

# SQL Cheat Sheet: FUNCTIONS and Implicit JOIN

Command	Syntax (MySQL/DB2)	Description	Example (MySQL/DB2)
COUNT	<pre>SELECT COUNT(column_name) FROM table_name WHERE condition;</pre>	<b>COUNT</b> function returns the number of rows that match a specified criterion.	<pre>SELECT COUNT(dep_id) FROM employees;</pre>
AVG	<pre>SELECT AVG(column_name) FROM table_name WHERE condition;</pre>	<b>AVG</b> function returns the average value of a numeric column.	<pre>SELECT AVG(salary) FROM employees;</pre>
SUM	<pre>SELECT SUM(column_name) FROM table_name WHERE condition;</pre>	<b>SUM</b> function returns the total sum of a numeric column.	<pre>SELECT SUM(salary) FROM employees;</pre>
MIN	<pre>SELECT MIN(column_name) FROM table_name WHERE condition;</pre>	<b>MIN</b> function returns the smallest value of the SELECTED column.	<pre>SELECT MIN(salary) FROM employees;</pre>
MAX	<pre>SELECT MAX(column_name) FROM table_name WHERE condition;</pre>	<b>MAX</b> function returns the largest value of the SELECTED column.	<pre>SELECT MAX(salary) FROM employees;</pre>
ROUND	<pre>SELECT ROUND(2number, decimals, operation) AS RoundValue;</pre>	<b>ROUND</b> function rounds a number to a specified number of decimal places.	<pre>SELECT ROUND(salary) FROM employees;</pre>
LENGTH	<pre>SELECT LENGTH(column_name) FROM table;</pre>	<b>LENGTH</b> function returns the length of a string (in bytes).	<pre>SELECT LENGTH(f_name) FROM employees;</pre>
UCASE	<pre>SELECT UCASE(column_name) FROM table;</pre>	<b>UCASE</b> function displays the column name in each table in uppercase.	<pre>SELECT UCASE(f_name) FROM employees;</pre>
LCASE	<pre>SELECT LCASE(column_name) FROM table;</pre>	<b>LCASE</b> function displays the column name in each table in lowercase.	<pre>SELECT LCASE(f_name) FROM employees;</pre>
DISTINCT	<pre>SELECT DISTINCT column_name FROM table;</pre>	<b>DISTINCT</b> function is used to display data without duplicates.	<pre>SELECT DISTINCT UCASE(f_name) FROM employees;</pre>
DAY	<pre>SELECT DAY(column_name) FROM table</pre>	<b>DAY</b> function returns the day of the month for a given date.	<pre>SELECT DAY(b_date) FROM employees where emp_id = 'E1002';</pre>

CURRENT_DATE	<pre>SELECT CURRENT_DATE;</pre>	<code>CURRENT_DATE</code> is used to display the current date.	<pre>SELECT CURRENT_DATE;</pre>
DATEDIFF()	<pre>SELECT DATEDIFF(date1, date2);</pre>	<code>DATEDIFF()</code> is used to calculate the difference between two dates or time stamps. The default value generated is the difference in number of days.	<pre>SELECT DATEDIFF(CURRENT_DATE, date_column) FROM table;</pre>
FROM_DAYS()	<pre>SELECT FROM_DAYS(number_of_days);</pre>	<code>FROM_DAYS()</code> is used to convert a given number of days to YYYY-MM-DD format.	<pre>SELECT FROM_DAYS(DATEDIFF(CURRENT_DATE, date_column)) FROM table;</pre>
DATE_ADD()	<pre>SELECT DATE_ADD(date, INTERVAL n type);</pre>	<code>DATE_ADD()</code> is used to calculate the date after lapse of mentioned number of units of date type, i.e. if n=3 and type=DAY, the result is a date 3 days after what is mentioned in date column. The type variable can also be months or years.	<pre>SELECT DATE_ADD(date, INTERVAL 3 DAY);;</pre>
DATE_SUB()	<pre>SELECT DATE_SUB(date, INTERVAL n type);</pre>	<code>DATE_SUB()</code> is used to calculate the date prior to the record date by mentioned number of units of date type, i.e. if n=3 and type=DAY, the result is a date 3 days before what is mentioned in date column. The type variable can also be months or years.	<pre>SELECT DATE_SUB(date, INTERVAL 3 DAY);;</pre>
Subquery	<pre>SELECT column_name [, column_name ] FROM table1 [, table2 ] WHERE column_name OPERATOR (SELECT column_name [, column_name ] FROM table1 [, table2 ] [WHERE])</pre>	<p><code>Subquery</code> is a query within another SQL query and embedded within the WHERE clause.</p> <p>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.</p>	<pre>SELECT emp_id, f_name, l_name, salary FROM employees where salary &lt; (SELECT AVG(salary) FROM employees);</pre> <pre>SELECT * FROM ( SELECT emp_id, f_name, l_name, dep_id FROM employees) AS emp4all;</pre> <pre>SELECT * FROM employees WHERE job_id IN (SELECT job_id FROM jobs);</pre>

Implicit Inner Join	<pre>SELECT column_name(s) FROM table1, table2 WHERE table1.column_name = table2.column_name;</pre>	<p><b>Implicit Inner Join</b> combines two or more records but displays only matching values in both tables. Inner join applies only the specified columns.</p>	<pre>SELECT * FROM employees, jobs where employees.job_id = jobs.job_id;</pre>
Implicit Cross Join	<pre>SELECT column_name(s) FROM table1, table2;</pre>	<p><b>Implicit Cross Join</b> is defined as a Cartesian product where the number of rows in the first table is multiplied by the number of rows in the second table.</p>	<pre>SELECT * FROM employees, jobs;</pre>