**GLOBAL BANKING ANALYSIS**

**Agenda**

Global Bank is undergoing a digital transformation to enhance its fraud detection capabilities, improve customer segmentation, and optimize its reporting systems. As a data engineer, your task is to design and implement a robust data pipeline that can handle real-time transaction data from multiple sources, process it efficiently, and prepare it for advanced analytics and reporting.

Your specific objectives are:

Design and implement a scalable data ingestion system that can handle 100 to 500 transactions per second from 10 to 50 global branches and digital channels.

Implement a Delta Lake architecture with Bronze, Silver, and Gold layers:

Bronze: Raw ingestion of streaming transaction data

Silver: Cleansed and enriched data, joining transactions with customer and branch information

Gold: Aggregated data for reporting, fraud detection results, and customer segments

Create a data processing pipeline that cleanses, transforms, and enriches the transaction data with customer and branch information.

Implement a real-time fraud detection system based on rule-based flagging and transaction pattern analysis.

Develop a customer segmentation system based on transaction behavior and account information.

Create an automated reporting system that generates daily, weekly, and monthly summaries of transaction volumes, potential fraud cases, and customer segment distributions.

Ensure data quality, implement schema evolution capabilities, and manage data retention policies across all layers.

Optimize Delta Lake tables for both streaming writes and batch reads to enhance overall system performance.

Implement conceptual multi-tenant isolation within the same Databricks workspace to support different teams or departments.

Optimize the data storage and retrieval process to support quick analytics queries and real-time dashboards.

Create a real-time dashboard showing transaction volumes across different channels

Visualize the geographical distribution of transactions and potential fraud cases

Build interactive charts for customer segment analysis

Design a time series visualization for transaction patterns and anomalies

**Flow Diagram:**



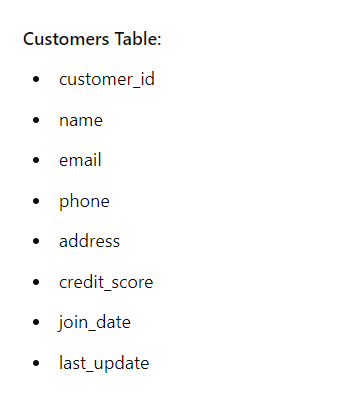
**Limitation:**

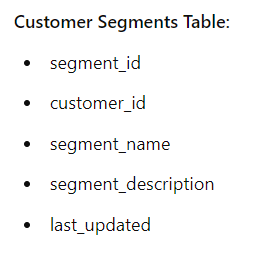
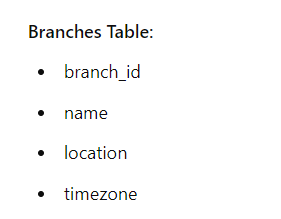
* In this project, I have used the community version of Databricks due to its neupro labs cloud restrictions
* Since I’m using community version so unable to create pipeline or job cluster

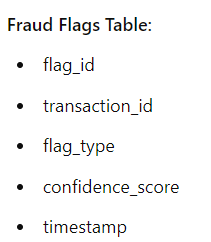
**Approach**:

* I have done entire process using Medallion architecture and spark streaming except customer, branch table

**Tables, Schema**





**Medallion architecture**

**Data Pipeline Description**

1. **Ingest Data to Bronze Layer:**

* Raw data from the Faker API is ingested into the Bronze layer through spark streaming.

1. **Cleansing and Transformation in Silver Layer:**

* Data is cleansed and transformed, moving from the Bronze layer to the Silver layer through both batch & spark streaming.

Cleansing Rules:

