

**Roll No :  
205119019**

**Subject :  
DBMS Lab**

## Question 1

- 1.1      CREATE TABLE EMP (EMPNO NUMBER(6),EMPNAME VARCHAR(20) NOT NULL,JOB VARCHAR(10) NOT NULL,MGR NUMBER(8),DEPTNO NUMBER(6),SAL NUMBER(7,2));
- 1.2      ALTER TABLE EMP ADD COMMI NUMBER(5);
- 1.3      ALTER TABLE EMP MODIFY JOB VARCHAR(12);
- 1.4      CREATE TABLE DEPT (DEPTNO NUMBER(5) PRIMARY KEY,DNAME VARCHAR(20),DLOC VARCHAR(20));
- 1.5      ALTER TABLE EMP MODIFY EMPNO INT PRIMARY KEY;  
          ALTER TABLE DEPT ADD FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO);
- 1.6      ALTER TABLE EMP ADD CHECK(EMPNO>100);
- 1.7      ALTER TABLE EMP MODIFY SAL FLOAT DEFAULT 5000 NOT NULL;
- 1.8      ALTER TABLE EMP ADD DOB DATE;

## Question 2

- 2.1     INSERT INTO DEPT VALUES(10,'MANAGEMENT','MAIN BLOCK');  
         INSERT INTO DEPT VALUES(20,'DEVELOPMENT','MANUFACTURING UNIT');  
         INSERT INTO DEPT VALUES(30,'MAINTAINANCE','MAIN BLOCK');  
         INSERT INTO DEPT VALUES(40,'TRANSPORT','ADMIN BLOCK');  
         INSERT INTO DEPT VALUES(50,'SALES','HEAD OFFICE');
- 2.2     INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7369,'SMITH','CLERK',7566,'17-DEC80',800,0,20);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7399,'ASANT','SALESMAN',7566,'20-FEB81',1600,300,20);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7499,'ALLEN','SALESMAN',7698,'20-FEB81',1600,300,30);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7521,'WARD','SALESMAN',7698,'22-FEB82',1250,500,30);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7566,'JONES','MANAGER',7839,'02-APR81',5975,500,20);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7698,'BLAKE','MANAGER',7839,'01-MAY79',9850,1400,30);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7611,'SCOTT','HOD',7839,'12-JUN76',3000,NULL,10);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB ,MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7839,'CLARK','CEO',NULL,'16-MAR72',9900,NULL,10);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7368,'FORD','SUPERVIS',7366,'17-DEC80',800,0,20);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7599,'ALLEY','SALESMAN',7698,'20-FEB81',1600,300,30);  
         INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)  
VALUES(7421,'DRANK','CLERCK',7698,'22-JAN82',1250,500,30);
- 2.3     UPDATE EMP SET COMMI=1000 WHERE JOB='MANAGER';
- 2.4     CREATE TABLE EMPLOYEE (EMPNO INTEGER PRIMARY KEY,ENAME VARCHAR(20) NOT NULL,JOB  
VARCHAR(30) NOT NULL,MGR INTEGER,DEPTNO INTEGER,SAL INTEGER,COMM INTEGER,DOB  
VARCHAR(10));  
INSERT INTO EMPLOYEE SELECT\*FROM EMP;
- 2.5     DELETE FROM EMPLOYEE WHERE JOB='SUPERVIS';
- 2.6     DELETE FROM EMPLOYEE WHERE EMPNO=7599;
- 2.7     SELECT \* FROM EMP ORDER BY SAL;
- 2.8     SELECT \* FROM EMP ORDER BY SAL DESC;
- 2.9     SELECT \* FROM EMP WHERE DEPTNO=30;
- 2.10    SELECT DISTINCT DEPTNO FROM EMP;

- 2.11    `SELECT * FROM EMP ORDER BY EMPNAME;`
- 2.12    `create table manager as select * from EMP where JOB='MANAGER';`
- 2.13    `select * from emp where commi IS null;`

