

## **Practical-01**

**AIM:** Define the elements of the ER (ENTITY-RELATIONSHIP) model and draw the ER diagram of the database.

### **ER MODEL**

- ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
- It develops a conceptual design for the database. It also develops a very.
- In ER modelling, the database structure is portrayed as a diagram called an entity-relationship diagram.

### **COMPONENT OF ER MODEL:**

#### **1. ENTITY:**

An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

Consider an organization as an example - manager, product, employee, department etc. can be taken as an entity.

##### **a. Weak Entity:**

An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.

##### **b. Strong Entity:**

A strong entity set is an entity set that contains sufficient attributes to uniquely identify all its entities. In other words, a primary key exists for a strong entity set. Primary key of a strong entity set is represented by underlining it.

#### **2. ATTRIBUTE:**

The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

**a. Key attribute:**

The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.

**b. Composite attribute:**

An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.

**c. Multivalued attribute:**

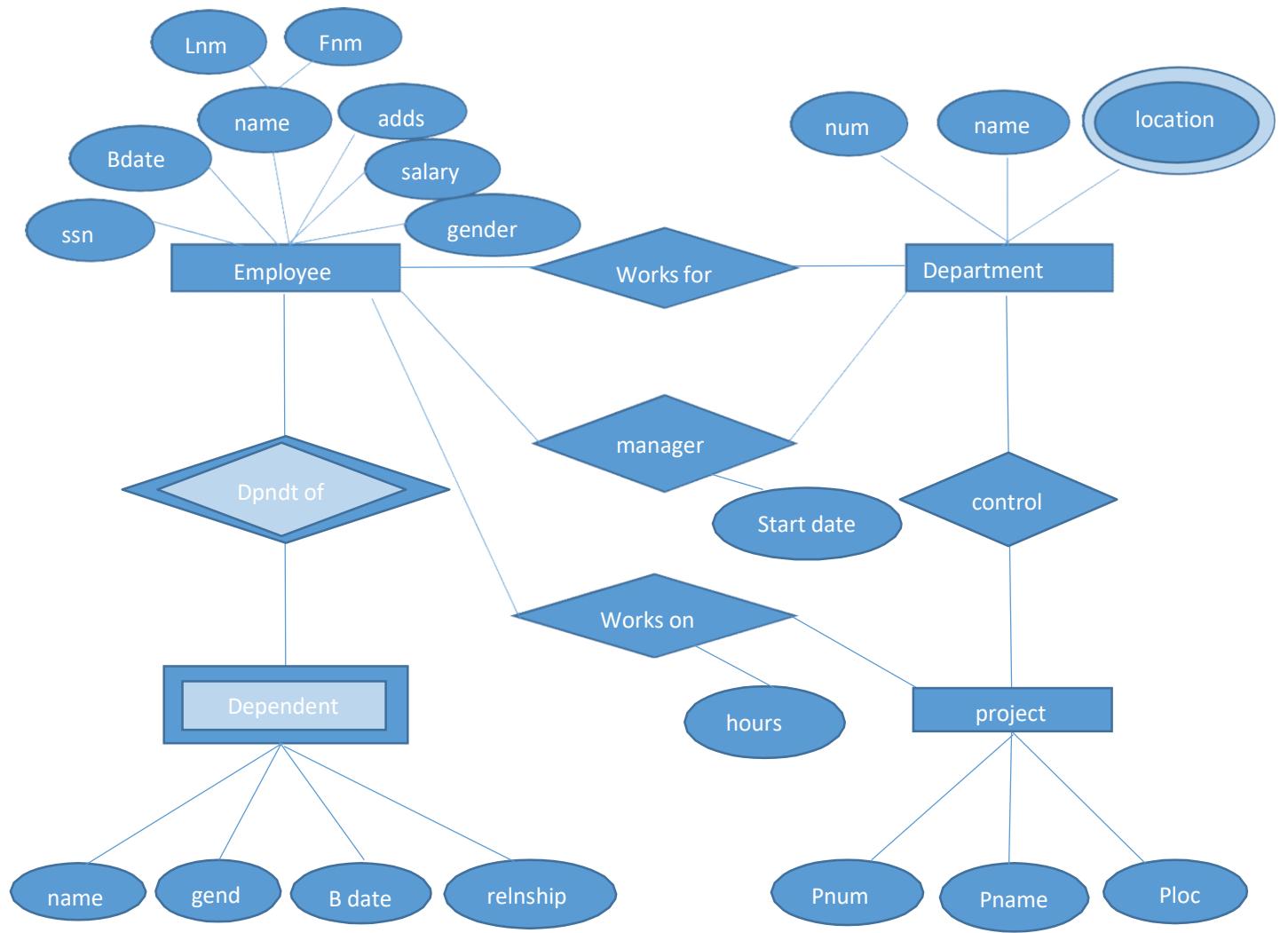
An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.

**d. Derived attribute:**

An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

**3. RELATIONSHIP:**

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.



**E-R design for Company Database**



## Practical- 02

**AIM-** To draw Relational model corresponding to ER Model.

### **Relational Model:**

Relational model can represent as a table with columns and rows. Each row is known as a tuple. Each table of the column has a name or attribute.

**Domain:** It contains a set of atomic values that an attribute can take.

**Attribute:** It contains the name of a column in a particular table. Each attribute  $A_i$  must have a domain,  $\text{dom}(A_i)$

**Relational instance:** In the relational database system, the relational instance is represented by a finite set of tuples. Relation instances do not have duplicate tuples.

