**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;

/