Bank of Sait - Database System Design

Written by:
Kanav Saini

Table of Contents

https://github.com/kanavv-tech/Data-science/upload/main

Bank of Sait

Data integrity, security, and efficiency are critical in today's banking environment. The Bank of Sait (BOS) realised that to handle its wide range of banking operations—from managing client data to monitoring financial transactions—it needed a strong and dependable database system. To streamline operations, maintain regulatory compliance, and improve customer service, BOS developed a secure and effective database system, which is examined in this case study.

Mission Statement

To support banking activities, the BOS database system's major goal is to create and maintain a dependable, effective, and secure infrastructure.

This system uses automated routine operations and centralised data management to guarantee excellent service delivery, regulatory compliance, and efficient decision-making.

Business Rules and Objectives

The following guidelines were put into place to ensure the dependability and integrity of financial transactions:

1. Centralised Administration of Data

- Keep client, account, and transaction data in a single system to guarantee consistency and accessibility.
- By giving client accounts and transactions unique IDs, you can prevent data duplication while saving money on storage and improving productivity.
- To offer a comprehensive banking experience, facilitate smooth integration with outside financial services.

2. Accuracy and Security of Transactions

- Automate tracking and recording of transactions to reduce fraud and human mistake.
- To safeguard private financial information from online attacks, use encryption and multi-factor authentication.
- Make sure that transactions are processed in real time to enhance client satisfaction and minimise account balance disparities.
- Create audit logs to ensure transparency in all financial transactions and to track compliance and fraud.

3.Efficiency in Operations

- Reduce errors by automating repetitive processes like credit card issue, fund transfers, and loan approvals.

- Increase workflow productivity by incorporating analytics powered by AI to forecast client financial requirements and offer customised banking solutions.
- Enhance data access by putting cloud-based storage options into place, which will enable bank staff to quickly access and update customer data.

4.Data-Informed Decision Making

- Create insights on consumer spending habits, behaviour, and creditworthiness to provide individualised financial products.
- By examining transaction patterns and customer interactions, you can improve banking operations and allocate resources more effectively.
- Forecast possible defaults, evaluate loan performance, and enhance risk assessment models by utilising predictive analytics.
- Improve consumer engagement with tailored promotions and services by using big data to hone marketing strategy.

Subjects

1. **Database security:** It is the process of protecting financial data by putting multi-factor authentication, encryption, and fraud detection systems into place. This involves protecting sensitive client data and preventing unwanted access using cutting-edge cryptographic algorithms.

2. **Data optimisation:** It refers to methods for reducing redundancy, improving query performance, and streamlining data storage. This includes caching techniques, normalisation, and indexing algorithms to speed up response times and save storage expenses.

3. **Automated Banking Solution:** AI-powered systems for fraud detection, transaction monitoring, and improving customer service are known as automated banking solutions. For smooth customer service, these include chatbot integrations and machine learning models that examine transaction trends to identify irregularities.

4. **Regulatory Compliance:** Ensuring compliance with financial rules and regulations, including Basel III, GDPR, and PCI DSS, is known as regulatory compliance. For banks to comply with regulatory requirements and uphold openness, compliance frameworks, audit trails, and reporting procedures must be put in place.

List of Tables

- **Branches:** Keeps track of branch names, addresses, and resources.
- **Clients:** Preserves client information, including name, birthdate, phone number, and identity documentation.
- **Accounts:** Keeps track of account types and balances (business, checking, savings, etc.).
- **employees:** Lists bank workers along with their positions and contact information.
- **Transactions:** Records transfers, deposits, and withdrawals together with amounts and timestamps.
- **Loans:** Shops gave out loans to consumers and collected the remaining balances.
- **Credit Cards:** Monitors credit card balances, expiration dates, and current balances.

