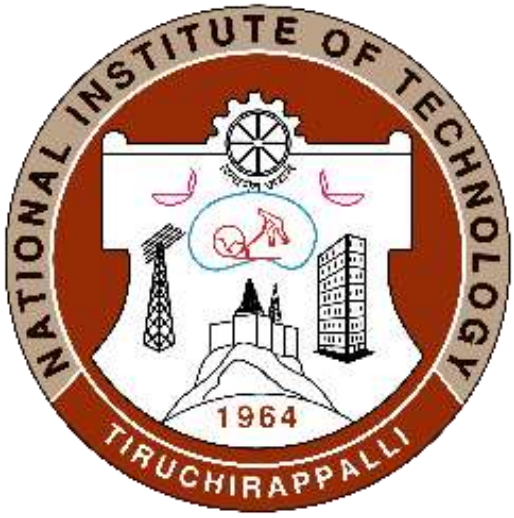

DBMS Lab Manual

Name: Vishesh Kumar Chauarsia

Roll No. : 205119109

Exercise 1:

Problem 1.1: Create a table called EMP with the following structure.

SQL> CREATE table emp(empno number(6),ename varchar2(20),job varchar2(10),mgr number(4),deptno number(3),sal number(7,2));

Problem 1.2: Add a column commission to the emp table Commission numeric null allowed.

SQL> alter table emp add(commission number(10));

Problem 1.3: Modify the column width of the job field of emp table.

SQL> alter table emp modify(job varchar2(20));

Problem 1.4: Create dept table with the following structure.

SQL> create table dept(deptno number(2)primary key,dname varchar2(10),loc varchar2(10));

Problem 1.5: Add constraints to the emp table that empno as the primary key and deptno as the foreign key.

SQL> alter table emp add constraint p primary key (empno);

SQL> alter table emp add constraint f foreign key (deptno)references dept(deptno);

Problem 1.6: Add constraints to the emp table to check the empno value while entering (i.e) empno > 100.

SQL> alter table emp add constraint c check(empno>100);

Problem 1.7: Salary value by default is 5000, otherwise as entered values

SQL> alter table emp modify(sal number(7,2) default '5000');

Problem 1.8: Add columns Dob to the emp table.

SQL> alter table emp add(dob date);

Exercise 2:

Problem 2.1: Insert 3 records into dept table.

```
SQL> insert into dept values(10,'management','main block');

SQL> insert into dept  values(20,'develop','manufact');

SQL> insert into dept  values(30,'maintain','mainblock');

SQL> insert into dept  values(40,'transport','adminblock');

SQL> insert into dept  values(50,'sales','headoffice');
```

Problem 2.2: Insert 10 records into emp table.

```
SQL> insert into emp values(7369,'SMITH','CLERK',7566,20,800,0,'17-DEC-80');

SQL> insert into emp values(7399,'ASANT','SALESMAN',7566,20,1600,300,'20-FEB-81');

SQL> insert into emp values(7499,'ALLEN','SALESMAN',7698,30,1600,300,'20-FEB-81');

SQL> insert into emp values(7521,'WARD','SALESMAN',7698,30,1250,500,'22-FEB-82');

SQL> insert into emp values(7566,'JONES','MANAGER',7839,20,5975,500,'02-APR-81');

SQL> insert into emp values(7698,'BLAKE','MANAGER',7839,30,9820,1400,'01-MAY-79');

SQL> insert into emp values(7611,'SCOTT','HOD',7839,30,3000,NULL,'12-JUN-76');

SQL> insert into emp values(7839,'CLARK','CEO',NULL,10,9900,NULL,'16-MAR-72');

SQL> insert into emp values(7368,'FORD','SUPERVIS',7366,20,800,0,'17-DEC-80');

SQL> insert into emp values(7599,'ALLEY','SALESMAN',7698,30,1600,300,'20-FEB-81');

SQL> insert into emp values(7421,'DRANK','CLERK',7698,30,1250,500,'22-JAN-82');
```

Problem 2.3: Update the emp table to set the default commission of all employees to Rs 1000/- who are working as managers

```
SQL>  UPDATE EMP SET COMMISSION=1000 WHERE JOB='MANAGER';
```

Problem 2.4: Create a pseudo table employee with the same structure as the table emp and insert rows into the table using select clauses.

