Disciplinary

Database Management System

CS3003

Assignment 2

Sheikh Muhammed Tadeeb (AU19B1014)

❖ Problem Statement:

The following things needs to be accomplished: -

1) Creating a database using DDL statements in the MySQL command-line tool.
2) Populating the database row-by-row from the provided raw data using command line terminal of MySQL.
3) Writing SQL queries to the database using DML and DDL.

❖ PART -1 (Creating database and tables using DDL):

- **Step1) Creating a database**.

```
create database company;
```

- **Step2) Entering into that database.**

```
use company;
```

Employee:

- **Step3) Creating tables in that database.**

```
CREATE TABLE employee (
SSN int(8) Primary key ,
F_name varchar(20) NOT NULL,
M_name varchar(20),
L_name varchar(20) NOT NULL,
Address varchar(30) NOT NULL,
gender char(2)
);
```

Department:

```
➢ CREATE TABLE department (
DName varchar(30) NOT NULL,
 Dnumber int(2) Primary key,
Mgr_start_date DATE NOT NULL,
Mgr_ssn int(8),
CONSTRAINT FK FOREIGN KEY (Mgr_ssn) REFERENCES employee (SSN) );
```

OR

-
  ```
  CREATE TABLE department (
  DName varchar(30) NOT NULL,
  Dnumber int(2) Primary key,
  Mgr_start_date DATE NOT NULL,
  Mgr_ssn int(8) REFERENCES employee(SSN)
  );
  ```

Dept_location:

-
  ```
  create table dept_location (
  Dnumber int(2) Primary key,
  Dlocation varchar(30) NOT NULL
  );
  ```

Project:

-
  ```
  create table project (
  Pname char(30),
  Pnumber int(4) PRIMARY KEY,
  Plocation char(20),
  Dnum int(2)
  );
  ```

Dependent:

-
  ```
  CREATE TABLE dependent (
  Essn int(8),
  dependent_name varchar(30),
  Gender char(1) DEFAULT ‘-‘,
  Relation varchar(20),
  PRIMARY KEY (Essn, dependent_name)
  );
  ```

Works_on:

-
  ```
  CREATE TABLE Works_on (
  Essn int(8),
  Pno int(4) CHECK (Pno>0),
  Hours TIME
  );
  ```

- **Step4) Altering some values in table using ADD, DROP, RENAME, CHANGE & MODIFY.**

  - ALTER table employee ADD Salary int(20) CHECK (Salary > 1000);
  - ALTER table employee ADD Dno int(2) CHECK (Dno>0);
  - ALTER table employee DROP COLUMN Salary, DROP COLUMN Dno;
  - ALTER table employee ADD (Salary int(20) CHECK (Salary > 1000), Dno int CHECK (Dno > 0));
  - Alter table employee ADD Foreign key (Dno) REFERENCES department(Dnumber);
  - Alter table employee modify Dno int(2) NOT NULL ;
  - Alter table employee ADD ex int UNIQUE;
  - Alter table employee drop ex;
  - Alter table employee MODIFY Address varchar(20) NOT NULL AFTER Salary;
  - Alter table employee RENAME Employees;
  - Alter table employee CHANGE ssn Ssn int(8) ;
  - Alter table employee ADD COLUMN Super_ssn int(8) REFERENCES employee(Ssn);
  - Alter table employee ADD FOREIGN KEY (Super_ssn) REFERENCES employee(Ssn);
  - Alter table employee ADD COLUMN Bdate DATE AFTER L_name;
  - Alter table dept_location ADD primary key (Dnumber);
  - Alter table dept_location ADD FOREIGN KEY (Dnumber) REFERENCES department(Dnumber);
  - Alter table project ADD FOREIGN KEY (Dnum) REFERENCES department(Dnumber);
  - Alter table project MODIFY Dnum int(2) NOT NULL ;

- ALTER TABLE dependent ADD FOREIGN KEY (Essn) REFERENCES employee(Ssn);
- Alter table dependent ADD COLUMN Bdate DATE AFTER Gender ;
- ALTER table Works_on ADD PRIMARY KEY(Essn,Pno);
- ALTER table Works_on ADD FOREIGN KEY(Essn) REFERENCES employee(Ssn),ADD FOREIGN KEY(Pno) REFERENCES project(Pnumber);
- Alter table dependent modify dependent_name varchar(30) NULL ;
- ALTER table works_on ADD FOREIGN KEY (Pno) REFERENCES project(Pnumber) ON DELETE CASCADE ON UPDATE CASCADE;

❖ Conclusion (Part-1)

S.no	Analysis
1)	We can’t enter spaces in column names.
2)	We can’t define two primary key using column constraint but we can do this using table constraint.
3)	We can’t give size to DATE datatype.
4)	Once declared we can’t convert a primary key to a foreign key.
5)	We can use modify only to change datatype, size, order and only NOT NULL constraint.
6)	We can use ALTER command along with ADD, MODIFY, RENAME, DROP & CHANGE for redefining the definition of a table.
7)	We need to insert size of at least one varchar datatype (if used) otherwise MySQL will throw an error.
8)	Char is Fixed length field i.e. use it when we need faster execution.
9)	Varchar is variable length field i.e. use it when we need to consume less memory.

❖ PART-2 (Populating the database and tables using DML)

Department:

Step1) Inserting values in department table.

- Insert into department values ('Quality',5,1988-05-22, NULL);

- Insert into department (Dname, Dnumber, Mgr_start_date, Mgr_ssn) values
  ('Analysis',4,'1995-01-01’, NULL),
  ('Research',1,'1981-06-19’, NULL);

Employee:

Step2) Inserting values in employee table.

- insert into employee (Ssn,F_name,M_name,L_name,Bdate,gender,Salary,Address,Dno,Super_ssn) values

  (1234,'John','B','Smith',1965-01-09,'M',30000,'12 Amarpura, Ujjain',5,3334),

  (3334,'Franklin','T','Wong','1950-12-08','M',40000,'21 Shivhare colony, Alandi',5,8886),

  (9998,'Alicia', NULL,'Sharma','1968-01-19','F',25000,'32 Mangal colony, Pune',4,9876),

  (9876,'Pradeep', NULL,'Kumar','1950-05-08','M',100000,'2 Azad Nagar, Pune',4,8886),

  (6668,'Ramesh', NULL,'Jain','1962-09-15','M',200000,'23 Nanakheda, Ujjain',5,3334),

  (4534,'Preeti','L','Pal','1972-07-31','F',25000,'2 Shastri Nagar, Pune',5,3334),

  (9879,'Jabbar', NULL,'Ahmad','1969-03-29','M',38000,'22 Poonam colony, Indore',4,9876),

  (8886,'Smith','H','Borg','1937-11-10','M',55000,'12 Hira Mil, Indore',1,NULL);

Dept_location:

Step3) Inserting values in dept_location table.

-
  ```
  insert into dept_location (Dnumber, Dlocation) values
  (1,'Indore'),
  (4,'Pune'),
  (5,'Pune');
  ```

Project:

Step4) Inserting values in project table.

-
  ```
  INSERT INTO project (Pname, Pnumber, Plocation, Dnum) values
  ('X',1,'Pune',5),
  ('Y',2,'Pune',5),
  ('Z',3,'Pune',4),
  ('P',4,'Indore',1),
  ('Q',5,'Ujjain',5);
  ```

Works_on:

-
  ```
  INSERT INTO works_on (Essn, Pno, Hours) values
  (1234,1,'33:05:00'),
  (1234,2,'07:05:00'),
  (6668,3,'40:00:00'),
  (4534,1,'20:00:00'),
  (4534,2,'20:00:00'),
  (3334,4,'10:80:00'),
  (9998,4,'30:10:00'),
  (9876,5,'15:00:00'),
  (8886,5, NULL);
  ```

Dependent:

-
  ```
  insert into dependent (Essn, dependent_name, Gender, Bdate, Relation) values
  (3334,'Ram','M','1950-04-05','Son'),
  (3334,'Simran','F','1983-10-25','Daughter'),
  (3334,'Sunita','F','1942-02-28','Spouse'),
  (9876,'Sunil','M','1988-01-04','Son'),
  (1234,'Sumit','M','1967-05-05','Son');
  ```

Step2) Using INSERT, UPDATE & DELETE

-
  ```
  UPDATE department SET Mgr_start_date = '1988-05-22';
  ```

-
  ```
  SET FOREIGN_KEY_CHECKS = 0;
  ```

- SET FOREIGN_KEY_CHECKS = 1;

- Delete from dept_location;

- insert into dept_location (Dnumber, Dlocation) values
  (1,'Indore'),
  (4,'Pune'),
  (5,'Ujjain');
- UPDATE dept_location SET Dlocation = 'Pune' where Dnumber = 5;

- DELETE from works_on where Essn = 9998;

- Insert into works_on values (9998,4,'30:10:00');

- Update employee SET Bdate = '1965-01-09' where F_name = 'John';

- Update project SET Pname = 'C' where Pnumber = 5;

❖ Conclusion (Part-2):

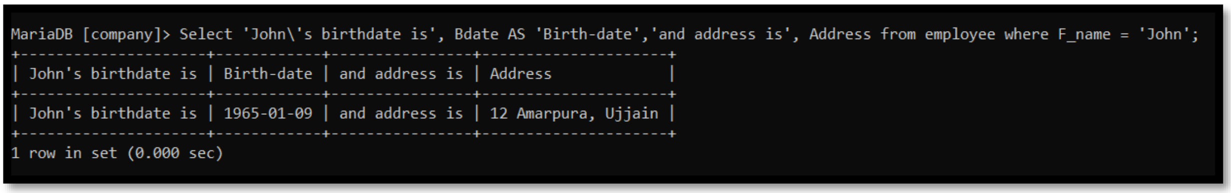
S.no	Analysis
1)	The difference between MODIFY and UPDATE is MODIFY is used to reconstruct the columns of table whereas UPDATE is used for redefining table content.
2)	Difference between DELETE & DROP is delete is used to remove table content whereas DROP is used to remove the whole table.
3)	Difference between ADD and Insert is also the same.