https://github.com/kanavv-tech/Data-science/upload/main

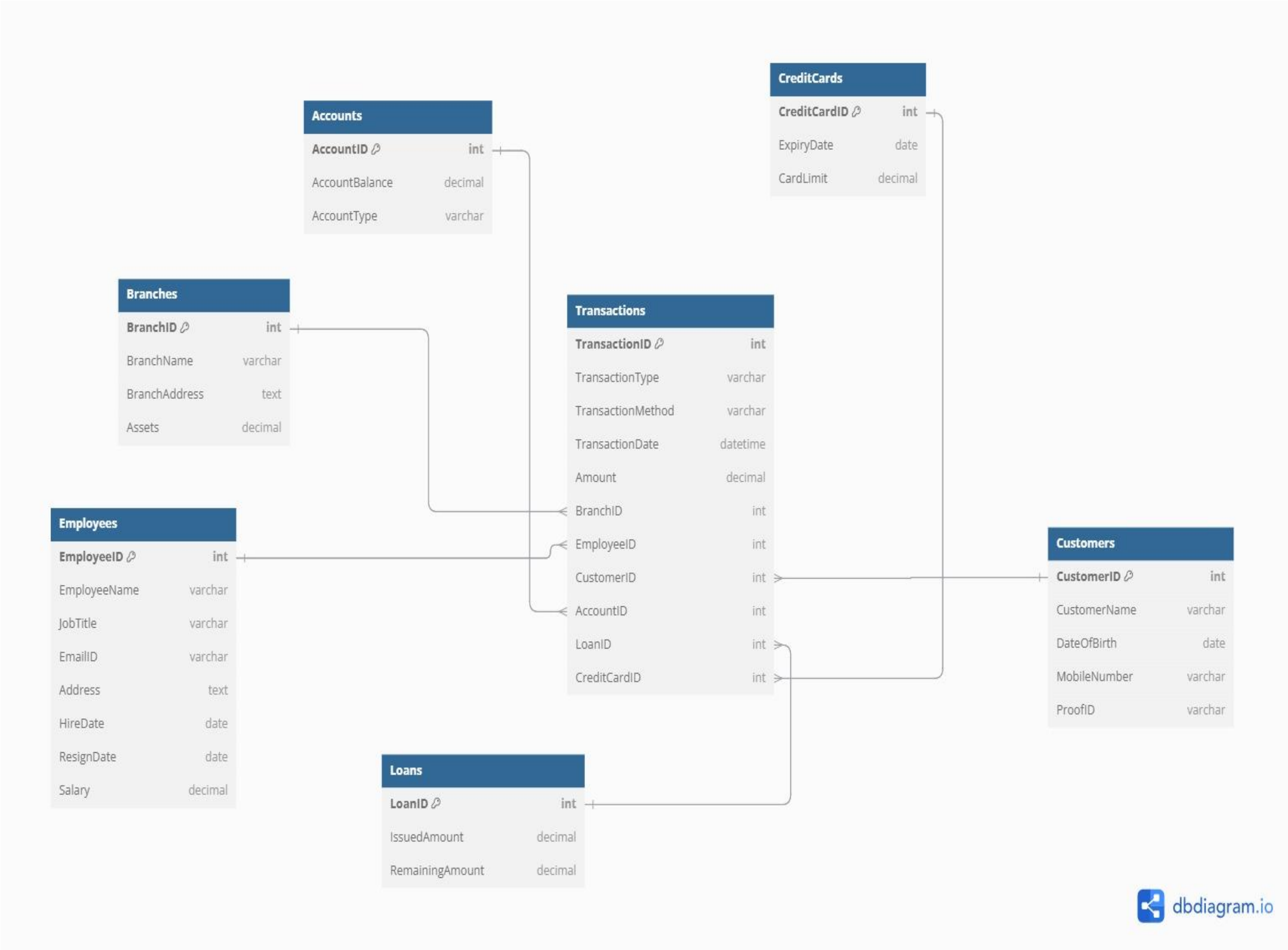
List of Attributes

- **Customer ID:** A special number assigned to each customer.
- **Account Type:** Indicates if the account is for business, savings, checking, etc.
- **Transaction Date:** A financial operations timestamp.
- **Loan Amount:** The sum of all loans granted and those still pending.
- **Branch ID:** Indicates which branch is in charge of a certain transaction.

Entity-Relationship (ER) Diagram and Relationships

The system utilizes an **ER model** to define relationships between entities:

- One Branch can have many Transactions (1: N).
- One Employee can handle many Transactions (1: N).
- One Customer can perform many Transactions (1: N).
- One Account can be linked to many Transactions (1: N).
- One Loan can be associated with multiple Transactions (1: N).
- One Credit Card can have several Transactions (1: N).