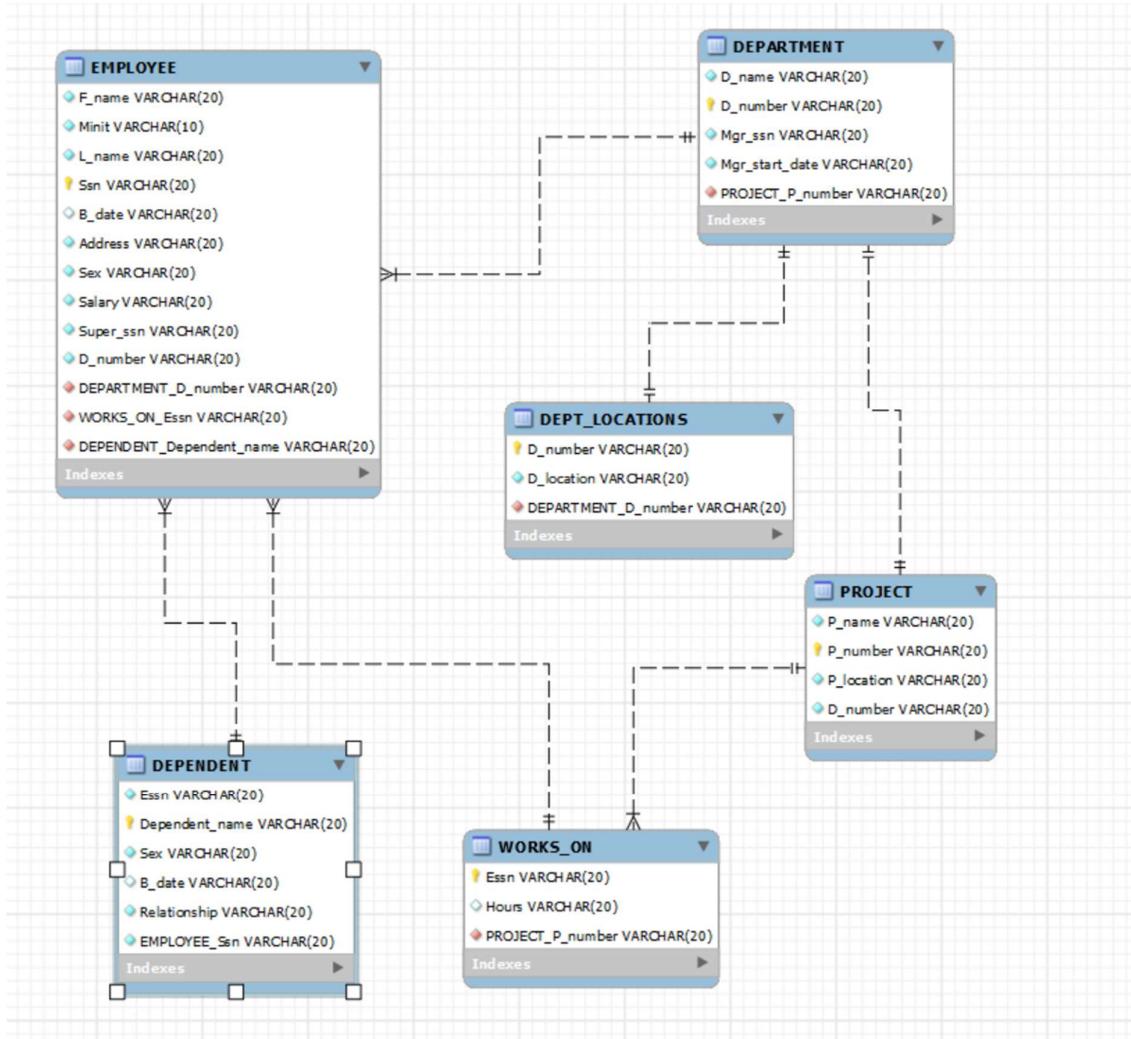
**Relational schema:** A relational schema contains the name of the relation and name of all columns or attributes.

**Relational key:** In the relational key, each row has one or more attributes. It can identify the row in the relation uniquely.

### **Properties of Relations:**

- Name of the relation is distinct from all other relations.
- Each relation cell contains exactly one atomic (single) value
- Each attribute contains a distinct name
- Attribute domain has no significance
- tuple has no duplicate value
- Order of tuple can have a different sequence

## Implementation :



**PRACTICAL FILE**  
**OF**  
**DATABASE MANAGEMENT SYSTEM LAB**  
**SUB-CODE: BCSE-510**  
[COMPUTER SCIENCE AND ENGINEERING]



**MAHARISHI  
MARKANDESHWAR  
(DEEMED TO BE UNIVERSITY)**  
**Mullana-Ambala, Haryana**

(Established under Section 3 of UGC Act, 1956)

(Accredited by NAAC with Grade 'A++')

**Submitted To:**

*Dr. Neeraj Mangla*

(Associate professor)

Department: CSE

MMEC Mullana

**Submitted By:**

*Shahnawaz Alam*

Roll no: 11202722 (B1)

Class: B. Tech. (2<sup>nd</sup> Y)

Session: 2021-2022

**MAHARISHI MARKANDESHWAR ENGINEERING COLLEGE**  
**MAHARISHI MARKANDESHWAR (DEEMED TO BE UNIVERSITY)**  
**MULLANA-AMBALA, HARYANA (INDIA) – 133207**  
(Established Under Section 3 of the UGC Act, 1956)  
(Accredited by NAAC with Grade 'A++')

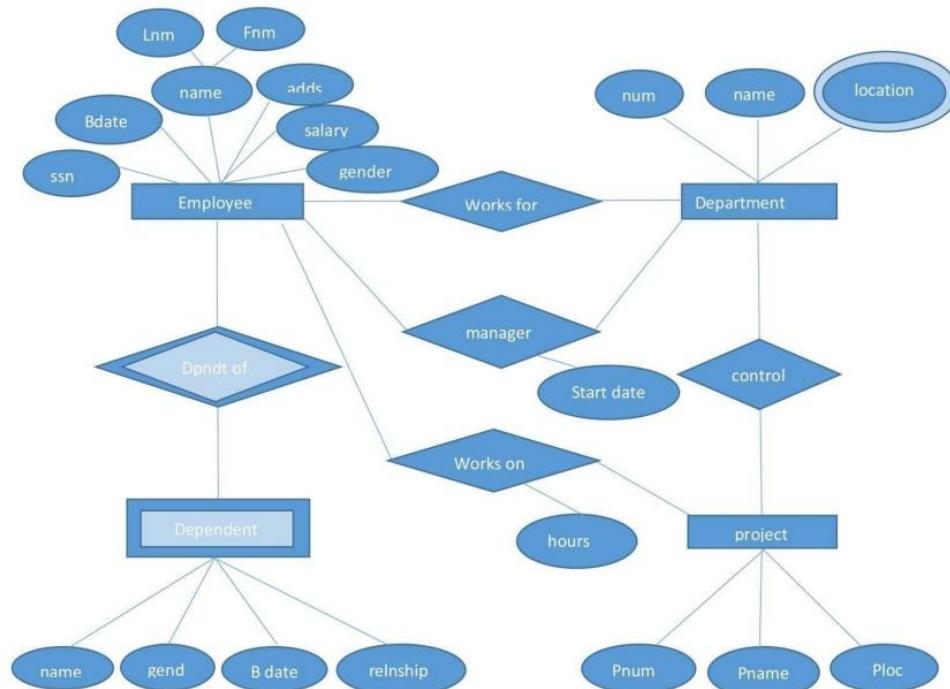
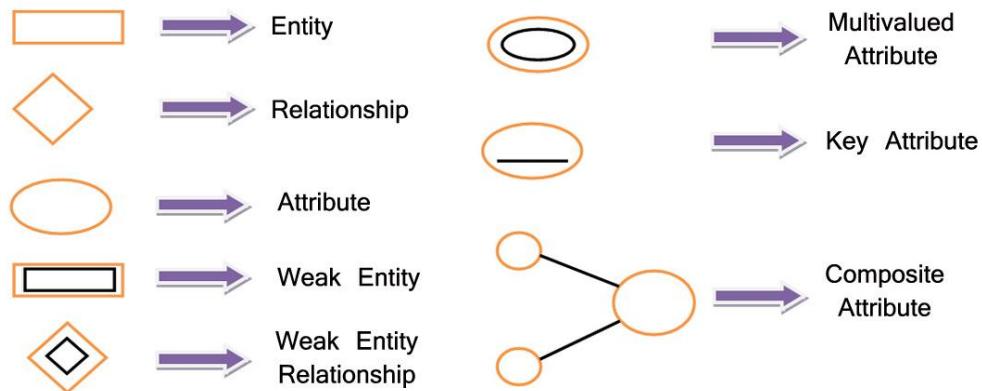
**Composite Attribute:** An attribute that is composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.