SQL> CREATE TABLE EMPLOYEE AS SELECT * FROM EMP;

Problem 2.5: Delete only those who are working as supervisors.
SQL> DELETE FROM EMPLOYEE WHERE JOB='SUPERVIS';

Problem 2.6: Delete the rows whose empno is 7599.
SQL> DELETE FROM EMPLOYEE WHERE EMPNO=7599;

Problem 2.7: List the records in the emp table orderby salary in ascending order.
SQL> SELECT * FROM EMP ORDER BY SAL;

Problem 2.8: List the records in the emp table orderby salary in descending order.
SQL> SELECT * FROM EMP ORDER BY SAL DESC;

Problem 2.9: Display only those employees whose deptno is 30.
SQL> SELECT * FROM EMP WHERE DEPTNO=30;

Problem 2.9: Display only those employees whose deptno is 30.
SQL> SELECT * FROM EMP WHERE DEPTNO=30;

Problem 2.11: List the records in sorted order of their employees.
SQL> SELECT * FROM EMP ORDER BY ENAME;

Problem 2.12: create a manager table from the emp table which should hold details aonly about the managers.
SQL> CREATE TABLE MANAGER AS SELECT * FROM EMP WHERE JOB='MANAGER';

Problem 2.13: List the employee names whose commission is null.
SQL> SELECT ENAME FROM EMP WHERE COMMISSION =NULL;

Problem 2.14: List the employee names and the department name in which they are working.

SQL> SELECT ENAME,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO;

Exercise 3:

Problem 3.1: Select all employees from department numbers 7369,7499.

SQL> SELECT * FROM EMP WHERE DEPTNO BETWEEN 7369 AND 7499;

Problem 3.2: Display all the details of the records whose employee name starts with 'S'.

SQL> SELECT * FROM EMP WHERE ENAME LIKE 'S%';

Problem 3.3: Display all the details of the records whose employee name does not starts with 'S'.

SQL> SELECT * FROM EMP WHERE ENAME NOT LIKE 'S%';

Problem 3.4: Display the rows whose empno ranges from 7500 to 7600.

SQL> SELECT * FROM EMP WHERE EMPNO BETWEEN 7500 AND 7600;

Problem 3.5: Display the rows whose empno not in range from 7500 to 7600.

SQL> SELECT * FROM EMP WHERE EMPNO NOT BETWEEN 7500 AND 7600;

Problem 3.6: Calculate the square root of the salary of all employees.

SQL> SELECT SAL,SQRT(SAL) FROM EMP;

Problem 3.7: Count the total records in the emp table.

SQL> SELECT COUNT(*) FROM EMP;

Problem 3.8: Calculate the total and average salary amount of the emptable.

SQL> SELECT SUM(SAL),AVG(SAL) FROM EMP;

Problem 3.9: Determine the max and min salary and rename the column as max_salary and min_salary.

SQL> SELECT MAX(SAL) AS MAX_SALARY,MIN(SAL) AS MIN_SALARY FROM EMP;

Problem 3.10: Display total salary spent for employees.

SQL> SELECT SUM(SAL) FROM EMP;

Problem 3.11: Display total salary spent for each job category.

SQL> SELECT JOB, SUM(SAL) FROM EMP GROUP BY JOB;

Problem 3.12: Display the month name of date "14-jul-09" in full.

SQL> SELECT DATE.TO_WORDS(DOB) FROM EMP;

Problem 3.13: Display the Dob of all employees in the format "dd-mm-yy".

SQL> SELECT ENAME,DOB FROM EMP;

Problem 3.14: Display the date two months after the Dob of employees.

SQL> SELECT ENAME,ADD_MONTHS(DOB,2) FROM EMP;

Problem 3.15: Display the last date of that month in "05-Oct-09".

SQL> SELECT LAST_DAY('05-OCT-09')FROM DUAL;

Problem 3.16: Display the rounded date in the year format, month format, day format in the employees.

SQL> SELECT ROUND(TO_DATE('1-JUN-2009','DD-MM-YY'),'YEAR') FROM DUAL;

SQL> SELECT ROUND(TO_DATE(DOB,'DD-MM-YY'),'MONTH') FROM EMP;

SQL> SELECT ROUND(TO_DATE(DOB,'DD-MM-YY'),'YEAR') FROM EMP;

SQL> SELECT ROUND(TO_DATE(DOB,'DD-MM-YY'),'DAY') FROM EMP;

Problem 3.17: Display the date 60 days before current date.