CreditCards	
CreditCardID	int
ExpiryDate	date
CardLimit	decimal

Accounts	
AccountID	int
AccountBalance	decimal
AccountType	varchar

Branches	
BranchID	int
BranchName	varchar
BranchAddress	text
Assets	decimal

Transactions	
TransactionID	int
TransactionType	varchar
TransactionMethod	varchar
TransactionDate	datetime
Amount	decimal
BranchID	int
EmployeeID	int
CustomerID	int
AccountID	int
LoanID	int
CreditCardID	int

Employees	
EmployeeID	int
EmployeeName	varchar
JobTitle	varchar
EmailID	varchar
Address	text
HireDate	date
ResignDate	date
Salary	decimal

Customers	
CustomerID	int
CustomerName	varchar
DateOfBirth	date
MobileNumber	varchar
ProofID	varchar

Loans	
LoanID	int
IssuedAmount	decimal
RemainingAmount	decimal

FINAL LIST OF TABLES:

https://github.com/kanavv-tech/Data-science/upload/main

1.Branches

Branches	
Branchid	int
BranchName	Char
BranchAddress	Varchar
Asset	decimal

2.Account:

Account	
AccountId	int
AccountBalance	Decimal
AccountType	varchar

3.Employee:

Emoloyee	
EmployeeId	int
Employee Name	varchar
Job title	varchar
Address	text
EmailId	varchar
HireDate	date

4.Transaction:

Transaction	
TransactionId	int
TransactionType	varchar

https://github.com/kanavv-tech/Data-science/upload/main

TransactionMethod	varchar
TransactionDate	date
Amount	decimal

5.Loans:

Loans	
LoanID	int
IssuedAmount	Decimal
RemainingAmount	decimal

6.Customer:

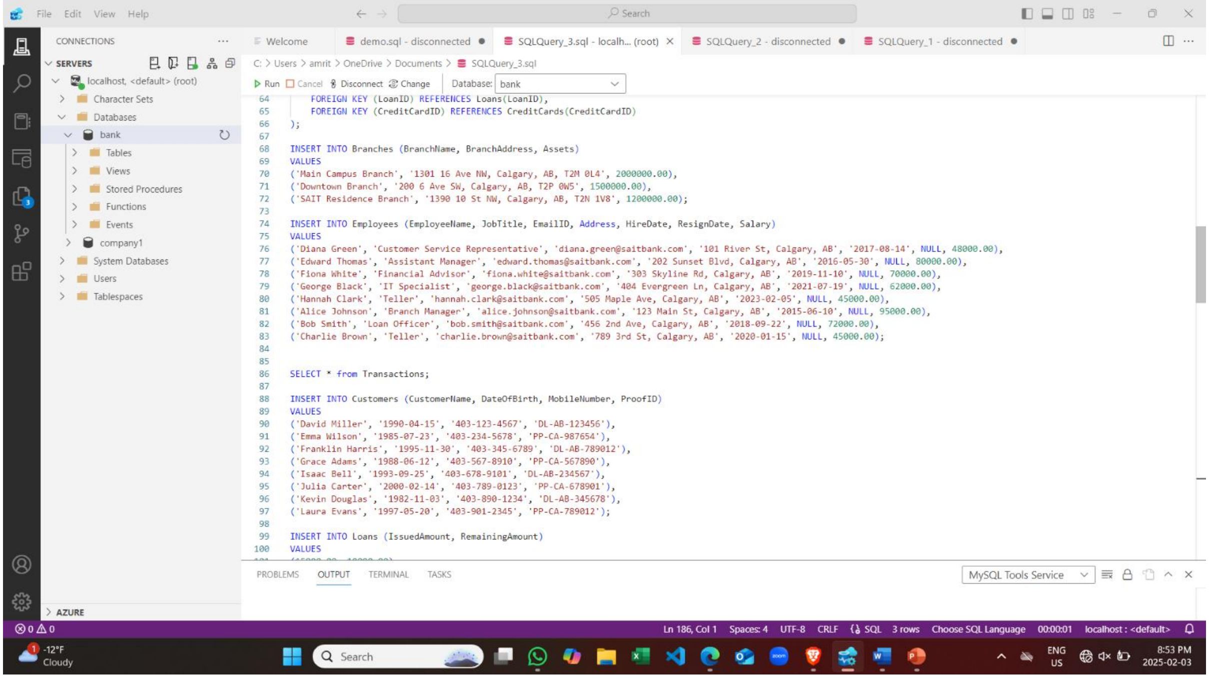
Customer	
CustomerId	int
CustomerName	varchar
DateOfBirth	Date
MobileNumber	Varchar
Proofid	varchar

7.Credit Card

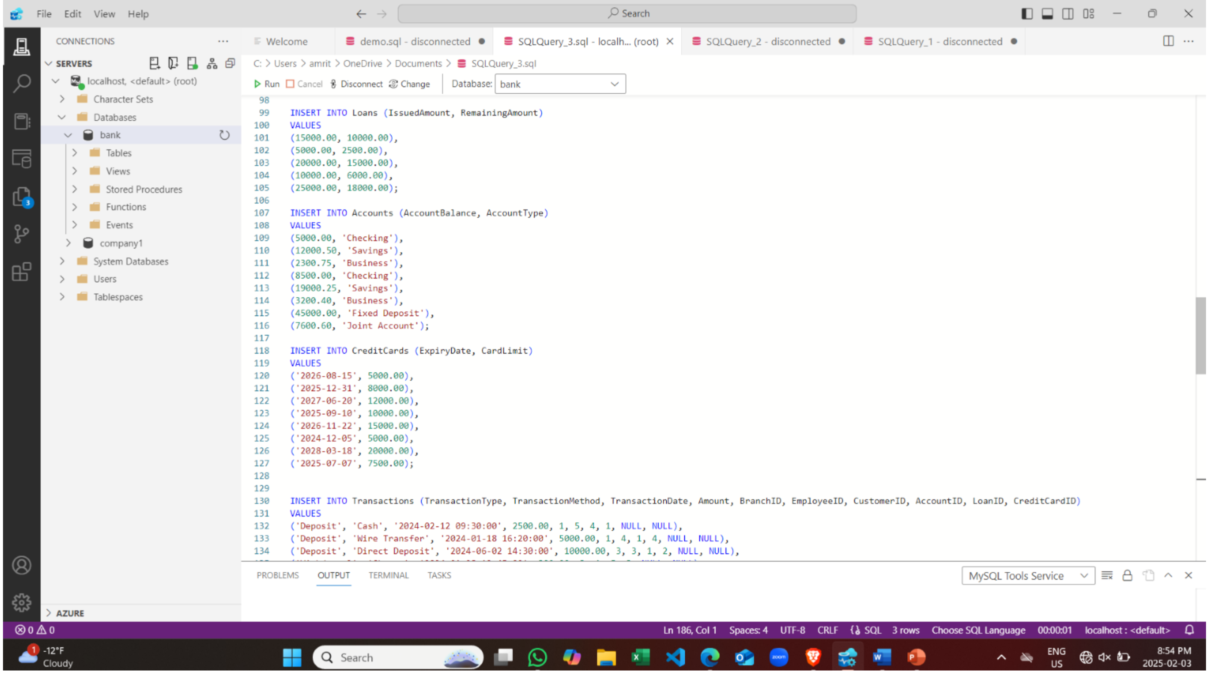
Credit Card	
CreditCardId	int
ExpiryDate	date
CardLimit	decimal

Designing database:

https://github.com/kanavv-tech/Data-science/upload/main

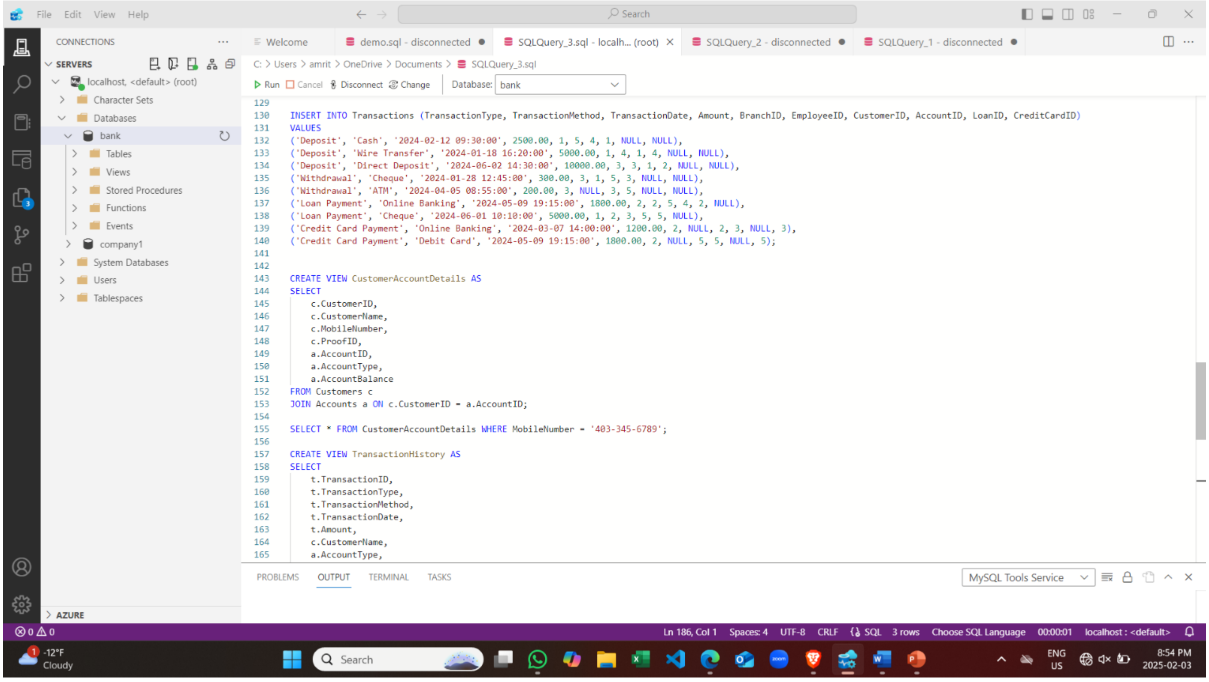


```sql
    FOREIGN KEY (LoanID) REFERENCES Loans(LoanID),
    FOREIGN KEY (CreditCardID) REFERENCES CreditCards(CreditCardID)
);

INSERT INTO Branches (BranchName, BranchAddress, Assets)
VALUES
('Main Campus Branch', '1301 16 Ave NW, Calgary, AB, T2M 0L4', 2000000.00),
('Downtown Branch', '200 6 Ave SW, Calgary, AB, T2P 0W5', 1500000.00),
('SAIT Residence Branch', '1390 10 St NW, Calgary, AB, T2N 1V8', 1200000.00);

INSERT INTO Employees (EmployeeName, JobTitle, EmailID, Address, HireDate, ResignDate, Salary)
VALUES
('Diana Green', 'Customer Service Representative', 'diana.green@saitbank.com', '101 River St, Calgary, AB', '2017-08-14', NULL, 48000.00),
('Edward Thomas', 'Assistant Manager', 'edward.thomas@saitbank.com', '202 Sunset Blvd, Calgary, AB', '2016-05-30', NULL, 80000.00),
('Fiona White', 'Financial Advisor', 'fiona.white@saitbank.com', '303 Skyline Rd, Calgary, AB', '2019-11-10', NULL, 70000.00),
('George Black', 'IT Specialist', 'george.black@saitbank.com', '404 Evergreen Ln, Calgary, AB', '2021-07-19', NULL, 62000.00),
('Hannah Clark', 'Teller', 'hannah.clark@saitbank.com', '505 Maple Ave, Calgary, AB', '2023-02-05', NULL, 45000.00),
('Alice Johnson', 'Branch Manager', 'alice.johnson@saitbank.com', '123 Main St, Calgary, AB', '2015-06-10', NULL, 95000.00),
('Bob Smith', 'Loan Officer', 'bob.smith@saitbank.com', '456 2nd Ave, Calgary, AB', '2018-09-22', NULL, 72000.00),
('Charlie Brown', 'Teller', 'charlie.brown@saitbank.com', '789 3rd St, Calgary, AB', '2020-01-15', NULL, 45000.00);


SELECT * from Transactions;

INSERT INTO Customers (CustomerName, DateOfBirth, MobileNumber, ProofID)
VALUES
('David Miller', '1990-04-15', '403-123-4567', 'DL-AB-123456'),
('Emma Wilson', '1985-07-23', '403-234-5678', 'PP-CA-987654'),
('Franklin Harris', '1995-11-30', '403-345-6789', 'DL-AB-789012'),
('Grace Adams', '1988-06-12', '403-567-8910', 'PP-CA-567890'),
('Isaac Bell', '1993-09-25', '403-678-9101', 'DL-AB-234567'),
('Julia Carter', '2000-02-14', '403-789-0123', 'PP-CA-678901'),
('Kevin Douglas', '1982-11-03', '403-890-1234', 'DL-AB-345678'),
('Laura Evans', '1997-05-20', '403-901-2345', 'PP-CA-789012');

INSERT INTO Loans (IssuedAmount, RemainingAmount)
VALUES
```



```sql

INSERT INTO Loans (IssuedAmount, RemainingAmount)
VALUES
(15000.00, 10000.00),
(5000.00, 2500.00),
(20000.00, 15000.00),
(10000.00, 6000.00),
(25000.00, 18000.00);

INSERT INTO Accounts (AccountBalance, AccountType)
VALUES
(5000.00, 'Checking'),
(12000.50, 'Savings'),
(2300.75, 'Business'),
(8500.00, 'Checking'),
(19000.25, 'Savings'),
(3200.40, 'Business'),
(45000.00, 'Fixed Deposit'),
(7600.60, 'Joint Account');

INSERT INTO CreditCards (ExpiryDate, CardLimit)
VALUES
('2026-08-15', 5000.00),
('2025-12-31', 8000.00),
('2027-06-20', 12000.00),
('2025-09-10', 10000.00),
('2026-11-22', 15000.00),
('2024-12-05', 5000.00),
('2028-03-18', 20000.00),
('2025-07-07', 7500.00);


INSERT INTO Transactions (TransactionType, TransactionMethod, TransactionDate, Amount, BranchID, EmployeeID, CustomerID, AccountID, LoanID, CreditCardID)
VALUES
('Deposit', 'Cash', '2024-02-12 09:30:00', 2500.00, 1, 5, 4, 1, NULL, NULL),
('Deposit', 'Wire Transfer', '2024-01-18 16:20:00', 5000.00, 1, 4, 1, 4, NULL, NULL),
('Deposit', 'Direct Deposit', '2024-06-02 14:30:00', 10000.00, 3, 3, 1, 2, NULL, NULL),
```

