E.V.K.DEEPTHI(AFO311771)

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

EmployeePosition Table:

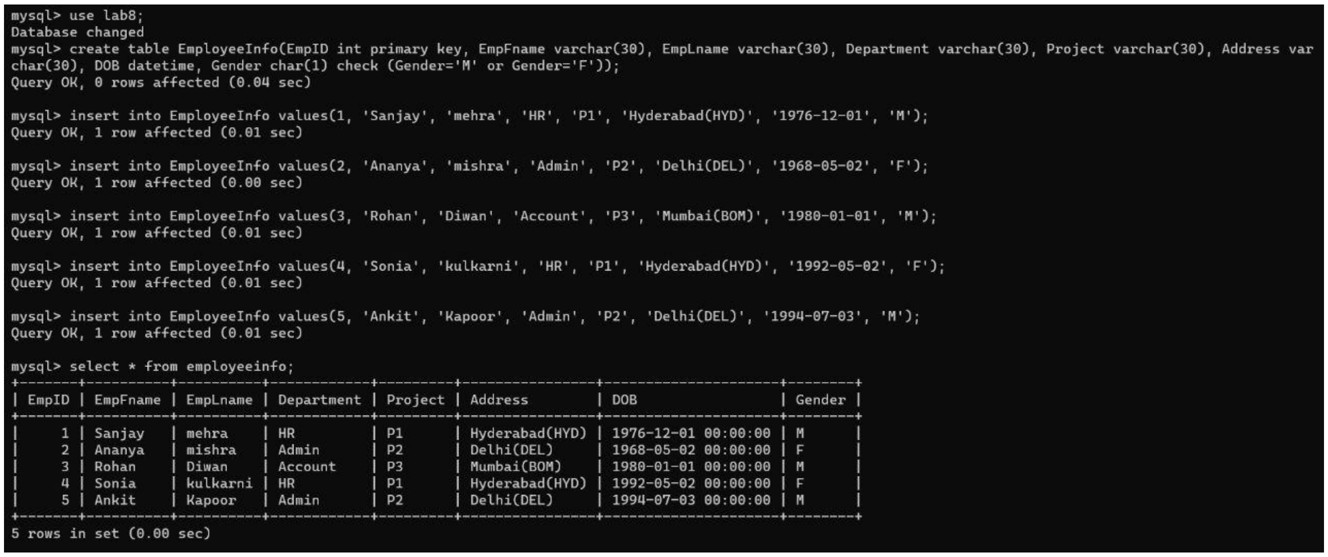
EmpID	EmpPosition	DateOfJoining	Salary
1	Manager	01/05/2022	500000
2	Executive	02/05/2022	75000
3	Manager	01/05/2022	90000
2	Lead	02/05/2022	85000
1	Executive	01/05/2022	300000

- create table 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');

