**Name :-YASH KARIL**

**Roll No.:-205119111**

**Subject :- DBMS LAB MANUAL**

**1. Data Definition Language (DDL) commands.**

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));
2. ALTER TABLE EMP ADD COMMI NUMBER(5);
3. ALTER TABLE EMP MODIFY JOB VARCHAR(12);
4. CREATE TABLE DEPT (DEPTNO NUMBER(5) PRIMARY KEY,DNAME VARCHAR(20),DLOC VARCHAR(20));
5. ALTER TABLE EMP MODIFY EMPNO INT PRIMARY KEY;

ALTER TABLE DEPT ADD FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO;

1. ALTER TABLE EMP ADD CHECK(EMPNO>100);
2. ALTER TABLE EMP MODIFY SAL FLOAT DEFAULT 5000 NOT NULL;
3. ALTER TABLE EMP ADD DOB DATE;

**2. Data Manipulation Language (DML) commands.**

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');

1. 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);

1. UPDATE EMP SET COMMI=1000 WHERE JOB='MANAGER';
2. 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;

1. DELETE FROM EMPLOYEE WHERE JOB='SUPERVIS';
2. DELETE FROM EMPLOYEE WHERE EMPNO=7599;
3. SELECT \* FROM EMP ORDER BY SAL;
4. SELECT \* FROM EMP ORDER BY SAL DESC;
5. SELECT \* FROM EMP WHERE DEPTNO=30;
6. SELECT DISTINCT DEPTNO FROM EMP;
7. SELECT \* FROM EMP ORDER BY EMPNAME;
8. CREATE TABLE MANAGER AS SELECT \* FROM EMP WHERE JOB='MANAGER';
9. SELECT \* FROM EMP WHERE COMMI IS NULL;
10. SELECT E.ENAME,D.DNAMEFROM EMP E,DEPT D WHERE E.DEPTNO = D.DEPTNO;

**3. In Built functions.**

1. select \* from EMP where DEPTNO in (20,10);
2. select \* from EMPLOYEE where ENAME like 'S%';
3. select \* from EMPLOYEE where ENAME not like 'S%';
4. select \* from EMPLOYEE where EMPNO between 7500 and 7600 ;
5. Select \* from EMPLOYEE where EMPNO not between 7500 and 7600 ;
6. select sqrt(SAL) from EMP;
7. select COUNT(\*) from EMP;
8. select SUM(SAL),AVG(SAL) from EMP;
9. select min(SAL) AS MIN\_SAL, MAX(SAL) AS MAX\_SAL from EMP;
10. select SUM(SAL) from EMP;
11. select JOB,SUM(SAL) from EMP GROUP BY JOB;
12. select to\_char(to\_date('14-jul-09'),'month') from dual;
13. select to\_date(DOB,'DD-MM-YY') from EMP;
14. select add\_months(DOB,2) from EMP;
15. select last\_day('05-oct-09') from dual;
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;

1. select(sysdate-60) from dual;
2. select EMPNAME ,SAL , SAL+0.15\* SAL from EMP;
3. select EMPNAME from EMP where Empname like 'B%' or Empname like 'C%';
4. select EMPNAME,SAL,MGR from EMP where SAL in (select min(SAL) from EMP group by MGR);
5. select dname, count (empname) from emp, dept where emp.deptno=dept.deptno group by dname;
6. select empname from emp where length(empname)<=5;
7. select empname from emp where mgr in (7399,7698,7566);
8. select count(DISTINCT JOB ) from EMP;
9. select max(sal)-min(sal) from emp;
10. select count(distinct deptno) from emp;
11. select empname,dob from emp where to\_char(dob,'mon')='feb';
12. select EMPname from EMP where to\_char(dob,'MON') like to\_char(sysdate, 'MON');
13. select empname from emp where empname like 'S%H';
14. select empname from emp where sal>5000;

