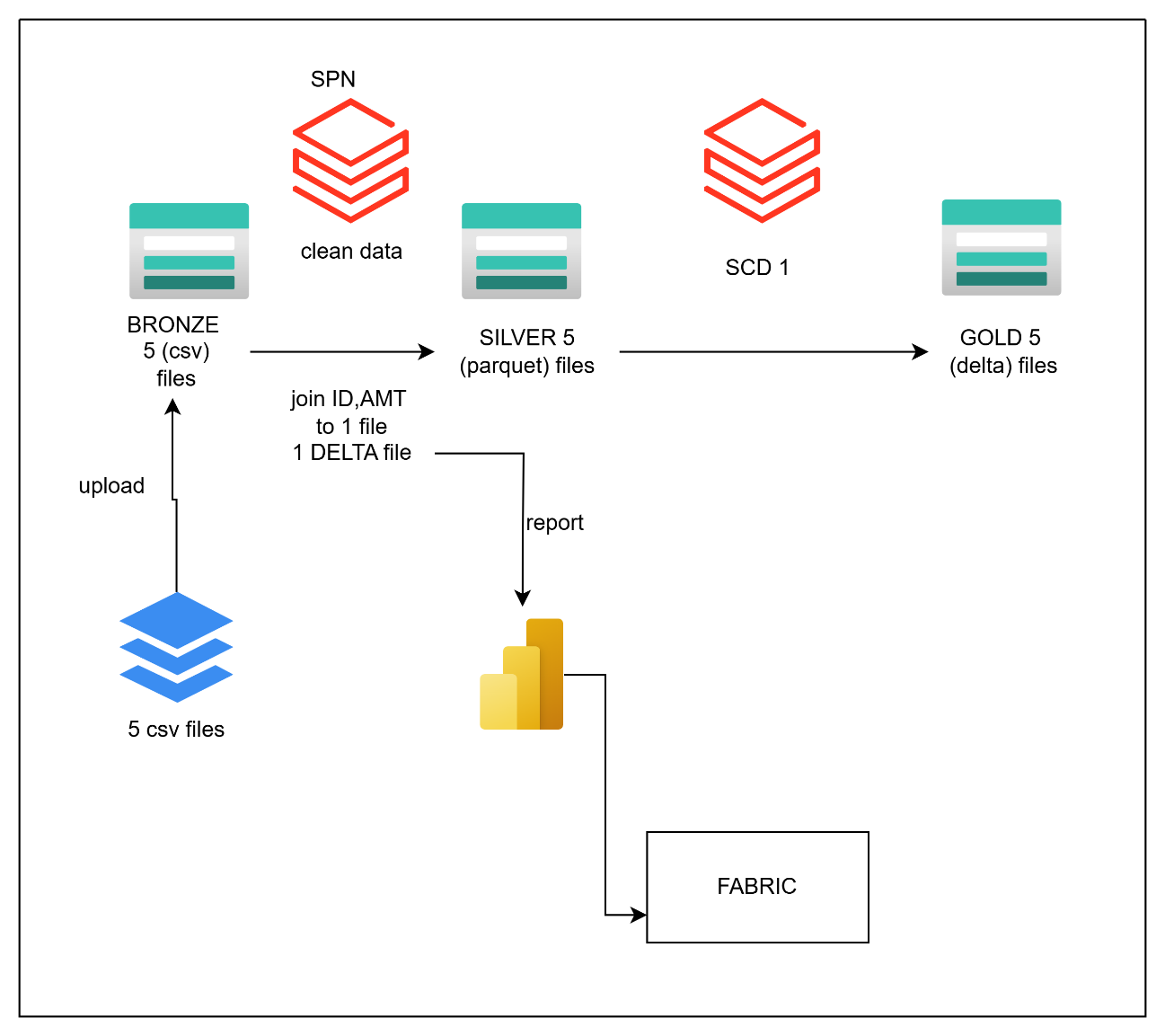
**FOR A**

**CUSTOMER**

**OBJECTIVE**

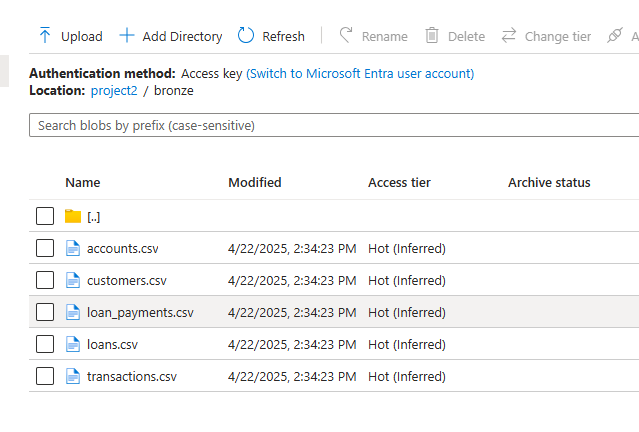
The project aims to design and implement a robust data pipeline for processing customer account data. This includes copying data from ADLS GEN2 (Bronze layer) and transforming the data in the silver layer using Data bricks Notebooks and GOLD Storage into the SCD Type 1 Delta Table in ADLS GEN2. The pipeline aims to ensure efficient, accurate, and scalable data processing to support downstream analytics and reporting needs.

**ARCHITECTURE**

****

**STEP-1: DATA INGESTION (UPLOAD TO BRONZE ADLS GEN 2)**

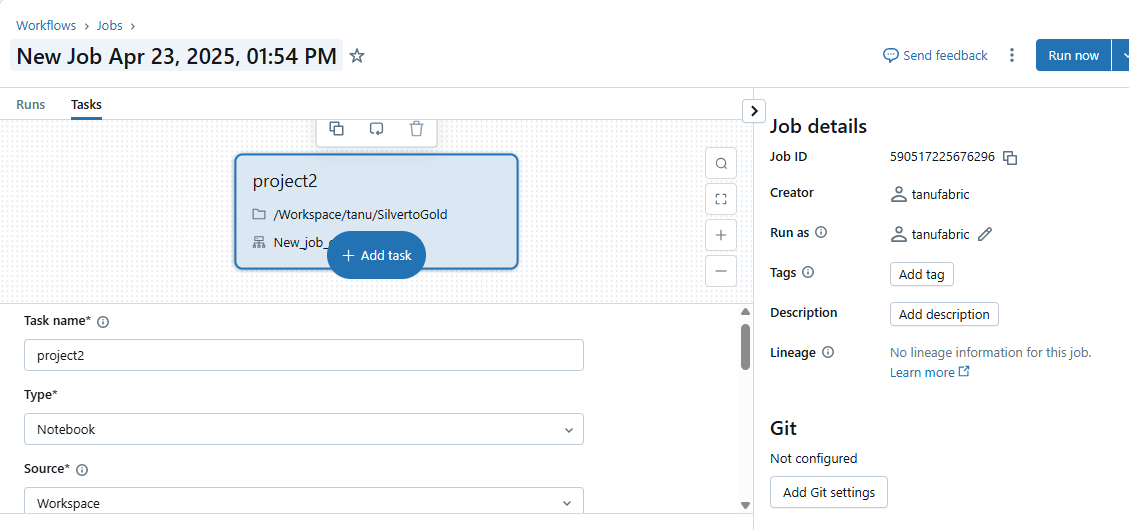
Here we need to load the files from our local system to directly bronze container in ADLS GEN 2. The files are accounts.csv, customers.csv, loan\_payments.csv, loans.csv, transactions.csv using upload option.

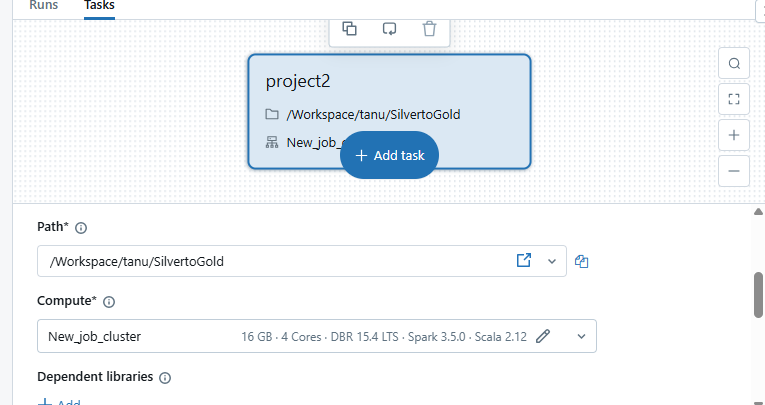


**STEP-2: Use Databricks Notebooks to remove the duplicates**

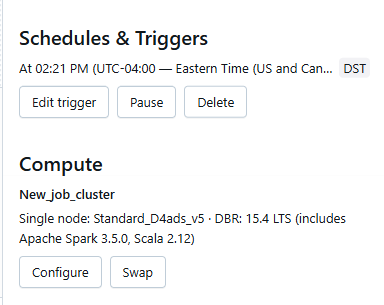
To implement this we have to log in to Databricks workspace and then create a folder and a notebook inside it.

Next create a job compute to schedule the notebook

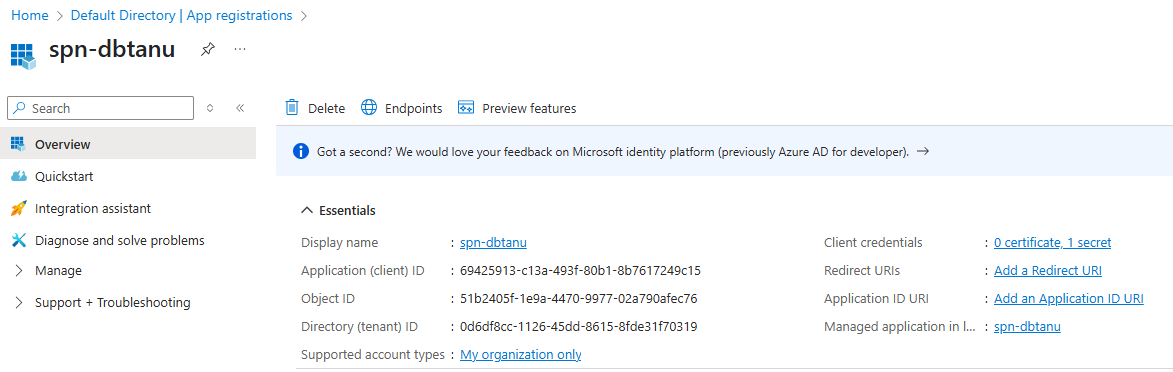




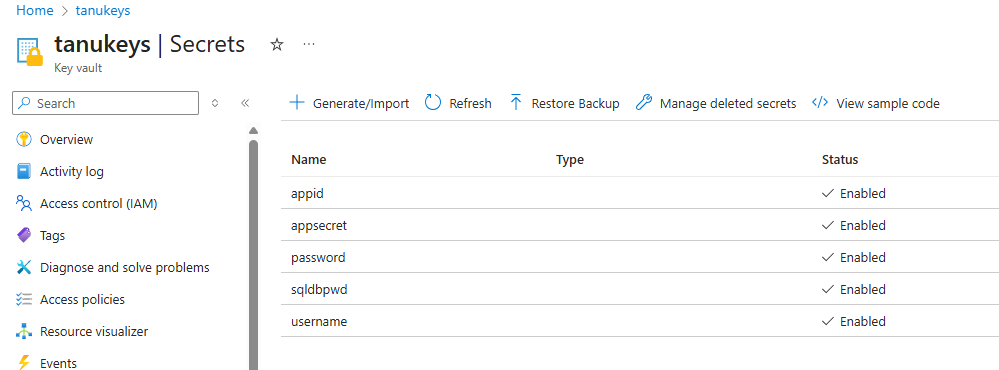
Schedule the notebooks



To mount ADLS GEN 2 on Databricks workspace let’s create a service principle and provide storage blob contributor role to it

