MCA SEM-1
DBMS LAB PROGRAMS

1. **Create a table called Employee with the following structure**.

Name	Type
Empno	Number
Ename	Varchar2(20)
Job	Varchar2(20)
Mgr	Number
Sal	Number

a. **Add a column commission with domain to the Employee table**
b. **Insert any five records into the table.**
c. **Update the column details of job**
d. **Rename the column of Employ table using alter command.**
e. **Delete the employee whose empno is19.**

1.**Create a table called Employee**

```
Sql>createtable Employee (Empnonumber,Ename varchar2(20),job varchar2(20), Mgrnumber,Sal number);
Sql> Select * from Employee;
```

a.**Add a column commission with domain to the Employee table**

```
Sql> Altertable employee add commission number;
```

b.**Insert any five records into the table**

```
Sql> INSERT INTO Employee VALUES (1, 'King', 'ITmanager', '100', '20000' );
Sql> INSERT INTO Employee VALUES (5, 'blake', 'IT', '200', '30000' );
Sql> INSERT INTO Employee VALUES (9, 'raj', 'manager', '300', '40000' );
Sql> INSERT INTO Employee VALUES (19, 'clarke', 'Assistant', '400', '50000' );
Sql> INSERT INTO Employee VALUES (25, 'mohan', 'clerk', '500', '60000' );
```

Output:

Empno	Ename	Job	Mgr	Sal
1	King	ITmanager	100	20000
5	blake	IT	200	30000
9	raj	manager	300	40000
19	clarke	Assistant	400	50000
25	mohan	clerk	500	60000

c**. Update the column details of job**

```
Sql> UPDATE EMPLOYEE SET JOB = 'MANAGER' WHERE JOB IS NULL;
```

d.**Rename the column of Employ table using alter command**

```
Sql>ALTER TABLE Employee RENAME COLUMN Ename TO Employname;
```

e.**Delete the employee whose empno is19**

```
Sql>DELETEempno FROM Employee WHERE empno=19;
```

output:

Empno	Ename	Job	Mgr	Sal
1	King	'ITmanager'	100	20000
5	blake	IT	200	30000
9	raj	manager	300	40000
25	mohan	clerk	500	60000

1**. Createdepartmenttablewiththefollowingstructure.**

Name	Type
Deptno	Number
Deptname	Varchar2(20)
Location	Varchar2(20)

a. Add column designation to the department table.
b. Insert values into the table
c. List the records of emp table grouped by dept no.
d. Update the record where dept no is9.
e. Delete any column data from the table

1. **Create department table**

```
Sql>CREATE TABLE dept (Deptnonumber,Deptname varchar2(20), location varchar2(20) );
```

a.**Add column designation to the department table.**

```
Sql>Altertabledept add designation varchar2(20);
```

b.**Insert values into the table.**

```
Sql> insert into deptvalues(101, „cse‟, „ nellore‟, „assistant‟);
Sql> insert into deptvalues(102, „Ece‟, „ tpty‟, „assistant‟);
Sql> insert into deptvalues(103, „eee‟, „banglore‟, „HR‟);
Sql> insert into deptvalues(104, „civil‟, „Hyd‟, „manager‟);
Sql> insert into deptvalues(101, „cse‟, „ chittoor‟, „assistant‟);
```

Output:

Deptno	Deptname	Location	designation
101	CSC	nellore	Assistant
102	Ece	Tpty	Assistant
103	eee	Bangalore	HR
104	Civil	Hyd	Manager
101	Csc	Chittoor	Assistant

c.**List the records of emp table grouped by dept no.**

```
Sql>SELECT empno from emp, dept , GROUP BY deptno;
```

d**. Update the record where dept no is9.**

```
Sql> Update table dept set deptno=9 where location= „tpty‟;
```

Output:

e.**Delete any column data from the table**

```
Sql>DELETE location FROM dept;
```

Output:

Deptno	Deptname	designation
101	CSC	Assistant
102	Ece	Assistant
103	eee	HR
104	Civil	Manager
101	Csc	Assistant

QUERIES USING DDL AND DML

I)

a.**Create a user and grant all permissions to the user**

```
Sql>create user <user-name> for login <login-name>
GRANT ALL to<user-name>
```

b.**Insert the any three records in the employee table and use rollback. Check the result.**

```
Sql> INSERT INTO Employee VALUES (1, 'King', 'ITmanager', '100', '20000' );
Sql> INSERT INTO Employee VALUES (5, 'blake', 'IT', '200', '30000' );
Sql> INSERT INTO Employee VALUES (9, 'raj', 'manager', '300', '40000' );
ROLLBACK;

SELECT * FROM Employee;
```

c.**Add primary key constraint and not null constraint to the employee table**

```
Sql>createtable Employee
 (EmpnonumberPRIMARY KEY,Ename varchar2(20) NOT NULL,job varchar2(20), Mgrnumber,Sal
number);
```

d**. Insert null values to the employee table and verify the result.**

```
Sql> INSERT INTO Employee VALUES (1, 'King', null, '100', '20000' );
        INSERT INTO Employee VALUES (5, 'blake', 'IT ,null , '30000' );
```

