MYSQL WHERE CLAUSE

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

Syntax:

Here is generic SQL syntax of SELECT command with WHERE clause to fetch data from MySQL table:

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

- You can use one or more tables separated by comma to include various conditions using a WHERE clause, but WHERE clause is an optional part of SELECT command.

- You can specify any condition using WHERE clause.

- You can specify more than one conditions using **AND** or **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 given value with the field value available in MySQL table. If given value from outside is equal to the available field value in MySQL table, then it returns that row.

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

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

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

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

It is a common practice to search records using **Primary Key** to make search fast.

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

Fetching Data from Command Prompt:

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

Example:

Following example will return all the records from **tutorials_tbl** table for which author name is **Sanjay**:

```
root@host# mysql -u root -p password;
Enter password:*******
mysql> use TUTORIALS;
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 using **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 PHP Script:

You can use same SQL SELECT command with WHERE CLAUSE into PHP function **mysql_query**. This function is used to execute SQL command and later another PHP function **mysql_fetch_array** can be used to fetch all the selected data. This function returns row as an associative array, a numeric array, or both. This function returns FALSE if there are no more rows.

Example:

Following example will return all the records from **tutorials_tbl** table for which author name is **Sanjay**:

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $conn )
{
  die('Could not connect: ' . mysql_error());
}
$sql = 'SELECT tutorial_id, tutorial_title,
```

```php
                tutorial_author, submission_date
        FROM tutorials_tbl
        WHERE tutorial_author="Sanjay"';

mysql_select_db('TUTORIALS');
$retval = mysql_query( $sql, $conn );
if(! $retval )
{
  die('Could not get data: ' . mysql_error());
}
while($row = mysql_fetch_array($retval, MYSQL_ASSOC))
{
    echo "Tutorial ID :{$row['tutorial_id']}  <br> ".
         "Title: {$row['tutorial_title']} <br> ".
         "Author: {$row['tutorial_author']} <br> ".
         "Submission Date : {$row['submission_date']} <br> ".
         "--------------------------------<br>";
}
echo "Fetched data successfully\n";
mysql_close($conn);
?>
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js