Assignment-10

Procedures, Functions and Triggers:

1. Create a Procedure INCR_OF_EMP in which you will pass empno and amount of increment. If the salary of the specified employee is NULL in Employee table raised the exception 'Salary Missing' and display the message 'This Employee has NULL salary' else update the salary of the employee.

Invoke the INCR_OF_EMP procedure from SQL prompt with some empno and increment amount and follow the consequences.

```
SQL>CREATE PROCEDURE incr_of_emp (emp_no number, amt_incr number)
  2  IS
  3   psal emp.SAL%type;
  4   salary_missing EXCEPTION;
  5  BEGIN
  6   Select SAL into psal from emp where empno=emp_no;
  7    IF psal is NULL then
  8            RAISE salary_missing;
  9   ELSE
 10            Update emp set SAL=SAL+amt_incr where empno=emp_no;
 11   END IF;
 12   EXCEPTION
 13   When salary_missing Then
 14            dbms_output.put_line (emp_no ||'Has salary NULL');
 15   When NO_DATA_FOUND Then
 16            dbms_output.put_line (emp_no||'No such Employee');
 17    END;
 18    /
```

Procedure created.

```
SQL> EXEC incr_of_emp (7876,1000);
```

PL/SQL procedure successfully completed.

2. Create a function REVIEW_EMP in which we will pass empno as parameter and return the incremented values of salary based on the following condition:
If salary <= 1500 then increment will be 20%,
If salary >15000and salary <= 30000 then increment will be 30%
Else increment will be 40%.

Call the REVIEW_EMP function from a PL/SQL Block with empno = 7698 and display the result on the screen.

```
SQL>CREATE OR REPLACE FUNCTION REVIEW_EMP (emp_no number)
  2  RETURN NUMBER
  3  IS
  4     incr emp.SAL%type;
  5     net emp.SAL%type;
  6     vempno emp.empno%type;
  7     vsal emp.SAL%type;
  8     vcomm emp.comm%type;
  9 BEGIN
 10     Select empno,SAL,nvl(comm,0) into vempno,vsal,vcomm from emp where
 11        empno=emp_no;
 12     net:=vsal+vcomm;
 13     IF vsal<=1500 then
 14              incr:=net*0.2;
 15     ELSIF vsal>1500 AND vsal<=3000 Then
 16              incr:=net*0.3;
 17     ELSE
 18              incr:=net*0.4;
 19     END IF;
 20     RETURN(incr);
 21     END;
 22        /
```

```
SQL> DECLARE
  2     i number;
  3  BEGIN
  4     i:=REVIEW_EMP(7698);
  5     dbms_output.put_line(i);
  6  END;
  7  /
855

PL/SQL procedure successfully completed.
```

3. Create a trigger UPPER_STRING for the employee table, which makes the entry in ENAME column in UPPER CASE.

SQL> CREATE TABLE emp_new
2 (ID NUMBER(2),
3 ename VARCHAR2(20));

SQL> CREATE OR REPLACE TRIGGER UPPER_STRING
2 BEFORE INSERT OR UPDATE OF ename ON emp_new
3 FOR EACH ROW
4 BEGIN
5 :new.ename:=UPPER(:new.ename);
6 END;
7 /
Trigger created.

4. Write a trigger TOTAL_SALARY to maintain a derived column TOTSAL of Dept table that stores total salary of all employees in a department.

SQL> CREATE OR REPLACE TRIGGER total_salary
2 AFTER DELETE OR INSERT OR UPDATE OF deptno,sal on emp
3 FOR EACH ROW
4 BEGIN
5 IF DELETING OR (UPDATING AND :old.deptno!=:new.deptno)THEN
6 UPDATE dept set totsal=totsal-:old.sal where deptno=:old.deptno;
7 END IF;
8 IF INSERTING OR (UPDATING AND :old.deptno!=:new.deptno)THEN
9
10 UPDATE dept set totsal=totsal + :new.sal where deptno=old.deptno;
11 END IF;
12 IF (UPDATING AND :old.deptno=:new.deptno AND :old.sal!=:new.sal)THEN
13 UPDATE dept set totsal=totsal-:old.sal+:new.sal where
deptno=:new.deptno;
14 END IF;
15 END;
16 /

Trigger created.

5. Create a INSTEAD OF TRIGGER in which, if a delete statement is fired on a VIEW, say V1 for an employee Name, then the corresponding employee's records gets deleted from Employee table.

SQL> CREATE VIEW VI AS
2 SELECT empno,ename,emp.deptno,job,sal,dname,loc from emp,dept

```
  3  where emp.deptno=dept.deptno;
```

View created.

```
SQL> CREATE OR REPLACE TRIGGER T1
  2  INSTEAD OF DELETE ON VI
  3  FOR EACH ROW
  4  BEGIN
  5     DELETE from emp where deptno=:old.deptno;
  6     DELETE from dept where deptno=:old.deptno;
  7  END;
  8  /
```

Trigger created.

6. Write a SQL query by which you will see the code that defines the TOTAL_SALARY trigger.

SQL>Select DESCRIPTION FROM USER_TRIGGERS WHEN TRIGGER_NAME LIKE "TOTAL_SALARY";

7. Drop the TOTAL_SALARY trigger from the database.

SQL>DROP trigger TOTAL_SALARY;