SQL query cheat sheet

🧠 SQL Query Cheat Sheet (from zero → advanced)

Save this. This covers **90%+ of real-world SQL queries** you'll ever write — interviews + daily work.

1️⃣ BASIC SELECT

sql

```sql
SELECT * FROM employees;
SELECT name, salary FROM employees;
```

Alias:

Copy code

sql

```sql
SELECT name AS emp_name, salary AS pay FROM employees;
```

Copy code

2️⃣ WHERE (Filtering)

sql

```sql
SELECT * FROM employees WHERE salary > 50000;
SELECT * FROM employees WHERE dept = 'IT';
```

Multiple conditions:

Copy code

sql

```sql
WHERE salary > 50000 AND dept = 'IT';
WHERE dept = 'IT' OR dept = 'HR';
WHERE NOT dept = 'HR';
```

Copy code

3 IN, BETWEEN, LIKE

sql

```sql
WHERE dept IN ('IT', 'HR');
WHERE salary BETWEEN 30000 AND 60000;
```

LIKE (pattern matching):

Copy code

sql

```sql
WHERE name LIKE 'A%';      -- starts with A
WHERE name LIKE '%a';      -- ends with a
WHERE name LIKE '%an%';    -- contains an
```

Copy code

4 ORDER BY & LIMIT

sql

```sql
ORDER BY salary ASC;
ORDER BY salary DESC;
```

Top N:

Copy code

sql

```sql
SELECT * FROM employees ORDER BY salary DESC LIMIT 5;
```

Copy code

5 AGGREGATE FUNCTIONS

sql

```sql
SELECT COUNT(*) FROM employees;
SELECT SUM(salary) FROM employees;
SELECT AVG(salary) FROM employees;
SELECT MAX(salary), MIN(salary) FROM employees;
```

Copy code

6 GROUP BY (VERY IMPORTANT)

sql

```sql
SELECT dept, COUNT(*)
FROM employees
GROUP BY dept;
```

With aggregates:

Copy code

sql

```sql
SELECT dept, AVG(salary)
FROM employees
GROUP BY dept;
```

Copy code

7️⃣ HAVING (filter AFTER GROUP BY)

❌ Wrong:

sql

```sql
WHERE COUNT(*) > 5
```

Copy code

✅ Correct:

sql

```sql
SELECT dept, COUNT(*)
FROM employees
GROUP BY dept
HAVING COUNT(*) > 5;
```

Copy code

8️⃣ DISTINCT

sql

```sql
SELECT DISTINCT dept FROM employees;
```

Copy code

9️⃣ JOINS (MOST ASKED)

INNER JOIN

sql

```sql
SELECT e.name, d.dept_name
FROM employees e
INNER JOIN departments d
ON e.dept_id = d.id;
```

Copy code

LEFT JOIN

sql

```sql
SELECT e.name, d.dept_name
FROM employees e
LEFT JOIN departments d
ON e.dept_id = d.id;
```

Copy code

RIGHT JOIN

sql

```sql
SELECT e.name, d.dept_name
FROM employees e
RIGHT JOIN departments d
ON e.dept_id = d.id;
```

Copy code

FULL OUTER JOIN (if supported)

sql

```sql
SELECT *
FROM employees e
FULL OUTER JOIN departments d
ON e.dept_id = d.id;
```

Copy code

🔟 NULL Handling

sql

```sql
WHERE manager_id IS NULL;
```

```
WHERE manager_id IS NOT NULL;
```

Copy code

Replace NULL:

sql

```
SELECT COALESCE(manager_id, 0) FROM employees;
```

Copy code

1️⃣1️⃣ SUBQUERIES

Single value:

sql

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

Multiple values:

sql

```
SELECT *
FROM employees
WHERE dept_id IN (
    SELECT id FROM departments WHERE location = 'HYD'
);
```

1️⃣2️⃣ EXISTS

Copy code

sql

```
SELECT *
FROM employees e
WHERE EXISTS (
    SELECT 1
    FROM departments d
```

```
    WHERE d.id = e.dept_id
);
```

Copy code

1️⃣3️⃣ CASE (IF-ELSE in SQL)

sql

```
SELECT name,
CASE
    WHEN salary >= 80000 THEN 'High'
    WHEN salary >= 50000 THEN 'Medium'
    ELSE 'Low'
END AS salary_level
FROM employees;
```

1️⃣4️⃣ WINDOW FUNCTIONS (ADVANCED 🔥)

Copy code

ROW_NUMBER

sql

```
SELECT *,
ROW_NUMBER() OVER (ORDER BY salary DESC) AS rn
FROM employees;
```

RANK vs DENSE_RANK

Copy code

sql

```
SELECT *,
RANK() OVER (ORDER BY salary DESC) AS rnk,
DENSE_RANK() OVER (ORDER BY salary DESC) AS drnk
FROM employees;
```

Partition (group without collapse)

Copy code

sql

```
SELECT *,
AVG(salary) OVER (PARTITION BY dept) AS dept_avg
```

```
FROM employees;
```

Copy code

1️⃣5️⃣ TOP-N PER GROUP (INTERVIEW FAVORITE)

sql

```
SELECT *
FROM (
    SELECT *,
    ROW_NUMBER() OVER (PARTITION BY dept ORDER BY salary DESC) rn
    FROM employees
) t
WHERE rn <= 3;
```

1️⃣6️⃣ UPDATE / DELETE

Copy code

sql

```
UPDATE employees
SET salary = salary + 5000
WHERE dept = 'IT';
```

sql

```
DELETE FROM employees
WHERE salary < 20000;
```

Copy code

1️⃣7️⃣ CREATE / ALTER

sql

```
CREATE TABLE employees (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    salary INT,
    dept_id INT
);
```

sql

```sql
ALTER TABLE employees ADD age INT;
ALTER TABLE employees DROP age;
```

Copy code

1️⃣8️⃣ INDEX (Performance)

sql

```sql
CREATE INDEX idx_salary ON employees(salary);
```

Copy code

🧠 GOLDEN ORDER OF SQL EXECUTION (VERY IMPORTANT)

text

```
FROM
JOIN
WHERE
GROUP BY
HAVING
SELECT
ORDER BY
LIMIT
```

🎯 MOST ASKED INTERVIEW QUESTIONS

Copy code

- Difference between `WHERE` and `HAVING`
- `INNER` vs `LEFT JOIN`
- `RANK` vs `DENSE_RANK`
- Find **Nth highest salary**
- Top N records per group
- Difference between `DELETE`, `TRUNCATE`, `DROP`
- `EXISTS` vs `IN`

If you want, next I can:

- 🔥 Give **50 interview SQL problems with answers**
- 🔥 Provide **real employee database schema + queries**
- 🔥 Share **MySQL vs PostgreSQL vs Oracle differences**
- 🔥 Create **one-page printable PDF cheat sheet**

Just tell me 😄