



SQL Case Study Challenge 4 Financial Analysis

By Dr. Neha Chaudhry



Introduction

You are a Finance Analyst working for '[The Big Bank](#)'
You have been tasked with finding out about your
customers and their banking behaviour. Examine the
accounts they hold and the type of transactions they make
to develop greater insight into your customers.



About Dataset

The financial analysis case study challenge involves analyzing a dataset containing information about customers, branches, accounts, and transactions.

The dataset is structured using relational database tables and requires SQL queries to extract insights.



Tools Used



Table 1

Create and Insert in Customers Table

```
-- Create the Customers table
CREATE TABLE Customers (
CustomerID INT PRIMARY KEY,
FirstName VARCHAR(50) NOT NULL,
LastName VARCHAR(50) NOT NULL,
City VARCHAR(50) NOT NULL,
State VARCHAR(2) NOT NULL
);

-- Populate the Customers table
INSERT INTO Customers (CustomerID, FirstName, LastName, City, State)
VALUES (1, 'John', 'Doe', 'New York', 'NY'),
(2, 'Jane', 'Doe', 'New York', 'NY'),
(3, 'Bob', 'Smith', 'San Francisco', 'CA'),
(4, 'Alice', 'Johnson', 'San Francisco', 'CA'),
(5, 'Michael', 'Lee', 'Los Angeles', 'CA'),
(6, 'Jennifer', 'Wang', 'Los Angeles', 'CA');
```

Table 2

Create and Insert in Branches Table

```
-- Create the Branches table
CREATE TABLE Branches (
BranchID INT PRIMARY KEY,
BranchName VARCHAR(50) NOT NULL,
City VARCHAR(50) NOT NULL,
State VARCHAR(2) NOT NULL
);

-- Populate the Branches table
INSERT INTO Branches (BranchID, BranchName, City, State)
VALUES (1, 'Main', 'New York', 'NY'),
(2, 'Downtown', 'San Francisco', 'CA'),
(3, 'West LA', 'Los Angeles', 'CA'),
(4, 'East LA', 'Los Angeles', 'CA'),
(5, 'Uptown', 'New York', 'NY'),
(6, 'Financial District', 'San Francisco', 'CA'),
(7, 'Midtown', 'New York', 'NY'),
(8, 'South Bay', 'San Francisco', 'CA'),
(9, 'Downtown', 'Los Angeles', 'CA'),
(10, 'Chinatown', 'New York', 'NY'),
(11, 'Marina', 'San Francisco', 'CA'),
(12, 'Beverly Hills', 'Los Angeles', 'CA'),
(13, 'Brooklyn', 'New York', 'NY'),
(14, 'North Beach', 'San Francisco', 'CA'),
(15, 'Pasadena', 'Los Angeles', 'CA');
```

Table 3

Create and Insert in Accounts Table

```
-- Create the Accounts table
CREATE TABLE Accounts (
    AccountID INT PRIMARY KEY,
    CustomerID INT NOT NULL,
    BranchID INT NOT NULL,
    AccountType VARCHAR(50) NOT NULL,
    Balance DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (BranchID) REFERENCES Branches(BranchID)
);
```

```
-- Populate the Accounts table
INSERT INTO Accounts (AccountID, CustomerID, BranchID, AccountType, Balance)
VALUES (1, 1, 5, 'Checking', 1000.00),
       (2, 1, 5, 'Savings', 5000.00),
       (3, 2, 1, 'Checking', 2500.00),
       (4, 2, 1, 'Savings', 10000.00),
       (5, 3, 2, 'Checking', 7500.00),
       (6, 3, 2, 'Savings', 15000.00),
       (7, 4, 8, 'Checking', 5000.00),
       (8, 4, 8, 'Savings', 20000.00),
       (9, 5, 14, 'Checking', 10000.00),
       (10, 5, 14, 'Savings', 50000.00),
       (11, 6, 2, 'Checking', 5000.00),
       (12, 6, 2, 'Savings', 10000.00),
       (13, 1, 5, 'Credit Card', -500.00),
       (14, 2, 1, 'Credit Card', -1000.00),
       (15, 3, 2, 'Credit Card', -2000.00);
```

