SQL Queries – Complete Cheat Sheet (Beginner to Advanced)

1️⃣ Database Queries

```sql
CREATE DATABASE db_name;
CREATE DATABASE IF NOT EXISTS db_name;
SHOW DATABASES;
USE db_name;
DROP DATABASE db_name;
```

2️⃣ Table Queries

```sql
CREATE TABLE table_name (
  id INT PRIMARY KEY,
  name VARCHAR(50),
  age INT
);

SHOW TABLES;
DESCRIBE table_name;
DROP TABLE table_name;
TRUNCATE TABLE table_name;
```

3️⃣ Insert Data

```sql
INSERT INTO table_name VALUES (1, 'Anurag', 22);

INSERT INTO table_name (name, age)
VALUES ('Rahul', 25);

INSERT INTO table_name VALUES
(1, 'A', 20),
(2, 'B', 21);
```

4️⃣ Select (Read Data)

```sql
SELECT * FROM table_name;
SELECT name, age FROM table_name;

SELECT DISTINCT age FROM table_name;
```

5️⃣ WHERE Conditions

```sql
SELECT * FROM table_name WHERE age > 18;
SELECT * FROM table_name WHERE name = 'Anurag';
SELECT * FROM table_name WHERE age BETWEEN 18 AND 25;
SELECT * FROM table_name WHERE name IN ('A', 'B');
SELECT * FROM table_name WHERE name LIKE 'A%';
```

6️⃣ Update Data

```sql
UPDATE table_name
SET age = 23
WHERE id = 1;
```

7️⃣ Delete Data

```sql
DELETE FROM table_name WHERE id = 1;
DELETE FROM table_name; -- deletes all rows
```

8️⃣ Aggregate Functions

```sql
SELECT COUNT(*) FROM table_name;
SELECT SUM(age) FROM table_name;
SELECT AVG(age) FROM table_name;
SELECT MIN(age) FROM table_name;
SELECT MAX(age) FROM table_name;
```

9️⃣ GROUP BY & HAVING

```sql
SELECT age, COUNT(*)
FROM table_name
GROUP BY age;

SELECT age, COUNT(*)
FROM table_name
GROUP BY age
HAVING COUNT(*) > 1;
```

🔟 ORDER BY & LIMIT

```sql
SELECT * FROM table_name ORDER BY age ASC;
SELECT * FROM table_name ORDER BY age DESC;
SELECT * FROM table_name LIMIT 5;
```

1️⃣1️⃣ Joins

```sql
-- INNER JOIN
SELECT * FROM A
INNER JOIN B ON A.id = B.id;

-- LEFT JOIN
SELECT * FROM A
LEFT JOIN B ON A.id = B.id;

-- RIGHT JOIN
SELECT * FROM A
RIGHT JOIN B ON A.id = B.id;

-- FULL JOIN (MySQL workaround)
SELECT * FROM A
LEFT JOIN B ON A.id = B.id
UNION
SELECT * FROM A
RIGHT JOIN B ON A.id = B.id;
```

1️⃣2️⃣ Subqueries

```
SELECT * FROM table_name
WHERE age > (SELECT AVG(age) FROM table_name);
```

1️⃣3️⃣ Constraints

```
PRIMARY KEY
FOREIGN KEY
UNIQUE
NOT NULL
CHECK
DEFAULT
```

1️⃣4️⃣ Indexes

```
CREATE INDEX idx_name ON table_name(name);
DROP INDEX idx_name ON table_name;
```

1️⃣5️⃣ Views

```
CREATE VIEW view_name AS
SELECT name, age FROM table_name;

SELECT * FROM view_name;
DROP VIEW view_name;
```

1️⃣6️⃣ Stored Procedures

```
CREATE PROCEDURE getUsers()
BEGIN
  SELECT * FROM table_name;
END;
```

```
CALL getUsers();
```

1️⃣7️⃣ Transactions

```
START TRANSACTION;
INSERT INTO table_name VALUES (1, 'A', 20);
COMMIT;
ROLLBACK;
```

1️⃣8️⃣ Useful Operators

```
AND, OR, NOT
IN, BETWEEN, LIKE
IS NULL, IS NOT NULL
```

🔝 Interview Tip

👉Master **SELECT, JOIN, GROUP BY, WHERE, Subquery** — 80% questions come from these.

If you want: - MySQL-only queries - PostgreSQL-only queries - SQL interview questions - Practice problems with solutions

Tell me 👍