Name : Mayuresh Rajendrraa Deshmukh

Class : TE A

Roll No : TECOA124

Batch : A2

Assignment No. 7

Title: Trigger

1. Write a update, delete trigger on clientmstr table. The System should keep track of the
records 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)

```
SQL> select * from clientmstr24;

      ID NAME
---------- --------------------
       1 ashwin
       2 mayuresh
       3 rutuj
       4 jay
       5 saloni

SQL> select * from audit_trdae24;

no rows selected


SQL> create or replace trigger cl after delete or update on clientmstr24 for each row
  2
  3  declare
  4
  5    op varchar2(6);
  6  begin
  7    if updating then
  8            op:='update';
  9    else
 10            op:='delete';
 11     end if;
 12
 13  insert into audit_trdae24 values(:old.id, :old.name, op);
 14
 15  end;
 16  /

Trigger created.

SQL> update clientmstr24 set name='ashwin mohan' where id=1;

1 row updated.

SQL> delete from clientmstr24 where id=5;

1 row deleted.

SQL> select * from clientmstr24;

        ID NAME
---------- --------------------
         1 ashwin mohan
         2 mayuresh
         3 rutuj
         4 jay

SQL> select * from audit_trdae24;

        ID NAME                 ACTION
---------- -------------------- ------
         1 ashwin               update
         5 saloni               delete
```

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 less than 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> select * from trigemp;

       ENO NAME           SALARY
---------- ---------- ----------
```

```
         1 ashwin         20000
         2 hemangi         49000
         3 rutuj          51000
         4 mayuresh        99000
         5 bhavesh        90000
         6 ram            65000

6 rows selected.

SQL> select * from tracking;

no rows selected

SQL> create or replace trigger emptrig before insert or update on trigemp for each row
  2
  3  declare
  4    op varchar(10);
  5  begin
  6    if updating and :new.salary<50000 then
  7          insert into tracking values(:old.eno,:new.salary,'update');
  8          raise_application_error(-20003,'salary must be greater than 50000');
  9    elsif inserting and :new.salary<50000 then
 10          insert into tracking values(:new.eno,:new.salary,'insert');
 11          raise_application_error(-20003,'salary must be greater than 50000');
 12     elsif updating and :new.salary>50000 then
 13          insert into tracking values(:old.eno,:new.salary,'update');
 14     elsif inserting and :new.salary>50000 then
 15          insert into tracking values(:new.eno,:new.salary,'insert');
 16
 17
 18     end if;
 19  end;
 20  /

Trigger created.
SQL> insert into trigemp values(7,'xyz',90000);

1 row created.

SQL> update trigemp set salary=80000 where eno=1;

1 row updated.

SQL> select * from tracking;

       ENO     SALARY ACTION
---------- ---------- ----------
         7      90000 insert
         1      80000 update
```

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> select * from emp_sal_trig;

       ENO       SAL1       SAL2       SAL3
---------- ---------- ---------- ----------
         1      20000      30000      40000
         2      45000      67000      70000

SQL> select * from emp_new;

no rows selected

SQL> create or replace trigger sal3ck before insert on emp_sal_trig for each row
  2  declare
  3     avg_sal number(10,2);
  4  begin
  5
  6     if :new.sal1>50000 or :new.sal2>50000 or :new.sal3>50000  then
  7             avg_sal:=(:new.sal1+:new.sal2+:new.sal3)/3;
  8             insert into emp_new values(:new.eno,avg_sal);
  9     end if;
 10  end;
 11  /

Trigger created.

SQL> insert into emp_sal_trig values(3,10000,20000,30000);

1 row created.

SQL> insert into emp_sal_trig values(4,60000,20000,30000);

1 row created.

SQL> select * from emp_new;

       ENO    AVG_SAL
---------- ----------
         4   36666.67

SQL> insert into emp_sal_trig values(5,60000,60000,60000);

1 row created.

SQL> select * from emp_new;

       ENO    AVG_SAL
---------- ----------
         4   36666.67
         5      60000
```

```
SQL> insert into emp_sal_trig values(6,2000,5000,70000);

1 row created.

SQL> select * from emp_new;

       ENO    AVG_SAL
---------- ----------
         4   36666.67
         5      60000
         6   25666.67
SQL> select * from emp_sal_trig;

       ENO       SAL1       SAL2       SAL3
---------- ---------- ---------- ----------
         1      20000      30000      40000
         2      45000      67000      70000
         3      10000      20000      30000
         4      60000      20000      30000
         5      60000      60000      60000
         6       2000       5000      70000

6 rows selected.
```