

**NAME : NITISH KUMAR RATHOUR**

**ROLL NO : 205119063**

**SUBJECT : DBMS LAB MANUAL**

### **PROBLEM 1**

#### PROBLEM 1.1

```
create table EMP(  
    EMPNO INT(6) NOT NULL PRIMARY KEY,  
    ENAME VARCHAR(20) NOT NULL,  
    JOB VARCHAR(10) NOT NULL,  
    MGR INT(4),  
    DEPTNO INT(4),  
    SAL FLOAT(7, 2)  
);
```

---

#### PROBLEM 1.2

```
ALTER TABLE EMP ADD COMMISSION INT;
```

---

#### PROBLEM 1.3

```
ALTER TABLE EMP MODIFY JOB VARCHAR(20);
```

---

#### PROBLEM 1.4

```
CREATE TABLE DEPT(  
  DEPTNO INT(2) NOT NULL PRIMARY KEY,  
  DNAME VARCHAR(50),  
  LOC VARCHAR(50)  
);
```

---

#### PROBLEM 1.5

```
ALTER TABLE EMP ADD CONSTRAINT FK_DEPTNO FOREIGN KEY  
(DEPTNO) REFERENCES DEPT(DEPTNO);
```

---

#### PROBLEM 1.6

```
ALTER TABLE EMP ADD CHECK (EMPNO > 100);
```

---

#### PROBLEM 1.7

```
ALTER TABLE EMP MODIFY SAL FLOAT(7,2) NOT NULL DEFAULT 5000;
```

---

#### PROBLEM 1.8

```
ALTER TABLE EMP ADD DOB DATE;
```

---

---

## **PROBLEM 2**

### PROBLEM 2.1

```
INSERT INTO DEPT VALUES (10,'MANAGEMENT','MAIN BLOCK'),  
(20,'DEVELOPMENT','MAIN BLOCK'),  
(30,'MAINTAINANCE','MAIN BLOCK'),  
(40,'TRANSPORT','ADMIN BLOCK'),  
(50,'SALES','HEAD OFFICE');
```

-----

### PROBLEM 2.2

```
INSERT INTO EMP VALUES (7369,'SMITH','CLERK',7566,20,800,0,'1980-  
12-17'),  
    (7399,'ASANT','SALESMAN',7566,20,1600,300,'1981-02-20'),  
    (7499,'ALLEN','SALESMAN',7698,30,1600,300,'1981-02-20'),  
    (7521,'WARD','SALESMAN',7698,30,1250,500,'1982-02-22'),  
    (7566,'JONES','MANAGER',7839,20,5975,500,'1981-04-02'),  
    (7698,'BLAKE','MANAGER',7839,30,9850,1400,'1979-05-01'),  
    (7611,'SCOTT','HOD',7839,10,3000,NULL,'1976-07-12'),  
    (7839,'CLARK','CEO',NULL,20,800,0,'1972-03-16'),  
    (7368,'FORD','SUPERVIS',7366,20,800,0,'1980-12-12'),  
    (7599,'ALLEY','SALESMAN',7698,30,1600,300,'1981-02-20'),  
    (7421,'DRANK','CLERCK',7698,30,1250,500,'1982-01-22');
```

-----

### PROBLEM 2.3

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

-----

### PROBLEM 2.4

```
CREATE TABLE EMPPSEUDO AS SELECT * FROM EMP;
```

-----

PROBLEM 2.5

DELETE FROM EMP WHERE JOB = 'SUPERVISOR';

---

PROBLEM 2.6

DELETE FROM EMP WHERE EMPNO = 7599;

---

PROBLEM 2.7

SELECT \* FROM EMP ORDERBY SAL;

---

PROBLEM 2.8

SELECT \* FROM EMP ORDER BY SAL;

---

PROBLEM 2.9

SELECT \* FROM EMP WHERE DEPTNO = 30;

---

PROBLEM 2.10

SELECT DISTINCT DEPTNO FROM DEPT;

---

PROBLEM 2.11

SELECT \* FROM EMP ORDER BY ENAME;

---

PROBLEM 2.12

CREATE TABLE MANAGER AS SELECT \* FROM EMP WHERE JOB =  
'MANAGER';

---

PROBLEM 2.13

SELECT \* FROM EMP WHERE COMMISSION = NULL;