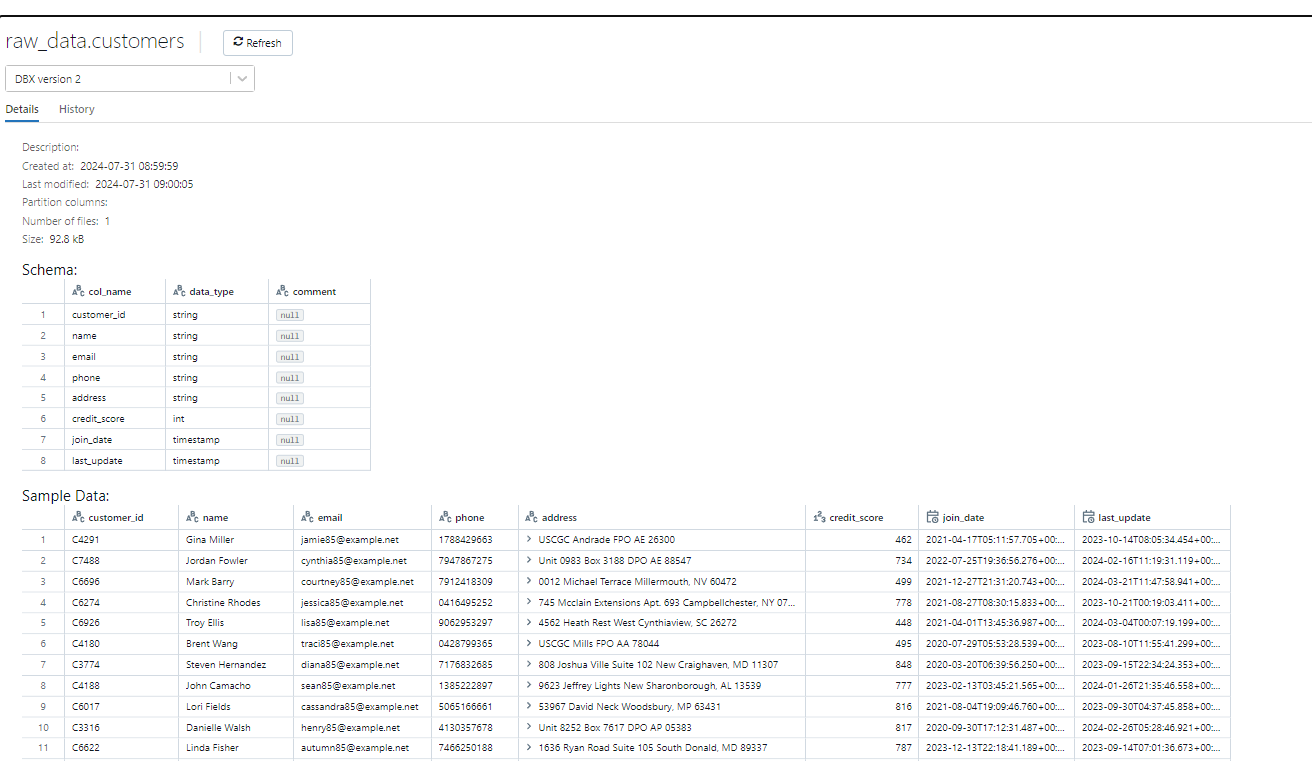
* Drops rows with any null values.
* Ensures columns are of the correct data type based on the schema.
* Removes duplicate rows.
* Trims whitespace and removes special characters from text columns.
* Handles outliers using the IQR method.
* Ensures data consistency and checks for non-null and non-NaN values**.**

1. **Aggregation and Analysis in Gold Layer:**

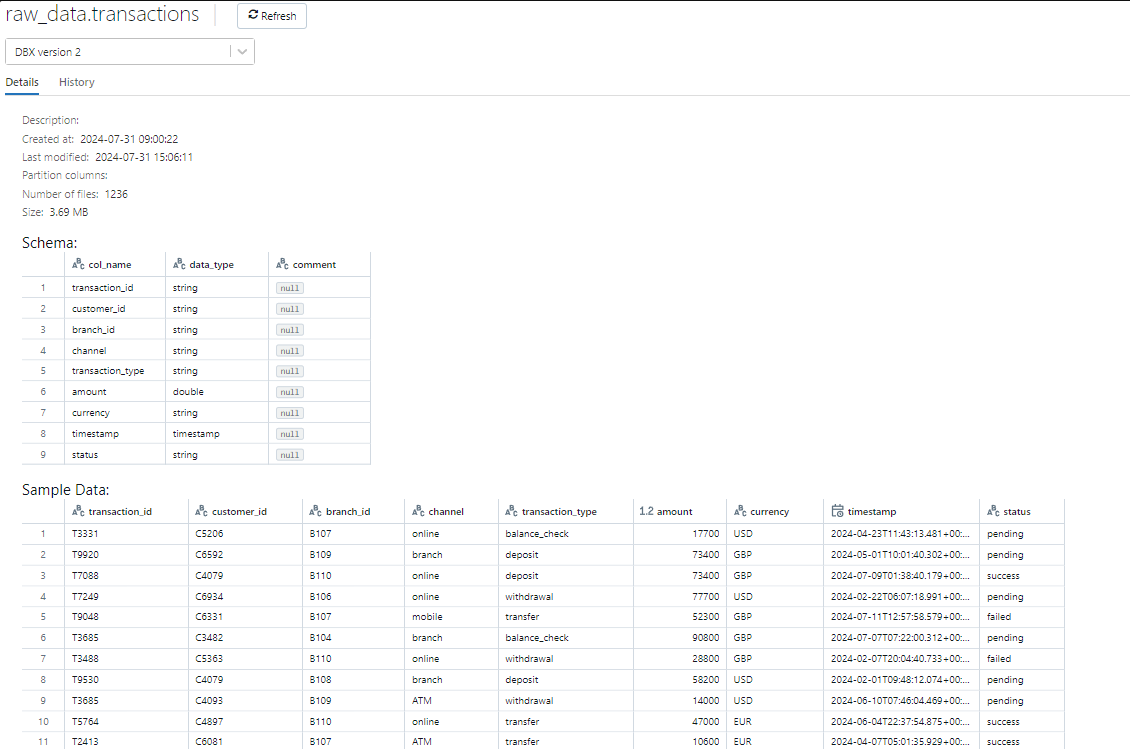
* Data is aggregated and prepared for business-level use in the Gold layer through spark streaming.