## Question 3

- 3.1      `select * from EMP where DEPTNO in (20,10);`
- 3.2      `select * from EMPLOYEE where ENAME like 'S%';`
- 3.3      `select * from EMPLOYEE where ENAME 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) AS MIN_SAL, MAX(SAL) AS 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(DOB,'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 EMPNAME ,SAL , SAL+0.15* SAL from EMP;`
- 3.19     `select EMPNAME from EMP where Empname like 'B%' or Empname like 'C%';`
- 3.20     `select EMPNAME,SAL,MGR from EMP where SAL in (select min(SAL) from EMP group by MGR);`
- 3.21     `select dname, count (empname) from emp, dept where emp.deptno=dept.deptno group by dname;`
- 3.22     `select empname from emp where length(empname)<=5;`
- 3.23     `select empname from emp where mgr in (7399,7698,7566);`

- 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 EMPname from EMP 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;`

## Question 4

- 4.1     SELECT \* FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND  
(DEPT.DNAME='MAINTAINANCE' OR DEPT.DNAME='DEVELOPMENT');
- 4.2     SELECT EMPNAME,SAL FROM EMP WHERE SAL>(SELECT MIN(SAL) FROM EMP) AND JOB LIKE 'M%';
- 4.3     SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNAME='JONES');
- 4.4     SELECT \* FROM EMP WHERE SAL>(SELECT MAX(SAL) FROM EMP WHERE DEPTNO=30);
- 4.5     SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNAME='JONES') AND  
SAL>=(SELECT SAL FROM EMP WHERE EMPNAME='FORD');
- 4.6     SELECT EMPNAME,JOB FROM EMP WHERE DEPTNO=20 AND JOB=ANY(SELECT JOB FROM EMP  
E,DEPT D WHERE E.DEPTNO=D.DEPTNO AND D.DNAME='MANAGEMENT');
- 4.7     SELECT \* FROM EMP OUTER WHERE SAL>(SELECT AVG(SAL) FROM EMP WHERE  
DEPTNO=OUTER.DEPTNO);
- 4.8     SELECT EMPNAME,JOB,DNAME FROM EMP E,DEPT D WHERE E.DEPTNO=D.DEPTNO;
- 4.9     SELECT \* FROM EMP WHERE JOB=ANY(SELECT E.JOB FROM DEPT D,EMP E WHERE  
D.DEPTNO=E.DEPTNO AND DLOC='MAIN BLOCK') AND DEPTNO!=(SELECT DEPTNO FROM DEPT WHERE  
DLOC='MAIN BLOCK');
- 4.10    SELECT \* FROM EMP WHERE DEPTNO=10 AND JOB=ANY(SELECT JOB FROM EMP,DEPT WHERE  
DEPT.DEPTNO=EMP.DEPTNO AND DEPT.DNAME='DEVELOPMENT');
- 4.11    SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNAME='FORD') AND  
SAL=(SELECT SAL FROM EMP WHERE EMPNAME='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);
- //OR
- SELECT DNAME FROM DEPT D WHERE 2<=(SELECT COUNT(EMPNAME) FROM EMP WHERE  
JOB='SALESMAN' AND DEPTNO=D.DEPTNO);
- 4.13    SELECT \* FROM EMP WHERE DEPTNO=20 AND JOB=ANY(SELECT JOB FROM EMP WHERE  
DEPTNO=30);
- 4.14    SELECT \* FROM EMP WHERE SAL>ANY(SELECT MAX(SAL) FROM EMP WHERE DEPTNO=20 OR  
DEPTNO=30 GROUP BY DEPTNO);
- 4.15    SELECT MAX(SAL) FROM EMP GROUP BY DEPTNO HAVING MAX(SAL)>9000;
- 4.16    SELECT MAX(SAL) FROM EMP GROUP BY EMPNAME HAVING MIN(SAL)>1000 AND MIN(SAL)<5000;

