



FAILED BANKING TRANSACTION ANALYSIS

**PRESENTED BY
CHETHANA G**

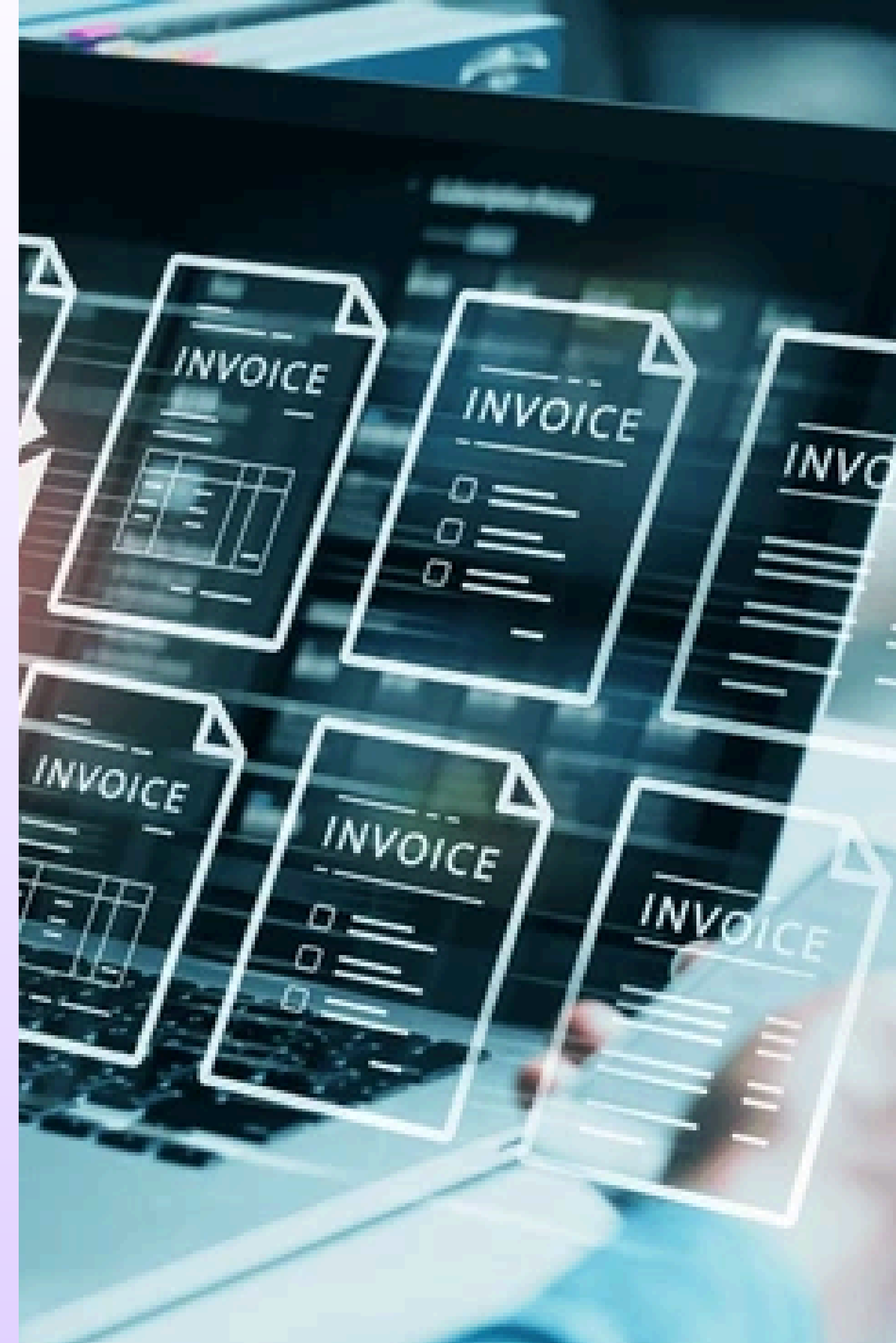
INTRODUCTION

In the modern banking ecosystem, transaction failures can have significant operational and customer experience implications. This project aims to build a scalable, cloud-native solution for analyzing failed banking transactions collected from various branches across multiple cities.