**Raw Data Generation**

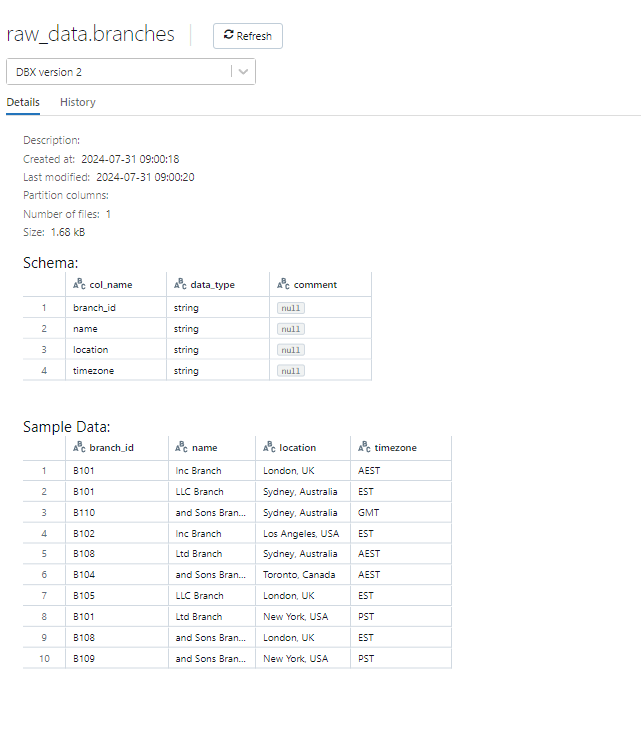
**Customer Table**

****

**Transaction Table**

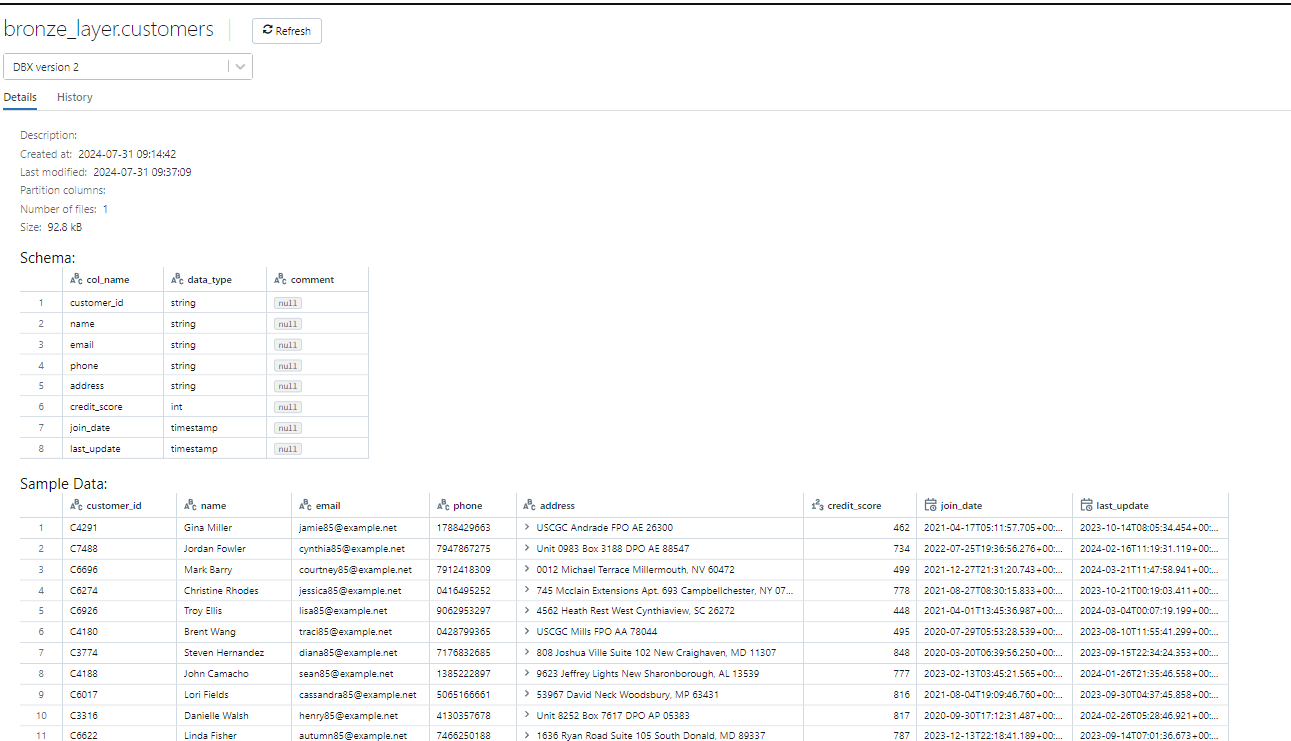
****

**Branch Table**

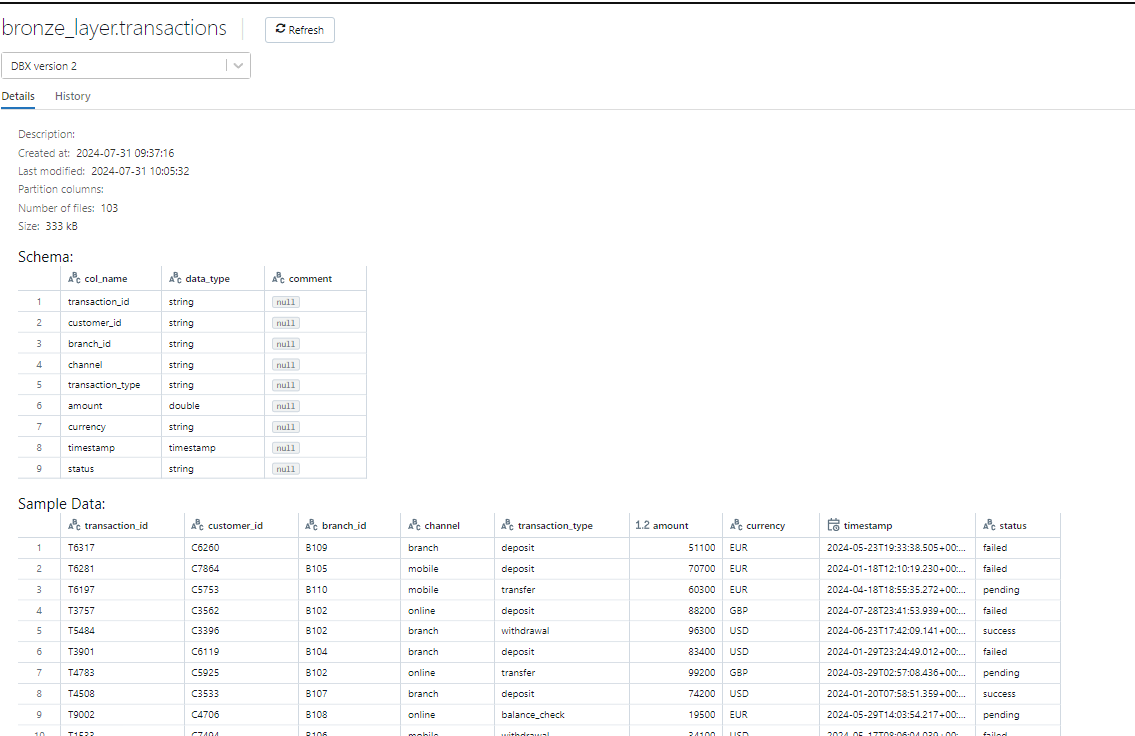
****

**Ingest Raw data to Bronze layer**

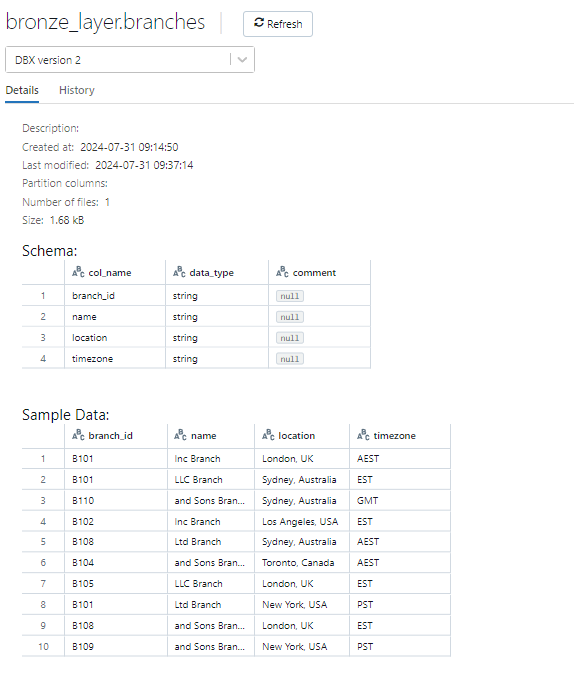
**Customer Table**

****

**Transaction table**