//extra table creation

```
CREATE TABLE ACCDEPT (DEPTNO NUMBER(4),DNAME VARCHAR(15),DCITY VARCHAR(20));
```

```
INSERT ALL INTO ACCDEPT (DNAME,DEPTNO,DCITY) VALUES(10,'MANAGEMENT','MAIN BLOCK') INTO  
ACCDEPT (DNAME,DEPTNO,DCITY) VALUES(20,'DEVELOPMENT','MANUFACTURING UNIT') INTO ACCDEPT  
(DNAME,DEPTNO,DCITY) VALUES(30,'MAINTAINANCE','MAIN BLOCK') SELECT 1 FROM DUAL;
```

```
4.17  SELECT A.DNAME FROM DEPT D,ACCDEPT A WHERE D.DEPTNO=A.DEPTNO;
```

```
4.18  SELECT EMPNAME FROM EMP WHERE DEPTNO!=ANY(SELECT DEPTNO FROM ACCDEPT);
```

```
4.19  SELECT * FROM EMP LEFT JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;
```

```
4.20  SELECT * FROM EMP RIGHT JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;
```

```
4.21  SELECT * FROM EMP FULL JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;
```

```
4.22  SELECT E.EMPNAME,M.EMPNAME FROM EMP E,EMP M WHERE E.MGR=M.EMPNO;
```

```
4.23  SELECT E.EMPNAME,M.SAL FROM EMP E,EMP M WHERE E.MGR=M.EMPNO;
```

```
4.24  SELECT E.EMPNAME,E.JOB,E.EMPNO,D.DNAME,D.DLOC FROM EMP E, DEPT D WHERE  
E.DEPTNO=E.DEPTNO AND D.DEPTNO=E.DEPTNO;
```

```
4.25  SELECT E.EMPNO,E.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 AND  
E.EMPNAME!=P.EMPNAME;
```



## Question 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;`

## Question 6

--6.1

```
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;
/
```

--6.2

```
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;
/
```

--6.3

```
declare
a number;
b number;
begin
a:=&a;
b:=&b;
if a=b then
dbms_output.put_line('BOTH ARE EQUAL');
```

```

elseif a>b then
dbms_output.put_line('A IS GREATER');
else
dbms_output.put_line('B IS GREATER');
end if;
end;
/

```

```

--6.4
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);
dbms_output.put_line('ENTER THE MARKS');
begin
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');
elseif per>75 then
dbms_output.put_line('GRADE A');
elseif per>65 and per<75 then
dbms_output.put_line('GRADE B');
elseif 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;
/

```

```

--6.5
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 = ' || sum1);
end;
/

```

```

--6.6
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 NUMBER = ' || rev);
end;
/

```

```

--6.7
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;
/

```

```
--6.8
declare
n number;
f number:=1;
begin
n:=&n;
for i in 1..n
loop
f:=f*i;
end loop;
dbms_output.put_line('Factorial ' || n || ' is ' || f);
end;
/
```

```
--6.9
create table areas(radius number(10),area number(6,2));
```

```
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;
/
```

```
--6.10
create table acct(name varchar2(10),cur_bal number(10),acctno number(6,2));
insert into stud values('&sname',&rollno,&marks);
select * from acct;
```

```
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;
/
```

## Question 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;

## Question 8

```
8.1 CREATE OR REPLACE TRIGGER first
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 value);
end if;
end if;
END;
```

```
8.2 CREATE OR REPLACE TRIGGER trgr_to_delete
BEFORE DELETE ON DEPT FOR EACH ROW
DECLARE
CURSOR get_emp( p_deptno NUMBER ) IS
select EMPNO, ENAME , JOB, MGR, SAL, COMM, DOB
from EMP
WHERE deptno=p_deptno;
BEGIN
dbms_output.put_line( 'Delete dept = ' || :old.deptno );
dbms_output.put_line( '- dept name = ' || :old.dname );
dbms_output.put_line( '- dept loc = ' || :old.loc );
FOR get_emp_rec IN get_emp( :old.deptno ) LOOP
dbms_output.put( '- emp ( ' || get_emp_rec.empno );
dbms_output.put( ', ' || get_emp_rec.ename );
dbms_output.put( ', ' || get_emp_rec.job );
dbms_output.put( ', ' || get_emp_rec.mgr );
dbms_output.put( ', ' || get_emp_rec.sal );
dbms_output.put( ', ' || get_emp_rec.comm );
dbms_output.put( ', ' || get_emp_rec.job );
dbms_output.put_line( ' ' );
END LOOP;
END;
```

```
8.3 CREATE TRIGGER AfterDELETEDTrigger after delete on [EMP Table]
FOR DELETE
AS
INSERT INTO [logtable](
empno
,[ename]
,[job]
```

```
,[mgr]
,[deptno]
,[sal]
,[dob])
SELECT empno
,[ename]
,[job]
,[mgr]
,[deptno]
,[sal]
,[dob]
,CAST( SERVERPROPERTY('MachineName') AS VARCHAR2(50))
,CAST( SERVERPROPERTY('ServerName') AS VARCHAR2(50))
,GETDATE()
FROM DELETED;
PRINT 'We Successfully Fired the AFTER DELETE Triggers in SQL Server and inserted a values into log
Table.'
GO
```