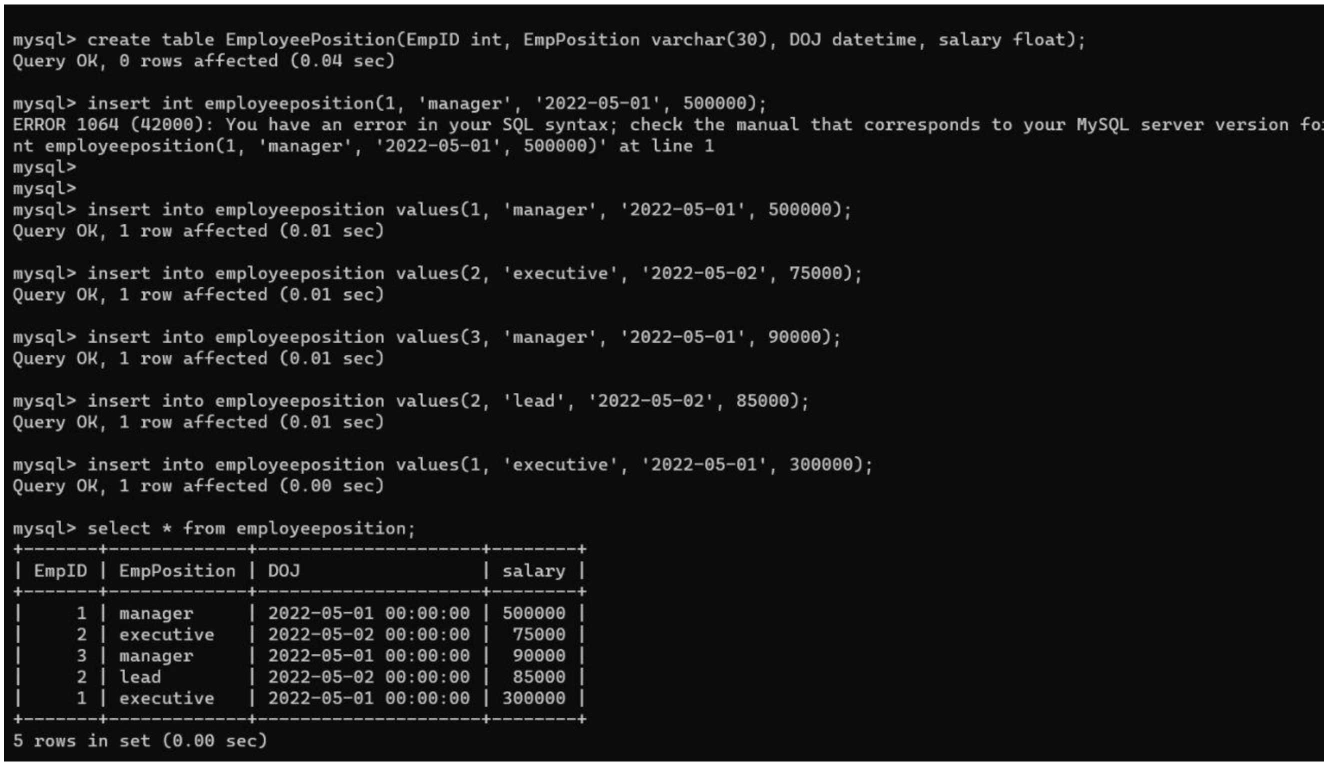
  - insert into EmployeeInfo values(1, 'Sanjay', 'mehra', 'HR', 'P1', 'Hyderabad(HYD)', '1976-12-01', 'M');
  - insert into EmployeeInfo values(2, 'Ananya', 'Mishra', 'Admin', 'P2', 'Delhi(DEL)', '1968-05-02', 'F');
  - insert into EmployeeInfo values(3, 'Rohan', 'Diwan', 'Account', 'P3', 'Mumbai(BOM)', '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');

- 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);



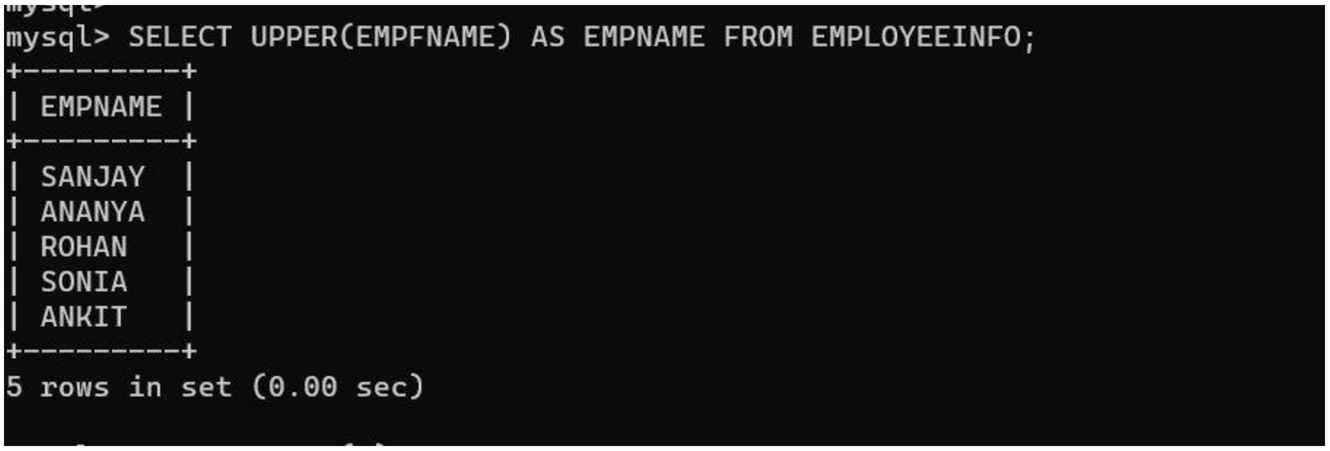
```
mysql> use lab8;
Database changed
mysql> create table EmployeeInfo(EmpID int primary key, EmpFname varchar(30), EmpLname varchar(30), Department varchar(30), Project varchar(30), Address var
char(30), DOB datetime, Gender char(1) check (Gender='M' or Gender='F'));
Query OK, 0 rows affected (0.04 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> insert into EmployeeInfo values(3, 'Rohan', 'Diwan', 'Account', 'P3', 'Mumbai(BOM)', '1980-01-01', 'M');
Query OK, 1 row affected (0.01 sec)

mysql> insert into EmployeeInfo values(4, 'Sonia', 'kulkarni', 'HR', 'P1', 'Hyderabad(HYD)', '1992-05-02', 'F');
Query OK, 1 row affected (0.01 sec)

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 (0.04 sec)

mysql> insert int employeeposition(1, 'manager', '2022-05-01', 500000);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version fo
nt employeeposition(1, 'manager', '2022-05-01', 500000)' at line 1
mysql>
mysql>
mysql> insert into employeeposition values(1, 'manager', '2022-05-01', 500000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employeeposition values(2, 'executive', '2022-05-02', 75000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employeeposition values(3, 'manager', '2022-05-01', 90000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employeeposition values(2, 'lead', '2022-05-02', 85000);
Query OK, 1 row affected (0.01 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 |
|     3 | manager     | 2022-05-01 00:00:00 |  90000 |
|     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)
```

