**8. Write and execute suitable database triggers .Consider row level and statement level triggers.**

**Ans :**

.  **Row-Level Trigger**:

* We will create a row-level trigger that automatically updates the EnrollmentDate in the Enrollment table whenever a new enrollment is added or an existing enrollment is updated. For simplicity, the trigger will set the EnrollmentDate to the current date (SYSDATE) when a row is inserted or updated.

 **Statement-Level Trigger**:

* We will create a statement-level trigger that prevents any deletion from the Student table if a student is enrolled in any course. The trigger will check if the student is enrolled in any course by looking at the Enrollment table. If the student is enrolled, the deletion will be canceled, and an error message will be displayed.

**Complete SQL Code with Triggers**:

**-- Step 1: Create the database (if not already created)**

CREATE DATABASE IndianCollegeDB;

**-- Step 2: Use the created database**

USE IndianCollegeDB;

**-- Step 3: Create the Instructor table with Primary Key**

CREATE TABLE Instructor (

InstructorID INT PRIMARY KEY,

Name VARCHAR(100),

Department VARCHAR(100)

);

**-- Step 4: Create the Student table with Primary Key**

CREATE TABLE Student (

StudentID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100),

Email VARCHAR(100)

);

**-- Step 5: Create the Course table with Foreign Key reference to Instructor table**

CREATE TABLE Course (

CourseID INT PRIMARY KEY,

Title VARCHAR(100),

InstructorID INT,

FOREIGN KEY (InstructorID) REFERENCES Instructor(InstructorID)

);

**-- Step 6: Create the Enrollment table (junction table) with Foreign Keys**

CREATE TABLE Enrollment (

EnrollmentID INT PRIMARY KEY AUTO\_INCREMENT,

StudentID INT,

CourseID INT,

EnrollmentDate DATE,

FOREIGN KEY (StudentID) REFERENCES Student(StudentID),

FOREIGN KEY (CourseID) REFERENCES Course(CourseID)

);

**-- Step 7: Insert sample data into Instructor table**

INSERT INTO Instructor (InstructorID, Name, Department) VALUES

(1, 'Dr. Rajesh Kumar', 'Computer Science'),

(2, 'Prof. Meena Agarwal', 'Mathematics'),

(3, 'Dr. Arvind Sharma', 'Physics');

**-- Step 8: Insert sample data into Student table**

INSERT INTO Student (Name, Email) VALUES

('Amit Patel', 'amit.patel@example.com'),

('Priya Sharma', 'priya.sharma@example.com'),

('Ravi Kumar', 'ravi.kumar@example.com'),

('Neha Singh', 'neha.singh@example.com'),

('Vikram Joshi', 'vikram.joshi@example.com');

**-- Step 9: Insert sample data into Course table**

INSERT INTO Course (CourseID, Title, InstructorID) VALUES

(1, 'Introduction to Programming', 1),

(2, 'Data Structures', 1),

(3, 'Calculus I', 2),

(4, 'Quantum Mechanics', 3),

(5, 'Linear Algebra', 2);

**-- Step 10: Insert sample data into Enrollment table**

INSERT INTO Enrollment (StudentID, CourseID, EnrollmentDate) VALUES

(1, 1, '2025-04-01'),

(2, 2, '2025-04-02'),

(3, 3, '2025-04-03'),

(4, 4, '2025-04-04'),

(5, 5, '2025-04-05');

**-- Step 11: Create a Row-Level Trigger to automatically update EnrollmentDate**

CREATE OR REPLACE TRIGGER trg\_update\_enrollment\_date

AFTER INSERT OR UPDATE ON Enrollment

FOR EACH ROW

BEGIN

**-- Set the EnrollmentDate to the current date whenever a new enrollment is inserted or updated**

:NEW.EnrollmentDate := SYSDATE;

DBMS\_OUTPUT.PUT\_LINE('Enrollment Date set to ' || TO\_CHAR(SYSDATE, 'YYYY-MM-DD') || ' for EnrollmentID ' || :NEW.EnrollmentID);

END;

/

**-- Step 12: Create a Statement-Level Trigger to prevent deletion of students enrolled in any course**

CREATE OR REPLACE TRIGGER trg\_prevent\_delete\_student

BEFORE DELETE ON Student

FOR EACH ROW

DECLARE

v\_enrollment\_count INT;

BEGIN

-- Check if the student is enrolled in any course by looking in the Enrollment table

SELECT COUNT(\*) INTO v\_enrollment\_count

FROM Enrollment

WHERE StudentID = :OLD.StudentID;

IF v\_enrollment\_count > 0 THEN

**-- If the student is enrolled in any course, prevent the deletion**

RAISE\_APPLICATION\_ERROR(-20001, 'Cannot delete student. This student is enrolled in one or more courses.');

END IF;

END;

/

**-- Step 13: Execute the Row-Level Trigger (Example: Insert a new enrollment)**

INSERT INTO Enrollment (StudentID, CourseID) VALUES (2, 3);

**-- This will trigger the row-level trigger and automatically set the EnrollmentDate to the current date**

**-- Step 14: Execute the Statement-Level Trigger (Example: Try to delete a student enrolled in a course)**

DELETE FROM Student WHERE StudentID = 2;

**-- This will trigger the statement-level trigger, and the deletion will be prevented if the student is enrolled in any course**

**-- Step 15: Show the data from the Enrollment table to check EnrollmentDate after the trigger**

SELECT \* FROM Enrollment;

**-- Step 16: Show the data from the Student table (no deletion should happen for enrolled students)**

SELECT \* FROM Student;