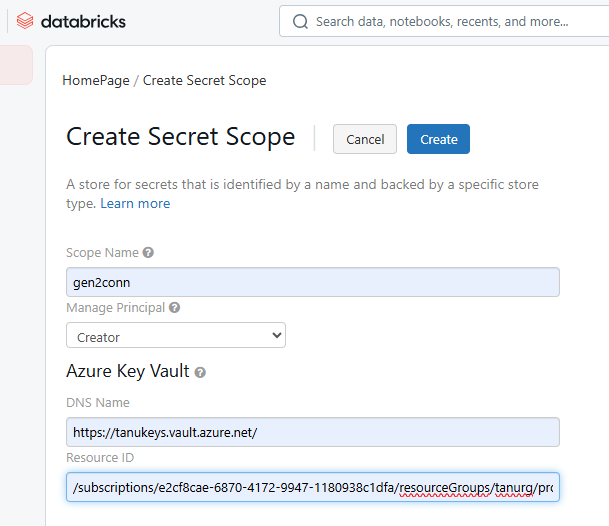


Here we store this Application ID , Application secret in Azure key vault as follows appid,appsecret.

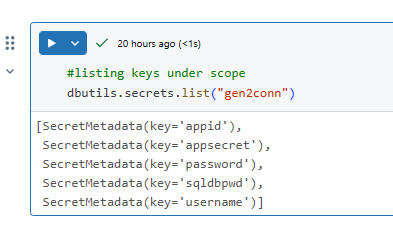


Now we go to Databricks and create a scope for azure key vault using vault URI , Resource ID

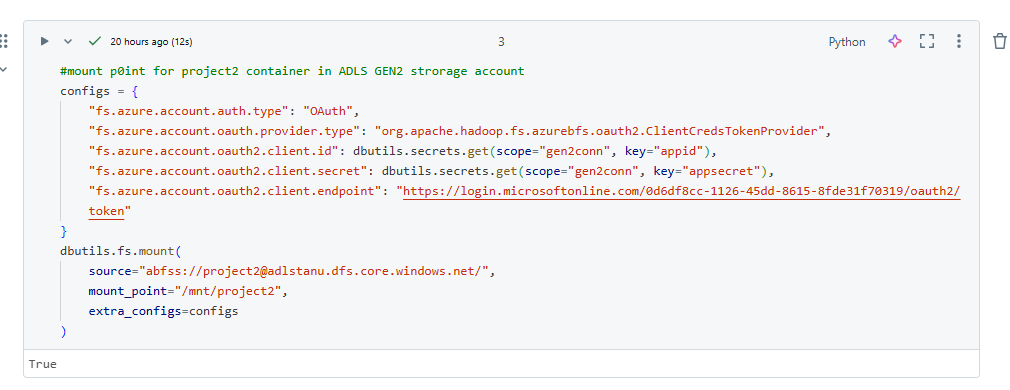
https://adb-552146958663388.8.azuredatabricks.net/?o=552146958663388#secrets/createScope



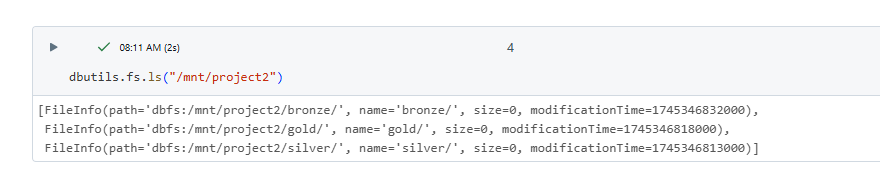
List all secrets under the scope using dbutils.secrets.list("gen2conn")



Mount point for project2 container in ADLS GEN2 using appid, appsecret, scope, tenantID in service principle

