MASTER OF COMPUTER APPLICATIONS

SEMESTER – 2nd

ROLL NO : 205119067

DATABASE MANAGEMENT SYSTEM LAB MANUAL

Submitted to:

Dr.U. Vignesh Sir

Submitted By:

Pranshu Mishra

Roll No: 205119067

PROBLEM 1.01:

```
CREATE TABLE EMP(EMPNO NUMBER(6),ENAME VARCHAR2(20)  NOT NULL,JOB VARCHAR2(10)  NOT
NULL,MGR NUMBER(4),DEPTNO NUMBER(3),SAL NUMBER(7,2),CONSTRAINT PK_EMP PRIMARY
KEY(EMPNO));
```

PROBLEM 1.02:

```
ALTER TABLE EMP ADD COMMISSION NUMBER(7,2);
```

PROBLEM 1.03:

```
ALTER TABLE EMP MODIFY(JOB VARCHAR2(20));
```

PROBLEM 1.04:

```
CREATE TABLE DEPT(DEPTNO NUMBER(2),DNAME VARCHAR2(10)  NOT NULL,LOC
VARCHAR2(10),CONSTRAINT PK_DEPT PRIMARY KEY(DEPTNO));
```

PROBLEM 1.05:

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

PROBLEM 1.06:

```
ALTER TABLE EMP ADD CONSTRAINT C1 CHECK (EMPNO>100);
```

PROBLEM 1.07:

```
ALTER TABLE EMP MODIFY SAL DEFAULT 5000;
```

PROBLEM 1.08:

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

PROBLEM 2.01:

```
ALTER TABLE DEPT MODIFY(DNAME VARCHAR2(20));
ALTER TABLE DEPT MODIFY(LOC VARCHAR2(20));
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');
```

PROBLEM 2.02:

```
INSERT INTO EMP VALUES(7369,'SMITH','CLERK',7566,20,800,0,'17-DEC-1980');
INSERT INTO EMP VALUES(7399,'ASANT','SALESMAN',7566,20,1600,300,'20-FEB-1981');
INSERT INTO EMP VALUES(7499,'ALLEN','SALESMAN',7698,30,1600,300,'20-FEB-1981');
INSERT INTO EMP VALUES(7521,'WARD','SALESMAN',7698,30,1250,500,'22-FEB-1982');
INSERT INTO EMP VALUES(7566,'JONES','MANAGER',7839,20,5975,500,'02-APR-1981');
INSERT INTO EMP VALUES(7698,'BLAKE','MANAGER',7839,30,9850,1400,'01-MAY-1979');
INSERT INTO EMP VALUES(7611,'SCOTT','HOD',7839, 10,3000,NULL,'12-JUN-1976');
INSERT INTO EMP VALUES(7839,'CLARK','CEO',NULL ,10,9900,NULL,'16-MAR-1972');
INSERT INTO EMP VALUES(7368,'FORD','SUPERVIS',7366,20,800,0,'17-DEC-1980');
INSERT INTO EMP VALUES(7599,'ALLEY','SALESMAN',7698,30,1600,300,'20-FEB-1981');
INSERT INTO EMP VALUES(7421,'DRANK','CLERCK',7698,30,1250,500,'22-JAN-1982');
```

PROBLEM 2.03:

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

PROBLEM 2.04:

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

PROBLEM 2.05:

```
DELETE FROM EMPLOYEE WHERE JOB='SUPERVISOR';
```

PROBLEM 2.06:

```
DELETE FROM EMPLOYEE WHERE EMPNO=7599;
```

PROBLEM 2.07:

```
SELECT * FROM EMPLOYEE ORDER BY SAL;
```

PROBLEM 2.08:

```
SELECT * FROM EMPLOYEE ORDER BY SAL DESC;
```

PROBLEM 2.09:

```
SELECT * FROM EMPLOYEE WHERE DEPTNO=30;
```

PROBLEM 2.10:

```
SELECT DISTINCT DEPTNO FROM EMPLOYEE;
```

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 IS NULL;
```

PROBLEM 2.14:

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

PROBLEM 3.01:

```
SELECT * FROM EMP WHERE DEPTNO=7369 OR DEPTNO=7499;
```

PROBLEM 3.02:

```
SELECT * FROM EMP WHERE SUBSTR(ENAME,1,1) IN ('S');
```

PROBLEM 3.03:
```
SELECT * FROM EMP WHERE SUBSTR(ENAME,1,1) NOT IN ('S');
```

PROBLEM 3.04:
```
SELECT * FROM EMP WHERE EMPNO BETWEEN 7500 AND 7600;
```

PROBLEM 3.05:
```
SELECT * FROM EMP WHERE EMPNO NOT BETWEEN 7500 AND 7600;
```

PROBLEM 3.06:
```
SELECT SQRT(SAL) FROM EMP;
```

PROBLEM 3.07:
```
SELECT COUNT(*) FROM EMP;
```

PROBLEM 3.08:
```
SELECT SUM(SAL),AVG(SAL) FROM EMP;
```

PROBLEM 3.09:
```
SELECT MAX(SAL) AS MAX_SALARY,MIN(SAL) AS MIN_SALARY 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 TO_CHAR(TO_DATE('14-JUL-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('05-OCT-09') FROM DUAL;
```

PROBLEM 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;
```

PROBLEM 3.17:
```
SELECT (SYSDATE-60) FROM DUAL;
```

PROBLEM 3.18:
```
SELECT ENAME,SAL,0.15*SAL AS RAISE FROM EMP;
```

PROBLEM 3.19:
```
SELECT * FROM EMP WHERE SUBSTR(ENAME,1,1) IN ('B','C');
```

PROBLEM 3.20:
```
SELECT ENAME,SAL,MGR FROM EMP WHERE SAL IN (SELECT MIN(SAL) FROM EMP GROUP BY MGR);
```

PROBLEM 3.21:
```
SELECT COUNT(EMPNO),(SELECT DNAME FROM DEPT WHERE DEPT.DEPTNO=EMP.DEPTNO)DNAME FROM
EMP GROUP BY DEPTNO;
```

PROBLEM 3.22:
```
SELECT ENAME FROM EMP WHERE LENGTH(ENAME)<=5;
```

PROBLEM 3.23:
```
SELECT ENAME,MGR FROM EMP WHERE MGR IN(77499,7566,7611);
```

PROBLEM 3.24:
```
SELECT COUNT(DISTINCT(JOB)) FROM EMP;
```

PROBLEM 3.25:
```
SELECT MAX(SAL)-MIN(SAL) FROM EMP;
```

PROBLEM 3.26:
```
SELECT COUNT(DISTINCT(DEPTNO)) FROM EMP;
```

PROBLEM 3.27:
```
SELECT ENAME,DOB FROM EMP WHERE TO_CHAR(DOB,'MM') IN ('02');
```

PROBLEM 3.28:
```
SELECT ENAME FROM EMP WHERE TO_CHAR(DOB,'MM') IN (EXTRACT(MONTH FROM SYSDATE));
```

PROBLEM 3.29:
```
SELECT ENAME FROM EMP WHERE ENAME LIKE 'S%H';
```

PROBLEM 3.30:
```
SELECT ENAME FROM EMP WHERE SAL>6000;
```

PROBLEM 4.01:

SELECT ENAME,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND (DNAME='MAINTAINANCE' OR DNAME='DEVELOPMENT');

PROBLEM 4.02:
SELECT ENAME,SAL FROM EMP WHERE SAL>(SELECT MIN(SAL) FROM EMP) AND JOB LIKE ('M%');

PROBLEM 4.03:
SELECT ENAME FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='JONES') AND ENAME NOT IN ('JONES');

PROBLEM 4.04:
SELECT * FROM EMP WHERE SAL>(SELECT MAX(SAL) FROM EMP WHERE DEPTNO=30);

PROBLEM 4.05:
SELECT ENAME FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='JONES') AND SAL>=(SELECT SAL FROM EMP WHERE ENAME='FORD') AND ENAME NOT IN ('JONES');

PROBLEM 4.06:
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 4.07:
SELECT ENAME,DEPTNO,SAL FROM EMP E1 WHERE SAL > (SELECT AVG(SAL) FROM EMP E2 WHERE E1.DEPTNO=E2.DEPTNO);

PROBLEM 4.08:
SELECT ENAME,JOB,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO;

PROBLEM 4.09:
SELECT ENAME FROM EMP WHERE JOB IN(SELECT JOB FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND LOC='MAIN BLOCK') AND DEPTNO NOT IN (SELECT DEPTNO FROM DEPT WHERE LOC='MAIN BLOCK');

PROBLEM 4.10:
SELECT ENAME FROM EMP WHERE DEPTNO=10 AND JOB IN(SELECT JOB FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND DNAME='DEVELOPMENT');

PROBLEM 4.11:
SELECT ENAME FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='FORD') AND SAL=(SELECT SAL FROM EMP WHERE ENAME='FORD') AND ENAME NOT IN ('FORD');

PROBLEM 4.12:
SELECT DNAME FROM DEPT WHERE (SELECT COUNT(*) FROM EMP WHERE JOB='SALESMAN' AND DEPT.DEPTNO=EMP.DEPTNO) >= 2;
PROBLEM 4.13:

```
SELECT ENAME FROM EMP WHERE DEPTNO=20 AND JOB IN(SELECT JOB FROM EMP WHERE DEPTNO=30);
```

