

S.N.R.Likhitha(AF0312909)

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

Query:

```
SELECT UPPER(EmpFname) AS Empname FROM EmployeeInfo;
```

```
mysql> SELECT UPPER(EmpFname) AS Empname FROM EmployeeInfo;
ERROR 1054 (42S22): Unknown column 'EmpFna' in 'field list'
mysql> select * from EmployeeInfo;
+-----+-----+-----+-----+-----+-----+-----+-----+
| EmpId | EmpFname | EmpLname | Department | Project | Address | DOB | Gender |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Sanjay | Mehra | HR | P1 | Hyderabad(HYD) | 1976-12-01 | M |
| 2 | Ananya | Mishra | Admin | P2 | Delhi(DEL) | 1968-05-02 | F |
| 3 | Rohan | Diwan | Account | P3 | Mumbai(BOM) | 1980-01-01 | M |
| 4 | Sonia | Kulakarni | HR | P1 | Hyderabad(HYB) | 1992-05-02 | F |
| 5 | Ankit | Kapoor | Admin | P2 | Delhi(DEL) | 1994-07-03 | M |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT UPPER(EmpFna) AS EmpName FROM EmployeeInfo;
ERROR 1054 (42S22): Unknown column 'EmpFna' in 'field list'
mysql> SELECT UPPER(EmpFname) AS Empname FROM EmployeeInfo;
+-----+
| Empname |
+-----+
| SANJAY |
| ANANYA |
| ROHAN |
| SONIA |
| ANKIT |
+-----+
5 rows in set (0.00 sec)
```

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

Query:

```
SELECT COUNT(*) AS NumberOfEmployees FROM EmployeeInfo WHERE
Department = 'HR';
```

```
mysql> SELECT COUNT(*) AS NumberOfEmployees FROM EmployeeInfo WHERE Department = 'HR';
+-----+
| NumberOfEmployees |
+-----+
| 2 |
+-----+
1 row in set (0.01 sec)
```

3. Write a query to get the current date.

Query: SELECT CURRENT_DATE AS CurrentDate;

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

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

Query:

```
SELECT SUBSTRING(EmpLname, 1, 4) AS FirstFourCharacters FROM EmployeeInfo;
```

```
mysql> SELECT SUBSTRING(EmpLname, 1, 4) AS FirstFourCharacters FROM EmployeeInfo;
+-----+
| FirstFourCharacters |
+-----+
| Mehr                |
| Mish                |
| Diwa                |
| Kula                |
| Kapo                |
+-----+
5 rows in set (0.00 sec)
```

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

Query:

```
SELECT SUBSTRING_INDEX(Address, ',', -1) AS PlaceName
```

-> FROM EmployeeInfo;

```
mysql> SELECT SUBSTRING_INDEX(Address, ',', -1) AS PlaceName
-> FROM EmployeeInfo;
+-----+
| PlaceName |
+-----+
| Hyderabad(HYD) |
| Delhi(DE) |
| Mumbai(BOM) |
| Hyderabad(HYB) |
| Delhi(DE) |
+-----+
```

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

Query:

- create table employeeDetails as select *from EmployeeInfo;
- select *from employeeDetails;

```
mysql> create table employeeDetails as select *from EmployeeInfo;
Query OK, 5 rows affected (0.03 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select *from employeeDetails
-> select *from employeeDetails;
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 'select *from employeeDetails' at line 2
mysql>
mysql>
mysql>
mysql> select *from employeeDetails;
```

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

5 rows in set (0.00 sec)

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

Query:

SELECT * FROM EmployeePosition WHERE Salary BETWEEN 50000 AND 100000;

```
mysql> SELECT * FROM EmployeePosition WHERE Salary BETWEEN 50000 AND 100000;
```

EmpId	EmpPosition	DateOfJoining	Salary
2	Executive	2022-05-02	75000
3	Manager	2022-05-01	90000
4	Lead	2022-05-02	85000

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

Query:

SELECT EmpFname, EmpLname FROM EmployeeInfo WHERE EmpFname LIKE 'S%';

```
mysql> SELECT EmpFname, EmpLname FROM EmployeeInfo WHERE EmpFname LIKE 'S%';
+-----+-----+
| EmpFname | EmpLname |
+-----+-----+
| Sanjay   | Mehra    |
| Sonia    | Kulakarni|
+-----+-----+
2 rows in set (0.03 sec)
```

9. Write a query to fetch top N records.

Query: 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    | Kulakarni | HR          |
| 5     | Ankit    | Kapoor   | Admin       |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

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.

