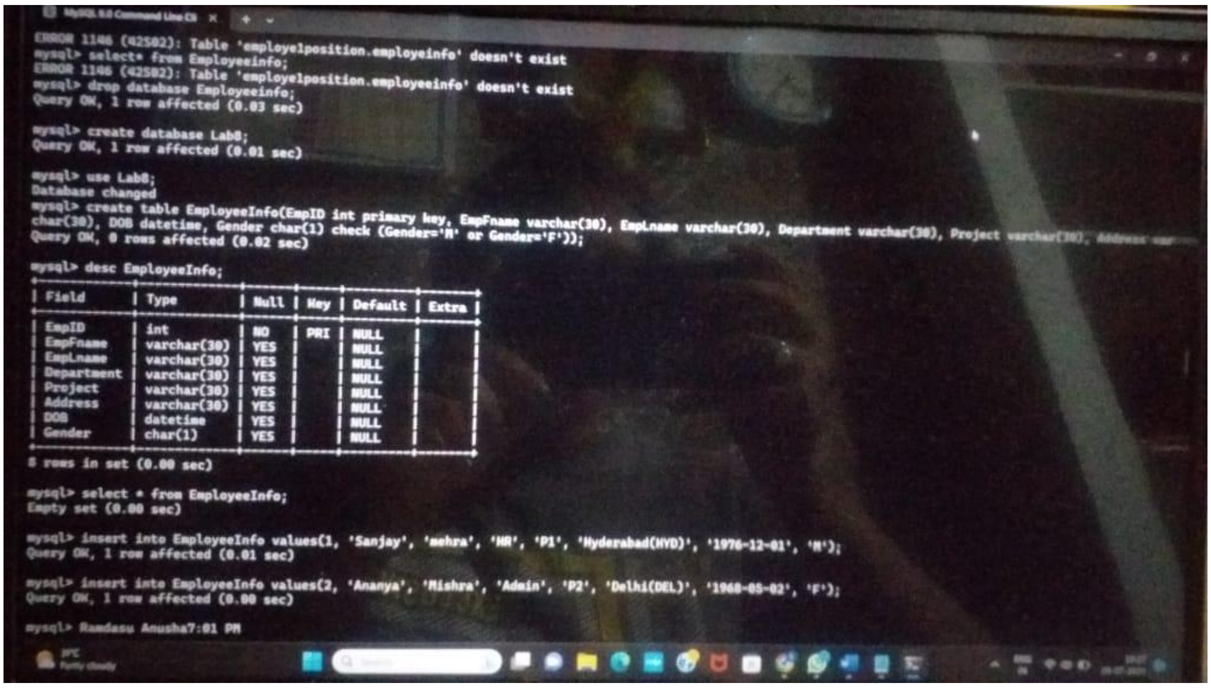
VIJAYA LAKSHMI SRAVANTI

EmployeeInfo Table:

EmpID	EmpFname	EmpLname	Department	Project	Address	DOB	Gender
1	Sanjay	Mehra	HR	P1	Hyderabad(HYD)	01/12/1976	M
2	Ananya	Mishra	Admin	P2	Delhi(DEL)	02/05/1968	F
3	Rohan	Diwan	Account	P3	Mumbai(BOM)	01/01/1980	M
4	Sonia	Kulkarni	HR	P1	Hyderabad(HYD)	02/05/1992	F
5	Ankit	Kapoor	Admin	P2	Delhi(DEL)	03/07/1994	M

- create table Employeeinfo(EMPID int primary key, EMPFNAME varchar(30), EMPLNAME varchar(30), Department varchar(30),Project varchar(30), Address varchar(30), DOB datetime, Gender char(1) check (Gender='M' or Gender='F');



```
ERROR 1146 (42S02): Table 'employelposition.employeinfo' doesn't exist
mysql> select* from Employeeinfo;
ERROR 1146 (42S02): Table 'employelposition.employeeinfo' doesn't exist
mysql> drop database Employeeinfo;
Query OK, 1 row affected (0.03 sec)

mysql> create database Lab8;
Query OK, 1 row affected (0.01 sec)

mysql> use Lab8;
Database changed
mysql> create table EmployeeInfo(EmpID int primary key, EmpFname varchar(30), EmpLname varchar(30), Department varchar(30), Project varchar(30), Address var
char(30), DOB datetime, Gender char(1) check (Gender='M' or Gender='F'));
Query OK, 0 rows affected (0.02 sec)

mysql> desc EmployeeInfo;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| EmpID      | int         | NO   | PRI | NULL    |       |
| EmpFname   | varchar(30) | YES  |     | NULL    |       |
| EmpLname   | varchar(30) | YES  |     | NULL    |       |
| Department | varchar(30) | YES  |     | NULL    |       |
| Project    | varchar(30) | YES  |     | NULL    |       |
| Address    | varchar(30) | YES  |     | NULL    |       |
| DOB        | datetime    | YES  |     | NULL    |       |
| Gender     | char(1)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

mysql> select * from EmployeeInfo;
Empty set (0.00 sec)

mysql> insert into EmployeeInfo values(1, 'Sanjay', 'mehra', 'HR', 'P1', 'Hyderabad(HYD)', '1976-12-01', 'M');
Query OK, 1 row affected (0.01 sec)

mysql> insert into EmployeeInfo values(2, 'Ananya', 'Mishra', 'Admin', 'P2', 'Delhi(DEL)', '1968-05-02', 'F');
Query OK, 1 row affected (0.00 sec)

mysql> Ramdasu Anusha7:01 PM
```

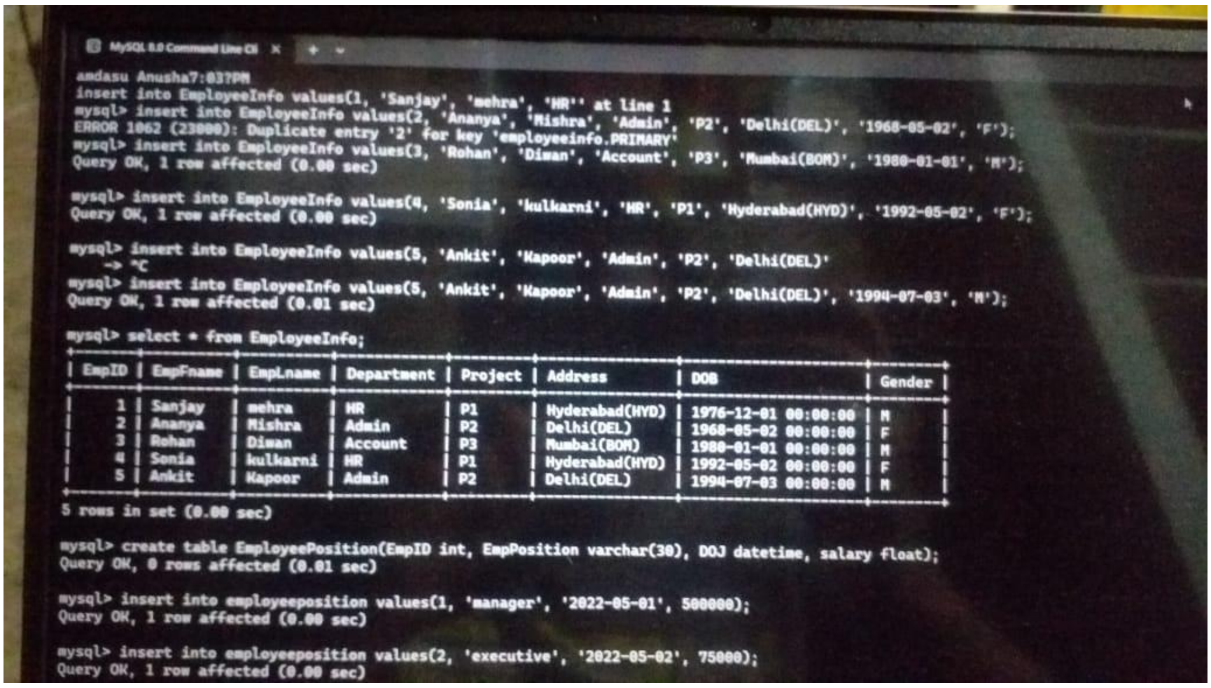
insert into Employeeinfo values(1, 'SANJAY', 'MEHRA', 'HR', 'P1', 'Hyderabad(HYD)', '1976-12-01', 'M');

· insert into Employeeinfo values(2, 'ANANYA', 'MISHRA', 'ADMIN', 'P2', 'Delhi(DEL)', '1968-05-02', 'F');

· insert into Employeeinfo values(3, 'ROHAN', 'DIWAN', 'ACCOUNT', 'P3', 'Mumbai(BOM)', '1980-01-01', 'M');

· insert into Employeeinfo values(4, 'SONIA', 'kULKARNI', 'HR', 'P1', 'Hyderabad(HYD)', '1992-05-02', 'F');

- insert into Employeeinfo values(5, 'ANKIT', 'KAPOOR', 'ADMIN', 'P2', 'Delhi(DEL)', '1994-07-03', 'M');



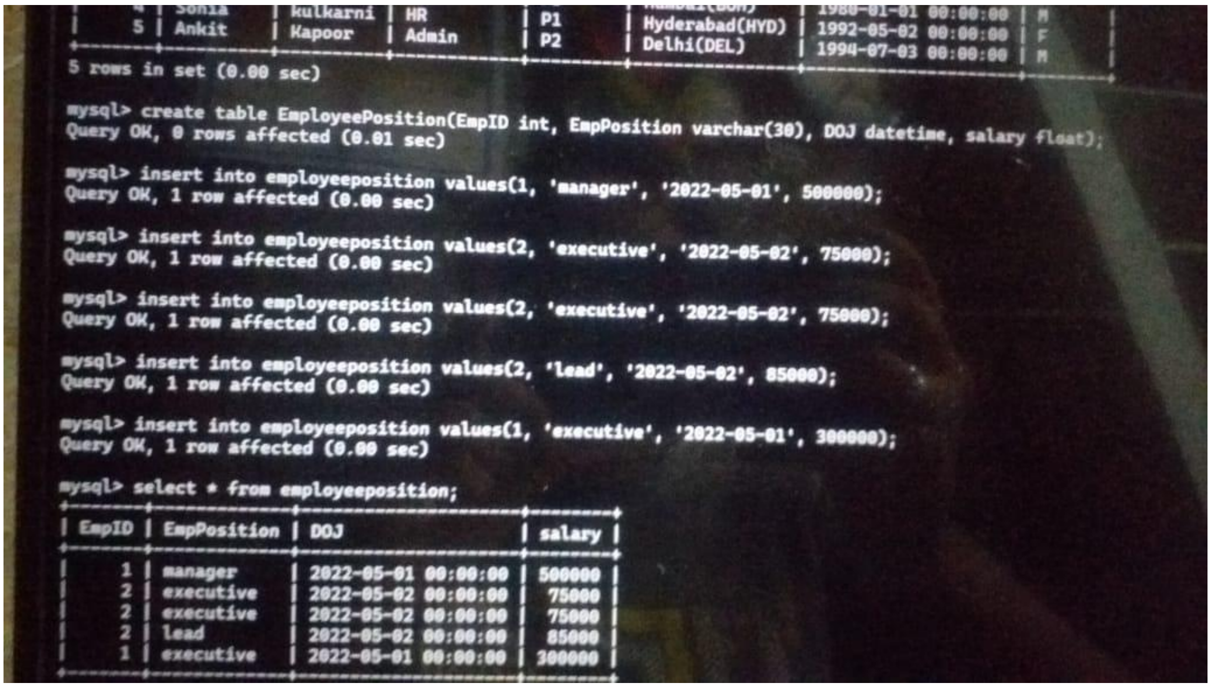
```
amdasu Anusha7:03?PM
insert into EmployeeInfo values(1, 'Sanjay', 'mehra', 'HR'' at line 1
mysql> insert into EmployeeInfo values(2, 'Ananya', 'Mishra', 'Admin', 'P2', 'Delhi(DEL)', '1968-05-02', 'F');
ERROR 1062 (23000): Duplicate entry '2' for key 'employeeinfo.PRIMARY'
mysql> insert into EmployeeInfo values(3, 'Rohan', 'Diwan', 'Account', 'P3', 'Mumbai(BOM)', '1980-01-01', 'M');
Query OK, 1 row affected (0.00 sec)

mysql> insert into EmployeeInfo values(4, 'Sonia', 'kulkarni', 'HR', 'P1', 'Hyderabad(HYD)', '1992-05-02', 'F');
Query OK, 1 row affected (0.00 sec)

mysql> insert into EmployeeInfo values(5, 'Ankit', 'Kapoor', 'Admin', 'P2', 'Delhi(DEL)'
    -> ^C
mysql> insert into EmployeeInfo values(5, 'Ankit', 'Kapoor', 'Admin', 'P2', 'Delhi(DEL)', '1994-07-03', 'M');
Query OK, 1 row affected (0.01 sec)

mysql> select * from EmployeeInfo;
+-------+---------+----------+------------+---------+----------------+---------------------+--------+
| EmpID | EmpFname | EmpLname | Department | Project | Address        | DOB                 | Gender |
+-------+---------+----------+------------+---------+----------------+---------------------+--------+
|     1 | Sanjay  | mehra    | HR         | P1      | Hyderabad(HYD) | 1976-12-01 00:00:00 | M      |
|     2 | Ananya  | Mishra   | Admin      | P2      | Delhi(DEL)     | 1968-05-02 00:00:00 | F      |
|     3 | Rohan   | Diwan    | Account    | P3      | Mumbai(BOM)    | 1980-01-01 00:00:00 | M      |
|     4 | Sonia   | kulkarni | HR         | P1      | Hyderabad(HYD) | 1992-05-02 00:00:00 | F      |
|     5 | Ankit   | Kapoor   | Admin      | P2      | Delhi(DEL)     | 1994-07-03 00:00:00 | M      |
+-------+---------+----------+------------+---------+----------------+---------------------+--------+
5 rows in set (0.00 sec)

mysql> create table EmployeePosition(EmpID int, EmpPosition varchar(30), DOJ datetime, salary float);
Query OK, 0 rows affected (0.01 sec)

mysql> insert into employeeposition values(1, 'manager', '2022-05-01', 500000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into employeeposition values(2, 'executive', '2022-05-02', 75000);
Query OK, 1 row affected (0.00 sec)
```

EmployeePosition Table:

EmpID	EmpPosition	DateOfJoining	Salary
1	Manager	01/05/2022	500000
2	Executive	02/05/2022	75000
3	Manager	01/05/2022	90000
2	Lead	02/05/2022	85000
1	Executive	01/05/2022	300000

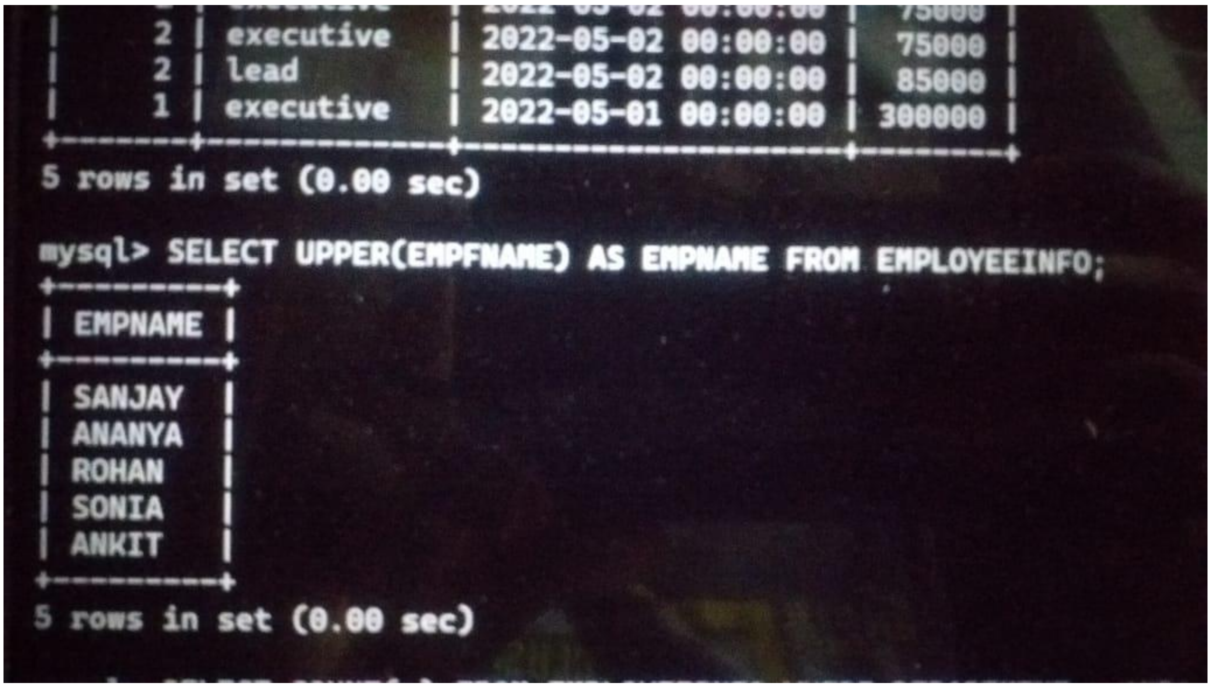
create table EmployeePosition(EmpID int, EmpPosition varchar(30), DOJ datetime, salary float);

