CODING NINJAS

Data Analytics With SQL

Grouping

Group BY

- Used to group rows that have the same values.
- It summarizes data from the database.
- The GROUP BY clause returns one row for each group.

Group BY

- Can be used with -
  - Sum
  - Count
  - Min
  - Max
  - Avg

Grouping with multiple columns

- Syntax -
  - *SELECT * FROM table GROUP BY col1, col2*
- This will display the total records in each group and sub group.

Having

Having

- WHERE keyword can not be used with group functions.
- We have to use HAVING clause in SELECT statement to specify filter conditions for grouped results.
- If the GROUP BY clause is omitted, the HAVING clause behaves like the WHERE clause.

Order By

- used to sort the result-set in ascending or descending order.
- Sorts the records in ascending order by default.
- To sort the records in descending order, use DESC keyword.

In and not In

IN

- The IN operator is a shorthand for multiple OR conditions.
- The IN operator allows you to determine if a specified value matches any value from a list or from a subquery.
- Syntax -

  SELECT column1,column2,...FROM table_name

  WHERE (expr|column_1) IN ('value1','value2',...);
- The values in the list must be separated by a comma (,).

Not IN

- You can combine the IN operator with the NOT operator to determine if a value does not match any value in a list or a subquery.

Between

Between

- We can use BETWEEN clause to replace a combination of "greater than equal AND less than equal" conditions.
- Syntax -
    - SELECT *column_name(s)* FROM *table_name* WHERE *column_name* BETWEEN *value1* AND *value2;*
- It will return the records where *expression* is within the range of *value1* and *value2* (inclusive).
- Values can be numbers, text or dates.

Between

- Can be used with different commands
  - Select
  - Update
  - Delete
  - IN

Not Between

- It will return all rows where value does not lie in the given range.

Like

Like

- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
- There are two wildcards used in conjunction with the LIKE operator:
  - % : The percent sign represents zero, one, or multiple characters
  - _ : The underscore represents a single character
- Syntax -
  - *SELECT column1, column2, ...FROM table_name WHERE column LIKE pattern;*

Examples

Statements	Description
LIKE 'S%'	It finds any value which starts with 'S'.
LIKE '%S%'	It finds any value which have 'S' in any position.
LIKE '_SS%'	It finds any value which have 'SS' in the second and third positions.
LIKE 'S_%_%'	It finds any value which starts with 'S' and have at least three characters in length.
LIKE '%S'	It finds any value which ends with 'S'.
LIKE '_S%P'	It finds any value which have 'S' in the second position and ends with 'P'.
LIKE 'S___P'	It finds any value in a five digit numbers which start with 'S' and ends with 'P'.

Escape Characters

- Let's say you wanted to search for a % or a _ character in the MySQL LIKE condition. You can do this using an Escape character.
  - *SELECT * FROM table_name WHERE column_name LIKE 'G\%';*
- We can override the default escape character in MySQL by providing the ESCAPE modifier as follows:
  - *SELECT * FROM table_name WHERE column_name LIKE 'G!%' ESCAPE '!';*