**Name:Partha Malakar**

**Id:20-42908-1**

**Section:A**

**ADVABNCED DATABASE MANAGEMENT SYSTEM**

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.

**Table create:**

drop table Cource

create table SStudent(

SNUM NUMBER(4) CONSTRAINT PK\_SStudent PRIMARY KEY,

SNAME varchar(255),

STANDING NUMBER(4),

GPA varchar(255)

);

create table Cource(

CNUM NUMBER(4) CONSTRAINT PK\_Cource PRIMARY KEY,

CTITLE varchar(255),

CRHR int,

STANDING NUMBER(4),

CAPACITY int

);

create table Enroll(

EID NUMBER(4) CONSTRAINT PK\_Enroll PRIMARY KEY,

S# NUMBER(4) CONSTRAINT FK\_S# REFERENCES SStudent,

C# NUMBER(4) CONSTRAINT FK\_C# REFERENCES Cource,

GRADE varchar(255)

);

**insert value into sstudent table**

INSERT INTO SStudent VALUES (111,'Andy',4,'');

INSERT INTO SStudent VALUES (222,'Betty',2,'');

INSERT INTO SStudent VALUES (333,'Cindy',3,'');

**insert value into Cource table**

INSERT INTO Cource VALUES (240,'Intro to MIS',3,2,5);

INSERT INTO Cource VALUES (301,'Statistics',3,3,5);

INSERT INTO Cource VALUES (380,'Database',3,3,3);

**insert value into Enroll table**

INSERT INTO Enroll VALUES (1,111,240,'A');

INSERT INTO Enroll VALUES (2,333,240,'B');

SELECT \* FROM SStudent

**answer1:**

declare

d\_no SStudent.SNUM %type:=:SNUM;

a number(4);

begin

select sum(CRHR) into a from cource where CNUM in(select C# from enroll where S# = d\_no);

if(a>=0 or a<=30) then

update SStudent set STANDING = 1 where SNUM=d\_no;

elsif(a>=31 or a<=60) then

update SStudent set STANDING = 2 where SNUM=d\_no;

elsif(a>=61 or a<=90) then

update SStudent set STANDING = 3 where SNUM=d\_no;

elsif(a>91) then

update SStudent set STANDING = 4 where SNUM=d\_no;

end if;

end;select \* from SStudent

**Answer2:**

create or replace procedure AddCourse(psnum in SStudent.SNUM%type,pcnum in Cource.CNUM%type)

is

A number(2);

B number(2);

C number(2);

D number(2);

begin

A:=checkValidStudent(psnum);

B:=checkValidCourse(pcnum);

C:=CheckStanding(snum,cnum);

D:=checkClassCapacity(cnum);

if(A=0 and B=0 and C=0 and D=0) then

Insert into enroll values(3,psnum ,pcnum,' ');

dbms\_output.put\_line('inserted');

else

dbms\_output.put\_line('invalid');

endif;

end;

create or replace function checkValidStudent(psnum in sstudent.SNUM%type)

return number

cursor c1

is

i number;

begin

for i in c1 loop

if (i.SNUM = psnum) then

return 1;

else

return 0;

endif;

end loop;

end;

create or replace function checkValidCourse(pcnum in cource.CNUM%type)

return number

is

cursor c1;

i number;

begin

for i in c1 loop

if (i.CNUM = pcnum) then

return 1;

else

return 0;

endif;

end loop;

end;

create or replace function CheckStanding(psnum in sstudent.SNUM%type,pcnum in cource.CNUM%type)

return number

is

e SStudent.STANDING%type;

m Cource.STANDING%type;

begin

select Standing into e from SStudent where SNUM=psnum;

select Standing into m from Cource where CNUM=pcnum;

if (e=m) then

return 1;

else

return 0;

endif;

end;

create or replace function checkClassCapacity(pcnum in cource.CNUM%type)

return number

is

a number(4);

begin

select sum(CRHR) into a from cource where CNUM in(select C# from enroll where S# = 111);

if (a<15) then

return 1;

else

return 0;

endif;

end;

declare

psnum SStudent.SNUM %type:=:SNUM;

pcnum Cource.CNUM %type:=:CNUM;

begin

AddCourse(psnum,pcnum);

end;