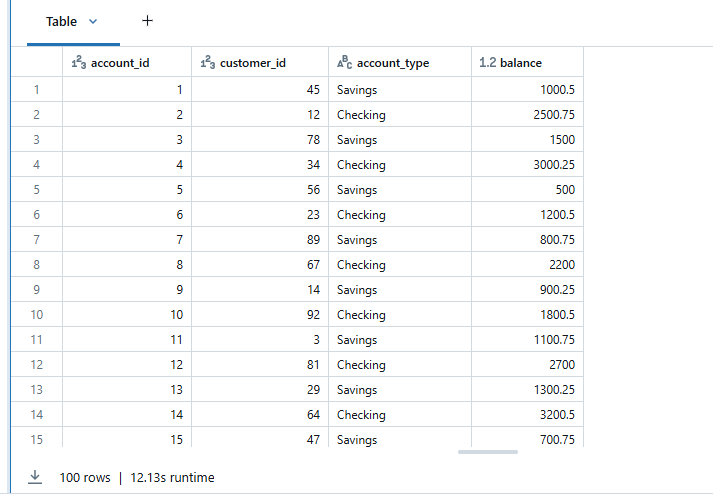


List folders inside project2 container

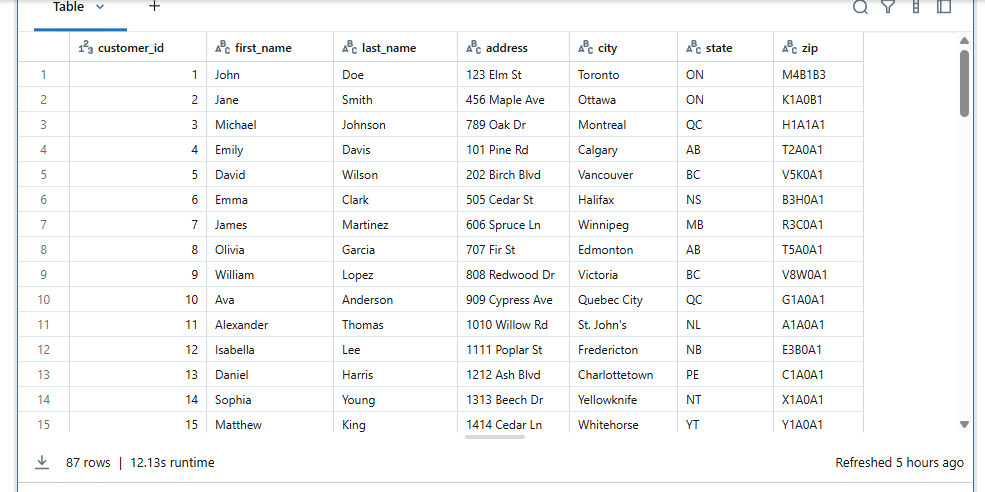


Reading all five files using data frames separately and displaying them

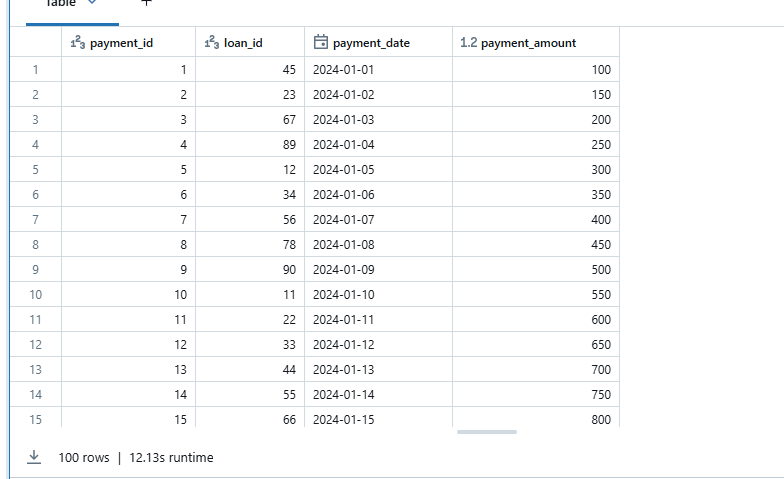




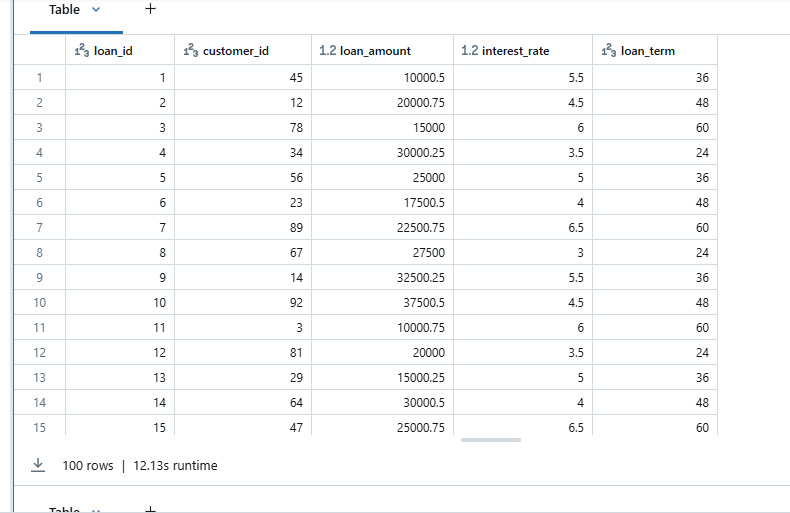
Customers.csv



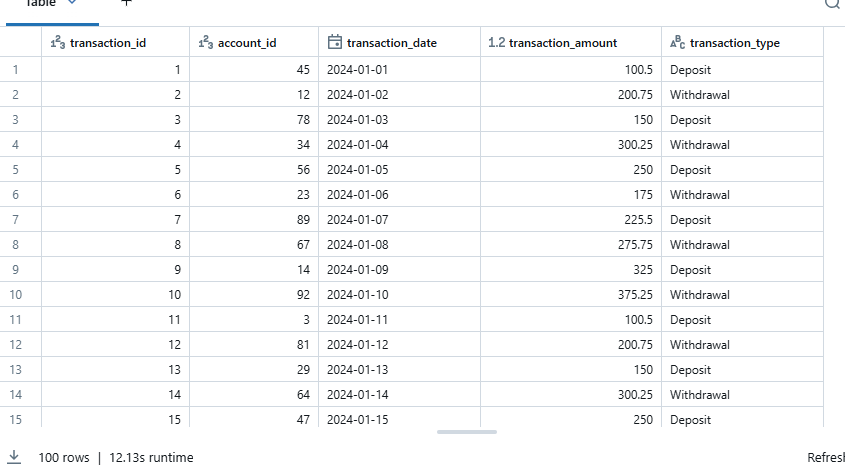
Loan\_payments.csv



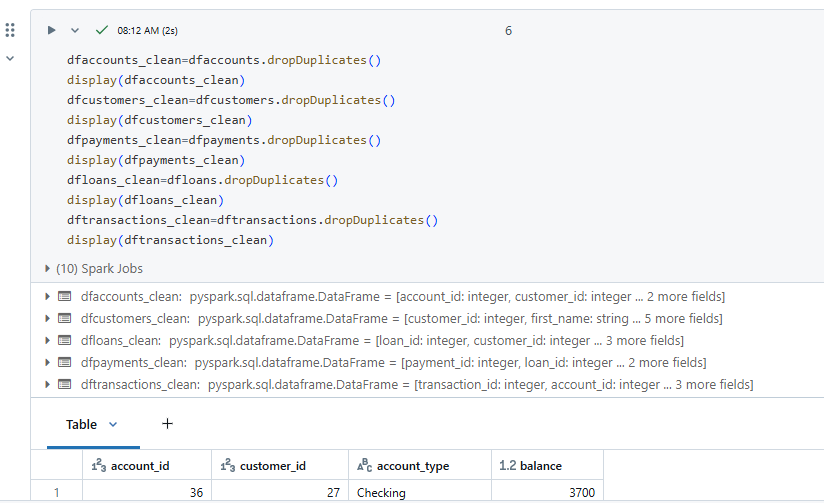
Loans.csv



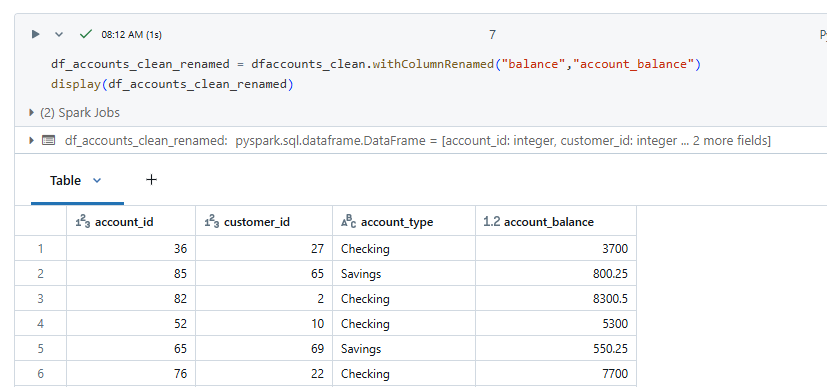
Transactions.csv



Dropping duplicates in all files using dropDuplicates()



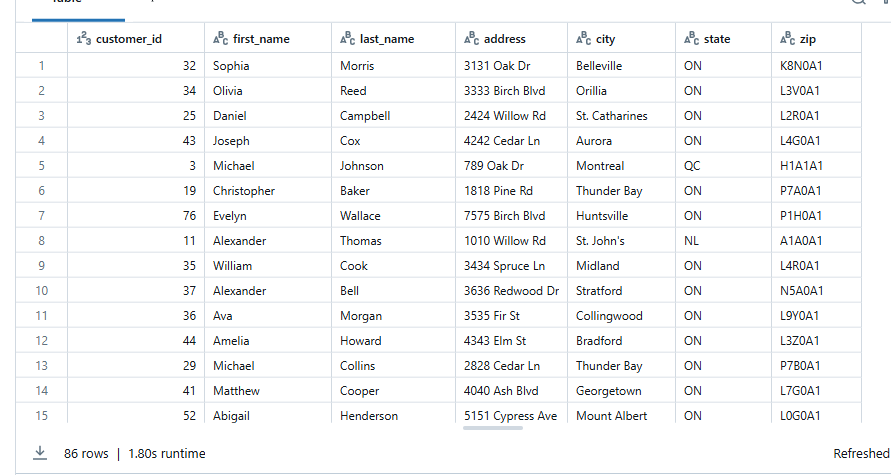
Renaming balance column in accounts.csv to account\_balance



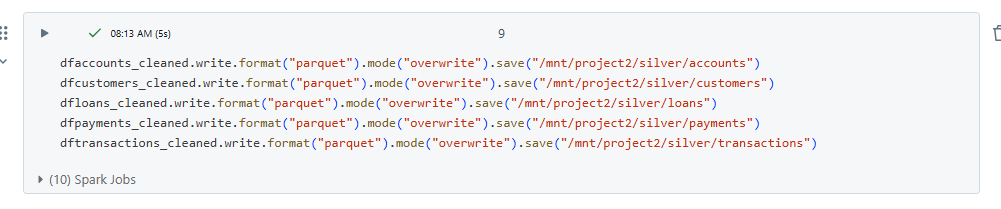
Dropping any null values in all files



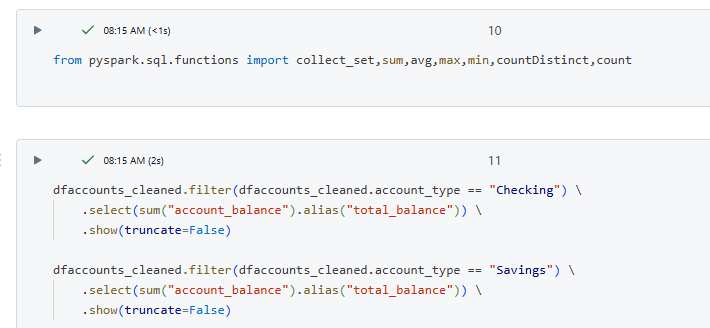
We have 87rows with one null record in customers.csv now it becomes 86 rows

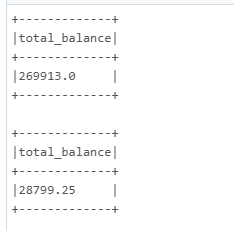


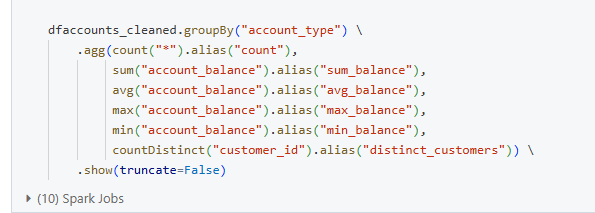
Writing back all the five csv files to ADLS GEN 2 Silver folder in parquet format

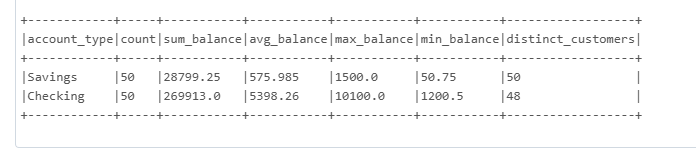


Now applying aggregate functions on accounts file





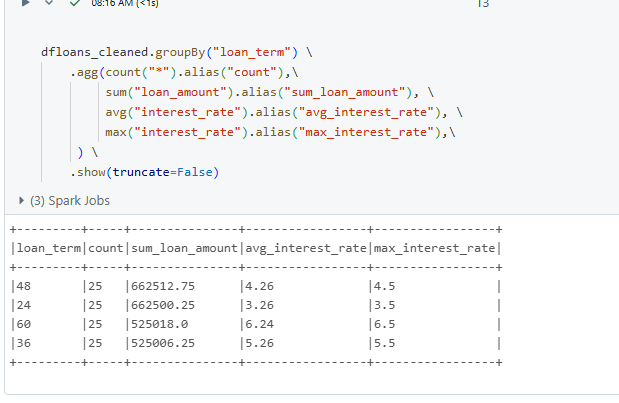




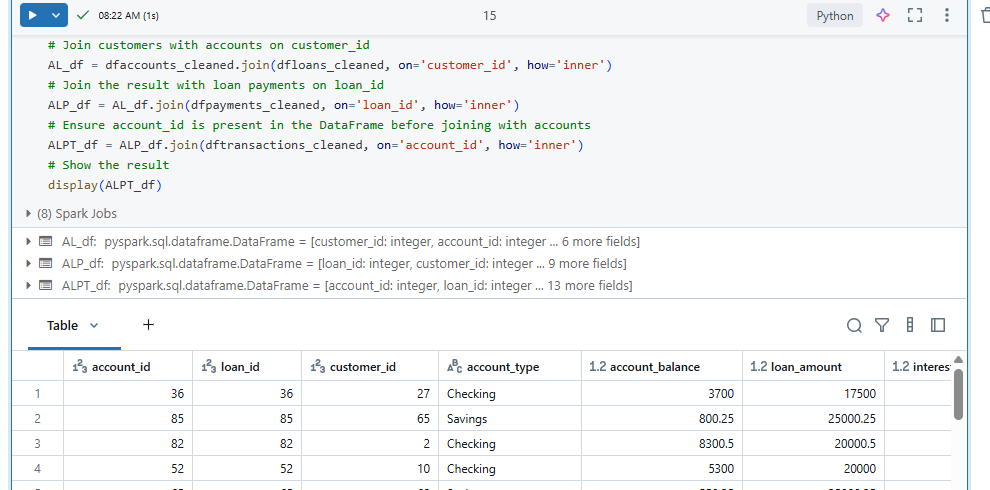
Aggregate functions(sum,avg,max,min,countdistinct) on loan\_payments file as follows



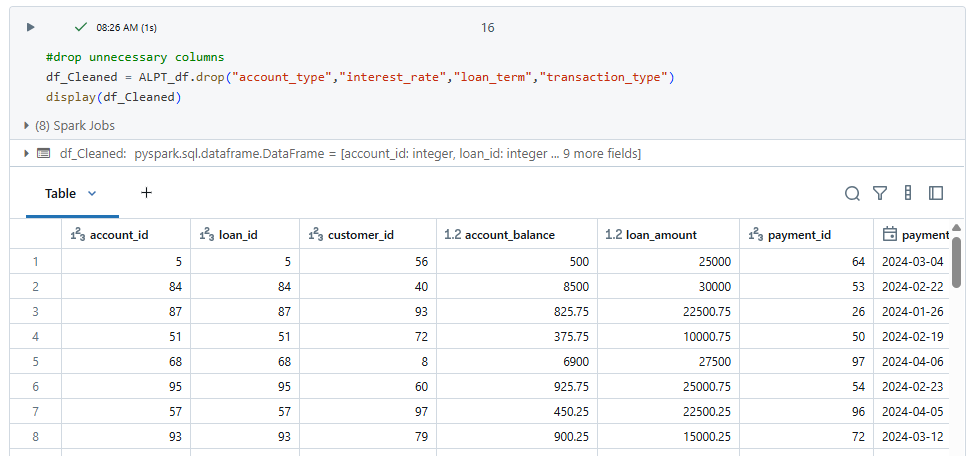
Aggregate funtions (sum,avg,max,count) on loans file