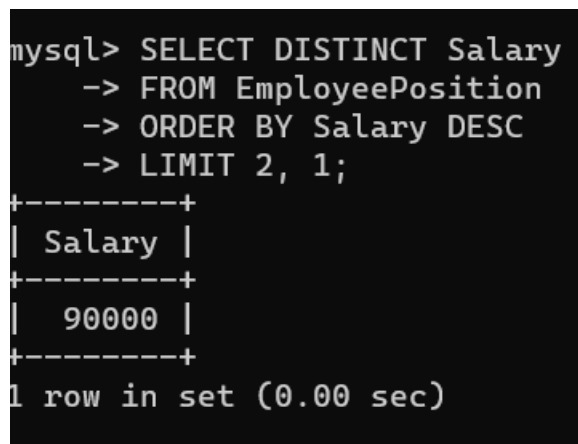
Query: select concat(EmpFname,',',EmpLname) as FullName from EmployeeInfo;

```
mysql> select concat(EmpFname,',',EmpLname) as FullName from EmployeeInfo;
+-----+
| FullName          |
+-----+
| SanjayMehra      |
| AnanyaMishra     |
| RohanDiwan       |
| SoniaKulakarni   |
| AnkitKapoor      |
+-----+
5 rows in set (0.01 sec)
```

11. To find the second and third highest salary in the EmployeePosition table.

Query:

```
SELECT DISTINCT Salary
-> FROM EmployeePosition
-> ORDER BY Salary DESC
-> LIMIT 2, 1;
```



```
mysql> SELECT DISTINCT Salary
-> FROM EmployeePosition
-> ORDER BY Salary DESC
-> LIMIT 2, 1;
+-----+
| Salary |
+-----+
| 90000  |
+-----+
1 row in set (0.00 sec)
```

12. Explain with example Unique Key, Primary Key and Foreign Key.

1. Unique Key:

A **Unique Key** is a database constraint that ensures that the values in a specific column (or set of columns) are unique and cannot be duplicated within a table. It is used to maintain data integrity and prevent duplicate entries. While a table can have multiple unique keys, each unique key must have a distinct name.

Example: Consider a table called "Students" with the following columns: StudentID, Name, and Email. To ensure that each student has a unique email address, you can define the "Email" column as a unique key. This constraint guarantees that no two students can have the same email address in the "Students" table.

2. Primary Key:

A **Primary Key** is a special type of Unique Key that uniquely identifies each record in a table. It is a column or a combination of columns that uniquely identify each row, and it must have a unique value for each record. Every table should have a **Primary Key**, and it must be unique and not allow **NULL** values.

Example: Continuing with the "Students" table, let's say the "StudentID" column is designated as the **Primary Key**. This means that each student in the table will have a unique StudentID, and no two students can have the same StudentID. The **Primary Key** is essential for efficiently identifying and accessing specific rows in the table.

3.Foreign Key:

A **Foreign Key** is a column or a set of columns in a table that refers to the **Primary Key** of another table. It establishes a link or relationship between two tables and ensures referential integrity. The **Foreign Key** constraint helps maintain consistency and prevents actions that would leave orphaned records when working with related tables.

Example: Suppose you have another table called "Courses" with columns: CourseID, CourseName, and InstructorID. The "InstructorID" column represents the instructor who teaches each course. To establish a relationship with the "Students" table, you can create a **Foreign Key** that references the "StudentID" column in the "Students" table. This **Foreign Key** ensures that only existing StudentIDs from the "Students" table can be assigned as instructors in the "Courses" table.