- a. **Multivalued Attribute:** An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent a multivalued attribute.
- b. **Derived Attribute:** An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

### 3. RELATIONSHIP:

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

## SYMBOLS OF DIFFERENT COMPONENTS



ER MODEL FOR A COMPANY

## PRACTICAL-05

**AIM: To Insert Data in the Tables of Company DBMS**  
**Implementation:**

### 1. Login into company database

```
mysql> Show databases;
+-----+
| Database |
+-----+
| Company |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0,00 sec)

mysql> use Company;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

### 2. Show tables

```
Database changed
mysql> show tables;
+-----+
| Tables_in_Company |
+-----+
| DEPARTMENT |
| DEPENDENT |
| DEPT_LOCATIONS |
| EMPLOYEE |
| PROJECT |
| WORKS_ON |
+-----+
6 rows in set (0,00 sec)
```

### 3. Inserting values in EMPLOYEE table of company database

```
mysql> insert into EMPLOYEE values('Ratan','S','Kumar','111777111','#0-OA0-2013','xyz_212','M','30000','9876666
12','2');
Query OK, 1 row affected (0,01 sec)

mysql> insert into EMPLOYEE values('Tina','I','Kumari','777222333','07-T21-2001','PNB_MG1','F','40000','98766668
76','5');
Query OK, 1 row affected (0,02 sec)

mysql> insert into EMPLOYEE values('Tanjilina','J','Antunes','234789456','18-N31-2000','PNB_MG1','F','40000','98
7666456','8');
Query OK, 1 row affected (0,02 sec)

mysql> insert into EMPLOYEE values('Ronak','N','Kumar','111999000','1R-9R1-2001','III_HYD6','M','50000','987666
000','4');
Query OK, 1 row affected (0,02 sec)

mysql> insert into EMPLOYEE values('Nitu','N','Kaur','987456321','1U-SU2-2001','PNB_MG4','F','60000','987666246
','9');
Query OK, 1 row affected (0,02 sec)
```

## 4. Show inserted values in employee table

```
mysql> select*from EMPLOYEE;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| F_name | Minit | L_name | Ssn      | B_date    | Address   | Sex     | Salary   | Super_ssn | Dno   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Riddle  | K     | Sharma  | 111333666 | 10-AAA-2001 | 666_xyz   | M       | 40000   | 987666543 | 6
| Kalyan  | T     | Sharma  | 111222666 | 20-A1A-2019 | 333_xyz   | M       | 40000   | 987666763 | 7
| Anmol   | R     | Sharma  | 111111111 | K2-A1A-2001 | 111_xyz   | M       | 50000   | 987666111 | 3
| Ratan   | S     | Kumar   | 111777111 | #0-OAO-2013 | xyz_212   | M       | 30000   | 987666212 | 2
| Tina    | I     | Kumari  | 777222333 | 07-T21-2001 | PNB_MG1   | F       | 40000   | 987666876 | 5
| Tanjina  | J     | Antunes | 234789456 | 18-N31-2000 | PNB_MG1   | F       | 40000   | 987666456 | 8
| Ronak   | N     | Kumar   | 111999000 | 1R-9R1-2001 | III_HYD6  | M       | 50000   | 987666000 | 4
| Nitu    | N     | Kaur    | 987456321 | 1U-5U2-2001 | PNB_MG4   | F       | 60000   | 987666246 | 9
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0,00 sec)

mysql> ■
```

## 5. Inserting value in department table

```
mysql> describe DEPARTMENT;
+-----+-----+-----+-----+-----+
| Field   | Type    | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| D_name   | varchar(20) | YES |   | NULL   |       |
| D_number | varchar(20) | YES |   | NULL   |       |
| Mgr_ssn  | varchar(20) | YES |   | NULL   |       |
| Mgr_start_date | varchar(20) | YES |   | NULL   |       |
+-----+-----+-----+-----+-----+
4 rows in set (0,00 sec)

mysql> insert into DEPARTMENT values('Headquarters','1','888665555','10-Jan-93');
Query OK, 1 row affected (0,02 sec)

mysql> insert into DEPARTMENT values('Administration','4','788665645','18-Jun-85');
Query OK, 1 row affected (0,01 sec)

mysql> insert into DEPARTMENT values('Research','5','333665645','22-may-95');
Query OK, 1 row affected (0,02 sec)

mysql> insert into DEPARTMENT values('Automation','7','123456789','08-sep-97');
Query OK, 1 row affected (0,01 sec)

mysql> select*from DEPARTMENT;
+-----+-----+-----+-----+
| D_name   | D_number | Mgr_ssn   | Mgr_start_date |
+-----+-----+-----+-----+
| Headquarters | 1        | 888665555 | 10-Jan-93    |
| Administration | 4        | 788665645 | 18-Jun-85    |
| Research   | 5        | 333665645 | 22-may-95    |
| Automation  | 7        | 123456789 | 08-sep-97    |
+-----+-----+-----+-----+
4 rows in set (0,01 sec)

mysql> ■
```

## 6. Inserting values in dept\_locations

```

mysql> describe DEPT_LOCATIONS;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| D_number | varchar(20) | YES | | NULL | |
| D_location | varchar(60) | YES | | NULL | |
+-----+-----+-----+-----+-----+
2 rows in set (0,00 sec)

mysql> insert into DEPT_LOCATIONS values('7','Houston');
Query OK, 1 row affected (0,03 sec)

mysql> insert into DEPT_LOCATIONS values('6','Stafford');
Query OK, 1 row affected (0,03 sec)

mysql> insert into DEPT_LOCATIONS values('5','Bellaire');
Query OK, 1 row affected (0,01 sec)

mysql> insert into DEPT_LOCATIONS values('4','Sugarland');
Query OK, 1 row affected (0,04 sec)

mysql> insert into DEPT_LOCATIONS values('4','Houston');
Query OK, 1 row affected (0,03 sec)

mysql> select*from DEPT_LOCATIONS;
+-----+-----+
| D_number | D_location |
+-----+-----+
| 7 | Houston |
| 6 | Stafford |
| 5 | Bellaire |
| 4 | Sugarland |
| 4 | Houston |
+-----+-----+
5 rows in set (0,00 sec)

mysql>

```

## 7. Inserting values in project table

```

mysql> describe PROJECT;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| P_name | varchar(20) | YES | | NULL | |
| P_number | varchar(20) | YES | | NULL | |
| P_location | varchar(60) | YES | | NULL | |
| Dnum | varchar(20) | YES | | NULL | |
+-----+-----+-----+-----+
4 rows in set (0,00 sec)

mysql> insert into PROJECT values('Project_A','1','Bellaire','5');
Query OK, 1 row affected (0,02 sec)

mysql> insert into PROJECT values('Project_B','2','Sugarland','5');
Query OK, 1 row affected (0,02 sec)