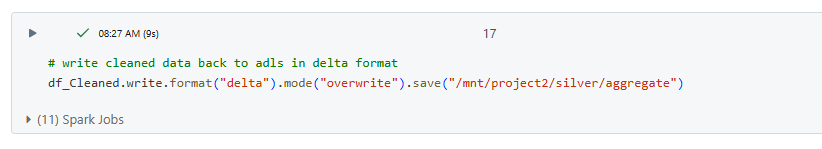
Merging all files using inner join into a single data frame

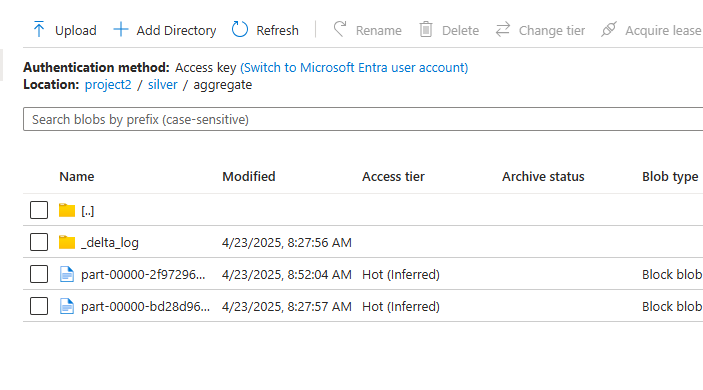


Dropping unnecessary columns from the joined data frame



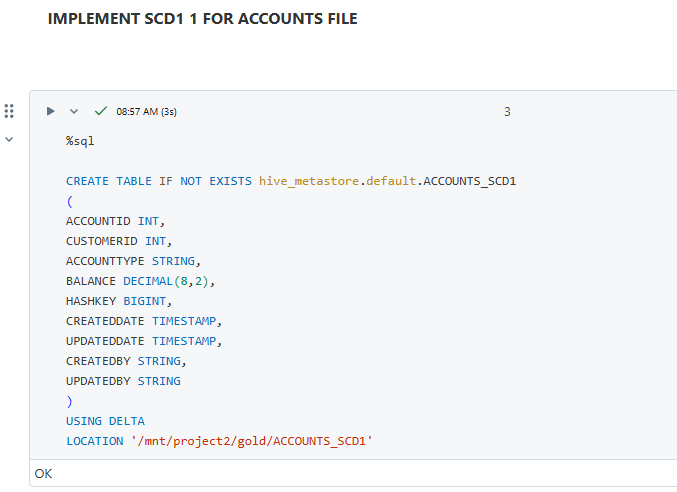
Writing this joined data frame back to silver folder in delta format



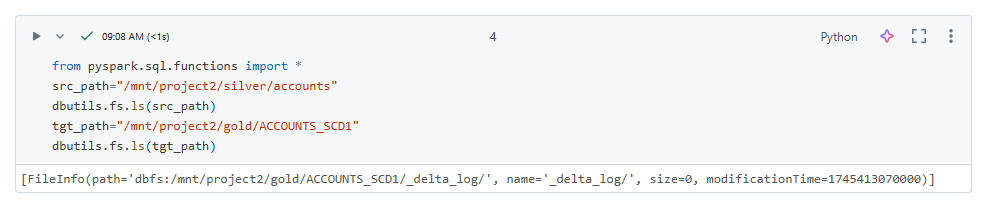


**STEP-3: Use Notebooks using SCD Type technique (SCD 1)**

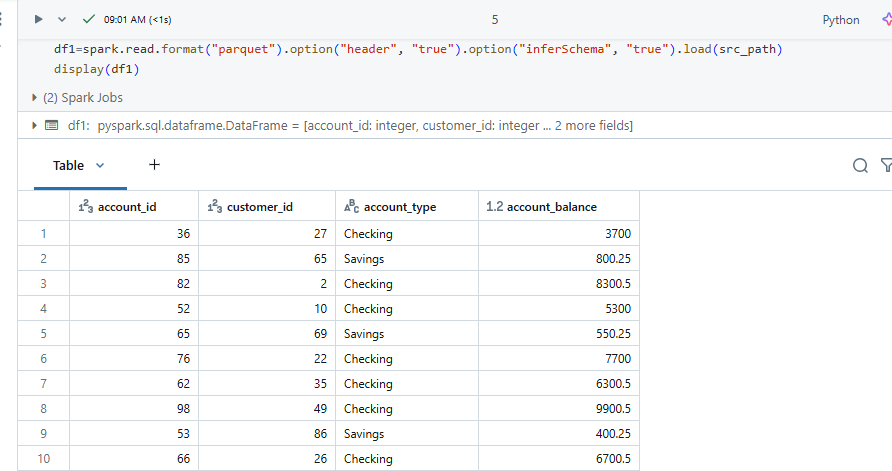
To implement this logic first we have to create a delta table in gold layer as follows



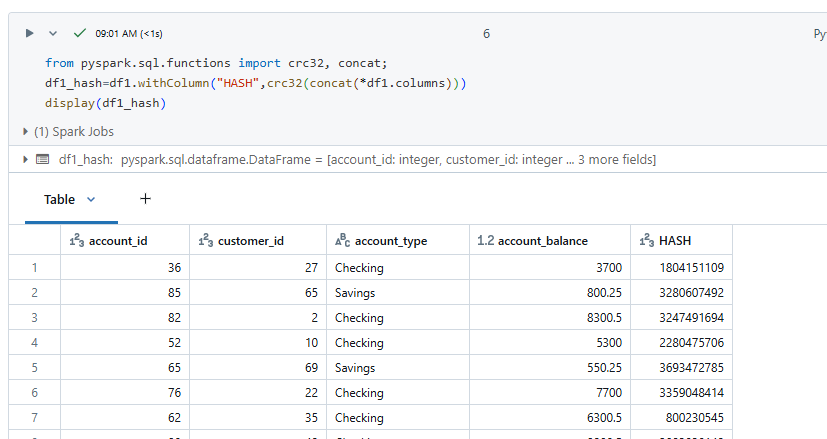
Consider source path and target path as follows



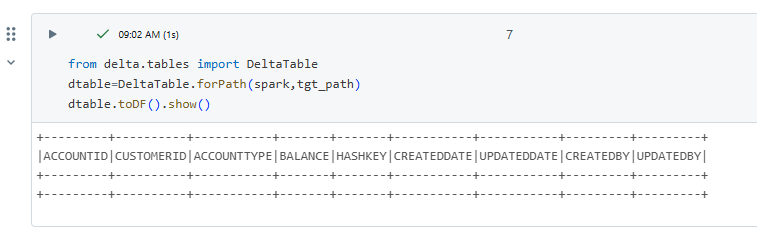
Load the source path into a data frame df1



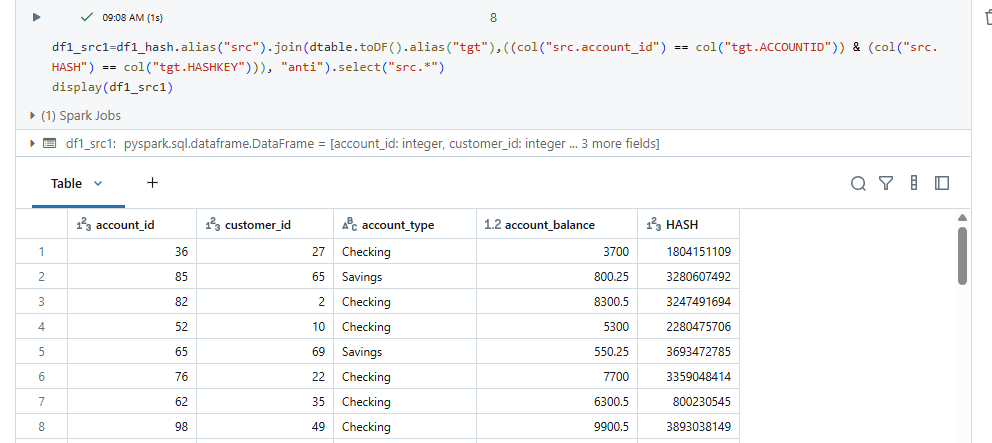
Generate a hash key for the file using CRC32



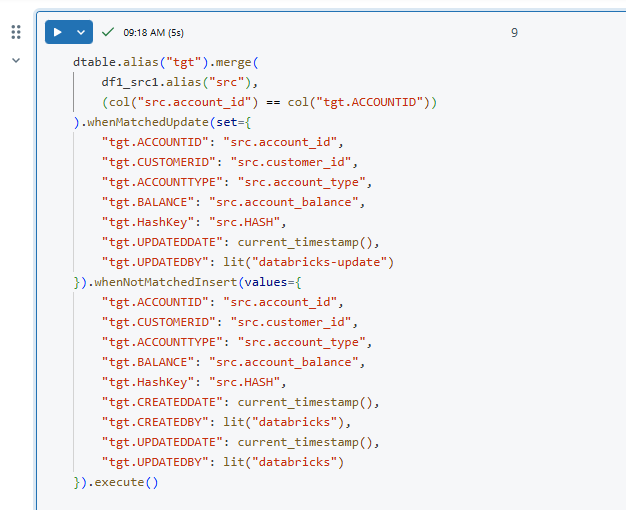
Consider the delta table in target path and load it



