**Advanced Database Management System**

**Lab Exercise**

SStudent

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

Course

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

Enroll

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

* For a given student, if his total credit hours is between 0-30, update his standing to “1”; if his total 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 STUDENT table; 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.

Ans:

1.

declare

total\_cr number;

begin

select count(cnum)\*3 into total\_cr from enroll where snum = ‘111’;

if total\_cr between 0 and 30 then

update sstudent set standing = 1 where snum = ‘111’;

elsif total\_cr between 31 and 60 then

update sstudent set standing = 2 where snum = ‘111’;

elsif total\_cr between 61 and 90 then

update sstudent set standing = 3 where snum = ‘111’;

elsif total\_cr >91

update sstudent set standing = 4 where snum = ‘111’;

end if;

end;

2.

create or replace procedure AddCourse(psnum sstudent.snum%type, pcnum course.cnum%type)

is

isValidcourse number;

isValidSudent number;

isStillRoom number;

takenCredit number;

prevgrade enroll.grade%type;

isValidTocarry number;

begin

isValidcourse:=CheckValidCourse(pcnum);

isValidSudent:=CheckValidStudent(psnum);

isStillRoom:=CheckClassCapacity(pcnum);

isValidTocarry:=CheckStanding(psnum, pcnum);

select count(enroll.cnum)\*3 into takenCredit from enroll where grade is null and snum = psnum;

select grade into prevgrade from enroll where cnum = pcnum and snum = psnum;

if isValidcourse = 1 and isValidSudent = 1

and isStillRoom = 1 and isValidTocarry = 1

and takenCredit < 13 and prevgrade is null

then

insert into enroll (snum,cnum) values (psnum,pcnum);

end if;

end;

3.

create or replace function CheckValidStudent(psnum sstudent.snum%type) return number is

sid sstudent.snum%type;

begin

select snum into sid from sstudent where snum = psnum;

if sid is null then

return 0;

else

return 1;

end if;

end;

4.

create or replace function CheckValidCourse(pcnum course.cnum%type) return number is

pid course.cnum%type;

begin

select cnum into pid from course where cnum = pcnum;

if pid is null then

return 0;

else

return 1;

end if;

end;

5.

create or replace function CheckStanding(ssnum sstudent.snum%type , ccnum

course.cnum%type) return number

is

sst sstudent.standing%type;

cst course.standing%type;

begin

select standing into sst from sstudent where snum = ssnum;

select standing into cst from course where cnum = ccnum;

if sst >= cst then

return 1;

else

return 0;

end if;

end;

6.

create or replace function CheckClassCapacity(ccnum course.cnum%type) return number

is

cts course.capacity%type;

cfill course.capacity%type;

begin

select count(enroll.cnum) into cfill from enroll where enroll.cnum = ccnum;

select capacity into cts from course where cnum = ccnum;

if cts > cfill then

return 1;

else

return 0;

end if;

end;