---

PROBLEM 2.14

```
SELECT E.ENAME, D.DNAME FROM EMP E, DEPT D WHERE E.DEPTNO =  
D.DEPTNO;
```

## **PROBLEM 1**

### PROBLEM 3.1

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

### 3.2

```
select * from employee where empname like 's%';
```

### 3.3

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

### 3.4

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

### 3.5

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

### 3.6

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

3.7

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

3.8

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

3.9

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

3.10

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

3.11

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

3.12

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

3.13

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

3.14

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

3.15

```
select last_day('05-oct-09') from dual;
```

3.16

```
select round(to_date(dob),'month') from emp;
```

```
select round(to_date(dob),'year') from emp;
```

```
select round(to_date(dob),'day') from emp;
```

3.17

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

3.18

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

3.19

```
select ename from emp where ename like 'B%' or ename like 'C%';
```

3.20

```
select ename, sal,mgr from emp where sal in (select min(sal) from emp  
group by mgr);
```

3.21

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

3.22

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

3.23

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

3.24

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

3.25

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

3.26

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

3.27

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



3.28

select pname from programmer where to\_char(dob,'MON') like  
to\_char (sysdate, 'MON');

3.29

select empname from emp where empname like 'S%H';

3.30

select empname from emp where sal>5000;

## PROBLEM 4

4.1

select ENAME,DNAME from EMP,DEPT where DNAME='MAINTAINANCE' OR  
DNAME='DEVELOPMENT' ;

4.2

SELECT ename FROM emp WHERE sal >(SELECT MIN(sal)FROM emp) AND JOB  
LIKE ('M%');

4.3

SELECT ename FROM EMP WHERE job =( SELECT job FROM emp WHERE  
eNAME='JONES');

4.4

```
SELECT * FROM emp WHERE sal >ANY( SELECT sal FROM emp WHERE  
DEPTNO=30 );
```

4.5

```
SELECT * FROM EMP WHERE job =( SELECT job FROM emp WHERE  
eNAME='JONES') AND SAL>=( SELECT sal FROM emp WHERE ENAME='FORD');
```

4.6

```
SELECT ename, job FROM emp WHERE DEPTNO=10 AND JOB IN(SELECT JOB  
FROM emp,dept WHERE EMP.DEPTNO=DEPT.DEPTNO AND  
Dname='MANAGEMENT');
```

4.7

```
SELECT * FROM emp WHERE sal >(SELECT AVG(SAL)FROM emp);
```

4.8

```
SELECT ENAME,JOB,DNAME FROM EMP,DEPT WHERE  
EMP.DEPTNO=DEPT.DEPTNO;
```

4.9

```
SELECT * FROM EMP WHERE job in (SELECT job FROM emp,dept WHERE  
emp.deptno=dept.deptno and LOC='MAIN BLOCK');
```

4.10

```
SELECT * FROM emp WHERE DEPTNO=10 AND JOB IN(SELECT JOB FROM  
emp,dept WHERE EMP.DEPTNO=DEPT.DEPTNO AND Dname='development');
```

4.11

```
SELECT * FROM EMP WHERE job =( SELECT job FROM emp WHERE  
eNAME='FORD') AND SAL=( SELECT SAL FROM emp WHERE eNAME='FORD');
```

4.12:

```
SELECT DNAME FROM DEPT WHERE DEPTNO=ANY(SELECT DEPTNO FROM
(SELECT COUNT(JOB) AS NO,DEPTNO FROM EMP WHERE JOB='SALESMAN'
GROUP BY DEPTNO) WHERE NO>=2);
```

4.13:

```
SELECT * FROM emp WHERE deptno=20 and job=ANY( SELECT job FROM emp
WHERE DEPTNO=30 );
```

4.14:

```
SELECT eNAME FROM emp WHERE sal >ANY( SELECT sal FROM emp WHERE
DEPTNO IN (20,30));
```

4.15

```
select max(sal) from emp where sal>9000;
```

4.16

```
select min(sal) from emp where sal between 1000 and 5000;
```

4.17

```
select * from emp,dept where emp.deptno=dept.deptno;
```

4.18

```
select * from emp,dept where not emp.deptno=dept.deptno;
```

4.19

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

4.20

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

4.21

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

4.22

```
select ename,job from emp where job='manager';
```

4.23

```
select ename,job,sal from emp where job='manager';
```

4.24

```
select ename,job,dname,loc from emp natural join dept;
```

4.25

```
select e.empno, empname, e.job, m.empname from emp e, emp m where  
e.mgr=m.empno;
```

