-- Creating tables with necessary constraints

CREATE TABLE Account (

Acc\_no INT PRIMARY KEY,

branch\_name VARCHAR(50),

balance DECIMAL(10, 2),

FOREIGN KEY (branch\_name) REFERENCES Branch(branch\_name)

);

CREATE TABLE Branch (

branch\_name VARCHAR(50) PRIMARY KEY,

branch\_city VARCHAR(50),

assets DECIMAL(10, 2)

);

CREATE TABLE Customer (

cust\_name VARCHAR(50) PRIMARY KEY,

cust\_street VARCHAR(50),

cust\_city VARCHAR(50)

);

CREATE TABLE Depositor (

cust\_name VARCHAR(50),

acc\_no INT,

PRIMARY KEY (cust\_name, acc\_no),

FOREIGN KEY (cust\_name) REFERENCES Customer(cust\_name),

FOREIGN KEY (acc\_no) REFERENCES Account(Acc\_no)

);

CREATE TABLE Loan (

loan\_no INT PRIMARY KEY,

branch\_name VARCHAR(50),

amount DECIMAL(10, 2),

FOREIGN KEY (branch\_name) REFERENCES Branch(branch\_name)

);

CREATE TABLE Borrower (

cust\_name VARCHAR(50),

loan\_no INT,

PRIMARY KEY (cust\_name, loan\_no),

FOREIGN KEY (cust\_name) REFERENCES Customer(cust\_name),

FOREIGN KEY (loan\_no) REFERENCES Loan(loan\_no)

);

-- 1. Find all customers who have both account and loan.

SELECT DISTINCT c.cust\_name

FROM Customer c

JOIN Depositor d ON c.cust\_name = d.cust\_name

JOIN Borrower b ON c.cust\_name = b.cust\_name;

-- 2. Find all customers who have an account or loan or both.

SELECT DISTINCT cust\_name FROM Depositor

UNION

SELECT DISTINCT cust\_name FROM Borrower;

-- 3. Find all customers who have an account but no loan.

SELECT DISTINCT c.cust\_name

FROM Customer c

JOIN Depositor d ON c.cust\_name = d.cust\_name

LEFT JOIN Borrower b ON c.cust\_name = b.cust\_name

WHERE b.loan\_no IS NULL;

-- 4. Find the average account balance at ‘Wadia College’ branch.

SELECT AVG(balance) AS avg\_balance

FROM Account

WHERE branch\_name = 'Wadia College';

-- 5. Find the number of depositors at each branch.

SELECT branch\_name, COUNT(DISTINCT acc\_no) AS num\_depositors

FROM Depositor d

JOIN Account a ON d.acc\_no = a.Acc\_no

GROUP BY branch\_name;