Table 4

Create and Insert in Transactions Table

— Create the Transactions table

```
CREATE TABLE Transactions (
    TransactionID INT PRIMARY KEY,
    AccountID INT NOT NULL,
    TransactionDate DATE NOT NULL,
    Amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)
);
```

— Populate the Transactions table

```
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount)
VALUES (1, 1, '2022-01-01', -500.00),
       (2, 1, '2022-01-02', -250.00),
       (3, 2, '2022-01-03', 1000.00),
       (4, 3, '2022-01-04', -1000.00),
       (5, 3, '2022-01-05', 500.00),
       (6, 4, '2022-01-06', 1000.00),
       (7, 4, '2022-01-07', -500.00),
       (8, 5, '2022-01-08', -2500.00),
       (9, 6, '2022-01-09', 500.00),
       (10, 6, '2022-01-10', -1000.00),
       (11, 7, '2022-01-11', -500.00),
       (12, 7, '2022-01-12', -250.00),
       (13, 8, '2022-01-13', 1000.00),
       (14, 8, '2022-01-14', -1000.00),
       (15, 9, '2022-01-15', 500.00);
```

Table 4

Create and Insert in Transactions Table

— Create the Transactions table

```
CREATE TABLE Transactions (
    TransactionID INT PRIMARY KEY,
    AccountID INT NOT NULL,
    TransactionDate DATE NOT NULL,
    Amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)
);
```

— Populate the Transactions table

```
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount)
VALUES (1, 1, '2022-01-01', -500.00),
       (2, 1, '2022-01-02', -250.00),
       (3, 2, '2022-01-03', 1000.00),
       (4, 3, '2022-01-04', -1000.00),
       (5, 3, '2022-01-05', 500.00),
       (6, 4, '2022-01-06', 1000.00),
       (7, 4, '2022-01-07', -500.00),
       (8, 5, '2022-01-08', -2500.00),
       (9, 6, '2022-01-09', 500.00),
       (10, 6, '2022-01-10', -1000.00),
       (11, 7, '2022-01-11', -500.00),
       (12, 7, '2022-01-12', -250.00),
       (13, 8, '2022-01-13', 1000.00),
       (14, 8, '2022-01-14', -1000.00),
       (15, 9, '2022-01-15', 500.00);
```

Dataset

Customers

CustomerID	FirstName	LastName	City	State
1	John	Doe	New York	NY
2	Jane	Doe	New York	NY
3	Bob	Smith	San Francisco	CA
4	Alice	Johnson	San Francisco	CA
5	Michael	Lee	Los Angeles	CA
6	Jennifer	Wang	Los Angeles	CA

Accounts

AccountID	CustomerID	BranchID	AccountType	Balance
1	1	5	Checking	1000
2	1	5	Savings	5000
3	2	1	Checking	2500
4	2	1	Savings	####
5	3	2	Checking	7500
6	3	2	Savings	####
7	4	8	Checking	5000
8	4	8	Savings	####
9	5	14	Checking	####
10	5	14	Savings	####
11	6	2	Checking	5000
12	6	2	Savings	####
13	1	5	Credit Card	-500
14	2	1	Credit Card	-1000
15	3	2	Credit Card	-2000

Transactions

TransactionID	AccoutID	TransactionDate	Amount
1	1	2022-01-01	-500
2	1	2022-01-02	-250
3	2	2022-01-03	1000
4	3	2022-01-04	-1000
5	3	2022-01-05	500
6	4	2022-01-06	1000
7	4	2022-01-07	-500
8	5	2022-01-08	-2500
9	6	2022-01-09	500
10	6	2022-01-10	-1000
11	7	2022-01-11	-500
12	7	2022-01-12	-250
13	8	2022-01-13	1000
14	8	2022-01-14	-1000
15	9	2022-01-15	500

