**Lab 1: PRIMARY KEY Constraint**

**Exercise:**

Create a table Students with a StudentID column as the primary key.

CREATE TABLE Students (

StudentID INT PRIMARY KEY,

Name VARCHAR(50),

Age INT

);

**Solution:**

INSERT INTO Students (StudentID, Name, Age) VALUES (1, 'John Doe', 20);

INSERT INTO Students (StudentID, Name, Age) VALUES (2, 'Jane Smith', 22);

-- This will fail because StudentID must be unique

INSERT INTO Students (StudentID, Name, Age) VALUES (1, 'Mark Johnson', 21);

**Lab 2: FOREIGN KEY Constraint**

**Exercise:**

Create a table Courses and link it to Students using a foreign key.

CREATE TABLE Courses (

CourseID INT PRIMARY KEY,

CourseName VARCHAR(100),

StudentID INT,

FOREIGN KEY (StudentID) REFERENCES Students(StudentID)

);

**Solution:**

INSERT INTO Students (StudentID, Name, Age) VALUES (3, 'Alice Brown', 23);

INSERT INTO Courses (CourseID, CourseName, StudentID) VALUES (101, 'Math', 3);

-- This will fail because StudentID 99 does not exist in Students

INSERT INTO Courses (CourseID, CourseName, StudentID) VALUES (102, 'Science', 99);

**Lab 3: UNIQUE Constraint**

**Exercise:**

Ensure that the Email column in the Students table contains unique values.

ALTER TABLE Students ADD Email VARCHAR(100) UNIQUE;

**Solution:**

INSERT INTO Students (StudentID, Name, Age, Email) VALUES (4, 'Bob White', 21, 'bob@example.com');

-- This will fail because the email is duplicated

INSERT INTO Students (StudentID, Name, Age, Email) VALUES (5, 'Eve Adams', 22, 'bob@example.com');

**Lab 4: CHECK Constraint**

**Exercise:**

Ensure that Age in Students table must be greater than or equal to 18.

ALTER TABLE Students ADD CONSTRAINT chk\_Age CHECK (Age >= 18);

**Solution:**

INSERT INTO Students (StudentID, Name, Age) VALUES (6, 'Chris Green', 19);

-- This will fail because Age is less than 18

INSERT INTO Students (StudentID, Name, Age) VALUES (7, 'David Black', 17);

**Lab 5: DEFAULT Constraint**

**Exercise:**

Set a default value for Age in Students as 18.

ALTER TABLE Students ADD CONSTRAINT df\_Age DEFAULT 18 FOR Age;

**Solution:**

INSERT INTO Students (StudentID, Name) VALUES (8, 'Emma White');

SELECT \* FROM Students WHERE StudentID = 8; -- Age will be set to 18 automatically

**Lab 6: NOT NULL Constraint**

**Exercise:**

Ensure that Name in Students cannot be NULL.

ALTER TABLE Students ALTER COLUMN Name VARCHAR(50) NOT NULL;

**Solution:**

-- This will fail because Name cannot be NULL

INSERT INTO Students (StudentID, Age) VALUES (9, 20);

**Lab 7: COMPOSITE PRIMARY KEY Constraint**

**Exercise:**

Create a Enrollments table with a composite primary key (StudentID, CourseID).

CREATE TABLE Enrollments (

StudentID INT,

CourseID INT,

Grade CHAR(1),

PRIMARY KEY (StudentID, CourseID),

FOREIGN KEY (StudentID) REFERENCES Students(StudentID),

FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)

);

**Solution:**

INSERT INTO Enrollments (StudentID, CourseID, Grade) VALUES (3, 101, 'A');

-- This will fail because (3, 101) already exists

INSERT INTO Enrollments (StudentID, CourseID, Grade) VALUES (3, 101, 'B');

**Lab 8: UNIQUE Constraint on Multiple Columns**

**Exercise:**

Ensure that a combination of CourseID and StudentID in Enrollments must be unique.

ALTER TABLE Enrollments ADD CONSTRAINT uq\_Course\_Student UNIQUE (CourseID, StudentID);

**Solution:**

INSERT INTO Enrollments (StudentID, CourseID, Grade) VALUES (4, 101, 'B');

-- This will fail because (4, 101) must be unique

INSERT INTO Enrollments (StudentID, CourseID, Grade) VALUES (4, 101, 'A');

**Lab 9: CASCADE DELETE Constraint**

**Exercise:**

Modify Courses table so that deleting a student also deletes related courses.

ALTER TABLE Courses DROP CONSTRAINT Courses\_StudentID\_FK;

ALTER TABLE Courses ADD CONSTRAINT Courses\_StudentID\_FK

FOREIGN KEY (StudentID) REFERENCES Students(StudentID) ON DELETE CASCADE;

**Solution:**

DELETE FROM Students WHERE StudentID = 3;

-- The related course with StudentID = 3 will also be deleted

**Lab 10: CASCADE UPDATE Constraint**

**Exercise:**

Modify Courses table so that updating StudentID in Students table updates it in Courses.

ALTER TABLE Courses DROP CONSTRAINT Courses\_StudentID\_FK;

ALTER TABLE Courses ADD CONSTRAINT Courses\_StudentID\_FK

FOREIGN KEY (StudentID) REFERENCES Students(StudentID) ON UPDATE CASCADE;

**Solution:**

UPDATE Students SET StudentID = 10 WHERE StudentID = 3;

-- The related StudentID in Courses will also be updated to 10