NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI-15

DEPARTMENT OF COMPUTER APPLICATIONS

BATCH-2019-22

DBMS LAB MANUAL

SUBMITTED BY

SHILPI GUPTA
205119095
SECTION:A
MCA(I Year)

SUBMITTED TO

Prof. Dr.U.Vignesh

EXERCISE-1

SQL

Data Definition Language (DDL) commands in RDBMS.

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

EMPNO NUMBER(6)
ENAME VARCHAR2(20)
JOB VARCHAR2(10)
MGR NUMBER(4)
DEPTNO NUMBER(3)
SAL NUMBER(7,2)
Allow NULL for all columns except ename and job.

Solution:

```
CREATE TABLE empl(empno number(6) primary key not null, ename
varchar2(20),job varchar2(20), 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.
Solution:

```
alter table empl add(comm number(5));
```

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

```
alter table empl modify(job varchar2(21));
```

Problem 1.4: Create dept table with the following structure.
Name Type
--------------------------------- ---------------------------
DEPTNO NUMBER(2)
DNAME VARCHAR2(10)
LOC VARCHAR2(10)

Deptno as the primarykey
Solution:

```
create table depart(deptno number(3),dname varchar2(15),loc varchar2(20));
```

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

```
 alter table depart modify(deptno primary key);
```

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

```
 alter table empl add foreign key(deptno) references depart(deptno);
```

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

```
alter table empl add check(empno>100);
```

Problem 1.8: Add columns Dob to the emp table.
Solution:

```
alter table empl modify sal default 5000;
```

EXERCISE-2

Data Manipulation Language (DML) commands in RDBMS

Problem 2.1: Insert 3 records into dept table.

Solution:

```
    insert into depart values(10,'MANAGEMENT','MAIN BLOCK');
    insert into depart values(20,'DEVELOPMENT','MANUFACTURING');
    insert into depart values(30,'MAINTAINANCE','MAIN BLOCK');
    insert into depart values(40,'TRANSPORT','ADMIN BLOCK');
    insert into depart values(50,'SALES','HEAD OFFICE');
```

```
DNAME DEPTNO DLOC
-------- ---------- ------- -- ---------- --------------
10 MANAGEMENT MAIN BLOCK
20 DEVELOPMENT MANUFACTURING UNIT
30 MAINTAINANCE MAIN BLOCK
40 TRANSPORT ADMIN BLOCK
50 SALES HEAD OFFICE
```

Problem 2.2: Insert 10 records into emp table.

Solution:

```
insert into empl values(7369,'smith','clerk',7566,20,800,0,'17-dec-1980');
    insert into empl values(7399,'asant','salesman',7566,20,1600,300,'20-feb-1981');
    insert into empl values(7499,'allen','salesman',7698,30,1600,300,'20-feb-1981');
    insert into empl values(7521,'ward','salesman',7698,30,1250,500,'22-feb-1982');
    insert into empl values(7566,'jones','manager',7839,20,5975,500,'02-apr-1981');
    insert into empl values(7698,'blake','manager',7839,30,9850,1400,'01-may-1979');
    insert into empl values(7611,'scott','hod',7839,10,3000,NULL,'12-jun-1976');
    insert into empl values(7839,'clark','ceo',NULL,10,9900,NULL,'16-mar-1972');
    insert into empl values(7368,'ford','supervisor',7366,20,800,0,'17-dec-1980');
```

```
     insert into empl values(7599,'alley','salesman',7698,30,1600,300,'20-feb-
1981');
     insert into empl values(7421,'drank','clerck',7698,30,1250,500,'22-jan-
1982');
```

