FinTech Transactions Analysis (SQL + Pandas)

By: Tushar Kalra

Date: 22 Oct 2025

Objective: Diagnose failures, quantify revenue, and surface actionable KPIs. This report demonstrates end-to-end analyst workflow: data audit, transformation, KPIs, and insights.

Executive Summary

This analysis quantifies payment reliability and revenue for a hypothetical fintech app. We compute daily transaction KPIs, identify failure hotspots by method, and highlight high-value customers. Outcome: a concise KPI pack and a prioritised improvement backlog.

Key KPIs

- Transaction Success Rate (TSR) = successful / total
- Failed Transactions % by method (UPI, Card, Wallet)
- Average Ticket Size = revenue / successful transactions
- Revenue by day and by user (Top Spenders)

Dataset and Assumptions

Input: transactions.csv with columns: txn_id, user_id, ts, amount, status(success/failed), method(UPI/Card/Wallet).
Assumptions: timestamps are UTC; refunds excluded from revenue; test txns removed.

SQL: Daily KPIs (sketch)

```
SELECT
  DATE(ts) AS day,
  COUNT(*) AS total_txn,
  SUM(CASE WHEN status='success' THEN 1 ELSE 0 END) AS success_txn,
  ROUND(SUM(CASE WHEN status='success' THEN 1 ELSE 0 END)*100.0/COUNT(*),2) AS tsr_pct,
  SUM(CASE WHEN status='success' THEN amount ELSE 0 END) AS revenue
FROM transactions
GROUP BY 1
ORDER BY 1;
```

Pandas: Minimal Notebook Flow

```
import pandas as pd
df = pd.read_csv('transactions.csv', parse_dates=['ts'])
df['day'] = df['ts'].dt.date
# Daily KPIs
daily = df.groupby('day').agg(
    total=('txn_id','count'),
    success=('status', lambda s: (s=='success').sum()),
    revenue=('amount', lambda a: a.where(df.loc[a.index,'status']=='success').sum()))
```

```
).reset_index()
daily['tsr_pct'] = (daily['success']/daily['total']*100).round(2)

# Failure by method
fail_by_method = (df[df.status=='failed']
    .groupby('method').size().sort_values(ascending=False))
```

Insights Template

- UPI failures spike between 8-9 PM; add smart-retry + alternate route.
- Card declines concentrated on specific BINs; add issuer-specific messaging.
- Top 5% users contribute 40% revenue; enable proactive support for them.

Next Actions

- Implement payment retry with fallback method selection.
- Expose TSR by method in live dashboard; set alert < 95%.
- Create VIP user list for high-touch support.

```
).reset_index()
daily['tsr_pct'] = (daily['success']/daily['total']*100).round(2)

# Failure by method
fail_by_method = (df[df.status=='failed']
    .groupby('method').size().sort_values(ascending=False))
```