SQL> SELECT ADD_MONTHS(SYSDATE,-2) FROM DUAL;

Problem 3.18: List all employee names , salary and 15% rise in salary.
SQL> SELECT ENAME,SAL,SAL*.15 FROM EMP;
SQL> SELECT ENAME,SAL,SAL+(SAL*.15) FROM EMP;

Problem 3.19: List all employees which starts with either B or C.
SQL> SELECT ENAME FROM EMP WHERE ENAME LIKE 'B%' or ename like'C%';

Problem 3.20: Display lowest paid employee details under each manager.
SQL> SELECT ALL ENAME,SAL,MGR FROM EMP WHERE SAL IN (SELECT MIN(SAL) FROM EMP GROUP BY MGR);

Problem 3.21: Display number of employees working in each department and their department name.
SQL> select count(*),emp.deptno from emp,dept where emp.deptno=dept.deptno group by emp.deptno;

Problem 3.22: Display the employee names whose name contains up to 5 characters.
SQL> SELECT ENAME FROM EMP WHERE LENGTH(ENAME)<=5;

Problem 3.23: List all employee names and their manager whose manager is 77499 or 7566 0r 7611.
SQL> SELECT ENAME FROM EMP WHERE MGR IN (7499,7611,7566);

Problem3.24: Find how many job titles are available in employee table.
SQL> select count(distinct job) from emp;

Problem 3.25 : What is the difference between maximum and minimum salaries of employees in the organization?
SQL> SELECT MAX(SAL)-MIN(SAL) AS DIFF FROM EMP;

Problem 3.26: Find no.of dept in employee table.
SQL> SELECT COUNT(DEPTNO) FROM EMP;

Problem 3.27: Display the names and dob of all employees who were born in Feburary.
SQL> select ename,dob,dname from emp,dept where emp.deptno=dept.deptno and extract(month from dob)=2;

Problem 3.28: List out the employee names who will celebrate their birthdays during current month.
SQL> select ename,dob,dname from emp,dept where emp.deptno=dept.deptno and extract(month from sysdate)=extract(month from dob);

Problem 3.29: List out the employee names whose names starts with s and ends with h.
SQL> SELECT ENAME FROM EMP WHERE ENAME LIKE 'S%H';

Problem 3.30: List out the employee names whose salary is greater than 5000,6000
SQL> select * from emp where sal>5000 and sal>6000;

Exercise 4:

Problem 4.1: Select all employees from 'maintainance' and 'development' dept.
SQL> SELECT * FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND (DNAME='MAINTAIN' OR DNAME='DEVELOP');

Problem 4.2: Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with 'M'.
SQL> select ename,sal from emp where sal>(select min(sal) from emp)and job like 'M%';

Problem 4.3: Issue a query to find all the employees who work in the same job as jones.
SQL> SELECT ENAME FROM EMP,DEPT WHERE EMP.DEPTNO==DEPT.DEPTNO;

Problem 4.4: Issue a query to display information about employees who earn more than any employee in dept 30.
SQL> SELECT * FROM EMP E WHERE SAL IN (SELECT MAX(E1.SAL) FROM EMP E1 WHERE E1.DEPTNO==E.DEPTNO);

Problem 4.5: Display the employees who have the same job as jones and whose salary >= fords.

SQL> SELECT * FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='JONES') AND SAL>(SELECT SAL FROM EMP WHERE ENAME='FORD');

Problem 4.6: Write a query to display the name and job of all employees in dept 20 who have a job that someone in the Management dept as well.

SQL> SELECT ENAME,JOB FROM EMP e WHERE e.DEPTNO=20 AND JOB in (SELECT JOB FROM EMP,dept WHERE DEPT.DNAME='management');

Problem 4.7: Issue a query to list all the employees who salary is > the average salary of their own dept.

SQL> select * from emp e where sal>(select avg(sal) from emp where deptno=e.deptno group by deptno);

Problem 4.8: Write a query that would display the empname, job where each employee works and the name of their dept.

SQL> select ename,job,dname from emp,dept where emp.deptno=dept.deptno;

Problem 4.9: Write a query to list the employees having the same job as employees located in ' mainblock'.(use multiple subquery)

SQL> select * from emp where job in(select job from emp,dept where emp.deptno=dept.deptno and loc='main block');

Problem 4.10: Write a query to list the employees in dept 10 with the same job as anyone in the development dept.