Output:

Empno	Ename	Job	Mgr	Sal
1	King	NULL	100	20000
5	blake	IT	NULL	30000

II)

a.**Insert values in the department table and use commit.**

```
Sql> insert into deptvalues(101, „cse‟, „ nellore‟, „assistant‟);
Sql> insert into deptvalues(102, „Ece‟, „ tpty‟, „assistant‟);
Sql> insert into deptvalues(103, „eee‟, „banglore‟, „HR‟);
Sql> insert into deptvalues(104, „civil‟, „Hyd‟, „manager‟);
Sql> insert into deptvalues(101, „cse‟, „ chittoor‟, „assistant‟);

Select * from dept where Deptno=101;
Sql>COMMIT;
```

Output:

Deptno	Deptname	Location	designation
101	CSC	nellore	Assistant
101	Csc	Chittoor	Assistant

b.**Add constraints like unique and not null to the department table**.

```
Sql>CREATE TABLE dept (Deptno number NOT NULL UNIQUE,Deptname varchar2(20), location
varchar2(20) );
```

c.**Delete any three records in the department table**.

```
Sql>delete from dept where designation = ‘Assistant’;
```

Deptno	Deptname	Location	designation
103	eee	Bangalore	HR
104	Civil	Hyd	Manager

d. **Add constraint primary key and foreign key to the table**.

```
Sql>createtable Employee
 (Empnonumber PRIMARY KEY,Ename varchar2(20) NOT NULL,job varchar2(20), Mgrnumber,Sal number);
```

Empno	Ename	Job	Mgr	Sal
1	King	'ITmanager'	100	20000
5	blake	IT	200	30000
9	raj	manager	300	40000
25	mohan	clerk	500	60000

```
Sql>CREATE TABLE dept (Deptno number NOT NULLPRIMARY KEY ,Deptname varchar2(20), location varchar2(20), Empno number FOREIGN KEY REFERENCESEmployee(Empno));
```

Deptno	Deptname	Location	designation	Empno
101	CSC	nellore	Assistant	1
102	Ece	Tpty	Assistant	5
103	eee	Bangalore	HR	9
104	Civil	Hyd	Manager	25

QUERIES USING AGGREGATE FUNCTIONS

a.**Display lowest paid employee details under each department**.

```
Sql>select ename,sal,job from employee e, dept d
where e.depid=d.depid and sal in
(SELECT MIN(sal) FROM employee GROUP BY depid)
```

b.**Display number of employees working in each department and their department number**

```
Sql>select depid,count(*) as Noof_mployees from employee group by depid;
```

Output:

depid	Noofemployees
1	2
2	2
3	3

c.**List all employees which start with either B or C.**

```
Sql>select * from employee where ename like 'B%' or ename like 'C%';
```

Output:

Empno	Ename	Job	Mgr	Sal
5	blake	IT	200	30000
19	clarke	Assistant	400	50000

d.**Display only these ename of employees where the maximum salary is greater than or equal to 30000**

```
Sql>select ename,sal from employee where sal>=30000
```

Output:

Ename	Sal
blake	30000
raj	40000
clarke	50000
mohan	60000

e.List all employee names, salary and 15% rise in salary.

```
Sql>select ename,sal,(sal*15/100) as increasedsal from employee;
```

Output:

Ename	Sal	increasedsal
King	20000	3000
blake	30000	4500
raj	40000	6000
clarke	50000	7500
mohan	60000	9000

f.**Count the number of employees in department20**

Sql>select * from employee where depid=20;

g.**Find minimum, maximum, average salary of all employees.**
Sql>select min(sal) as Minsal, max(sal) as Maxsal, avg(sal) as Avgsal from employee;

Minsal	Maxsal	Avgsal
20000	50000	32142

h.**List the employee names in descending order.**
Sql>select * from employee order by enamedesc;

i.**List the employee id, names in ascending order by empid.**
Sql>select empno,ename from employee order by empnoasc.