Q1. Write a query to fetch the EmpFname from the EmployeeInfo table in theupper case and use the ALIAS name as EmpName.

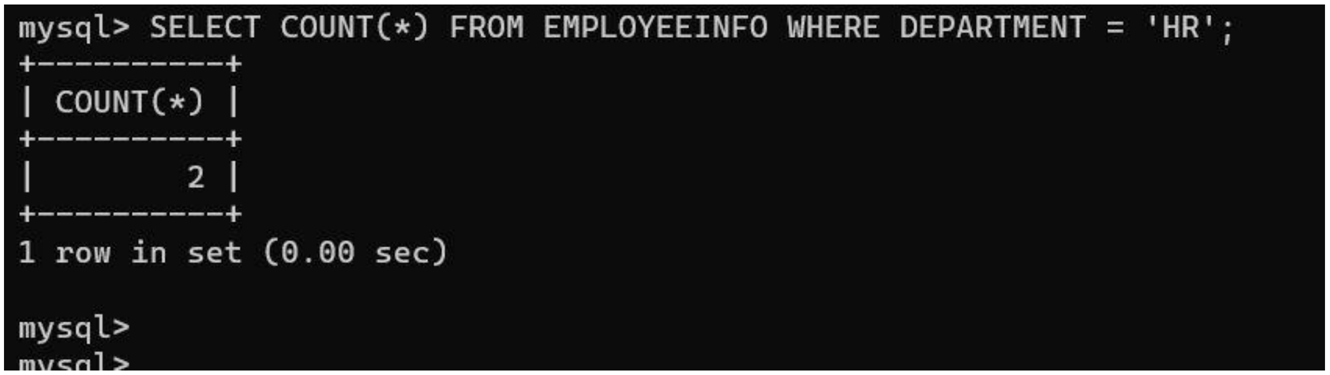
Ans: SELECT UPPER(EMPFNAME) AS EMPNAME FROM EMPLOYEEINFO;



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

Q2. Write a query to fetch the number of employees working in the department ‘HR’.

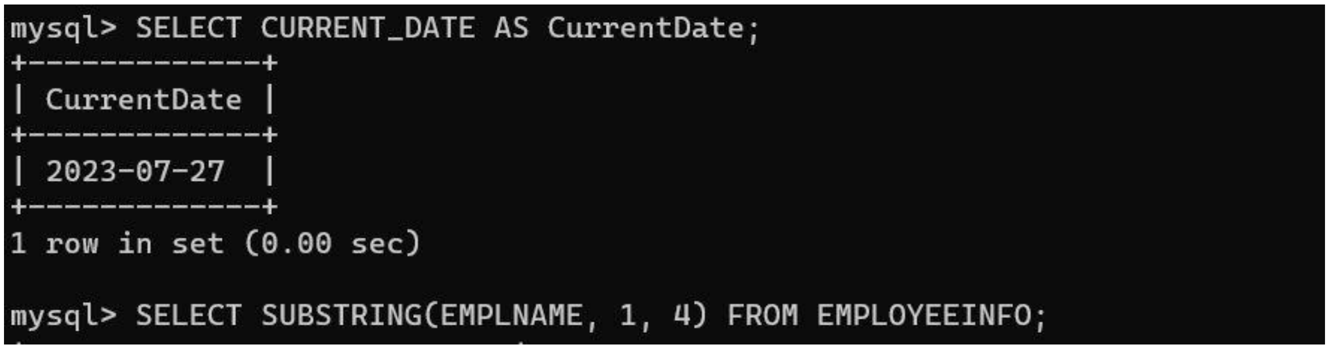
Ans: SELECT COUNT(*) FROM EMPLOYEEINFO WHERE DEPARTMENT = 'HR';