mysql> insert into PROJECT values('Project_C','3','Sugarland','4');
Query OK, 1 row affected (0,03 sec)

mysql> insert into PROJECT values('New_benefits','6','Houston','5');
Query OK, 1 row affected (0,02 sec)

mysql> insert into PROJECT values('Computerization','7','Stafford','1');
Query OK, 1 row affected (0,02 sec)

mysql> select*from PROJECT;
+-----+-----+-----+-----+
| P_name | P_number | P_location | Dnum |
+-----+-----+-----+-----+
| Project_A | 1 | Bellaire | 5 |
| Project_B | 2 | Sugarland | 5 |
| Project_C | 3 | Sugarland | 4 |
| New_benefits | 6 | Houston | 5 |
| Computerization | 7 | Stafford | 1 |
+-----+-----+-----+-----+
5 rows in set (0,00 sec)

mysql> ■

```

## 8. Inserting values in works\_on table

```

mysql> describe WORKS_ON;
+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| Essn  | varchar(20) | YES  |     | NULL    |       |
| Pno   | int      | YES  |     | NULL    |       |
| Hours | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+
3 rows in set (0,00 sec)

mysql> insert into WORKS_ON values('123876890','1','7');
Query OK, 1 row affected (0,02 sec)

mysql> insert into WORKS_ON values('123876432','2','3.5');
Query OK, 1 row affected (0,04 sec)

mysql> insert into WORKS_ON values('123876123','11','12.4');
Query OK, 1 row affected (0,01 sec)

mysql> insert into WORKS_ON values('123876341','4','17.1');
Query OK, 1 row affected (0,02 sec)

mysql> insert into WORKS_ON values('123876872','1','20');
Query OK, 1 row affected (0,02 sec)

mysql> select*from WORKS_ON;
+-----+-----+-----+
| Essn  | Pno   | Hours |
+-----+-----+-----+
| 123876890 | 1 | 7 |
| 123876432 | 2 | 3.5 |
| 123876123 | 11 | 12.4 |
| 123876341 | 4 | 17.1 |
| 123876872 | 1 | 20 |
+-----+-----+-----+
5 rows in set (0,00 sec)

mysql>

```

## 9. Inserting value in dependent

```

mysql> describe DEPENDENT;
+-----+-----+-----+-----+
| Field   | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| Essn    | varchar(20) | YES  |     | NULL    |       |
| Dependent_name | varchar(20) | YES  |     | NULL    |       |
| Sex     | varchar(10)  | YES  |     | NULL    |       |
| B_date  | varchar(20)  | YES  |     | NULL    |       |
| Relationship | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+
5 rows in set (0,00 sec)

mysql> insert into DEPENDENT values('123876872','Tina','F','31-jun-82','Daughter');
Query OK, 1 row affected (0,02 sec)

mysql> insert into DEPENDENT values('123876872','Anjali','F','09-04-99','Spouse');
Query OK, 1 row affected (0,02 sec)

mysql> insert into DEPENDENT values('123876873','Randhir','M','12-06-98','Son');
Query OK, 1 row affected (0,01 sec)

mysql> insert into DEPENDENT values('123876453','Sushil','M','02-11-96','Spouse');
Query OK, 1 row affected (0,02 sec)

mysql> insert into DEPENDENT values('123876403','Venkat','M','07-05-99','Son');
Query OK, 1 row affected (0,01 sec)

mysql> select*from DEPENDENT;
+-----+-----+-----+-----+
| Essn    | Dependent_name | Sex | B_date   | Relationship |
+-----+-----+-----+-----+
| 123876872 | Tina        | F   | 31-jun-82 | Daughter    |
| 123876872 | Anjali       | F   | 09-04-99  | Spouse     |
| 123876873 | Randhir     | M   | 12-06-98  | Son        |
| 123876453 | Sushil       | M   | 02-11-96  | Spouse     |
| 123876403 | Venkat       | M   | 07-05-99  | Son        |
+-----+-----+-----+-----+
5 rows in set (0,00 sec)

mysql>

```

**Result: Various data has been inserted into the company database**

## PRACTICAL-06

**AIM:** To perform various select statements in mysql.

**Implementation:**

- 1. Retrieve the birth date and address of the employee(s) whose name is ‘John B. Smith’.**

```
mysql> select Bdate,Address from employee where Fname ='John' and Midname ='B' and Lname ='Smith';
+-----+-----+
| Bdate | Address        |
+-----+-----+
| 1965-01-09 | 731 fondren,Houston,TX |
+-----+-----+
1 row in set (0.21 sec)
```

- 2. Retrieve the name and address of all employees who work for the ‘Research’ department.**

```
mysql> select Fname,Midname,Lname from employee,department where Dname ='Research' and Dnumber =Dno;
+-----+-----+-----+
| Fname | Midname | Lname  |
+-----+-----+-----+
| John  | B       | Smith |
| Franklin | T       | Wong   |
| Joyce | A       | English |
| Ramesh | K       | Narayan |
+-----+-----+-----+
4 rows in set (0.02 sec)
```

- 3. Select all employees who work in department no 5.**

```
mysql> select Fname,Midname,Lname  from employee where Dno =5;
+-----+-----+-----+
| Fname | Midname | Lname  |
+-----+-----+-----+
| John  | B       | Smith |
| Franklin | T       | Wong   |
| Joyce | A       | English |
| Ramesh | K       | Narayan |
+-----+-----+-----+
4 rows in set (0.02 sec)
```

- 4. 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.**

```
mysql> select Pnumber,Dnumber,Lname,Address,Bdate from employee,project,department where Plocation ='Stafford' and Dno =Dnumber and Mgr_SSN =SSN;
+-----+-----+-----+-----+-----+
| Pnumber | Dnumber | Lname | Address        | Bdate      |
+-----+-----+-----+-----+-----+
| 10     | 4       | Wallace | 291 berry,bellaire,TX | 1941-06-20 |
| 10     | 1       | Bong    | 450 stone,Houston,TX  | 1937-11-10 |
| 10     | 5       | Wong    | 638 voss,Houston,TX  | 1955-12-08 |
| 30     | 4       | Wallace | 291 berry,bellaire,TX | 1941-06-20 |
| 30     | 1       | Bong    | 450 stone,Houston,TX  | 1937-11-10 |
| 30     | 5       | Wong    | 638 voss,Houston,TX  | 1955-12-08 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

**5. Retrieve the name of each employee who works on all the projects controlled by department number 5.**

```
mysql> select Fname,Midname,Lname from employee,project,department where Dnumber =5 and Dnum =Dnumber;
+-----+-----+-----+
| Fname | Midname | Lname |
+-----+-----+-----+
| John  | B      | Smith |
| John  | B      | Smith |
| John  | B      | Smith |
| Franklin | T      | Wong |
| Franklin | T      | Wong |
| Franklin | T      | Wong |
| Joyce  | A      | English |
| Joyce  | A      | English |
| Joyce  | A      | English |
| Ramesh  | K      | Narayan |
| Ramesh  | K      | Narayan |
| Ramesh  | K      | Narayan |
| James   | E      | Bong  |
| James   | E      | Bong  |
| James   | E      | Bong  |
| Jennifer | S      | Wallace |
| Jennifer | S      | Wallace |
| Jennifer | S      | Wallace |
| Ahmed   | V      | Jabbar |
| Ahmed   | V      | Jabbar |
| Ahmed   | V      | Jabbar |
| Alicia  | J      | Zelaya |
| Alicia  | J      | Zelaya |
| Alicia  | J      | Zelaya |
+-----+-----+-----+
24 rows in set (0.05 sec)
```

**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.**

```
mysql> select Pnumber from employee,department,project where Lname ='Smith' and Dno =Dnum and Dnumber =Dno;
+-----+
| Pnumber |
+-----+
| 1       |
| 2       |
| 3       |
+-----+
3 rows in set (0.00 sec)
```

**7. For each employee, retrieve the employee’s first and last name and the first and last name of his or her immediate supervisor.**

```
mysql> select e.Fname,e.Midname,e.Lname from employee as e,employee as s where s.Super_SSN =e.SSN;
+-----+-----+-----+
| Fname | Midname | Lname |
+-----+-----+-----+
| Franklin | T      | Wong |
| Franklin | T      | Wong |
| Franklin | T      | Wong |
| James   | E      | Bong  |
| James   | E      | Bong  |
| Jennifer | S      | Wallace |
| Jennifer | S      | Wallace |
+-----+-----+-----+
7 rows in set (0.10 sec)
```

**8. Select all EMPLOYEE SSN.**

```
mysql> select SSN from employee;
+-----+
| SSN      |
+-----+
| 888665555 |
| 123456789 |
| 453453453 |
| 666884444 |
| 333445555 |
| 987654321 |
| 987987987 |
| 999887777 |
+-----+
8 rows in set (0.00 sec)
```

**9. Select all combinations of EMPLOYEE SSN and DEPARTMENT Dname.**

```
mysql> select SSN,Dname from employee,department where Mgr_SSN =SSN and Dno =Dnumber;
+-----+-----+
| SSN      | Dname      |
+-----+-----+
| 888665555 | Headquarters |
| 987654321 | Administration |
| 333445555 | Research    |
+-----+-----+
3 rows in set (0.00 sec)
```

**10. Retrieve the salary of every employee.**

```
mysql> select salary from employee;
+-----+
| salary |
+-----+
| 30000 |
| 40000 |
| 25000 |
| 38000 |
| 55000 |
| 43000 |
| 25000 |
| 25000 |
+-----+
8 rows in set (0.00 sec)
```

## 11. Retrieve the distinct salary values.

```
mysql> select distinct salary from employee;
+-----+
| salary |
+-----+
| 30000 |
| 40000 |
| 25000 |
| 38000 |
| 55000 |
| 43000 |
+-----+
6 rows in set (0.33 sec)
```

## 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.

```
mysql> select Dname,Lname,Fname,Midname,Pname from department,employee,works_on,project where Dnumber =Dno and SSN =ESSN and Pno =Pnumber order by Dname,Lname,Fname;
+-----+-----+-----+-----+-----+
| Dname | Lname | Fname | Midname | Pname |
+-----+-----+-----+-----+-----+
| Administration | Jabbar | Ahmed | V | computerization |
| Administration | Jabbar | Ahmed | V | Newbenefits |
| Administration | Zelaya | Alicia | J | computerization |
| Administration | Zelaya | Alicia | J | Newbenefits |
| Headquarters | Bong | James | E | Reorganization |
| Research | English | Joyce | A | ProductX |
| Research | English | Joyce | A | ProductY |
| Research | Narayan | Ramesh | K | ProductZ |
| Research | Smith | John | B | ProductX |
| Research | Smith | John | B | ProductY |
| Research | Wong | Franklin | T | ProductY |
| Research | Wong | Franklin | T | ProductZ |
| Research | Wong | Franklin | T | computerization |
| Research | Wong | Franklin | T | Reorganization |
+-----+-----+-----+-----+-----+
14 rows in set (0.10 sec)
```

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

```
mysql> select Fname,Midname,Lname from employee where SSN in(select ESSN from dependent where Fname =dependent_name and sex =sex);
+-----+-----+-----+
| Fname | Midname | Lname |
+-----+-----+-----+
| John | B | Smith |
+-----+-----+-----+
1 row in set (0.01 sec)
```

## 14. Retrieve the names of all employees who do not have supervisors.

```
mysql> select Fname,Midname,Lname from employee where Super_SSN is NULL;
+-----+-----+-----+
| Fname | Midname | Lname |
+-----+-----+-----+
| James | E | Bong |
+-----+-----+-----+
1 row in set (0.00 sec)
```

**Result :** Various select statement of Company's Database in mysql are performed.

## PRACTICAL-07

**AIM: To Describe Mysql Operators Including Group By Clause And Use Of Sub Queries.**

**Implementation:**

**1. Retrieve the names of all employees who have two or more dependents.**

```
mysql> select Fname,Midname,Lname from employee where(select count(*) from dependent where SSN =ESSN)>=2;
+-----+-----+-----+
| Fname | Midname | Lname |
+-----+-----+-----+
| John  | B       | Smith |
| Franklin | T       | Wong |
+-----+-----+-----+
2 rows in set (0.07 sec)
```

**2. Retrieve all employees whose address is in Houston, Texas.**

```
mysql> select Fname,Midname,Lname from employee where Address like '%Houston,TX%';
+-----+-----+-----+
| Fname | Midname | Lname |
+-----+-----+-----+
| John  | B       | Smith |
| Franklin | T       | Wong |
| Joyce | A       | English |
| Ramesh | K       | Narayan |
| James  | E       | Bong |
| Ahmed  | V       | Jabbar |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

**3. Find all employees who were born during the 1950s.**

```
mysql> select Fname,Midname,Lname from employee where Bdate between '1950-01-01' and '1959-12-31';
+-----+-----+-----+
| Fname | Midname | Lname |
+-----+-----+-----+
| Franklin | T       | Wong |
+-----+-----+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select Fname,Midname,Lname from employee where SSN in(select ESSN from dependent where Fname =dependent_name and sex =sex);
+-----+-----+-----+
| Fname | Midname | Lname |
+-----+-----+-----+
| John  | B       | Smith |
+-----+-----+-----+
1 row in set (0.01 sec)
```

**5. Retrieve the Social Security numbers of all employees who work on project numbers 1, 2, or 3.**

```
mysql> select distinct ESSN from works_on where Pno in(1,2,3);
+-----+
| ESSN      |
+-----+
| 123456789 |
| 453453453 |
| 333445555 |
| 666884444 |
+-----+
4 rows in set (0.07 sec)
```

**6. Retrieve the names of all employees who do not have supervisors.**

```
mysql> select Fname,Midname,Lname from employee where Super_SSN is NULL;
+-----+-----+-----+
| Fname | Midname | Lname |
+-----+-----+-----+
| James | E       | Bong  |
+-----+-----+-----+
1 row in set (0.00 sec)
```

**7. Retrieve the total number of employees in the company.**

