**1. Consider the following bank database relations, where the primary keys are underlined.**

Branch (branch-name, branch-city, assets)

Customer (customer-name, customer-street, customer-city)

Loan (loan-number, branch-name, amount)

Borrower (customer-name, loan-number)

Account (account-number, branch-name, balance)

Depositor (customer-name, account number)

**Write down the SQL expressions for the following queries**.

**i.** Find all customers who have an account but no loan in the bank.

**ii.** Delete all loan amount between 10000/- and 25000/-

**iii.** Find the names of all customers who have a loan at Perryridge branch.

**iv.** Delete all loans with amounts in the range 0 to 500.

**DDL:**

CREATE DATABASE IF NOT EXISTS q1\_bank\_database;

USE q1\_bank\_database;

CREATE TABLE Branch (

branch\_name VARCHAR(50),

branch\_city VARCHAR(50),

assets DECIMAL(12, 2),

PRIMARY KEY (branch\_name)

);

CREATE TABLE Customer (

customer\_name VARCHAR(50),

customer\_street VARCHAR(100),

customer\_city VARCHAR(50),

PRIMARY KEY (customer\_name)

);

CREATE TABLE Loan (

loan\_number INT,

branch\_name VARCHAR(50),

amount DECIMAL(12, 2),

PRIMARY KEY (loan\_number),

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

);

CREATE TABLE Borrower (

customer\_name VARCHAR(50),

loan\_number INT,

PRIMARY KEY (customer\_name, loan\_number),

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

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

);

CREATE TABLE Account (

account\_number INT,

branch\_name VARCHAR(50),

balance DECIMAL(12, 2),

PRIMARY KEY (account\_number),

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

);

CREATE TABLE Depositor (

customer\_name VARCHAR(50),

account\_number INT,

PRIMARY KEY (customer\_name, account\_number),

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

FOREIGN KEY (account\_number) REFERENCES Account(account\_number)

);

**DML:**

INSERT INTO Branch (branch\_name, branch\_city, assets)

VALUES ('Perryridge', 'Brooklyn', 150000),

('Brighton', 'New York', 200000),

('Downtown', 'Brooklyn', 100000);

INSERT INTO Customer (customer\_name, customer\_street, customer\_city)

VALUES ('Alice', '123 Main St', 'Brooklyn'),

('Bob', '456 Elm St', 'New York'),

('Charlie', '789 Oak St', 'Brooklyn'),

('David', '101 Pine St', 'Brooklyn');

INSERT INTO Loan (loan\_number, branch\_name, amount)

VALUES (1, 'Perryridge', 20000),

(2, 'Brighton', 30000),

(3, 'Downtown', 15000),

(4, 'Perryridge', 1000);

INSERT INTO Borrower (customer\_name, loan\_number)

VALUES ('Alice', 1),

('Bob', 2),

('Charlie', 3),

('David', 4);

INSERT INTO Account (account\_number, branch\_name, balance)

VALUES (101, 'Perryridge', 5000),

(102, 'Brighton', 10000),

(103, 'Downtown', 7500),

(104, 'Perryridge', 200);

INSERT INTO Depositor (customer\_name, account\_number)

VALUES ('Alice', 101),

('Bob', 102),

('Charlie', 103),

('David', 104);

**Write down the SQL expressions for the following queries**.

1. Find all customers who have an account but no loan in the bank.

SELECT DISTINCT D.customer\_name

FROM Depositor D

LEFT JOIN Borrower B ON D.customer\_name = B.customer\_name

WHERE B.customer\_name IS NULL;

1. Delete all loan amount between 10000/- and 25000/-

DELETE FROM Loan

WHERE amount BETWEEN 10000 AND 25000;

1. Find the names of all customers who have a loan at Perryridge branch.

SELECT DISTINCT B.customer\_name

FROM Borrower B

JOIN Loan L ON B.loan\_number = L.loan\_number

WHERE L.branch\_name = 'Perryridge';