SQL> select * from emp where deptno=10 and job in (select job from emp,dept where emp.deptno=dept.deptno and dept.dname='develop');

Problem 4.11: Write a query to list the employees with the same job and salary as 'ford'.

SQL> select * from emp where job =(select job from emp where ename='FORD') and sal=(select sal from emp where ename='FORD');

Problem 4.12: Write a query to list all depts. with at least 2 salesman.

SQL> SELECT DNAME FROM DEPT WHERE (SELECT COUNT(*) FROM EMP WHERE JOB='SALESMAN')>=2;

Problem 4.13: Write a query to list the employees in dept 20 with the same job as anyone in dept 30.

SQL> SELECT * FROM EMP WHERE DEPTNO=20 AND JOB IN(SELECT JOB FROM EMP WHERE DEPTNO=30);

Problem 4.14: List out the employee names who get the salary greater than the maximum salaries of dept with dept no 20,30

SQL> SELECT ENAME FROM EMP WHERE SAL>(SELECT MAX(SAL) FROM EMP WHERE DEPTNO=20 OR DEPTNO=30);

Problem 4.15:Display the maximum salaries of the departments whose maximum salary is greater than 9000.

SQL> SELECT DEPTNO,MAX(SAL) FROM EMP WHERE SAL >9000 GROUP BY DEPTNO;

Problem 4.16:Display the maximum salaries of the departments whose minimum salary is greater than 1000 and lesser than 5000.

SQL> SELECT DEPTNO, MAX(SAL) FROM EMP GROUP BY DEPTNO HAVING MIN(SAL)>1000 AND MIN(SAL)<5000;

Create the following table :
AccDept.(Accredited Department by quality council)

DNAME DEPTNO DCity

10	MANAGEMENT	MAIN BLOCK
20	DEVELOPMENT	MANUFACTURING UNIT
30	MAINTAINANCE	MAIN BLOCK

Problem 4.17: Display the departments that are accredited by the quality council.

SQL> select * from dept;
select * from dept,accdept where accdept.deptno=dept.deptno;

Problem 4.18: Display the employees of departments which are not accredited by the quality council
SQL> select * from dept,accdept where accdept.deptno<dept.deptno;

Problem 4.19: Display all the employees and the departments implementing a left outer join.
SQL> select * from emp,dept where emp.deptno(+)=dept.deptno

Problem 4.20: Display the employee name and department name in which they are working implementing a right outer join.
SQL> select ename,dname from emp right outer join dept on emp.deptno=dept.deptno;

Problem 4.21: Display the employee name and department name in which they are working implementing a full outer join.
SQL> select ename,dname from emp full outer join dept on emp.deptno=dept.deptno;

Problem 4.22: Write a query to display their employee names and their managers name.
SQL> select e.ename,m.ename from emp e,emp m where e.mgr=m.empno;

Problem 4.23: Write a query to display their employee names and their managers salary for every employee .
SQL> select e.ename,m.sal as mgrsal from emp e,emp m where e.mgr=m.empno;

Problem 4.24: Write a query to output the name , job, empno, deptname and location for each dept, even if there are no employees.
SQL> select ename,job,empno,dname,loc from emp right outer join dept on emp.deptno=dept.deptno;

Problem 4.25: Find the name of the manager for each employee. Include the following in the output: empno, empname, job and his manager's name.
SQL> select e.ename,m.ename,e.empno,e.job from emp e,emp m where e.mgr=m.empno;

Problem 4.26: Display the details of those who draw the same salary.
SQL> select e.ename,m.ename from emp e, emp m where e.sal=m.sal and e.empno=m.empno;

Exercise 5:

Problem 5.1: Display all the dept numbers available with the dept and accdept tables avoiding duplicates.

SQL> select deptno from dept union select deptno from accdept;

Problem 5.2: Display all the dept numbers available with the dept and accdept tables.

SQL>select deptno from dept union all select deptno from accdept;

Problem 5.3: Display dept no available in both the dept and acc dept tables.

SQL> select deptno from dept intersect select deptno from accdept;

Problem 5.4: Display all the dept numbers available in dept and not in accdept tables.

SQL> select deptno from dept minus select deptno from accdept;

Problem 5.5: The organization wants to display only the details of the employees those who are managers.(horizontal portioning)

SQL> create view managers as select ename from emp where job='manager';
select * from managers;

Problem 5.6: The organization wants to display only the details like empno,empname,deptno,deptname of the employees .(vertical portioning)

SQL> create view general as select enmpno,ename,emp.deptno,dname from emp,dept where emp.deptno=dept.deptno;
select * from general;

Problem 5.7: The organization wants to display only the details like empno,empname,deptno,deptname of the all the employees except the HOD and CEO .(full portioning)

SQL> create view allv as select enmpno,ename,emp.deptno,dname from emp,dept where emp.deptno=dept.deptno and job!='CEO' and job!='HOD';
select * from allv;

Problem 5.8: Display all the views generated.

SQL> Select * from users;

Problem 5.9: Execute the DML commands on the view created.

SQL> select * from general;

Problem 5.10: Drop a view.

SQL> Drop view managers;

Exercise 6:

Program 6.1:write a pl/sql program to swap two numbers with out taking third variable

```
SQL>
declare
a number(10);
b number(10);
begin
a:=&a;
b:=&b;
dbms_output.put_line('THE PREV VALUES OF A AND B WERE');
dbms_output.put_line(a);
dbms_output.put_line(b);
a:=a+b;
b:=a-b;
a:=a-b;
dbms_output.put_line('THE VALUES OF A AND B ARE');
dbms_output.put_line(a);
dbms_output.put_line(b);
end;
```

Program 6.2: write a pl/sql program to swap two numbers by taking third variable

```
SQL>
declare
a number(10);
b number(10);
c number(10);
begin
dbms_output.put_line('THE PREV VALUES OF A AND B WERE');
dbms_output.put_line(a);
dbms_output.put_line(b);
a:=&a;
b:=&b;
c:=a;
a:=b;
b:=c;
dbms_output.put_line('THE VALUES OF A AND B ARE');
dbms_output.put_line(a);
dbms_output.put_line(b);
end;
```

Program 6.3: Write a pl/sql program to find the largest of two numbers

```
SQL>
declare
a number;
b number;
begin
```

```
a:=&a;
b:=&b;
if a=b then
dbms_output.put_line('BOTH ARE EQUAL');
elsif a>b then
dbms_output.put_line('A IS GREATER');
else
dbms_output.put_line('B IS GREATER');
end if;
end;
```

Program 6.4:write a pl/sql program to find the total and average of 6 subjects and display the grade

```
SQL>
declare
java number(10);
dbms number(10);
co number(10);
se number(10); es
number(10); ppl
number(10); total
number(10); avgs
number(10); per
number(10);
begin
dbms_output.put_line('ENTER THE MARKS');
java:=&java;
dbms:=&dbms;
co:=&co;
se:=&se;
es:=&es;
ppl:=&ppl;
total:=(java+dbms+co+se+es+ppl);
per:=(total/600)*100;
if java<40 or dbms<40 or co<40 or se<40 or es<40 or ppl<40 then
dbms_output.put_line('FAIL');
if per>75 then
dbms_output.put_line('GRADE A');
elsif per>65 and per<75 then
dbms_output.put_line('GRADE B');
elsif per>55 and per<65 then
dbms_output.put_line('GRADE C');
else
dbms_output.put_line('INVALID INPUT');
end if;
dbms_output.put_line('PERCENTAGE IS '||per);
```

```
dbms_output.put_line('TOTAL IS '||total);
end;
```

Program 6.5: Write a pl/sql program to find the sum of digits in a given number

```
SQL>
declare
a number;
d number:=0;
sum1 number:=0;
begin
a:=&a;
while a>0
loop
d:=mod(a,10);
sum1:=sum1+d;
a:=trunc(a/10);
end loop;
dbms_output.put_line('sum is'|| sum1);
end;
```

Program 6.6:write a pl/sql program to display the number in reverse order

```
SQL>
declare
a number;
d number:=0;
sum1 number:=0;
begin
a:=&a;
dbms_output.put_line('No. In reverse order');
while a>0
loop
d:=mod(a,10);
a:=trunc(a/10);
dbms_output.put_line(d);
end loop;
end;
```