```
EMPNO ENAME JOB MGR DOB SAL COMM DEPTNO
-------- ---------- --------- ---------- --------- ---------- ---------- ---------- ---------- ------------
-----
7369 SMITH CLERK 7566 17-DEC-80 800 0 20
7399 ASANT SALESMAN 7566 20-FEB-81 1600 300 20
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-82 1250 500 30
7566 JONES MANAGER 7839 02-APR-81 5975 500 20
7698 BLAKE MANAGER 7839 01-MAY-79 9850 1400 30
7611 SCOTT HOD 7839 12-JUN-76 3000 10
7839 CLARK CEO 16-MAR-72 9900 10
7368 FORD SUPERVIS 7366 17-DEC-80 800 0 20
7599 ALLEY SALESMAN 7698 20-FEB-81 1600 300 30
7421 DRANK CLERCK 7698 22-JAN-82 1250 500 30
```

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

```
update empl set comm=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.
Solution:

Problem 2.5: Delete only those who are working as supervisors.
Solution:

```
delete from employee where job =” supervisor”;
```

Problem 2.6: Delete the rows whose empno is 7599.
Solution:

```
delete from employee where empno=7599;
```

Problem 2.7: List the records in the emp table orderby salary in ascending order.

Solution:

```
select * from employee order by sal;
```

Problem 2.8: List the records in the emp table orderby salary in descending order.

Solution:

```
select * from employee order by sal desc;
```

Problem 2.9: Display only those employees whose deptno is 30.

Solution:

```
select * from employee where deptno =30;
```

Problem 2.10: Display deptno from the table employee avoiding the duplicated values.

Solution:

```
select distinct deptno from employee;
```

Problem 2.11: List the records in sorted order of their employees.

Solution:

```
select * from emp order by empname ;
```

Problem 2.12: create a manager table from the emp table which should hold details aonly about the managers.

Solution:

```
create table manager as select * from emp where job=”manager” ;
```

Problem 2.13: List the employee names whose commission is null.

Solution:

```
select * from emp where comm=null ;
```

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

Solution:

```
select empname,dname from emp,dept where emp.deptno=dept.deptno;
```

EXERCISE-3

In Built functions in RDBMS

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

```
 select * from emp where deptno in(7369,7499);
```

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

```
 select * from employee where empname like ‘s%’;
```

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

```
select * from employee where empname not like ‘s%’;
```

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

```
select * from employee where empno between 7500 and 7600 ;
```

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

```
select * from employee where empno not between 7500 and 7600 ;
```

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

```
select sqrt(sal) from emp;
```

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

```
select count(*) from emp;
```

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

```
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.
Solution:

```
select min(sal) “min_sal”, max(sal) “max_sal” from emp;
```

Problem 3.10: Display total salary spent for employees.
Solution:

```
select sum (sal) from emp ;
```

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

```
select job,sum (sal) from emp group by job;
```

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

```
select to_char(to_date(’14-jul-09’),’month’) from dual;
```

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

```
select to_date(doj,’DD-MM-YY’) from emp;
```

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

```
 select add_months(dob,2) from emp;
```

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

```
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.
Solution:

```
 select round(to_date(dob),’month’) from emp;
 select round(to_date(dob),’year’) from emp;
 select round(to_date(dob),’day’) from emp;
```

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

```
select(sysdate-60) from dual;
```

Problem 3.18: List all employee names , salary and 15% rise in salary.
Solution:

```
select ename , sal , sal+0.15* sal from emp
```

Problem 3.19: List all employees which starts with either B or C.
Solution:

```
select ename from emp where ename like ‘B%’ or ename like ‘C%’
```

Problem 3.20: Display lowest paid employee details under each manager.
Solution:

```
select 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.
Solution:

```
select dname, count (ename) from emp, dept where emp.deptno=dept.deptno
group by
dname
```

Problem 3.22: Display the employee names whose name contains up to 5 characters.
Solution:

```
select empname from emp where length (empname) <=5;
```

Problem 3.23: List all employee names and their manager whose manager is 77499 or 7566 0r 7611.
Solution:

```
select ename from emp where mgr in(7602,7566,7789);
```

Problem3.24: Find how many job titles are available in employee table.
Solution:

```
select count (distinct job) from emp;
```

Problem 3.25 : What is the difference between maximum and minimum salaries of employees in the organization?
Solution:

```
select max(sal)-min(sal) from emp;
```

Problem 3.26: Find no.of dept in employee table.
Solution:

```
select count(distinct deptno) from emp;
```

Problem 3.27: Display the names and dob of all employees who were born in Feburary.
Solution:

```
select empname , dob from emp where to_char (dob,’MON’)=’FEB’;
```

Problem 3.28: List out the employee names who will celebrate their birthdays during current month.
Solution:

```
select pname from programmer where to_char(dob,’MON’) like to_char
(sysdate, ‘MON’);
```

Problem 3.29: List out the employee names whose names starts with s and ends with h.
Solution: select ename from emp where ename like ‘s%h’;

Problem 3.30: List out the employee names whose salary is greater than 5000,6000
Solution:

```
select ename from empl where sal>5000 and sal>6000;
```

EXERCISE-4

Nested Queries & Joins in RDBMS

Problem 4.1: Select all employees from ‘maintainance’ and ‘development’ dept.

Solution:

```
select ename,dname from emp,dept where emp.deptno=dept.deptno and
dname in('maintenance','development');
```

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’.

Solution:

```
select ename,sal from emp where sal in (select sal from emp where job like
'M%');
```

Problem 4.3: Issue a query to find all the employees who work in the same job as jones.

Solution:

```
select * from emp where job in( select job from emp where ename = 'Jones');
```

Problem 4.4: Issue a query to display information about employees who earn more than any employee in dept 30.

Solution:

```
select * from emp where sal > any(select sal from emp where deptno = 30);
```

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

Solution:

```
select * from emp where sal > ( select sal from emp where ename='Ford') and
job= ( select job from emp where ename='Jones');
```

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.

Solution:

```
select ename,job from emp where deptno=20 and job in(select job from
dept,emp where dept.deptno=emp.deptno and dname = 'management');
```

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

Solution:

```
select ename,deptno,sal from emp e1 where sal > (select avg(sal) from emp e2
where e1.deptno=e2.deptno);
```

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

Solution:

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

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

Solution:

```
select ename from emp where job = (select job from emp,dept where
loc='main block' and emp.deptno=dept.deptno);
```

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

Solution:

```
select * from emp where deptno = 10 and job in (select job from emp,dept
where emp.deptno=dept.deptno and dname='development');
```

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

Solution:

```
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.

Solution:

```
select dname from dept where (select count(*) from emp where
job='SALESMAN' and dept.deptno=emp.deptno )  >= 2;
```

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

Solution:

```
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

Solution:

```
SQL> select ename from emp where sal > ( select max(sal) from emp where
deptno = any(20,30) );
```

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

Solution:

```
select 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.

Solution:

```
select max(sal) from emp where  deptno = (select deptno from emp group by
deptno  having  min(sal) between 1000 and 5000);
```

JOINS

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

EQUI-JOIN

~~~~~~~~~

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

Solution:

```
create table accdept as select * from dept where deptno in(10,20,30);
select dept.dname from dept,accdept where dept.dname=accdept.dname;
```

NON-EQUIJOIN

~~~~~~~~~~~~

Problem 4.18: Display the employees of departments which are not accredited by the quality council

Solution:

```
select ename from emp where deptno in (select deptno from dept where
deptno not in ( select deptno from accdept) );
```

LEFTOUT-JOIN

~~~~~~~~~~~~

Problem 4.19: Display all the employees and the departments implementing a left outer join.

Solution:

```
select ename,dname from emp left join dept on dept.deptno=emp.deptno;
```

RIGHTOUTER-JOIN

~~~~~~~~~~~~~~~

Problem 4.20: Display the employee name and department name in which they are working implementing a right outer join.