- insert into employeeposition values(1, 'MANAGER', '2022-05-01', 500000);
- insert into employeeposition values(2, 'EXECUTIVE', '2022-05-02', 75000);
- insert into employeeposition values(3, 'MANAGER', '2022-05-01', 90000);
- insert into employeeposition values(2, 'LEAD', '2022-05-02', 85000);
- insert into employeeposition values(1, 'EXECUTIVE', '2022-05- 01', 300000);



```
|     4 | Sonia  | kulkarni | HR    |    |               | 1980-01-01 00:00:00 | M |
|     5 | Ankit  | Kapoor   | Admin | P1 | Hyderabad(HYD) | 1992-05-02 00:00:00 | F |
|       |        |          |       | P2 | Delhi(DEL)     | 1994-07-03 00:00:00 | M |
+-------+--------+----------+-------+----+----------------+---------------------+---+
5 rows in set (0.00 sec)

mysql> create table EmployeePosition(EmpID int, EmpPosition varchar(30), DOJ datetime, salary float);
Query OK, 0 rows affected (0.01 sec)

mysql> insert into employeeposition values(1, 'manager', '2022-05-01', 500000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into employeeposition values(2, 'executive', '2022-05-02', 75000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into employeeposition values(2, 'executive', '2022-05-02', 75000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into employeeposition values(2, 'lead', '2022-05-02', 85000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into employeeposition values(1, 'executive', '2022-05-01', 300000);
Query OK, 1 row affected (0.00 sec)

mysql> select * from employeeposition;
+-------+-------------+---------------------+--------+
| EmpID | EmpPosition | DOJ                 | salary |
+-------+-------------+---------------------+--------+
|     1 | manager     | 2022-05-01 00:00:00 | 500000 |
|     2 | executive   | 2022-05-02 00:00:00 |  75000 |
|     2 | executive   | 2022-05-02 00:00:00 |  75000 |
|     2 | lead        | 2022-05-02 00:00:00 |  85000 |
|     1 | executive   | 2022-05-01 00:00:00 | 300000 |
+-------+-------------+---------------------+--------+
```

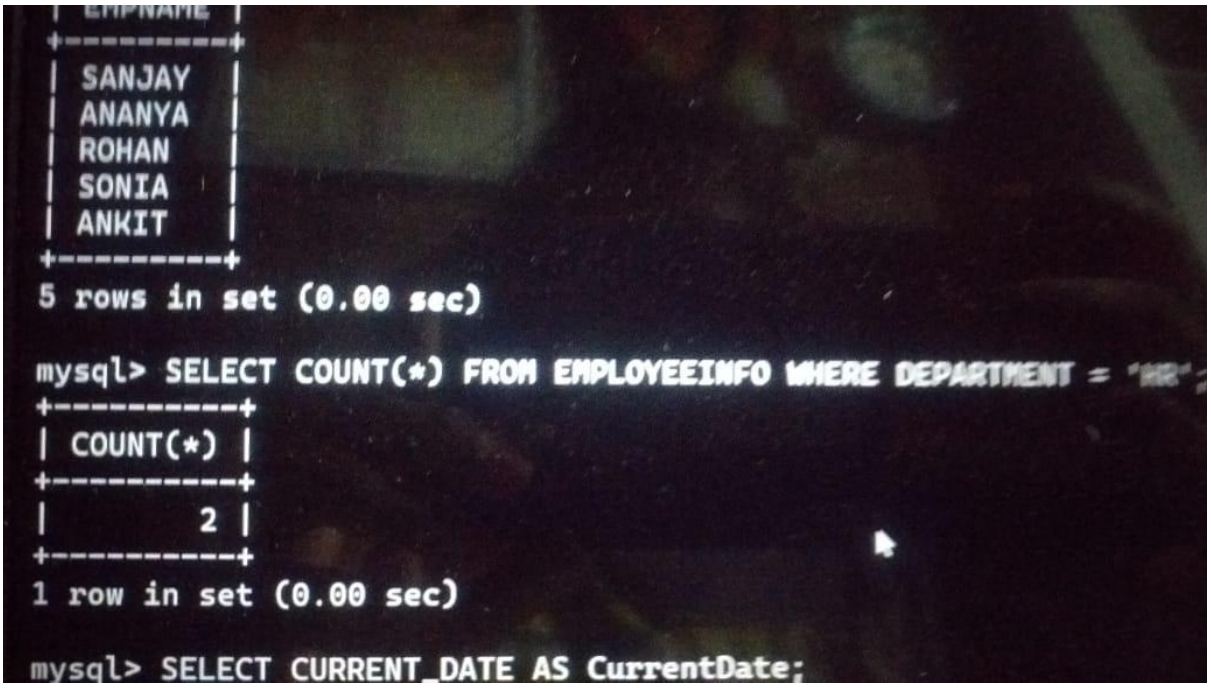
1. Write a query to fetch the EmpFname from the EmployeeInfo table in the upper case and use the ALIAS name as EmpName.



```
|        2 | executive     | 2022-05-02 00:00:00 |  75000 |
|        2 | lead          | 2022-05-02 00:00:00 |  85000 |
|        1 | executive     | 2022-05-01 00:00:00 | 300000 |
+---------+---------------+---------------------+--------+
5 rows in set (0.00 sec)

mysql> SELECT UPPER(EMPFNAME) AS EMPNAME FROM EMPLOYEEINFO;
+---------+
| EMPNAME |
+---------+
| SANJAY  |
| ANANYA  |
| ROHAN   |
| SONIA   |
| ANKIT   |
+---------+
5 rows in set (0.00 sec)
```

Query:

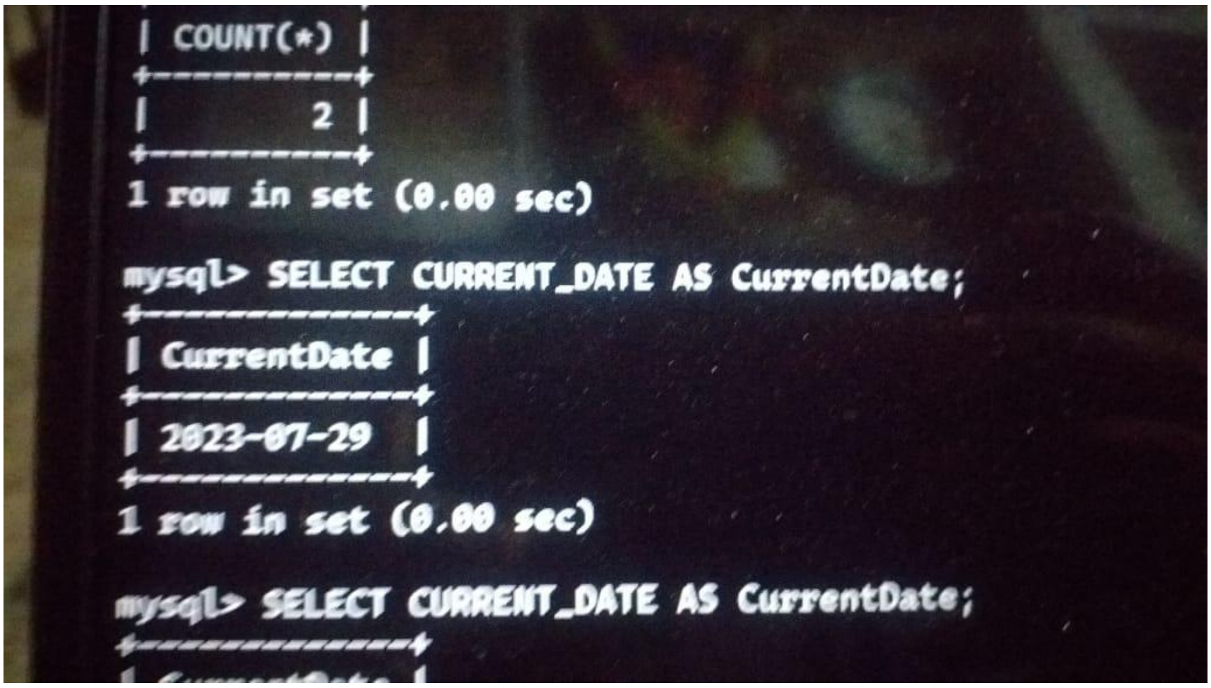
2. Write a query to fetch the number of employees working in the department 'HR'.



```
| EMPNAME  |
+----------+
| SANJAY   |
| ANANYA   |
| ROHAN    |
| SONIA    |
| ANKIT    |
+----------+
5 rows in set (0.00 sec)

mysql> SELECT COUNT(*) FROM EMPLOYEEINFO WHERE DEPARTMENT = 'HR';
+----------+
| COUNT(*) |
+----------+
|        2 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT CURRENT_DATE AS CurrentDate;
```

