Set 1

Q1. Create an Employee Attendance table and use UPDATE queries to modify attendance:

```
CREATE TABLE EmployeeAttendance (
    EmpID INT PRIMARY KEY,
    EmpName VARCHAR(50),
    DaysPresent INT
);

INSERT INTO EmployeeAttendance VALUES (1, 'Amit', 18), (2, 'Riya', 20), (3, 'Suresh', 15);

UPDATE EmployeeAttendance SET DaysPresent = DaysPresent + 1 WHERE EmpID = 1;

SELECT * FROM EmployeeAttendance;
```

Q2. Create a Product table with UNIQUE and CHECK constraints:

```
CREATE TABLE Product (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(50) UNIQUE,
    Price DECIMAL(10,2) CHECK (Price > 0)
);

INSERT INTO Product VALUES (1, 'Pen', 10.5), (2, 'Notebook', 25.0);
```

Q3. Use RIGHT JOIN on Customers and Orders:

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Customers
RIGHT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

Set 3

Q1. Create a Books table and use SELECT queries:

```
CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    Title VARCHAR(100),
    Author VARCHAR(50),
    Price DECIMAL(8,2)
);

INSERT INTO Books VALUES (1, 'Wings of Fire', 'A.P.J. Abdul Kalam', 300), (2, 'The Alchemist', 'Paulo Coelho', 250);

SELECT * FROM Books WHERE Price > 250;
SELECT * FROM Books ORDER BY Price DESC;
SELECT * FROM Books LIMIT 1;
```

Q2. Employee Attendance UPDATE and SELECT:

```
UPDATE EmployeeAttendance SET DaysPresent = DaysPresent + 1 WHERE EmpID = 1;

SELECT * FROM EmployeeAttendance;
```

Q3. SUM and COUNT:

```
SELECT SUM(Price * Quantity) AS TotalRevenue, COUNT(*) AS TotalProducts
FROM Products;
```

Set 4

Q1. Design tables and perform INSERT, UPDATE:

```
CREATE TABLE Department (
    DeptID INT PRIMARY KEY,
    DeptName VARCHAR(50)
);

CREATE TABLE Employee (
    EmpID INT PRIMARY KEY,
    EmpName VARCHAR(50),
    DeptID INT,
    FOREIGN KEY (DeptID) REFERENCES Department(DeptID)
);

INSERT INTO Department VALUES (1, 'HR'), (2, 'IT');
INSERT INTO Employee VALUES (101, 'Sneha', 1), (102, 'Rahul', 2);

UPDATE Employee SET DeptID = 1 WHERE EmpID = 102;
```

Q2. Use multiple data types:

```
CREATE TABLE EmployeeDetails (
    EmpID INT,
    EmpName VARCHAR(50),
    JoiningDate DATE,
    Salary INT
);
```

Q3. Insert, delete and display:

```
INSERT INTO EmployeeDetails VALUES (1, 'Amit', '2023-01-01', 25000);
DELETE FROM EmployeeDetails WHERE EmpID = 1;
SELECT * FROM EmployeeDetails;
```

Set 8

Q1. CHECK constraint on Age:

```
CREATE TABLE Employee (
    EmpID INT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT CHECK (Age >= 18)
);

INSERT INTO Employee VALUES (1, 'Aarti', 25), (2, 'Manoj', 30);
```

Q2. NOT NULL and DEFAULT:

```
CREATE TABLE Staff (
    StaffID INT PRIMARY KEY,
```

```
    Name VARCHAR(50) NOT NULL,
    City VARCHAR(50) DEFAULT 'Mumbai'
);

INSERT INTO Staff (StaffID, Name) VALUES (1, 'Karan');
```

Q3. SELECT top 3 highest salaries:

```
SELECT * FROM Employee
ORDER BY Salary DESC
LIMIT 3;
```