**6. Write and execute PL/SQL stored procedure and function to perform a suitable task on the database. Demonstrate its use.**

**Ans :**

### 1. **Stored Procedure**: A stored procedure will be created to insert a new student into the Student table. It will take the student's name and email as parameters.

### 2. **Function**: A function will be created to calculate and return the total number of students enrolled in a specific course. It will take a CourseID as input and return the count of students enrolled in that course.

**Complete PL/SQL Code:**

**-- Step 1: Create the database**

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 Stored Procedure to Add a New Student**

CREATE OR REPLACE PROCEDURE AddNewStudent(

p\_StudentName IN VARCHAR(100),

p\_Email IN VARCHAR(100)

)

IS

BEGIN

-- Insert a new student into the Student table

INSERT INTO Student (Name, Email)

VALUES (p\_StudentName, p\_Email);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Student ' || p\_StudentName || ' has been added successfully.');

END;

/

**-- Step 12: Create a Function to Get Total Enrolled Students for a Given Course**

CREATE OR REPLACE FUNCTION GetTotalEnrolledStudents(

p\_CourseID IN INT

)

RETURN INT

IS

total\_enrollment INT;

BEGIN

-- Count the number of students enrolled in the given course

SELECT COUNT(\*)

INTO total\_enrollment

FROM Enrollment

WHERE CourseID = p\_CourseID;

RETURN total\_enrollment;

END;

/

**-- Step 13: Execute the Stored Procedure to Add a New Student**

EXEC AddNewStudent('Suresh Verma', 'suresh.verma@example.com');

**-- Step 14: Call the Function to Get the Total Number of Students Enrolled in Course 1**

DECLARE

v\_TotalStudents INT;

BEGIN

**-- Get the total number of students enrolled in Course 1**

v\_TotalStudents := GetTotalEnrolledStudents(1);

DBMS\_OUTPUT.PUT\_LINE('Total students enrolled in Course 1: ' || v\_TotalStudents);

END;

/

**-- Step 15: Show the List of All Students**

SELECT \* FROM Student;

**-- Step 16: Show the List of All Enrollments (Student-Course Pairings)**

SELECT S.Name AS StudentName, C.Title AS CourseTitle, E.EnrollmentDate

FROM Student S

JOIN Enrollment E ON S.StudentID = E.StudentID

JOIN Course C ON E.CourseID = C.CourseID;

**-- Step 17: Show the Total Number of Students Enrolled in Each Course**

SELECT C.Title AS CourseTitle, COUNT(E.StudentID) AS TotalEnrolled

FROM Course C

LEFT JOIN Enrollment E ON C.CourseID = E.CourseID

GROUP BY C.Title;