```
mysql> select count(*) from employee;
+-----+
| count(*) |
+-----+
|      8   |
+-----+
1 row in set (0.04 sec)
```

**8. Retrieve the number of employees in the ‘Research’ department.**

```
mysql> select count(*) from employee,department where Dname ='Research' and Dnumber =Dno;
+-----+
| count(*) |
+-----+
|      4   |
+-----+
1 row in set (0.04 sec)
```

**9. Count the number of distinct salary values in the database.**

```
mysql> select distinct salary,count(*) from employee;
+-----+-----+
| salary | count(*) |
+-----+-----+
| 30000 |      8 |
+-----+-----+
1 row in set (0.00 sec)
```

**10. For each department, retrieve the department number, the number of employees in the department, and their average salary.**

```
mysql> select Dno,count(*),avg(salary) from employee group by Dno;
+-----+-----+-----+
| Dno | count(*) | avg(salary) |
+-----+-----+-----+
| 1   |      1 | 55000.0000 |
| 4   |      3 | 31000.0000 |
| 5   |      4 | 33250.0000 |
+-----+-----+-----+
3 rows in set (0.10 sec)
```

**11. For each project, retrieve the project number, the project name, and the number of employees who work on that project.**

```
mysql> select Pnumber,Pname,count(*) from project,employee,works_on where pnumber =Pno and SSN =ESSN group by Pnumber,Pname;
+-----+-----+-----+
| Pnumber | Pname       | count(*) |
+-----+-----+-----+
| 1     | ProductX    | 2 |
| 2     | ProductY    | 3 |
| 3     | ProductZ    | 2 |
| 10    | computerization | 3 |
| 20    | Reorganization | 2 |
| 30    | Newbenefits  | 2 |
+-----+-----+-----+
6 rows in set (0.03 sec)
```

**12. For each project on which more than two employees work, retrieve the project number, the project name, and the number of employees who work on the project.**

```
mysql> select Pnumber,Pname,Dno from employee,project group by Pnumber having count(*)>2;
+-----+-----+-----+
| Pnumber | Pname       | Dno |
+-----+-----+-----+
| 30    | Newbenefits | 1 |
| 20    | Reorganization | 1 |
| 10    | computerization | 1 |
| 3     | ProductZ    | 1 |
| 2     | ProductY    | 1 |
| 1     | ProductX    | 1 |
+-----+-----+-----+
6 rows in set (0.14 sec)
```

**13. For each project, retrieve the project number, the project name, and the number of employees from department 5 who work on the project.**

```
mysql> select Pnumber,Pname,Dno,count(*) from project,employee,department where Dnumber =5 and Dnum =Dnumber and Dnumber =Dno;
+-----+-----+-----+
| Pnumber | Pname   | Dno   | count(*) |
+-----+-----+-----+
|       1 | ProductX |      5 |        12 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

**14. Select the department number and the number of its employees who are making more than \$40,000.**

```
mysql> select Dnumber,count(*) from department,employee where Dnumber =Dno and salary>40000 and(select Dno from employee group by Dno having count(*)>5);
+-----+-----+
| Dnumber | count(*) |
+-----+-----+
|    NULL |      0 |
+-----+-----+
1 row in set (0.00 sec)
```

**Result: Hence the mysql operators including group by clauses and use of sub queries are described.**

## PRACTICAL-08

**AIM:** To Describe MySQL Conversion Functions.

1. DATE: It converts the value into DATE datatype that responsible for the date portion only. It always results in the "YYYY-MM-DD" format.

```
mysql> select date("2022-06-12");
+-----+
| date("2022-06-12") |
+-----+
| 2022-06-12          |
+-----+
1 row in set (0.01 sec)
```

2. DATETIME: It converts the value into the DATETIME data type that responsible for the date and time portion both. It always results in the "YYYY-MM-DD HH:MM:SS" format.

```
mysql> select convert("2018-11-30",datetime);
+-----+
| convert("2018-11-30",datetime) |
+-----+
| 2018-11-30 00:00:00          |
+-----+
1 row in set (0.01 sec)
```

3. TIME: It converts the value into a TIME data type that responsible for the time portion only. It always results in the "HH:MM:SS" format.

```
mysql> select time("21:05:12");
+-----+
| time("21:05:12") |
+-----+
| 21:05:12          |
+-----+
1 row in set (0.01 sec)
```

4. CHAR: It converts a value to the CHAR data type, which has a fixed-length string.

```
mysql> select convert(TutorialTracks,char character set utf8mb4);
ERROR 1054 (42S22): Unknown column 'TutorialTracks' in 'field list'
mysql> select convert('TutorialTracks',char character set utf8mb4);
+-----+
| convert('TutorialTracks',char character set utf8mb4) |
+-----+
| TutorialTracks                                     |
+-----+
1 row in set (0.01 sec)
```

5. SIGNED: It converts a value to SIGNED datatype, which has signed 64-bit integer.

```
mysql> select convert(convert(4-7,unsigned),signed);
+-----+
| convert(convert(4-7,unsigned),signed) |
+-----+
| -3                                |
+-----+
1 row in set (0.01 sec)
```

6. UNSIGNED: It converts a value to the UNSIGNED datatype, which has unsigned 64-bit integer.

```
mysql> select convert(4-7,unsigned);
+-----+
| convert(4-7,unsigned) |
+-----+
| 18446744073709551613 |
+-----+
1 row in set (0.01 sec)
```

7. DECIMAL: It converts a value to the DECIMAL data type, which has a decimal string.

```
mysql> create table items;
ERROR 4028 (HY000): A table must have at least one visible column.
mysql> create table items
    -> (price decimal(5,2));
Query OK, 0 rows affected (0.09 sec)

mysql> insert into items(price)
    -> values(343.04),
    -> (309.98383),
    -> (345.0840);
Query OK, 3 rows affected, 2 warnings (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 2

mysql> select *from items;
+-----+
| price |
+-----+
| 343.04 |
| 309.98 |
| 345.08 |
+-----+
3 rows in set (0.01 sec)
```

8. BINARY: It converts a value to the BINARY data type, which has a binary string.

```
mysql> select binary "TutorialTracks";
+-----+
| binary "TutorialTracks" |
+-----+
| 0x5475746F7269616C547261636B73 |
+-----+
1 row in set, 1 warning (0.01 sec)
```