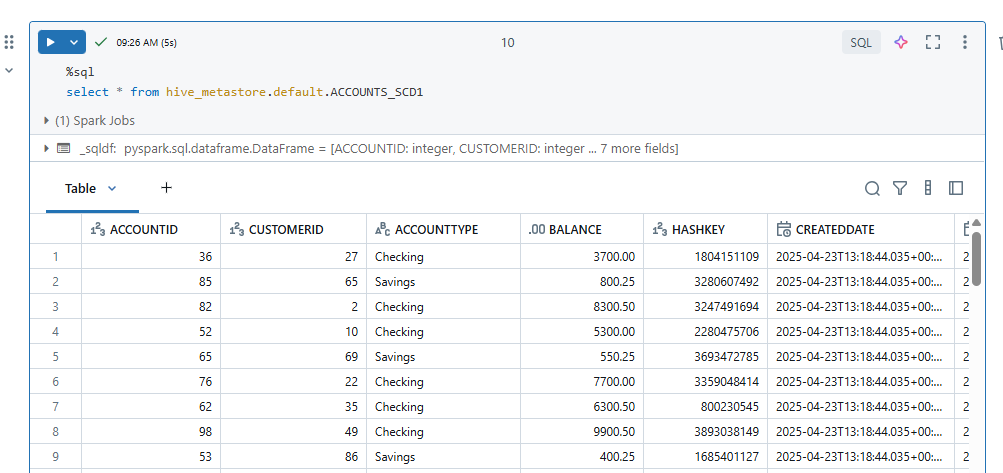
Join source and target using anti

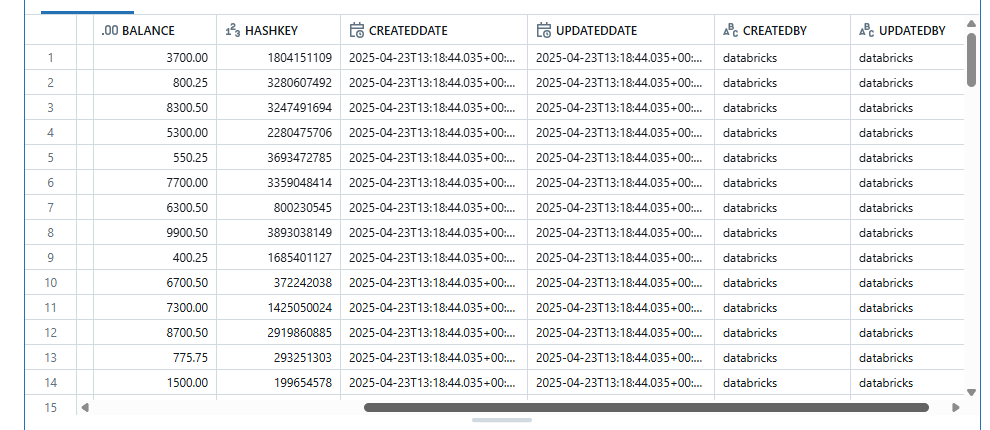


Merge columns for insertion and updating data using source and target

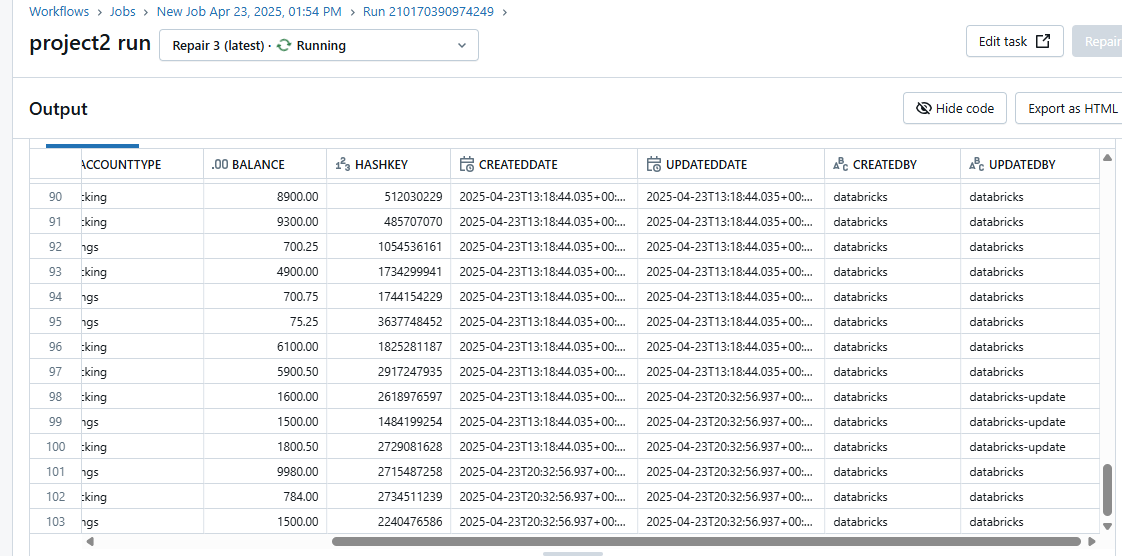


Now see the data in the delta table using %sql command



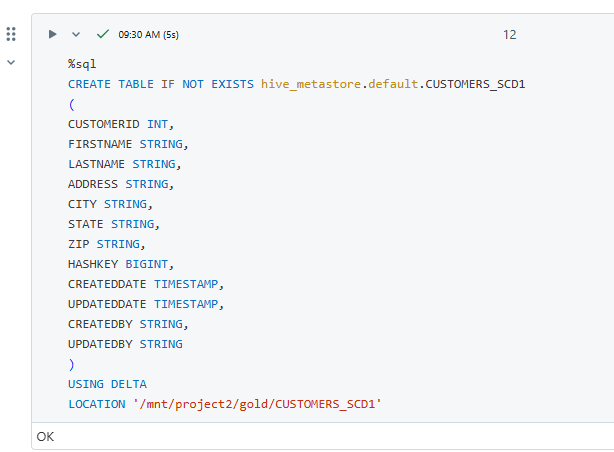


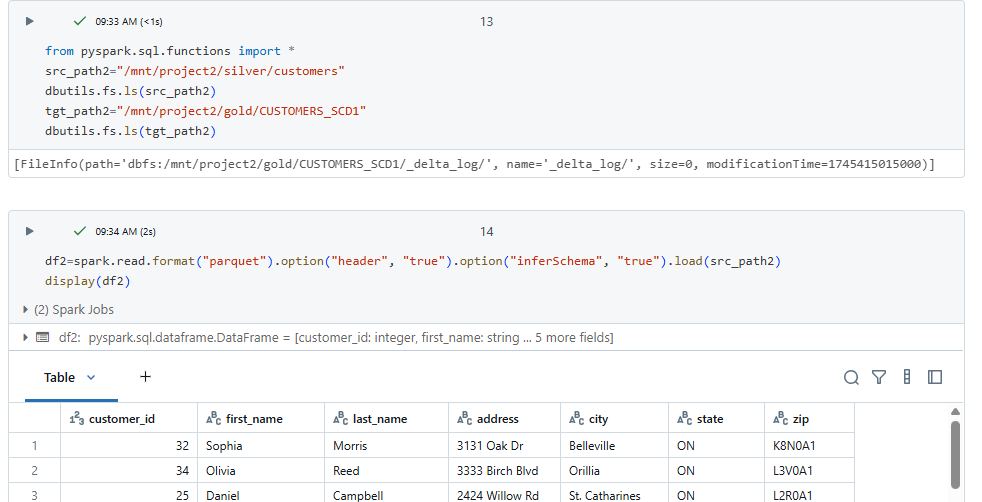
Now let us take a sample updated data file in source see the updates

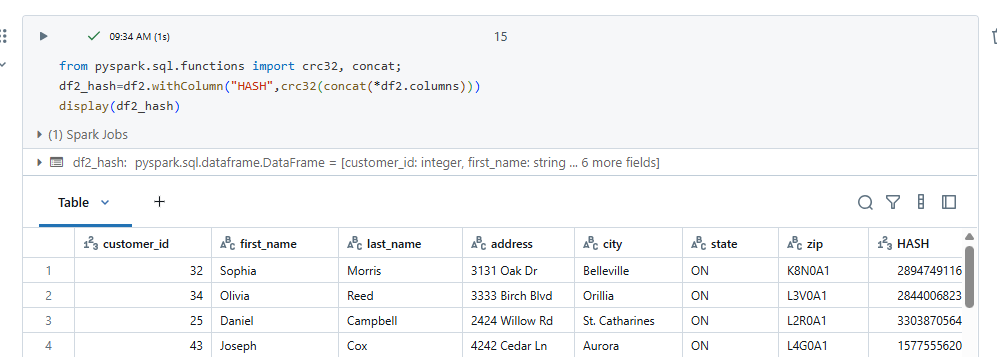


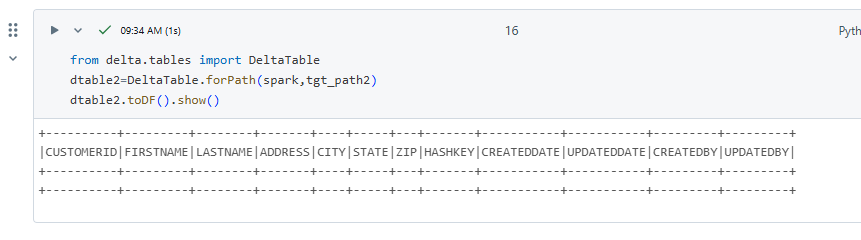
**IMPLEMENT SCD 1 FOR CUSTOMERS FILE**

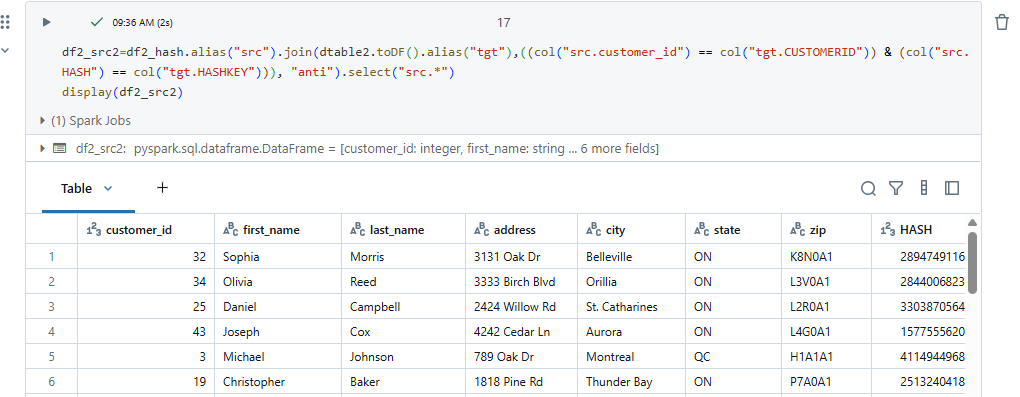
To implement this logic first we have to create a delta table in gold layer as follows