1. Delete all loans with amounts in the range 0 to 500.

DELETE FROM Loan

WHERE amount BETWEEN 0 AND 500;

**2. Consider the employee database consisting of the following relations, where the primary keys are underlined.**

Employee (employee-id, employee-name, street, city)

Works (employee-id, company-name, salary)

Company (company-name, city)

Manager (employee-id, manager-name)

**Write down the SQL expressions for the following queries:**

**i**. Find the company that has the most employees.

**ii**. Find the average salaries at each company.

**iii.** Find all employees who live in Barisal city, but their company is not in Barisal.

1. Find the names of all employees who work for First Bank Corporation.

**DDL:**

CREATE DATABASE IF NOT EXISTS q2\_company\_database;

USE q2\_company\_database;

CREATE TABLE Employee (

employee\_id INT PRIMARY KEY,

employee\_name VARCHAR(100),

street VARCHAR(100),

city VARCHAR(50)

);

CREATE TABLE Works (

employee\_id INT,

company\_name VARCHAR(100),

salary DECIMAL(10, 2),

FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id)

);

CREATE TABLE Company (

company\_name VARCHAR(100) PRIMARY KEY,

city VARCHAR(50)

);

CREATE TABLE Manager (

employee\_id INT PRIMARY KEY,

manager\_name VARCHAR(100)

);

**DML:**

INSERT INTO Employee (employee\_id, employee\_name, street, city)

VALUES (1, 'John Doe', '123 Main St', 'New York'),

(2, 'Jane Smith', '456 Elm St', 'Los Angeles'),

(3, 'Alice Johnson', '789 Oak St', 'Chicago'),

(4, 'Bob Brown', '101 Pine St', 'Barisal');

INSERT INTO Works (employee\_id, company\_name, salary)

VALUES (1, 'ABC Corporation', 60000.00),

(2, 'XYZ Company', 70000.00),

(3, 'ABC Corporation', 80000.00),

(4, 'First Bank Corporation', 65000.00);

INSERT INTO Company (company\_name, city)

VALUES ('ABC Corporation', 'New York'),

('XYZ Company', 'Los Angeles'),

('First Bank Corporation', 'Chicago');

INSERT INTO Manager (employee\_id, manager\_name)

VALUES (1, 'Michael Johnson'),

(2, 'Emily Davis'),

(3, 'Michael Johnson'),

(4, 'Alice Johnson');

**Write down the SQL expressions for the following queries:**

1. Find the company that has the most employees.

SELECT company\_name

FROM Works

GROUP BY company\_name

ORDER BY COUNT(employee\_id) DESC

LIMIT 1;

1. Find the average salaries at each company.

SELECT company\_name, AVG(salary) AS average\_salary

FROM Works

GROUP BY company\_name;

1. Find all employees who live in Barisal city, but their company is not in Barisal.

SELECT e.employee\_name

FROM Employee e

JOIN Works w ON e.employee\_id = w.employee\_id

JOIN Company c ON w.company\_name = c.company\_name

WHERE e.city = 'Barisal' AND c.city != 'Barisal';

1. Find the names of all employees who work for First Bank Corporation.

SELECT employee\_name

FROM Employee

JOIN Works ON Employee.employee\_id = Works.employee\_id

WHERE company\_name = 'First Bank Corporation';

**3. Consider the banking database consisting of the banking database consisting of the following tables, where the primary keys are underlined**.

Branch (branch-name, branch-city, assets)

Customer (customer-name, customer-street, customer-city)

Loan-account (loan-number, branch-name, amount)

Borrower (customer-name, loan-number)

Saving-account (account number, branch-name, balance)

Depositor (customer-name, account number)

**Write down the SQL expressions for the following queries:**

**i.**  Find all customers of the bank who have both loan and a saving account.

**ii.**  Find all average account balance at each branch.

**iii.** Deduct 3% service charge from saving account balance that have both loan and a saving account otherwise deduct 5% service charge from saving account balance.

