**Name :** Rohit Bangar **Class :** TE-C **Roll No :** TECOC306 **Batch :** C1

-----------------------------------------------------------------------------------------------------

**Assignment-8**

**Title:** Database Trigger

1. **Write** **a** **update,** **delete** **trigger** **on** **clientmstr** **table.** **The** **System** **should** **keep** **track** **of** **therecords** **that** **ARE** **BEING** **updated** **or** **deleted.** **The** **old** **value** **of** **updated** **or** **deleted** **records** **should** **be** **added** **in** **audit\_trade** **table.** **(separate** **implementation** **using** **both** **row** **and** **statement** **triggers)**

**Row** **trigger:**

**SQL> select \* from client\_mstr;**

ACCNO BAL ---------- ----------

101 8000 102 5660 101 7000 104 6000 105 9000 106 4000 106 8004 102 7500

8 rows selected.

**SQL> create or replace trigger t1 after update or delete on client\_mstr**

**2 for each row**

**3**

**4 declare**

**5 op varchar(10); 6 begin**

**7 if updating then 8 op:='update';**

**9 elsif deleting then 10 op:='delete';**

**11 end if;**

**12 insert into audit\_trade values(:old.accno,:old.bal,op); 13 end;**

**14 /**

Trigger created.

**SQL> update client\_mstr set bal=9000 where accno=102;**

2 rows updated.

**SQL> delete from client\_mstr where accno=102;**

2 rows deleted.

**SQL> delete from client\_mstr where accno=106;**

2 rows deleted.

**SQL> update client\_mstr set bal=7000 where accno=105;**

1 row updated.

**SQL> select \* from audit\_trade;**

ACCNO BAL STATUS

---------- ---------- ----------102 5660 update 102 7500 update 102 9000 delete 102 9000 delete 106 4000 delete 106 8004 delete 105 9000 update

7 rows selected.

**SQL> select \* from client\_mstr;**

ACCNO BAL ---------- ----------

101 8000 101 7000 104 6000 105 7000

**Statement trigger:**

**SQL> select \* from client\_mstr;**

ACCNO BAL ---------- ----------

101 8000 101 7000 104 6000 105 7000 102 8000 103 7500 104 8500

7 rows selected.

**SQL> create or replace trigger t1 after update or delete on client\_mstr**

**2**

**3 declare**

**4 op varchar(10);**

**5 begin**

**6 if updating then**

**7 op:='update';**

**8 elsif deleting then**

**9 op:='delete';**

**10 end if;**

**11 insert into audit\_trade values('','',op);**

**12 end;**

**13 /**

Trigger created.

**SQL> delete from client\_mstr where accno=104;**

2 rows deleted.

**SQL> update client\_mstr set bal=9020 where accno=103;**

1 row updated.

**SQL> update client\_mstr set bal=9520 where accno=101;**

2 rows updated.

**SQL> select \* from audit\_trade;**

ACCNO BAL STATUS

---------- ---------- ----------delete update update

**SQL> select \* from client\_mstr;**

ACCNO BAL ---------- ----------

101 9520 101 9520 105 7000 102 8000 103 9020

2. **Write** **a** **before** **trigger** **for** **Insert,** **update** **event** **considering** **following** **requirement:**

**Emp(e\_no,** **e\_name,** **salary)**

I) **Trigger** **action** **should** **be** **initiated** **when** **salary** **is** **tried** **to** **be** **inserted** **is** **less** **than** **Rs.50,000/-**

II) **Trigger** **action** **should** **be** **initiated** **when** **salary** **is** **tried** **to** **be** **updated** **for** **value** **lessthan** **Rs.** **50,000/-**

**Action** **should** **be** **rejection** **of** **update** **or** **Insert** **operation** **by** **displaying** **appropriate** **error** **message.** **Also** **the** **new** **values** **expected** **to** **be** **inserted** **will** **be** **stored** **in** **new** **table** **Tracking(e\_no,** **salary).**

**SQL> create or replace trigger check\_salary before insert or update of salary on emp1**

**2 for each row**

**3 begin**

**4 if inserting then**

**5 if :new.salary<50000 then**

**6 raise\_application\_error(-20003,'Salary cannot be less than 50000');**

**7 end if;**

**8 insert into tracking values(:new.e\_no,:new.salary);**

**9**

**10 elsif updating then**

**11 if :new.salary<50000 then**

**12 raise\_application\_error(-20003,'Salary cannot be less than 50000');**

**13 end if;**

**14 insert into tracking values(:new.e\_no,:new.salary);**

**15 end if;**

**16 end; 17 /**

Trigger created.

**SQL> insert into emp1 values(1,'abc',60000);**

1 row created.

**SQL> insert into emp1 values(2,'xyz',70000);**

1 row created.

**SQL> insert into emp1 values(3,'lmn',65000);**

1 row created.

**SQL> select \* from emp1;**

E\_NO E\_NAME SALARY

---------- ---------- ----------1 abc 60000 2 xyz 70000 3 lmn 65000

**SQL> update emp1 set salary = 70000 where e\_no=2;**

1 row updated.

**SQL> select \* from emp1;**

E\_NO E\_NAME SALARY ---------- ---------- ----------

1 abc 60000 2 xyz 70000 3 lmn 65000

**SQL> select \* from tracking;**

E\_NO SALARY ---------- ----------

1 60000 2 70000 3 65000 2 70000

**3.** **Write** **a** **Database** **trigger** **for** **following** **requirements:**

**Employee** **salary** **of** **last** **three** **month** **is** **stored** **in** **the** **emp\_sal** **table.** **emp\_sal(emp\_no,** **sal1,sal2,sal3)** **before** **inserting** **salary** **into** **emp\_sal** **table,** **if** **salary** **of** **employee** **in** **any** **of** **the** **last** **three** **month** **is** **greater** **than** **Rs.** **50,000/-** **then** **entry** **of** **average** **salary** **along** **with** **emp\_no** **needs** **to** **be** **inserted** **into** **new** **table** **emp\_new(emp\_no,** **avg\_sal).**

**SQL> create or replace trigger avg\_salary before insert on emp\_sal**

**2 for each row**

**3 declare**

**4 aveg number(10);**

**5 begin**

**6 if :new.sal1>50000 or :new.sal2>50000 or :new.sal3>50000 then**

**7 aveg:=(:new.sal1 + :new.sal3 + :new.sal3)/3;**

**8 insert into emp\_new values(:new.emp\_no,aveg);**

**9 end if;**

**10 end; 11 /**

Trigger created.

**SQL> insert into emp\_sal values(1,50001,40000,30000);**

1 row created.

**SQL> insert into emp\_sal values(2,40001,40000,30000);**

1 row created.

**SQL> insert into emp\_sal values(3,40001,70000,30000);**

1 row created.

**SQL> insert into emp\_sal values(4,30000,25000,90000);**

1 row created.

**SQL> insert into emp\_sal values(5,30000,25000,30000);**

1 row created.

**SQL> select \* from emp\_sal;**

EMP\_NO SAL1 SAL2 SAL3 ---------- ---------- ---------- ----------

1 50001 40000 30000 2 40001 40000 30000 3 40001 70000 30000 4 30000 25000 90000 5 30000 25000 30000

**SQL> select \* from emp\_new;**

EMP\_NO AVG\_SAL ---------- ----------

1 36667 3 33334 4 70000