❖ PART-3 (Solving queries using DDL & DML)

TASK-1)

Q1. Retrieve the birth date and address of the employee whose name is John.
Sol:

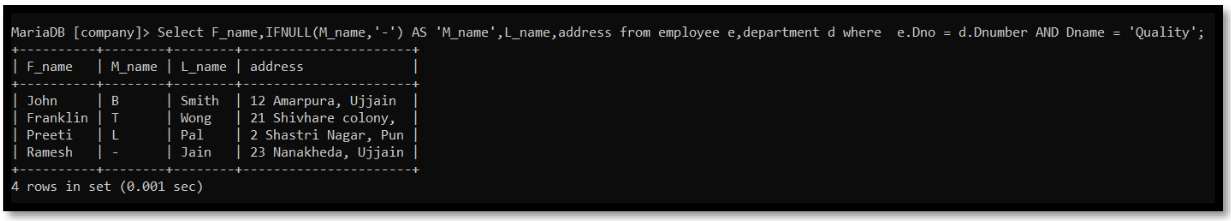
Select 'John\'s birthdate is', Bdate AS 'Birth-date’, ‘and address is', Address from employee where F_name = 'John';



```
MariaDB [company]> Select 'John\'s birthdate is', Bdate AS 'Birth-date','and address is', Address from employee where F_name = 'John';
+---------------------+------------+----------------+---------------------+
| John's birthdate is | Birth-date | and address is | Address             |
+---------------------+------------+----------------+---------------------+
| John's birthdate is | 1965-01-09 | and address is | 12 Amarpura, Ujjain |
+---------------------+------------+----------------+---------------------+
1 row in set (0.000 sec)
```

Q2. Retrieve the name and address of all the employees who work for the quality department.
Sol:

Select F_name, IFNULL(M_name,'-') AS 'M_name', L_name, address from employee e, department d where e.Dno = d.Dnumber AND Dname = 'Quality';



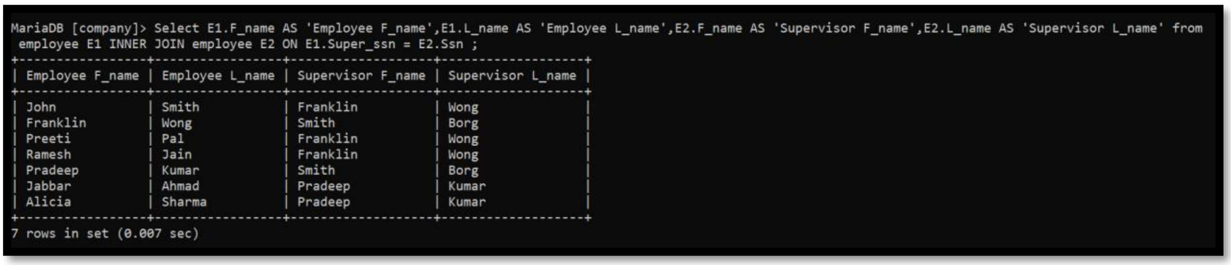
```
MariaDB [company]> Select F_name,IFNULL(M_name,'-') AS 'M_name',L_name,address from employee e,department d where  e.Dno = d.Dnumber AND Dname = 'Quality';
+----------+--------+--------+---------------------+
| F_name   | M_name | L_name | address             |
+----------+--------+--------+---------------------+
| John     | B      | Smith  | 12 Amarpura, Ujjain  |
| Franklin | T      | Wong   | 21 Shivhare colony,  |
| Preeti   | L      | Pal    | 2 Shastri Nagar, Pun |
| Ramesh   | -      | Jain   | 23 Nanakheda, Ujjain |
+----------+--------+--------+---------------------+
4 rows in set (0.001 sec)
```

Q3. For each employee retrieve the employees first and last name and first and last name of his /her immediate supervisor.
Sol:

Select E1.F_name AS 'Employee F_name',E1.L_name AS 'Employee L_name',E2.F_name AS 'Supervisor F_name',E2.L_name AS 'Supervisor L_name' from employee E1 INNER JOIN employee E2 ON E1.Super_ssn = E2.Ssn ;

OR

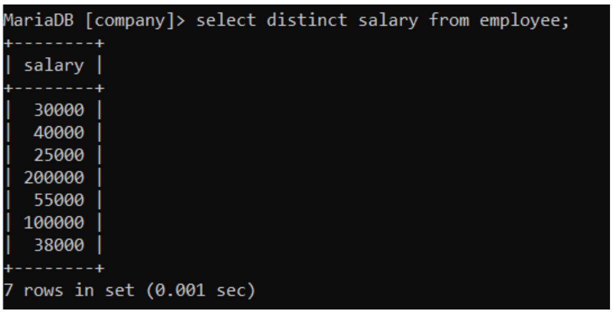
Select E1.F_name AS 'Employee F_name',E1.L_name AS 'Employee L_name',E2.F_name AS 'Supervisor F_name',E2.L_name AS 'Supervisor L_name' from employee E1 , employee E2 where E1.Super_ssn = E2.Ssn ;



```
MariaDB [company]> Select E1.F_name AS 'Employee F_name',E1.L_name AS 'Employee L_name',E2.F_name AS 'Supervisor F_name',E2.L_name AS 'Supervisor L_name' from
 employee E1 INNER JOIN employee E2 ON E1.Super_ssn = E2.Ssn ;
+-----------------+-----------------+-------------------+-------------------+
| Employee F_name | Employee L_name | Supervisor F_name | Supervisor L_name |
+-----------------+-----------------+-------------------+-------------------+
| John            | Smith           | Franklin          | Wong              |
| Franklin        | Wong            | Smith             | Borg              |
| Preeti          | Pal             | Franklin          | Wong              |
| Ramesh          | Jain            | Franklin          | Wong              |
| Pradeep         | Kumar           | Smith             | Borg              |
| Jabbar          | Ahmad           | Pradeep           | Kumar             |
| Alicia          | Sharma          | Pradeep           | Kumar             |
+-----------------+-----------------+-------------------+-------------------+
7 rows in set (0.007 sec)
```

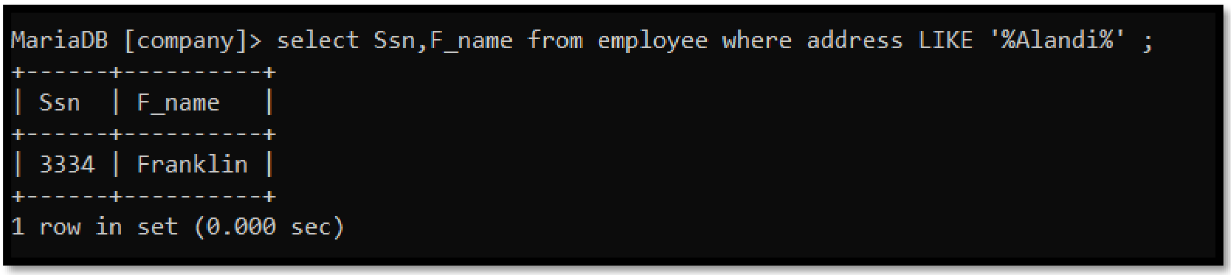
Q4. Retrieve all the distinct salary values.
Sol:

Select distinct salary from employee;



```
MariaDB [company]> select distinct salary from employee;
+--------+
| salary |
+--------+
|  30000 |
|  40000 |
|  25000 |
| 200000 |
|  55000 |
| 100000 |
|  38000 |
+--------+
7 rows in set (0.001 sec)
```