4.26

```
select e.empname, p.empname from emp e, emp p where e.sal=p.sal ad  
e.empname !=empname;
```

## **PROBLEM 5**

5.1    SELECT DEPTNO FROM DEPT UNION SELECT DEPTNO FROM ACCDEPT;

5.2    SELECT DEPTNO FROM DEPT UNION ALL SELECT DEPTNO FROM ACCDEPT;

5.3    SELECT DEPTNO FROM DEPT INTERSECT SELECT DEPTNO FROM ACCDEPT;

5.4    SELECT DEPTNO FROM DEPT MINUS SELECT DEPTNO FROM ACCDEPT;

5.5    CREATE VIEW MANAGERS AS SELECT \* FROM EMP WHERE  
JOB='MANAGER';  
      SELECT \* FROM MANAGERS;

5.6

```
5.7 CREATE VIEW EMP_ALL AS SELECT  
E.EMPNO,E.EMPNAME,D.DEPTNO,D.DNAME FROM EMP E, DEPT D WHERE  
E.DEPTNO=D.DEPTNO AND E.JOB NOT IN('HOD','CEO');  
SELECT * FROM EMP_ALL;
```

5.8

```
5.9 DROP VIEW EMP_ALL;
```

## **PROBLEM 6**

### PROBLEM 6.1

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

```
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;
/
```

#### PROBLEM 6.2

Write a pl/sql program to swap two numbers by taking third variable

declare

a number(10);

b number(10);

c 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);

```
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;
/
```

### PROBLEM 6.3

Write a pl/sql program to find the largest of two numbers.

```
declare
a number(10);
b number(10);
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;
```

```
/
```

#### PROBLEM 6.4

Write a pl/sql program to find the total and average of 6 subjects and display the grade.

```
declare
```

```
a number;
```

```
b number;
```

```
c number;
```

```
d number;
```

```
e number;
```

```
f number;
```

```
total number;
```

```
per number;
```

```
begin
```

```
dbms_output.put_line('enter marks of subject');
```

```
a:=&a;
```

```
b:=&b;
```

```
c:=&c;
```

```
d:=&d;
```

```
e:=&e;
```

```
f:=&f;
```



```

total:=(a+b+c+d+e+f);
per:=(total/600)*100;
if a<40 or b<40 or c<40 or d<40 or e<40 or f<40 then
dbms_output.put_line('Fail');
elsif per>75 then
dbms_output.put_line('Grade A');
elsif per>65 and per<=75 then
dbms_output.put_line('Grade B');
elsif per>50 and per<=65 then
dbms_output.put_line('Grade C');
else
dbms_output.put_line('Grade D');

end if;
dbms_output.put_line('Total=' || total);
dbms_output.put_line('Percentage=' || per);
end;
/

```

### PROBLEM 6.5

Write a pl/sql program to find the sum of digits in a given number.

```

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;
/

```

#### PROBLEM 6.6

Write a pl/sql program to display the number in reverse order

```

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('reverse no is' || rev);
end;
/
```

#### PROBLEM 6.7

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

```
declare
a number;
c number:=0;
i number;
begin
a:=&a;
for i in 2..a-1
loop
if mod(a,i)=0 then
c:=c+1;
end if;
```

```
end loop;
if c=0 then
dbms_output.put_line(a || 'is a prime number');
else
dbms_output.put_line(a || 'is not a prime number');
end if;
end;
/
```

#### PROBLEM 6.8

Write a pl/sql program to find the factorial of a given number.

```
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;
/
```

### PROBLEM 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.

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;

/

### PROBLEM 6.10

Write a pl/sql code block that will accept an account number from the user, check if the user's balance is less than minimum balance, only then deduct rs.100/- from the

balance. this process is fired on the acct table.

\*\*\*\*\*

## **PROBLEM 7**

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

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

7.3 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 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 create or replace procedure dep\_name(deptid number) as

begin

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

end;

## **PROBLEM 8**

## 8.1

```
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

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

BEGIN

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



END;

### 8.3

CREATE TRIGGER trig3 AFTER DELETE ON emp FOR EACH ROW

BEGIN

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

END;