-- 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 branches where average account balance > 15000.

SELECT branch\_name

FROM Account

GROUP BY branch\_name

HAVING AVG(balance) > 15000;

-- 2. Find number of tuples in the Customer relation.

SELECT COUNT(\*) AS num\_tuples FROM Customer;

-- 3. Calculate total loan amount given by the bank.

SELECT SUM(amount) AS total\_loan\_amount FROM Loan;

-- 4. Delete all loans with loan amount between 1300 and 1500.

DELETE FROM Loan WHERE amount BETWEEN 1300 AND 1500;

-- 5. Find the average account balance at each branch.

SELECT branch\_name, AVG(balance) AS avg\_balance

FROM Account

GROUP BY branch\_name;

-- 6. Find names of Customers and cities where names start with 'P'.

SELECT cust\_name, cust\_city

FROM Customer

WHERE cust\_name LIKE 'P%';