

SQL Cheat Sheet: Intermediate - LIKE, ORDER BY, GROUP BY

Command	Syntax (MySQL/DB2)	Description	Example (MySQL/DB2)
LIKE	SELECT column1, column2, ... FROM table_name WHERE column_name LIKE pattern;	Like operator is used in a WHERE clause to search for a specified pattern in a column. Two wildcards often used in conjunction with the LIKE operator are percent sign(%) and underscore sign(_), depending upon the SQL engine being used.	MySQL : <code>SELECT * FROM employees WHERE address LIKE '%Right_1%'</code> This command - will output all entries with Right_1 in the Address.
BETWEEN	SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value1 AND value2;	The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates. The BETWEEN operator is inclusive; both end values are included.	MySQL : <code>SELECT * FROM employees WHERE salary BETWEEN 4000 AND 8000;</code> This generates all records of employees with salaries between 4000 and 8000.
ORDER BY	SELECT column1, column2, ... FROM table_name ORDER BY column1, column2, ... ASC/DESC;	order BY keyword is used to sort the result set in ascending or descending order. The default is ascending. In case of multiple columns in ORDER BY, the sorting will be done in the sequence of the appearance of the arguments.	MySQL : <code>SELECT * FROM employees ORDER BY emp_id DESC, 1_name;</code> This command - first sorts the employees by their department IDs and then sorts them by their last names.
GROUP BY	SELECT column_name(s) FROM table_name GROUP BY column_name(s);	group BY clause is used in collaboration with the SELECT statement to arrange data into identical values into groups.	MySQL : <code>SELECT emp_id, COUNT(*) AS "EMPLOYEE_COUNT" FROM EMPLOYEE GROUP BY emp_id;</code> This command - gives department IDs and the number of employees as them, grouped by the department IDs.
HAVING	SELECT column_name(s) FROM table_name GROUP BY column_name(s) HAVING condition;	having clause is used in conjunction with group BY clause in collaboration with the SELECT statement in order to filter the data as per the given condition and thus group as per identical values of a specified parameter.	MySQL : <code>SELECT emp_id, COUNT(*) AS "EMPLOYEE_COUNT", AVG(salary) AS "AVG_SALARY" FROM EMPLOYEE GROUP BY emp_id HAVING count(*) > 5 ORDER BY avg_salary;</code>

Author(s)



Skills Network