PHP BASIC

PHP Introduction
PHP Getting Started
PHP Syntax
PHP Variables
PHP Constants
PHP Echo and Print
PHP Data Types
PHP Strings
PHP Operators
PHP If…Else
PHP Switch…Case
PHP Arrays
PHP Sorting Arrays
PHP Loops
PHP Functions
PHP Math Operations
PHP GET and POST

PHP ADVANCED

PHP Date and Time
PHP Include Files
PHP File system
PHP Parsing Directories
PHP File Upload
PHP File Download
PHP Cookies
PHP Sessions
PHP Send Email
PHP Form Handling
PHP Form Validation
PHP Filters
PHP Error Handling
PHP Classes and Objects
PHP Magic Constants
PHP JSON Parsing
PHP Regular Expressions
PHP Exception Handling

PHP & MySQL DATABASE

PHP MySQL Introduction
PHP MySQL Connect
PHP MySQL Create Database
PHP MySQL Create Table
PHP MySQL Insert
PHP MySQL Prepared
PHP MySQL Last Inserted ID
PHP MySQL Select
PHP MySQL Where
PHP MySQL Limit
PHP MySQL Order By
PHP MySQL Update
PHP MySQL Delete
PHP MySQL CRUD Application

PHP MySQL INSERT Query

‹ ›

In this tutorial you will learn how to insert records in a MySQL table using PHP.

Inserting Data into a MySQL Database Table

Now that you've understood how to create database and tables in MySQL. In this tutorial you will learn how to execute SQL query to insert records into a table.

The `INSERT INTO` statement is used to insert new rows in a database table.

Let's make a SQL query using the `INSERT INTO` statement with appropriate values, after that we will execute this insert query through passing it to the PHP `mysqli_query()` function to insert data in table. Here's an example, which insert a new row to the *persons* table by specifying values for the *first_name*, *last_name* and *email* fields.

Example Procedural Object Oriented PDO ⤓ Download

```php
<?php
/* Attempt MySQL server connection. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
$link = mysqli_connect("localhost", "root", "", "demo");

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Attempt insert query execution
$sql = "INSERT INTO persons (first_name, last_name, email) VALUES ('Peter', 'Parker', 'peterparker@mail.com')";
if(mysqli_query($link, $sql)){
    echo "Records inserted successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

// Close connection
mysqli_close($link);
?>
```

If you remember from the preceding chapter, the *id* field was marked with the `AUTO_INCREMENT` flag. This modifier tells the MySQL to automatically assign a value to this field if it is left unspecified, by incrementing the previous value by 1.

Inserting Multiple Rows into a Table

You can also insert multiple rows into a table with a single insert query at once. To do this, include multiple lists of column values within the `INSERT INTO` statement, where column values for each row must be enclosed within parentheses and separated by a comma.

Let's insert few more rows into the *persons* table, like this:

PHP MySQL Ajax Search
PHP MySQL Login System

PHP EXAMPLES

PHP Practice Examples
PHP FAQ's Answers

PHP REFERENCE

PHP String Functions
PHP Array Functions
PHP File System Functions
PHP Date/Time Functions
PHP Calendar Functions
PHP MySQLi Functions
PHP Filters
PHP Error Levels
More references

Example Procedural Object Oriented PDO ⤓ Download

```php
<?php
/* Attempt MySQL server connection. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
$link = mysqli_connect("localhost", "root", "", "demo");

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Attempt insert query execution
$sql = "INSERT INTO persons (first_name, last_name, email) VALUES
            ('John', 'Rambo', 'johnrambo@mail.com'),
            ('Clark', 'Kent', 'clarkkent@mail.com'),
            ('John', 'Carter', 'johncarter@mail.com'),
            ('Harry', 'Potter', 'harrypotter@mail.com')";
if(mysqli_query($link, $sql)){
    echo "Records added successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

// Close connection
mysqli_close($link);
?>
```

Now, go to phpMyAdmin (`http://localhost/phpmyadmin/`) and check out the *persons* table data inside *demo* database. You will find the value for the *id* column is assigned automatically by incrementing the value of previous *id* by 1.

Note: Any number of line breaks may occur within a SQL statement, provided that any line break does not break off keywords, values, expression, etc.

