**Customer 360 Data Engineering Project GCP Bootcamp 3**

Sakina Banu  
Date: July 11, 2025

# Project Objective

The goal of this project is to build an end-to-end Customer 360 data integration pipeline on Google Cloud Platform. It involves ingesting raw data from multiple sources, cleaning it using PySpark, loading the cleaned data into BigQuery, aggregating business insights into a gold layer, and visualizing the results in Looker Studio.

# Step 1: Upload Raw Data to Cloud Storage (Bronze Layer)

All 9 raw CSV datasets were uploaded to the GCS bucket `gs://bronze-customer360/`. This forms the bronze layer, where data is kept in its raw, untouched form.

# Step 2: Data Cleaning in Google Colab Using PySpark

The Cloud Shell Editor was used to submit a PySpark job to **Dataproc** by running a shell command that created the cluster and submitted the cleaning script customer360\_cleaning.py. This script processed the raw datasets stored in the bronze GCS bucket. Each dataset was cleaned by renaming columns, dropping nulls and duplicates, and formatting timestamp fields. The cleaned output was written back to GCS under gs://curated-silver/{TableName}\_cleaned/ folders, such as Customers\_cleaned, Products\_cleaned, and so on.

# Step 3: Load Cleaned Data into BigQuery (Silver Layer)

Cleaned datasets were loaded into BigQuery dataset `customer360\_silver` using the following bq commands:

bq load --source\_format=CSV --skip\_leading\_rows=1 --replace=true --autodetect \  
sakina-gcp:customer360\_silver.Customers \  
gs://curated-silver/Customers\_cleaned/part-\*.csv

Similar commands were used for all 9 datasets.

# Step 4: Create Aggregated Tables in BigQuery (Gold Layer)

Gold layer tables were created using `bq query` CLI.

1. 1. Average Order Value per Customer

bq query --use\_legacy\_sql=false "  
CREATE OR REPLACE TABLE customer360\_gold.avg\_order\_value\_summary AS  
SELECT  
 CustomerID,  
 COUNT(TransactionID) AS TotalOrders,  
 SUM(TransactionAmount) AS TotalSpent,  
 ROUND(SUM(TransactionAmount) / COUNT(TransactionID), 2) AS AvgOrderValue  
FROM customer360\_silver.OnlineTransactions  
WHERE TransactionAmount IS NOT NULL  
GROUP BY CustomerID;"

1. 2. Loyalty Points Tier Summary

bq query --use\_legacy\_sql=false "  
CREATE OR REPLACE TABLE customer360\_gold.loyalty\_points\_summary AS  
SELECT  
 CustomerID,  
 PointsEarned,  
 TierLevel,  
 CASE  
 WHEN PointsEarned >= 4000 THEN 'Loyalty Champion'  
 WHEN PointsEarned >= 2500 THEN 'High-Value Customer'  
 ELSE 'Standard'  
 END AS CustomerSegment  
FROM customer360\_silver.LoyaltyAccounts;"

1. 3. InStore vs Online Transactions by Date

bq query --use\_legacy\_sql=false "  
CREATE OR REPLACE TABLE customer360\_gold.instore\_vs\_online\_summary AS  
SELECT DATE(DateTime) AS TransactionDate, 'InStore' AS TransactionType, COUNT(\*) AS TotalTransactions  
FROM customer360\_silver.InStoreTransactions  
GROUP BY TransactionDate  
UNION ALL  
SELECT DATE(DateTime), 'Online', COUNT(\*)  
FROM customer360\_silver.OnlineTransactions  
GROUP BY DATE(DateTime);"

1. 4. Agent Resolution Summary

bq query --use\_legacy\_sql=false "  
CREATE OR REPLACE TABLE customer360\_gold.agent\_resolution\_summary AS  
SELECT  
 AgentID,  
 COUNT(InteractionID) AS TotalInteractions,  
 SUM(CASE WHEN ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END) AS ResolvedCount,  
 ROUND(  
 SAFE\_DIVIDE(  
 SUM(CASE WHEN ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END),  
 COUNT(InteractionID)  
 ), 2  
 ) AS ResolutionRate  
FROM customer360\_silver.CustomerServiceInteractions  
GROUP BY AgentID;"

# Step 5: Build Looker Studio Dashboard

The final gold tables were connected to Looker Studio to build an interactive dashboard. It included visualizations for AOV, loyalty tiers, in-store vs online trends, and agent resolution rate.

# Conclusion

The project demonstrates a real-world GCP data engineering pipeline. From raw ingestion to transformation, automated loading, analytics, and dashboarding, each step reflects industry-standard practices.