MCA SEM-2nd

Section-A

DBMS Lab Assignment

Name-Ayush Singh

Roll no-205119023

**************************exercise 1****************************

1.1 CREATE TABLE EMPLOYEE (EMPNO NUMBER PRIMARY KEY,ENAME VARCHAR(20) NOT NULL,JOB VARCHAR(20) NOT NULL,MGR NUMBER,DEPTNO NUMBER,SAL NUMBER);

1.2 ALTER TABLE EMPLOYEE ADD (COMM NUMBER) ;

1.3 ALTER TABLE EMPLOYEE MODIFY (JOB VARCHAR(30));

1.4 CREATE TABLE DEPARTMENT(DEPTNO NUMBER PRIMARY KEY,DNAME VARCHAR(20),LOC VARCHAR(40));

1.5 ALTER TABLE EMPLOYEE ADD CONSTRAINT F_KEY FOREIGN KEY (DEPTNO) REFERENCES DEPARTMENT(DEPTNO);

1.6 ALTER TABLE EMPLOYEE ADD CONSTRAINT CHECK_CONSTRAINT CHECK (EMPNO>100);

1.7 ALTER TABLE EMPLOYEE modify (sal number default 5000);

1.8 ALTER TABLE EMPLOYEE ADD (DOB VARCHAR(10));

DESC EMPLOYEE;

DESC DEPARTMENT;

****************************exercise2******************************

2.1 INSERT INTO DEPARTMENT VALUES(10, 'MANAGEMENT','MAIN BLOCK');

INSERT INTO DEPARTMENT VALUES(20, 'DEVELOPMENT','MANUFACTURING');

INSERT INTO DEPARTMENT VALUES(30, 'MAINTAINANCE','UNIT MAN BLOCK');

INSERT INTO DEPARTMENT VALUES(40, 'TRANSPORT','ADMIN BLOCK');

INSERT INTO DEPARTMENT VALUES(50, 'SALES','HEAD OFFICE');

2.2 INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7369,'SMITH','CLERK',7566,'17-DEC-80',800,0,20);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7399,'ASANT','SALESMAN',7566,'20-FEB-81',1600,300,20);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7499,'ALLEN','SALESMAN',7698,'20-FEB-81',1600,300,30);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7521,'WARD','SALESMAN',7698,'22-FEB-82',1250,500,30);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7566,'JONES','MANAGER',7839,'02-APR-81',5975,500,20);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7698,'BLAKE','MANAGER',7839,'01-MAY-79',9850,1400,30);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL , DEPTNO) VALUES(7611,'SCOTT','HOD',7839,'12-JUN-76',3000,NULL,10);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB ,DOB ,SAL , DEPTNO) VALUES(7839,'CLARK','CEO','16-MAR-72',9900,NULL,10);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7368,'FORD','SUPERVIS',7366,'17-DEC-80',800,0,20);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7599,'ALLEY','SALESMAN',7698,'20-FEB-81',1600,300,30);

INSERT INTO EMPLOYEE(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7421,'DRANK','CLERCK',7698,'22-JAN-82',1250,500,30);

2.3 UPDATE EMPLOYEE SET COMM=1000 WHERE JOB='MANAGER';

2.4 CREATE TABLE PSEUDOEMPLOYEE AS (SELECT * FROM EMPLOYEE);

2.5 DELETE FROM EMPLOYEE WHERE JOB='SUPERVIS';

2.6 DELETE FROM EMPLOYEE WHERE EMPNO=7599;

2.7 SELECT * FROM EMPLOYEE ORDER BY SAL;

2.8 SELECT * FROM EMPLOYEE ORDER BY SAL DESC;

2.9 SELECT * FROM EMPLOYEE WHERE DEPTNO=30;

2.10 SELECT DISTINCT DEPTNO FROM EMPLOYEE;

2.11 SELECT * FROM EMPLOYEE ORDER BY ENAME;

2.12 create table manager as select * from EMPLOYEE where JOB='MANAGER';

2.13 select * from EMPLOYEE where COMM=NULL ;

2.14 select ENAME,DNAME from EMPLOYEE,DEPARTMENT where EMPLOYEE.DEPTNO=DEPARTMENT.DEPTNO ;

