

SQL

CHEAT SHEET

(Crash course)



By @engineer_bhaiya_yt

SQL Detailed Cheatsheet

1. What is SQL?

Structured Query Language (SQL) is a standard language for managing and manipulating relational databases. It allows users to create, retrieve, update, and delete data efficiently. SQL is widely used in database management systems like MySQL, PostgreSQL, SQL Server, and Oracle.

2. Data Types

SQL provides different data types to store various kinds of data in a structured format.

Numeric

- **INT** – Integer (whole number) used for storing whole numbers without decimals.
- **DECIMAL(p, s)** – Fixed precision number, used when exact decimal values are required.
- **FLOAT / DOUBLE** – Floating-point numbers used for approximate values with decimal points.

String

- **VARCHAR(n)** – Variable-length string, saves space by using only the necessary length.
- **CHAR(n)** – Fixed-length string, always consumes the specified length of space.
- **TEXT** – Large text storage, used for storing long text data.

Date & Time

- **DATE** – Stores date in YYYY-MM-DD format.
- **DATETIME** – Stores both date and time in YYYY-MM-DD HH:MI:SS format.
- **TIMESTAMP** – Stores Unix timestamp, which represents the number of seconds since 1970-01-01.
- **TIME** – Stores time in HH:MI:SS format.

3. DDL (Data Definition Language)

DDL is used to define and modify the structure of database objects like tables, indexes, and schemas.

Create Table

```
CREATE TABLE employees (  
    id INT PRIMARY KEY,  
    name VARCHAR(50),  
    age INT,  
    department VARCHAR(50),  
    salary DECIMAL(10,2)  
);
```

This statement creates a table named `employees` with specific columns.

Alter Table

```
ALTER TABLE employees ADD COLUMN join_date DATE;  
ALTER TABLE employees DROP COLUMN age;  
ALTER TABLE employees MODIFY COLUMN salary FLOAT;
```

The `ALTER TABLE` statement allows modifying an existing table structure.

Drop Table

```
DROP TABLE employees;
```

The `DROP TABLE` command permanently deletes a table and its data.

4. DML (Data Manipulation Language)

DML is used to manipulate existing records in a database.

Insert Data

```
INSERT INTO employees (id, name, department, salary) VALUES (1, 'John Doe', 'HR', 50000);
```

Inserts a new record into the **employees** table.

Update Data

```
UPDATE employees SET salary = 55000 WHERE id = 1;
```

Updates specific fields of existing records.

Delete Data

```
DELETE FROM employees WHERE id = 1;
```

Removes records from a table.

5. DQL (Data Query Language)

DQL is used to fetch data from the database.

Select Statements

```
SELECT * FROM employees;  
SELECT name, salary FROM employees;  
SELECT DISTINCT department FROM employees;
```

Retrieves data from a table with optional filters.

Where Clause

```
SELECT * FROM employees WHERE salary > 50000;  
SELECT * FROM employees WHERE department = 'HR' AND salary > 50000;
```

Filters data based on specific conditions.

Order By

```
SELECT * FROM employees ORDER BY salary DESC;
```

Sorts query results in ascending or descending order.

Limit

```
SELECT * FROM employees LIMIT 5;
```

Limits the number of records returned by a query.

6. Joins

Joins are used to combine rows from multiple tables based on related columns.

Inner Join

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

Returns matching rows from both tables.

Left Join

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

Returns all rows from the left table and matched rows from the right table.

Right Join

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

Returns all rows from the right table and matched rows from the left table.

9. Indexes

Indexes speed up query performance by creating pointers to data.

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

Indexes enhance search operations but require extra storage.

10. Views

Views store query results as a virtual table.

```
CREATE VIEW high_salary AS SELECT * FROM employees WHERE salary > 60000;  
SELECT * FROM high_salary;  
DROP VIEW high_salary;
```

11. Stored Procedures

Stored procedures contain a series of SQL statements that execute together.

```
DELIMITER //  
CREATE PROCEDURE GetHighSalaryEmployees()  
BEGIN  
    SELECT * FROM employees WHERE salary > 60000;  
END //  
DELIMITER ;  
  
CALL GetHighSalaryEmployees();
```

12. Transactions

Transactions ensure database consistency.

```
START TRANSACTION;  
UPDATE employees SET salary = salary + 5000 WHERE department = 'HR';  
COMMIT;
```

```
-- Rollback Example  
START TRANSACTION;  
DELETE FROM employees WHERE department = 'HR';  
ROLLBACK;
```

COMMIT saves changes, while **ROLLBACK** undoes changes.

13. Triggers

Triggers execute automatically in response to events.

```
CREATE TRIGGER before_insert_employee  
BEFORE INSERT ON employees  
FOR EACH ROW  
SET NEW.salary = IF(NEW.salary < 30000, 30000, NEW.salary);
```

14. Common Table Expressions (CTEs)

CTEs improve readability and reusability of queries.

```
WITH DepartmentSalary AS (  
    SELECT department, AVG(salary) AS avg_salary FROM employees GROUP BY  
    department  
)  
SELECT * FROM DepartmentSalary;
```

*Every **SELECT** query brings you closer to mastering data.*

💡 Join our WhatsApp channel for daily job updates and SQL tips! 🚀

<https://whatsapp.com/channel/0029VajSCJP2JI8EEDRRKp0F>