Assignment 3

1. Create a database named StudentManagement.

→ CREATE DATABASE StudentManagement;

2. Define the following tables with appropriate constraints:

Table 1: Students

Columns:

StudentID (Integer, PRIMARY KEY, NOT NULL)

FirstName (VARCHAR(50), NOT NULL)

LastName (VARCHAR(50))

Email (VARCHAR(100), UNIQUE)

DOB (DATE, NOT NULL)

CourseID (Integer, FOREIGN KEY references Courses(CourseID))

→ USE StudentManagement;

CREATE TABLE Student\_180

(

Student\_ID INT PRIMARY KEY NOT NULL,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50),

Email VARCHAR(50) UNIQUE,

DOB DATE NOT NULL,

Course\_ID INT NOT NULL,

FOREIGN KEY (Course\_ID) REFERENCES Courses\_180(Course\_ID)

);

Table 2: Courses

Columns:

CourseID (Integer, PRIMARY KEY, NOT NULL)

CourseName (VARCHAR(100), NOT NULL, UNIQUE)

Credits (Integer, DEFAULT 3)

**→** CREATE TABLE Courses\_180

(

Course\_ID INT PRIMARY KEY NOT NULL,

Course\_name VARCHAR(100) UNIQUE NOT NULL,

Credits INT DEFAULT 3

);

3. Insert records into the tables while ensuring the constraints are not violated.

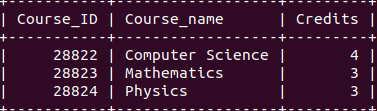
→ INSERT INTO Courses\_180 (Course\_ID, Course\_name, Credits)

VALUES

(28822, 'Computer Science', 4),

(28823, 'Mathematics', 3),

(28824, 'Physics', 3);



INSERT INTO Student\_180 (Student\_ID, FirstName, LastName, Email, DOB, Course\_ID)

VALUES

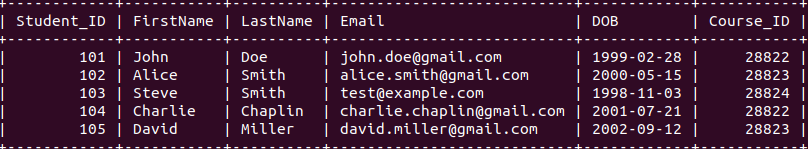
(101, 'John', 'Doe', 'john.doe@gmail.com', '1999-02-28', 28822),

(102, 'Alice', 'Smith', 'alice.smith@gmail.com', '2000-05-15', 28823),

(103, 'Steve', 'Smith', 'Steve.smith@gmail.com', '1998-11-03', 28824),

(104, 'Charlie', 'Chaplin', 'charlie.chaplin@gmail.com', '2001-07-21', 28822),

(105, 'David', 'Miller', 'david.miller@gmail.com', '2002-09-12', 28823);



4. Test the Constraints

Attempt to violate the constraints and observe the results:

1.

Test 1: Insert a NULL value into a NOT NULL column.

→ INSERT INTO Student\_180 (Student\_ID, FirstName, LastName, Email, DOB, Course\_ID)

VALUES

(106, NULL, 'Brown', 'test.brown@gmail.com', '2000-08-10', 28822);



2.

Test 2: Insert a duplicate value in the UNIQUE column.

→ INSERT INTO Student\_180 (Student\_ID, FirstName, LastName, Email, DOB, Course\_ID)

VALUES

(107, 'Emma', 'Jones', 'john.doe@gmail.com', '1999-05-22', 28823);



3.

Test 3: Insert a record with an invalid FOREIGN KEY reference.

→ INSERT INTO Student\_180 (Student\_ID, FirstName, LastName, Email, DOB, Course\_ID)

VALUES

(108, 'Oliver', 'Williams', 'oliver.williams@gmail.com', '2001-03-14', 99999);



5. Attempt following questions:

1. Write a query to display the names of students who were born after the year

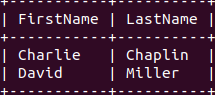
2000.

→

SELECT FirstName, LastName

FROM Student\_180

WHERE YEAR(DOB) > 2000;



2. Write a query to update the course credits for "Computer Science" to 4.

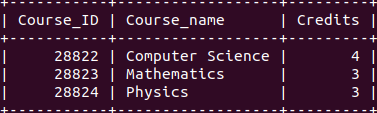
→

UPDATE Courses\_180

SET Credits = 4

WHERE Course\_name = 'Computer Science';

SELECT\* FROM Courses\_180;



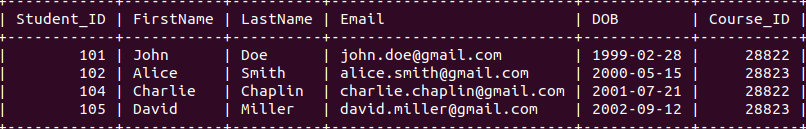
3.Write a query to delete a student record whose email is ['test@example.com](mailto:'test@example.com)'.

→

DELETE FROM Student\_180

WHERE Email = 'test@example.com';

SELECT\* FROM Student\_180;



4. Write a query to find students who are not enrolled in any course.

→ SELECT FirstName, LastName

FROM Student\_180

WHERE Course\_ID IS NULL;



5. Test the UNIQUE constraint by inserting a duplicate email into the Students

table.

→ INSERT INTO Student\_180 (Student\_ID, FirstName, LastName, Email, DOB, Course\_ID)

VALUES

(109, 'Liam', 'Taylor', 'john.doe@gmail.com', '2002-04-10', 28823);

SELECT\* FROM Student\_180;



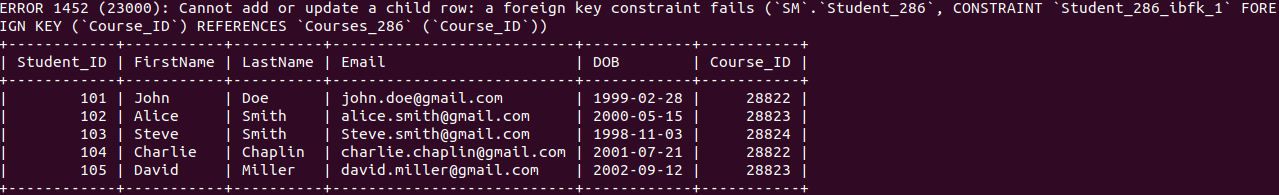
6. Test the FOREIGN KEY constraint by inserting a student with a CourseID

that does not exist in the Courses table.

→ INSERT INTO Student\_180 (Student\_ID, FirstName, LastName, Email, DOB, Course\_ID)

VALUES

(110, 'Sophia', 'Brown', 'sophia.brown@gmail.com', '2001-06-18', 99999);

 SELECT\* FROM Student\_180;