

 **BASIC TABLE (for practice)**

STUDENT table

id	name	marks	dept
1	Ravi	85	CSE
2	Anu	78	ECE
3	John	90	CSE
4	Priya	88	MECH
5	Amit	70	CSE

 **SQL PRACTICE QUESTIONS**

1. Display all records from STUDENT table

SELECT * FROM student;

2. Display only name and marks

SELECT name, marks FROM student;

3. Find students with marks greater than 80

SELECT * FROM student WHERE marks > 80;

4. Display students from CSE department

```
SELECT * FROM student WHERE dept = 'CSE';
```

5. Find highest marks

```
SELECT MAX(marks) FROM student;
```

```
SELECT name  
FROM students  
WHERE marks = (SELECT MAX(marks) FROM  
students);
```

6. Find lowest marks

```
SELECT MIN(marks) FROM student;
```

7. Count total students

```
SELECT COUNT(*) FROM student;
```

8. Display students in ascending order of marks

```
SELECT * FROM student ORDER BY marks ASC;
```

9. Display students in descending order of marks

```
SELECT * FROM student ORDER BY marks DESC;
```

10. Find average marks

```
SELECT AVG(marks) FROM student;
```

11. Count students department-wise

```
SELECT dept, COUNT(*)
```

```
FROM student
```

```
GROUP BY dept;
```

12. Find departments having more than 1 student

```
SELECT dept, COUNT(*)
```

```
FROM student
```

```
GROUP BY dept
```

```
HAVING COUNT(*) > 1;
```

13. Find second highest marks

```
SELECT MAX(marks)
```

```
FROM student
```

```
WHERE marks < (SELECT MAX(marks) FROM student);
```

14. Display names starting with 'A'

```
SELECT * FROM student
```

```
WHERE name LIKE 'A%';
```

15. Display students whose name ends with 'i'

```
SELECT * FROM student
```

```
WHERE name LIKE '%i';
```

16. Display distinct departments

```
SELECT DISTINCT dept FROM student;
```

17. Find total marks of all students

```
SELECT SUM(marks) FROM student;
```

18. Update marks of Amit to 75

```
UPDATE student
```

```
SET marks = 75
```

```
WHERE name = 'Amit';
```

19. Delete record of Anu

```
DELETE FROM student
```

```
WHERE name = 'Anu';
```

20. Add new column 'age'

```
ALTER TABLE student ADD age INT;
```

JOINS PRACTICE

EMPLOYEE table

empid	name	deptid
1	Ravi	101
2	Anu	102
3	John	101

DEPARTMENT table

deptid	deptname
101	CSE
102	ECE

21. Inner Join Example

```
SELECT e.name, d.deptname  
FROM employee e  
INNER JOIN department d  
ON e.deptid = d.deptid;
```

22. Left Join Example

```
SELECT e.name, d.deptname  
FROM employee e
```

```
LEFT JOIN department d
```

```
ON e.deptid = d.deptid;
```

23. Find employees working in CSE

```
SELECT e.name
```

```
FROM employee e
```

```
JOIN department d
```

```
ON e.deptid = d.deptid
```

```
WHERE d.deptname = 'CSE';
```

ADVANCED INTERVIEW QUESTIONS

24. Find duplicate names

```
SELECT name, COUNT(*)
```

```
FROM student
```

```
GROUP BY name
```

```
HAVING COUNT(*) > 1;
```

25. Find top 3 students

```
SELECT * FROM student
```

```
ORDER BY marks DESC
```

LIMIT 3;

26. Display students whose marks between 70 and 90

SELECT * FROM student

WHERE marks BETWEEN 70 AND 90;

27. Rename table

RENAME TABLE student TO student_details;

28. Delete all records but keep structure

TRUNCATE TABLE student;

29. Delete table completely

DROP TABLE student;

30. Create a view

CREATE VIEW cse_students AS

SELECT * FROM student WHERE dept='CSE';

● SAMPLE TABLES (assume these)

EMPLOYEE

emp_id	name	salary	dept_id
1	Ravi	50000	10
2	Anu	60000	20
3	John	70000	10
4	Priya	80000	30
5	Amit	60000	20