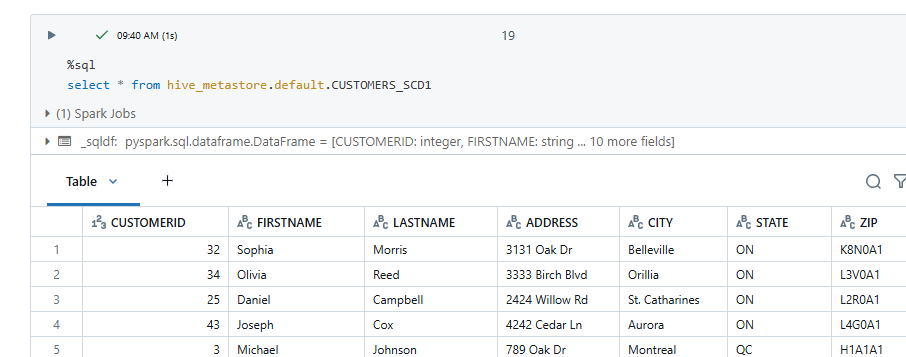




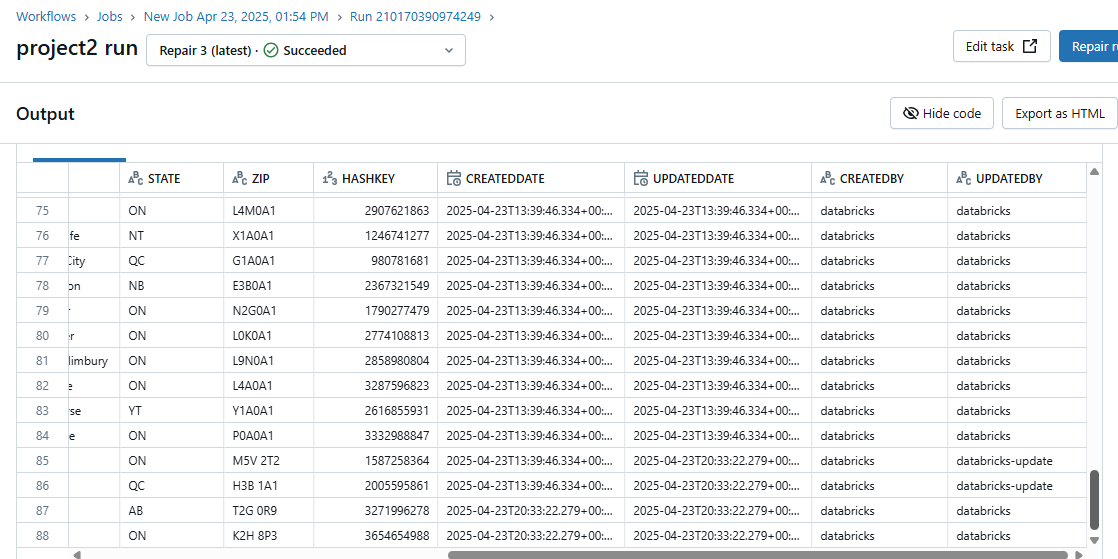




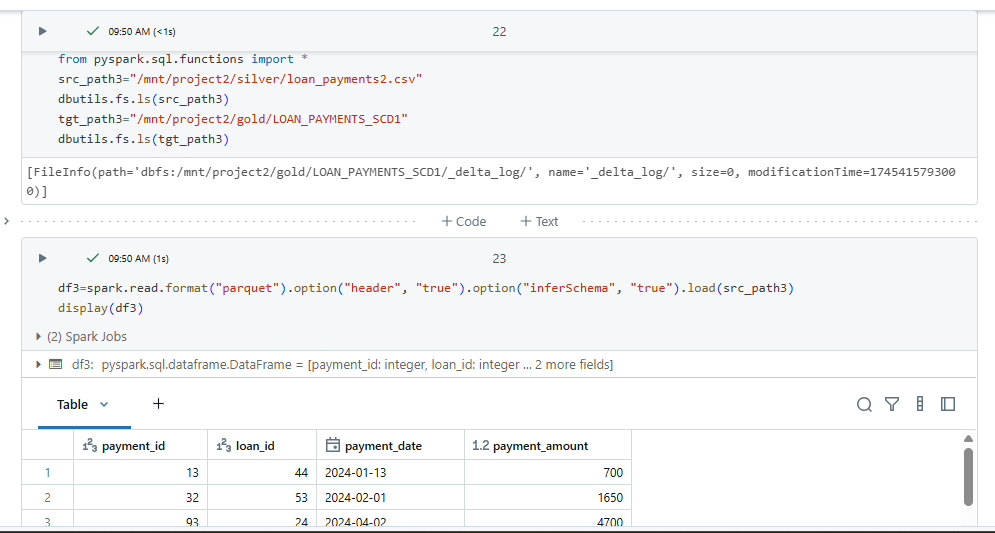


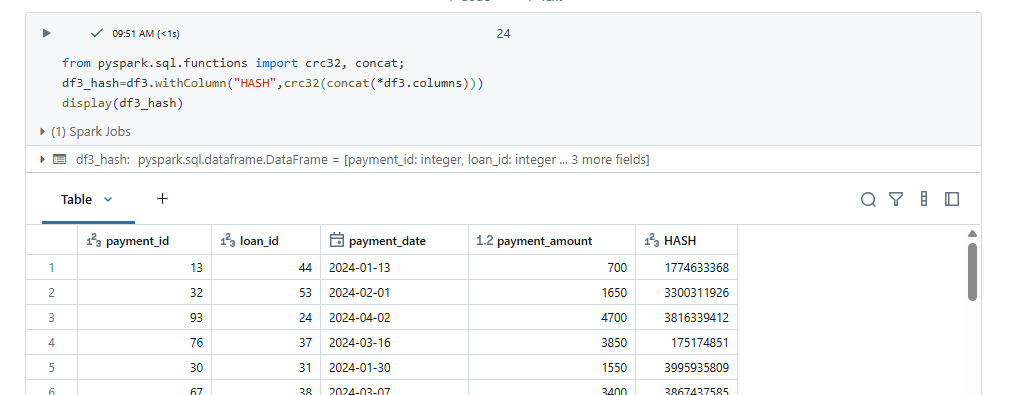


To see the updated data we pass a sample file and see the update results



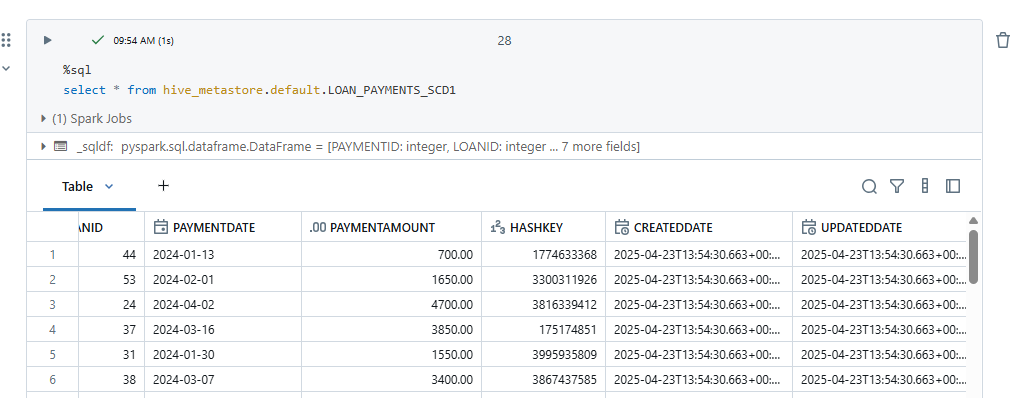
**IMPLEMENT SCD 1 FOR LOAN\_PAYMENTS FILE**  