**Result: Described MySQL Conversion Functions.**

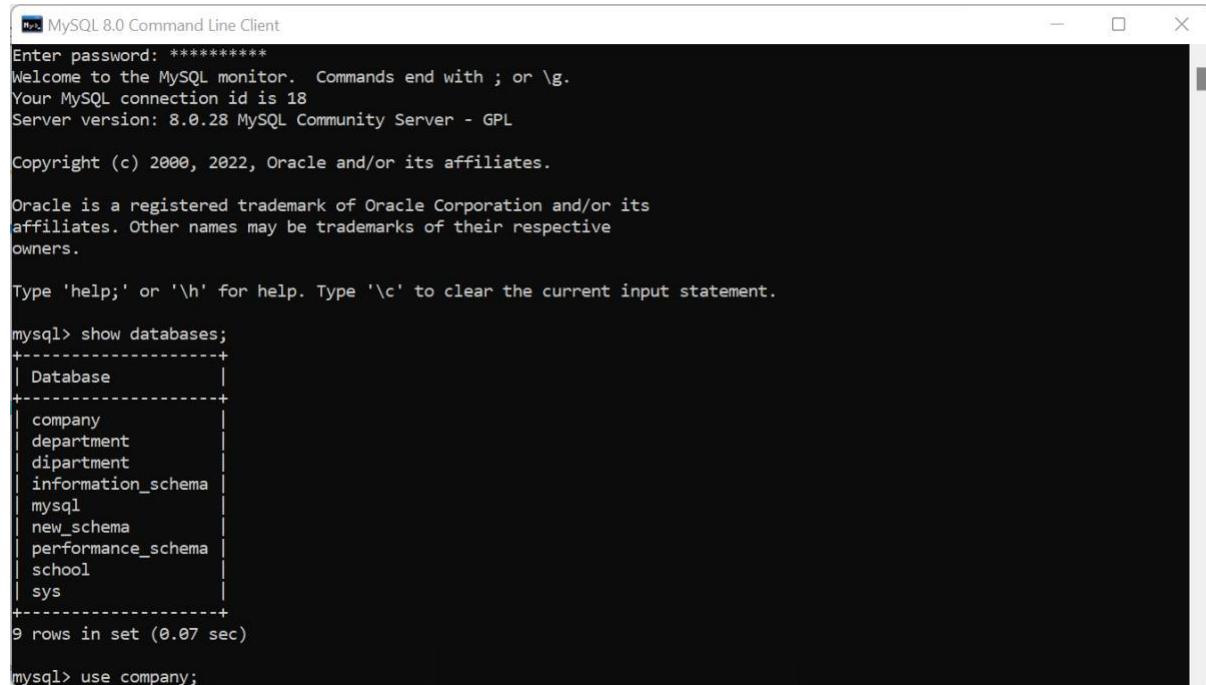
## PRACTICAL-09

**AIM:- To describe mysql operation including Group by or Order by clause.**

1. The **GROUP BY** statement groups rows that have the same values into summary rows, like "find the number of customers in each country".
2. The **Order BY** statement is often used with aggregate functions (**COUNT()**, **MAX()**, **MIN()**, **SUM()**, **AVG()**) to group the result-set by one or more columns.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

```
SELECT COUNT(emo_id), emp_name
FROM employee
GROUP BY emp_name;
```



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client". It displays the MySQL monitor startup message, including the password prompt, welcome message, connection ID, server version, copyright information, and a trademark notice. The user then runs the command "show databases;" which lists nine databases: company, department, dipartment, information\_schema, mysql, new\_schema, performance\_schema, school, and sys. The output shows "9 rows in set (0.07 sec)". Finally, the user runs "use company;" to switch to the company database.

```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| company   |
| department |
| dipartment |
| information_schema |
| mysql      |
| new_schema |
| performance_schema |
| school     |
| sys        |
+-----+
9 rows in set (0.07 sec)

mysql> use company;
```

```
MySQL 8.0 Command Line Client

mysql> use company;
Database changed
mysql> show tables;
+-----+
| Tables_in_company |
+-----+
| employee |
+-----+
1 row in set (0.03 sec)

mysql> describe employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emo_id | int | YES | | NULL |
| emp_name | varchar(12) | YES | | NULL |
| age | int | YES | | NULL |
| salary | int | YES | | NULL |
| aadhar_no | int | YES | | NULL |
+-----+-----+-----+-----+-----+
5 rows in set (0.05 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| emo_id | emp_name | age | salary | aadhar_no |
+-----+-----+-----+-----+-----+
| 112000 | Raman | 22 | 10000 | NULL |
| 112010 | Kumodh | 32 | 50000 | NULL |
| 112000 | Raman | 22 | 10000 | NULL |
| 12345 | harshit | 20 | 100000 | NULL |
| 112022 | Saksham | 20 | 20000 | NULL |
| 112227 | Nitish | 21 | 45000 | NULL |
+-----+-----+-----+-----+-----+
6 rows in set (0.05 sec)
```

```
MySQL 8.0 Command Line Client

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| emo_id | emp_name | age | salary | aadhar_no |
+-----+-----+-----+-----+-----+
| 112000 | Raman | 22 | 10000 | NULL |
| 112010 | Kumodh | 32 | 50000 | NULL |
| 112000 | Raman | 22 | 10000 | NULL |
| 12345 | harshit | 20 | 100000 | NULL |
| 112022 | Saksham | 20 | 20000 | NULL |
| 112227 | Nitish | 21 | 45000 | NULL |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select emp_name, sum(salary) as "total paid" from 1st_emp_paid_sal group by emp_name;
ERROR 1146 (42S02): Table 'company.1st_emp_paid_sal' doesn't exist
mysql> select emp_name, sum(salary) as "total paid" from group by emp_name;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
for the right syntax to use near 'group by emp_name' at line 1
mysql> select * from employee order by name;
ERROR 1054 (42S22): Unknown column 'name' in 'order clause'
mysql> select * from employee order by emp_name;
+-----+-----+-----+-----+-----+
| emo_id | emp_name | age | salary | aadhar_no |
+-----+-----+-----+-----+-----+
| 12345 | harshit | 20 | 100000 | NULL |
| 112010 | Kumodh | 32 | 50000 | NULL |
| 112227 | Nitish | 21 | 45000 | NULL |
| 112000 | Raman | 22 | 10000 | NULL |
| 112000 | Raman | 22 | 10000 | NULL |
| 112022 | Saksham | 20 | 20000 | NULL |
+-----+-----+-----+-----+-----+
```

```
MySQL 8.0 Command Line Client
mysql> select * from employee where salary>=20000 order by emp_name;
+-----+-----+-----+-----+
| emo_id | emp_name | age | salary | aadhar_no |
+-----+-----+-----+-----+
| 12345 | harshit | 20 | 100000 | NULL |
| 112010 | Kumodh | 32 | 50000 | NULL |
| 112227 | Nitish | 21 | 45000 | NULL |
| 112022 | Saksham | 20 | 20000 | NULL |
+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> select * from employee where age>=22 order by emp_name;
+-----+-----+-----+-----+
| emo_id | emp_name | age | salary | aadhar_no |
+-----+-----+-----+-----+
| 112010 | Kumodh | 32 | 50000 | NULL |
| 112000 | Raman | 22 | 10000 | NULL |
| 112000 | Raman | 22 | 10000 | NULL |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from employee where salary>=20000 order by emp_name asc ;
+-----+-----+-----+-----+
| emo_id | emp_name | age | salary | aadhar_no |
+-----+-----+-----+-----+
| 12345 | harshit | 20 | 100000 | NULL |
| 112010 | Kumodh | 32 | 50000 | NULL |
| 112227 | Nitish | 21 | 45000 | NULL |
| 112022 | Saksham | 20 | 20000 | NULL |
+-----+-----+-----+-----+
```

```
MySQL 8.0 Command Line Client
mysql> select * from employee where salary>=20000 order by salary desc;
+-----+-----+-----+-----+
| emo_id | emp_name | age | salary | aadhar_no |
+-----+-----+-----+-----+
| 12345 | harshit | 20 | 100000 | NULL |
| 112010 | Kumodh | 32 | 50000 | NULL |
| 112227 | Nitish | 21 | 45000 | NULL |
| 112022 | Saksham | 20 | 20000 | NULL |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from employee where salary>=20000 order by emp_name desc;
+-----+-----+-----+-----+
| emo_id | emp_name | age | salary | aadhar_no |
+-----+-----+-----+-----+
| 112022 | Saksham | 20 | 20000 | NULL |
| 112227 | Nitish | 21 | 45000 | NULL |
| 112010 | Kumodh | 32 | 50000 | NULL |
| 12345 | harshit | 20 | 100000 | NULL |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client
mysql> select emp_name, count(*) from employee group by emp_name;
+-----+-----+
| emp_name | count(*) |
+-----+-----+
| Raman    |      2 |
| Kumodh   |      1 |
| harshit  |      1 |
| Saksham  |      1 |
| Nitish   |      1 |
+-----+-----+
5 rows in set (0.04 sec)

mysql> select salary, count(*) from employee group by salary;
+-----+-----+
| salary | count(*) |
+-----+-----+
| 10000  |      2 |
| 50000  |      1 |
| 100000 |      1 |
| 20000  |      1 |
| 45000  |      1 |
+-----+-----+
5 rows in set (0.00 sec)

mysql> select age, count(*) from employee group by age;
+-----+-----+
| age  | count(*) |
+-----+-----+
| 22   |      2 |
| 32   |      1 |
+-----+-----+
2 rows in set (0.00 sec)
```

```
MySQL 8.0 Command Line Client
mysql> select age, count(*) from employee group by age;
+-----+-----+
| age  | count(*) |
+-----+-----+
| 22   |      2 |
| 32   |      1 |
| 20   |      2 |
| 21   |      1 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> ^C
mysql> select salary, count(*),sum(salary) from employee group by salary;
+-----+-----+-----+
| salary | count(*) | sum(salary) |
+-----+-----+-----+
| 10000 |      2 |     20000 |
| 50000 |      1 |     50000 |
| 100000 |     1 |   100000 |
| 20000 |      1 |     20000 |
| 45000 |      1 |     45000 |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

## PRACTICAL-10

**AIM :- To perform various join on company database.**

**Description:-**

1. **INNER JOIN:** The inner join clause joins two tables based on a condition which is known as a join predicate.

Only rows will be joined which will satisfy the condition.

The inner join clause compares each row from first table with every row in second table. The two rows are joined which satisfy the condition.

**Syntax:-**

