

SQL Cheat Sheet: FUNCTIONS and Implicit JOIN



Command	Syntax	Description	Example
COUNT	<code>SELECT COUNT(column_name) FROM table_name WHERE condition;</code>	<code>COUNT</code> function returns the number of rows that matches a specified criterion.	<code>SELECT COUNT(dep_id) FROM employees;</code>
AVG	<code>SELECT AVG(column_name) FROM table_name WHERE condition;</code>	<code>AVG</code> function returns the average value of a numeric column.	<code>SELECT AVG(salary) FROM employees;</code>
SUM	<code>SELECT SUM(column_name) FROM table_name WHERE condition;</code>	<code>SUM</code> function returns the total sum of a numeric column.	<code>SELECT SUM(salary) FROM employees;</code>
MIN	<code>SELECT MIN(column_name) FROM table_name WHERE condition;</code>	<code>MIN</code> function returns the smallest value of the <code>SELECT</code> d column.	<code>SELECT MIN(salary) FROM employees;</code>
MAX	<code>SELECT MAX(column_name) FROM table_name WHERE condition;</code>	<code>MAX</code> function returns the largest value of the <code>SELECT</code> d column.	<code>SELECT MAX(salary) FROM employees;</code>
ROUND	<code>SELECT ROUND(2number, decimals, operation) AS RoundValue;</code>	<code>ROUND</code> function rounds a number to a specified number of decimal places.	<code>SELECT ROUND(salary) FROM employees;</code>
LENGTH	<code>SELECT LENGTH(column_name) FROM table;</code>	<code>LENGTH</code> function returns the length of a string (in bytes).	<code>SELECT LENGTH(f_name) FROM employees;</code>
UCASE	<code>SELECT UCASE(column_name) FROM table;</code>	<code>UCASE</code> function that displays the column name in each table in uppercase.	<code>SELECT UCASE(f_name) FROM employees;</code>
DISTINCT	<code>SELECT DISTINCT(column_name) FROM table;</code>	<code>DISTINCT</code> function is used to display data without duplicates.	<code>SELECT DISTINCT(UCASE(f_name)) FROM employees;</code>
DAY	<code>SELECT DAY(column_name) FROM table</code>	<code>DAY</code> function returns the day of the month for a given date	<code>SELECT DAY(b_date) FROM employees where emp_id = 'E1002';</code>
CURRENT DATE	<code>SELECT (CURRENT DATE - COLUMN) FROM table;</code>	<code>CURRENT DATE</code> is used to display the current date.This can be subtracted from the previous date to get the difference.	<code>SELECT YEAR(CURRENT DATE - b_date) As AGE, CURRENT_DATE, b_date FROM employees;</code>
Subquery	<code>SELECT column_name [, column_name] FROM table1 [, table2] WHERE column_name OPERATOR (SELECT column_name [, column_name] FROM table1 [, table2] [WHERE])</code>	<code>Subquery</code> is a query within another SQL query and embedded within the <code>WHERE</code> clause. A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.	<code>SELECT emp_id, fname, lname, salary FROM employees where salary < (SELECT AVG(salary) FROM employees);</code> <code>SELECT * FROM (SELECT emp_id, f_name, l_name, dep_id FROM employees) AS emp4all;</code> <code>SELECT * FROM employees WHERE job_id IN (SELECT job_ident FROM jobs);</code>
Implicit Inner Join	<code>SELECT column_name(s) FROM table1, table2 WHERE table1.column_name = table2.column_name;</code>	<code>Implicit Inner Join</code> combines the two or more records but displays only matching values in both tables. Inner join applies only the specified columns.	<code>SELECT * FROM employees, jobs where employees.job_id = jobs.job_ident;</code>

Implicit Cross Join	<code>SELECT column_name(s) FROM table1, table2;</code>	Implicit Cross Join defines as a Cartesian product where the number of rows in the first table multiplied by the number of rows in the second table..	<code>SELECT * FROM employees, jobs;</code>
---------------------	---	---	---

Author(s)

[Lakshmi Holla](#)

Changelog

Date	Version	Changed by	Change Description
2021-07-28	1.0	Lakshmi Holla	Initial Version