Query:

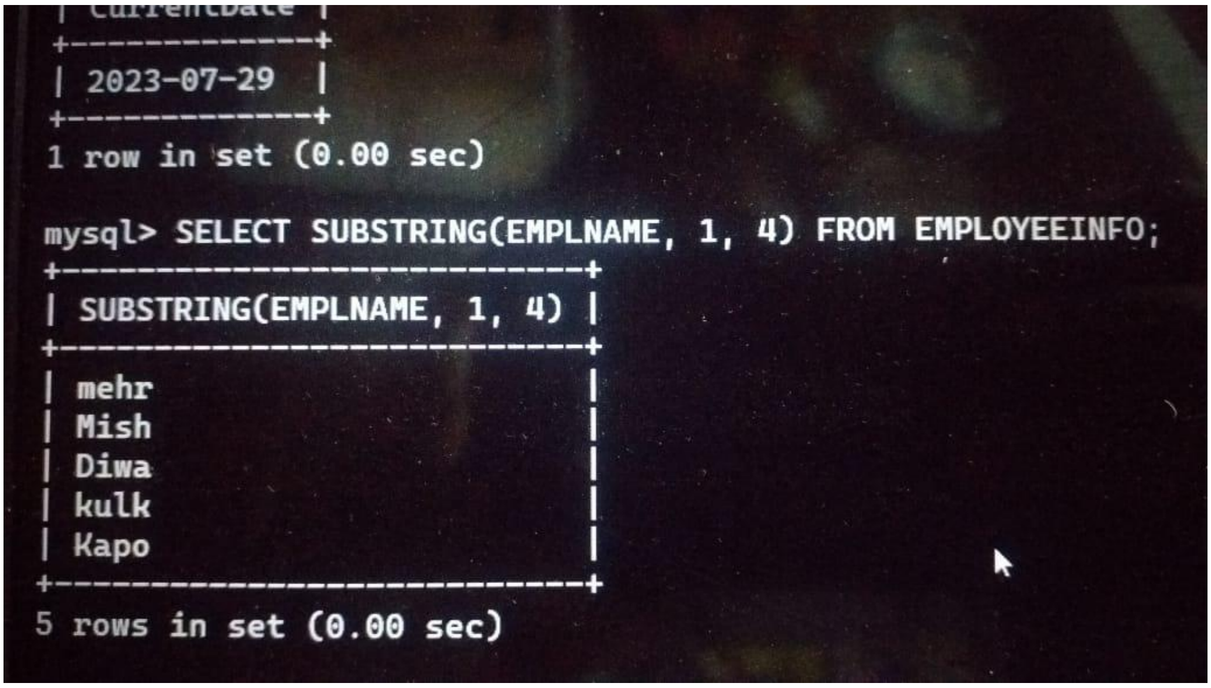
3. Write a query to get the current date.



```
| COUNT(*) |
+----------+
|        2 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT CURRENT_DATE AS CurrentDate;
+-------------+
| CurrentDate |
+-------------+
| 2023-07-29  |
+-------------+
1 row in set (0.00 sec)

mysql> SELECT CURRENT_DATE AS CurrentDate;
+-------------+
```

Query:

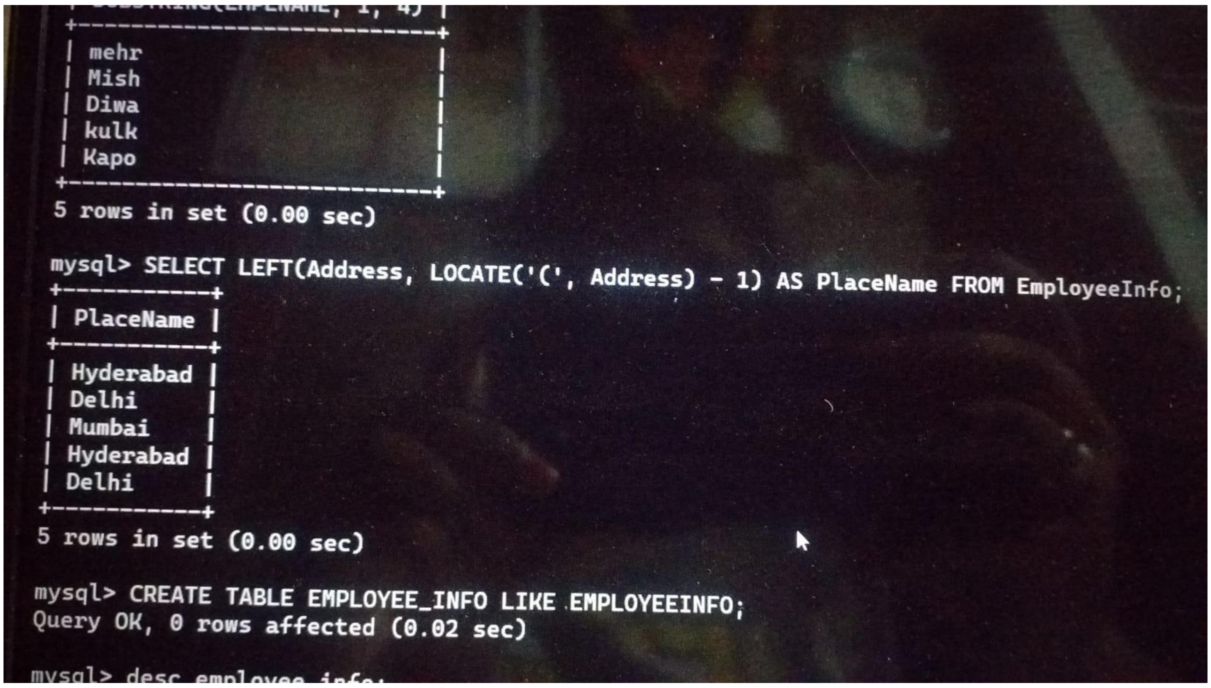
4. Write a query to retrieve the first four characters of EmpLname from the EmployeeInfo table.



```
+-------------+
| 2023-07-29  |
+-------------+
1 row in set (0.00 sec)

mysql> SELECT SUBSTRING(EMPLNAME, 1, 4) FROM EMPLOYEEINFO;
+---------------------------+
| SUBSTRING(EMPLNAME, 1, 4) |
+---------------------------+
| mehr                      |
| Mish                      |
| Diwa                      |
| kulk                      |
| Kapo                      |
+---------------------------+
5 rows in set (0.00 sec)
```

Query:

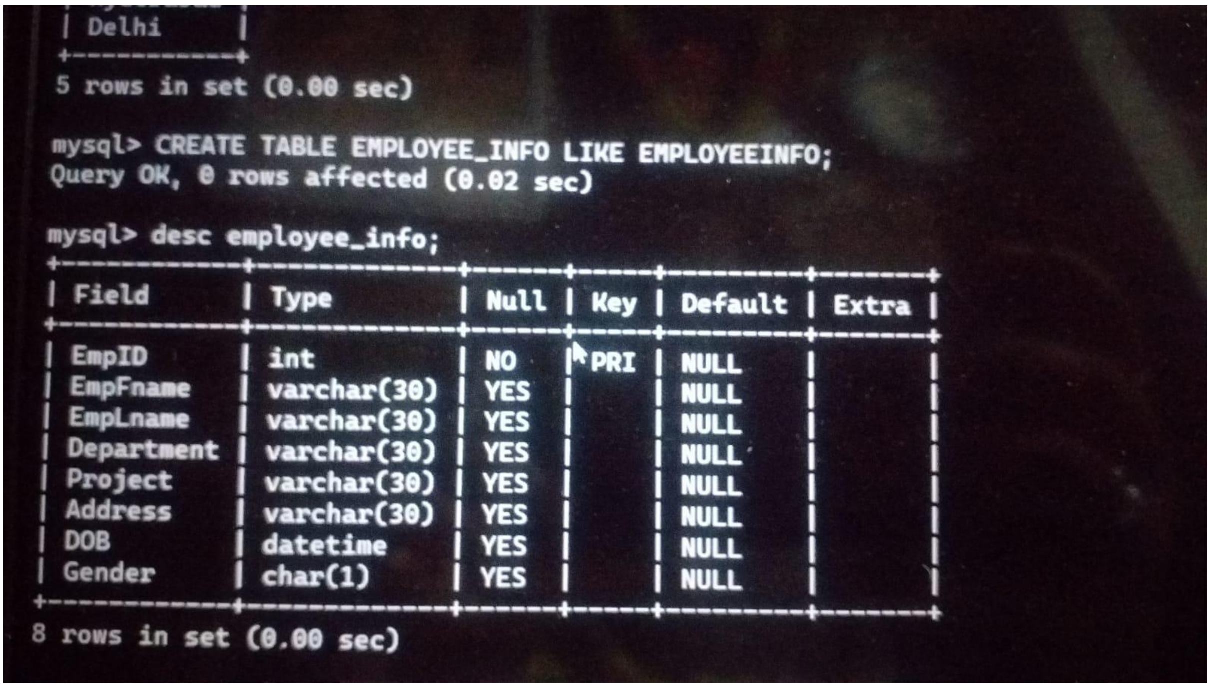
5. Write a query to fetch only the place name(string before brackets) from the Address column of EmployeeInfo table.



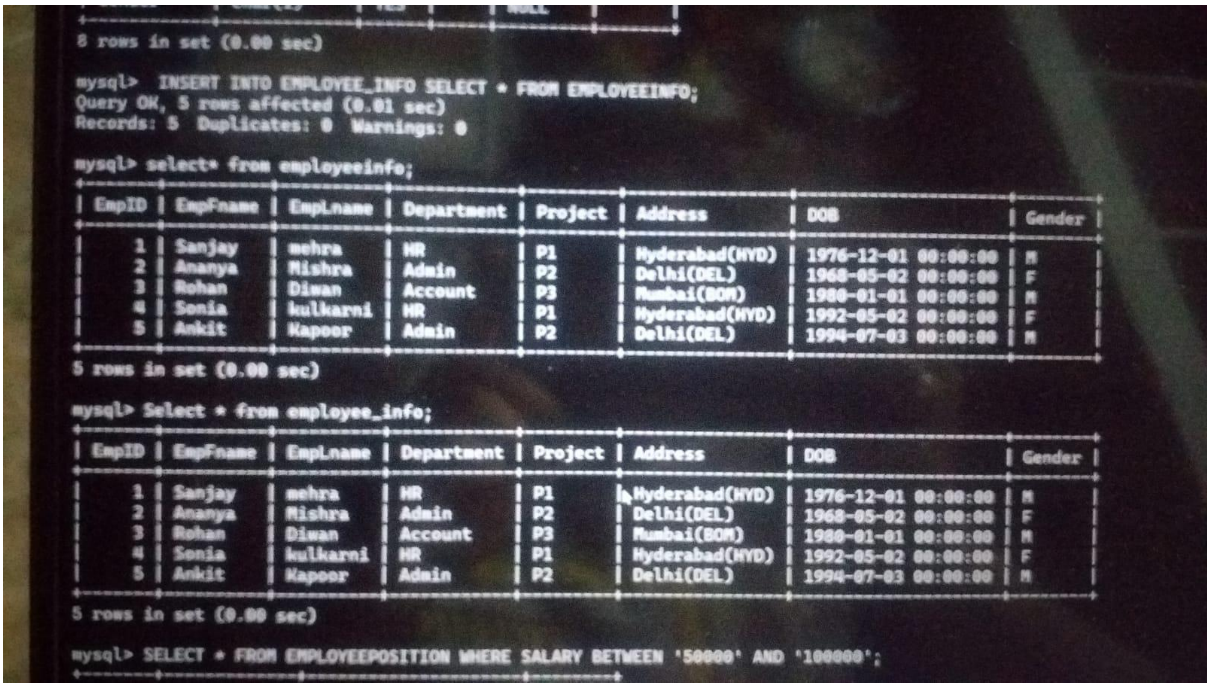
```
+---------------------------+
| mehr                      |
| Mish                      |
| Diwa                      |
| kulk                      |
| Kapo                      |
+---------------------------+
5 rows in set (0.00 sec)

mysql> SELECT LEFT(Address, LOCATE('(', Address) - 1) AS PlaceName FROM EmployeeInfo;
+-----------+
| PlaceName |
+-----------+
| Hyderabad |
| Delhi     |
| Mumbai    |
| Hyderabad |
| Delhi     |
+-----------+
5 rows in set (0.00 sec)

mysql> CREATE TABLE EMPLOYEE_INFO LIKE EMPLOYEEINFO;
Query OK, 0 rows affected (0.02 sec)
```

Query:

6. Write a query to create a new table that consists of data and structure copied from the other table.



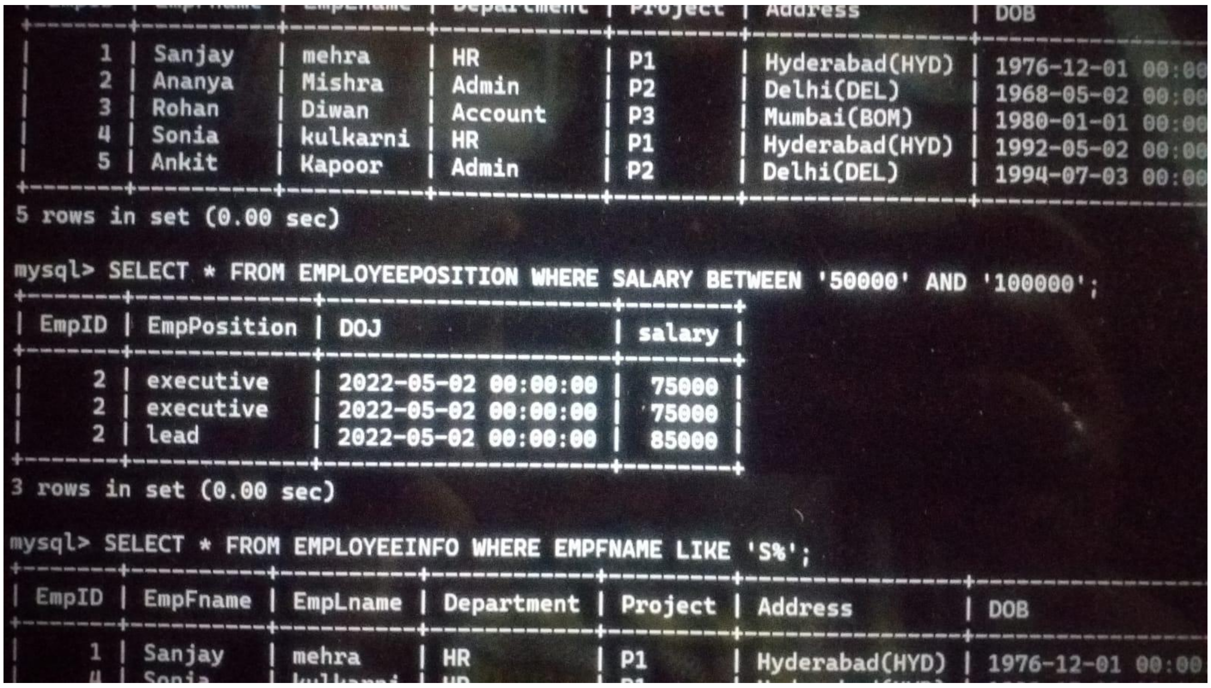
```
| Delhi        |
+--------------+
5 rows in set (0.00 sec)

mysql> CREATE TABLE EMPLOYEE_INFO LIKE EMPLOYEEINFO;
Query OK, 0 rows affected (0.02 sec)

mysql> desc employee_info;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| EmpID      | int         | NO   | PRI | NULL    |       |
| EmpFname   | varchar(30) | YES  |     | NULL    |       |
| EmpLname   | varchar(30) | YES  |     | NULL    |       |
| Department | varchar(30) | YES  |     | NULL    |       |
| Project    | varchar(30) | YES  |     | NULL    |       |
| Address    | varchar(30) | YES  |     | NULL    |       |
| DOB        | datetime    | YES  |     | NULL    |       |
| Gender     | char(1)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```



```
+------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

mysql>  INSERT INTO EMPLOYEE_INFO SELECT * FROM EMPLOYEEINFO;
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select* from employeeinfo;
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
| EmpID | EmpFname | EmpLname | Department | Project | Address        | DOB                 | Gender |
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
|     1 | Sanjay   | mehra    | HR         | P1      | Hyderabad(HYD) | 1976-12-01 00:00:00 | M      |
|     2 | Ananya   | Mishra   | Admin      | P2      | Delhi(DEL)     | 1968-05-02 00:00:00 | F      |
|     3 | Rohan    | Diwan    | Account    | P3      | Mumbai(BOM)    | 1980-01-01 00:00:00 | M      |
|     4 | Sonia    | kulkarni | HR         | P1      | Hyderabad(HYD) | 1992-05-02 00:00:00 | F      |
|     5 | Ankit    | Kapoor   | Admin      | P2      | Delhi(DEL)     | 1994-07-03 00:00:00 | M      |
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
5 rows in set (0.00 sec)

mysql> Select * from employee_info;
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
| EmpID | EmpFname | EmpLname | Department | Project | Address        | DOB                 | Gender |
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
|     1 | Sanjay   | mehra    | HR         | P1      | Hyderabad(HYD) | 1976-12-01 00:00:00 | M      |
|     2 | Ananya   | Mishra   | Admin      | P2      | Delhi(DEL)     | 1968-05-02 00:00:00 | F      |
|     3 | Rohan    | Diwan    | Account    | P3      | Mumbai(BOM)    | 1980-01-01 00:00:00 | M      |
|     4 | Sonia    | kulkarni | HR         | P1      | Hyderabad(HYD) | 1992-05-02 00:00:00 | F      |
|     5 | Ankit    | Kapoor   | Admin      | P2      | Delhi(DEL)     | 1994-07-03 00:00:00 | M      |
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEEPOSITION WHERE SALARY BETWEEN '50000' AND '100000';
+-------+--------------------+-------------+--------+
```

Query:

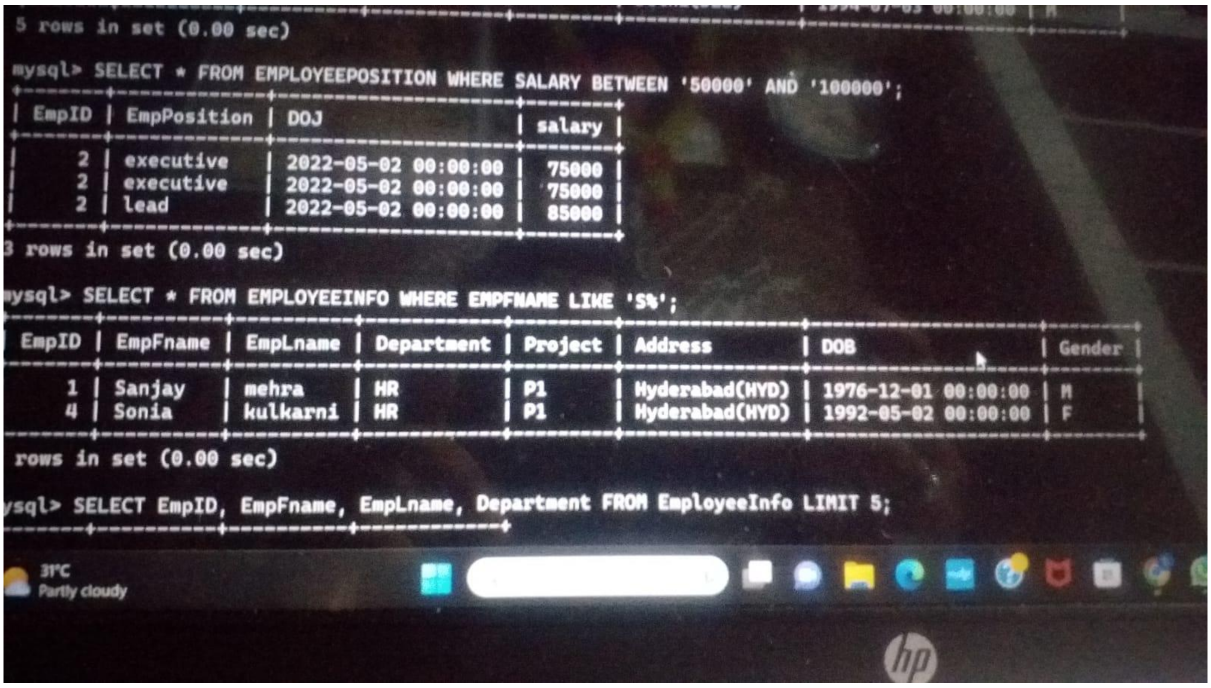
7. Write query to find all the employees whose salary is between 50000 to 100000.



```
|     1 | Sanjay   | mehra    | HR         | P1      | Hyderabad(HYD) | 1976-12-01 00:00
|     2 | Ananya   | Mishra   | Admin      | P2      | Delhi(DEL)     | 1968-05-02 00:00
|     3 | Rohan    | Diwan    | Account    | P3      | Mumbai(BOM)    | 1980-01-01 00:00
|     4 | Sonia    | kulkarni | HR         | P1      | Hyderabad(HYD) | 1992-05-02 00:00
|     5 | Ankit    | Kapoor   | Admin      | P2      | Delhi(DEL)     | 1994-07-03 00:00
+-------+----------+----------+------------+---------+----------------+-----------------
5 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEEPOSITION WHERE SALARY BETWEEN '50000' AND '100000';
+-------+-------------+---------------------+--------+
| EmpID | EmpPosition | DOJ                 | salary |
+-------+-------------+---------------------+--------+
|     2 | executive   | 2022-05-02 00:00:00 |  75000 |
|     2 | executive   | 2022-05-02 00:00:00 |  75000 |
|     2 | lead        | 2022-05-02 00:00:00 |  85000 |
+-------+-------------+---------------------+--------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEEINFO WHERE EMPFNAME LIKE 'S%';
+-------+----------+----------+------------+---------+----------------+-----------------
| EmpID | EmpFname | EmpLname | Department | Project | Address        | DOB
+-------+----------+----------+------------+---------+----------------+-----------------
|     1 | Sanjay   | mehra    | HR         | P1      | Hyderabad(HYD) | 1976-12-01 00:00
```

Query:

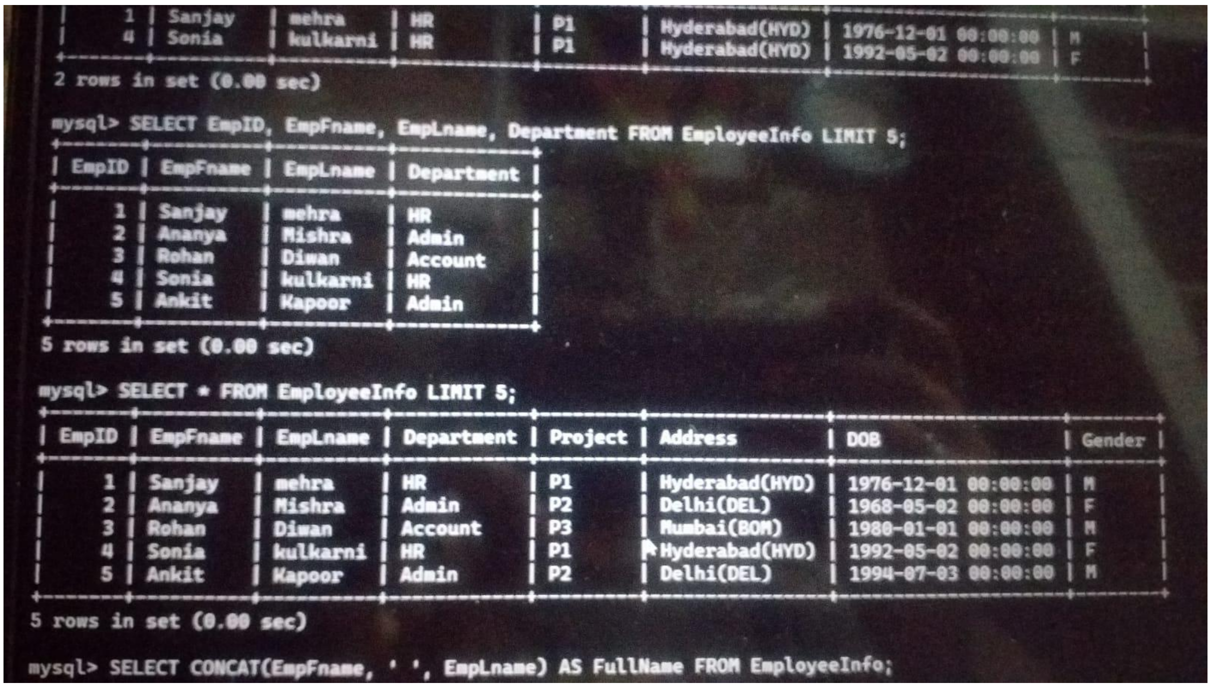
8. Write a query to find the names of employees that begin with ‘S’