https://github.com/kanavv-tech/Data-science/upload/main



```sql

INSERT INTO Transactions (TransactionType, TransactionMethod, TransactionDate, Amount, BranchID, EmployeeID, CustomerID, AccountID, LoanID, CreditCardID)
VALUES
('Deposit', 'Cash', '2024-02-12 09:30:00', 2500.00, 1, 5, 4, 1, NULL, NULL),
('Deposit', 'Wire Transfer', '2024-01-18 16:20:00', 5000.00, 1, 4, 1, 4, NULL, NULL),
('Deposit', 'Direct Deposit', '2024-06-02 14:30:00', 10000.00, 3, 3, 1, 2, NULL, NULL),
('Withdrawal', 'Cheque', '2024-01-28 12:45:00', 300.00, 3, 1, 5, 3, NULL, NULL),
('Withdrawal', 'ATM', '2024-04-05 08:55:00', 200.00, 3, NULL, 3, 5, NULL, NULL),
('Loan Payment', 'Online Banking', '2024-05-09 19:15:00', 1800.00, 2, 2, 5, 4, 2, NULL),
('Loan Payment', 'Cheque', '2024-06-01 10:10:00', 5000.00, 1, 2, 3, 5, 5, NULL),
('Credit Card Payment', 'Online Banking', '2024-03-07 14:00:00', 1200.00, 2, NULL, 2, 3, NULL, 3),
('Credit Card Payment', 'Debit Card', '2024-05-09 19:15:00', 1800.00, 2, NULL, 5, 5, NULL, 5);


CREATE VIEW CustomerAccountDetails AS
SELECT
    c.CustomerID,
    c.CustomerName,
    c.MobileNumber,
    c.ProofID,
    a.AccountID,
    a.AccountType,
    a.AccountBalance
FROM Customers c
JOIN Accounts a ON c.CustomerID = a.AccountID;

SELECT * FROM CustomerAccountDetails WHERE MobileNumber = '403-345-6789';

CREATE VIEW TransactionHistory AS
SELECT
    t.TransactionID,
    t.TransactionType,
    t.TransactionMethod,
    t.TransactionDate,
    t.Amount,
    c.CustomerName,
    a.AccountType,
```

CONCLUSION

The Bank of Sait database system is a prime example of how a safe and well-organised database solution can increase client happiness, lower operating costs, and streamline banking operations.

Strict business rules, sophisticated database techniques, and a well-designed schema enable the system to fulfil its goals of centralising data administration, automating repetitive tasks, and improving data security.

https://github.com/kanavv-tech/Data-science/upload/main