**4. Nested Quries & Joins in RDBMS.**

1. SELECT \* FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND (DEPT.DNAME='MAINTAINANCE' OR DEPT.DNAME='DEVELOPMENT');
2. SELECT EMPNAME,SAL FROM EMP WHERE SAL>(SELECT MIN(SAL) FROM EMP) AND JOB LIKE 'M%';
3. SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNAME='JONES');
4. SELECT \* FROM EMP WHERE SAL>(SELECT MAX(SAL) FROM EMP WHERE DEPTNO=30);
5. SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNAME='JONES') AND SAL>=(SELECT SAL FROM EMP WHERE EMPNAME='FORD');
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');
7. SELECT \* FROM EMP OUTER WHERE SAL>(SELECT AVG(SAL) FROM EMP WHERE DEPTNO=OUTER.DEPTNO);
8. SELECT EMPNAME,JOB,DNAME FROM EMP E,DEPT D WHERE E.DEPTNO=D.DEPTNO;
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');
10. SELECT \* FROM EMP WHERE DEPTNO=10 AND JOB=ANY(SELECT JOB FROM EMP,DEPT WHERE DEPT.DEPTNO=EMP.DEPTNO AND DEPT.DNAME='DEVELOPMENT');
11. SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNAME='FORD') AND SAL=(SELECT SAL FROM EMP WHERE EMPNAME='FORD');
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);

1. SELECT \* FROM EMP WHERE DEPTNO=20 AND JOB=ANY(SELECT JOB FROM EMP WHERE DEPTNO=30);
2. SELECT \* FROM EMP WHERE SAL>ANY(SELECT MAX(SAL) FROM EMP WHERE DEPTNO=20 OR DEPTNO=30 GROUP BY DEPTNO);
3. SELECT MAX(SAL) FROM EMP GROUP BY DEPTNO HAVING MAX(SAL)>9000;
4. 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;

1. SELECT A.DNAME FROM DEPT D,ACCDEPT A WHERE D.DEPTNO=A.DEPTNO;
2. SELECT EMPNAME FROM EMP WHERE DEPTNO!=ANY(SELECT DEPTNO FROM ACCDEPT);
3. SELECT \* FROM EMP LEFT JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;
4. SELECT \* FROM EMP RIGHT JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;
5. SELECT \* FROM EMP FULL JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;
6. SELECT E.EMPNAME,M.EMPNAME FROM EMP E,EMP M WHERE E.MGR=M.EMPNO;
7. SELECT E.EMPNAME,M.SAL FROM EMP E,EMP M WHERE E.MGR=M.EMPNO;
8. 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;
9. SELECT E.EMPNO,E.EMPNAME,E.JOB,M.EMPNAME FROM EMP E,EMP M WHERE E.MGR=M.EMPNO;
10. SELECT E.EMPNAME,P.EMPNAME FROM EMP E,EMP P WHERE E.SAL=P.SAL AND E.EMPNAME!=P.EMPNAME;

**5. Set Operators & View in RDBMS.**

1. SELECT DEPTNO FROM DEPT UNION SELECT DEPTNO FROM ACCDEPT;
2. SELECT DEPTNO FROM DEPT UNION ALL SELECT DEPTNO FROM ACCDEPT;
3. SELECT DEPTNO FROM DEPT INTERSECT SELECT DEPTNO FROM ACCDEPT;
4. SELECT DEPTNO FROM DEPT MINUS SELECT DEPTNO FROM ACCDEPT;
5. CREATE VIEW MANAGERS AS SELECT \* FROM EMP WHERE JOB='MANAGER';

SELECT \* FROM MANAGERS;

1. CREATE VIEW GENERAL AS SELECT EMPNO,ENAME,EMP.DEPTNO,DNAME FROM EMP, DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO;
2. 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;

1. SELECT VIEW\_NAME FROM USER\_VIEWS;
2. INSERT INTO MANAGER VALUES(720,’CLAVE’,20,’COMPUTER’);
3. DROP VIEW EMP\_ALL;

**6. Control Structures.**

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;

/

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;

/

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

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: write a pl/sql program to find the total and average of 6 subjects and display

the grade.

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: 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 = '|| sum1);

end;

/

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 NUMBER = '|| rev);

end;

/

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 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: 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('Factorial '|| n ||' is '|| f);

end;

/

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.

Store the radius and the corresponding values of calculated area in an empty table named

areas ,consisting of two columns radius & area.

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: write a pl/sql code block that will accept an account number from the

user,check if the users balance is less than minimum balance,only then deduct rs.100/- from

the balance.this process is fired on the acct table.

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:

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;

/

**8. Triggers**

8.1:

Write a trigger to ensure that DEPT TABLE does not contain duplicate of null values in DEPTNO column

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:

Write a trigger to carry out the following action: on deleting a deptno from DEPT table, all the records with that

DEPTNO has to be deleted from the EMP table

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:

Write a Trigger to carry out the following action: on deleting any records from the emp table,the same values must

be inserted into the log table.

CREATE TRIGGER AfterDELETETrigger 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

/