Program 6.7:Write a pl/sql program to check whether the given number is prime or not

```
SQL>
declare
a number;
d number:=2;
flag number:=0
begin
```

```
a:=&a;
while d<trunc(a/2)
loop
if mod(a,d)=0 then;
dbms_output.put_line(‘The given no. Is not prime no.’);
flag=1;
endif
d:=d+1;
end loop;
if flag=0 then
dbms_output.put_line('The no. is not a prime no.');
end;
```

Program 6.8: Write a pl/sql program to find the factorial of a given number

```
SQL>
Declare
a number;
fac number:=1;
begin
a:=&a;
while a>0
loop
fac:=fac*a;
a:=a-1;
end loop;
end;
```

Program 6.9:write a pl/sql code block to calculate the area of a circle for a value of radius varying from 3 to 7.

```
SQL>
Declare
r number:=3;
area number;
pi  number:=3.14;
begin
r:=&r;
while r<8
loop
area:=2*pi*r;
dbms_output.put_line(‘The area is’|| area);
insert into areas(radius,area) values(r,area);
end loop;
end;
```

Store the radius and the corresponding values of calculated area in an empty table named areas ,consisting of two columns radius & area

TABLE NAME:AREAS

RADIUS
AREA
SQL> create table areas(radius number(10),area number(6,2));

Program 6.10:write a pl/sql code block that will accept an account number from the user,check if the users balance is less than minimum balance,only then deduct rs.100/- from the balance.this process is fired on the acct table.

```
SQL>
declare
accn number;
begin
accn:=&accn;
while select acct,balance from acct;
loop
if accn=acc then
if balance>min(balance)
balance:=balance-100;
endif;
endif;
end loop;
end;
```

Exercise 7:

7.1 Write a procedure to add an amount of Rs.1000 for the employees whose salaries is greater than 5000 and who belongs to the deptno passed as an argument.

```
SQL> create or replace procedure salary(deptid number) as
        begin
            update emp set sal=sal+1000 where sal>5000 AND deptno=deptid;
        end;
```

7.2 Write a PL/SQL block to update the salary of the employee with a 10% increase whose empno is to be passed as an argument for the procedure.

```
SQL> create or replace procedure salary1(empid number) as
        begin
            update emp set sal=sal+sal*(0.1) where empno=empid;
        end;
```

7.3 Write a function to find the salary of the employee who is working in the deptno 20(to be passed as an argument).

```
SQL> create or replace procedure get_sal(dept number) as
      begin
          for s in (select * from emp where deptno = dept)
          loop
            dbms_output.put_line(s.sal);
          end loop;
       end;
```

7.4 Write a function to find the nature of job of the employee whose deptno is 20(to be passed as an argument)

```
SQL> create or replace procedure get_nature(dept number) as
      begin
         for s in (select * from emp where deptno = dept)
         loop
            dbms_output.put_line(s.job);
          end loop;
       end;
```

7.5 Write a PL/SQL block to obtain the department name of the employee who works for deptno 30.

```
SQL> create or replace procedure dep_name(deptid number)  as
      begin
        select dept.dname from dept,emp where emp.deptno=dept.deptno;
      end;
```

Exercise 8:

8.1 Write a Trigger to ensure that DEPT TABLE does not contain duplicate of null values in DEPTNO column.

```
SQL> CREATE OR RELPLACE TRIGGER trig1 before insert on DEPT for each row DECLARE a
number;
        BEGIN
                if(:new.DEPTNO is Null) then
                        raise_application_error(-20001,'error:: DEPTNO cannot be null');
                else
                        select count(*) into a from DEPT where DEPTNO =:new.DEPTNO;
                        if(a=1) then
                                raise_application_error(-20002,'error:: cannot have duplicate DEPTNo ');
                        end if;
                end if;
        END;
```

8.2 Write a Trigger to carry out the following action: on deleting a deptno from dept table , all the records with that deptno has to be deleted from the emp table

```
SQL>  CREATE [OR REPLACE] TRIGGER trig2 Afterdelete on DEPT FOR EACH ROW
        BEGIN
                DELETE FROM emp WHERE emp.deptno=:new.deptno;
        END;
```

8.3 Write a Trigger to carry out the following action: on deleting any records from the emp table,the same values must be inserted into the log table.

```
SQL>  CREATE TRIGGER trig3 AFTER DELETE ON emp FOR EACH ROW
        BEGIN
                INSERT INTO log(val1, val2, ...) VALUES (old.val1, old.val2, ...);
        END;
```