LAB 8

Aniket sambher

Reg no-190905466

Section-A

Roll no-58

1.

```
CREATE OR REPLACE PROCEDURE listDept (deptName Student.dept_name%type) IS
        CURSOR c1 (deptName student.dept_name%type) IS
                SELECT name FROM Student WHERE dept_name = deptName;
        CURSOR c2 (deptName course.dept_name%type) IS
                SELECT course_id FROM Course WHERE dept_name = deptName;
BEGIN
        DBMS_OUTPUT.PUT_LINE('---------------------------------');
        DBMS_OUTPUT.PUT_LINE('-- DEPARTMENT STUDENTs --');
        FOR row IN c1(deptName) LOOP
                DBMS_OUTPUT.PUT_LINE('    ' || row.name);
        END LOOP;
        DBMS_OUTPUT.PUT_LINE('-----------------------------------------');
        DBMS_OUTPUT.PUT_LINE('-- COURSES --');
        FOR row IN c2 (deptName) LOOP
                DBMS_OUTPUT.PUT_LINE('    ' || row.course_id);
        END LOOP;
END;
/

DECLARE
BEGIN
        listDept('Music');
END;
/
```

2.

```
create or replace procedure course_popular(d_name department.dept_name%type) is
c_name course.title%type;
cursor c1 is select title from (course natural join takes) where dept_name = d_name group by course_id,title
having count(*) = (select max(freq) from (select count(*)as freq from (course natural join takes)
where dept_name = d_name group by course_id,title));
begin
   dbms_output.put('Most popular courses in  '||d_name||' : ');
   for c_tit in c1
      loop
         dbms_output.put(c_tit.title || ', ');
      end loop;
   dbms_output.put_line('');
end;
/

declare
cursor c1 is select distinct dept_name from department;
begin
for dName in c1
   loop
      course_popular(dName.dept_name);
   end loop;
end;
/
```

Triggers

1.//first we create a table log_change_takes

create table log_change_Takes (Time_Of_Change date,ID varchar(5),ccourseid varchar(8),sec_id varchar(8),semester varchar(6),year numeric(4,0),grade varchar(2));

```
create or replace trigger takes_log
before insert or update or delete on takes
for each row
begin
case
when deleting then
insert into log_change_Takes
values(current_timestamp,:old.id,:old.course_id,:old.sec_id,:old.semester,:old.year,:old.grade);
else
insert into log_change_Takes
values(current_timestamp,:new.id,:new.course_id,:new.sec_id,:new.semester,:new.year,:new.grade
);
end case;
end;
/
```

2.

```
create table Old_Data_Instructor(
ID varchar(5),
name varchar(20) not null,
dept_name varchar(20),
salary numeric(8,2) check (salary > 29000),
primary key (ID),
foreign key (dept_name) references department
on delete set null
);
```

```
create or replace trigger log_Instructor
before update of salary on instructor
for each row
begin
insert into Old_Data_Instructor values(:old.ID,:old.name,:old.dept_name,:old.salary);
end;
/
```

3.

```
create or replace trigger instructorTrigger
before insert on instructor
for each row
declare
deptBud number(10);
begin
        select budget into deptBud from department where dept_name=:new.dept_name;
  if regexp_like(:new.name, '[^A-Za-z]')
  then
    RAISE_APPLICATION_ERROR(-20000,'Invalid name');
        end if;
        if :new.salary<=0 then
    RAISE_APPLICATION_ERROR(-20000,'Salary should be positive');
        end if;
  if :new.salary>deptBud then
    RAISE_APPLICATION_ERROR(-20000,'Salary should be less than budget');
  end if;
end;
/
```

4.

```
create table Client_master(
   client_no varchar(5) primary key,
   name varchar(20),
   address varchar(20),
   Bal_due number
   );
create table auditclient(
   client_no varchar(5) primary key,
   name varchar(20),
   bal_due number,
   operation varchar(20),
   userid varchar(5) default('00000'),
   opdate date
   );
create or replace trigger client_audit
        before update or delete on Client_master
        for each row
        begin
                case
                        when updating then
                                insert into auditclient
values(:OLD.client_no,:OLD.name,:OLD.bal_due,'upd',NULL,sysdate);
                        when deleting then
                                insert into auditclient
values(:OLD.client_no,:OLD.name,:OLD.bal_due,'del',NULL,sysdate);
                end case;
        end;
        /
```