1. **Database Design:**
   * Tables: Customers, Accounts, Transactions, Loans.
   * Relationships:
     + A customer can have multiple accounts.
     + An account can have multiple transactions.
     + A customer can have multiple loans.
2. **Table Definitions:**
   * Create tables with appropriate columns and data types.
   * Define primary keys and foreign keys.
3. **Sample Data Insertion:**
   * Insert some sample data into the tables.
4. **Sample Queries:**
   * Queries to track account balances, analyze transaction history, and manage loan records.

**1. Database Design**

**Tables and Relationships:**

* **Customers**: Stores customer information.
* **Accounts**: Stores account information.
* **Transactions**: Stores transaction details.
* **Loans**: Stores loan details.

Code:

CREATE DATABASE BankingSystem;

USE BankingSystem;

-- Create Customers Table

CREATE TABLE Customers (

CustomerID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DateOfBirth DATE,

Address VARCHAR(100)

);

-- Create Accounts Table

CREATE TABLE Accounts (

AccountID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT,

AccountType VARCHAR(20),

Balance DECIMAL(15, 2),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

-- Create Transactions Table

CREATE TABLE Transactions (

TransactionID INT AUTO\_INCREMENT PRIMARY KEY,

AccountID INT,

TransactionDate DATE,

Amount DECIMAL(15, 2),

TransactionType VARCHAR(20),

FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)

);

-- Create Loans Table

CREATE TABLE Loans (

LoanID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT,

LoanAmount DECIMAL(15, 2),

InterestRate DECIMAL(5, 2),

StartDate DATE,

EndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

-- Insert into Customers

INSERT INTO Customers (FirstName, LastName, DateOfBirth, Address)

VALUES

('John', 'Doe', '1980-01-01', '123 Main St'),

('Jane', 'Smith', '1985-05-15', '456 Oak St');

-- Insert into Accounts

INSERT INTO Accounts (CustomerID, AccountType, Balance)

VALUES

(1, 'Savings', 1000.00),

(1, 'Checking', 1500.00),

(2, 'Savings', 2000.00);

-- Insert into Transactions

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

VALUES

(1, '2024-01-01', 100.00, 'Deposit'),

(1, '2024-01-02', 50.00, 'Withdrawal'),

(2, '2024-01-03', 200.00, 'Deposit'),

(3, '2024-01-04', 500.00, 'Deposit');

-- Insert into Loans

INSERT INTO Loans (CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES

(1, 5000.00, 5.00, '2024-01-01', '2029-01-01'),

(2, 10000.00, 4.50, '2024-02-01', '2030-02-01');

SELECT

c.FirstName,

c.LastName,

a.AccountType,

a.Balance

FROM

Customers c

JOIN

Accounts a ON c.CustomerID = a.CustomerID;

SELECT

t.TransactionID,

t.TransactionDate,

t.Amount,

t.TransactionType,

a.AccountID,

c.FirstName,

c.LastName

FROM

Transactions t

JOIN

Accounts a ON t.AccountID = a.AccountID

JOIN

Customers c ON a.CustomerID = c.CustomerID

ORDER BY

t.TransactionDate DESC;

**Schema Diagram**

1. **Customers Table**: Contains customer information.
   * CustomerID (Primary Key)
   * FirstName
   * LastName
   * DateOfBirth
   * Address
2. **Accounts Table**: Contains account information.
   * AccountID (Primary Key)
   * CustomerID (Foreign Key referencing Customers.CustomerID)
   * AccountType
   * Balance
3. **Transactions Table**: Contains transaction details.
   * TransactionID (Primary Key)
   * AccountID (Foreign Key referencing Accounts.AccountID)
   * TransactionDate
   * Amount
   * TransactionType
4. **Loans Table**: Contains loan details.
   * LoanID (Primary Key)
   * CustomerID (Foreign Key referencing Customers.CustomerID)
   * LoanAmount
   * InterestRate
   * StartDate
   * EndDate