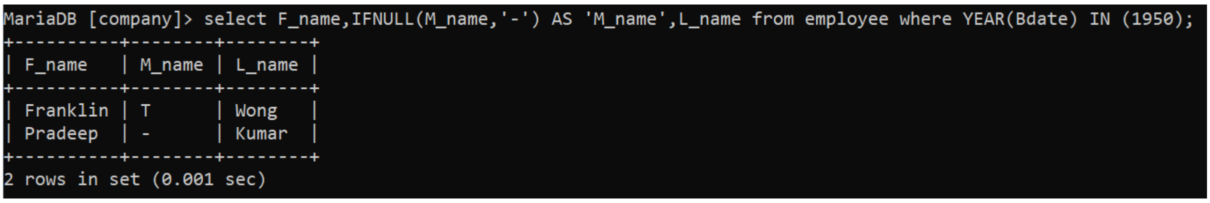
Q5. Retrieve all employees whose address is in Alandi.
Sol:
Select Ssn, F_name from employee where address LIKE '%Alandi%';



```
MariaDB [company]> select Ssn,F_name from employee where address LIKE '%Alandi%' ;
+------+----------+
| Ssn  | F_name   |
+------+----------+
| 3334 | Franklin |
+------+----------+
1 row in set (0.000 sec)
```

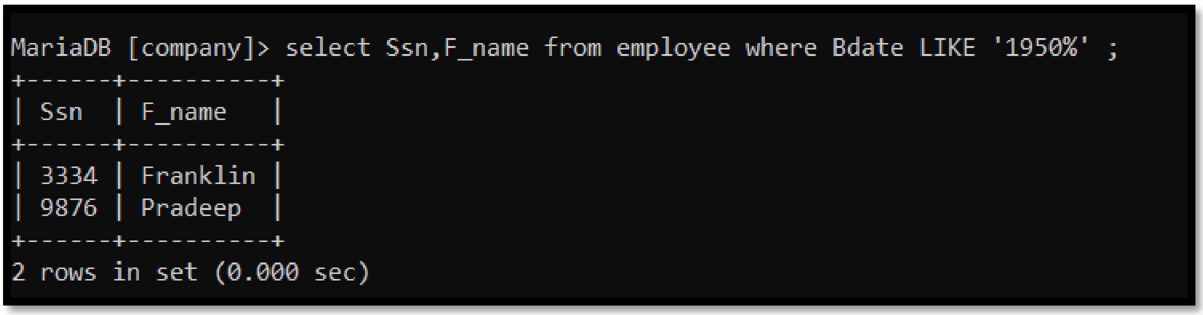
Q6. Find all employees who were born during the 1950s.
Sol:

Select Ssn, F_name from employee where Bdate LIKE '1950%';
OR
select F_name, IFNULL(M_name,'-') AS 'M_name', L_name from employee where YEAR(Bdate) IN (1950);



```
MariaDB [company]> select F_name,IFNULL(M_name,'-') AS 'M_name',L_name from employee where YEAR(Bdate) IN (1950);
+----------+--------+--------+
| F_name   | M_name | L_name |
+----------+--------+--------+
| Franklin | T      | Wong   |
| Pradeep  | -      | Kumar  |
+----------+--------+--------+
2 rows in set (0.001 sec)
```

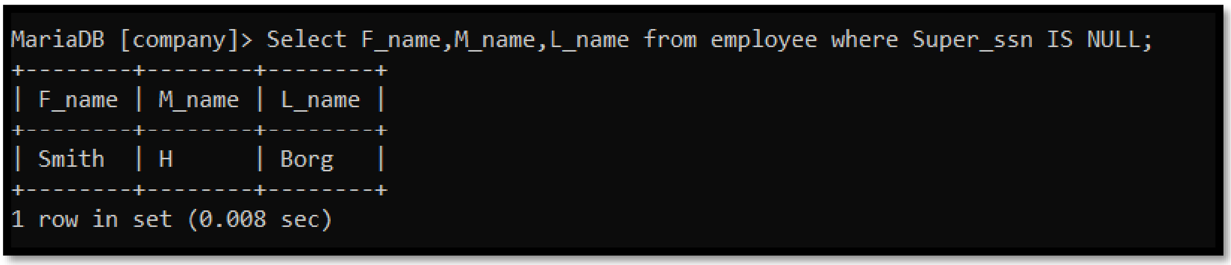
OR



```
MariaDB [company]> select Ssn,F_name from employee where Bdate LIKE '1950%' ;
+------+----------+
| Ssn  | F_name   |
+------+----------+
| 3334 | Franklin |
| 9876 | Pradeep  |
+------+----------+
2 rows in set (0.000 sec)
```

Q7. Retrieve the name of all employees who do not have supervisors.
Sol:

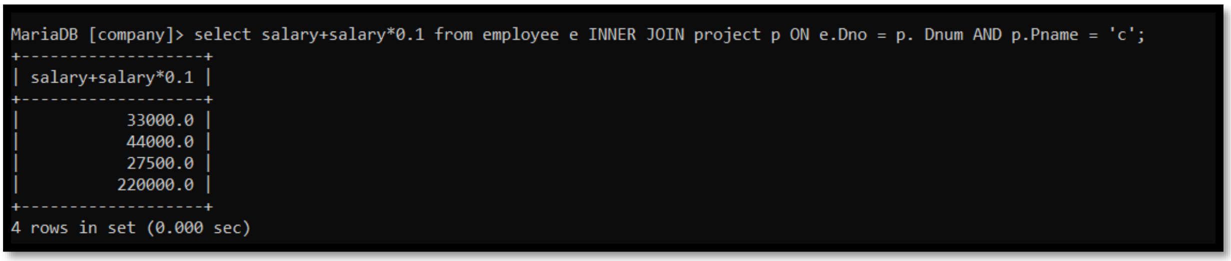
Select F_name, M_name, L_name from employee where Super_ssn IS NULL;



```
MariaDB [company]> Select F_name,M_name,L_name from employee where Super_ssn IS NULL;
+--------+--------+--------+
| F_name | M_name | L_name |
+--------+--------+--------+
| Smith  | H      | Borg   |
+--------+--------+--------+
1 row in set (0.008 sec)
```

Q8. Show the resulting salary if every employee working on c product is given a 10% rise.
Sol:

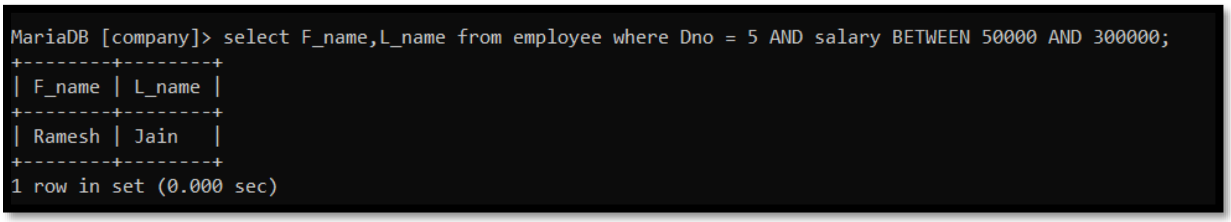
Select salary+salary*0.1 from employee e INNER JOIN project p ON e.Dno = p. Dnum AND p.Pname = 'c';



```
MariaDB [company]> select salary+salary*0.1 from employee e INNER JOIN project p ON e.Dno = p. Dnum AND p.Pname = 'c';
+-------------------+
| salary+salary*0.1 |
+-------------------+
|           33000.0 |
|           44000.0 |
|           27500.0 |
|          220000.0 |
+-------------------+
4 rows in set (0.000 sec)
```

Q9. Retrieve all employees in department number 5 whose salary is between 50000 and 300000.
Sol:

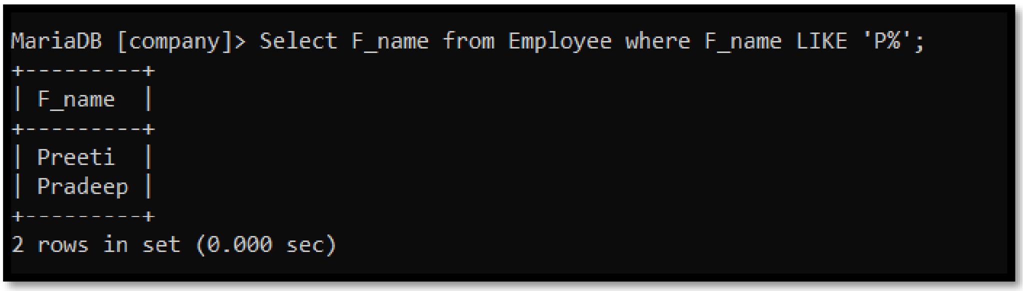
Select F_name, L_name from employee where Dno = 5 AND salary BETWEEN 50000 AND 300000;



```
MariaDB [company]> select F_name,L_name from employee where Dno = 5 AND salary BETWEEN 50000 AND 300000;
+--------+--------+
| F_name | L_name |
+--------+--------+
| Ramesh | Jain   |
+--------+--------+
1 row in set (0.000 sec)
```

Q10. Display the name of all employees starting with "P".
Sol:

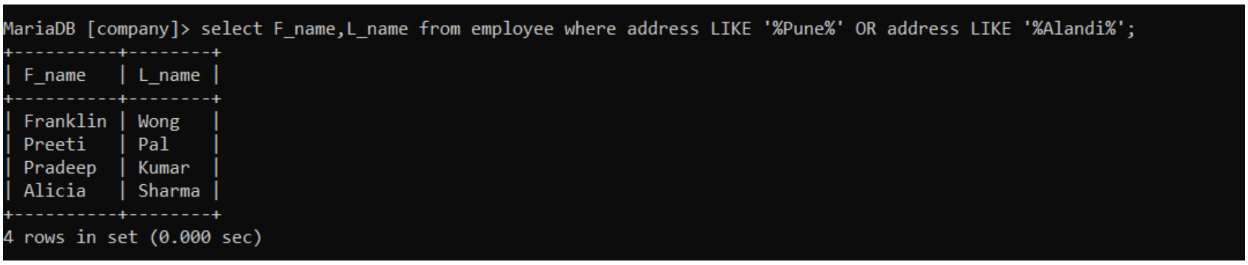
Select F_name from Employee where F_name LIKE 'P%';



```
MariaDB [company]> Select F_name from Employee where F_name LIKE 'P%';
+---------+
| F_name  |
+---------+
| Preeti  |
| Pradeep |
+---------+
2 rows in set (0.000 sec)
```

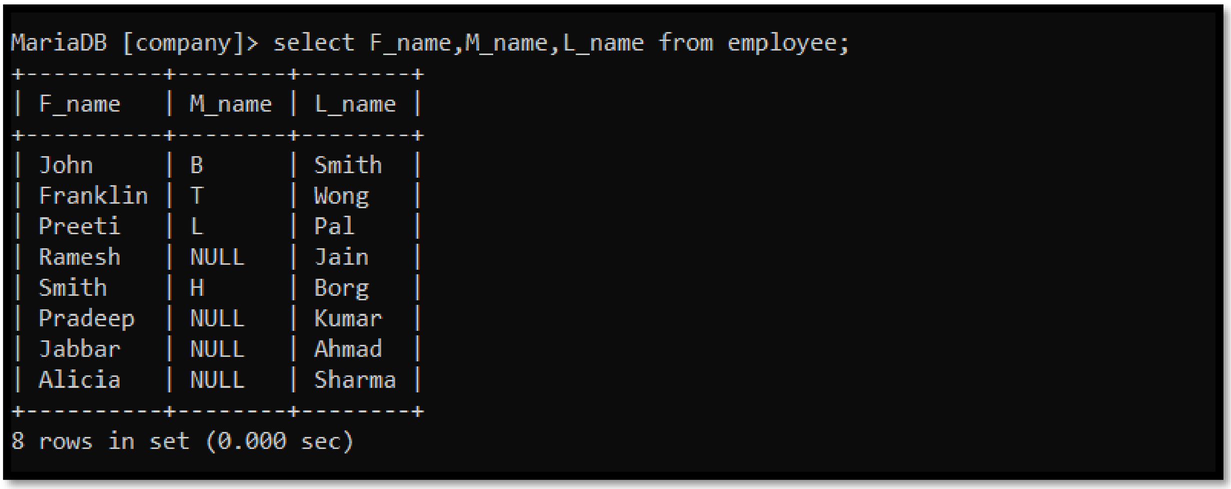
Q11. Retrieve all employees who reside in Pune or in Alandi.
Sol:

Select F_name, L_name from employee where address LIKE '%Pune%' OR address LIKE '%Alandi%';



```
MariaDB [company]> select F_name,L_name from employee where address LIKE '%Pune%' OR address LIKE '%Alandi%';
+----------+--------+
| F_name   | L_name |
+----------+--------+
| Franklin | Wong   |
| Preeti   | Pal    |
| Pradeep  | Kumar  |
| Alicia   | Sharma |
+----------+--------+
4 rows in set (0.000 sec)
```

Q12. Display the full name of all the employees.
Sol
Select F_name, M_name, L_name from employee;

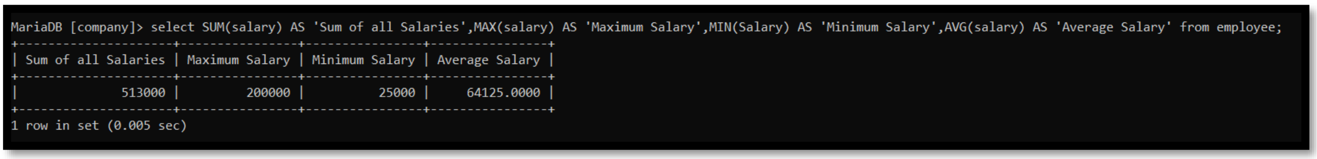


```
MariaDB [company]> select F_name,M_name,L_name from employee;
+----------+--------+--------+
| F_name   | M_name | L_name |
+----------+--------+--------+
| John     | B      | Smith  |
| Franklin | T      | Wong   |
| Preeti   | L      | Pal    |
| Ramesh   | NULL   | Jain   |
| Smith    | H      | Borg   |
| Pradeep  | NULL   | Kumar  |
| Jabbar   | NULL   | Ahmad  |
| Alicia   | NULL   | Sharma |
+----------+--------+--------+
8 rows in set (0.000 sec)
```

TASK-2)

Q1. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary.
Sol:

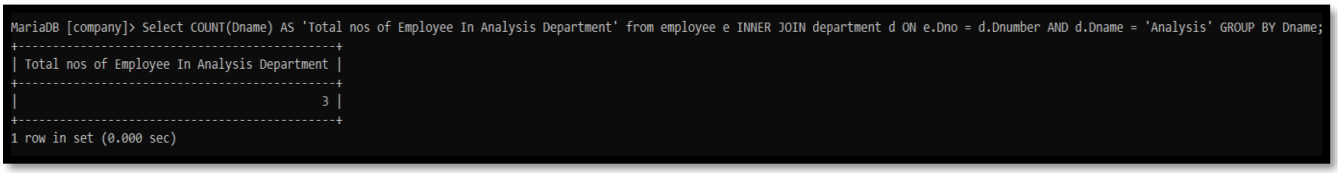
Select SUM(salary) AS 'Sum of all Salaries', MAX(salary) AS 'Maximum Salary', MIN(Salary) AS 'Minimum Salary', AVG(salary) AS 'Average Salary' from employee;



```
MariaDB [company]> select SUM(salary) AS 'Sum of all Salaries',MAX(salary) AS 'Maximum Salary',MIN(Salary) AS 'Minimum Salary',AVG(salary) AS 'Average Salary' from employee;
+---------------------+----------------+----------------+----------------+
| Sum of all Salaries | Maximum Salary | Minimum Salary | Average Salary |
+---------------------+----------------+----------------+----------------+
|              513000 |         200000 |          25000 |     64125.0000 |
+---------------------+----------------+----------------+----------------+
1 row in set (0.005 sec)
```

Q2. Retrieve the total number of employees in the analysis department.
Sol:

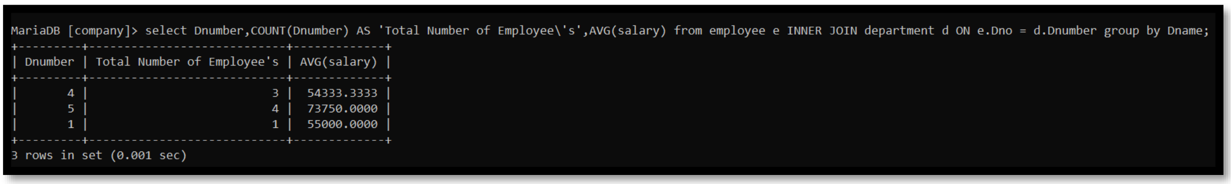
Select COUNT(Dname) AS 'Total nos of Employee In Analysis Department' from employee e INNER JOIN department d ON e.Dno = d.Dnumber AND d.Dname = 'Analysis' GROUP BY Dname;



```
MariaDB [company]> Select COUNT(Dname) AS 'Total nos of Employee In Analysis Department' from employee e INNER JOIN department d ON e.Dno = d.Dnumber AND d.Dname = 'Analysis' GROUP BY Dname;
+----------------------------------------------+
| Total nos of Employee In Analysis Department |
+----------------------------------------------+
|                                            3 |
+----------------------------------------------+
1 row in set (0.000 sec)
```

Q3. For each department retrieve the department number, the number of employees in the department and their average salary.
Sol:

