Assignment 7

Q1. Q.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> create table client_mstr176(id int, name char(10), amount int);

Table created.

SQL> create table audit176(id int, name char(10), amount int);

Table created.

SQL>insert into client_mstr176 values(1, 'ABC', 98000);

1 row created.

SQL> insert into client_mstr176 values(2, 'PQR', 78000);

1 row created.

SQL> insert into client_mstr176 values(3, 'XYZ', 125000);

1 row created.

SQL> insert into client_mstr176 values(4, 'KLM', 58000);

1 row created.

SQL> select * from client_mstr176;

ID	NAME	AMOUNT
1	ABC	98000
2	PQR	78000
3	XYZ	125000
4	KLM	58000

```
SQL> create trigger t1
after update or delete
on client_mstr176
for each row
begin
insert into audit176 values(:old.id, :old.name, :old.amount);
end;
/
```

Trigger created.

```
SQL> update client_mstr176
set amount = 97500
where id = 3;
1 row updated.
SQL> update client_mstr176
set amount = 50000
where id = 2;
1 row update.
SQL> select * from audit176;
        ID       NAME            AMOUNT
---------- ---------- --------------------
         3       XYZ             125000
         2       PQR             78000

SQL> drop trigger t1;
Trigger dropped.
SQL> create trigger t1
after update or delete
on client_mstr176
declare
op char(10);
begin
if updating then
op := 'updated';
end if;
insert into audit176 values(op);
end;
/
Trigger created.
SQL> delete from client_mstr176
where id = 3;
1 row deleted.

SQL> select * from audit176;
```

OPERATION

---

deleted

Q 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 errormessage. Also the new values expected to be inserted will be stored in new table Tracking(e_no, salary).

```
create trigger t1
before insert or update of salary
on emp176
for each row
declare
msg1 char(20);
msg2 char(20);
low_amt exception;
begin
if :new.salary<50000 then
:new.salary := NULL;
raise low_amt;
else
if updating then
insert into tracking176 values(:old.e_no, :new.salary);
else
insert into tracking176 values(:new.e_no, :new.salary);
end if;
end if;
exception
when low_amt then
dbms_output.put_line('Please Enter Amount greater than 50000, NULL inserted');
end;
/
```

Trigger created.

SQL> insert into emp176 values(1, ‘ABC’, 30000);

1 row created.

SQL> insert into emp176 values(2, ‘XYZ’, 28000);

Please Enter Amount greater than 50000, NULL inserted

1 row created.

SQL> select * from emp176;

E_NO	E_NAME	SALARY
1	ABC	300000
2	XYZ	

SQL> select * from tracking176;

E_NO	SALARY
1	300000

SQL> update emp176

set salary = 500000

SQL> select * from emp176;

E_NO	E_NAME	SALARY
1	ABC	300000
2	XYZ	

SQL> update emp176

set salary = 500000

where e_no =2;

1 row updated.

SQL> select * from emp176;

E_NO	E_NAME	SALARY

1 ABC 300000

2 XYZ 500000

SQL> update emp176

set salary = 25000

where e_no = 1;

Please Enter Amount greater than 50000, NULL inserted

1 row updated.

SQL> select * from emp176;

E_NO	E_NAME	SALARY
1	ABC	
2	XYZ	500000

SQL> select * from tracking176;

E_NO	SALARY
1	300000
2	500000

Q.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 table emp_sal(emp_no int, sall int, sal2 int, sal3 int);

Table created.

SQL> create table emp_new(emp_no int, avg_sal int);

Table created.

SQL> create trigger t1

before insert on emp_sal

for each row

declare

average int;

```
temp char(10);
begin
if (:new.sal1> 50000 OR :new.sal2> 50000 OR :new.sal3>50000) then
average := (:new.sal1+:new.sal2+:new.sal3)/3;
insert into emp_new values(:new.emp_no, average);
end if;
end;
/
Trigger created.

SQL> insert into emp_sal values(1, 30000, 67000, 125000);
1 row created.
SQL> insert into emp_sal values(2, 1000, 2000, 500);
1 row created.
SQL> select * from emp_sal;
```

EMP_NO	SAL1	SAL2	SAL3
1	30000	67000	125000
2	1000	2000	500

```
SQL> select * from emp_new;
```

EMP_NO	AVG_SAL
1	74000