1. Select all columns from a table

sql

CopyEdit

SELECT * FROM employees;

🔹 **2. Find employees in a specific department**

sql

CopyEdit

SELECT name, department

FROM employees

WHERE department = 'HR';

🔹 **3. Get the total number of employees**

sql

CopyEdit

SELECT COUNT(*) FROM employees;

🔹 **4. Get the average salary**

sql

CopyEdit

SELECT AVG(salary) AS avg_salary FROM employees;

🔹 **5. Get maximum and minimum salary**

sql

CopyEdit

SELECT MAX(salary) AS max_salary, MIN(salary) AS min_salary

FROM employees;

🔹 **6. Sort employees by salary in descending order**

sql

CopyEdit

SELECT name, salary

FROM employees

ORDER BY salary DESC;

🔹 **7. Group employees by department and count them**

sql

CopyEdit

SELECT department, COUNT(*) AS total_employees

FROM employees

GROUP BY department;

🔹 **8. Filter grouped data using HAVING**

sql

CopyEdit

SELECT department, COUNT(*)

FROM employees

GROUP BY department

HAVING COUNT(*) > 5;

🔹 **9. Find duplicate entries by email**

sql

CopyEdit

```
SELECT email, COUNT(*)
FROM users
GROUP BY email
HAVING COUNT(*) > 1;
```

🔹 **10. Get employees hired in the last 30 days**

sql

CopyEdit

```
SELECT *
FROM employees
WHERE hire_date >= CURRENT_DATE - INTERVAL 30 DAY;
```

🔹 **11. Inner join example: Employees and Departments**

sql

CopyEdit

```
SELECT e.name, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.id;
```

🔹 **12. Left join: Show all employees even if they don't belong to a department**

sql

CopyEdit

```
SELECT e.name, d.department_name
FROM employees e
LEFT JOIN departments d ON e.department_id = d.id;
```

🔹 **13. Subquery to find employees earning above average**

sql

CopyEdit

```
SELECT name, salary
FROM employees
WHERE salary > (
  SELECT AVG(salary) FROM employees
);
```

🔹 **14. Find second highest salary**

sql

CopyEdit

```
SELECT MAX(salary)
FROM employees
WHERE salary < (
  SELECT MAX(salary) FROM employees
);
```

🔹 **15. Update salary for a specific employee**

sql

CopyEdit

```
UPDATE employees
SET salary = salary + 5000
WHERE name = 'John';
```

🔹 **16. Delete employees in a department**

sql

CopyEdit

DELETE FROM employees

WHERE department = 'Temporary';

17. Get employees whose name starts with 'A'

sql

CopyEdit

SELECT * FROM employees

WHERE name LIKE 'A%';

🔸 **18. Get employees whose name ends with 'n'**

sql

CopyEdit

SELECT * FROM employees

WHERE name LIKE '%n';

🔸 **19. Get employees with names containing 'an'**

sql

CopyEdit

SELECT * FROM employees

WHERE name LIKE '%an%';

🔸 **20. Get the top 5 highest paid employees**

sql

CopyEdit

SELECT name, salary FROM employees

ORDER BY salary DESC

LIMIT 5;

🔸 **21. Find employees between age 25 and 35**

sql

CopyEdit

SELECT * FROM employees

WHERE age BETWEEN 25 AND 35;

🔸 **22. Find employees not in 'HR' or 'Finance'**

sql

CopyEdit

SELECT * FROM employees

WHERE department NOT IN ('HR', 'Finance');

🔸 **23. List all distinct departments**

sql

CopyEdit

SELECT DISTINCT department FROM employees;

🔸 **24. Rename a column using alias**

sql

CopyEdit

```
SELECT name AS employee_name, salary AS monthly_salary
FROM employees;
```

🔸 **25. Find the number of employees who joined each year**

sql

CopyEdit

```
SELECT YEAR(joining_date) AS year, COUNT(*) AS total_employees
FROM employees
GROUP BY YEAR(joining_date);
```

🔸 **26. Find employees with NULL phone numbers**

sql

CopyEdit

```
SELECT * FROM employees
WHERE phone_number IS NULL;
```

🔸 **27. Get total salary paid department-wise**

sql

CopyEdit

```
SELECT department, SUM(salary) AS total_salary
FROM employees
GROUP BY department;
```

🔸 **28. Find the department with the highest average salary**

sql

CopyEdit

SELECT department, AVG(salary) AS avg_salary

FROM employees

GROUP BY department

ORDER BY avg_salary DESC

LIMIT 1;

🔸 **29. Find common employees between two tables**

sql

CopyEdit

SELECT name FROM employees

INTERSECT

SELECT name FROM contractors;

(Note: INTERSECT works in some DBs like PostgreSQL, but not in MySQL. For MySQL, use INNER JOIN.)