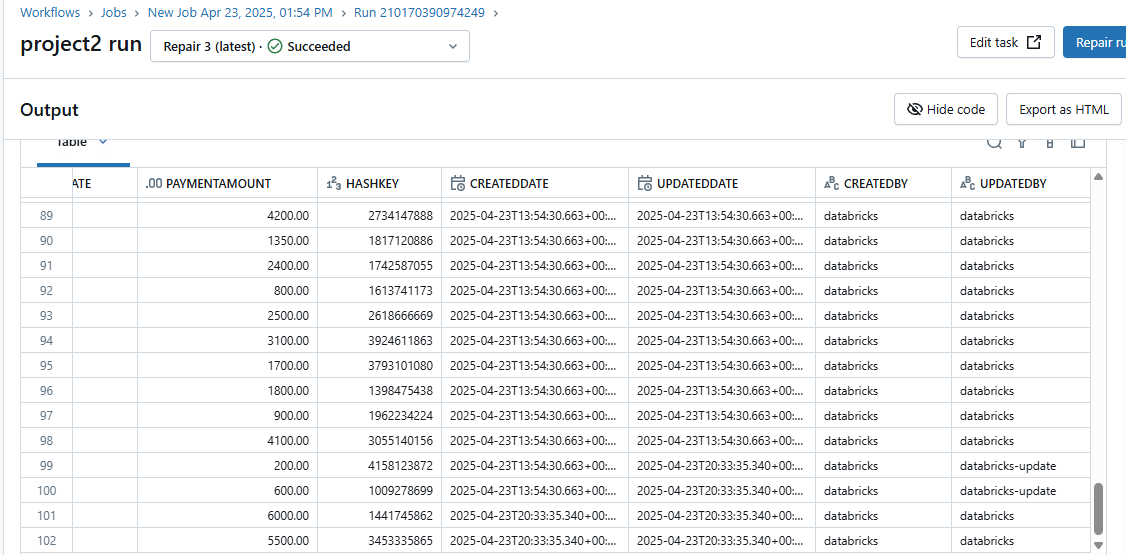






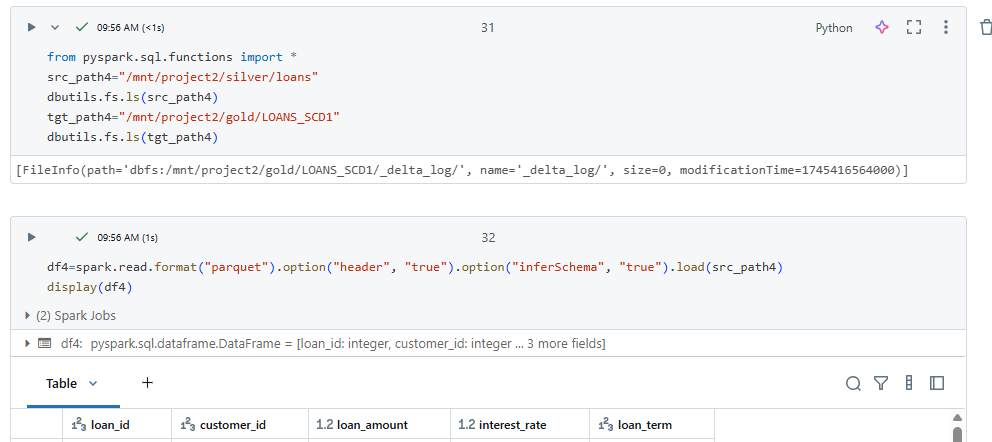


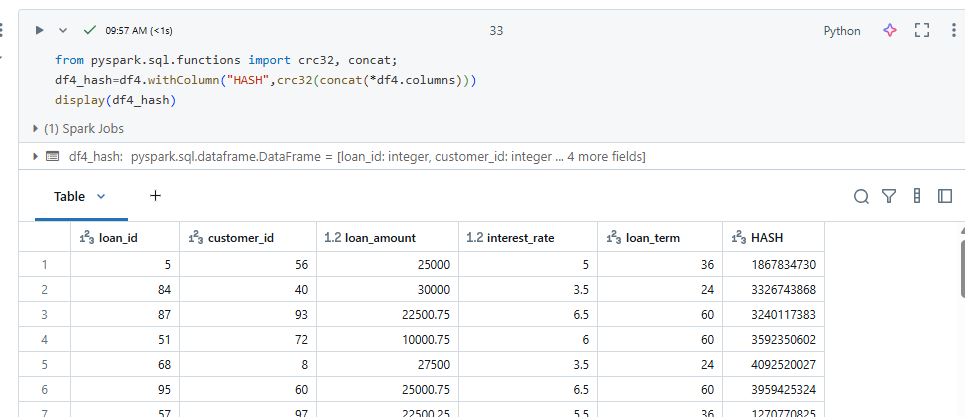
We can see updates as follows

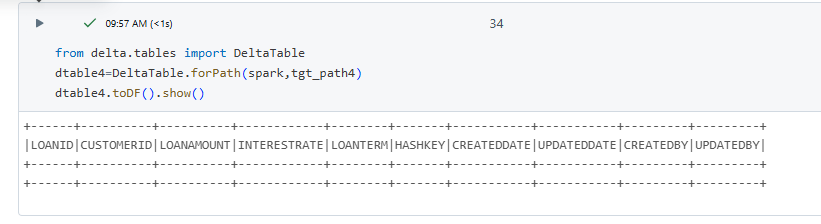


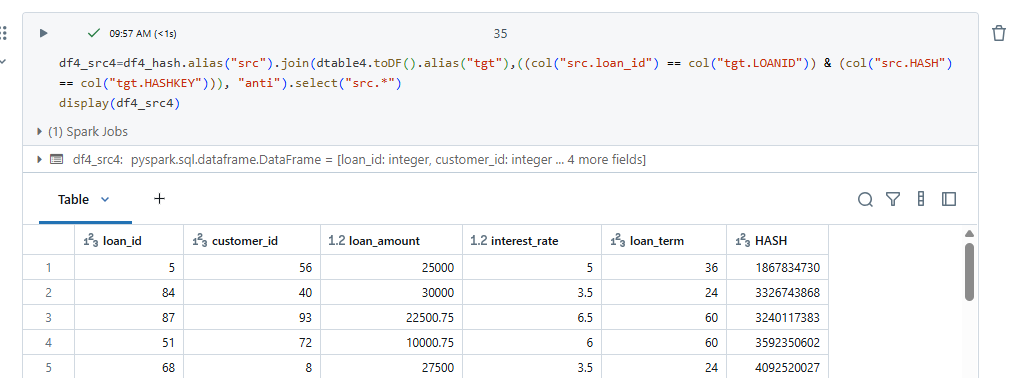
**IMPLEMENT SCD 1 FOR LOANS FILE**



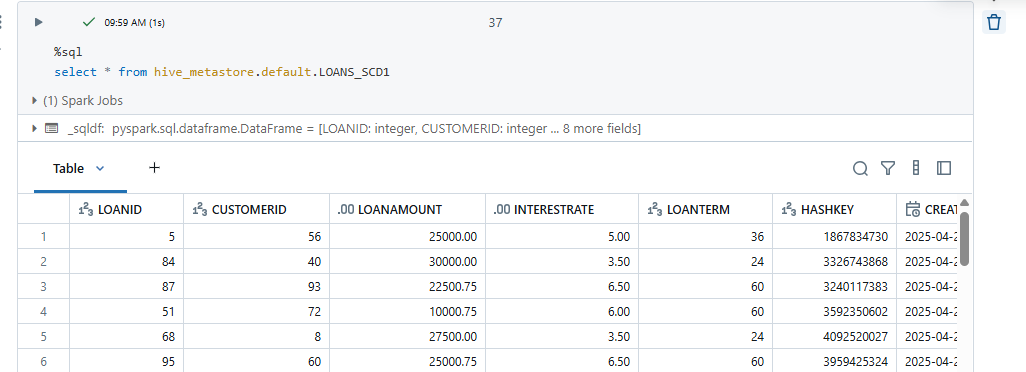




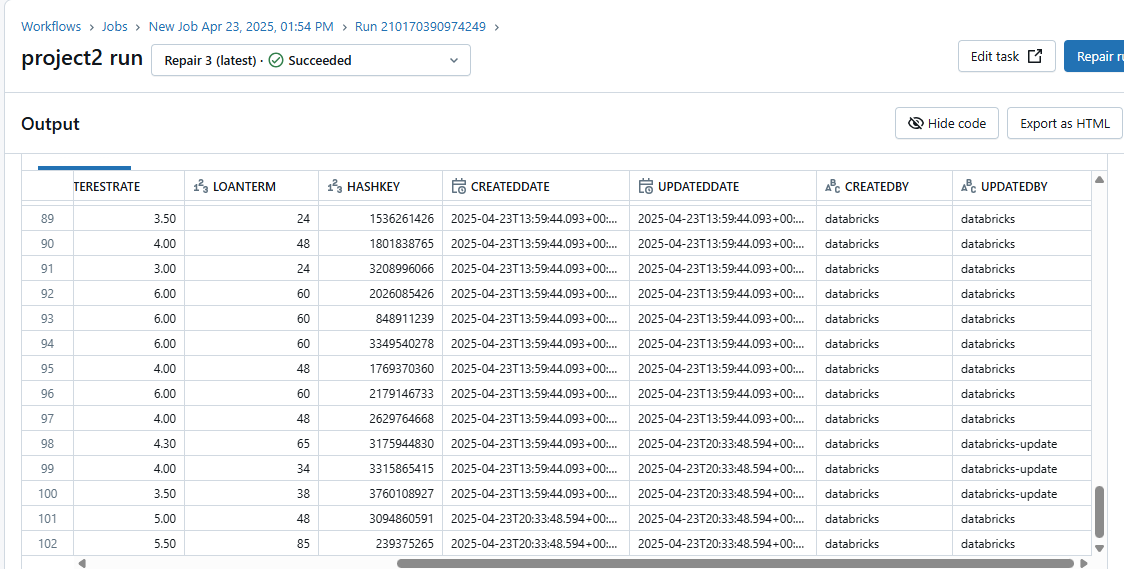




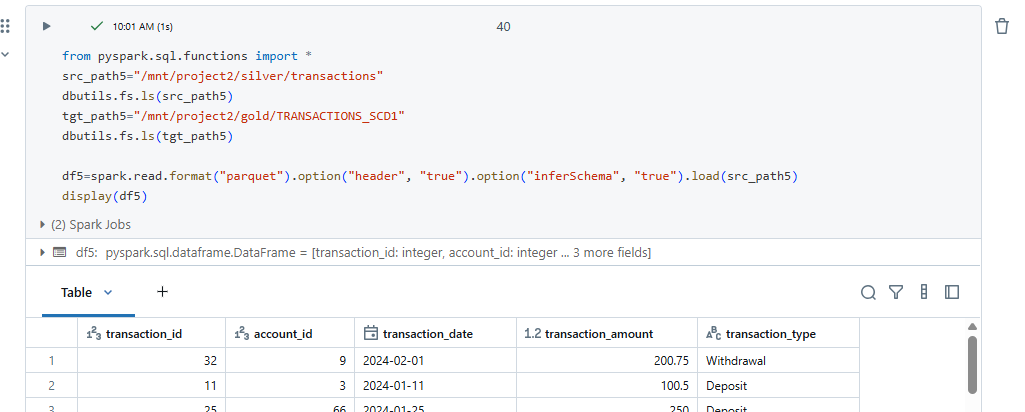


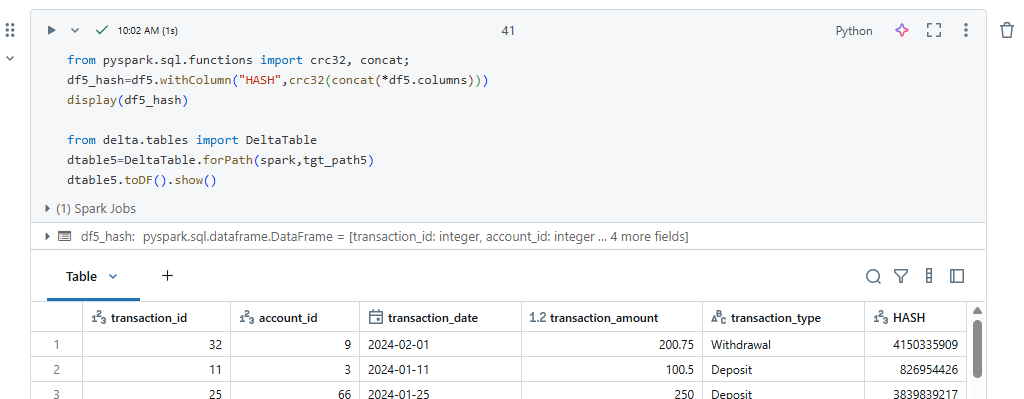


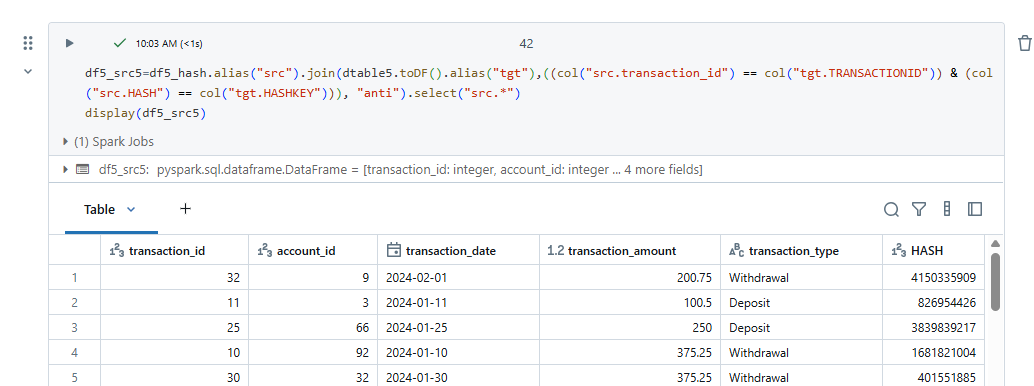
Updated data as follows

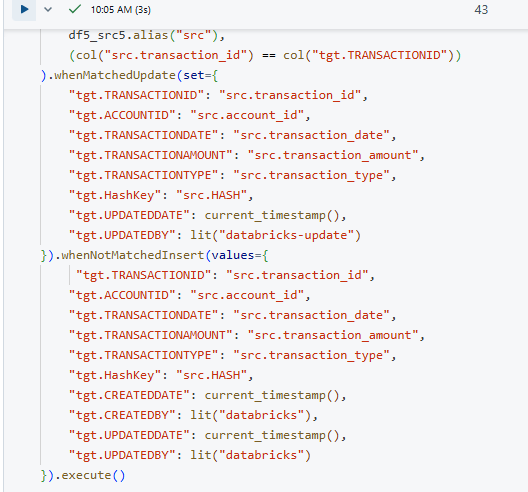


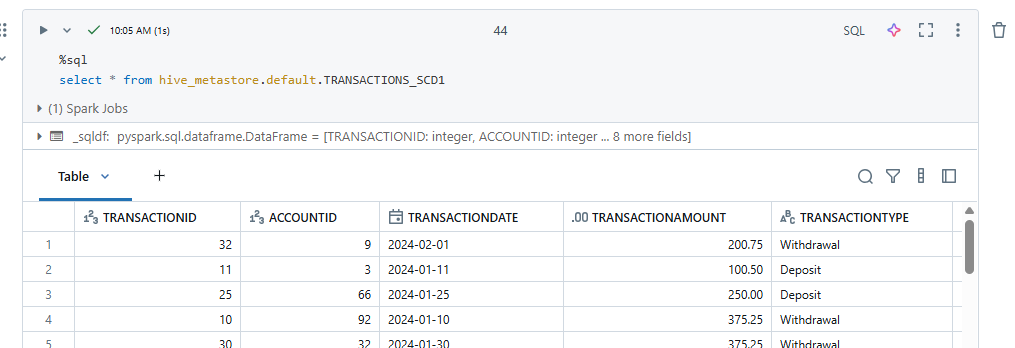
**IMPLEMENT SCD 1 FOR TRANSACTIONS FILE**  