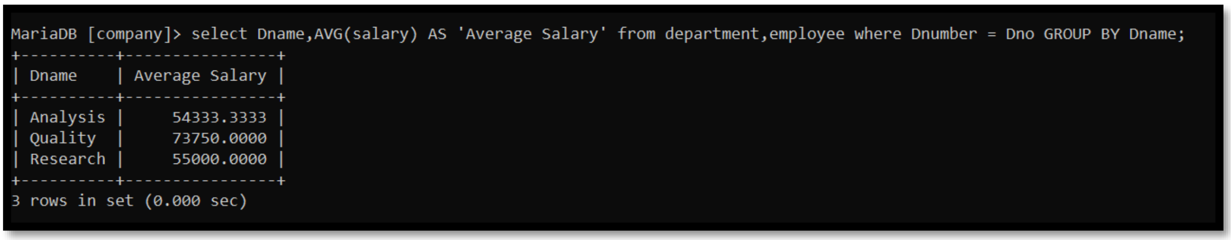
Select Dnumber, COUNT(Dnumber) AS 'Total Number of Employee\'s', AVG(salary) from employee e INNER JOIN department d ON e.Dno = d.Dnumber group by Dname;



```
MariaDB [company]> select Dnumber,COUNT(Dnumber) AS 'Total Number of Employee\'s',AVG(salary) from employee e INNER JOIN department d ON e.Dno = d.Dnumber group by Dname;
+---------+---------------------------+------------+
| Dnumber | Total Number of Employee's | AVG(salary) |
+---------+---------------------------+------------+
|       4 |                         3 |  54333.3333 |
|       5 |                         4 |  73750.0000 |
|       1 |                         1 |  55000.0000 |
+---------+---------------------------+------------+
3 rows in set (0.001 sec)
```

Q4. For each department retrieve the department name and the average salary.
Sol:

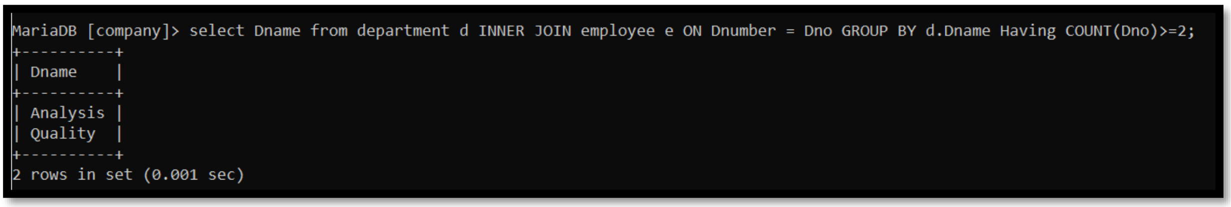
Select Dname, AVG(salary) AS 'Average Salary' from department, employee where Dnumber = Dno GROUP BY Dname;



```
MariaDB [company]> select Dname,AVG(salary) AS 'Average Salary' from department,employee where Dnumber = Dno GROUP BY Dname;
+----------+----------------+
| Dname    | Average Salary |
+----------+----------------+
| Analysis |     54333.3333 |
| Quality  |     73750.0000 |
| Research |     55000.0000 |
+----------+----------------+
3 rows in set (0.000 sec)
```

Q5. Retrieve the name of the department with a minimum of 2 employees.
Sol:

Select Dname from department d INNER JOIN employee e ON Dnumber = Dno GROUP BY d.Dname Having COUNT(Dno)>=2;



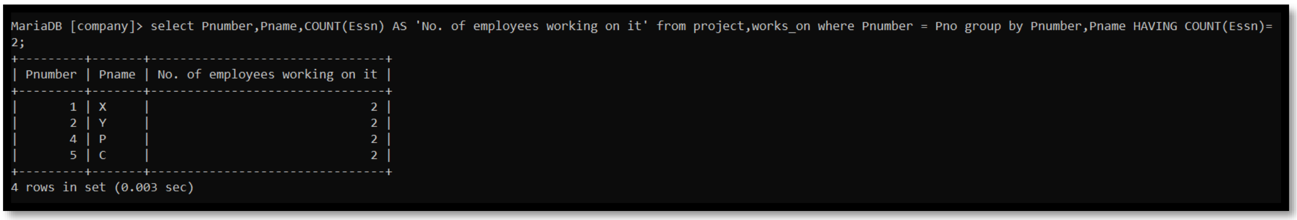
```
MariaDB [company]> select Dname from department d INNER JOIN employee e ON Dnumber = Dno GROUP BY d.Dname Having COUNT(Dno)>=2;
+----------+
| Dname    |
+----------+
| Analysis |
| Quality  |
+----------+
2 rows in set (0.001 sec)
```

Q6. For each project on which more than 2 employees work to retrieve the project number, project name and the number of employees who work on the project.
Sol:

SELECT Pnumber,Pname,COUNT(Essn) AS 'No. of employees working on it' from project,works_on where Pnumber = Pno group by Pnumber,Pname HAVING COUNT(Essn)=2;

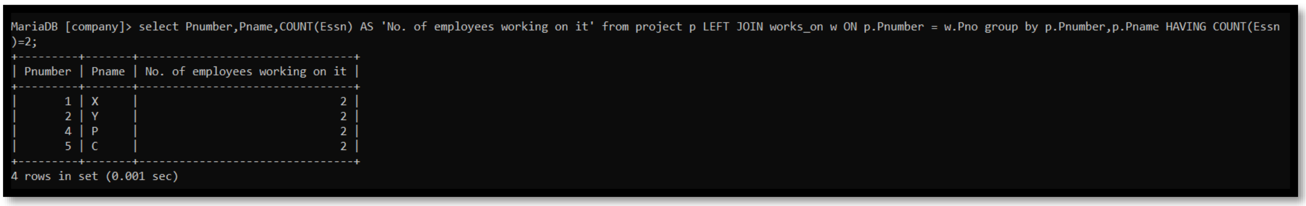
OR

Select Pnumber,Pname,COUNT(Essn) AS 'No. of employees working on it' from project p LEFT JOIN works_on w ON p.Pnumber = w.Pno group by p.Pnumber,p.Pname HAVING COUNT(Essn)=2;



```
MariaDB [company]> select Pnumber,Pname,COUNT(Essn) AS 'No. of employees working on it' from project,works_on where Pnumber = Pno group by Pnumber,Pname HAVING COUNT(Essn)=
2;
+---------+-------+--------------------------------+
| Pnumber | Pname | No. of employees working on it |
+---------+-------+--------------------------------+
|       1 | X     |                              2 |
|       2 | Y     |                              2 |
|       4 | P     |                              2 |
|       5 | C     |                              2 |
+---------+-------+--------------------------------+
4 rows in set (0.003 sec)
```

OR

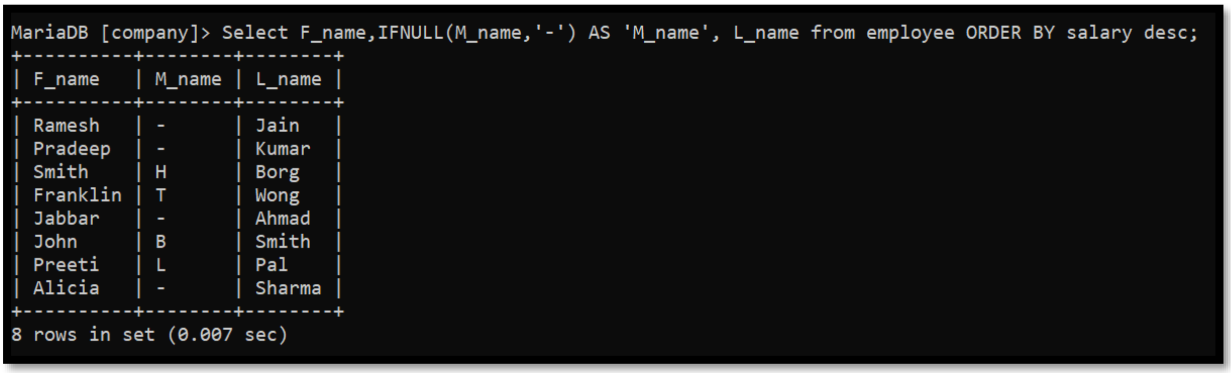


```
MariaDB [company]> select Pnumber,Pname,COUNT(Essn) AS 'No. of employees working on it' from project p LEFT JOIN works_on w ON p.Pnumber = w.Pno group by p.Pnumber,p.Pname HAVING COUNT(Essn
)=2;
+---------+-------+--------------------------------+
| Pnumber | Pname | No. of employees working on it |
+---------+-------+--------------------------------+
|       1 | X     |                              2 |
|       2 | Y     |                              2 |
|       4 | P     |                              2 |
|       5 | C     |                              2 |
+---------+-------+--------------------------------+
4 rows in set (0.001 sec)
```

Q7. Display all the employees according to their highest salary.

