PHP MySQL SELECT Query

In this tutorial you'll learn how to select records from a MySQL table using PHP.

Selecting Data From Database Tables

So far you have learnt how to create database and table as well as inserting data. Now it's time to retrieve data what have inserted in the preceding tutorial. The SQL `SELECT` statement is used to select the records from database tables. Its basic syntax is as follows:

```
SELECT column1_name, column2_name, columnN_name FROM table_name;
```

Let's make a SQL query using the `SELECT` statement, after that we will execute this SQL query through passing it to the PHP `mysqli_query()` function to retrieve the table data.

Consider our *persons* database table has the following records:

```
+----+------------+-----------+----------------------+
| id | first_name | last_name | email                |
+----+------------+-----------+----------------------+
|  1 | Peter      | Parker    | peterparker@mail.com |
|  2 | John       | Rambo     | johnrambo@mail.com   |
|  3 | Clark      | Kent      | clarkkent@mail.com   |
|  4 | John       | Carter    | johncarter@mail.com  |
|  5 | Harry      | Potter    | harrypotter@mail.com |
+----+------------+-----------+----------------------+
```

The PHP code in the following example selects all the data stored in the *persons* table (using the asterisk character (*) in place of column name selects all the data in the table).

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 select query execution
$sql = "SELECT * FROM persons";
if($result = mysqli_query($link, $sql)){
    if(mysqli_num_rows($result) > 0){
        echo "<table>";
            echo "<tr>";
                echo "<th>id</th>";
                echo "<th>first_name</th>";
                echo "<th>last_name</th>";
                echo "<th>email</th>";
            echo "</tr>";
        while($row = mysqli_fetch_array($result)){
```

```
            echo "<tr>";
                echo "<td>" . $row['id'] . "</td>";
                echo "<td>" . $row['first_name'] . "</td>";
                echo "<td>" . $row['last_name'] . "</td>";
                echo "<td>" . $row['email'] . "</td>";
            echo "</tr>";
        }
        echo "</table>";
        // Free result set
        mysqli_free_result($result);
    } else{
        echo "No records matching your query were found.";
    }
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

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

Explanation of Code (Procedural style)

In the example above, the data returned by the `mysqli_query()` function is stored in the `$result` variable. Each time `mysqli_fetch_array()` is invoked, it returns the next row from the result set as an array. The while loop is used to loops through all the rows in the result set. Finally the value of individual field can be accessed from the row either by passing the field index or field name to the `$row` variable like `$row['id']` or `$row[0]`, `$row['first_name']` or `$row[1]`, `$row['last_name']` or `$row[2]`, and `$row['email']` or `$row[3]`.

If you want to use the `for` loop you can obtain the loop counter value or the number of rows returned by the query by passing the `$result` variable to the `mysqli_num_rows()` function. This loop counter value determines how many times the loop should run.

« PREVIOUS PAGE NEXT PAGE »