Bank Management System: SQL Schema + Queries

F2023332037

Sajeela Noor

CREATE TABLE Branch (

BranchID INT PRIMARY KEY, Name VARCHAR(100),

Address VARCHAR(255), City VARCHAR(100), Phone VARCHAR(20)

);

CREATE TABLE Employee (

EmployeeID INT PRIMARY KEY, Name VARCHAR(100),

Email VARCHAR(100), Phone VARCHAR(20), Position VARCHAR(50),

BranchID INT NOT NULL, HireDate DATE,

FOREIGN KEY (BranchID) REFERENCES Branch(BranchID)

);

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY, Name VARCHAR(100),

Email VARCHAR(100), Phone VARCHAR(20), Address VARCHAR(255),

RegisteredBranchID INT NOT NULL,

FOREIGN KEY (RegisteredBranchID) REFERENCES Branch(BranchID)

);

CREATE TABLE AccountType (

AccountTypeID INT PRIMARY KEY, TypeName VARCHAR(50),

Description TEXT

);

CREATE TABLE Account (

AccountID INT PRIMARY KEY, CustomerID INT NOT NULL,

BranchID INT NOT NULL, AccountTypeID INT NOT NULL,

Balance DECIMAL(15,2), DateOpened DATE,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (BranchID) REFERENCES Branch(BranchID),

FOREIGN KEY (AccountTypeID) REFERENCES AccountType(AccountTypeID)

);

CREATE TABLE TransactionType (

TransactionTypeID INT PRIMARY KEY, TypeName VARCHAR(50)

);

CREATE TABLE Transaction (

TransactionID INT PRIMARY KEY, AccountID INT NOT NULL,

TransactionTypeID INT NOT NULL, Amount DECIMAL(15,2),

Date DATETIME, Description VARCHAR(255), PerformedBy INT,

FOREIGN KEY (AccountID) REFERENCES Account(AccountID),

FOREIGN KEY (TransactionTypeID) REFERENCES TransactionType(TransactionTypeID),

FOREIGN KEY (PerformedBy) REFERENCES Employee(EmployeeID)

);

CREATE TABLE LoanType (

LoanTypeID INT PRIMARY KEY, TypeName VARCHAR(50), Description TEXT

);

CREATE TABLE Loan (

LoanID INT PRIMARY KEY, CustomerID INT NOT NULL,

BranchID INT NOT NULL, LoanTypeID INT NOT NULL,

LoanAmount DECIMAL(15,2), InterestRate DECIMAL(5,2),

Status VARCHAR(50), IssueDate DATE, ApprovedBy INT,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (BranchID) REFERENCES Branch(BranchID),

FOREIGN KEY (LoanTypeID) REFERENCES LoanType(LoanTypeID),

FOREIGN KEY (ApprovedBy) REFERENCES Employee(EmployeeID)

);

CREATE TABLE Payment (

PaymentID INT PRIMARY KEY, LoanID INT NOT NULL,

Amount DECIMAL(15,2), PaymentDate DATE, CollectedBy INT,

FOREIGN KEY (LoanID) REFERENCES Loan(LoanID),

FOREIGN KEY (CollectedBy) REFERENCES Employee(EmployeeID)

);

# Advanced Queries

-- A. Customers and their account balances

SELECT C.Name, A.AccountID, AT.TypeName, A.Balance

FROM Customer C

JOIN Account A ON C.CustomerID = A.CustomerID

JOIN AccountType AT ON A.AccountTypeID = AT.AccountTypeID;

-- B. Total transactions per account by type

SELECT A.AccountID, TType.TypeName, COUNT(\*) AS Total

FROM Account A

JOIN Transaction T ON A.AccountID = T.AccountID

JOIN TransactionType TType ON T.TransactionTypeID = TType.TransactionTypeID

GROUP BY A.AccountID, TType.TypeName;

-- C. Loans approved by each employee

SELECT E.Name, COUNT(\*) AS LoansApproved

FROM Loan L JOIN Employee E ON L.ApprovedBy = E.EmployeeID

GROUP BY E.EmployeeID;

-- D. Loan repayment summary

SELECT C.Name, L.LoanID, SUM(P.Amount) AS TotalPaid

FROM Customer C

JOIN Loan L ON C.CustomerID = L.CustomerID

JOIN Payment P ON L.LoanID = P.LoanID

GROUP BY C.CustomerID, L.LoanID;

-- E. Branch-wise total account balances

SELECT B.Name, SUM(A.Balance) AS TotalBalance

FROM Branch B JOIN Account A ON B.BranchID = A.BranchID

GROUP BY B.BranchID;

-- F. Customers without loans

SELECT C.Name

FROM Customer C LEFT JOIN Loan L ON C.CustomerID = L.CustomerID

WHERE L.LoanID IS NULL;

-- G. Total deposits & withdrawals per branch

SELECT B.Name, TType.TypeName, SUM(T.Amount) AS Total

FROM Branch B

JOIN Account A ON B.BranchID = A.BranchID

JOIN Transaction T ON A.AccountID = T.AccountID

JOIN TransactionType TType ON T.TransactionTypeID = TType.TransactionTypeID

GROUP BY B.BranchID, TType.TypeName;

-- Insert into Branch

INSERT INTO Branch VALUES (1, 'Main Branch', '123 Main St', 'Lahore', '0421234567');