PROBLEM 4.14:

```
SELECT ENAME FROM EMP WHERE SAL>(SELECT MAX(SAL) FROM EMP WHERE DEPTNO=20 OR
DEPTNO=30);
```

PROBLEM 4.15:

```
SELECT MAX(SAL),DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND SAL > 9000 GROUP
BY DNAME;
```

PROBLEM 4.16:

```
SELECT MAX(SAL),DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO HAVING MIN(SAL)>1000
AND MIN(SAL)<5000 GROUP BY DNAME;
```

PROBLEM 4.17:

```
CREATE TABLE ACCDEPT AS SELECT * FROM DEPT WHERE DEPTNO IN (10,20,30);
SELECT DEPT.DNAME FROM DEPT,ACCDEPT WHERE DEPT.DEPTNO=ACCDEPT.DEPTNO;
```

PROBLEM 4.18:

```
SELECT ENAME FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME NOT IN
(SELECT DEPT.DNAME FROM DEPT,ACCDEPT WHERE DEPT.DEPTNO=ACCDEPT.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 RIGHT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;
```

PROBLEM 4.21:

```
SELECT ENAME,DNAME FROM EMP FULL OUTER JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;
```

PROBLEM 4.22:

```
SELECT A.ENAME AS EMPLOYEE,B.ENAME AS MANAGER FROM EMP A,EMP B WHERE A.MGR=B.EMPNO;
```

PROBLEM 4.23:

```
SELECT A.ENAME AS EMPLOYEE,B.SAL AS MANAGER_SALARY FROM EMP A,EMP B WHERE
A.MGR=B.EMPNO;
```

PROBLEM 4.24:

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

PROBLEM 4.25:

```
SELECT A.EMPNO,A.ENAME AS EMPLOYEE,A.JOB,B.ENAME AS MANAGER FROM EMP A,EMP B WHERE
A.MGR=B.EMPNO;
```

PROBLEM 4.26:

```
SELECT ENAME,SAL FROM EMP WHERE SAL IN (SELECT SAL FROM EMP GROUP BY SAL HAVING
COUNT(*)>1);
```

PROBLEM 5.01:

```
SELECT DEPTNO FROM DEPT UNION SELECT DEPTNO FROM ACCDEPT;
```

PROBLEM 5.02:

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

PROBLEM 5.03:

```
SELECT DEPTNO FROM DEPT INTERSECT SELECT DEPTNO FROM ACCDEPT;
```

PROBLEM 5.04:

```
SELECT DEPTNO FROM DEPT MINUS SELECT DEPTNO FROM ACCDEPT;
```

PROBLEM 5.05:

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

PROBLEM 5.06:

```
 CREATE VIEW GENERALS AS SELECT EMP.EMPNO, EMP.ENAME, EMP.DEPTNO, DEPT.DNAME FROM EMP
JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;
```

PROBLEM 5.07:

```
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;
```

PROBLEM 5.08:

```
SELECT VIEW_NAME FROM USER_VIEWS;
```

PROBLEM 5.09:

```
DROP VIEW EMP_ALL;
```

PROBLEM 5.10:

```
DROP VIEW ALL;
```

PROBLEM 6.01:

```
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 6.02:

```
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 6.03:

```
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;
/
```

PROBLEM 6.04:

```
declare
java number(10);
dbmsnumber(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');
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>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;
/
```

PROBLEM 6.05:

```
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;
/
```

PROBLEM 6.06:

```
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;
/
```

PROBLEM 6.07:

```
declare
a number;
c number:=0;
inumber;
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;
/
```

PROBLEM 6.08:

```
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;
/
```

PROBLEM 6.09:

```
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;
/
```

PROBLEM 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);
mcbnumber(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;
/
```

PROBLEM 7.01:

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

PROBLEM 7.02:

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

PROBLEM 7.03:

```
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;
/
```

PROBLEM 7.04:

```
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;
/
```

PROBLEM 7.05:

```
create or replace procedure dep_name(deptid number)  as
```

```
begin
select dept.dname from dept,emp where emp.deptno=dept.deptno;
end;
/
```

PROBLEM 8.01:

```
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;
/
```

PROBLEM 8.02:

```
CREATE [OR REPLACE] TRIGGER trig2 Afterdelete on DEPT FOR EACH ROW
BEGIN
DELETE FROM emp WHERE emp.deptno=:new.deptno;
END;
/
```

PROBLEM 8.03:

```
CREATE TRIGGER trig3 AFTER DELETE ON emp FOR EACH ROW
BEGIN
INSERT INTO log(val1, val2, ...) VALUES (old.val1, old.val2, ...);
END;
/
```