

# UPI Failure Analysis Project

This project analyzes UPI (Unified Payments Interface) transaction failures using SQL for data extraction and Power BI for visualization. The objective is to identify key patterns and insights that explain why UPI payments fail across regions, banks, and time frames.

---

## Problem Statement

UPI transactions in India are rapidly growing, but failure rates due to technical and operational issues impact user experience and trust. This project investigates failure trends and their underlying causes to improve service reliability.

---

## Tools & Technologies

- **SQL (MySQL/PostgreSQL):** Data cleaning and querying
  - **Power BI:** Dashboard development
  - **Excel/CSV:** Initial data storage
- 

## Dataset Overview

The dataset contains ~6000 UPI transactions with the following columns: - transaction\_id - bank\_name - region - transaction\_status (Success/Failure) - failure\_reason - transaction\_time - amount

---

## KPIs and SQL Queries

### 1. Total Transactions

```
sql SELECT COUNT(*) AS total_transactions FROM upi_transactions;
```

### 2. Total Failures

```
sql SELECT COUNT(*) AS failed_transactions FROM upi_transactions  
WHERE transaction_status = 'Failure';
```

### 3. Failure Rate (%)

```
sql SELECT ROUND(100.0 * (SELECT COUNT(*) FROM upi_transactions  
WHERE transaction_status = 'Failure') / (SELECT COUNT(*) FROM  
upi_transactions), 2 ) AS failure_rate_percentage;
```

## 4. Peak Failure Hours

```
sql SELECT EXTRACT(HOUR FROM transaction_time) AS hour, COUNT(*)  
AS failures FROM upi_transactions WHERE transaction_status =  
'Failure' GROUP BY hour ORDER BY failures DESC LIMIT 3;
```

## 5. Region-wise Failure Breakdown

```
sql SELECT region, COUNT(*) AS total_failures FROM  
upi_transactions WHERE transaction_status = 'Failure' GROUP BY  
region ORDER BY total_failures DESC;
```

## 6. Bank-wise Failure Breakdown

```
sql SELECT bank_name, COUNT(*) AS failed_txns FROM  
upi_transactions WHERE transaction_status = 'Failure' GROUP BY  
bank_name ORDER BY failed_txns DESC;
```

## 7. Failure Reasons Distribution

```
sql SELECT failure_reason, COUNT(*) AS count FROM  
upi_transactions WHERE transaction_status = 'Failure' GROUP BY  
failure_reason ORDER BY count DESC;
```

---

# Power BI Dashboard Features

- **KPI Cards:** Total transactions, failure rate, total failures
- **Line Graph:** Monthly/Hourly transaction failures
- **Bar Chart:** Failure reasons, region-wise failures, bank-wise breakdown
- **Filters/Slicers:** Region, bank, time

---

# Key Insights

- Peak UPI failures occur between **6 PM to 8 PM**.
- **SBI** recorded the **highest number of failures**.
- The leading cause of failure is **server issues**, accounting for nearly **14%** of all failed transactions.

---

# Conclusion

This project highlights how SQL + Power BI can drive actionable insights in fintech operations. Optimizing infrastructure during peak hours and improving server reliability can significantly reduce UPI failure rates.

---

## Repository Structure (Suggested)

```
UPI-Failure-Analysis/ ├── data/ | └─ upi_transactions.csv ──
queries/ | └─ upi_failure_kpis.sql ├── dashboard/ | └─
upi_dashboard.pbix ├── README.md └─ LICENSE
```

---

Feel free to fork, clone, and explore!