TECHNOLOGIES USED

- CSV, Python
- Google Cloud Storage
- Dataproc (PySpark)
- Cloud SQL (MySQL)
- BigQuery
- Looker Studio



IMPLEMENTATION

1. Data Ingestion

- Used gsutil to upload multiple CSV files from different branches into a Google Cloud Storage (GCS) bucket.
- Each file contains transaction data for 7 days from 3 cities × 5 branches = 15 files.

2. Data Processing with PySpark (Dataproc)

- Read all CSV files.
- Remove blank/null rows.
- Merge data into a single CSV file.

3. Failed Transaction Filtering

- PySpark job filtered only rows where Status = 'Failed'.
- Cleaned failed transaction data was saved to a new CSV.

4. Cloud SQL - MySQL

- Connected to Cloud SQL MySQL instance.
- Created a table to store failed transactions.

5. BigQuery Integration

- Connected BigQuery with Cloud SQL using external connection..
- Execute analytical queries on failed transactions by city, reason, branch, etc.

RESULT OF THE QUERY

Untitled query

Run

Save

Download

Share

Schedule

Open in

More

1

SELECT *

2

FROM EXTERNAL_QUERY(

3

"canvas-sentinel-457706-m1.us.bankdata",

4

"SELECT * FROM Transactions WHERE BranchCity = 'Bangalore'"

5

);

This query will process 0 B when run.

Query results

Save results

Open in

Job information

Results

Chart

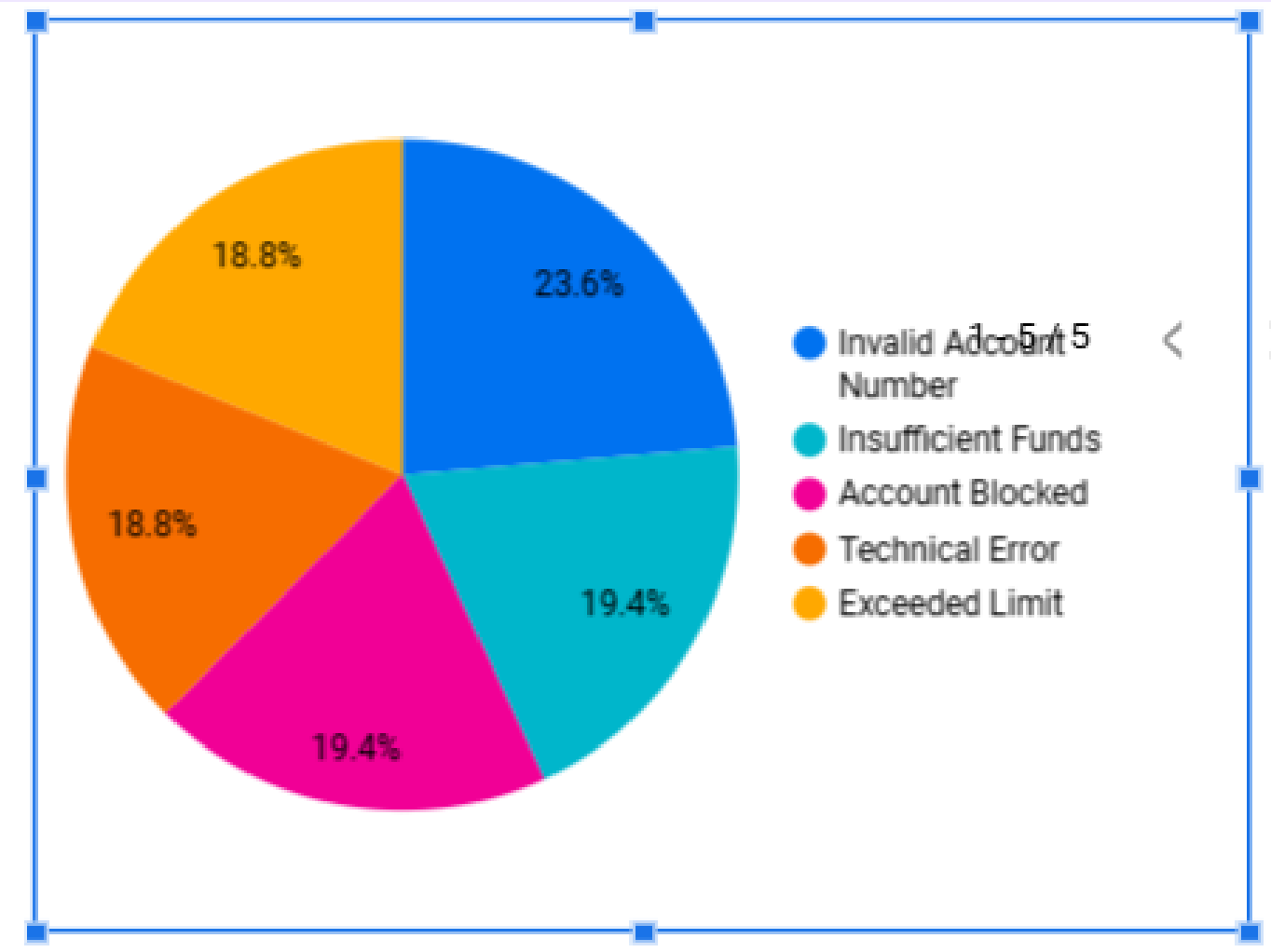
JSON

Execution details

Execution graph

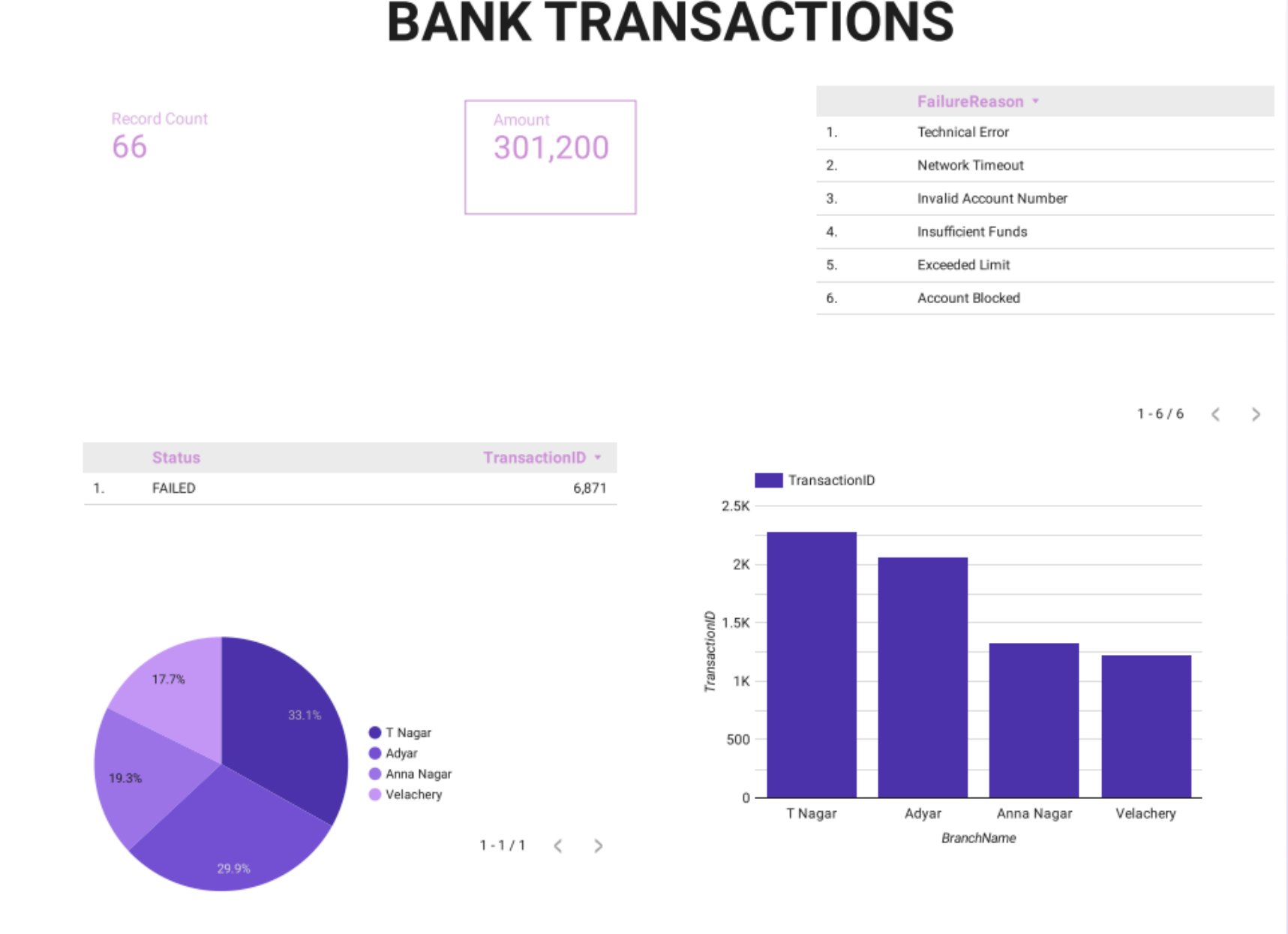
| Row | TransactionID | Date | Amount | Status | BranchID | BranchName | BranchCity |
|-----|---------------|------------|--------|--------|----------|-------------|------------|
| 1 | 4 | 2025-05-04 | 9134 | FAILED | WHI4 | Whitefield | Bangalore |
| 2 | 5 | 2025-05-02 | 3978 | FAILED | KOR2 | Koramangala | Bangalore |
| 3 | 6 | 2025-05-04 | 9066 | FAILED | KOR6 | Koramangala | Bangalore |
| 4 | 7 | 2025-05-07 | 9794 | FAILED | MG 4 | MG Road | Bangalore |
| 5 | 9 | 2025-05-02 | 3634 | FAILED | MG 7 | MG Road | Bangalore |
| 6 | 10 | 2025-05-02 | 3414 | FAILED | IND7 | Indiranagar | Bangalore |
| 7 | 13 | 2025-05-04 | 7331 | FAILED | IND6 | Indiranagar | Bangalore |

