

Hoja de trucos de SQL: FUNCIONES y JOIN implícito

Comando	Sintaxis (MySQL/DB2)	Descripción	Ejemplo (MySQL/DB2)
COUNT	SELECT COUNT(column_name) FROM table_name WHERE condition;	La función COUNT devuelve el número de filas que coinciden con un criterio especificado.	SELECT COUNT(dep_id) FROM employees;
AVG	SELECT AVG(column_name) FROM table_name WHERE condition;	La función AVG devuelve el valor promedio de una columna numérica.	SELECT AVG(salary) FROM employees;
SUM	SELECT SUM(column_name) FROM table_name WHERE condition;	La función SUM devuelve la suma total de una columna numérica.	SELECT SUM(salary) FROM employees;
MIN	SELECT MIN(column_name) FROM table_name WHERE condition;	La función MIN devuelve el valor más pequeño de la columna SELECCIONADA.	SELECT MIN(salary) FROM employees;
MAX	SELECT MAX(column_name) FROM table_name WHERE condition;	La función MAX devuelve el valor más grande de la columna SELECCIONADA.	SELECT MAX(salary) FROM employees;
ROUND	SELECT ROUND(2number, decimals, operation) AS RoundValue;	La función ROUND redondea un número a un número especificado de decimales.	SELECT ROUND(salary) FROM employees;
LENGTH	SELECT LENGTH(column_name) FROM table;	La función LENGTH devuelve la longitud de una cadena (en bytes).	SELECT LENGTH(f_name) FROM employees;
UCASE	SELECT UCASE(column_name) FROM table;	La función UCASE muestra el nombre de la columna en cada tabla en mayúsculas.	SELECT UCASE(f_name) FROM employees;
LCASE	SELECT LCASE(column_name) FROM table;	La función LCASE muestra el nombre de la columna en cada tabla en minúsculas.	SELECT LCASE(f_name) FROM employees;
DISTINCT	SELECT DISTINCT column_name FROM table;	La función DISTINCT se utiliza para mostrar datos sin duplicados.	SELECT DISTINCT UCASE(f_name) FROM employees;
DAY	SELECT DAY(column_name) FROM table	La función DAY devuelve el día del mes para una fecha dada.	SELECT DAY(b_date) FROM employees where emp_id = 'E1002';
CURRENT_DATE	SELECT CURRENT_DATE;	CURRENT_DATE se utiliza para mostrar la fecha actual.	SELECT CURRENT_DATE;
DATEDIFF()	SELECT DATEDIFF(date1, date2);	DATEDIFF() se utiliza para calcular la diferencia entre dos fechas o marcas de tiempo. El valor predeterminado generado es la diferencia en el número de días.	SELECT DATEDIFF(CURRENT_DATE, date_column) FROM table;
FROM_DAYS()	SELECT FROM_DAYS(number_of_days);	FROM_DAYS() se utiliza para convertir un número dado de días al formato YYYY-MM-DD.	SELECT FROM_DAYS(DATEDIFF(CURRENT_DATE, date_column)) FROM table;
DATE_ADD()	SELECT DATE_ADD(date, INTERVAL n type);	DATE_ADD() se utiliza para calcular la fecha después de transcurrido el número mencionado de unidades de tipo de fecha, es decir, si n=3 y type=DÍA, el resultado es una fecha 3 días después de lo que se menciona en la columna de fecha. El valor de tipo también puede ser meses o años.	SELECT DATE_ADD(date, INTERVAL 3 DAY);;
DATE_SUB()	SELECT DATE_SUB(date, INTERVAL n type);	DATE_SUB() se utiliza para calcular la fecha anterior a la fecha del registro por el número mencionado de unidades de tipo de fecha, es decir, si n=3 y type=DÍA, el resultado es una fecha 3 días antes de lo que se menciona en la columna de fecha. El valor de tipo también puede ser meses o años.	SELECT DATE_SUB(date, INTERVAL 3 DAY);;
Subconsulta	SELECT column_name [, column_name] FROM table1 [, table2] WHERE column_name OPERATOR (SELECT column_name [, column_name] FROM table1 [, table2] [WHERE])	<p>Una subconsulta es una consulta dentro de otra consulta SQL y está incrustada dentro de la cláusula WHERE.</p> <p>Una subconsulta se utiliza para devolver datos que se utilizarán en la consulta principal como condición para restringir aún más los datos que se van a recuperar.</p>	<p>SELECT emp_id, f_name, l_name, salary FROM employees where salary < (SELECT AVG(salary) FROM employees);</p> <p>SELECT * FROM (SELECT emp_id, f_name, l_name, dep_id FROM employees) AS emp4all;</p> <p>SELECT * FROM employees WHERE job_id IN (SELECT job_id FROM jobs);</p>
Join interno implícito	SELECT column_name(s) FROM table1, table2 WHERE table1.column_name = table2.column_name;	El Join interno implícito combina dos o más registros pero muestra solo los valores coincidentes en ambas tablas. El join interno aplica solo a las columnas especificadas.	SELECT * FROM employees, jobs where employees.job_id = jobs.job_id;

Join cruzado implicito	<code>SELECT column_name(s) FROM table1, table2;</code>	El Join cruzado implícito se define como un producto cartesiano donde el número de filas en la primera tabla se multiplica por el número de filas en la segunda tabla.	<code>SELECT * FROM employees, jobs;</code>
---------------------------	---	--	---

Autor(es)

Lakshmi Holla
Abhishek Gagneja

