# Problem Statement

A regional bank wants to better understand its customer base to improve service offerings, enhance customer satisfaction, and identify opportunities for growth. The bank is particularly interested in the behavior patterns of account usage, the effectiveness of its loan products, and overall customer engagement.

# Analysis Questions and SQL Queries

## 1. How many customers have multiple accounts, and what is the distribution of the number of accounts per customer?

SELECT CustomerID, COUNT(AccountID) AS NumberOfAccounts  
FROM Accounts  
GROUP BY CustomerID  
HAVING NumberOfAccounts > 1  
ORDER BY NumberOfAccounts DESC;

A screenshot of a computer

Description automatically generated

This query helps identify customers with multiple accounts, which can indicate higher engagement or diverse financial needs.

## 2. What is the average balance across different account types?

SELECT AccountType, AVG(Balance) AS AverageBalance  
FROM Accounts  
GROUP BY AccountType;

A screenshot of a computer

Description automatically generated

Understanding the average balance per account type (Savings, Checking, Loan) can help the bank tailor its financial products.

## 3. Identify the top 5 customers by total transaction amount in the last 6 months.

SELECT a.CustomerID, SUM(t.Amount) AS TotalTransactionAmount  
FROM Transactions t  
JOIN Accounts a ON t.AccountID = a.AccountID  
WHERE t.TransactionDate > CURDATE() - INTERVAL 6 MONTH  
GROUP BY a.CustomerID  
ORDER BY TotalTransactionAmount DESC  
LIMIT 5;

A screenshot of a computer

Description automatically generated

This query identifies the bank's most active customers by transaction volume, which could be useful for targeted marketing campaigns or loyalty programs.

## 4. How many loans have a balance greater than $5,000, and what is their total value?

SELECT COUNT(AccountID) AS NumberOfLoans, SUM(Balance) AS TotalLoanValue  
FROM Accounts  
WHERE AccountType = 'Loan' AND Balance < -5000;

A screenshot of a computer

Description automatically generated

This query provides insights into the size and value of larger loans, which can help assess the bank's risk exposure and loan product performance.

## 5. What is the monthly transaction volume (number of transactions) trend over the past year?

SELECT MONTH(TransactionDate) AS Month, YEAR(TransactionDate) AS Year, COUNT(TransactionID) AS NumberOfTransactions  
FROM Transactions  
WHERE TransactionDate >= CURDATE() - INTERVAL 1 YEAR  
GROUP BY YEAR(TransactionDate), MONTH(TransactionDate)  
ORDER BY Year, Month;

A screenshot of a data

Description automatically generated

Analyzing transaction volume trends can help the bank understand customer activity patterns and plan for peak periods.  
  
  
  
**Data Manipulation Language:**

CREATE TABLE Customers (

CustomerID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL

);

CREATE TABLE Accounts (

AccountID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT NOT NULL,

AccountType ENUM('Savings', 'Checking', 'Loan') NOT NULL,

Balance DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

CREATE TABLE Transactions (

TransactionID INT AUTO\_INCREMENT PRIMARY KEY,

AccountID INT NOT NULL,

TransactionType ENUM('Deposit', 'Withdrawal') NOT NULL,

Amount DECIMAL(10, 2) NOT NULL,

TransactionDate DATE NOT NULL,

FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)

);

INSERT INTO Customers (FirstName, LastName, Email) VALUES

('John', 'Doe', 'john.doe@email.com'),

('Jane', 'Doe', 'jane.doe@email.com'),

('Michael', 'Smith', 'michael.smith@email.com'),

('Emily', 'Jones', 'emily.jones@email.com'),

('William', 'Brown', 'william.brown@email.com'),

('Olivia', 'Johnson', 'olivia.johnson@email.com'),

('Noah', 'Davis', 'noah.davis@email.com'),

('Ava', 'Miller', 'ava.miller@email.com'),

('Ethan', 'Wilson', 'ethan.wilson@email.com'),

('Sophia', 'Moore', 'sophia.moore@email.com'),

('Alexander', 'Johnson', 'alexander.johnson@example.com'),

('Linda', 'Williams', 'linda.williams@example.com'),

('Ethan', 'Brown', 'ethan.brown@example.com'),

('Sophia', 'Davis', 'sophia.davis@example.com'),

('Aiden', 'Miller', 'aiden.miller@example.com'),

('Olivia', 'Wilson', 'olivia.wilson@example.com'),

('Lucas', 'Moore', 'lucas.moore@example.com'),

('Ava', 'Taylor', 'ava.taylor@example.com'),

('Mason', 'Anderson', 'mason.anderson@example.com'),

('Isabella', 'Thomas', 'isabella.thomas@example.com');

INSERT INTO Accounts (CustomerID, AccountType, Balance) VALUES

(1, 'Checking', 1500.00),

(1, 'Savings', 4500.50),

(2, 'Savings', 5500.00),

(3, 'Checking', 2000.00),

(4, 'Savings', 1500.00),

(5, 'Checking', 2500.00),

(5, 'Loan', -5000.00),

(6, 'Checking', 3500.00),

(7, 'Savings', 7500.00),

(8, 'Checking', 950.00),

(11, 'Savings', 5000.00),

(12, 'Checking', 1500.00),

(7, 'Savings', 2000.00),

(14, 'Checking', 2500.00),

(15, 'Loan', -10000.00),

(5, 'Savings', 3000.00),

(17, 'Checking', 4500.00),

(18, 'Savings', 7500.00),

(19, 'Checking', 1200.00),

(20, 'Loan', -5000.00);

INSERT INTO Transactions (AccountID, TransactionType, Amount, TransactionDate) VALUES

(1, 'Deposit', 500.00, '2023-01-10'),

(2, 'Withdrawal', 200.00, '2023-01-12'),

(2, 'Deposit', 1000.00, '2023-02-15'),

(3, 'Withdrawal', 550.00, '2023-02-20'),

(4, 'Deposit', 645.00, '2023-03-05'),

(5, 'Withdrawal', 300.00, '2023-03-10'),

(6, 'Deposit', 1200.00, '2023-04-25'),

(7, 'Withdrawal', 1250.00, '2023-05-30'),

(8, 'Deposit', 200.00, '2023-06-15'),

(9, 'Withdrawal', 100.00, '2023-07-20'),

(10, 'Deposit', 1500.00, '2023-08-10'),

(11, 'Deposit', 1000.00, '2023-01-10'),

(12, 'Withdrawal', 200.00, '2023-01-12'),

(13, 'Deposit', 1500.00, '2023-02-01'),

(14, 'Withdrawal', 100.00, '2023-02-05'),

(15, 'Deposit', 500.00, '2023-02-20'),

(16, 'Withdrawal', 600.00, '2023-03-15'),

(17, 'Deposit', 700.00, '2023-03-20'),

(18, 'Withdrawal', 400.00, '2023-04-10'),

(19, 'Deposit', 300.00, '2023-04-15'),

(20, 'Withdrawal', 500.00, '2023-05-01');

A screenshot of a computer

Description automatically generated