

ROLL NO:- 205119075

DBMS LAB MANUAL

<u>Index</u>	<u>Sign</u>	<u>Date!</u>
<u>Exercise 1</u> DDL commands	88	
<u>Exercise - 2</u> - DML commands	92/1/20	31/1/20
<u>Exercise - 3</u> (INBUILT function)	93/1/20	31/1/20
<u>Exercise - 4</u> (Nested Queries join in RDBMS)	93/1/20	28/2/20
<u>Exercises?</u> Set operation & view	93/1/20	28/02/20
<u>Exercise 5</u> Control structures.	93/2/20	06/03/20

Exercise 1.8 → SQL

problem 1.1 Create a table called Emp with the following structure.

Name	Type
EMPNO	Number(6)
ENAME	VARCHAR(20)
JOB	VARCHAR(10)
MGR	Number(4)
DEPTNO	Number(3)
SAL	Number(7,2)

above all columns except MGR and SAL

Ans Syntax of creating table is like this,

Create table Emp(EMPNO int(6), ENAME varchar(10),
JOB varchar(10), MGR int(4),
DEPTNO int(3), SAL decimal(7,2))

to give null Syntax column Data type Null/Not Null
like putting int so giving null
and not null

problem 1.2 Add a column commission for the emp table
Commission numeric null allowed.

→ Syntax

alter table Emp add
add commission int null;

Problem 1.3 modify the column width of the job field
of emp table.

Syntax ALTER TABLE Emp
modify job CHAR(10);

problem 1.4 Create dept table with the following structure.

Name	Type
DEPTNO	Number(2)
DNAME	VARCHAR(20)
LOC	VARCHAR(20)

DeptNo as the primary key.

Syntax:

```
CREATE TABLE dept(DEPTNO INT(2) primary key,  
DNAME VARCHAR(20),  
LOC VARCHAR(20));
```

Problem 1.5

Add constraints to the emp table
that empno as the primary key and
DeptNo as the foreign key.

Syntax:

```
alter table emp  
add constraint deptno foreignkey(DEPTNO)  
References DEPT(DEPTNO);
```

Problem 1.6

alter table emp

```
add check (empno > 100);
```

problem 1-7

alter table Emp
modify (SAL ~~5000~~; DEFAULT 5000);

problem 1-8

alter table Emp

Add DOB Int;

RDB
DB2 commands.

Exercise 2

problem 2-1

Insert into dept VALUES (&DEPTNO, '&DNAME', '&LOC');

4007

After entering in main block

- (1) — Enter value of dept No: 10
- (2) — Enter value of d Name manag
- (3) — enter value of loc main block

(2) → 10
(3) → development
(4) → manufac

(5) → 10
(6) → manufac
(7) → admin block

Problem 2c Insert 10 records into emp table

Insert into Emp VALUES (& EMNO, & ENAME, & JOB, & MGR,
 , & DEPTNO, & SAL, & DB, & Commission)

① Enter value of empno: 7369
 Enter value of ename: Smith
 Enter value of job: close
 " " mgr 7521
 " " deptno: 20
 " " sal : 80
 " " " db: 17
 " " " commission: 0

② " " " empno 7369
 " " " " " 8011T

" " " " " SALESMAN

" " " " " 10
 " " " " " 160
 " " " " " 20
 " " " " " 20

③ 7499

ALLEN

SALESMAN

7298

20

6145

500

30

④ 7521

WARD

MANAGER

7690

21

1000

600

20

⑤ 7658

CLERK

SALESMAN

28

150

16000

6--

50

⑥ 8445

SAL

MAN

7598

30

0525

700

50

⑦ 8998

SAL

MOB

7602

40

8100

800

50

⑧ 8998

KAL

MAN

7690

21

0280

700

70

⑨ 8598

SAB

KAL

7525

22

8500

8000

80

⑩ 9598

RAS

SAL

7695

25

7520

900

800

Ques 2-3. Update the emp table to set the default
Commission of all employees
to Rs 100% where workers are
manager.

Ans update emp set comm=100
where job = 'MANAGER'

Problem

2-4

Create table employee.

or select * from

Problem

2-5

delete from emp

where job = "supervisor";

Problem 2-6

delete from emp

where empno = 7599;

Problem 2-7

select * from emp

order by sal;

problem 2-8

select * from emp

order by sal desc;

problem 2-9

select * from emp

where deptno = 30;

problem 2-10

select distinct deptno

from emp;

~~Ques Ans~~
37/11/2022

In Out function :-

Q3.1 Problem 3.1

Select * from emp
where deptno in (1349, 7499);

Problem 3.2

Select * from emp
where ename like 'S%';

