

SQL Quick Reference

A compact guide to the SQL clauses and functions analysts use most. Use it while practicing, cleaning data, summarizing results, or building joins.

Dialect note: SQL syntax differs across PostgreSQL, SQL Server, MySQL, Oracle, SQLite, and browser engines. Check the dialect column before using an example.

Function / clause	What it does	Example	Dialect	Level	Lab
SELECT	Chooses the columns or expressions returned by a query.	<code>SELECT name, district FROM tributes;</code>	Standard	Beginner	Yes
DISTINCT	Removes duplicate result rows.	<code>SELECT DISTINCT district FROM tributes;</code>	Standard	Beginner	Yes
WHERE	Keeps only rows that meet a condition.	<code>SELECT * FROM tributes WHERE score >= 8;</code>	Standard	Beginner	Yes
AND / OR / NOT	Combines or reverses filtering conditions.	<code>WHERE district = 12 AND score >= 8</code>	Standard	Beginner	Yes
IN	Matches any value in a list.	<code>WHERE district IN (4, 7, 12)</code>	Standard	Beginner	Yes
BETWEEN	Tests whether a value falls inside an inclusive range.	<code>WHERE score BETWEEN 7 AND 10</code>	Standard	Beginner	Yes
LIKE	Matches a text pattern using % and _ wildcards.	<code>WHERE name LIKE 'K%';</code>	Standard	Beginner	Yes
IS NULL	Finds missing values; use IS NOT NULL for present values.	<code>WHERE mentor IS NULL</code>	Standard	Beginner	Yes
ORDER BY	Sorts results ascending or descending.	<code>ORDER BY score DESC, name ASC</code>	Standard	Beginner	Yes
LIMIT	Returns only a specified number of rows.	<code>SELECT * FROM tributes LIMIT 5;</code>	PostgreSQL / MySQL / SQLite	Beginner	Yes
AS	Gives a column or table a temporary alias.	<code>SELECT score AS training_score FROM tributes;</code>	Standard	Beginner	Yes
COUNT()	Counts rows or non-null values.	<code>SELECT COUNT(*) AS tributes FROM tributes;</code>	Standard	Beginner	Yes
SUM()	Adds numeric values.	<code>SELECT SUM(points) FROM arena_events;</code>	Standard	Beginner	Yes
AVG()	Calculates the arithmetic mean.	<code>SELECT AVG(score) FROM tributes;</code>	Standard	Beginner	Yes
MIN() / MAX()	Returns the smallest or largest value.	<code>SELECT MIN(score), MAX(score) FROM tributes;</code>	Standard	Beginner	Yes
GROUP BY	Combines rows into groups for aggregate calculations.	<code>SELECT district, AVG(score) FROM tributes GROUP BY district;</code>	Standard	Beginner	Yes
HAVING	Filters groups after aggregation.	<code>GROUP BY district HAVING AVG(score) >= 8</code>	Standard	Intermediate	Yes
CASE	Creates conditional values inside a query.	<code>CASE WHEN score >= 8 THEN 'High' ELSE 'Other' END</code>	Standard	Intermediate	Yes

Function / clause	What it does	Example	Dialect	Level	Lab
COALESCE()	Returns the first non-null expression.	COALESCE(mentor, 'Unknown')	Standard	Beginner	Yes
NULLIF()	Returns NULL when two expressions are equal.	points / NULLIF(events, 0)	Standard	Intermediate	Yes
CAST()	Converts a value to another data type.	CAST(score AS DECIMAL(5,2))	Standard	Intermediate	Yes
ROUND()	Rounds a number to a chosen precision.	ROUND(AVG(score), 2)	Common	Beginner	Yes
LOWER() / UPPER()	Changes text to lower- or uppercase.	UPPER(name)	Standard	Beginner	Yes
TRIM()	Removes leading and trailing spaces.	TRIM(name)	Standard	Beginner	Yes
CONCAT()	Combines text values.	CONCAT(name, ' – District ', district)	Common	Beginner	Yes
INNER JOIN	Returns rows that match in both tables.	FROM tributes t JOIN arena_events e ON t.id = e.tribute_id	Standard	Intermediate	Yes
LEFT JOIN	Keeps every left-table row and matching right rows.	FROM tributes t LEFT JOIN arena_events e ON t.id = e.tribute_id	Standard	Intermediate	Yes
UNION	Stacks query results and removes duplicates.	SELECT name FROM tributes UNION SELECT name FROM mentors;	Standard	Intermediate	Yes
UNION ALL	Stacks query results and keeps duplicates.	SELECT name FROM tributes UNION ALL SELECT name FROM mentors;	Standard	Intermediate	Yes
WITH (CTE)	Names a temporary result used by the main query.	WITH ranked AS (SELECT * FROM tributes) SELECT * FROM ranked;	Standard	Intermediate	Yes
ROW_NUMBER()	Assigns a unique sequence number within a result.	ROW_NUMBER() OVER (ORDER BY score DESC)	Standard / PostgreSQL	Advanced	No
RANK()	Ranks rows with gaps after ties.	RANK() OVER (ORDER BY score DESC)	Standard / PostgreSQL	Advanced	No
LAG() / LEAD()	Reads a value from a previous or following row.	LAG(score) OVER (ORDER BY event_date)	Standard / PostgreSQL	Advanced	No
PERCENTILE_CONT()	Calculates an interpolated percentile, including the median.	PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY score)	PostgreSQL / SQL Server / Oracle	Advanced	No