



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

ADVERTISEMENTS

Stock baskets for first timers

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

```
1 <?php
2 /* Attempt MySQL server connection. Assuming you are running MySQL
3 server with default setting (user 'root' with no password) */
4 $link = mysqli_connect("localhost", "root", "", "demo");
5
6 // Check connection
7 if($link === false){
8     die("ERROR: Could not connect. " . mysqli_connect_error());
9 }
10
11 // Attempt select query execution
12 $sql = "SELECT * FROM persons";
13 if($result = mysqli_query($link, $sql)){
14     if(mysqli_num_rows($result) > 0){
15         echo "<table>";
16         echo "<tr>";
17         echo "<th>id</th>";
18         echo "<th>first_name</th>";
19         echo "<th>last_name</th>";
20         echo "<th>email</th>";
21         echo "</tr>";
22         while($row = mysqli_fetch_array($result)){
```

PHP EXAMPLES

PHP REFERENCE

```
23         echo "<tr>";
24         echo "<td>" . $row['id'] . "</td>";
25         echo "<td>" . $row['first_name'] . "</td>";
26         echo "<td>" . $row['last_name'] . "</td>";
27         echo "<td>" . $row['email'] . "</td>";
28         echo "</tr>";
29     }
30     echo "</table>";
31     // Free result set
32     mysqli_free_result($result);
33 } else{
34     echo "No records matching your query were found.";
35 }
36 } else{
37     echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
38 }
39
40 // Close connection
41 mysqli_close($link);
42 ?>
```

Explanation of Code (Procedural style)

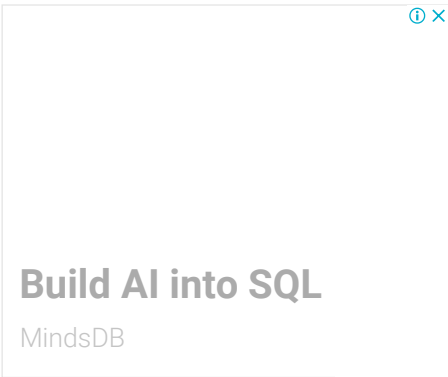
In the example above, the data returned by the `mysqli_query()` function is stored in the `$result` variable. Each time `mysqli_fetch_array()` is invoked, it returns the next row from the result set as an array. The `while loop` is used to loops through all the rows in the result set. Finally the value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

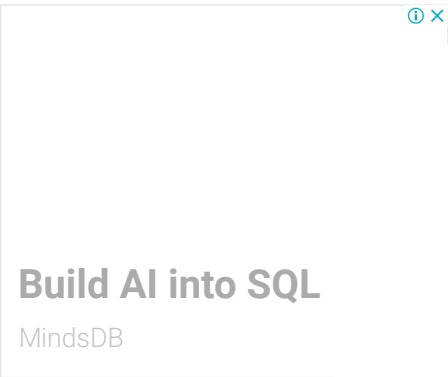
NEXT PAGE »

ADVERTISEMENTS



Build AI into SQL

MindsDB




Build AI into SQL

MindsDB

Bootstrap Code Snippets

Free ready to use UI design elements, templates and code snippets

[Browse Snippets](#)



ABOUT US

- Our Story
- Terms of Use
- Privacy Policy

CONTACT

- Contact Us
- Report Error
- Advertise

INTERACTIVE TOOLS

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

