

1. Find the second highest salary

Table: employees(id, name, salary)

Values:

1, Shubham, 5000

2, Debug, 6000

3, Wlth, 7000

```
SELECT DISTINCT salary
FROM employees
ORDER BY salary DESC
LIMIT 1 OFFSET 1;
```

(LIMIT ka use hum **number of rows restrict karne** ke liye karte hain jo query return karegi.

"Mujhe sirf top 3 rows chahiye."

Syntax:

```
SELECT * FROM table_name
LIMIT 3;
```

OFFSET ka use hota hai to **skip rows** from the top.

"Top 2 skip karke next 1 row chahiye."

Syntax:

```
sql
CopyEdit
SELECT * FROM table_name
LIMIT 1 OFFSET 2;)
```

2. Find employees with duplicate salaries

Table: employees(id, name, salary)

Values:

1, Shubham, 5000

2, Debug, 6000

3, Maurya, 7000

Expected Output: 5000, 2

Make sure yaha debug, 5000

```
SELECT name, COUNT(*) // Yaha count salary wise count krega because of group by salary
FROM employees
GROUP BY salary
HAVING COUNT(*) > 1;
```

3. Find employees who joined in the last 3 months

Table: employees(id, name, join_date)

Values:

1, Shubham, 2024-12-01,

2, Maurya, 2025-05-01

3, Debug, 2025-06-01

Expected Output: Maurya, Debug


```
SELECT name
FROM employees
WHERE join_date >= CURDATE() - INTERVAL 3 MONTH;
```

✓ **CURDATE()** ek MySQL function hai jo **aaj ki current date** deta hai.

For example: 2025-07-10 (agar aaj ki date hai).

✓ **INTERVAL** ek MySQL ka keyword hai jo tumhe date/time se kuch subtract/add karne deta hai.

👉 `CURDATE()` - `INTERVAL 3 MONTH` ka matlab hai:
Aaj ki date se **3 mahine pehle ki date** nikaal lo.

4. Get department-wise average salary 
Table: employees(id, name, department, salary)
Values:
1, Shubham, IT, 5000
2, Kumar, IT, 7000
3, Maurya, HR, 6000
Expected Output:(IT, 6000) (HR, 6000)

```
SELECT department, AVG(salary) as AVG_SALARY  
FROM employees  
GROUP BY department;
```

5. Fetch nth highest salary (e.g., 3rd)
Table: employees(id, name, salary)
Values:
1, Shubham, 3000
2, Kumar, 4000
3, Maurya, 5000
4, Debug, 6000
Expected Output:4000

Here we have to find the 3rd highest

```
SELECT DISTINCT salary  
FROM employees  
ORDER BY salary DESC  
LIMIT 1 OFFSET 2;
```

6. Find employees with salary more than their manager

Table: employees(id, name, salary, manager_id)

Values:

1, Shubham, 5000, NULL

2, Kumar, 6000, 1

3, Debug, 7000, 1

Expected Output: Kumar, Debug

```
SELECT e.name
```

```
FROM employees e
```

```
JOIN employees m ON e.manager_id = m.id
```

```
WHERE e.salary > m.salary;
```

7. Find all employees without a manager

Table: employees(id, name, manager_id)

Values:

1, Debug, NULL

2, Shubham, 1

Expected Output: Debug

```
SELECT *
```

```
FROM employees
```

```
WHERE manager_id IS NULL;
```

8. Get count of employees in each role

Table: employees(id, name, role)

Values:

1, A, Dev

2, B, Dev

3, C, QA

Expected Output:(Dev, 2) (QA, 1)

```
SELECT role, COUNT(*)
```

```
FROM employees
```

```
GROUP BY role;
```

9. Find top 3 highest paid employees

Table: employees(id, name, salary)

Values:

1, A, 3000

2, B, 4000

3, C, 5000

4, D, 6000

Expected Output:D, C, B

```
SELECT name
```

```
FROM employees
```

```
ORDER BY salary DESC
```

```
LIMIT 3 ;
```

10. Find common employees in two tables

Table: employees(name), contractors(name)

Values:

employees: A, B

contractors: B, C

Expected Output: B

```
SELECT e.name  
  
FROM employees e  
  
INNER JOIN contractors c  
  
ON e.name = c.name;
```

11. Find employees who earn the same as someone in 'Sales' department

Table: employees(id, name, salary, department)

Values:

1, A, 5000, Sales

2, B, 5000, IT

3, C, 6000, HR

Expected Output: A, B

```
SELECT name  
  
FROM employee  
  
WHERE salary IN  
  
(SELECT salary FROM employee WHERE department = "Sales");
```

12. List departments with more than 2 employees

Table: employees(id, name, department)

Values:

1 Alice, IT,

2 Bob , IT

3 Carol, IT

4 Dan ,HR

Expected Output:IT

```
SELECT department
FROM employees
GROUP BY department
HAVING COUNT(*) > 2;
```

13. Get employee names that start with A

Table: employees(id, name)

Values:

1, Shubham

2, Kumar

3, Maurya

Expected Output:Shubham, Maurya

O/P is wrong here

```
SELECT name
FROM employees
WHERE name LIKE 'A%';
```

14. Get employees who dont belong to any department

Table: employees(id, name, department_id)

Values:

1, A, NULL

2, B, 1

Expected Output:A

```
SELECT name
```

```
FROM employees
```

```
WHERE department_id IS NULL;
```

15. Join employees with department names

Table: employees(id, name, department_id), departments(id, department_name)

Values:

employees: 1, A, 10

employees: 2, B, 20

departments: 10, IT

departments:20, HR

Expected Output:(A, IT) (B, HR)

```
SELECT e.name, d.department_name
```

```
FROM employees e
```

```
INNER JOIN department d
```

```
ON e.department_id = d.id;
```

16. Find employees with the highest salary in each department

Table: employees(id, name, salary, department)

Example Data:

1, Shubham, 5000, IT

2, Kumar, 6000, IT

3, Debug, 7000, HR

Expected Output:(Kumar, IT) (Debug, HR)

```
SELECT name, department
FROM employees e
WHERE salary = (SELECT MAX(salary)
FROM employees
WHERE department = e.department);
```

17. Get number of employees who joined each year

Table: employees(id, name, join_date)

Example Data:

1, Shubham, 2022-01-01

2, Kumar, 2023-05-01

3, Mauya, 2023-06-01

Expected Output:(2022, 1) (2023, 2)

```
SELECT YEAR(join_date) AS join_year, COUNT(*)
FROM employees
GROUP BY YEAR(join_date);
```

18. Find employees whose name contains ar

Table: employees(id, name)

Values:

1, Shubham

2, Kumar

3, Maurya

```
SELECT name
```

```
FROM employees
```

```
WHERE name LIKE '%ar%';
```

20. Rank employees by salary

Table: employees(id, name, salary)

Values:

1, A, 3000

2, B, 4000

3, C, 5000

Expected Output:

C - Rank 1

B - Rank 2

A - Rank 3

```
SELECT name, RANK() OVER (ORDER BY salary DESC)
```

```
AS salary_rank
```

```
FROM employees;
```

20. Delete duplicate records (keeping one)

Table: employees(id, name, email)

Values:

1, DebugWithShubham, debugwithshubham@gmail.com

2, DebugWithShubham, debugwithshubham@gmail.com

3, Shubham, shubham@gmail.com

Expected Output:

Keep: id 1 or 2, and 3

```
DELETE FROM employees
```

```
WHERE id NOT IN(
```

```
SELECT MIN(id)
```

```
FROM employees
```

```
GROUP BY name, email);
```
