

---

# Banking Transaction Analyzer — Detailed Project Report

## 1. Introduction

Financial institutions need to monitor transaction patterns for fraud detection and customer segmentation. This project analyzes banking transactions using SQL to identify trends, detect anomalies, and evaluate account activity.

## 2. Problem Statement

What are the patterns in banking transactions across different account types and customer segments? How can SQL queries help detect unusual or suspicious transaction behavior?

## 3. Dataset Description

- **Customers:** Personal info including age, city, and account opening date.
- **Accounts:** Account IDs, types (Savings, Checking), and balances.
- **Transactions:** Credit and debit transactions with timestamps and amounts.

## 4. Methodology / SQL Techniques

- Join customers to accounts and transactions for holistic analysis.
- Use aggregate functions (`SUM`, `COUNT`) grouped by month, account, and customer.
- Use conditional aggregations (`CASE`) to separate credit and debit volumes.
- Filter suspicious transactions using threshold criteria in `WHERE` or `HAVING`.

## 5. Key Findings

- **Transaction Volumes:** Cities like New York and Chicago have higher transaction counts, aligning with population density.
- **Suspicious Activity:** Some accounts have debit transactions disproportionately higher than credits, possibly indicating risk.
- **Account Usage:** Savings accounts have fewer transactions but higher average balances than checking accounts.

## 6. Conclusion

The project shows how SQL can effectively monitor transaction behavior and identify unusual patterns that warrant further investigation, contributing to fraud prevention and better customer profiling.

## 7. Recommendations / Future Work

- Integrate real-time data streams for live fraud detection.
- Include customer demographics and credit history for deeper risk scoring.
- Build dashboards for bank staff to visualize key metrics dynamically.