02:38

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 columnN 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 f_name , l_name FROM employees WHERE address LIKE '%Elgin,IL%';` This command will output all entries with `Elgin,IL` 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 f_name, l_name, dep_id FROM employees ORDER BY dep_id DESC, l_name;` This displays the first name, last name, and department ID of employees, first 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 dep_id, COUNT(*) FROM employees GROUP BY dep_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 Holla
Abhishek Gagneja

Skills Network

IBM