**************************************exercie3***********************

3.1 select * from EMPLOYEE where DEPTNO in(7369,7499);

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 EMPLOYEE;

3.7 SELECT COUNT(*) FROM EMPLOYEE;

3.8 SELECT SUM(SAL),AVG(SAL) FROM EMPLOYEE;

3.9 select min(SAL) "MIN_SAL", MAX(SAL) "MAX_SAL" from EMPLOYEE;

3.10 SELECT SUM(SAL) FROM EMPLOYEE;

3.11 SELECT JOB,SUM(SAL) FROM EMPLOYEE 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 EMPLOYEE;

3.14 select add_months(DOB,2) from EMPLOYEE;

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

3.16 select round(to_date(dob),'month') from employee;

select round(to_date(dob),'year') from employee;

select round(to_date(dob),'day') from employee;

3.17 select(sysdate-60) from dual;

3.18 select ENAME ,SAL , SAL+0.15* SAL from EMPLOYEE;

3.19 select ENAME from EMPLOYEE where ENAME like 'B%' or ENAME like 'C%';

3.20 select ENAME,SAL,MGR from EMPLOYEE where SAL in (select min(SAL) from EMPLOYEE group by MGR);

3.21 select dname, count (ename) from employee, dept where employee.deptno=department.deptno group by dname;

3.22 select ename from employee where length (ename) <=5;

3.23 select ename from employee where mgr in(7602,7566,7789);

3.24 select count (distinct job) from employee;

3.25 select max(sal)-min(sal) from employee;

3.26 select count(distinct deptno) from employee;

3.27 select ename , dob from employee where to_char (dob,'MON')='FEB';

3.28 select ename from employee where to_char(dob,'MON') like to_char (sysdate, 'MON');

3.29 select ENAME from EMPLOYEE where ENAME LIKE ('s%h')

3.30 select ename from employee where sal>5000;

**************************exercise4****************************

4.1 select ENAME,DNAME from EMPLOYEE,DEPARTMENT where DNAME='MAINTAINANCE' OR DNAME='DEVELOPMENT' ;

4.2 SELECT ename,sal FROM employee WHERE sal >(SELECT MIN(saL)FROM employee) AND JOB LIKE ('M%');

4.3 SELECT ename FROM EMPLOYEE WHERE job =(SELECT job FROM employee WHERE eNAME='JONES');

4.4 SELECT * FROM employee WHERE sal >ANY(SELECT sal FROM employee WHERE DEPTNO=30);

4.5 SELECT * FROM EMPLOYEE WHERE job =(SELECT job FROM employee WHERE eNAME='JONES') AND SAL>=(SELECT sal FROM employee WHERE ENAME='FORD');

4.6 SELECT ename, job FROM employee WHERE DEPTNO=20 AND JOB IN(SELECT JOB FROM employee,department WHERE EMPLOYEE.DEPTNO=DEPARTMENT.DEPTNO AND Dname='MANAGEMENT');

4.7 SELECT * FROM employee WHERE sal >(SELECT AVG(SAL)FROM employee);

4.8 SELECT ENAME,JOB,DNAME FROM EMPLOYEE,DEPARTMENT WHERE EMPLOYEE.DEPTNO=DEPARTMENT.DEPTNO;

4.9 SELECT * FROM EMPLOYEE WHERE job in (SELECT job FROM employee,department WHERE employee.deptno=department.deptno and LOC='MAIN BLOCK');

4.10 SELECT * FROM employee WHERE DEPTNO=10 AND JOB IN(SELECT JOB FROM employee,department WHERE EMPLOYEE.DEPTNO=DEPARTMENT.DEPTNO AND Dname='development');

4.11 SELECT * FROM EMPLOYEE WHERE job =(SELECT job FROM employee WHERE eNAME='FORD') AND SAL=(SELECT SAL FROM employee WHERE eNAME='FORD');

4.12 SELECT DNAME,COUNT(*) FROM DEPARTMENT,EMPLOYEE WHERE EMPLOYEE.DEPTNO=DEPARTMENT.DEPTNO AND EMPLOYEE.JOB='SALESMAN' GROUP BY DNAME HAVING COUNT(*)>=2;

