

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 column1 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.	SELECT * FROM employees WHERE address LIKE '16141a%'; This command will output all entries with 16141a 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: begin and end values are included.	SELECT * FROM employees WHERE salary BETWEEN 40000 AND 80000; This generates all records of employees with salaries between 40000 and 80000.
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.	SELECT * FROM employees ORDER BY emp_id DESC, last_name; This displays the first name, last name, and department ID of employees, list sorted in descending order of department IDs and then sorted alphabetically as per 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 with identical values into groups.	SELECT emp_id, COUNT(*) FROM employees GROUP BY emp_id; This returns the department IDs and the number of employees in 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 then group as per identical values of a specified parameter.	SELECT dep_id, COUNT(*) AS "num_employees", AVG(salary) AS "avg_salary" FROM employees GROUP BY dep_id HAVING count(*) < 4 ORDER BY avg_salary;

Author(s)

[Lakshmi Nallu](#)  
[Srinivas Kumar](#)  
[Srinivas Kumar](#)

