SQL Cheat Sheet

Raghda Altaei

2024

Function Name	**Syntax**	**How it is Used**
SELECT	`SELECT column1, column2 FROM table_name;`	Retrieves specific columns from a table.

INNER JOIN	`SELECT t1.column, t2.column FROM table1 t1`
	`INNER JOIN table2 t2 ON t1.id = t2.id;`
		Combines rows from two tables when there is a match on the ON condition.

LEFT JOIN	`SELECT t1.column, t2.column FROM table1 t1`
	`LEFT JOIN table2 t2 ON t1.id = t2.id;`
		Returns all rows from the left table and matched rows from the right table.

RIGHT JOIN	`SELECT t1.column, t2.column FROM table1 t1`
	`RIGHT JOIN table2 t2 ON t1.id = t2.id;`
		Returns all rows from the right table and matched rows from the left table.

UNION	`SELECT column FROM table1 UNION`
	`SELECT column FROM table2;`
		Combines the result sets of two SELECT queries, removing duplicates.

GROUP BY and Aggregate Functions	`SELECT column, COUNT(*) FROM table_name`
	`GROUP BY column;`
		Groups rows with the same values and allows aggregate functions like COUNT, SUM, AVG.

ORDER BY	`SELECT column FROM table_name`
	`ORDER BY column DESC;`
		Sorts the result set in ascending (ASC) or descending (DESC) order.

LIMIT	`SELECT column FROM table_name LIMIT 10;`	Limits the number of rows returned by the query.
WHERE Clause with Comparison Operators	`SELECT column FROM table_name WHERE condition;`	Filters the result set based on a condition. Common operators: =, ¿, ¡, BETWEEN, LIKE.

Subqueries	`SELECT column FROM table_name`
	`WHERE column = (SELECT column FROM another_table);`
		A query within another query. The subquery executes first.

CASE Statement	`SELECT column, CASE WHEN condition THEN result`
	`ELSE result2 END FROM table_name;`
		Adds conditional logic to SQL queries for different outputs based on conditions.

Creating Aliases	`SELECT column AS alias_name FROM table_name;`	Renames a column or table temporarily for readability.

DISTINCT	`SELECT DISTINCT column FROM`
	`table_name;`
		Removes duplicate rows in the result set.

String Concatenation (SQLite)	`SELECT FirstName || LastName`
	`AS FullName FROM table_name;`
		Concatenates two or more strings.

DATE Functions	`SELECT * FROM table_name`
	`WHERE InvoiceDate >=`
	`DATE('now', '-15 years');`
		Returns records where the date is older than 15 years from today.

HAVING Clause	`SELECT column, COUNT(*) FROM`
	`table_name`
	`GROUP BY column HAVING`
	`COUNT(*) > 1;`
		Filters groups after they have been created by the GROUP BY clause.

IN and NOT IN	`SELECT column FROM table_name`
	`WHERE column IN ('value1',`
	`'value2', 'value3');`
		Filters rows where the column value matches one of the specified values.

IS NULL and IS NOT NULL	`SELECT column FROM table_name`
	`WHERE column IS NULL;`
		Checks for NULL values in a column.

String Pattern Matching	`SELECT column FROM table_name`
	`WHERE column LIKE 'A%';`
		Finds all rows where the column starts with 'A'. % represents any sequence of characters.

JOIN Multiple Tables	`SELECT t1.column, t2.column, t3.column FROM table1 t1`
	`JOIN table2 t2 ON t1.id = t2.id`
	`JOIN table3 t3 ON t2.id = t3.id;`
		Joins more than two tables by combining them on matching column values.

EXISTS Clause	`SELECT column FROM table_name`
	`WHERE EXISTS (SELECT 1 FROM another_table WHERE condition);`
		Checks for the existence of rows in a subquery.

CAST Function	`SELECT CAST(column AS data_type) FROM table_name;`	Converts a value from one data type to another, e.g., converting a number to a string.