NATIONAL INSTITUTE OF TECHNOLOGY,

TIRUCHIRAPALLI, TAMIL NADU

DBMS LAB MANUALS

**Submitted by:-**

NAME: - GOURAV KUMAR

ROLL NO.: - 205119033

SUBJECT: - CA702 - DBMS LAB MANUAL

SEMESTER: - 2nd SEMESTER

COURSE: - MASTER OF COMPUTER APPLICATIONS

DEPT.: - DEPARTMENT OF COMPUTER APPLICATIONS

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr No.** | **List of experiments** | **Remarks** |
| **1.** | Data Definition, Table Creation, Constraints. |  |
| **2.** | Insert, Select Commands, Update & Delete Commands. |  |
| **3.** | Inbuilt functions in RDBMS. |  |
| **4.** | Nested Queries & Join Queries. |  |
| **5.** | Set operators & Views. |  |
| **6.** | Control structures. |  |
| **7.** | Procedures and Functions. |  |
| **8.** | Triggers |  |

1. Data definition, Table creation and constraints

1.1 :- CREATE TABLE EMP (EMPNO INTEGER PRIMARY KEY,ENAME VARCHAR(20) NOT NULL,JOB VARCHAR(20) NOT NULL,MGR INTEGER,DEPTNO INTEGER,SAL INTEGER);

1.2 :- ALTER TABLE EMP ADD COMM INTEGER ;

1.3 :- ALTER TABLE EMP MODIFY JOB VARCHAR(30);

1.4 :-CREATE TABLE DEPT(DEPTNO INTEGER PRIMARY KEY,DNAME VARCHAR(20),LOC VARCHAR(40));

1.5:- ALTER TABLE EMP ADD FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO);

1.6 :- ALTER TABLE EMP ADD CHECK (EMPNO>100);

* 1. :- ALTER TABLE EMP modify sal integer default 5000;

1.8 :- ALTER TABLE EMP ADD DOB VARCHAR(10);

1. **Insert, Select Commands, Update & Delete Commands.**

2.1 :-

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

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

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

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

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

2.2 :-

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

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

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

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

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

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

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

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

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

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

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

2.3:-

UPDATE EMP SET COMM=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));

2.5 :- DELETE FROM EMPLOYEE WHERE JOB='SUPERVISOR';

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 ENAME;

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

2.13 :- SELECT \* FROM EMP WHERE COMM=NULL ;

2.14 :- SELECT ENAME,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO ;

**3.Inbuilt functions in RDBMS.**

3.1 :- SELECT \* FROM EMP 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 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(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 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 ENAME 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 FROML PROGRAMMER WHERE TO\_CHAR(DOB,’MON’) LIKE TO\_CHAR(SYSDATE,’MON’);

* 1. :- SELECT ENAME FROM EMP WHERE ENAME LIKE ('S%') AND ENAME LIKE('%H');
  2. :- SELECT ENAME FROM EMP WHERE SAL>5000 OR SAL>6000;

4. **Nested Queries And Joins in RDMS**

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='SALEMAN' GROUP BY DEPTNO) WHERE NO>1) ;

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 HAVING MAX(SAL)>9000 GROUP BY DEPTNO;

4.16 :- SELECT MAX(SAL) FROM EMP HAVING (MIN(SAL)>1000 AND MIN(SAL)<5000) GROUP BY DEPTNO;

4.17:- SELECT DEPT.DNAME FROM DEPT JOIN ACCDEPT ON DEPT.DEPTNO=ACCDEPT.DEPTNO;

4.18 :- SELECT ENAME FROM EMP JOIN ACCDEPT ON EMP.DEPTNO=ACCDEPT.DEPTNO WHERE EMP.DEPTNO!=ACCDEPT.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 E1.ENAME AS EMPNAME,E2.ENAME AS MGRNAME FROM EMP E1, EMP E2 WHERE E1.MGR=E2.ENMPNO;

4.23 :-

4.24 :-

4.25 :-

4.26 :-

**5. Set operators and Views in RDMS**

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 ENAME FROM EMP WHERE JOB='MANAGER';

SELECT \* FROM MANAGERS;

5.6 :- CREATE VIEW GENERAL AS SELECT ENMPNO,ENAME,EMP.DEPTNO,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO;

SELECT \* FROM GENERAL;

5.7 :- CREATE VIEW ALLVIEW AS SELECT ENMPNO,ENAME,EMP.DEPTNO,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND JOB!='CEO' AND JOB!='HOD';

SELECT \* FROM ALLVIEW;

5.8

5.9 :- SELECT \* FROM GENERAL;

5.10 :- DROP VIEW MANAGERS;

**6. Control Structures**

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

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(10);

b number(10);

begin

dbms\_output.put\_line('Enter values of a and b')

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

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');

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;

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;

**7.** **Procedures and Functions.**

7.1 :-

7.2 :-

create or replace procedure salary(empid number) as

begin

update emp set sal=sal+sal\*(0.1) where empno=empid;

end;

/

exec salary(7521);

7.3 :-

create or replace procedure get\_salary(dept number) as

begin

for s in (select \* from emp where deptno = dept)

loop

dbms\_output.put\_line(s.sal);

end loop;

end;

/

exec get\_salary(20);

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;

/

exec get\_nature(20);

7.5 :-

create or replace procedure dep\_name(dept number)

as

begin

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

end;

/

exec dep\_name(30);

8. **Triggers**

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;