Pl sql:

*--update the grade for student in specfic course and if he get f*

create or replace procedure update\_grade( v\_id in number , v\_course\_id in number , v\_grade in char)

as

actual\_grade char(2);

begin

select grade into actual\_grade

from enrollment

where student\_id=v\_id and course\_id = v\_course\_id;

if (actual\_grade !='F')

then

update enrollment

set grade = v\_grade

where student\_id = v\_id and course\_id = v\_course\_id ;

elsif(actual\_grade='F'and v\_grade in ('A+' ,'A-','A'))

then

update enrollment

set grade = 'B+'

where student\_id = v\_id and course\_id = v\_course\_id;

else

update enrollment

set grade = v\_grade

where student\_id = v\_id and course\_id = v\_course\_id ;

***dbms\_output.put\_line***('updated successfully');

end if;

end;

*--calling update if Student get F*

begin

update\_grade(1,3,'A');

end;

*-- trigger to remove the courses that not in the department if i update it*

create or replace trigger update\_dept\_trigger

after update of dept\_id on students

for each row

begin

if :new.dept\_id != :old.dept\_id then

delete from enrollment

where student\_id = :new.student\_id

and course\_id not in (

select course\_id

from courses

where dept\_id = :new.dept\_id );

end if;

end ;

*--update department for student*

create or replace procedure update\_dept\_id(v\_id in number , v\_dept\_id in number)

is

begin

update students

set dept\_id = v\_dept\_id ,

gpa = null

where student\_id = v\_id;

***dbms\_output.put\_line***('updated successfully');

end;

*--calling update*

begin

update\_dept\_id(1,2);

end;

*--update course*

set serveroutput on;

create or replace procedure update\_course(v\_id in number , v\_old\_course in number,v\_new\_course in number)

is

v\_dept\_id number;

v\_dept\_course number;

begin

select dept\_id into v\_dept\_id

from students

where student\_id =v\_id;

if v\_dept\_id is not null then

select dept\_id into v\_dept\_course

from courses

where course\_id = v\_new\_course;

if v\_dept\_id = v\_dept\_course then

update enrollment

set course\_id = v\_new\_course,grade=null

where student\_id=v\_id and course\_id=v\_old\_course ;

***DBMS\_OUTPUT.PUT\_LINE***('updated successfully');

else

RAISE\_APPLICATION\_ERROR(-20001, 'The new course does not belong to this department');

end if;

end if;

end ;

*--calling*

begin

update\_course(2,25,3);

end;

*---------------------------------*

*-- procedure to calculate gpa for one student*

set serveroutput on ;

create or replace procedure update\_gpa

is

total\_hours number :=0;

total\_grades number :=0;

v\_gpa number(3,2) := 0;

begin

for student\_record IN (SELECT DISTINCT student\_id FROM enrollment ) LOOP

v\_gpa := 0; *-- reset gpa*

total\_hours := 0;

total\_grades := 0; *-- Reset total hours*

declare

v\_student\_id number := student\_record.student\_id;

cursor enrollment\_cursor is

select e.student\_id, e.course\_id , e.grade, c.course\_hours

from enrollment e

join courses c on e.course\_id=c.course\_id

where e.student\_id=v\_student\_id;

begin

for course\_record in enrollment\_cursor

loop

total\_hours := total\_hours + course\_record.course\_hours;

case course\_record.grade

when 'A+' then total\_grades := total\_grades + (4 \* course\_record.course\_hours);

when 'A' then total\_grades := total\_grades + (4 \* course\_record.course\_hours);

when 'A-' then total\_grades := total\_grades + (3.7 \* course\_record.course\_hours);

when 'B+' then total\_grades := total\_grades + (3.3 \* course\_record.course\_hours);

when 'B' then total\_grades := total\_grades + (3 \* course\_record.course\_hours);

when 'B-' then total\_grades := total\_grades + (2.7 \* course\_record.course\_hours);

when 'C+' then total\_grades := total\_grades + (2.3 \* course\_record.course\_hours);

when 'C' then total\_grades := total\_grades + (2 \* course\_record.course\_hours);

when 'C-' then total\_grades := total\_grades + (1.7 \* course\_record.course\_hours);

when 'D+' then total\_grades := total\_grades + (1.3 \* course\_record.course\_hours);

when 'D' then total\_grades := total\_grades + (1 \* course\_record.course\_hours);

when 'F' then total\_grades := total\_grades + (0 \* course\_record.course\_hours);

*--handle if the grade updated to 0 when the student update the course*

when '0' then total\_grades := total\_grades + (0 \* course\_record.course\_hours);

end case;

end loop;

if total\_hours != 0 then

v\_gpa := total\_grades / total\_hours;

*--dbms\_output.put\_line(v\_student\_id||'-'|| v\_gpa);*

UPDATE students

SET gpa = v\_gpa

WHERE student\_id = v\_student\_id;

end if;

end;

end loop;

end;

*---------------------*

*--calling*

begin

update\_gpa;

end;

*---------------------*