**DDL:**

CREATE DATABASE IF NOT EXISTS q3\_bank\_database;

USE q3\_bank\_database;

CREATE TABLE Branch (

branch\_name VARCHAR(50) PRIMARY KEY,

branch\_city VARCHAR(50),

assets DECIMAL(12, 2)

);

CREATE TABLE Customer (

customer\_name VARCHAR(50) PRIMARY KEY,

customer\_street VARCHAR(100),

customer\_city VARCHAR(50)

);

CREATE TABLE Loan\_account (

loan\_number INT PRIMARY KEY,

branch\_name VARCHAR(50),

amount DECIMAL(12, 2),

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

);

CREATE TABLE Borrower (

customer\_name VARCHAR(50),

loan\_number INT,

PRIMARY KEY (customer\_name, loan\_number),

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

FOREIGN KEY (loan\_number) REFERENCES Loan\_account(loan\_number)

);

CREATE TABLE Saving\_account (

account\_number INT PRIMARY KEY,

branch\_name VARCHAR(50),

balance DECIMAL(12, 2),

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

);

CREATE TABLE Depositor (

customer\_name VARCHAR(50),

account\_number INT,

PRIMARY KEY (customer\_name, account\_number),

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

FOREIGN KEY (account\_number) REFERENCES Saving\_account(account\_number)

);

**DML:**

INSERT INTO Branch (branch\_name, branch\_city, assets)

VALUES ('Main Branch', 'New York', 1000000.00),

('Downtown Branch', 'New York', 750000.00),

('Uptown Branch', 'New York', 500000.00),

('Westside Branch', 'Los Angeles', 900000.00),

('Eastside Branch', 'Los Angeles', 600000.00);

INSERT INTO Customer (customer\_name, customer\_street, customer\_city)

VALUES ('John Smith', '123 Main St', 'New York'),

('Jane Doe', '456 Elm St', 'Los Angeles'),

('Alice Johnson', '789 Oak St', 'New York'),

('Bob Williams', '101 Pine St', 'Los Angeles');

INSERT INTO Loan\_account (loan\_number, branch\_name, amount)

VALUES (1, 'Main Branch', 50000.00),

(2, 'Downtown Branch', 30000.00),

(3, 'Uptown Branch', 20000.00),

(4, 'Westside Branch', 40000.00),

(5, 'Eastside Branch', 25000.00);

INSERT INTO Borrower (customer\_name, loan\_number)

VALUES ('John Smith', 1),

('Jane Doe', 2),

('Alice Johnson', 3),

('Bob Williams', 4),

('Alice Johnson', 5);

INSERT INTO Saving\_account (account\_number, branch\_name, balance)

VALUES (101, 'Main Branch', 10000.00),

(102, 'Downtown Branch', 15000.00),

(103, 'Uptown Branch', 20000.00),

(104, 'Westside Branch', 30000.00),

(105, 'Eastside Branch', 5000.00);

INSERT INTO Depositor (customer\_name, account\_number)

VALUES ('John Smith', 101),

('Jane Doe', 102),

('Alice Johnson', 103),

('Bob Williams', 104),

('Alice Johnson', 105);

**Write down the SQL expressions for the following queries:**

**i.**  Find all customers of the bank who have both loan and a saving account.

SELECT DISTINCT c.customer\_name

FROM Customer c

INNER JOIN Borrower b ON c.customer\_name = b.customer\_name

INNER JOIN Loan\_account la ON b.loan\_number = la.loan\_number

INNER JOIN Depositor d ON c.customer\_name = d.customer\_name

INNER JOIN Saving\_account sa ON d.account\_number = sa.account\_number;

**ii.**  Find all average account balance at each branch.

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

FROM Saving\_account

GROUP BY branch\_name;

**iii.** Deduct 3% service charge from saving account balance that have both loan and a saving account otherwise deduct 5% service charge from saving account balance.