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

mysql>
mysql>
```

Q3. Write a query to get the current date.

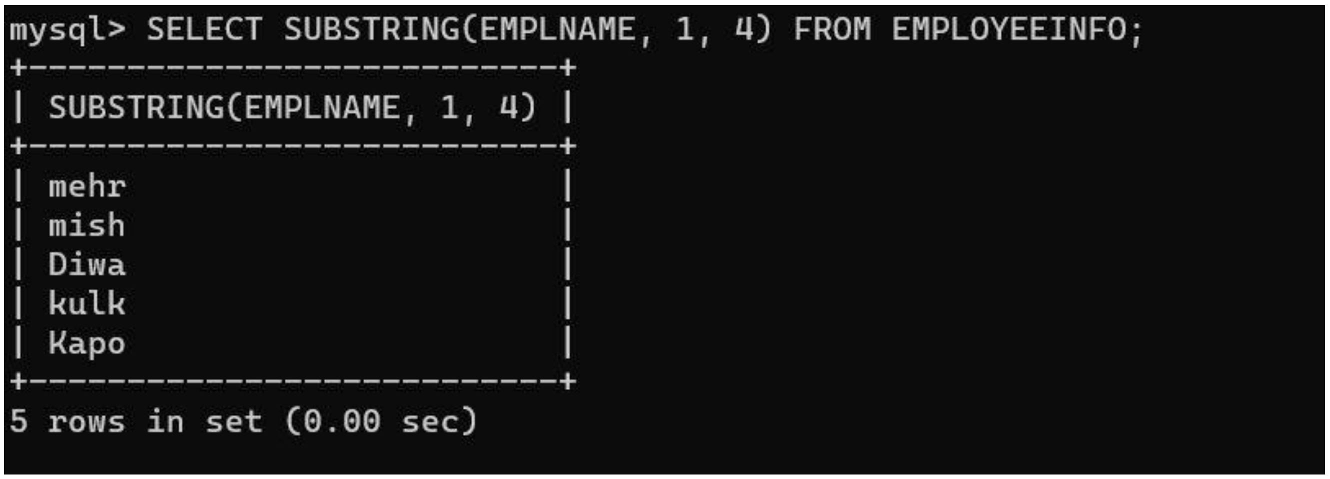
Ans: SELECT CURRENT_DATE AS CurrentDate;



```
mysql> SELECT CURRENT_DATE AS CurrentDate;
+-------------+
| CurrentDate |
+-------------+
| 2023-07-27  |
+-------------+
1 row in set (0.00 sec)

