# Day 36 – SQL Challenge: Banking Transactions & Fraud Detection

This challenge simulates real-world banking data. You’ll practice advanced SQL concepts like window functions, CTEs, fraud detection logic, and optimization strategies. The dataset contains customers, accounts, transactions, and branches.

## Tables & Sample Data

Customers Table:  
CustomerID | Name | Age | City  
-----------|------------|-----|---------  
1 | Rahul Shah | 32 | Mumbai  
2 | Neha Rao | 45 | Delhi  
3 | Arjun Mehta| 28 | Bangalore  
4 | Priya Iyer | 36 | Chennai

Accounts Table:  
AccountID | CustomerID | BranchID | AccountType | Balance  
----------|------------|----------|-------------|---------

101 | 1 | 11 | Savings | 50000  
102 | 2 | 12 | Current | 120000  
103 | 3 | 11 | Savings | 30000  
104 | 4 | 13 | Current | 80000

Transactions Table:  
TransactionID | AccountID | Amount | TransactionType | TransactionDate  
--------------|-----------|--------|-----------------|----------------  
1001 | 101 | 2000 | Debit | 2023-05-01  
1002 | 101 | 1500 | Credit | 2023-05-02  
1003 | 102 | 50000 | Debit | 2023-05-03  
1004 | 103 | 20000 | Debit | 2023-05-04  
1005 | 104 | 10000 | Credit | 2023-05-05  
1006 | 101 | 25000 | Debit | 2023-05-06

Branches Table:  
BranchID | BranchName | City  
---------|------------|---------  
11 | Andheri | Mumbai  
12 | Connaught | Delhi  
13 | T Nagar | Chennai

## Questions

1. Retrieve all customers along with their account type and balance.

2. Find the top 2 highest balance accounts per branch using window functions.

3. Identify customers who made more than 2 transactions in May 2023.

4. Write a query to calculate the total debit and credit amount per account.

5. Find accounts where the balance dropped below 10,000 at any point (simulate fraud detection).

6. List customers who do not have any transactions.

7. Calculate the average transaction amount per branch.

8. Use NTILE(4) to divide customers into quartiles based on their account balance.

9. Find customers who had consecutive high-value debits (>20,000) using LAG().

10. Calculate churn rate: % of customers with zero transactions in May 2023.

Bonus:

Optimize a query to count total transactions per branch. Compare COUNT(\*) with COUNT(1) and discuss indexing impact.