Sol:

Select F_name, IFNULL(M_name,'-') AS 'M_name', L_name from employee ORDER BY salary desc;

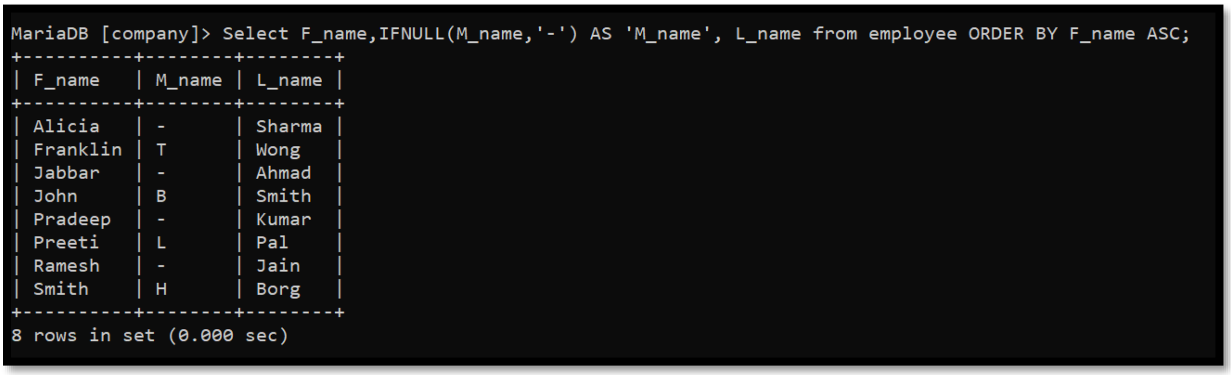


```
MariaDB [company]> Select F_name,IFNULL(M_name,'-') AS 'M_name', L_name from employee ORDER BY salary desc;
+----------+--------+--------+
| F_name   | M_name | L_name |
+----------+--------+--------+
| Ramesh   | -      | Jain   |
| Pradeep  | -      | Kumar  |
| Smith    | H      | Borg   |
| Franklin | T      | Wong   |
| Jabbar   | -      | Ahmad  |
| John     | B      | Smith  |
| Preeti   | L      | Pal    |
| Alicia   | -      | Sharma |
+----------+--------+--------+
8 rows in set (0.007 sec)
```

Q8. Display the entire employee name in alphabetical order.

Sol:

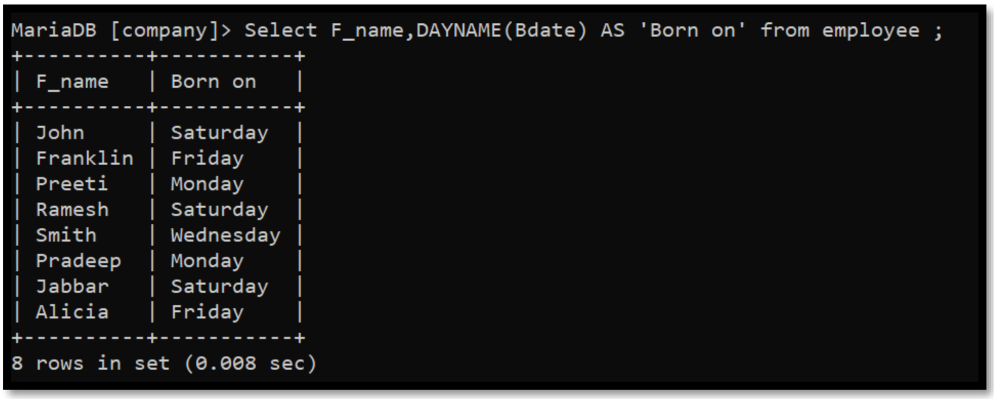
Select F_name, IFNULL(M_name,'-') AS 'M_name', L_name from employee ORDER BY F_name ASC;



```
MariaDB [company]> Select F_name,IFNULL(M_name,'-') AS 'M_name', L_name from employee ORDER BY F_name ASC;
+----------+--------+--------+
| F_name   | M_name | L_name |
+----------+--------+--------+
| Alicia   | -      | Sharma |
| Franklin | T      | Wong   |
| Jabbar   | -      | Ahmad  |
| John     | B      | Smith  |
| Pradeep  | -      | Kumar  |
| Preeti   | L      | Pal    |
| Ramesh   | -      | Jain   |
| Smith    | H      | Borg   |
+----------+--------+--------+
8 rows in set (0.000 sec)
```

Q9. Retrieve the first name and the day name of the birth of all the employees.
Sol:

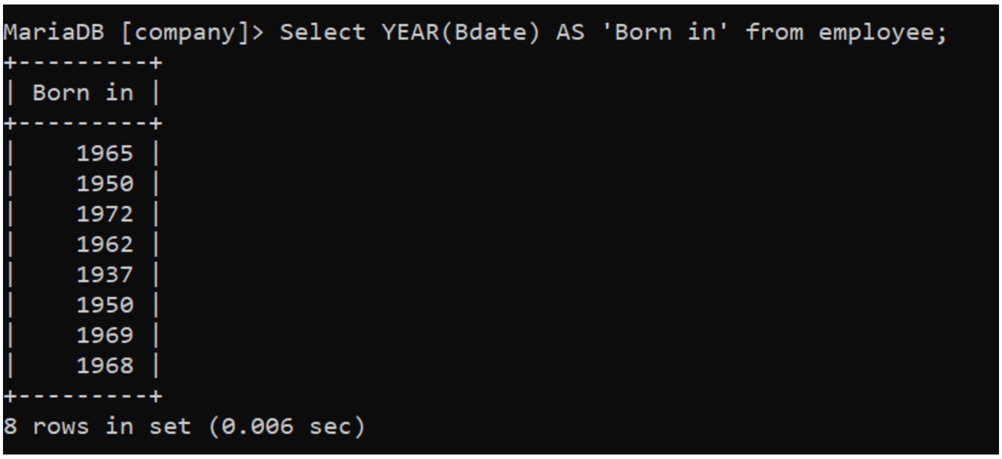
Select F_name, DAYNAME(Bdate) AS 'Born on' from employee;



```
MariaDB [company]> Select F_name,DAYNAME(Bdate) AS 'Born on' from employee ;
+----------+-----------+
| F_name   | Born on   |
+----------+-----------+
| John     | Saturday  |
| Franklin | Friday    |
| Preeti   | Monday    |
| Ramesh   | Saturday  |
| Smith    | Wednesday |
| Pradeep  | Monday    |
| Jabbar   | Saturday  |
| Alicia   | Friday    |
+----------+-----------+
8 rows in set (0.008 sec)
```

Q10. Retrieve the birth year of all employees.
Sol:

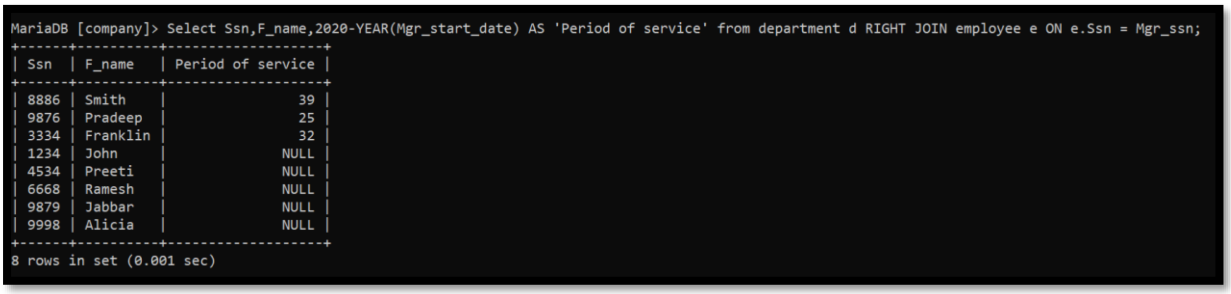
Select YEAR(Bdate) AS 'Born in' from employee;



```
MariaDB [company]> Select YEAR(Bdate) AS 'Born in' from employee;
+---------+
| Born in |
+---------+
|    1965 |
|    1950 |
|    1972 |
|    1962 |
|    1937 |
|    1950 |
|    1969 |
|    1968 |
+---------+
8 rows in set (0.006 sec)
```

Q11. Retrieve the period of service of all the employees as manager of the company.
Sol:

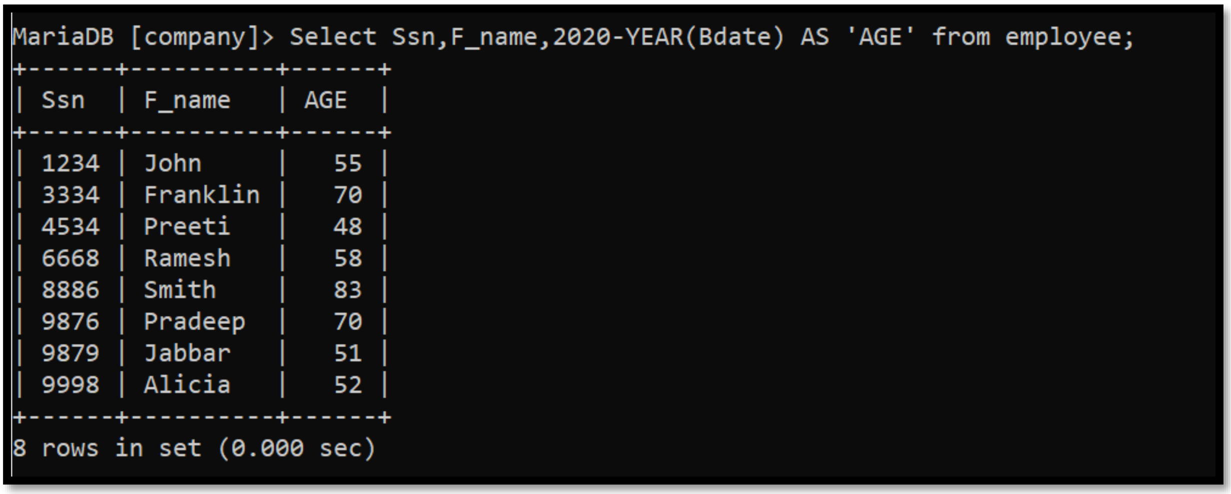
Select Ssn, F_name,2020-YEAR(Mgr_start_date) AS 'Period of service' from department d RIGHT JOIN employee e ON e.Ssn = Mgr_ssn;



```
MariaDB [company]> Select Ssn,F_name,2020-YEAR(Mgr_start_date) AS 'Period of service' from department d RIGHT JOIN employee e ON e.Ssn = Mgr_ssn;
+------+----------+-------------------+
| Ssn  | F_name   | Period of service |
+------+----------+-------------------+
| 8886 | Smith    |                39 |
| 9876 | Pradeep  |                25 |
| 3334 | Franklin |                32 |
| 1234 | John     |              NULL |
| 4534 | Preeti   |              NULL |
| 6668 | Ramesh   |              NULL |
| 9879 | Jabbar   |              NULL |
| 9998 | Alicia   |              NULL |
+------+----------+-------------------+
8 rows in set (0.001 sec)
```