```
5 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEEPOSITION WHERE SALARY BETWEEN '50000' AND '100000';
+-------+-------------+---------------------+--------+
| EmpID | EmpPosition | DOJ                 | salary |
+-------+-------------+---------------------+--------+
|     2 | executive   | 2022-05-02 00:00:00 |  75000 |
|     2 | executive   | 2022-05-02 00:00:00 |  75000 |
|     2 | lead        | 2022-05-02 00:00:00 |  85000 |
+-------+-------------+---------------------+--------+
3 rows in set (0.00 sec)

ysql> SELECT * FROM EMPLOYEEINFO WHERE EMPFNAME LIKE 'S%';
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
 EmpID | EmpFname | EmpLname | Department | Project | Address        | DOB                 | Gender |
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
     1 | Sanjay   | mehra    | HR         | P1      | Hyderabad(HYD) | 1976-12-01 00:00:00 | M      |
     4 | Sonia    | kulkarni | HR         | P1      | Hyderabad(HYD) | 1992-05-02 00:00:00 | F      |
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
 rows in set (0.00 sec)

ysql> SELECT EmpID, EmpFname, EmpLname, Department FROM EmployeeInfo LIMIT 5;
+-------+----------+----------+------------+
```

Query:

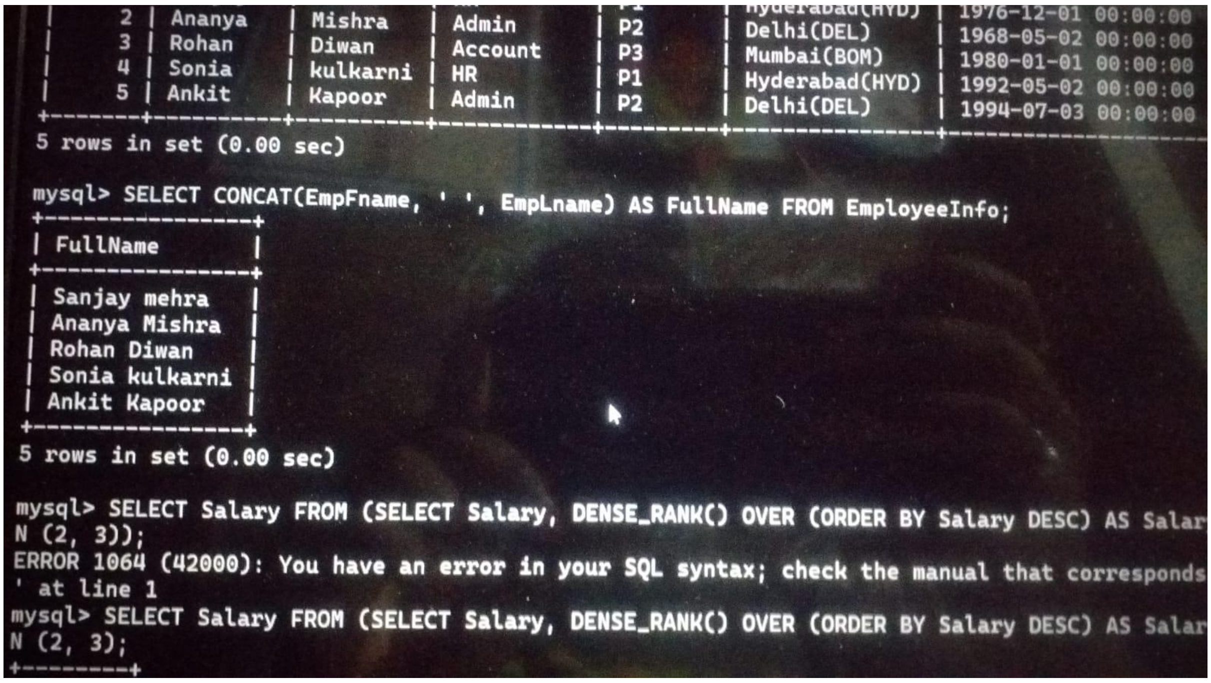
9. Write a query to fetch top N records.



```
|     1 | Sanjay  | mehra    | HR   | P1  | Hyderabad(HYD) | 1976-12-01 00:00:00 | M |
|     4 | Sonia   | kulkarni | HR   | P1  | Hyderabad(HYD) | 1992-05-02 00:00:00 | F |
+-------+---------+----------+------+-----+----------------+---------------------+---+
2 rows in set (0.00 sec)

mysql> SELECT EmpID, EmpFname, EmpLname, Department FROM EmployeeInfo LIMIT 5;
+-------+----------+----------+------------+
| EmpID | EmpFname | EmpLname | Department |
+-------+----------+----------+------------+
|     1 | Sanjay   | mehra    | HR         |
|     2 | Ananya   | Mishra   | Admin      |
|     3 | Rohan    | Diwan    | Account    |
|     4 | Sonia    | kulkarni | HR         |
|     5 | Ankit    | Kapoor   | Admin      |
+-------+----------+----------+------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM EmployeeInfo LIMIT 5;
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
| EmpID | EmpFname | EmpLname | Department | Project | Address        | DOB                 | Gender |
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
|     1 | Sanjay   | mehra    | HR         | P1      | Hyderabad(HYD) | 1976-12-01 00:00:00 | M      |
|     2 | Ananya   | Mishra   | Admin      | P2      | Delhi(DEL)     | 1968-05-02 00:00:00 | F      |
|     3 | Rohan    | Diwan    | Account    | P3      | Mumbai(BOM)    | 1980-01-01 00:00:00 | M      |
|     4 | Sonia    | kulkarni | HR         | P1      | Hyderabad(HYD) | 1992-05-02 00:00:00 | F      |
|     5 | Ankit    | Kapoor   | Admin      | P2      | Delhi(DEL)     | 1994-07-03 00:00:00 | M      |
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
5 rows in set (0.00 sec)