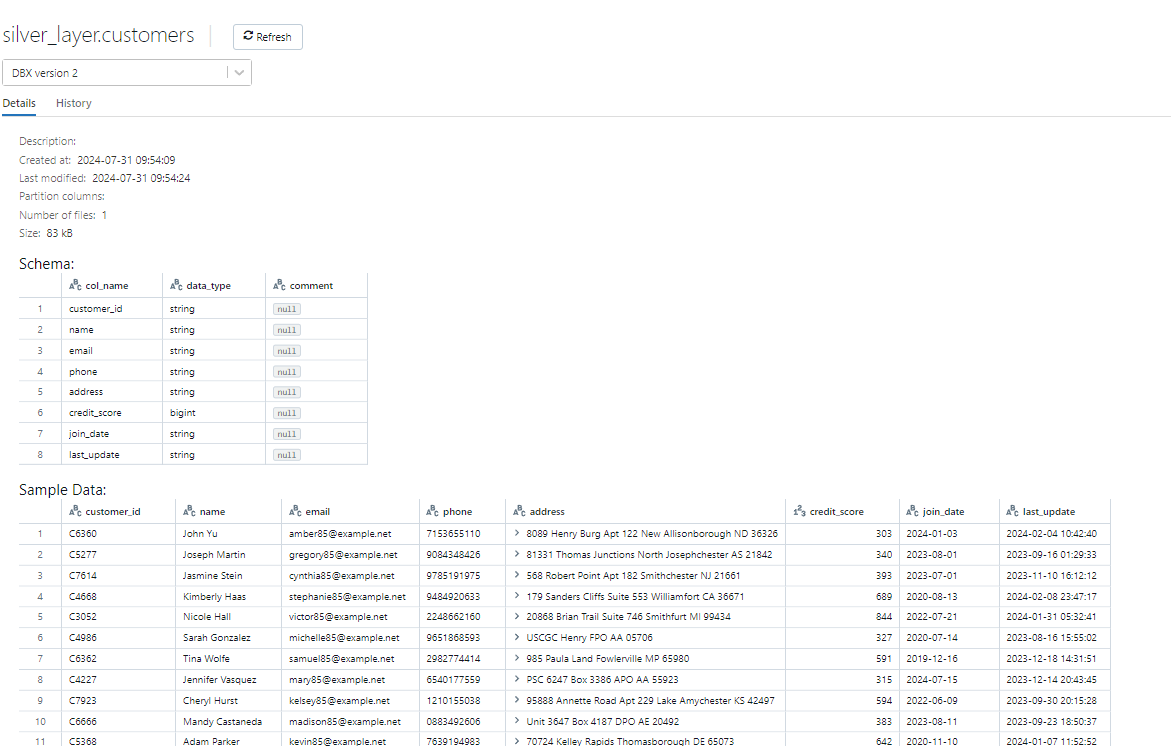
****

**Branch table**

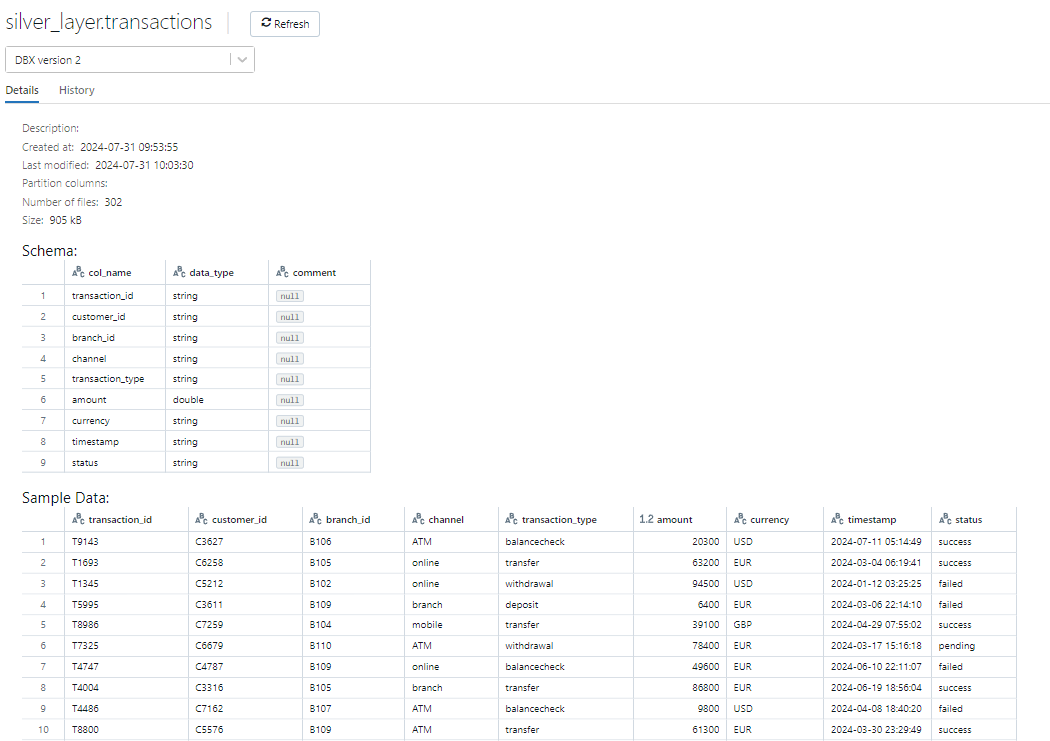
****

**Cleansing and load to Silver Layer**

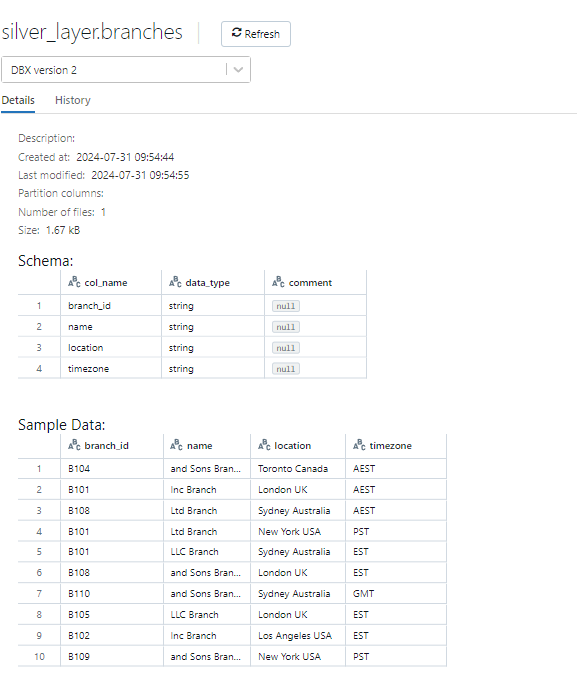