PROGRAMS ON PL/SQL

1.**Write a PL/SQL program to swap two numbers**

```
declare
-- declare variable num1, num2
-- and temp of datatype number
   num1 number;
   num2 number;
   temp number;
begin
   num1:=1000;
   num2:=2000;
-- print result before swapping
dbms_output.put_line('before');
dbms_output.put_line('num1 = '|| num1 ||' num2 = '|| num2);

-- swapping of numbers num1 and num2
temp := num1;
   num1 := num2;
   num2 := temp;
 -- print result after swapping
dbms_output.put_line('after');
dbms_output.put_line('num1 = '|| num1 ||' num2 = '|| num2);
end;
```

Output:

```
before
num1 = 1000 num2 = 2000
after
num1 = 2000 num2 = 1000
```

2.Write a PL/SQL program to find the largest of three numbers.

```
declare
a number:=10;
b number:=12;
c number:=5;
begin
dbms_output.put_line('a='||a||' b='||b||' c='||c);
if a>b AND a>c
then
dbms_output.put_line('a is greatest');
else
if b>a AND b>c
then
dbms_output.put_line('b is greatest');
else
dbms_output.put_line('c is greatest');
end if;
end if;
end;
/
```

Output:

```
a=10 b=12 c=5
b is greatest
```

3.Write a PL/SQL program to find the total and average of 6 subjects and display the grade.

```
DECLARE
total_marksNUMBER := 0;
average_marksNUMBER := 0;
   grade VARCHAR2(2);
BEGIN
   -- Calculate total marks
total_marks := 85 + 90 + 78 + 92 + 88+75;
   -- Calculate average marks
average_marks := total_marks / 6;
   -- Determine grade based on average marks
   IF average_marks>= 90 THEN
grade := 'A+';
   ELSIF average_marks>= 80 THEN
grade := 'A';
   ELSIF average_marks>= 70 THEN
grade := 'B';
   ELSIF average_marks>= 60 THEN
grade := 'C';
   ELSE
grade := 'F';
   END IF;
   -- Display results
   DBMS_OUTPUT.PUT_LINE('Total Marks: ' || total_marks);
   DBMS_OUTPUT.PUT_LINE('Average Marks: ' || average_marks);
   DBMS_OUTPUT.PUT_LINE('Grade: ' || grade);
END;
```

Output:

```
Total Marks:508
Average Marks:84.66
Grade:A
```

4.Write a PL/SQL program to find the sum of digits in a given umber.

```
DECLARE
--Declare variable n, temp_sumand r of datatype number
    n        INTEGER;
temp_sum INTEGER;
    r        INTEGER;
BEGIN
n := 123456;
temp_sum := 0;
    -- here we check condition with the help of while loop
    -- here <> symbol represent for not null
    WHILE n <> 0
 LOOP
r := MOD(n, 10);
temp_sum := temp_sum + r;
n := Trunc(n / 10);
END LOOP;
dbms_output.Put_line('sum of digits = ' || temp_sum);
END;
-- Program End
```

Output:

```
sum of digits = 21
```

5.**Write a PL/SQL program to display the number in reverse order.**

```
SET SERVEROUTPUT ON;
DECLARE
num NUMBER;
rev NUMBER;

BEGIN
-- & is used to read input from keyboard
num:=#
-- initialize rev to 0
rev:=0;
WHILE num>0 LOOP
-- mod function is used to find the modulus/ remainder of num when divided by 10
rev:=(rev*10) + mod(num,10);
-- floor function is used to obtain a result which is an integer
num:=floor(num/10);
END LOOP;
dbms_output.Put_line ('Reverse of the number is: ' || rev);
END;
/
```

Output:

```
Enter value for num : 157439
Reverse of the number is: 934751
```

6.**Write a PL/SQLprogram to check whether the given number is prime or not**

```
declare
n number;
i number;
temp number;
```

```
begin
n := 13;
i := 2;
temp := 1;

-- loop from i = 2 to n/2
  for i in 2..n/2
    loop
        if mod(n, i) = 0
        then
temp := 0;
            exit;
        end if;
    end loop;

    if temp = 1
    then
dbms_output.put_line(num||‘is a prime number');
    else
dbms_output.put_line(num||' is not a prime number');
    end if;
end;
```

Output:

```
13 is a prime number
```

7.**Write a PL/SQL program to find the factorial of a givennumber**

```
declare
n number;
fact number := 1;
inumber := 1;
c number := 0;
begin
/* take input from the user */
n:=&n;
if(n<0)
 then
 /* display error when a user entered a negative number */
dbms_output.put_line('factorial of negative number does not exist');
end if;
for i in 1..n
loop
fact := fact * i;
end loop;
dbms_output.put_line('factorial of '||n||' is '||fact);
end;
/
```

Output:

```
Enter value for n:5
Factorial of 5 is 120
```