🔸 **30. Delete duplicate rows from a table**

sql

CopyEdit

DELETE FROM employees

WHERE id NOT IN (

 SELECT MIN(id)

 FROM employees

 GROUP BY email

);

1. INNER JOIN

Query:

sql

CopyEdit

SELECT e.name, d.department_name

FROM employees e

INNER JOIN departments d ON e.department_id = d.id;

Explanation:

- INNER JOIN returns only the rows where there is a **match in both tables**.
- Use this when you only care about employees that have departments.

Sample Output:

name	department_name
Alice	IT
Bob	HR

🔹 **2. LEFT JOIN**

Query:

sql

CopyEdit

SELECT e.name, d.department_name

FROM employees e

LEFT JOIN departments d ON e.department_id = d.id;

Explanation:

- Returns **all employees**, even those **without a department** (will show NULL).

Sample Output:

name	department_name
Alice	IT
Bob	HR
Charlie	NULL

🔹 **3. RIGHT JOIN**

Query:

sql

CopyEdit

```
SELECT e.name, d.department_name
FROM employees e
RIGHT JOIN departments d ON e.department_id = d.id;
```

Explanation:

- Returns **all departments**, even if **no employees** are assigned.

Sample Output:

name	department_name
Alice	IT
NULL	Marketing

🔹 **4. FULL OUTER JOIN *(Not available in MySQL directly)***

PostgreSQL/Oracle:

sql

CopyEdit

SELECT e.name, d.department_name

FROM employees e

FULL OUTER JOIN departments d ON e.department_id = d.id;

Explanation:

- Combines LEFT JOIN and RIGHT JOIN.
- Returns **all records** from both tables, filling in NULLs when no match.

🔹 5. WHERE Clause

Query:

sql

CopyEdit

SELECT name FROM employees

WHERE department = 'IT';

Explanation:

- Filters records to only those matching the condition.
- WHERE clause is used to narrow down results.

🔹 6. LIKE Operator

Query:

sql

CopyEdit

SELECT * FROM employees

WHERE name LIKE 'A%';

Explanation:

- LIKE is used for pattern matching.
- 'A%': starts with A

- '%n': ends with n
- '%an%': contains 'an'

🔹 7. Find Duplicates (Group By + HAVING)

Query:

sql

CopyEdit

SELECT name, COUNT(*)

FROM employees

GROUP BY name

HAVING COUNT(*) > 1;

Explanation:

- Groups rows by name.
- Returns names that appear more than once (i.e., duplicates).

🔹 8. Delete Duplicates (Keep One Row)

Query:

sql

CopyEdit

DELETE e1

FROM employees e1

JOIN employees e2

ON e1.name = e2.name AND e1.id > e2.id;

Explanation:

- Deletes all duplicates of name except the one with the **lowest ID**.

🔹 9. INTERSECT (Common Records Between Tables)

PostgreSQL/Oracle:

sql

CopyEdit

SELECT name FROM employees

INTERSECT

SELECT name FROM contractors;

Explanation:

- Returns records **common to both** tables.

✅ Not supported in MySQL. Use:

sql

CopyEdit

SELECT e.name

FROM employees e

INNER JOIN contractors c ON e.name = c.name;

🔹 10. JOIN + WHERE + LIKE

Query:

sql

CopyEdit

SELECT e.name, d.department_name

FROM employees e

JOIN departments d ON e.department_id = d.id

WHERE d.department_name LIKE '%tech%';

Explanation:

- Combines a JOIN with filtering.

- Only returns employees in departments with names that **contain 'tech'**.