UPDATE Saving\_account

SET balance = CASE

WHEN EXISTS (

SELECT 1

FROM Depositor d

INNER JOIN Borrower b ON d.customer\_name = b.customer\_name

WHERE d.account\_number = Saving\_account.account\_number

) THEN balance \* 0.97

ELSE balance \* 0.95

END

WHERE account\_number IN (

SELECT account\_number

FROM Depositor

);

**4. Consider the employee database consisting of the following tables, where the primary keys are underlined.**

Employee (employee-name, street, city)

Works (employee-name, company-name, salary)

Company (company-name, city)

Manages (employee-name, manages-name)

**Write down the SQL expressions for the following queries:**

**i.** Find the names, cities and salaries of all employees who work for IFIC Bank Ltd.

**ii.** Find the total salaries of each company.

**iii.** Give all employees of First Bank Corporation a 20 percent salary raise.

**iv.** Find the names of all employees in this database who do not work for First Bank Corporation.

**DDL:**

CREATE DATABASE IF NOT EXISTS q4\_company\_database;

USE q4\_company\_database;

CREATE TABLE Employee (

employee\_name VARCHAR(50),

street VARCHAR(100),

city VARCHAR(50),

PRIMARY KEY (employee\_name)

);

CREATE TABLE Company (

company\_name VARCHAR(100),

city VARCHAR(50),

PRIMARY KEY (company\_name)

);

CREATE TABLE Works (

employee\_name VARCHAR(50),

company\_name VARCHAR(100),

salary DECIMAL(10,2),

PRIMARY KEY (employee\_name, company\_name),

FOREIGN KEY (employee\_name) REFERENCES Employee(employee\_name),

FOREIGN KEY (company\_name) REFERENCES Company(company\_name)

);

CREATE TABLE Manages (

employee\_name VARCHAR(50),

manages\_name VARCHAR(50),

PRIMARY KEY (employee\_name),

FOREIGN KEY (employee\_name) REFERENCES Employee(employee\_name),

FOREIGN KEY (manages\_name) REFERENCES Employee(employee\_name)

);

**DML:**

INSERT INTO Employee (employee\_name, street, city)

VALUES ('John Doe', '123 Main St', 'New York'),

('Alice Smith', '456 Elm St', 'Los Angeles'),

('Bob Johnson', '789 Oak St', 'Chicago');

INSERT INTO Company (company\_name, city)

VALUES ('IFIC Bank Ltd', 'New York'),

('First Bank Corporation', 'Los Angeles'),

('Second Company', 'Chicago');

INSERT INTO Works (employee\_name, company\_name, salary)

VALUES ('John Doe', 'IFIC Bank Ltd', 50000),

('Alice Smith', 'First Bank Corporation', 60000),

('Bob Johnson', 'Second Company', 55000);

INSERT INTO Manages (employee\_name, manages\_name)

VALUES ('Alice Smith', 'Bob Johnson');

**Write down the SQL expressions for the following queries:**

**i.** Find the names, cities and salaries of all employees who work for IFIC Bank Ltd.

SELECT E.employee\_name, E.city, W.salary

FROM Employee E

JOIN Works W ON E.employee\_name = W.employee\_name

WHERE W.company\_name = 'IFIC Bank Ltd';

**ii.** Find the total salaries of each company.

SELECT W.company\_name, SUM(W.salary) AS total\_salary

FROM Works W

GROUP BY W.company\_name;

**iii.** Give all employees of First Bank Corporation a 20 percent salary raise.

UPDATE Works

SET salary = salary \* 1.20

WHERE company\_name = 'First Bank Corporation';

**iv.** Find the names of all employees in this database who do not work for First Bank Corporation.

SELECT employee\_name

FROM Employee

WHERE employee\_name NOT IN (

SELECT employee\_name

FROM Works

WHERE company\_name = 'First Bank Corporation'