```
SELECT column_list
FROM table_1
INNER JOIN table_2 ON join_condition;
```

2. **LEFT OUTER JOIN:** The LEFT OUTER JOIN returns all rows from the left hand table specified in the ON condition and only those rows from the other table where the join condition is fulfilled.

**Syntax:-**

```
SELECT column_list
FROM table_1
LEFT [OUTER] join table2
ON table1.column=table2.column;
```

3. **RIGHT OUTER JOIN :** The MySQL Right Outer Join returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where he join condition is fulfilled.

**Syntax:-**

```
SELECT column_list
FROM table_1
RIGHT [OUTER] join table2
ON table1.column=table2.column;
```

4. **CROSS OUTER JOIN:** The cross join clause does not have a join condition. The cross join makes a Cartesian product of rows from the joined tables. The cross join combines each row from the first table with every row from the right table to make the result set.

**Syntax:-**

```
SELECT column_list
FROM table_1
CROSS JOIN join table2
```

5. **SELF JOIN:** A SELF JOIN is a join in which a table is joined with itself. To join a table itself means that each row of the table is combined with itself and with every other row of the table.

The self join can be viewed as a join of two copies of the same table.

**Syntax:-**

```
SELECT a.column_name,b.column.name....
FROM table1 a,table1 b
WHERE a.common_field=b.common_field;
```

## Operations and output

```
mysql> select * from deptt_head;
+-----+-----+-----+
| head_name | head_id | age |
+-----+-----+-----+
| vikas     | 2501    | 28  |
| Balwinder | 2502    | 30  |
| shyam     | 2503    | 35  |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select deptt_head.head_name,deptt_head.age,employee.emp_name from deptt_head inner join employee on deptt_head.head_id=employee.emp_id;
+-----+-----+-----+
| head_name | age   | emp_name |
+-----+-----+-----+
| vikas     | 28    | harry   |
| Balwinder | 30    | ricky   |
| shyam     | 35    | sharan  |
| shyam     | 35    | sharan  |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select deptt_head.head_name,deptt_head.age,employee.emp_name from deptt_head left join employee on deptt_head.head_id=employee.emp_id;
+-----+-----+-----+
| head_name | age   | emp_name |
+-----+-----+-----+
| vikas     | 28    | harry   |
| Balwinder | 30    | ricky   |
| shyam     | 35    | sharan  |
| shyam     | 35    | sharan  |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select deptt_head.head_name,deptt_head.age,employee.emp_name from deptt_head right join employee on deptt_head.head_id=employee.emp_id;
+-----+-----+-----+
| head_name | age   | emp_name |
+-----+-----+-----+
| vikas     | 28    | harry   |
| Balwinder | 30    | ricky   |
| shyam     | 35    | sharan  |
| NULL      | NULL   | avni    |
| NULL      | NULL   | avni    |
| shyam     | 35    | sharan  |
+-----+-----+-----+
```

```
select emp_name,head_name from employee cross join deptt_head' at line 2
mysql> select emp_name,head_name from employee cross join deptt_head;
+-----+-----+
| emp_name | head_name |
+-----+-----+
| harry    | vikas    |
| harry    | Balwinder|
| harry    | shyam    |
| ricky    | vikas    |
| ricky    | Balwinder|
| ricky    | shyam    |
| sharan   | vikas    |
| sharan   | Balwinder|
| sharan   | shyam    |
| avni     | vikas    |
| avni     | Balwinder|
| avni     | shyam    |
| avni     | vikas    |
| avni     | Balwinder|
| avni     | shyam    |
| sharan   | vikas    |
| sharan   | Balwinder|
| sharan   | shyam    |
+-----+-----+
18 rows in set (0.00 sec)
```

```
mysql> select a.emp_name,b.emp_id,a.salary from employee a,employee b where a.salary<b.salary;
+-----+-----+-----+
| emp_name | emp_id | salary |
+-----+-----+-----+
| harry    | 2502   | 10000
| harry    | 2503   | 10000
| ricky    | 2503   | 15000
| harry    | 2504   | 10000
| ricky    | 2504   | 15000
| sharan   | 2504   | 25000
| sharan   | 2504   | 25000
| harry    | 2504   | 10000
| ricky    | 2504   | 15000
| sharan   | 2504   | 25000
| sharan   | 2504   | 25000
| harry    | 2503   | 10000
| ricky    | 2503   | 15000
+-----+-----+-----+
13 rows in set (0.00 sec)

mysql>
```