Problem 3.3 Select * from emp
where ename not like 'S%';

problem 3.4

Select * from employee
where empno between 7500 and 7600;

problem 3.5

Select * from employee
where empno not between 7500 and 7600;

problem 3.6 :-

Select sum(sal) from emp;

problem 3.7 :- select count(*) from emp;

Select sum(sal).avg(sal)
from emp;

problem 3.8

Select max(sal) "max-sal" min(sal)
"min-sal" from emp;

Problem

3.10

Select sum(sal) from emp;

Problem 3.11

Select job, sum(sal) from emp
group by job;

Problem

3.12

Select job, sum(sal) from emp

Select to_date('11-jun-09',
'month') from dual;

problem 3.13

Select to_date(dob, 'dd-mm-yy')
from emp;

problem 3.14

Select add_months(dob, 2) from emp;

problem 3.15

Select last_day('105-oct-09') from dual;

problem 3.16

Select round(to_date(dob), 1 month) from emp;

Select round(to_date(dob), 1 year) from emp;

Select round(to_date(dob), 1 day) from emp;

problem 3.17

Select (sysdate - 60) from dual;

problem 3-18

Select ename, sal, 1/8 * sal + 0.10 * sal from emp

problem 3-19

Select ename from emp where
ename like 'C%' or ename like 'CY.'

problem 3-20

Select ename, sal, mgr from emp
where sal is (select max(sal) from emp
group by mgr);

problem 3-21

Select dname, count(ename) from emp, dept
where emp.deptno=dept.deptno
group by dname;

problem

3-22

Select empname from emp where length
(empname) <= 5;

problem

3-23

lakshmi

Select ename from emp where mgr
in (7602, 7516, 7983)

problem

3-24

Select count (distinct job) from emp

Prob/Ch 3-25

Select max(sal)-min(sal) from emp
This will give

Prob/Ch 3-26

Select count and

Select count all (distinct depno) from emp;

Prob/Ch 3-27

Select ename, dob from emp where to-
char(dob, 1 mon) = 'FEB'.

Prob/Ch 3-28

Select pname from programme where
to-cher(dob, 1 mon) like to char
(update (mon))

ans will be given

Prob/Ch 3-29

Select ename from employee
where ename like 'IS%'

Prob/Ch 3-30

Select ename, sal from emp
where sal > 50000 and sal < 60000

Ans will be given

Ques. write query to find no. of
employees whose salary is less than
50000 and greater than 60000

Problem 4.1

Select * from DEPT

WHERE DNAME = 'MAINTAINANCE' OR

DNAME = 'DEVELOPMENT';

Problem 4.2

→ Select ename from emp e, dept d
where e.deptno = d.deptno

and d.dname = 'DEVELOPMENT' or
d.dname = 'MAINTAINANCE'

Problem 4.2

Select ename, sal from emp

where sal > (Select min(sal) from emp
where job like 'M%'))

Problem 4.3

Select * from emp

where job = (select job from emp
where ename = 'JONAS');

Problem 4.4

Select * from emp

where dep > (select max(dep)
from emp)

where emp.deptno = 30;

Problem 4.6

Select * from emp
 where job = (select job from emp
 where name = 'JONATHAN') and
 sub>= (select sal from emp
 where name = 'FORD');

Problem 4.6 :-

Select ename, job from emp
 where dept no = 20 and
 job in (select job from emp, dept d
 where d.deptno = e.deptno
 and d.dname = 'MANAGEMENT'
 and d.loc = 'BOSTON')

Problem 4.7 :-

Select ename from emp
 group by deptno, e.ename
 having e.ename > (select avg(sal))

from emp e1
 inner join dept d on e1.deptno = d.deptno

problem 4.8 :-

Select ename, job, dname, loc from emp e, dept d
 where e.dept no = d.deptno;

Problem 4.9

Select ename from emp e
 where job in (select job from emp e1
 where e1.deptno = d1.deptno
 and d1.loc = 'MANCHESTER')

for 06/07 4.10

Select * from emp where
 deptno = 10 and job in (Select job from
 dept where deptno = 10)
 and emp.deptno = dept.deptno and
 emp.job = dept.job
 and emp.dname = 'development')

problem 4.11

Select * from emp e

where e.job = (Select job from emp
 where emp.dname = 'R&D')

and e.sal = (Select sal from
 emp where empno = 1000)

prob 4.12

Select dname (Count(*)) from emp, dept
 where Emp.deptno = dept.deptno group by
 dept.dname

order by department count(*) >= 2;

problem 4.13

Select e.empno, e.job from emp e

where e.deptno = 20 and

job = any (Select job from emp
 where emp.deptno = 20)

and department = 20

problem 4.17

Select ename from emp
where sal > any (select sal from emp
where deptno in (20,30));

problem 4.18

Select max(sal) from emp
group by deptno having
max(sal) > 9000;