-- Insert into Employee

INSERT INTO Employee VALUES (101, 'Ali Khan', 'ali@bank.com', '03211234567', 'Manager', 1, '2020-05-01');

-- Insert into Customer

INSERT INTO Customer VALUES (201, 'Sara Ahmed', 'sara@gmail.com', '03111234567', '45 Model Town', 1);

-- Insert into AccountType

INSERT INTO AccountType VALUES (1, 'Savings', 'Standard Savings Account');

-- Insert into Account

INSERT INTO Account VALUES (301, 201, 1, 1, 50000.00, '2022-01-15');

-- Insert Sample

INSERT INTO Branch VALUES (1, 'Main', '123 Main St', 'Lahore', '0421234567');

INSERT INTO Employee VALUES (101, 'Ali', 'ali@bank.com', '0321...', 'Manager', 1, '2020-05-01');

-- Update

UPDATE Account SET Balance = Balance + 5000 WHERE AccountID = 301;

-- Delete

DELETE FROM Transaction WHERE TransactionID = 1001;

-- Select

SELECT \* FROM Customer;

GRANT SELECT, INSERT ON BankDB.\* TO 'bankuser'@'localhost';

REVOKE DELETE ON BankDB.\* FROM 'bankuser'@'localhost';

# Schema

**A screenshot of a computer

AI-generated content may be incorrect.**

Figure Banking System Schema

# ER - Diagram

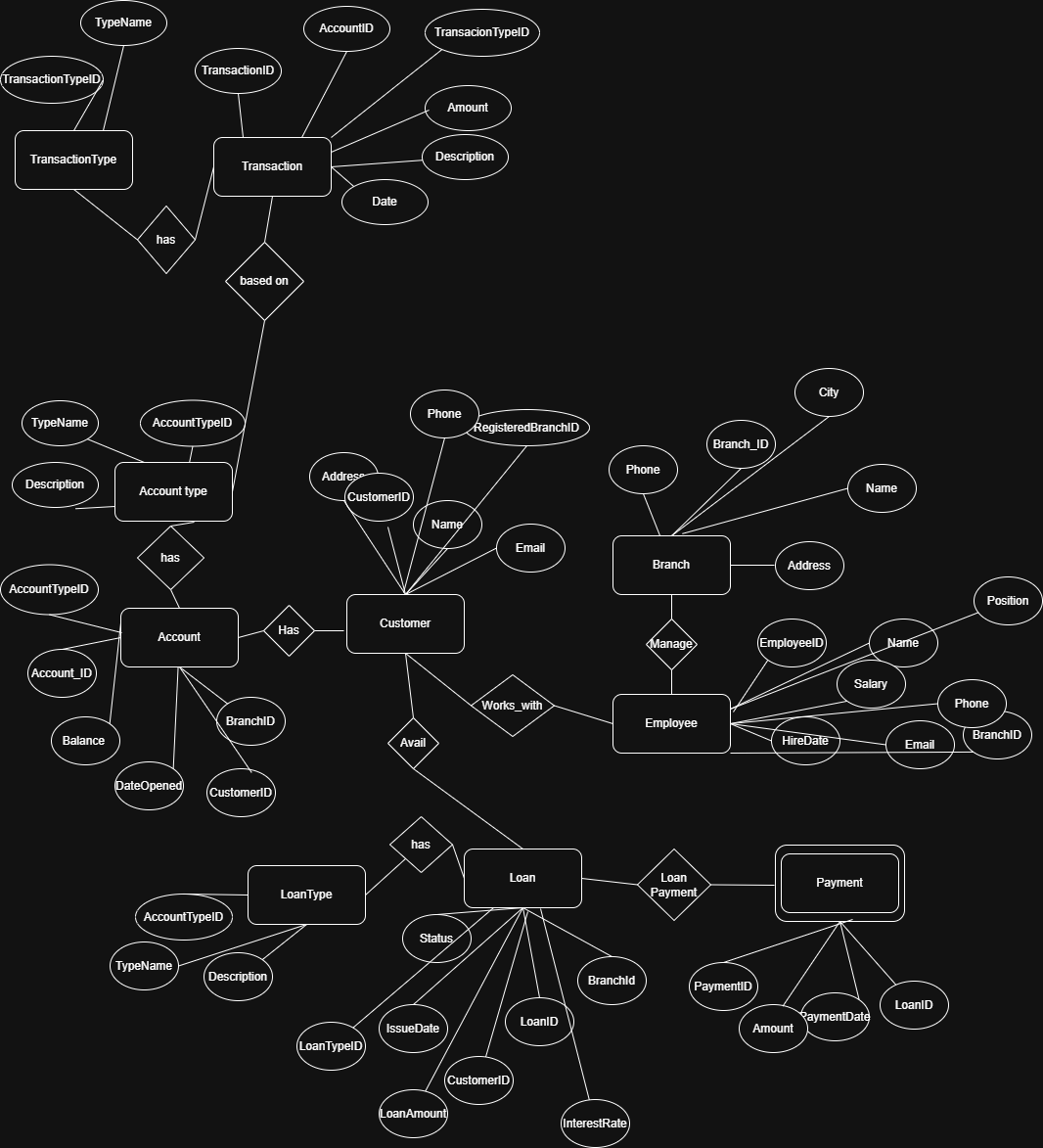
****

Figure Banking System ER - Diagram