mysql> SELECT CONCAT(EmpFname, ' ', EmpLname) AS FullName FROM EmployeeInfo;
```

Query:

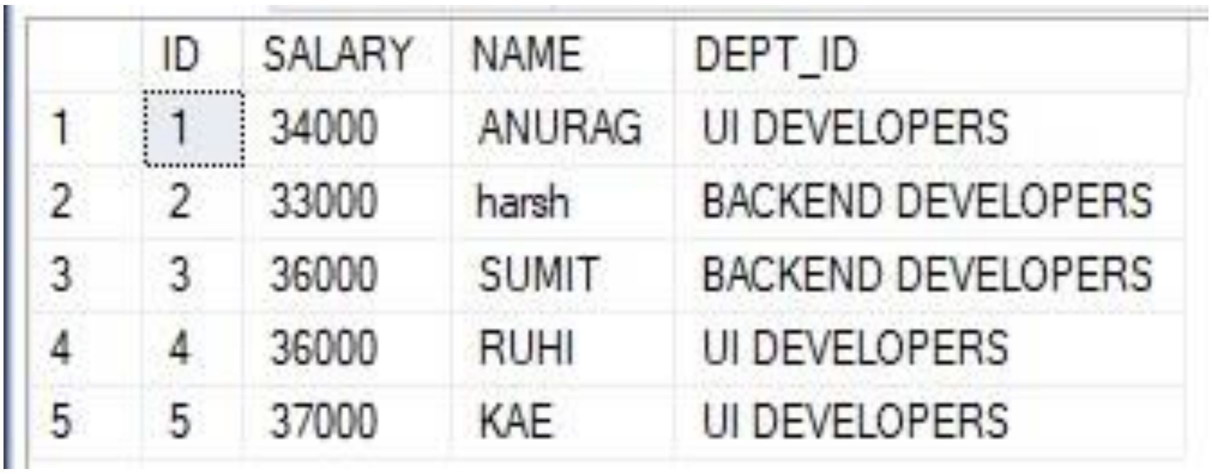
10. Write a query to retrieve the EmpFname and EmpLname in a single column as "FullName". The first name and the last name must be separated with space.



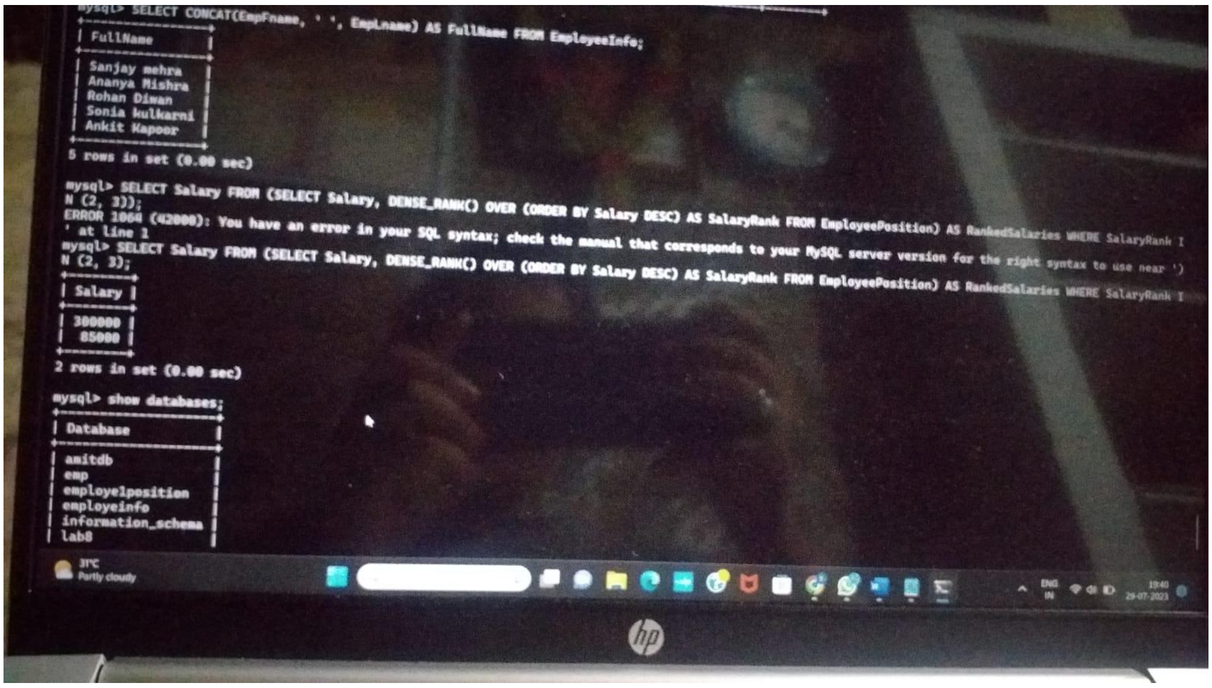
```
|     2 | Ananya   | Mishra   | Admin      | P2      | Delhi(DEL)     | 1968-05-02 00:00:00
|     3 | Rohan    | Diwan    | Account    | P3      | Mumbai(BOM)    | 1980-01-01 00:00:00
|     4 | Sonia    | kulkarni | HR         | P1      | Hyderabad(HYD) | 1992-05-02 00:00:00
|     5 | Ankit    | Kapoor   | Admin      | P2      | Delhi(DEL)     | 1994-07-03 00:00:00
+-------+----------+----------+------------+---------+----------------+---------------------
5 rows in set (0.00 sec)

mysql> SELECT CONCAT(EmpFname, ' ', EmpLname) AS FullName FROM EmployeeInfo;
+----------------+
| FullName       |
+----------------+
| Sanjay mehra   |
| Ananya Mishra  |
| Rohan Diwan    |
| Sonia kulkarni |
| Ankit Kapoor   |
+----------------+
5 rows in set (0.00 sec)

mysql> SELECT Salary FROM (SELECT Salary, DENSE_RANK() OVER (ORDER BY Salary DESC) AS Salar
N (2, 3));
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
' at line 1
mysql> SELECT Salary FROM (SELECT Salary, DENSE_RANK() OVER (ORDER BY Salary DESC) AS Salar
N (2, 3);
+--------+
```

11. To Find second and Third Highest salary in a table?

Example:



	ID	SALARY	NAME	DEPT_ID
1	1	34000	ANURAG	UI DEVELOPERS
2	2	33000	harsh	BACKEND DEVELOPERS
3	3	36000	SUMIT	BACKEND DEVELOPERS
4	4	36000	RUHI	UI DEVELOPERS
5	5	37000	KAE	UI DEVELOPERS



```
mysql> SELECT CONCAT(EmpFname, ' ', EmpLname) AS FullName FROM EmployeeInfo;
+-----------------+
| FullName        |
+-----------------+
| Sanjay mehra    |
| Ananya Mishra   |
| Rohan Diwan     |
| Sonia kulkarni  |
| Ankit Kapoor    |
+-----------------+
5 rows in set (0.00 sec)

mysql> SELECT Salary FROM (SELECT Salary, DENSE_RANK() OVER (ORDER BY Salary DESC) AS SalaryRank FROM EmployeePosition) AS RankedSalaries WHERE SalaryRank I
N (2, 3));
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 ')
' at line 1
mysql> SELECT Salary FROM (SELECT Salary, DENSE_RANK() OVER (ORDER BY Salary DESC) AS SalaryRank FROM EmployeePosition) AS RankedSalaries WHERE SalaryRank I
N (2, 3);
+--------+
| Salary |
+--------+
| 300000 |
|  85000 |
+--------+
2 rows in set (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| amitdb             |
| emp                |
| employe1position   |
| employeinfo        |
| information_schema |
| lab8               |
```

12. Explain with example

Unique Key

In SQL, a Unique Key is a type of database constraint that ensures the values in a specific column or a set of columns in a table are unique and not duplicated across rows. It guarantees that each value in the specified column(s) is unique and different from all other values in that column.

Key characteristics of Unique Key:

Uniqueness: Each value in the specified column(s) must be unique, and no two rows can have the same value in that column(s).

Null values: Unlike the Primary Key, a Unique Key can allow NULL values. However, if a column is part of a Unique Key constraint and has a NULL value, it can have only one NULL value because NULL is considered unique.

Table-level or Column-level: A Unique Key can be applied to a single column or a combination of columns, which means it can be defined at the column level or at the table level.

Multiple Unique Keys per table: A table can have multiple Unique Key constraints, either on different columns or combinations of columns.

EXAMPLE;

> Consider a table named "Employees" with the following columns:
>
> EmployeeID, Email, and PhoneNumber. If we want to make sure that each employee has a unique email address and phone number, we can define the Email and PhoneNumber columns as unique keys. This means that no two employees can have thesame email or phone number, but they can have the same EmployeeID

Primary Key

In SQL, a Primary Key is a special type of database constraint that uniquely identifies each record (row) in a table. It ensures that the values in the specified column or a set of columns are unique and not duplicated within the table. The Primary Key serves as a unique identifier for each row and is used to reference and relate records from other tables.

Key characteristics of Primary Key:

Uniqueness: Each value in the specified column(s) must be unique, and no two rows can have the same value in that column(s).

Non-null: A Primary Key cannot contain NULL values. Each row in the table must have a valid and unique value for the Primary Key column(s).

Table-level: A table can have only one Primary Key, and it is defined at the table level.

Indexed: The Primary Key is automatically indexed by the database management system to enhance data retrieval performance.

Foreign Key reference: The Primary Key is often used as a reference in other tables as a Foreign Key to establish relationships between tables.

Example:

Let's consider a table named "students" that stores information about students in a school. We want to ensure that each student has a unique student ID in the table. To achieve this, we can define a Primary Key constraint on the "student_id" column:

```
CREATE TABLE students (
student_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    age INT,
    grade VARCHAR(2)
);
```

In this example, the "students" table has a Primary Key constraint on the "student_id" column, specified by using the PRIMARY KEY keyword after the column definition. This constraint ensures that each student has a unique ID, and the "student_id" column cannot contain duplicate values.

Foreign Key

In SQL, a Foreign Key is a database constraint used to establish a relationship between two tables. It ensures the referential integrity of the data by linking data in one table to the data in another table. The Foreign Key in one table refers to the Primary Key in another table, creating a parent-child relationship between the two tables.

Key characteristics of Foreign Key:

References Primary Key: The Foreign Key column in one table refers to the Primary Key column of another table. This establishes a link between the two tables, where the Foreign Key column holds values that match the values in the Primary Key column of the related table.

Maintains Referential Integrity: The Foreign Key constraint ensures that the values in the Foreign Key column must match existing values in the related Primary Key column of the referenced table. It helps maintain data consistency and prevents invalid data relationships.

Multiple Foreign Keys: A table can have multiple Foreign Keys that reference different tables. Each Foreign Key establishes a separate relationship to its respective referenced table.

Can Allow NULL: By default, a Foreign Key can contain NULL values, which means it is not mandatory to have a related entry in the referenced table. However, you can also specify that the Foreign Key must be NOT NULL to enforce a mandatory relationship.