

SQL Essentials — Cheat Sheet

FREE CHEAT SHEET

One page from the complete **SQL Essentials Companion Guide**. [Get the full guide →](#)

Query Shape

```
SELECT columns
FROM table
JOIN other ON table.x = other.y
WHERE condition
GROUP BY columns
HAVING aggregate_condition
ORDER BY columns
LIMIT n;
```

Comparison & Logic

OPERATOR	MEANING
<code>=</code> , <code>!=</code> / <code><></code>	Equal / not equal
<code>></code> , <code><</code> , <code>>=</code> , <code><=</code>	Greater/less (or equal)
<code>AND</code> , <code>OR</code> , <code>NOT</code>	Combine conditions
<code>BETWEEN a AND b</code>	Inclusive range
<code>IN (...)</code>	Matches list
<code>LIKE '%X%'</code>	Pattern match
<code>IS NULL</code>	Test for no value

Join Types

JOIN	RETURNS
<code>INNER JOIN</code>	Matches in both tables
<code>LEFT JOIN</code>	All of left, matched where possible
<code>FULL JOIN</code>	All of both, matched where possible

Aggregate Functions

FUNCTION	RETURNS
<code>COUNT(*)</code>	Row count
<code>SUM(x)</code> / <code>AVG(x)</code>	Total / average
<code>MIN(x)</code> / <code>MAX(x)</code>	Smallest / largest

Statements

TASK	STATEMENT
Add a row	<code>INSERT INTO t (cols) VALUES (...)</code>
Change rows	<code>UPDATE t SET col = v WHERE ...</code>
Remove rows	<code>DELETE FROM t WHERE ...</code>
Define a table	<code>CREATE TABLE t (col type, ...)</code>
Named subquery	<code>WITH n AS (SELECT ...) SELECT ...</code>
Reusable query	<code>CREATE VIEW n AS SELECT ...</code>

Transaction Safety

```
BEGIN;
UPDATE ...;
-- check it looks right, then:
COMMIT; -- or ROLLBACK;
```

Dialect Quick Notes

FEATURE	POSTGRES	MYSQL	SQLITE
Row limit	<code>LIMIT n</code>	<code>LIMIT n</code>	<code>LIMIT n</code>
Auto-increment	<code>SERIAL</code>	<code>AUTO_INCREMENT</code>	<code>INTEGER PK</code>
Case-insensitive	<code>ILIKE</code>	<code>LIKE</code>	<code>LIKE</code>

Golden Rules

RULE

`WHERE` filters rows. `HAVING` filters groups, after aggregation.

RULE

`NULL` is never equal to anything. Use `IS NULL`, never `= NULL`.

RULE

Test every `UPDATE` / `DELETE` filter with `SELECT` first.

RULE

Never run `UPDATE` or `DELETE` without a `WHERE` clause.

RULE

Prefer `NOT EXISTS` over `NOT IN` for correlated subqueries.