mysql> SELECT SUBSTRING(EMPLNAME, 1, 4) FROM EMPLOYEEINFO;
```

Q4. Write a query to retrieve the first four characters of EmpLname from the EmployeeInfo table.

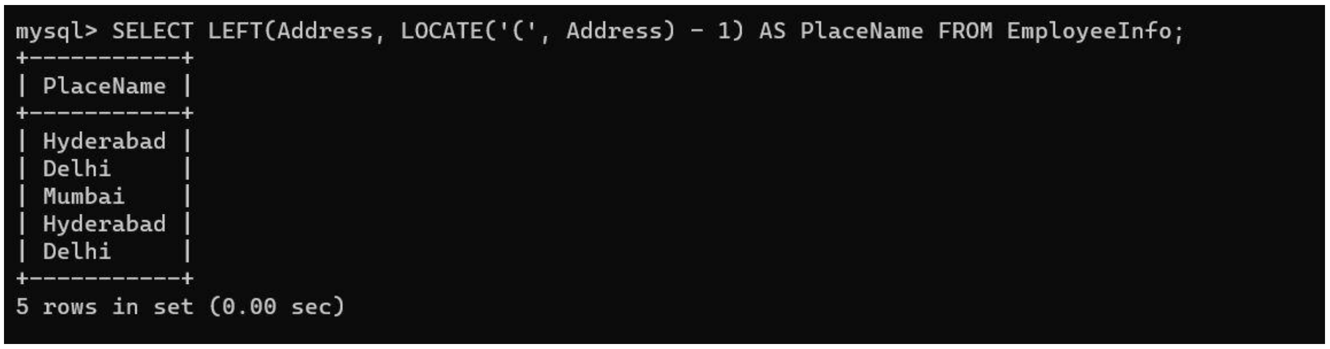
Ans: SELECT SUBSTRING(EMPLNAME, 1, 4) FROM EMPLOYEEINFO;



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

Q5. Write a query to fetch only the place name(string before brackets) from the Address column of EmployeeInfo table.

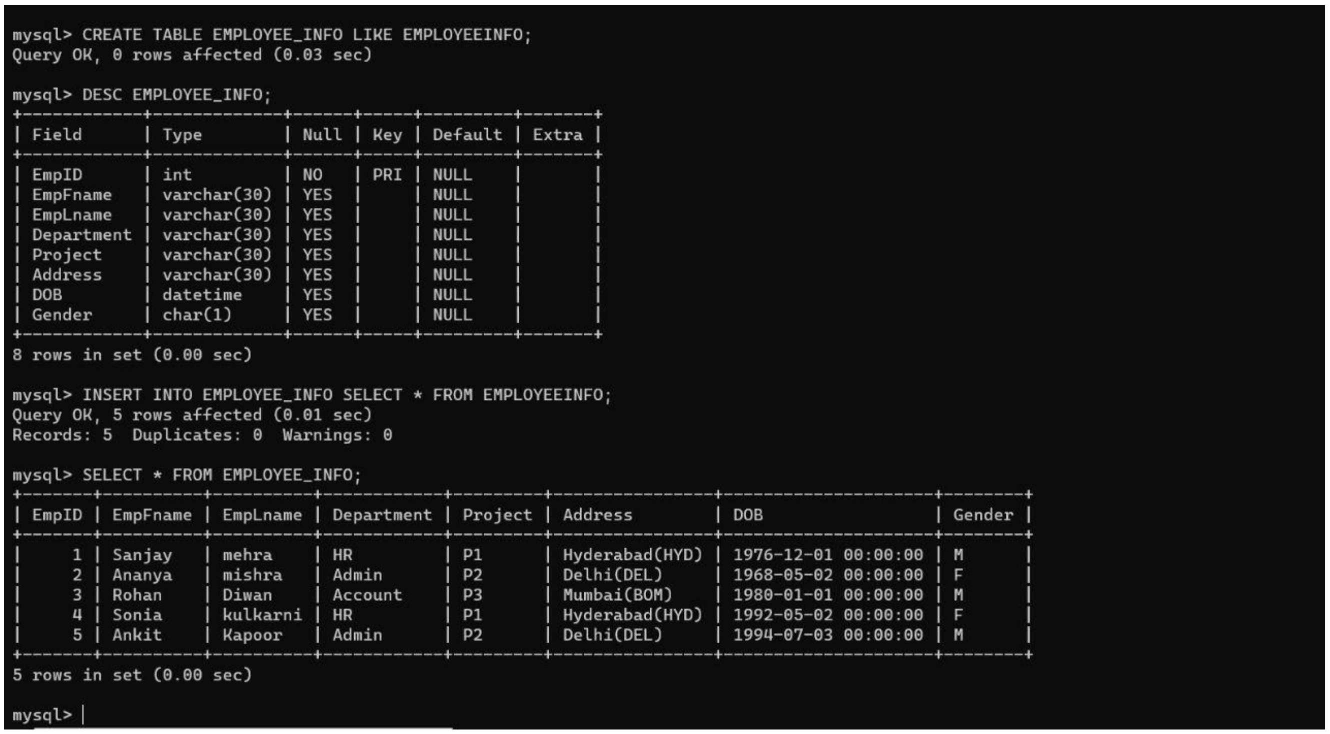
Ans: SELECT LEFT(Address, LOCATE('(', Address) - 1) AS PlaceName FROM EmployeeInfo;



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

Q6. Write a query to create a new table which consists of data and structure copied from the other table.

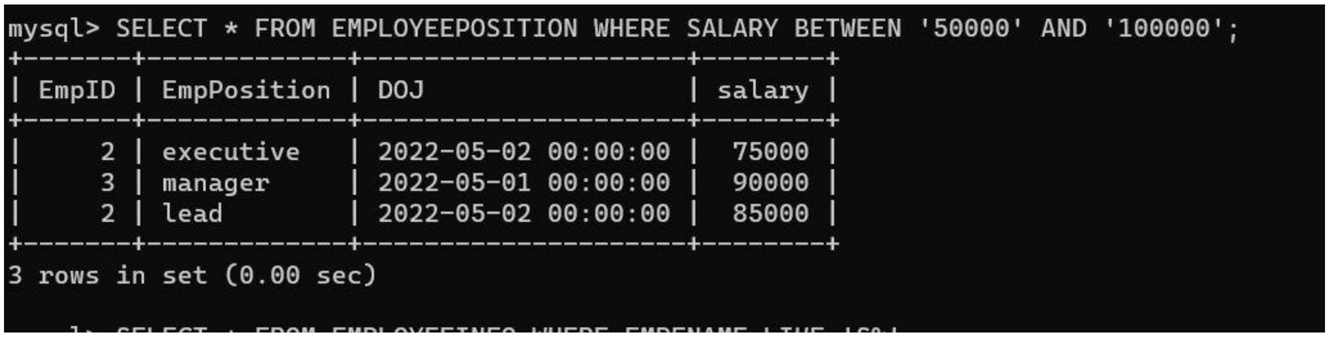
- ❖ CREATE TABLE EMPLOYEE_INFO LIKE EMPLOYEEINFO;
- ❖ INSERT INTO EMPLOYEE_INFO SELECT * FROM EMPLOYEEINFO;