**Customer Table**

****

**Transactions Table**

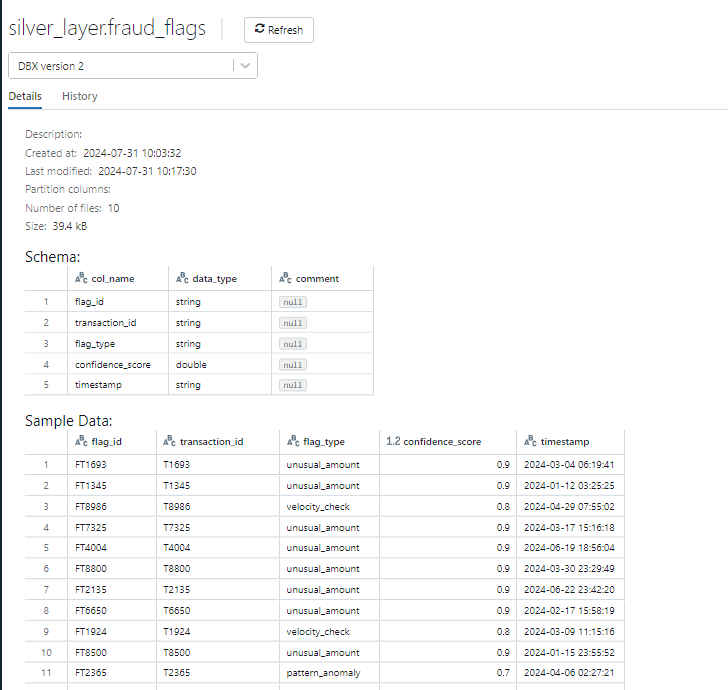
****

**Branch Table**

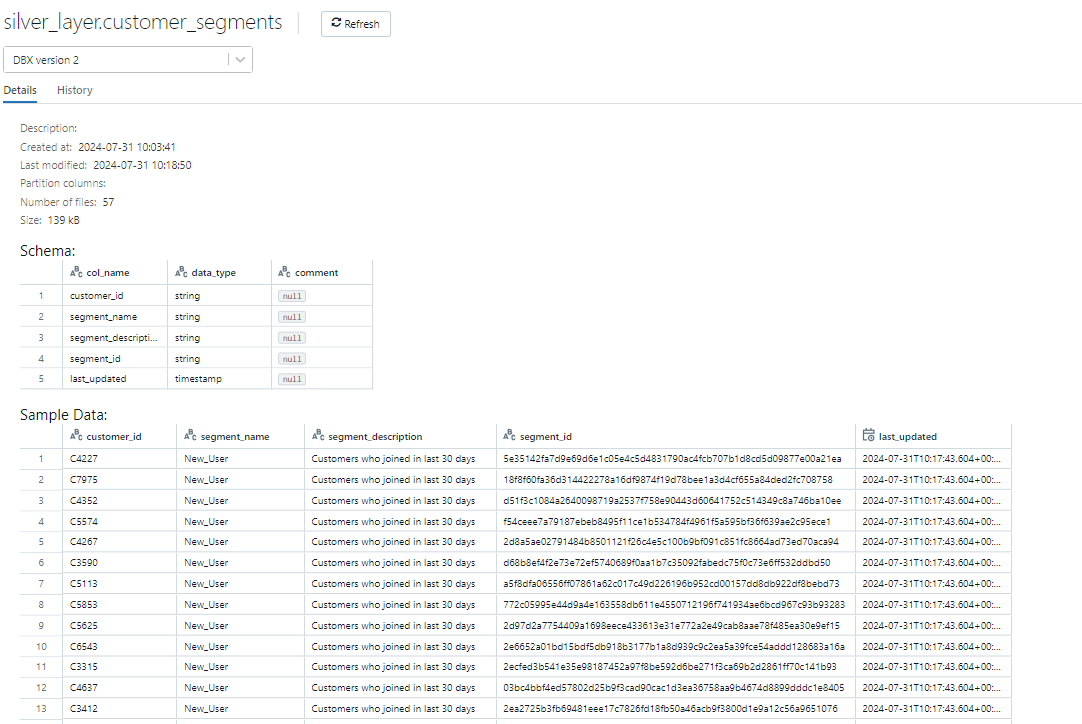
****

By aggregating data from the Transactions, Customer, and Branch tables, the fraud\_flags and customer\_segments tables were created.

**Fraud\_flags**

****

**Customer\_segements**

****