Branches

BranchID	BranchName	City	State
1	Main	New York	NY
2	Downtown	San Francisco	CA
3	West LA	Los Angeles	CA
4	East LA	Los Angeles	CA
5	Uptown	New York	NY
6	Financial District	San Francisco	CA
7	Midtown	New York	NY
8	South Bay	San Francisco	CA
9	Downtown	Los Angeles	CA
10	Chinatown	New York	NY
11	Marina	San Francisco	CA
12	Beverly Hills	Los Angeles	CA
13	Brooklyn	New York	NY
14	North Beach	San Francisco	CA
15	Pasadena	Los Angeles	CA

Questions

1. What are the names
of all the customers who
live in New York?

3. What is the total balance
of all checking accounts?

2. What is the total number of
accounts in the Accounts table?

4. What is the total
balance of all
accounts associated
with customers who
live in Los Angeles?

5. Which branch has the highest
average account balance?

Questions

6. Which customer has the highest current balance in their accounts?

8. Which branch has the highest total balance across all of its accounts?

7. Which customer has made the most transactions in the Transactions table?

9. Which customer has the highest total balance across all of their accounts, including savings and checking accounts?

10. Which branch has the highest number of transactions in the Transactions table?

Question 1

1. What are the names of all the customers who live in New York?



Query



```
SELECT CustomerID, CONCAT(FirstName, ' ', LastName) AS Customer_name  
FROM Customers  
WHERE City = 'New York';
```

Output

CustomerID	Customer_name
1	John Doe
2	Jane Doe

Question 2

2. What is the total number of accounts in the Accounts table?



Query

```
SELECT COUNT(*) AS TotalAccounts  
FROM Accounts;
```

Output

TotalAccounts
15

Question 3

3. What is the total balance of all checking accounts?

Query

```
SELECT ROUND(SUM(Balance), 0) AS Total_Balance  
FROM Accounts  
WHERE AccountType = 'Checking';
```

Output

Result Grid

Total_Balance
31000

Question 4

4. What is the total balance of all accounts associated with customers who live in Los Angeles?

**Query**

```
SELECT ROUND(SUM(a.Balance), 0) AS Total_Balance  
FROM Accounts a  
JOIN Customers c  
ON a.CustomerID = c.CustomerID  
WHERE c.City = 'Los Angeles';
```

snappify.com

Output

Total_Balance
75000



Question 5

5. Which branch has the highest average account balance?

Query

```
SELECT b.BranchID, b.BranchName,  
ROUND(AVG(a.Balance), 0) AS AvgBalance  
FROM Branches b  
JOIN Accounts a  
ON b.BranchID = a.BranchID  
GROUP BY b.BranchID, b.BranchName  
ORDER BY AvgBalance DESC  
LIMIT 1;
```

Output

BranchID	BranchName	AvgBalance
14	North Beach	30000

Question 6

6. Which customer has the highest current balance in their accounts?

**Query**

```
SELECT c.CustomerID,  
CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,  
ROUND(SUM(a.Balance), 0) AS TotalBalance  
FROM Customers c  
JOIN Accounts a  
ON c.CustomerID = a.CustomerID  
GROUP BY c.CustomerID, CustomerName  
ORDER BY TotalBalance DESC  
LIMIT 1;
```

Output

Result Grid | Filter Rows:

	CustomerID	CustomerName	TotalBalance
▶	5	Michael Lee	60000



Question 7

7. Which customer has made the most transactions in the Transactions table?



Query

```
SELECT c.CustomerID, c.FirstName, c.LastName,  
COUNT(t.TransactionID) AS TransactionCount  
FROM Customers c  
JOIN Accounts a  
ON c.CustomerID = a.CustomerID  
JOIN Transactions t ON a.AccountID = t.AccountID  
GROUP BY c.CustomerID, c.FirstName, c.LastName  
ORDER BY TransactionCount DESC;
```

Output



CustomerID	FirstName	LastName	TransactionCount
2	Jane	Doe	4
4	Alice	Johnson	4
1	John	Doe	3
3	Bob	Smith	3
5	Michael	Lee	1

Question 8

8. Which branch has the highest total balance across all of its accounts?

Query

```
SELECT b.BranchName,  
ROUND(SUM(a.Balance), 0) AS TotalBalance  
FROM Branches b  
JOIN Accounts a  
ON b.BranchID = a.BranchID  
GROUP BY b.BranchName  
ORDER BY TotalBalance DESC  
LIMIT 1;
```

Output



A diagram showing the flow of data from a MySQL database to a web browser. A red arrow points from a MySQL icon to a yellow-bordered table. The table has two columns: 'BranchName' and 'TotalBalance'. The row shows 'North Beach' with a value of '60000'. The table is displayed within a web browser window, indicated by a magnifying glass icon.

BranchName	TotalBalance
North Beach	60000

Question 9

9. Which customer has the highest total balance across all of their accounts, including savings and checking accounts?



Query

```
SELECT  
    a.CustomerID, CONCAT(FirstName, " ", LastName) AS Customer_name,  
    ROUND(SUM(a.balance), 0) AS total_balance  
FROM accounts a  
JOIN customers c  
ON a.CustomerID = c.CustomerID  
WHERE a.Accounttype IN ("Savings", "Checking")  
GROUP BY a.CustomerID, Customer_name  
ORDER BY total_balance DESC;
```

Output



	CustomerID	Customer_name	total_balance
▶	5	Michael Lee	60000
	4	Alice Johnson	25000
	3	Bob Smith	22500
	6	Jennifer Wang	15000
	2	Jane Doe	12500
	1	John Doe	6000

Question 10

10. Which branch has the highest number of transactions in the Transactions table?



Query

```
SELECT  
    b.BranchName,  
    COUNT(t.TransactionID) AS Number_of_transactions  
FROM branches b JOIN accounts a  
ON a.BranchID = b.BranchID  
JOIN transactions t ON a.AccountID = t.AccountID  
GROUP BY b.BranchName  
ORDER BY Number_of_transactions DESC;
```

Output



BranchName	Number_of_transactions
Main	4
South Bay	4
Downtown	3
Uptown	3
North Beach	1



Insights

The query retrieves the customer names by joining the Customers table and filtering for customers who live in New York. This information can be useful for targeted marketing or analysis focused on customers residing in New York.

Q1

This query calculates the count of accounts in the Accounts table. It provides an overview of the total number of accounts stored in the system, which can be useful for tracking account growth or monitoring database integrity.

Q2

The query calculates the sum of balances for all checking accounts in the Accounts table. This information is helpful for understanding the cumulative balance held in checking accounts, aiding financial analysis or assessing liquidity.

Q3

The query retrieves the total balance of accounts linked to customers residing in Los Angeles. This data can offer insights into the financial footprint of customers in Los Angeles and facilitate targeted marketing or branch-level analysis.

Q4



Insights

The query calculates the average account balance for each branch and identifies the branch with the highest average balance. This information can guide strategic decisions such as branch performance evaluation or resource allocation.

Q5

The query determines the customer with the highest current balance by calculating the total balance across their accounts. This insight is valuable for identifying high-value customers or prioritizing service for those with significant account balances.

Q6

The query identifies the customer who has made the most transactions by counting the number of transactions associated with each customer. This information is helpful for identifying active customers or analyzing transaction patterns.

Q7

The query calculates the total balance for branch and identifies the branch with the cumulative balance across its accounts. This can guide branch-level analysis, resource allocation, or help identify branches with the most significant financial impact.



Insights

The query calculates the total balance for each customer across all their accounts, including savings and checking accounts. It helps identify customers with the highest combined wealth, which can inform personalized services or financial planning.

Q9

The query calculates the number of transactions for each branch and identifies the branch with the highest count. This information can indicate the branch's activity level, transaction volume, or help identify branches requiring additional resources or attention.

Q10

Summary

In conclusion, this finance case study demonstrates the power of SQL in extracting meaningful insights from customer data and enhancing decision-making in the banking industry. By analyzing customer accounts and transactions, financial institutions can gain a deeper understanding of their customers' behavior, preferences, and needs.



This knowledge enables targeted marketing campaigns, personalized services, and resource allocation strategies that cater to specific customer segments. Furthermore, optimizing branch performance and identifying high-value customers contribute to improved profitability and customer satisfaction. Overall, leveraging SQL in financial analysis empowers finance analysts to drive growth, enhance customer experiences, and stay ahead in a highly competitive industry.



Thank You