problem 4.19

Select max(sal) from emp
group by deptno having
max(sal) between 1000 and 5000;

problem 4.1A

Select d-dname from dept d, accdept
where d.deptno = accdept.deptno;

problem 4.1B :-

Select e.empno, e.ename, e.job, e.sal,
d.deptno, d.dname from emp e, dept d
accdept a
where e.deptno = d.deptno and d.deptno
= a.deptno;

problem 4.19 :-

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

Problem 4.20

Select ename, dname from emp
 and right join dept, Dept on
 $\text{emp.deptno} = \text{dept.deptno}$

problem 4.21

Select ename, dname from emp
 full join dept on
 $\text{emp.deptno} = \text{dept.deptno}$

problem

4.22 Select e.empno, e.ename as "employees",
 m.mgr, m.ename as manager

m.mgr, m.ename
 from emp inner join emp m
 on e.mgr = m.empno;

problem 4.23

Select e.empno, ename as "employees",
 m.mgr, m.ename, m.job as "manager"
 m.mgr, m.ename
 from emp e inner join emp m
 on e.mgr = m.empno;

problem 4.24

Select e.empno, e.ename, e.job
 as employee, m.mgr, m.ename, as
 manager from emp e inner join
 employee on e.mgr = m.empno
 order by e.ename;

Problems

Select distinct e1-name, e1-sal
from emp e1 inner join emp e2 on e2-sal = e1-sal
and e1-name < e2-name
order by e1-sal;

*Vishnu
27/10/2020*

Exercises

Exercised Problems:-

✓ Select deptno from dept
union

Select deptno from accdept;

Problem 5-2 :-

✓ Select deptno from dept
union all
select deptno from accdept;

Problem 5-3 :-

Select deptno from dept

Intersect

Select deptno from accdept;

problem 5-4 :-

✓ Select deptno from dept

minus

Select deptno from accdept;

Problem 5-5 :-

Create view manager AS

Select *

from Emp

where job = 'manager';

problem 5-6

create view Neeled As
 select empno, emname, deptno, deptname
 From Emp;

problem 5-7

Create views EXP AS
 Select * from Emp
 where job not IN ('HOB', 'COO');
 Select * from MEMP;

problem 5-8

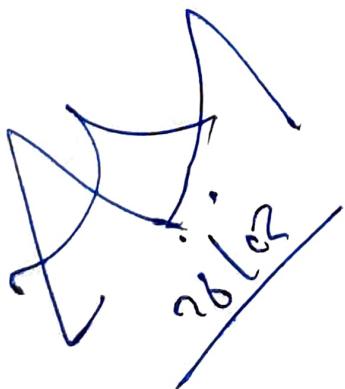
Select * from EXP;

problem 5-9

update view2 Set Comm = Comm * 10

problem 5-10

Drop view EXP;



EXERCISE :- 6

Program 6.1

```
declare
    n : number;
    i : number;
    flag : number;

begin
    i := 2;
    flag := 1; // Initialize flag to 1, assuming n is prime
    n := 8;
    for i := 2 to n/2 do
        if mod(n, i) = 0 then
            then
                flag := 0;
                exit;
        end if;
    end loop;
    if flag = 1 then
        dbms_output.put_line('prime');
    else
        dbms_output.put_line('not prime');
    end if;
end;
```

Program 6.2

```
a : number (10);
b : number (10);
c : number (10);

begin
    a := &a; // Address of variable a
    b := &b; // Address of variable b
    dbms_output.put_line('The prev values of A & B were');

```

```
dbms_output.put_line(a);
dbms_output.put_line(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)
```

Program 6.3

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

Program 6.4

Declare variables

java number(10);

dbms number(10);

co number(10);

float per;

for number(10); ppl

number(10) total

number(10); args

number(10); per

number(10);

dbms - output.put_line('Enter the marks');

begin

java := &java;

dbms := &dbms;

co := &co;

se := &se

es := &~~es~~ es

ppl := &ppl;

total := (java + dbms + co + se + es + ppl);

per := (total + 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 print 'FAIL'

dbms - output.put_line('FAIL');

else if per > 75 then

```

dbms_output.put_line('GRADE IS');
else
  dbms_output.put_line('INVALID INPUT');
end if;
dbms_output.put_line('PERCENTAGE IS'||P);
dbms_output.put_line('TOTAL IS'||T);
end;
/

```

program 6.5

```

declare
  a number:=0;
  d number:=0;
begin
  a:=89;
  while (a>0)
    loop
      d:= mod(a,10);
      sum1:= sum1+d;
      a:= trunc(a/10);
    end loop;
  dbms_output.put_line('sum=||sum1);
end;
/