DEPARTMENT

dept_id	dept_name
10	CSE
20	ECE
30	MECH

● GROUP BY & HAVING (VERY COMMON)

1 Count employees in each department

```
SELECT dept_id, COUNT(*)
```

```
FROM employee
```

```
GROUP BY dept_id;
```

2 Find average salary of each department

```
SELECT dept_id, AVG(salary)  
FROM employee  
GROUP BY dept_id;
```

3 Departments having more than 1 employee

```
SELECT dept_id, COUNT(*)  
FROM employee  
GROUP BY dept_id  
HAVING COUNT(*) > 1;
```

4 Departments where average salary > 60000

```
SELECT dept_id, AVG(salary)  
FROM employee  
GROUP BY dept_id  
HAVING AVG(salary) > 60000;
```

5 Find max salary in each department

```
SELECT dept_id, MAX(salary)  
FROM employee  
GROUP BY dept_id;
```

● JOINS (MOST IMPORTANT)

6 Display employee name with department name

```
SELECT e.name, d.dept_name
```

```
FROM employee e
```

```
JOIN department d
```

```
ON e.dept_id = d.dept_id;
```

7 Find employees working in CSE department

```
SELECT e.name
```

```
FROM employee e
```

```
JOIN department d
```

```
ON e.dept_id = d.dept_id
```

```
WHERE d.dept_name = 'CSE';
```

8 Count employees in each department (with join)

```
SELECT d.dept_name, COUNT(e.emp_id)
```

```
FROM employee e
```

```
JOIN department d
```

```
ON e.dept_id = d.dept_id
```

```
GROUP BY d.dept_name;
```

 **Departments having more than 1 employee (JOIN + HAVING)**

```
SELECT d.dept_name, COUNT(e.emp_id)
```

```
FROM employee e
```

```
JOIN department d
```

```
ON e.dept_id = d.dept_id
```

```
GROUP BY d.dept_name
```

```
HAVING COUNT(e.emp_id) > 1;
```

 **Employees without department (LEFT JOIN)**

```
SELECT e.name
```

```
FROM employee e
```

```
LEFT JOIN department d
```

```
ON e.dept_id = d.dept_id
```

```
WHERE d.dept_id IS NULL;
```

 **SUBQUERIES (INTERVIEW FAVORITE)**

  **Find employee with highest salary**

```
SELECT *
```

```
FROM employee
```

```
WHERE salary = (SELECT MAX(salary) FROM employee);
```

1 2 Find second highest salary

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

1 3 Employees earning more than average salary

```
SELECT *  
FROM employee  
WHERE salary > (SELECT AVG(salary) FROM employee);
```

1 4 Employees working in same department as Ravi

```
SELECT *  
FROM employee  
WHERE dept_id = (  
    SELECT dept_id FROM employee WHERE name = 'Ravi'  
)
```

1 5 Departments with employees earning > 70000

```
SELECT DISTINCT dept_id  
FROM employee
```

```
WHERE salary > 70000;
```

 **COMBINATION QUESTIONS (VERY HIGH CHANCE)**

1 6 Department with highest average salary

```
SELECT dept_id  
FROM employee  
GROUP BY dept_id  
ORDER BY AVG(salary) DESC  
LIMIT 1;
```

1 7 Employees whose salary is greater than department average

```
SELECT *  
FROM employee e  
WHERE salary > (  
    SELECT AVG(salary)  
    FROM employee  
    WHERE dept_id = e.dept_id  
);
```

1 8 Find duplicate salaries

```
SELECT salary, COUNT(*)
```

```
FROM employee  
  
GROUP BY salary  
  
HAVING COUNT(*) > 1;
```

1 9 Find departments with no employees

```
SELECT d.dept_name  
  
FROM department d  
  
LEFT JOIN employee e  
  
ON d.dept_id = e.dept_id  
  
WHERE e.emp_id IS NULL;
```

2 0 Employees earning highest salary in each department

```
SELECT *  
  
FROM employee e  
  
WHERE salary = (  
  
SELECT MAX(salary)  
  
FROM employee  
  
WHERE dept_id = e.dept_id  
  
);
```
