Name: Vinayak Madan Shete

Roll No.: SYCOC303 Course Name: Project Based Learning - II

Div: C Batch: C4 Course Code: BCE4409

==========

Problem Statement:

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)

1.1) Row-Level Trigger:

```
SQL> select *from clientmstr;

        ID NAME       PHONE_NO
---------- ---------- ----------
       101 ABC        8686868686
       102 DEF        4242424242
       103 GHI        6363636363
       104 JKL        5656565656

SQL> select *from audit_trade;

no rows selected

SQL>
```

```
SQL> --Row Trigger
SQL> create or replace trigger trig_ex after update or delete on clientmstr
  2  for each row
  3  DECLARE
  4  op varchar2(10);
  5  BEGIN
  6  if updating then
  7  op:='update';
  8  insert into audit_trade values(:old.id,:old.name,:old.phone_no,op);
  9  end if;
 10  if deleting then
 11  op:='delete';
 12  insert into audit_trade values(:old.id,:old.name,:old.phone_no,op);
 13  end if;
 14  END;
 15  /

Trigger created.

SQL>
```

```
SQL> update clientmstr set name='MNO' where id=104;

1 row updated.

SQL> select *from audit_trade;

        ID NAME         PHONE_NO OPERATION
---------- ---------- ---------- ----------
       104 JKL        5656565656 update

SQL> delete from clientmstr where id=104;

1 row deleted.

SQL> select *from audit_trade;

        ID NAME         PHONE_NO OPERATION
---------- ---------- ---------- ----------
       104 JKL        5656565656 update
       104 MNO        5656565656 delete

SQL>
```

1.2) Statement-Level Trigger:

```
SQL> select *from audit_trade;

no rows selected

SQL>
```

```
SQL> create or replace trigger trig_ex2 after update or delete on clientmstr
  2  DECLARE
  3  op varchar2(10);
  4  BEGIN
  5  if updating then
  6  op:='update';
  7  end if;
  8  if deleting then
  9  op:='delete';
 10  end if;
 11  insert into audit_trade values (", op);
 12  END;
 13  /
```

==========

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> create table EMP2
  2  (
  3  e_no number,
  4  salary number
  5  );

Table created.

SQL> create table Tracking
  2  (
  3  e_no number,
  4  salary number
  5  );

Table created.

SQL> CREATE OR REPLACE TRIGGER trig_ex3
  2  BEFORE INSERT OR UPDATE ON EMP2
  3  FOR EACH ROW
  4  DECLARE
  5      v_salary NUMBER;
  6  BEGIN
  7      -- Check if the new salary is less than Rs. 50,000/-
  8      IF :NEW.salary < 50000 THEN
  9          -- Display error message and reject the operation
 10    INSERT INTO Tracking (e_no, salary) VALUES (:NEW.e_no, :NEW.salary);
 11          RAISE_APPLICATION_ERROR(-20001, 'Salary cannot be less than Rs. 50,000/-');
 12      END IF;
 13  END;
 14  /

Trigger created.
```

```
SQL> insert into EMP2 values(101,2563);
insert into EMP2 values(101,2563)
            *
ERROR at line 1:
ORA-20001: Salary cannot be less than Rs. 50,000/-
ORA-06512: at "SYSTEM.TRIG_EX3", line 8
ORA-04088: error during execution of trigger 'SYSTEM.TRIG_EX3'


SQL> select *from Tracking;

no rows selected

SQL> insert into EMP2 values(101,256300);

1 row created.

SQL> select *from EMP2;

      E_NO     SALARY
---------- ----------
       101     256300
```

==========

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
  2  (
  3  emp_no number,
  4  sal1 number,
  5  sal2 number,
  6  sal3 number
  7  );

Table created.

SQL> create table emp_new
  2  (
  3  emp_no number,
  4  avg_sal number
  5  );

Table created.

SQL> CREATE OR REPLACE TRIGGER trig_ex4
  2  BEFORE INSERT ON emp_sal
  3  FOR EACH ROW
  4  DECLARE
  5      v_avg_sal NUMBER;
  6  BEGIN
  7      v_avg_sal := (:NEW.sal1 + :NEW.sal2 + :NEW.sal3) / 3;
  8      IF :NEW.sal1 > 50000 OR :NEW.sal2 > 50000 OR :NEW.sal3 > 50000 THEN
  9          INSERT INTO emp_new (emp_no, avg_sal) VALUES (:NEW.emp_no, v_avg_sal);
 10      END IF;
 11  END;
 12  /

Trigger created.

SQL>
```

```
Trigger created.

SQL> insert into emp_sal values(101,50012,32000,65000);

1 row created.

SQL> select *from emp_sal;

    EMP_NO       SAL1       SAL2       SAL3
---------- ---------- ---------- ----------
       101      50012      32000      65000

SQL> select *from emp_new;

    EMP_NO    AVG_SAL
---------- ----------
       101      49004

SQL>
```

==========