Advanced Database Management System

Assignment : 1

Name: S M ABU HURYRA

ID: 20-42480-1

Section: D

SStudent

SNUM	SNAME	STANDING	GPA
111	Andy	4	
222	Betty	2	
333	Cindy	3	

Course

CNUM	CTITLE	CRHR	STANDING	CAPACITY
240	Intro to	3	2	5
301	MIS	3	3	5
380	Statistics	3	3	3
	Database			

Enroll

EID	S#	C#	GRADE
1	111	240	A
2	333	240	B
…	.....	.....	....

- For a given student, if his total credit hours is between 0-30, update his standing to "1"; if histotal credit hours is between 31-60, update his standing to "2"; if his total credit hours is between 61- 90, update his standing to "3"; if his total credit hours is greater than 91, then update his standing to "4". Please write a PL/SQL program to do this.
- Write a procedure AddCourse(psnum, pcnum)that will enroll the student to a course. The program should check for the following things:

  1. The student must be a valid student.

  2. The course must be a valid course.

  3. There is still room in the class.

  4. After enrolling, the total credit hours of the student does not exceed 15 credit hours.

  5. The student is not currently enrolled in this class. You can check for current enrollment by a NULL grade.

  CheckValidStudent(psnum) that returns TRUE when the student exists in the STUDENTtable; FALSE otherwise.

  CheckValidCourse(pcnum)that returns TRUE when the course exists in the COURSE table; FALSE otherwise.

  CheckStanding(snum, cnum) that will return True is the student has appropriate standing to take the course, false otherwise.

  CheckClassCapacity(cnum) that will return true is the class still has room for one more student, and false otherwise.

1.

```
CREATE OR REPLACE FUNCTION update_student_status (p_credit_hours IN NUMBER, p_student_id
IN NUMBER)
RETURN VARCHAR2 IS
  v_student_status VARCHAR2(10);
BEGIN
  IF p_credit_hours BETWEEN 0 AND 30 THEN
    v_student_status := '1';
  ELSIF p_credit_hours BETWEEN 31 AND 60 THEN
    v_student_status := '2';
  ELSIF p_credit_hours BETWEEN 61 AND 90 THEN
    v_student_status := '3';
  ELSE
    v_student_status := '4';
  END IF;

  UPDATE student_table SET standing = v_student_status WHERE student_id = p_student_id;

  DBMS_OUTPUT.PUT_LINE('Student standing updated to ' || v_student_status);

  RETURN v_student_status;
END;
```

2.

```
CREATE OR REPLACE PROCEDURE AddCourse(psnum IN NUMBER, pcnum IN NUMBER) IS
  v_student_status VARCHAR2(10);
  v_course_status VARCHAR2(10);
  v_class_capacity NUMBER;
  v_total_credit_hours NUMBER;
BEGIN
  IF NOT CheckValidStudent(psnum) THEN
    DBMS_OUTPUT.PUT_LINE('Invalid student ID');
    RETURN;
  END IF;

  IF NOT CheckValidCourse(pcnum) THEN
    DBMS_OUTPUT.PUT_LINE('Invalid course ID');
    RETURN;
  END IF;

  IF    NOT    CheckClassCapacity(pcnum)    THEN
    DBMS_OUTPUT.PUT_LINE('Class      is      full');
    RETURN;
  END IF;

  IF NOT CheckStanding(psnum, pcnum) THEN
    DBMS_OUTPUT.PUT_LINE('Student does not have appropriate standing to take this course'); RETURN;
  END IF;

  SELECT grade INTO v_student_status FROM Enroll WHERE student_id = psnum AND course_id =
pcnum;

  IF v_student_status IS NOT NULL THEN DBMS_OUTPUT.PUT_LINE('Student is
    already enrolled in this class'); RETURN;
  END IF;
```

```
   SELECT SUM(credit_hours) INTO v_total_credit_hours FROM Enroll WHERE student_id = psnum
AND grade IS NOT NULL;

   IF v_total_credit_hours + (SELECT credit_hours FROM Course WHERE course_id = pcnum) > 15 THEN
      DBMS_OUTPUT.PUT_LINE('Total credit hours of the student will exceed 15 after enrolling in this
course');
      RETURN;
   END IF;

   INSERT INTO Enroll (student_id, course_id) VALUES (psnum, pcnum); DBMS_OUTPUT.PUT_LINE('Student

   with ID ' || psnum || ' has been enrolled in course with
ID ' || pcnum);
END;
```