```
mysql> CREATE TABLE EMPLOYEE_INFO LIKE EMPLOYEEINFO;
Query OK, 0 rows affected (0.03 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)

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 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> |
```

Q7. Write q query to find all the employees whose salary is between 50000 to 100000.

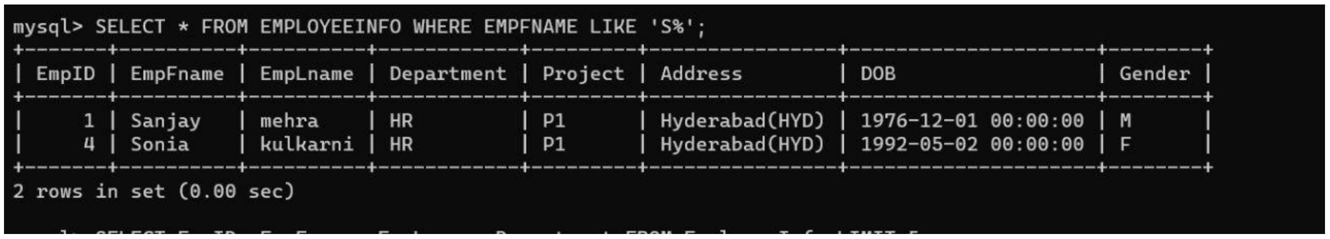
Ans: SELECT * FROM EMPLOYEEPOSITION WHERE SALARY BETWEEN '50000' AND '100000';



```
mysql> SELECT * FROM EMPLOYEEPOSITION WHERE SALARY BETWEEN '50000' AND '100000';
+-------+-------------+---------------------+--------+
| EmpID | EmpPosition | DOJ                 | salary |
+-------+-------------+---------------------+--------+
|     2 | executive   | 2022-05-02 00:00:00 |  75000 |
|     3 | manager     | 2022-05-01 00:00:00 |  90000 |
|     2 | lead        | 2022-05-02 00:00:00 |  85000 |
+-------+-------------+---------------------+--------+
3 rows in set (0.00 sec)
```

Q8. Write a query to find the names of employees that begin with ‘S’

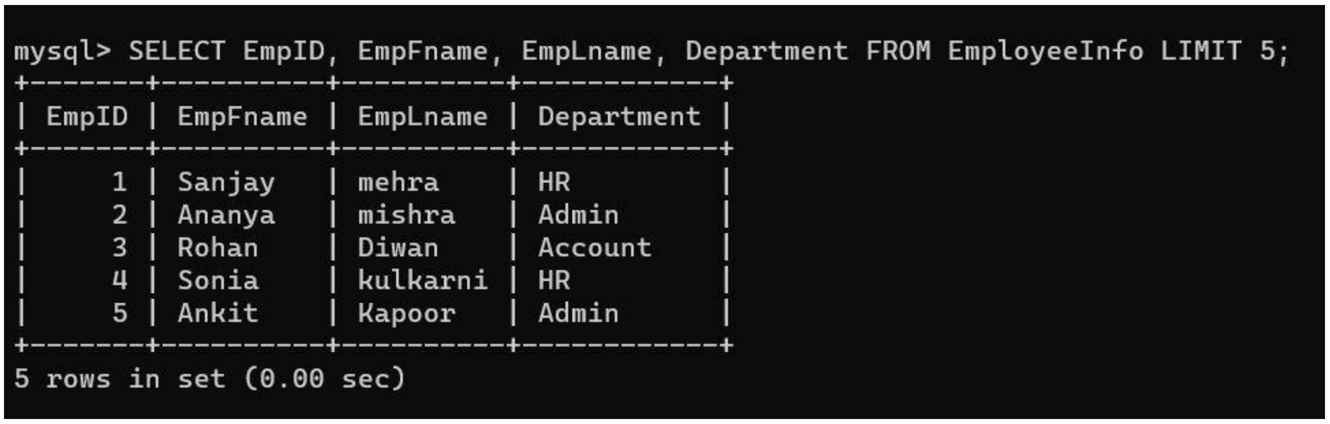
Ans: SELECT * FROM EMPLOYEEINFO WHERE EMPFNAME LIKE 'S%';



```
mysql> 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      |
+-------+----------+----------+------------+---------+----------------+---------------------+--------+
2 rows in set (0.00 sec)
```

Q9. Write a query to fetch top N records.

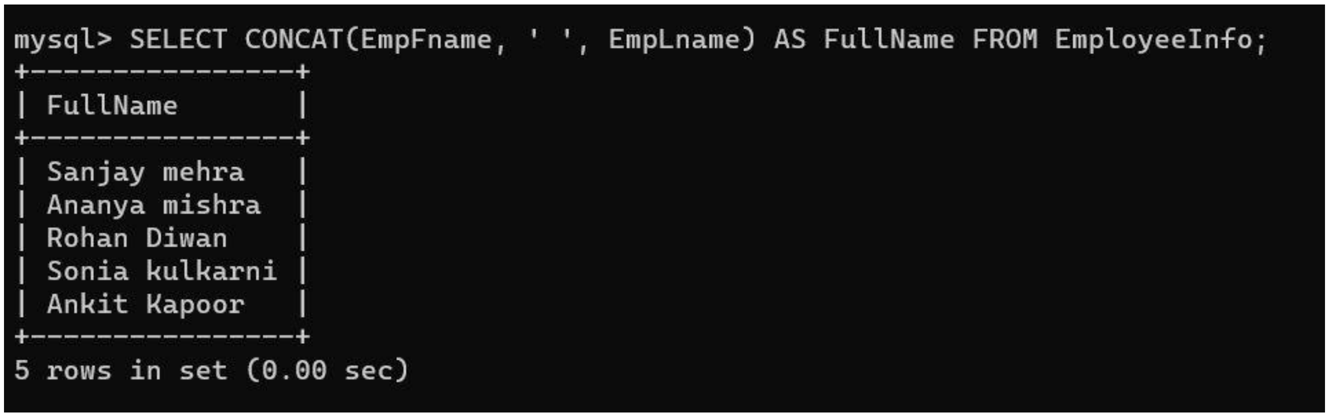
Ans: SELECT EmpID, EmpFname, EmpLname, Department FROM EmployeeInfo LIMIT 5;



```
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)
```

Q10. 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.

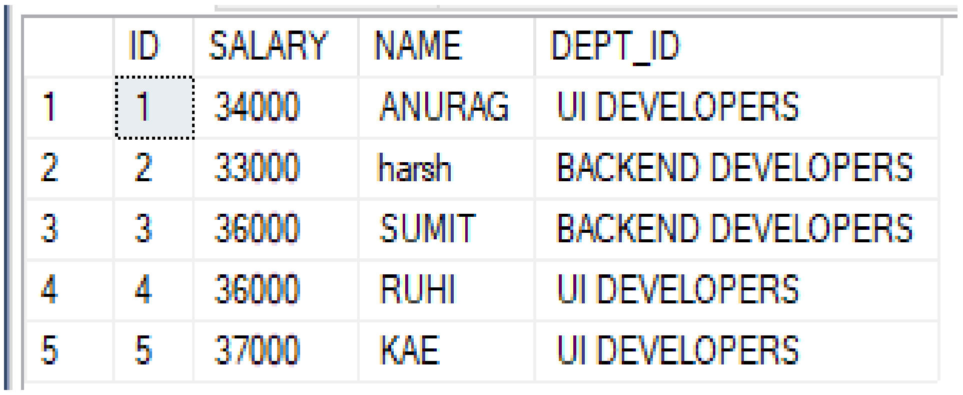
Ans: SELECT CONCAT(EmpFname, ' ', EmpLname) AS FullName FROM EmployeeInfo;



