**Nikhil bhalala**

**Csc352**

**--q1)**

DROP TRIGGER Dept\_Del ;

drop table Dept\_log;

CREATE TABLE Dept\_log(

Updated\_Date DATE default SYSDATE,

Updated\_By Varchar2 (15) default USER,

Action Varchar2 (30)

);

CREATE OR REPLACE TRIGGER Dept\_Del

AFTER DELETE ON Departments

BEGIN

INSERT INTO Dept\_log (Updated\_date, Updated\_By, Action)

VALUES (SYSDATE, User, 'Table Changed');

END Dept\_Del;

DELETE departments where department\_id = 150;

DELETE departments where department\_id > 249;

rollback;

--select \*from departments--;

--select \* from dept\_log;

**Output**

Graphical user interface, text, application

Description automatically generated

**--q2)**

DROP TRIGGER Dept\_Del\_Row;

DROP table Dept\_Del\_log;

CREATE TABLE Dept\_Del\_log (

Old\_Deptno number (4),

OLD\_Deptname Varchar2 (30),

OLD\_MgrID number (6),

OLD\_LocID number (4),

Updated\_Date DATE,

Updated\_By Varchar2 (15),

Action Varchar2 (30)

);

CREATE OR REPLACE TRIGGER Dept\_Del\_Row

AFTER DELETE ON departments

FOR EACH ROW

BEGIN

INSERT INTO Dept\_Del\_log VALUES

(:old.department\_id, :OLD.department\_name, :old.manager\_id, :old.location\_Id, Sysdate, user, 'tabel changed');

END Dept\_Del\_Row;

DELETE departments where department\_id = 150;

DELETE departments where department\_id > 249;

--select \* from Dept\_Del\_log;

--rollback;

**Output**

Graphical user interface, text, application

Description automatically generated

**--Q3)**

DROP TRIGGER Dept\_Del\_Row;

DROP table Dept\_Access\_log;

CREATE TABLE Dept\_Access\_log(

Deptno number (4),

OLD\_Deptname Varchar2 (30),

NEW\_Deptname Varchar2 (30),

OLD\_MgrID number (6),

NEW\_MgrID number (6),

OLD\_LocID number (4),

NEW\_LocID number (4),

Updated\_Date DATE,

Updated\_By Varchar2 (15),

Action Varchar2 (30)

);

CREATE OR REPLACE TRIGGER Change\_Dept

AFTER INSERT OR DELETE OR UPDATE OF manager\_id, location\_Id ON departments

FOR EACH ROW

BEGIN

IF INSERTING THEN

INSERT INTO Dept\_Access\_log VALUES

(:new.department\_id, null,:new.department\_name, 0,:new.manager\_id, 0, :new.location\_Id, Sysdate, user, 'table changed');

ELSIF UPDATING('manager\_id ') THEN

INSERT INTO Dept\_Access\_log VALUES

(:old.department\_id, :OLD.department\_name , null, :old.manager\_id,:new.manager\_id, :old.location\_Id, null, Sysdate, user, 'UPDATING\_manager\_id');

ELSIF UPDATING('location\_Id') THEN

INSERT INTO Dept\_Access\_log VALUES

(:old.department\_id, :OLD.department\_name,null, :old.manager\_id,null, :old.location\_Id, :new.location\_id, Sysdate, user, 'UPDATING\_location\_id');

ELSIF DELETING THEN

INSERT INTO Dept\_Access\_log VALUES

(:old.department\_id, :OLD.department\_name ,null, :old.manager\_id,null, :old.location\_Id,null, Sysdate, user, 'table changed');

ELSE

DBMS\_OUTPUT.PUT\_LINE('something goes wrong');

END IF;

END Change\_Dept;

Column OLD\_Deptname format A10

Column NEW\_Deptname format A10

Select \* from departments;

SELECT \* from Dept\_Access\_log ;

INSERT INTO departments VALUES (911, 'DeleteMe', 105, 1700);

UPDATE departments set manager\_id = 108 WHERE department\_id = 911;

UPDATE departments set location\_id = 1400 WHERE department\_id = 911;

DELETE departments where department\_id = 911;

--select \* from departments;

--select\*from dept\_access\_log;

--rollback;

**output**

Graphical user interface, text, application

Description automatically generated

**--q4)**

drop PACKAGE pkg\_func;

drop PACKAGE BODY pkg\_func;

CREATE OR REPLACE PACKAGE pkg\_func

IS

FUNCTION Emp\_Sal

( emp\_id number)

RETURN number;

FUNCTION Emp\_Sal

( emp\_mail employees.email%TYPE)

RETURN number;

END pkg\_func;

/

CREATE OR REPLACE PACKAGE BODY pkg\_func

is

FUNCTION Emp\_Sal

( emp\_id number)

RETURN number

IS

ret number (9,2);

BEGIN

SELECT salary into ret

FROM employees

WHERE employee\_id = emp\_id ;

RETURN ret;

END Emp\_Sal ;

FUNCTION Emp\_Sal

( emp\_mail employees.email%TYPE)

RETURN number

IS

ret number (9,2);

BEGIN

SELECT salary into ret

FROM employees

WHERE email = emp\_mail ;

RETURN ret;

END Emp\_Sal ;

end pkg\_func;

/

DECLARE

local\_variable\_al employees.employee\_id%type := Pkg\_func.Emp\_Sal ( 100 ) ;

local\_variable\_a2 employees.email%TYPE:= Pkg\_func.Emp\_Sal ( 'SKING' ) ;

begin

dbms\_output.put\_line(local\_variable\_al);

dbms\_output.put\_line(local\_variable\_a2);

END;

/

--rollback;

**output**

24000

24000