);

**5. Consider the following schemas for "car\_insurance" database relations, where the primary keys are underlined.**

Person (driver-id, name, address)

Car (license, model, year)

Accident (report-number, date, location)

Owns (driver-id, license)

Participate (driver-id, car, report-number, damage amount)

**Write down the SQL expressions for the following queries:**

**i.** Add a new accident to the database (assume any values for required attributes).

**ii.** Delete the Toyota belonging to "Simanto".

**iii.** Find the total number of people who owned cars that were involved in accidents in 2020.

**iv.** Update the damage amount for the car with license number "DHAKA 4000" in the accident with report number "AR 2197" to 30,000/-

**DDL:**

CREATE DATABASE IF NOT EXISTS q5\_car\_insurance\_database;

USE q5\_car\_insurance\_database;

CREATE TABLE Person (

driver\_id INT PRIMARY KEY,

name VARCHAR(100),

address VARCHAR(200)

);

CREATE TABLE Car (

license VARCHAR(20) PRIMARY KEY,

model VARCHAR(50),

year INT

);

CREATE TABLE Accident (

report\_number INT PRIMARY KEY,

date DATE,

location VARCHAR(100)

);

CREATE TABLE Owns (

driver\_id INT,

license VARCHAR(20),

PRIMARY KEY (driver\_id, license),

FOREIGN KEY (driver\_id) REFERENCES Person(driver\_id),

FOREIGN KEY (license) REFERENCES Car(license)

);

CREATE TABLE Participate (

driver\_id INT,

car VARCHAR(20),

report\_number INT,

damage\_amount DECIMAL(10, 2),

PRIMARY KEY (driver\_id, car, report\_number),

FOREIGN KEY (driver\_id) REFERENCES Person(driver\_id),

FOREIGN KEY (car) REFERENCES Car(license),

FOREIGN KEY (report\_number) REFERENCES Accident(report\_number)

);

**DML:**

INSERT INTO Person (driver\_id, name, address)

VALUES (1, 'Simanto', '123 Main St'),

(2, 'John Doe', '456 Elm St');

INSERT INTO Car (license, model, year)

VALUES ('DHAKA 1000', 'Toyota', 2018),

('DHAKA 2000', 'Honda', 2019);

INSERT INTO Accident (report\_number, date, location)

VALUES (100, '2020-05-20', 'Downtown'),

(101, '2021-07-15', 'Uptown');

INSERT INTO Owns (driver\_id, license)

VALUES (1, 'DHAKA 1000'),

(2, 'DHAKA 2000');

INSERT INTO Participate (driver\_id, car, report\_number, damage\_amount)

VALUES (1, 'DHAKA 1000', 100, 5000.00),

(2, 'DHAKA 2000', 101, 3000.00);

**Write down the SQL expressions for the following queries:**

**i.** Add a new accident to the database (assume any values for required attributes).

INSERT INTO Accident (report\_number, date, location)

VALUES (102, '2024-05-25', 'City Center');

**ii.** Delete the Toyota belonging to "Simanto".

DELETE FROM Car

WHERE license = (

SELECT license

FROM Owns

WHERE driver\_id = (

SELECT driver\_id

FROM Person

WHERE name = 'Simanto'

) AND model = 'Toyota'

);

**iii.** Find the total number of people who owned cars that were involved in accidents in 2020.

SELECT COUNT(DISTINCT P.driver\_id) AS total\_owners

FROM Participate P

JOIN Accident A ON P.report\_number = A.report\_number

WHERE YEAR(A.date) = 2020;

**iv.** Update the damage amount for the car with license number "DHAKA 4000" in the accident with report number "AR 2197" to 30,000/-.

UPDATE Participate

SET damage\_amount = 30000.00

WHERE car = 'DHAKA 4000'

AND report\_number = 'AR 2197';