SELECT * FROM EXTERNAL_QUERY("canvas-sentinel-457706-m1.us.bankdata",
"SELECT FailureReason, COUNT(*) AS frequency FROM Transactions WHERE
Status = 'FAILED' GROUP BY FailureReason ORDER BY frequency DESC LIMIT 5");



| | FailureReason | frequency ▾ |
|----|------------------------|-------------|
| 1. | Invalid Account Number | 39 |
| 2. | Insufficient Funds | 32 |
| 3. | Account Blocked | 32 |
| 4. | Technical Error | 31 |
| 5. | Exceeded Limit | 31 |

VISUALIZATION



REAL TIME USAGE

Customer Service Improvement

- Quickly identify the failed transactions and notify the customers in near real-time.
- Reduce complaint resolution time by providing actionable failure reasons.

Branch Performance Monitoring

- Monitor failure trends across branches and cities.
- Identify underperforming or technically challenged branches.

CONCLUSION

The Failed Banking Transaction Analysis project successfully demonstrates a scalable, cloud-native solution for managing and analyzing high volumes of banking transaction data across multiple branches and cities. By leveraging Google Cloud Platform (GCP) services such as Cloud Storage, Dataproc (PySpark), Cloud SQL, and BigQuery.

- Streamlines data ingestion from distributed sources.
- Ensures data quality through effective cleaning and transformation pipelines.
- Automates the identification and extraction of failed transactions.
- Stores structured data in a reliable MySQL environment for secure access and integration.
- Enables powerful analytical queries and visual dashboards that provide actionable insights into failure trends.

THANK YOU