```

Program 6.6 :-

```

declare
    n number
    i number
    rev number := 0
    r number;
begin
    n := &n;
    while n > 0
        loop
            r := mod(n, 10);
            rev := (rev * 10) + r;
            n := trunc(n / 10);
        end loop;
    dbms_output.put_line('reverse is ' || rev);
end;
/

```

Program 6.7 :-

```

declare
    n number
    i number
    flag number := 1;
begin
    i := 2
    flag := 1;
    n := &n

```

for i in range(n/2)

loop

if mod(ch,i) == 0

then

flag = 0

exit;

end if;

end loop;

if flag == 1

then

dbms_output.put_line('1 prime');

else

dbms_output.put_line('not prime');

end if

end;

/

end if; end loop; end program 6.8;

program 6.8

declara

n number;

fact number := 1;

i number;

begin

n := 8;

for i in 1..n

loop

fact := fact * i;

end loop;

dbms_output.put_line('factorial = ' || fact);

end;

program 6-9

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 area values (radius, area);

radius := radius + 1;

end loop;

end;

(Program 6-9 output after execution)

program 6-10

Create table act(name varchar(10),

cur_fdbk number(5), acc_no number(6,2);

Insert into std values ('&name', &rollno, &mark);

Select * from act;

declare

marks number(5);

mcg number(6,2);

and minbal constant number(7,2) := 1000.00;

and number(6,2) := 100.00;

begin
 mano := & mano;
 Select cur bal into mcb from acc where
 acctno = mano;
 if mcb < minbal then
 update acc set cur bal = cur bal - Am where
 acctno = mano;
 end if;
 end;
 /

Program 7-1

Create table or replace procedure salary
 (deptid number) as

begin
 update emp set sal = sal + 1000
 where sal > 5000 and deptid = deptno;
 end;

Program 7-2

Create or replace procedure salary_d1 (empid
 number) as

begin
 update emp set sal = sal + sal
 * 0.1 where empno = empid;

Program 7-3

Create table 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

Create or replace procedure get-name

(deptnumber) as

begin

for s in (Select * from Emp where
deptno = dept) loop

dbms_output.put_line(s.job);

end loop;

end;

create or replace procedure

get-sal (deptno)

program 7.5

Create or replace procedure

dep-name (deptid|number) by

begin

Select dept_name from dept, emp where

emp.dept_no = dept.dept_no;

(loop 2 - 3. Emp. department name)

end;

program 8.1

CREATE OR REPLACE TRIGGER trig1 before

insert or update for each row declare

insert or update

new DEPTNO number

and count(*) int;

begin

if (:new.DEPTN0 is null) then

begin

raise application error(-20001);

end if, else, if (:new.DEPTN0 is not

be null);

else

select Count(*) into a from DEPT where
DEPTNO = :new.DEPTN0;

if (a=1) then

you're application error (-2002);

error:: can not have duplicate

dept no;

end if;

end if;

END

if -1 < deptno < 10 then

Program 8.2

DEPARTMENT TRIGGER

CREATE OR REPLACE TRIGGER trgr_to_delete

before delete on DEPT FOR EACH ROW

DECLARE

CURSOR get_emp(p_deptno NUMBER) IS

Select empno, lname, job, mgr, sal, comm,
DOB

from EMP

WHERE deptno = p_deptno;

begin

dbms_output.put_line('Delete dept=11:');

old_deptno);

dbms_output.put_line('deptname=' || old_dname);

dbms_output.put_line('dept loc=' || old_loc);

FOR get_emp_rec IN get_emp(:old_deptno)

Loop

```

dbms_output.put('1-'dept_name  

                to char(10) || 'dept_name' || :old.dept_name,  

dbms_output.put_line('1-dept_loc' || :old.dept_loc,  

FOR get_emp_rec IN get_emp(:old.dept_no) Loop  

    dbms_output.put('1-'emp || get_emp_rec.empno);  

    dbms_output.put('1-' || get_emp_rec.ename,  

    dbms_output.put('1-' || get_emp_rec.job);  

    dbms_output.put('1-' || get_emp_rec.sal);  

    dbms_output.put('1-' || get_emp_rec.mng);  

    dbms_output.put('1-' || get_emp_rec.sal);  

    dbms_output.put('1-' || get_emp_rec.job);  

    dbms_output.put_line('1');  

END Loop;  

END;

```

Program ~~for~~ ~~to~~ ~~create~~ ~~trigger~~ ~~on~~ ~~dept~~ ~~table~~ ~~as~~

~~create~~ ~~trigger~~

CREATE TRIGGER AfterDelete
 after delete on [Emp table]
 FOR DELETE
 AS

INSERT INTO [logtable] (

 emphno
 , [emphno]
 , [job]
 , [mgr]
 , [deptno]
 , [sal]
 , [dob])