```
select Lname, Dname, Salary
from EMPLOYEE, DEPARTMENT
where Dno = Dnumber and Dname = 'Research'
order by Lname;
```

10. Select all staff members SSN, Fname, DepartmentName, Salary in ascending order by their Depatment, then by their salary in Descending order:

```
select Ssn, Fname, Dname , Salary
from DEPARTMENT, EMPLOYEE
where Dno = Dnumber
order by Dname ASC, Salary DESC;
```

11. What is the name of the department with the highest department number?

```
SELECT Dname , Dnumber
FROM DEPARTMENT
ORDER BY Dnumber DESC LIMIT 1;
```

12. Retrieve a list of employees and the projects they are working on, ordered by department and, within each department, ordered alphabetically by last name, then first name

```
SELECT D.Dname, E.Lname, E.Fname, P.Pname
FROM DEPARTMENT D, EMPLOYEE E, WORKS_ON W, PROJECT P
WHERE  D.Dnumber= E.Dno AND E.Ssn= W.Essn AND W.Pno= P.Pnumber
ORDER BY D.Dname, E.Lname, E.Fname;
```

RESULT: Successfully executed the queries using SQL DML Commands.

ExpNo:8

NESTED QUERIES , JOIN QUERIES AND SET OPERATORS

AIM: To perform nested Queries , joining Queries and set operations using DML command

QUERIES

1. Display all employee names and salary whose salary is greater than minimum salary of the company

```
select Fname,Lname,Salary
from EMPLOYEE
where Salary>(select min(Salary) from  EMPLOYEE);
```

2. Issue a query to display information about employees who earn more than any employee in dept no 5

```
select * from EMPLOYEE
where Salary>(select min(Salary) from EMPLOYEE where  Dno=5);
```

3. Display the details of those who draw the salary greater than the average salary.

```
select distinct *
from EMPLOYEE
where Salary >= (select avg(Salary) from  EMPLOYEE);
```

4. Write SQL Query which retrieves the name and address of every employee who works for the Research Department

```
select Fname, Lname, Address
from EMPLOYEE, DEPARTMENT
where Dno = Dnumber and Dname = 'Research';
```

5. Retrieve the name of each employee who has a dependent with the same first name and is the same sex as the employee.

```
Select E.Fname, E.Lname
From EMPLOYEE as E
where E.Ssn in ( Select Essn From  DEPENDENT as D where
E.Fname=D.Dependent_Name and E.Sex=D.Sex );
```

6. Make a list of all project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.

```
select Pno from WORKS_ON, EMPLOYEE
where Essn = Ssn and Lname = 'Smith'
UNION
select Pnumber from PROJECT P, DEPARTMENT D, EMPLOYEE E where
P.Dnum = D.Dnumber and D.Mgr_ssn = E.Ssn and E.Lname = 'Smith';
```

7. Write a query to display the name for all employees who work in a department with any employee whose Fname contains the letter 'h'

```
Select Fname from EMPLOYEE where Dno IN (Select Dno from EMPLOYEE
```

where Fname LIKE '%h%');

8 Retrieve all employees whose address Starts with Houston.

select Fname, Lname, Address
from EMPLOYEE
where Address LIKE 'Houston%';

9. Retrieve all employees whose address is Ends with Houston..

select Fname, Lname, Address
from EMPLOYEE
where Address LIKE '%Houston';

10. Find all employees who were born during the 1960s.

select Fname, Lname
from EMPLOYEE
where Bdate LIKE '__6_____';

11. Retrieve all employees in department 5 whose salary is between $30,000 and $40,000.

 # This is the use of in between

*SELECT * FROM EMPLOYEE*
WHERE (Salary BETWEEN 30000 AND 40000) AND Dno = 5;

 # this is euquqlent to <= and > =

*SELECT * FROM EMPLOYEE*
WHERE (Salary >= 30000 AND Salary <= 40000) AND Dno = 5;

12. Write a SQL query to find those employees who work in the same department where 'Ramesh' works.

 # Exclude all those records where first name is 'Ramesh'. Return first name, last name

select Fname, Lname, Dno from EMPLOYEE where dno = (select dno from

EMPLOYEE where Fname = 'Ramesh') and Fname <> 'Ramesh';

13 Display all the dept numbers available in Emp and not in dept tables

Minus is no more supported in mysql

```
select Dno
from EMPLOYEE left join  DEPARTMENT on Dno = Dnumber
where Dnumber is NULL;
```

14. Display all the dept numbers available in dept and not in Emp tables

```
select Dnumber
from EMPLOYEE right join  DEPARTMENT on Dno = Dnumber
where Dno is NULL;
```

15. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.

```
select Pnumber, Dnum, Lname, Address, Bdate
from PROJECT, DEPARTMENT, EMPLOYEE
where Dnum=Dnumber and Mgr_ssn=Ssn and Plocation='Stafford';
```

16. For each employee, retrieve the employee's first and last name and the first and last name of his or her immediate supervisor.

 # only employees who have a supervisor are included in the result

 # this is SELF JOIN

```
select E.Fname, E.Lname, S.Fname, S.Lname
from  EMPLOYEE AS E, EMPLOYEE AS S
where E.Super_ssn = S.Ssn;
```

17. For each employee, retrieve the employee's first and last name and the first and last name of his or her immediate supervisor, including those who have no immediate supervisors

```
select E.Fname, E.Lname, S.Fname, S.Lname
from  EMPLOYEE AS E left join EMPLOYEE AS S on  E.Super_ssn = S.Ssn;
```

18. List the details of employees having no immediate supervisor.

```
select *
```

from EMPLOYEE
where Super_ssn IS NULL;

19. Show the resulting salaries if every employee working on the 'ProductX' project is given a 10 percent raise.

#This is use of arithmetic expression in select clause

*select E.Fname, E.Lname, 1.1 * E.Salary AS Increased_sal*
from EMPLOYEE AS E, WORKS_ON AS W, PROJECT AS P
where E.Ssn=W.Essn AND W.Pno=P.Pnumber AND P.Pname='ProductX';

20. List the first name and last name of all employees who work in the same department as the manager with last name 'Wong',

select E.Fname, E.Lname from EMPLOYEE E where E.Dno = (select D.Dnumber from DEPARTMENT D where D.Mgr_ssn = (select E2.Ssn from EMPLOYEE E2 where E2.Lname = 'Wong'));

RESULT

The query was executed and output was successfully obtained.

Exp No-9
TCL COMMANDS

AIM: Implementation of SQL TCL commands Rollback, Commit, Savepoint.

TRANSACTIONAL CONTROL LANGUAGE (TCL): A transaction is a logical unit of work. All changes made to the database can be referred to as a transaction. Transaction changes can be mode permanent to the database only if they are committed a transaction begins with an executable SQL statement & ends explicitly with either role back or commit statement.

COMMIT

The basic syntax for using a COMMIT command in SQL is as follows :

BEGIN;