4.13 SELECT * FROM empLOYEE WHERE deptno=20 and job=ANY(SELECT job FROM employee WHERE DEPTNO=30);

4.14 SELECT eNAME FROM employee WHERE sal >ANY(SELECT sal FROM employee WHERE DEPTNO IN (20,30));

4.15 select dname,sum(sal) from department,employee where employee.deptno=department.deptno group by dname having sum(sal)>9000;

4.16 select dname,sum(sal) from department,employee where employee.deptno=department.deptno group by dname having sum(sal)>1000 and sum(sal)<5000;

4.17 CREATE TABLE accDEPARTMENT(DEPTNO number(2) PRIMARY KEY,DEPTNAME VARCHAR(20),dcity VARCHAR(40));

alter table accdepARTMENt add constraint fk foreign key(deptno) references department(deptno);

select deptno from department,accdept where department.deptno=accdept.deptno;

4.18SELECT ENAME,DNAME FROM EMPLOYEE,DEPARTMENT WHERE EMPLOYEE.DEPTNO=DEPARTMENT.DEPTNO AND DEPARTMENT.DEPTNO IN(SELECT DNO FROM ACCDEPARTMENT WHERE DEPARTMENT.DEPTNO!=ACCDEPARTMENT.DNO);

4.19 select ename,dname from empLOYEE left join depARTMENt on empLOYEE.deptno=depARTMENt.deptno;

4.20 select ename,dname from empLOYEE right join depARTMENt on empLOYEE.deptno=depARTMENt.deptno;

4.21 select ename,dname from empLOYEE full outer join depARTMENt on empLOYEE.deptno=depARTMENt.deptno;

4.22 SELECT E1.ENAME"EMPLOYEE",E2.ENAME"MANAGER" FROM EMPLOYEE E1 JOIN EMPLOYEE E2 ON(E1.EMPNO=E2.EMPNO);

4.23 SELECT E1.ENAME"EMPLOYEE,E2.SAL"MANAGER'S SALARY FROM EMPLOYEE E1 JOIN EMPLOYEE E2 ON(E1.EMPNO=E2.EMPNO);

4.24 select ename,job,dname,loc from employee natural join department;

*****************************Exercise5****************************
*****

5.1 select DEPTNO from department union select DEPTNO from accdepartment;

5.2 select (DISTINCT DEPTNO) from department union all select (DISTINCT DEPTNO) from accdepartment;

5.3 select DEPTNO from department intersect select DEPTNO from accdepartment;

5.4 select DEPTNO from department minus select DEPTNO from accdepartment;

5.5 create view managers as select * from employee where JOB='MANAGER';

5.6 create view general as select EMPNO,ENAME,DEPTNO,DNAME from Employee NATURAL JOIN department;

5.7 create view all as select EMPNO,ENAME,DEPTNO,DNAME from employee natural join department where job NOT IN('HOD','CEO');

5.8 select * from managers;

select * from general;

select * from all;

5.9 Delete from managers where empno=7566;

update general set ename='Michael' where empno=7399;

5.10 drop view general;

*******************************Exercise6***********************************

6.1 set serveroutput on;

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
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;
/
```

6.3

```
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;
/
```

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);
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');
```

```
end if;
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;
/
```

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 is'|| 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('no is'|| 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('the factorial is'|| f);
end;
/
```

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

PROGRAM:

```
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 SQL> create table acct(name varchar2(10),cur_bal number(10),acctno number(6,2));

```
SQL> insert into stud values('&sname',&rollno,&marks);

SQL> select * from acct;
```

PROGRAM:

```
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;

/
```

****************************Exercise7*************************
*****

```
7.1 create or replace procedure salary(deptid number) as
begin
update employee set sal=sal+1000
where sal>5000 and deptno=deptid;
end;
7.2 create or replace procedure salary1(empid number) as
begin
update employee 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 employee 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 employee 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 department.dname

from department,employee where employee.deptno=department.deptno;

end;

************************************Exercise8************************************

8.1 CREATE OR RELPLACE TRIGGER trig1 before insert on DEPARTMENT 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 DEPARTMENT 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 Afterdelete on DEPARTMENT FOR EACH ROW

BEGIN

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

END;

8.3 CREATE TRIGGER trig3 AFTER DELETE ON employee FOR EACH ROW

BEGIN

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

END;