**Question 1: Create a Student Table**

Create a table Students where:

* StudentID is the primary key.
* Name cannot be null.
* Age must be between 18 and 30.
* City should default to 'Delhi' if not provided.

CREATE TABLE Students (

StudentID INT PRIMARY KEY,

Name VARCHAR(50) NOT NULL,

Age INT CHECK(Age BETWEEN 18 AND 30),

City VARCHAR(30) DEFAULT 'Delhi'

);

**Question 2: Insert Data with Default**

Insert a student record without specifying the city, and check that it takes the default value.

INSERT INTO Students (StudentID, Name, Age)

VALUES (1, 'Ravi Kumar', 22);

-- To verify

SELECT \* FROM Students;

**Question 3: Create Employee Table with Salary Validation**

Create a table Employees where:

* EmpID is the primary key.
* EmpName cannot be null.
* Salary must always be greater than 5000.
* Department defaults to 'HR'.

CREATE TABLE Employees (

EmpID INT PRIMARY KEY,

EmpName VARCHAR(50) NOT NULL,

Salary DECIMAL(10,2) CHECK(Salary > 5000),

Department VARCHAR(30) DEFAULT 'HR'

);

**Question 4: Insert Invalid Salary**

Try inserting an employee with a salary less than 5000, and observe the constraint violation.

**INSERT INTO Employees (EmpID, EmpName, Salary)**

**VALUES (101, 'Priya Sharma', 4000);**

**-- This will fail due to CHECK constraint**

**Question 5: Product Table with Stock Constraint**

Create a Products table where:

* ProductID is the primary key.
* ProductName cannot be null.
* Price defaults to 100.
* Stock must be non-negative.

**CREATE TABLE Products (**

**ProductID INT PRIMARY KEY,**

**ProductName VARCHAR(50) NOT NULL,**

**Price DECIMAL(10,2) DEFAULT 100,**

**Stock INT CHECK(Stock >= 0)**

**);**

**Question 6: Insert with Constraint Violation**

Suppose the table Products requires Stock to be non-negative. Insert a product with a negative stock and explain the result.

**INSERT INTO Products (ProductID, ProductName, Stock)**

**VALUES (2, 'Monitor', -3);**

**-- Fails due to CHECK (Stock >= 0)**

**Question 7: Update with a Default Value**

Update the price of all products in the Products table to the default price if their price is currently null.

**UPDATE Products**

**SET Price = DEFAULT**

**WHERE Price IS NULL;**

**Question 8: Deleting with a Constraint**

Delete a student from the Students table whose age is 25.

**DELETE FROM Students**

**WHERE Age = 25;**

**Question 9: Update Violating CHECK Constraint**

Try to update an employee's salary in the Employees table to a value below the allowed minimum and explain what happens.

**UPDATE Employees**

**SET Salary = 4000**

**WHERE EmpID = 101;**

**-- Fails due to CHECK (Salary > 5000)**

**Question 10: Insert Into NOT NULL Column**

Attempt to insert a student record without a name and explain the result.

**INSERT INTO Students (StudentID, Age)**

**VALUES (2, 20);**

**-- Fails due to NOT NULL constraint on Name**

**Question 11: Set Default with UPDATE**

Set all employee departments to their default value.

**UPDATE Employees**

**SET Department = DEFAULT;**

**Question 12: Delete All Products with Default Price**

Delete all records in Products where the price is equal to the default.

**DELETE FROM Products**

**WHERE Price = 100;**

**Question 13: Multi-Column CHECK Constraint**

Create a table Orders where:

* OrderID is the primary key.
* Amount must be greater than zero.
* Discount must be less than Amount.
* CustomerName cannot be null.

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerName VARCHAR(50) NOT NULL,

Amount DECIMAL(10,2) CHECK (Amount > 0),

Discount DECIMAL(10,2),

CONSTRAINT chk\_valid\_discount CHECK (Discount < Amount)

);