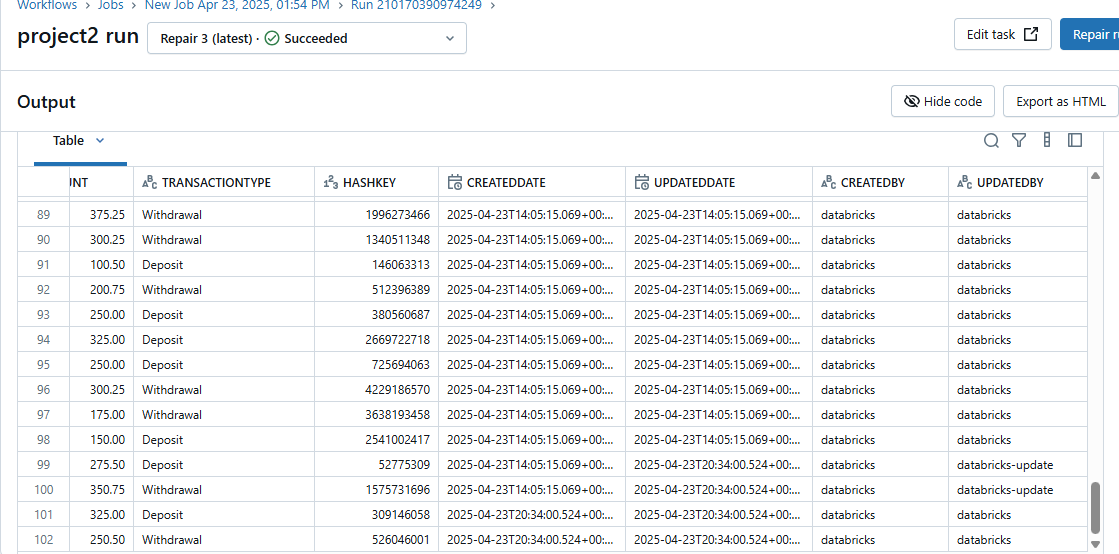




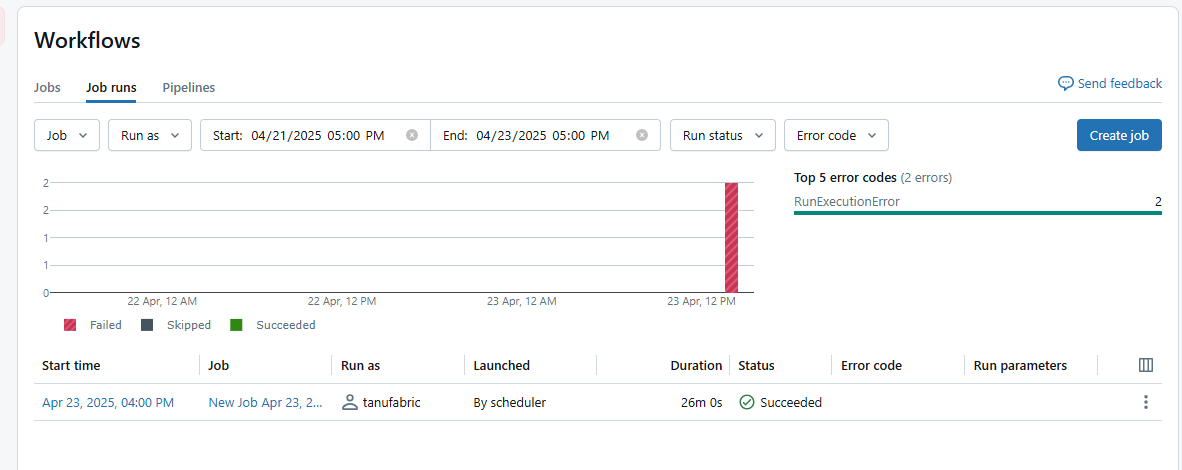




We can see the updated data as follows



We can schedule workflows and run the notebooks



**STEP- 4: USE POWER BI for DATA VISUALIZATION**

Go to power bi desktop

🡪home

🡪get data

🡪azure

🡪Azure data lake storage gen2

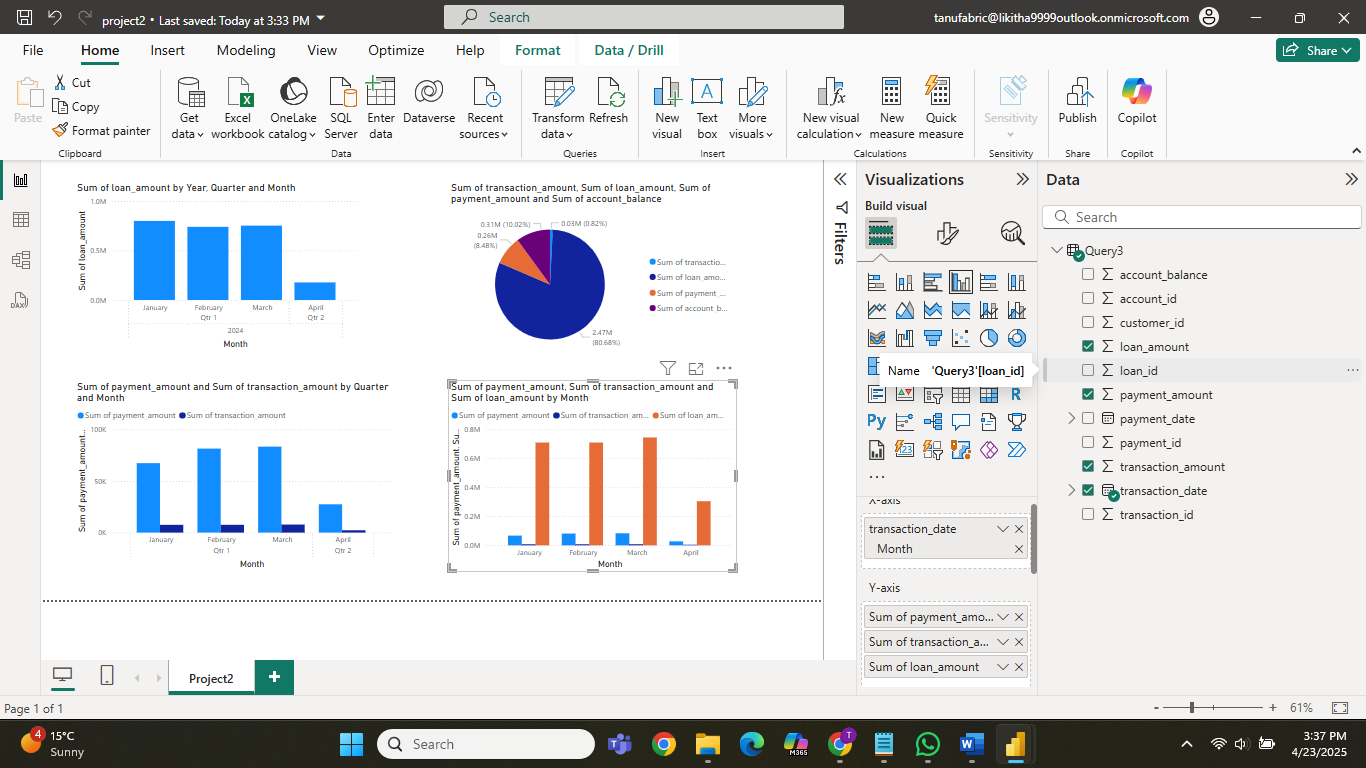
🡪url for connection to folder

🡪transaform data

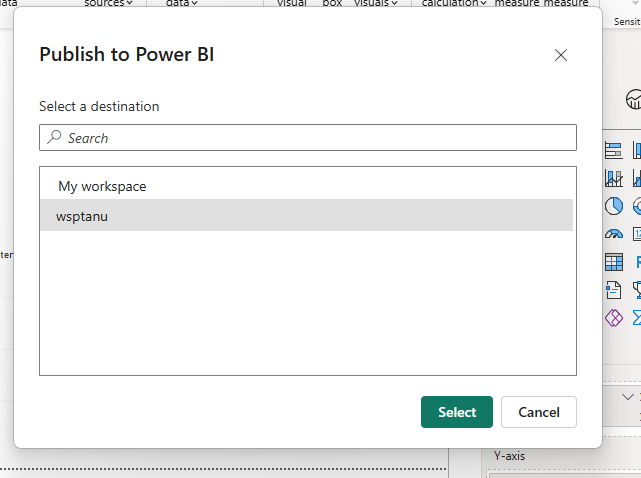
🡪combine files

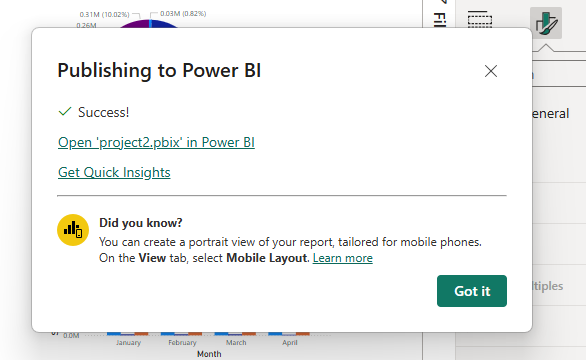
🡪power query

🡪access data for reports



Now save and publish to fabric workspace





Open fabric workspace