Q12. Retrieve the age of all the employees of the company.
Sol:

Select Ssn, F_name,2020-YEAR(Bdate) AS 'AGE' from employee;



```
MariaDB [company]> Select Ssn,F_name,2020-YEAR(Bdate) AS 'AGE' from employee;
+------+----------+------+
| Ssn  | F_name   | AGE  |
+------+----------+------+
| 1234 | John     |   55 |
| 3334 | Franklin |   70 |
| 4534 | Preeti   |   48 |
| 6668 | Ramesh   |   58 |
| 8886 | Smith    |   83 |
| 9876 | Pradeep  |   70 |
| 9879 | Jabbar   |   51 |
| 9998 | Alicia   |   52 |
+------+----------+------+
8 rows in set (0.000 sec)
```

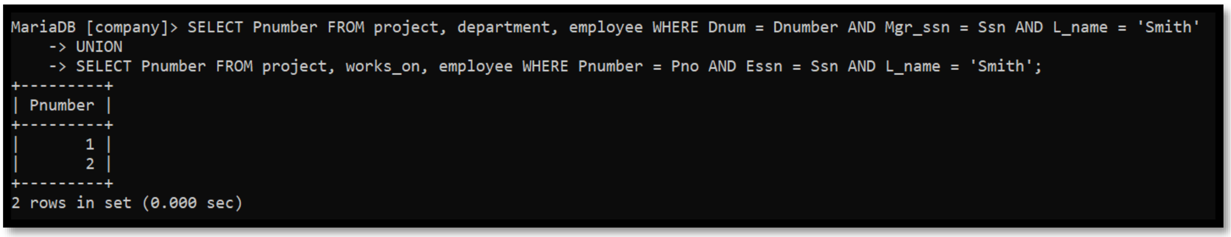
TASK-3

Q1. Make the 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.
Sol:

SELECT Pnumber FROM project, department, employee WHERE Dnum = Dnumber AND Mgr_ssn = Ssn AND L_name = 'Smith'

UNION

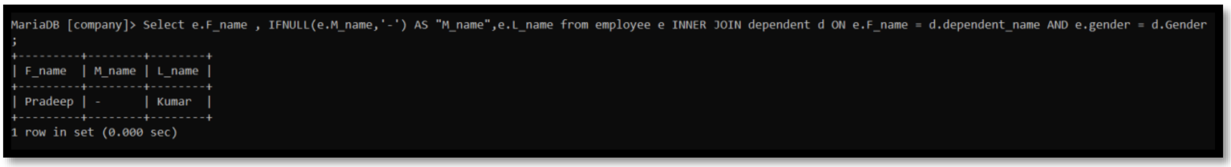
SELECT Pnumber FROM project, works_on, employee WHERE Pnumber = Pno AND Essn = Ssn AND L_name = 'Smith';



```
MariaDB [company]> SELECT Pnumber FROM project, department, employee WHERE Dnum = Dnumber AND Mgr_ssn = Ssn AND L_name = 'Smith'
    -> UNION
    -> SELECT Pnumber FROM project, works_on, employee WHERE Pnumber = Pno AND Essn = Ssn AND L_name = 'Smith';
+---------+
| Pnumber |
+---------+
|       1 |
|       2 |
+---------+
2 rows in set (0.000 sec)
```

Q2. Retrieve the name of each employee who has dependent with the same first name and same gender as the employee.
Sol:

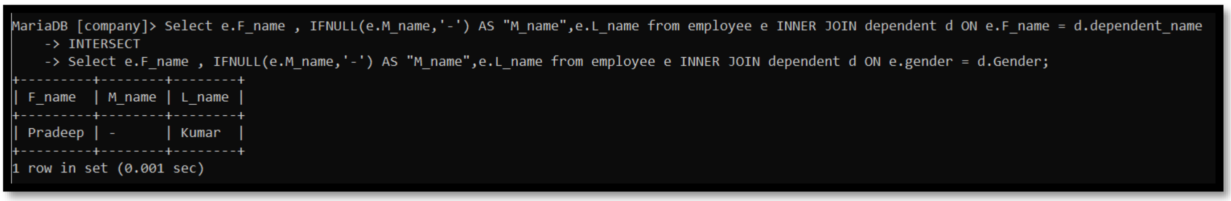
Select e.F_name , IFNULL(e.M_name,'-') AS "M_name",e.L_name from employee e INNER JOIN dependent d ON e.F_name = d.dependent_name AND e.gender = d.Gender;



```
MariaDB [company]> Select e.F_name , IFNULL(e.M_name,'-') AS "M_name",e.L_name from employee e INNER JOIN dependent d ON e.F_name = d.dependent_name AND e.gender = d.Gender
;
+---------+--------+--------+
| F_name  | M_name | L_name |
+---------+--------+--------+
| Pradeep | -      | Kumar  |
+---------+--------+--------+
1 row in set (0.000 sec)
```

OR

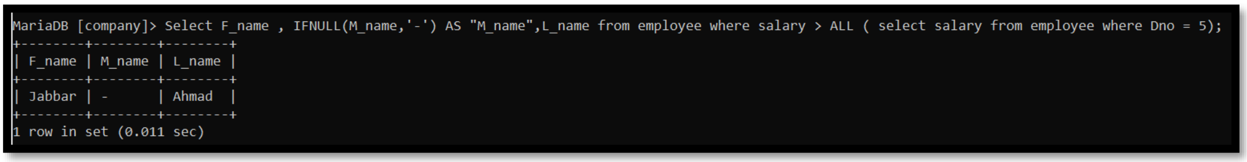
Select e.F_name , IFNULL(e.M_name,'-') AS "M_name",e.L_name from employee e
INNER JOIN dependent d ON e.F_name = d.dependent_name
INTERSECT
Select e.F_name , IFNULL(e.M_name,'-') AS "M_name",e.L_name from employee e
INNER JOIN dependent d ON e.gender = d.Gender;



```
MariaDB [company]> Select e.F_name , IFNULL(e.M_name,'-') AS "M_name",e.L_name from employee e INNER JOIN dependent d ON e.F_name = d.dependent_name
    -> INTERSECT
    -> Select e.F_name , IFNULL(e.M_name,'-') AS "M_name",e.L_name from employee e INNER JOIN dependent d ON e.gender = d.Gender;
+---------+--------+--------+
| F_name  | M_name | L_name |
+---------+--------+--------+
| Pradeep | -      | Kumar  |
+---------+--------+--------+
1 row in set (0.001 sec)
```

Q3. Find the name of employees whose salary is greater than the salary of all the department number 5.
Sol:

Select F_name, IFNULL(M_name,'-') AS "M_name", L_name from employee where salary > ALL (select salary from employee where Dno = 5);



```
MariaDB [company]> Select F_name , IFNULL(M_name,'-') AS "M_name",L_name from employee where salary > ALL ( select salary from employee where Dno = 5);
+--------+--------+--------+
| F_name | M_name | L_name |
+--------+--------+--------+
| Jabbar | -      | Ahmad  |
+--------+--------+--------+
1 row in set (0.011 sec)
```

Q4. Retrieve the social security number of all the employees who work on the project number 1,2 and 3.
Sol:

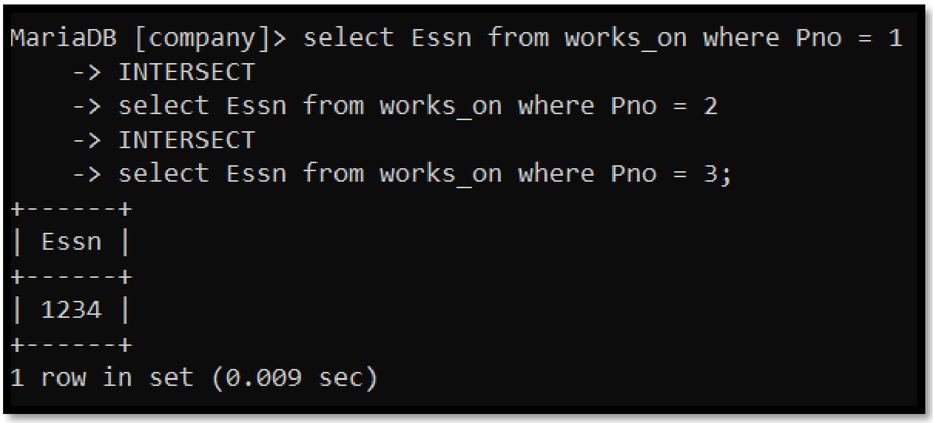
Select Essn from works_on where Pno = 1

INTERSECT

Select Essn from works_on where Pno = 2

INTERSECT

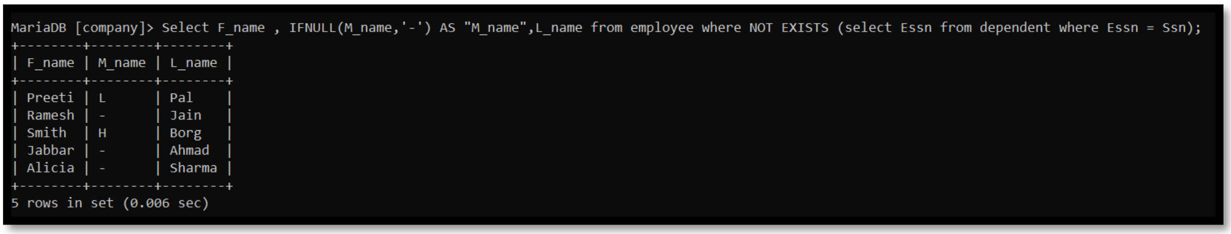
Select Essn from works_on where Pno = 3;



```
MariaDB [company]> select Essn from works_on where Pno = 1
    -> INTERSECT
    -> select Essn from works_on where Pno = 2
    -> INTERSECT
    -> select Essn from works_on where Pno = 3;
+------+
| Essn |
+------+
| 1234 |
+------+
1 row in set (0.009 sec)
```

Q5. Retrieve the name of the employees who have no dependence.
Sol:

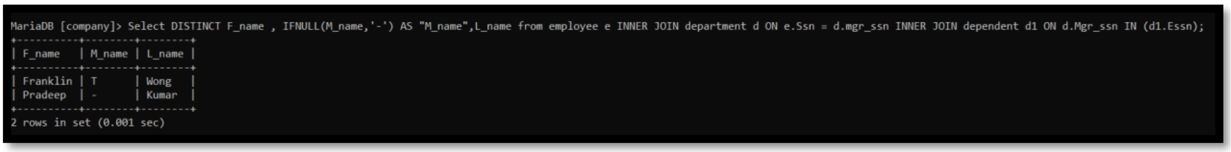
Select F_name, IFNULL(M_name,'-') AS "M_name", L_name from employee   where NOT EXISTS (select Essn from dependent where Essn = Ssn);



```
MariaDB [company]> Select F_name , IFNULL(M_name,'-') AS "M_name",L_name from employee where NOT EXISTS (select Essn from dependent where Essn = Ssn);
+--------+--------+--------+
| F_name | M_name | L_name |
+--------+--------+--------+
| Preeti | L      | Pal    |
| Ramesh | -      | Jain   |
| Smith  | H      | Borg   |
| Jabbar | -      | Ahmad  |
| Alicia | -      | Sharma |
+--------+--------+--------+
5 rows in set (0.006 sec)
```

Q6. List the names of the manager who has atleast one dependent.
Sol:

Select DISTINCT F_name , IFNULL(M_name,'-') AS "M_name",L_name from employee e INNER JOIN department d ON e.Ssn = d.mgr_ssn INNER JOIN dependent d1 ON d.Mgr_ssn IN (d1.Essn);



```
MariaDB [company]> Select DISTINCT F_name , IFNULL(M_name,'-') AS "M_name",L_name from employee e INNER JOIN department d ON e.Ssn = d.mgr_ssn INNER JOIN dependent d1 ON d.Mgr_ssn IN (d1.Essn);
+----------+--------+--------+
| F_name   | M_name | L_name |
+----------+--------+--------+
| Franklin | T      | Wong   |
| Pradeep  | -      | Kumar  |
+----------+--------+--------+
2 rows in set (0.001 sec)
```

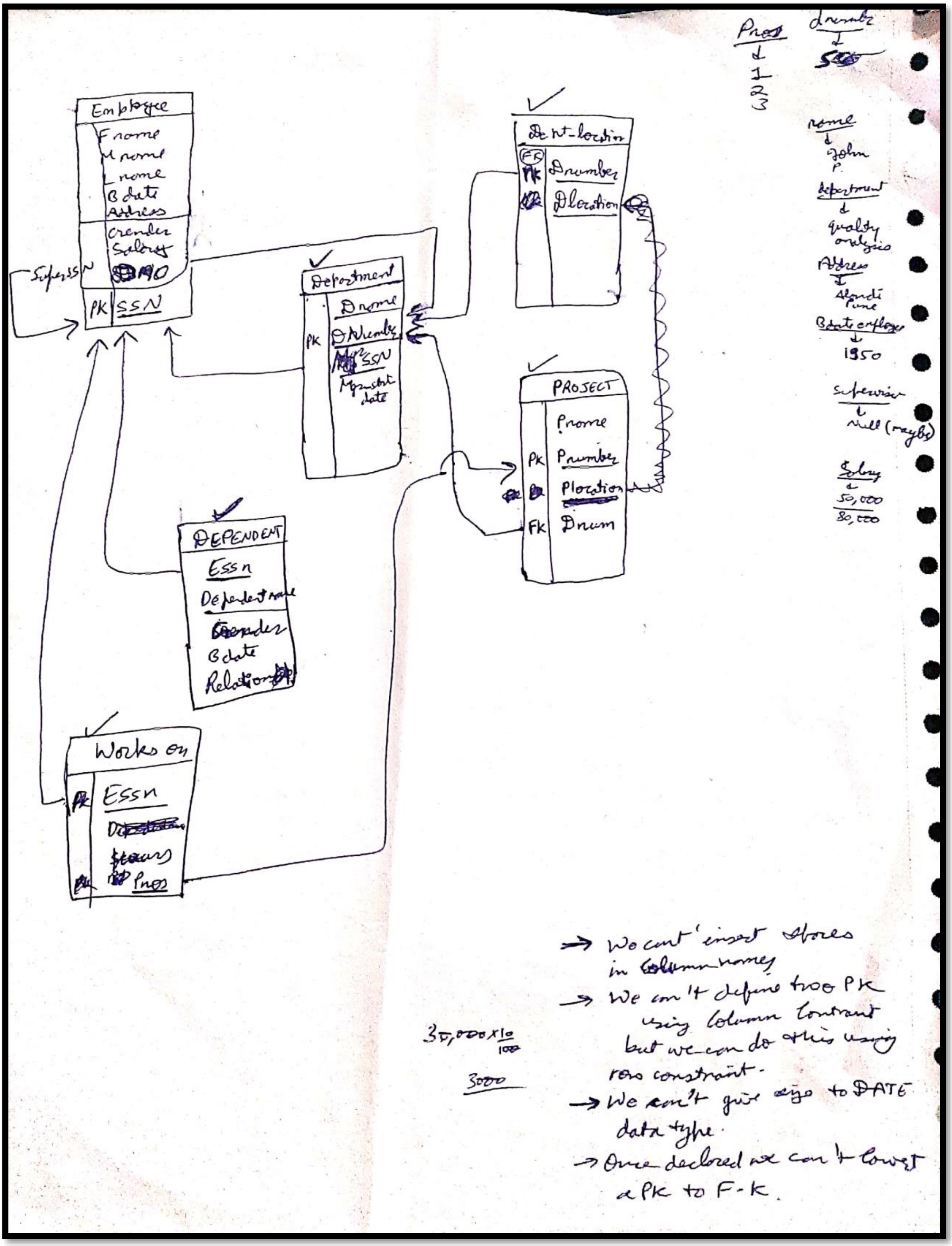
❖ DCL command:

CREATE VIEW AS
Select Ssn, F_name, M_name, L_name, Bdate, gender, Dno, Super_ssn from employee;

❖ Conclusion (Part-3):

S.no	Analysis
1)	FULL JOIN can’t be used in MySQL
2)	For single row subquery comparison we use comparator operator like >, <, >=, <= ,!= etc…
3)	For multi-row subquery comparison we use comparator operator like ANY, ALL, EXISTS, NOT EXISTS, IN, NOT IN etc…
4)	INTERSECT could be used in place of AND.
5)	UNION could be used in place of OR.
6)	ORDER BY cause can’t be used in subqueries.

- **Paper Work:**



→ We cant insert spaces in column names

→ We can't define two PK using column constraint but we can do this using row constraint.

→ We can't give size to DATE data type.

→ Once declared we can't convert a PK to F-K.