Solution:

```
select ename,dname from emp right join dept on dept.deptno=emp.deptno;
```

FULLOUTER-JOIN

~~~~~~~~~~~~~~

Problem 4.21: Display the employee name and department name in which they are working implementing a full outer join.

Solution:

```
select ename,dname from emp full outer join dept on
dept.deptno=emp.deptno;
```
~~~~~~~~~~~~~~

SELFJOIN
~~~~~~~~~~~

Problem 4.22: Write a query to display their employee names and their managers name.
Solution:

```
 select a.ename as Employee, b.ename as Manager from emp a, emp b where
a.mgr=b.empno;
```

Problem 4.23: Write a query to display their employee names and their managers salary for every employee .
Solution:

```
select a.ename as Employee, b.ename as Manager,b.sal as manager_salary
from emp a, emp b where a.mgr=b.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.
Solution:

```
select ename,job,empno,dname,loc from emp full 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.
Solution:

```
select a.empno,a.ename as employee,a.job,b.ename as manager_name from
emp a, emp b where a.mgr=b.empno;
```

Problem 4.26: Display the details of those who draw the same salary.
Solution:

```
select ename,sal from emp where sal in (select sal from emp group by sal
having count(*)>1);
```
~~~~~~~~~~~

EXERCISE-5

Set operators & Views in RDBMS

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

```
select deptno from dept union select deptno from accdept;
```

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

```
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.
Solution:

```
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.
Solution:

```
select deptno from dept minus select deptno from accdept;
```

Views

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

```
create view manager1 as select * from emp where job='manager';
```

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

```
create view general as select empno,ename,emp.deptno from emp,dept
where emp.deptno=dept.deptno;
```

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)

Solution:

```
create view all1 as select empno,ename,emp.deptno,dname from emp,dept
where emp.deptno=dept.deptno;
```

Problem 5.8: Display all the views generated.

Solution:

```
select view_name from user_views;
```

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

Solution:

```
insert into manager values(7201,'CLAVE',20,'Computer');
```

Problem 5.10: Drop a view.

Solution:

```
drop view all1;
```

EXERCISE-6

Control Structures

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

Solution:

```
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

Solution:

```
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
Solution:

```
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
Solution:

```
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

Solution:

```
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

Solution:

```
declare
a number;
rev number;
d number;
begin
a:=&a;
rev:=0;
while a>0
loop
d:=mod(a,10);
rev:=(rev*10)+d;
a:=trunc(a/10);
end loop;
dbms_output.put_line('no is'|| rev);
end;
```

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

Solution:

```
declare
a number;
c number:=0;
i number;
begin
a:=&a;
for i in 1..a
loop
if mod(a,i)=0 then
c:=c+1;
end if;
end loop;
if c=2 then
dbms_output.put_line(a ||'is a prime number');
else
dbms_output.put_line(a ||'is not a prime number');
end if;
end;
```

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

Solution:

```
declare
n number;
f number:=1;
begin
n:=&n;
for i in 1..n
loop
f:=f*i;
end loop;
dbms_output.put_line('the factorial is'|| f);
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.

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

Solution:

```
declare
pi constant number(4,2):=3.14;
radius number(5):=3;
area number(6,2);
begin
while radius<7 loop
area:=pi*power(radius,2);
insert into areas values(radius,area);
radius:=radius+1;
end loop;
end;
```

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.

Solution:

```
declare
mano number(5);
mcb number(6,2);
minibal constant number(7,2):=1000.00;
fine number(6,2):=100.00;
```

```
begin
mano:=&mano;
select cur_bal into mcb from acct where acctno=mano;
if mcb<minibal then
update acct set cur_bal=cur_bal-fine where acctno=mano;
end if;
end;
```

EXERCISE-7

Procedures and Functions

Program 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.

Solution:

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

Program 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.

Solution:

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

Program 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).

Solution:

```
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;
```

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

Solution:

```
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;
```

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

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

EXERCISE-8

Triggers

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

```
        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;
```

Program 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
Solution:

```
CREATE [OR REPLACE] TRIGGER trig2 After delete on DEPT FOR EACH ROW

    BEGIN

        DELETE FROM emp WHERE emp.deptno=:new.deptno;

    END;
```

Program 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.
Solution:

```
    CREATE TRIGGER trig3 AFTER DELETE ON emp FOR EACH ROW

    BEGIN
        INSERT INTO log(val1, val2, ...) VALUES (old.val1, old.val2, ...);

     END;
```

EXERCISE-9-10

Project work