```
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)
```

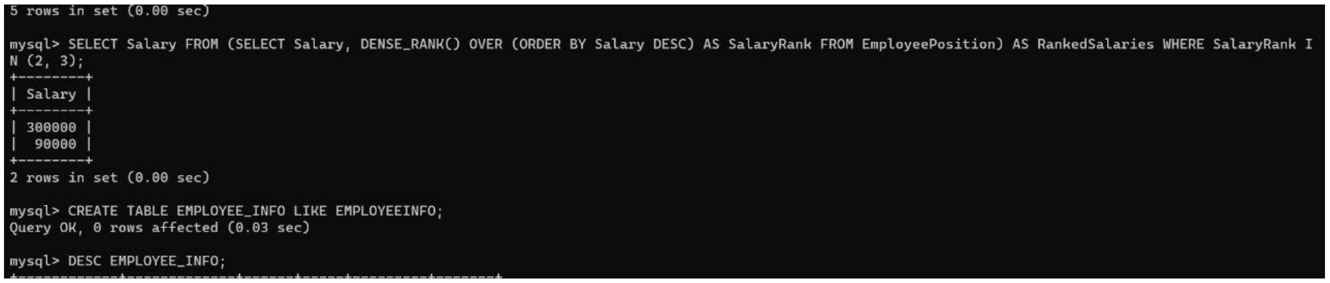
Q11.   To Find second and Third Highest salary in a table?

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

Ans: SELECT SALARY FROM (SELECT SALARY, DENSE_RANK() OVER (ORDER BY SALARY DESC) AS SALARYRANK FROM EMPLOYEEPOSTION) AS RANKSALARIES WHERE SALARYRANK IN (2, 3);



```
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);
+--------+
| Salary |
+--------+
| 300000 |
|  90000 |
+--------+
2 rows in set (0.00 sec)

mysql> CREATE TABLE EMPLOYEE_INFO LIKE EMPLOYEEINFO;
Query OK, 0 rows affected (0.03 sec)

mysql> DESC EMPLOYEE_INFO;
```

Q12.   Explain with example Unique Key , Primary Key, Foreign Key?

Ans: In relational databases, unique key, primary key, and foreign key are important concepts that help ensure data integrity and define relationships between tables. Let's explain each of them with examples:

1. Unique Key: A unique key is a column or a combination of columns in a table that ensures each row's values are unique within that table. It helps to prevent duplicate entries in the specified column(s). Unlike a primary key, a table can have multiple unique keys.

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 the same email or phone number, but they can have the same EmployeeID.

2.Primary Key: A primary key is a special kind of unique key that uniquely identifies each record in a table. It cannot contain NULL values, and there can be only one primary key per table. The primary key is used to establish relationships between tables, as it is often referenced by foreign keys in other tables.

Example: Let's continue with the "Employees" table example. Here, the EmployeeID column could be set as the primary key. It will ensure that each employee has a unique ID, and no NULL values are allowed in this column. The primary key will act as a unique identifier for each employee.

3. Foreign Key: A foreign key is a column or a combination of columns in one table that refers to the primary key in another table. It establishes a relationship between two tables, allowing you to enforce referential integrity, which means that the values in the foreign key must correspond to values in the primary key of the referenced table.

Example: Now, consider a new table called "Tasks," which includes columns TaskID, TaskName, and AssignedTo. In the "AssignedTo" column, we want to specify which employee is assigned to each task. To establish this relationship, we can set the "AssignedTo" column as a foreign key, referencing the "EmployeeID" column from the "Employees" table. This ensures that the value in the "AssignedTo" column must exist in the "EmployeeID" column of the "Employees" table, preventing the assignment of tasks to non-existing employees.

Unique keys ensure uniqueness within a table, primary keys uniquely identify each record in a table, and foreign keys establish relationships between tables, referencing the primary key of another table. These concepts are fundamental in database design and play a crucial role in maintaining data consistency and integrity.