Insert Data into a Database from an HTML Form

In the previous section, we have learned how to insert data into database from a PHP script. Now, we'll see how we can insert data into database obtained from an HTML form. Let's create an HTML form that can be used to insert new records to *persons* table.

Step 1: Creating the HTML Form

Here's a simple HTML form that has three text `<input>` fields and a submit button.

Example ⤓ Download

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Add Record Form</title>
</head>
<body>
<form action="insert.php" method="post">
    <p>
        <label for="firstName">First Name:</label>
        <input type="text" name="first_name" id="firstName">
    </p>
    <p>
        <label for="lastName">Last Name:</label>
        <input type="text" name="last_name" id="lastName">
    </p>
    <p>
        <label for="emailAddress">Email Address:</label>
        <input type="text" name="email" id="emailAddress">
    </p>
    <input type="submit" value="Submit">
</form>
</body>
</html>
```

Step 2: Retrieving and Inserting the Form Data

When a user clicks the submit button of the add record HTML form, in the example above, the form data is sent to 'insert.php' file. The 'insert.php' file connects to the MySQL database server, retrieves forms fields using the PHP `$_REQUEST` variables and finally execute the insert query to add the records. Here is the complete code of our 'insert.php' file:

Example   Procedural   Object Oriented   PDO   ⤓ Download

```
<?php
/* Attempt MySQL server connection. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
$link = mysqli_connect("localhost", "root", "", "demo");

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Escape user inputs for security
$first_name = mysqli_real_escape_string($link, $_REQUEST['first_name']);
$last_name = mysqli_real_escape_string($link, $_REQUEST['last_name']);
$email = mysqli_real_escape_string($link, $_REQUEST['email']);

// Attempt insert query execution
$sql = "INSERT INTO persons (first_name, last_name, email) VALUES ('$first_name', '$last_name', '$email')";
if(mysqli_query($link, $sql)){
    echo "Records added successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

// Close connection
mysqli_close($link);
?>
```

In the next chapter we will extend this insert query example and take it one step further by implementing the prepared statement for better security and performance.

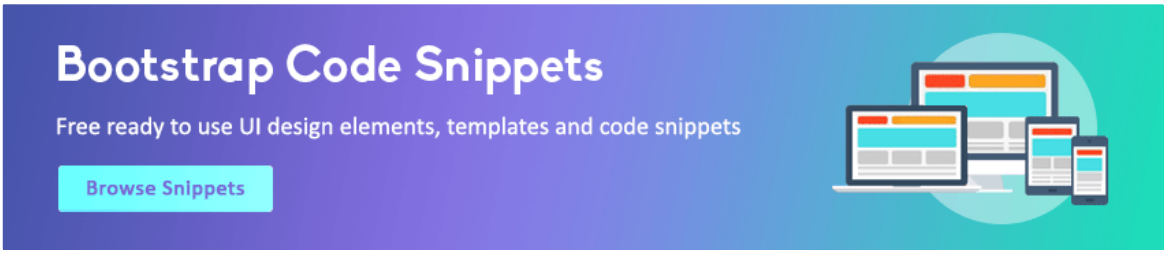
Note: The `mysqli_real_escape_string()` function escapes special characters in a string and create a legal SQL string to provide security against SQL injection.

This is very basic example of inserting the form data in a MySQL database table. You can extend this example and make it more interactive by adding validations to the user inputs before inserting it to the database tables. Please check out the tutorial on PHP form validation to learn more about sanitizing and validating user inputs using PHP.

« PREVIOUS PAGE   NEXT PAGE »

Build AI into SQL

MindsDB


Bootstrap Code Snippets
Free ready to use UI design elements, templates and code snippets
Browse Snippets

Is this website helpful to you? Please give us a like, or share your feedback to help us improve. Connect with us on Facebook and Twitter for the latest updates.

ABOUT US

- Our Story
- Terms of Use
- Privacy Policy

CONTACT

- Contact Us
- Report Error
- Advertise

INTERACTIVE TOOLS

- Font Awesome Icon Search Utility
- Title & Meta Length Calculator
- Bootstrap Button Generator
- Bootstrap Icon Finder
- HTML Formatter
- HTML Color Picker
- SQL Playground
- HTML Editor

Share This: