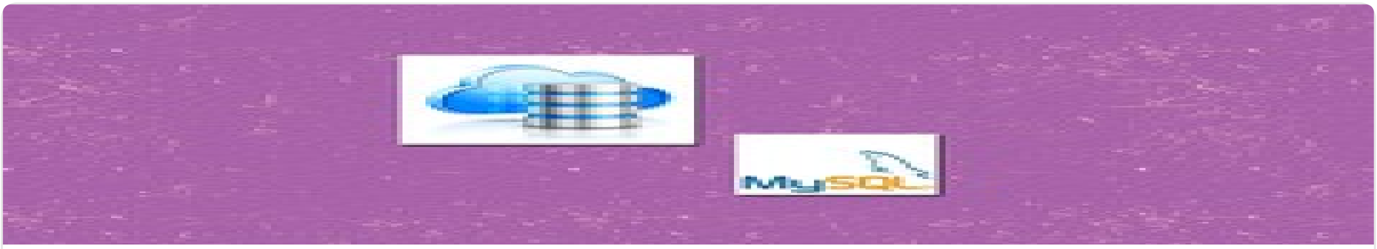
MySQL - Insert Query



To insert data into a MySQL table, you would need to use the SQL **INSERT INTO** command. You can insert data into the MySQL table by using the mysql> prompt or by using any script like PHP.

Syntax

Here is a generic SQL syntax of INSERT INTO command to insert data into the MySQL table −

```
INSERT INTO table_name ( field1, field2,...fieldN )
   VALUES
   ( value1, value2,...valueN );
```

To insert string data types, it is required to keep all the values into double or single quotes. For example **"value"**.

Inserting Data from the Command Prompt

To insert data from the command prompt, we will use SQL INSERT INTO command to insert data into MySQL table tutorials_tbl.

Example

The following example will create 3 records into **tutorials_tbl** table −

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

mysql> INSERT INTO tutorials_tbl
   ->(tutorial_title, tutorial_author, submission_date)
   ->VALUES
   ->("Learn PHP", "John Poul", NOW());
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO tutorials_tbl
   ->(tutorial_title, tutorial_author, submission_date)
   ->VALUES
   ->("Learn MySQL", "Abdul S", NOW());
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO tutorials_tbl
   ->(tutorial_title, tutorial_author, submission_date)
   ->VALUES
   ->("JAVA Tutorial", "Sanjay", '2007-05-06');
Query OK, 1 row affected (0.01 sec)
mysql>
```

NOTE − Please note that all the arrow signs (->) are not a part of the SQL command. They are indicating a new line and they are created automatically by the MySQL prompt while pressing the enter key without giving a semicolon at the end of each line of the command.

In the above example, we have not provided a tutorial_id because at the time of table creation, we had given AUTO_INCREMENT option for this field. So MySQL takes care of inserting these IDs automatically. Here, **NOW()** is a MySQL function, which returns the current date and time.

Inserting Data Using a PHP Script

PHP uses **mysqli query()** or **mysql_query()** function to insert a record into a MySQL table. This function takes two parameters and returns TRUE on success or FALSE on failure.

Syntax

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

Sr.No.	Parameter & Description
1	**$sql** Required - SQL query to insert record into a table.
2	**$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

This example will take three parameters from the user and will insert them into the MySQL table − −

Copy and paste the following example as mysql_example.php −

```
<html>
   <head><title>Add New Record in MySQL Database</title></head>
   <body>
      <?php
         if(isset($_POST['add'])) {
            $dbhost = 'localhost';
            $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 />');

            if(! get_magic_quotes_gpc() ) {
               $tutorial_title = addslashes ($_POST['tutorial_title']);
               $tutorial_author = addslashes ($_POST['tutorial_author']);
            } else {
               $tutorial_title = $_POST['tutorial_title'];
               $tutorial_author = $_POST['tutorial_author'];
            }

            $submission_date = $_POST['submission_date'];
            $sql = "INSERT INTO tutorials_tbl ".
               "(tutorial_title,tutorial_author, submission_date) "."VALUE
               "('$tutorial_title','$tutorial_author','$submission_date')"

            if ($mysqli->query($sql)) {
               printf("Record inserted successfully.<br />");
            }
            if ($mysqli->errno) {
               printf("Could not insert record into table: %s<br />", $mysq
            }
            $mysqli->close();
         } else {
      ?>
      <form method = "post" action = "<?php $_PHP_SELF ?>">
         <table width = "600" border = "0" cellspacing = "1" cellpadding =
            <tr>
               <td width = "250">Tutorial Title</td>
               <td><input name = "tutorial_title" type = "text" id = "tuto
            </tr>
            <tr>
               <td width = "250">Tutorial Author</td>
               <td><input name = "tutorial_author" type = "text" id = "tut
            </tr>
            <tr>
               <td width = "250">Submission Date [   yyyy-mm-dd ]</td>
               <td><input name = "submission_date" type = "text" id = "sub
               </tr>
               <tr>
                  <td width = "250"> </td>
                  <td></td>
               </tr>
               <tr>
                  <td width = "250"> </td>
                  <td><input name = "add" type = "submit" id = "add"  value =
               </tr>
            </table>
         </form>
      <?php
         }
      ?>
      </body>
   </html>
```

Output

Access the mysql_example.php deployed on apache web server, enter details and verify the output on submitting the form.

```
Record inserted successfully.
```

While doing a data insert, it is best to use the function **get_magic_quotes_gpc()** to check if the current configuration for magic quote is set or not. If this function returns false, then use the function **addslashes()** to add slashes before the quotes.

You can put many validations around to check if the entered data is correct or not and can take the appropriate action.