SQL Commands Cheat Sheet

1. SELECT

Used to select data from a database.

Example:

```
SELECT * FROM Employees;
```

2. WHERE

Filters records.

Example:

```
SELECT * FROM Employees WHERE department = 'HR';
```

3. ORDER BY

Sorts result in ascending or descending order.

Example:

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

4. GROUP BY

Groups rows sharing a property.

Example:

```
SELECT department, COUNT(*) FROM Employees GROUP BY department;
```

5. HAVING

Filters groups after GROUP BY.

Example:

```
SELECT department, COUNT(*) FROM Employees GROUP BY department HAVING COUNT(*) > 5;
```

6. LIMIT

Limits number of results returned.

Example:

```
SELECT * FROM Employees LIMIT 5;
```

7. DISTINCT

Returns only distinct values.

Example:

```
SELECT DISTINCT job_title FROM Employees;
```

8. JOINS

Combines rows from two or more tables based on related columns.

- INNER JOIN: Only matching rows

- LEFT JOIN: All from left + matched from right

- RIGHT JOIN: All from right + matched from left

- FULL OUTER JOIN: All rows from both sides

9. SUBQUERIES

A query inside another query.

Example:

```
SELECT name FROM Employees WHERE salary > (SELECT AVG(salary) FROM Employees);
```

10. CASE

Conditional logic.

Example:

```
SELECT name, CASE WHEN salary > 10000 THEN 'High' ELSE 'Low' END FROM Employees;
```

11. IN / NOT IN

Checks for values in a list.

Example:

```
SELECT name FROM Employees WHERE department IN ('HR', 'Sales');
```

12. BETWEEN

Filters in a range.

Example:

```
SELECT * FROM Employees WHERE salary BETWEEN 5000 AND 10000;
```

13. LIKE

Pattern matching.

Example:

```
SELECT * FROM Employees WHERE name LIKE 'A%';
```

14. IS NULL / IS NOT NULL

Checks for null values.

Example:

```
SELECT * FROM Employees WHERE manager_id IS NULL;
```

15. ALIAS (AS)

Temporary names for columns or tables.

Example:

```
SELECT name AS employee_name FROM Employees;
```

16. INSERT INTO

Adds new data.

Example:

```
INSERT INTO Employees (name, salary) VALUES ('John', 50000);
```

17. UPDATE

Modifies existing data.

Example:

```
UPDATE Employees SET salary = 60000 WHERE id = 1;
```

18. DELETE

Deletes records.

Example:

```
DELETE FROM Employees WHERE salary < 30000;
```

19. CREATE TABLE

Creates a new table.

Example:

```
CREATE TABLE Students (id INT, name VARCHAR(100));
```

20. DROP TABLE

Deletes a table.

Example:

```
DROP TABLE Students;
```

21. ALTER TABLE

Modifies table structure.

Example:

```
ALTER TABLE Students ADD age INT;
```

22. UNION / UNION ALL

Combines results of two SELECTs.

Example:

```
SELECT name FROM A UNION SELECT name FROM B;
```