MySQL - WHERE Clause

MySQL Database Training For Beginners

39 Lectures 5.5 hours

Simon Sez IT

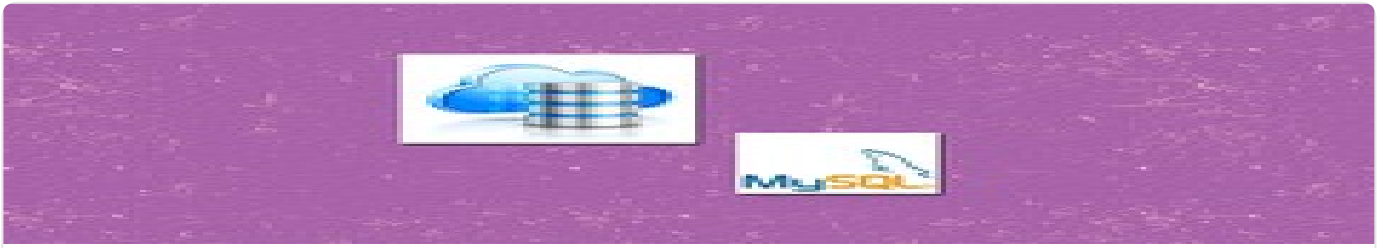
More Detail

Python Programming With MySQL Database: From Scratch

152 Lectures 16 hours

Metla Sudha Sekhar

More Detail



Learn MySQL From Scratch For Data Science And Analytics

87 Lectures 5.5 hours

Metla Sudha Sekhar

More Detail

We have seen the SQL **SELECT** command to fetch data from a MySQL table. We can use a conditional clause called the **WHERE Clause** to filter out the results. Using this WHERE clause, we can specify a selection criteria to select the required records from a table.

Syntax

The following code block has a generic SQL syntax of the SELECT command with the WHERE clause to fetch data from the MySQL table −

```
SELECT field1, field2,...fieldN table_name1, table_name2...
[WHERE condition1 [AND [OR]] condition2.....
```

- You can use one or more tables separated by a comma to include various conditions using a WHERE clause, but the WHERE clause is an optional part of the SELECT command.
- You can specify any condition using the WHERE clause.
- You can specify more than one condition using the **AND** or the **OR** operators.
- A WHERE clause can be used along with DELETE or UPDATE SQL command also to specify a condition.

The **WHERE** clause works like an **if condition** in any programming language. This clause is used to compare the given value with the field value available in a MySQL table. If the given value from outside is equal to the available field value in the MySQL table, then it returns that row.

Here is the list of operators, which can be used with the **WHERE** clause.

Assume field A holds 10 and field B holds 20, then −

Operator	Description	Example
=	Checks if the values of the two operands are equal or not, if yes, then the condition becomes true.	(A = B) is not true.
!=	Checks if the values of the two operands are equal or not, if the values are not equal then the condition becomes true.	(A != B) is true.
>	Checks if the value of the left operand is greater than the value of the right operand, if yes, then the condition becomes true.	(A > B) is not true.
<	Checks if the value of the left operand is less than the value of the right operand, if yes then the condition becomes true.	(A < B) is true.
>=	Checks if the value of the left operand is greater than or equal to the value of the right operand, if yes, then the condition becomes true.	(A >= B) is not true.
<=	Checks if the value of the left operand is less than or equal to the value of the right operand, if yes, then the condition becomes true.	(A <= B) is true.

The WHERE clause is very useful when you want to fetch the selected rows from a table, especially when you use the **MySQL Join**. Joins are discussed in another chapter.

It is a common practice to search for records using the **Primary Key** to make the search faster.

If the given condition does not match any record in the table, then the query would not return any row.

Fetching Data from the Command Prompt

This will use the SQL SELECT command with the WHERE clause to fetch the selected data from the MySQL table – **tutorials_tbl**.

Example

The following example will return all the records from the **tutorials_tbl** table for which the author name is **Sanjay**.

```
root@host# mysql -u root -p password;
Enter password:*******
mysql> use TUTORIALS;
Database changed
```

```
Database changed
mysql> SELECT * from tutorials_tbl WHERE tutorial_author = 'Sanjay';
+-------------+----------------+-----------------+-----------------+
| tutorial_id | tutorial_title | tutorial_author | submission_date |
+-------------+----------------+-----------------+-----------------+

|      3      | JAVA Tutorial  |      Sanjay     |    2007-05-21   |
+-------------+----------------+-----------------+-----------------+
1 rows in set (0.01 sec)

mysql>
```

Unless performing a **LIKE** comparison on a string, the comparison is not case sensitive. You can make your search case sensitive by using the **BINARY** keyword as follows −

```
root@host# mysql -u root -p password;
Enter password:*******
mysql> use TUTORIALS;
Database changed
mysql> SELECT * from tutorials_tbl \
   WHERE BINARY tutorial_author = 'sanjay';
Empty set (0.02 sec)

mysql>
```

Fetching Data Using a PHP Script

PHP uses **mysqli query()** or **mysql_query()** function to select records in a MySQL table using where clause. This function takes two parameters and returns TRUE on success or FALSE on failure.

Syntax

```
$mysqli->query($sql,$resultmode)
```

Sr.No.	Parameter & Description
1	**$sql** Required - SQL query to select records in a MySQL table using Where Clause.
2	**$resultmode** Optional - Either the constant MYSQLI_USE_RESULT or MYSQLI_STORE_RESULT depending on the desired behavior. By default, MYSQLI_STORE_RESULT is used.

Example

Try the following example to select a record using where clause in a table −

Copy and paste the following example as mysql_example.php −

```
<html>
   <head>
      <title>Using Where Clause</title>
   </head>
   <body>
      <?php
         $dbhost = 'localhost';
         $dbuser = 'root';
         $dbpass = 'root@123';
         $dbname = 'TUTORIALS';

         $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

         if($mysqli->connect_errno ) {
            printf("Connect failed: %s<br />", $mysqli->connect_error);
            exit();
         }
         printf('Connected successfully.<br />');

         $sql = 'SELECT tutorial_id, tutorial_title, tutorial_author, subm
            FROM tutorials_tbl where tutorial_author = "Mahesh"';

         $result = $mysqli->query($sql);

         if ($result->num_rows > 0) {
            while($row = $result->fetch_assoc()) {
               printf("Id: %s, Title: %s, Author: %s, Date: %d <br />",
                  $row["tutorial_id"],
```

```
                  $row["tutorial_title"],
                  $row["tutorial_author"],
                  $row["submission_date"]);
            }
         } else {
            printf('No record found.<br />');
         }
         mysqli_free_result($result);
         $mysqli->close();
      ?>
   </body>
</html>
```

Output

Access the mysql_example.php deployed on apache web server and verify the output. Here we've entered multiple records in the table before running the select script.

```
Connected successfully.
Id: 1, Title: MySQL Tutorial, Author: Mahesh, Date: 2021
Id: 2, Title: HTML Tutorial, Author: Mahesh, Date: 2021
Id: 3, Title: PHP Tutorial, Author: Mahesh, Date: 2021
```