**PHP MySQLi Functions: mysqli\_query, mysqli\_connect, mysqli\_fetch\_array**

PHP has a rich collection of built in functions for manipulating MySQL databases.

**In this tutorial, you will learn-**

* [mysqli\_connect function](https://www.guru99.com/mysql-php-and-other-database-access-methods.html#1)
* [mysqli\_select\_db function](https://www.guru99.com/mysql-php-and-other-database-access-methods.html#2)
* [mysqli\_query function](https://www.guru99.com/mysql-php-and-other-database-access-methods.html#3)
* [mysqli\_num\_rows function](https://www.guru99.com/mysql-php-and-other-database-access-methods.html#4)
* [mysqli\_fetch\_array function](https://www.guru99.com/mysql-php-and-other-database-access-methods.html#5)
* [mysqli\_close function](https://www.guru99.com/mysql-php-and-other-database-access-methods.html#6)
* [PHP Data Access Object PDO](https://www.guru99.com/mysql-php-and-other-database-access-methods.html#7)

**PHP mysqli\_connect function**

The PHP mysql connect function is used to connect to a MySQL database server.

It has the following syntax.

<?php;

$db\_handle = mysqli\_connect($db\_server\_name, $db\_user\_name, $db\_password);

?>

HERE,

* “$db\_handle” is the database connection resource variable.
* “mysqli\_connect(…)” is the function for php database connection
* “$server\_name” is the name or IP address of the server hosting MySQL server.
* “$user\_name” is a valid user name in MySQL server.
* “$password” is a valid password associated with a user name in MySQL server.

**PHP mysqli\_select\_db function**

The mysqli\_select\_db function is used to select a database.

It has the following syntax.

<?php

mysqli\_select\_db($db\_handle,$database\_name);

?>

  HERE,

* “mysqli\_select\_db(…)” is the database selection function that returns either true or false
* “$database\_name” is the name of the database
* “$link\_identifier” is optional, it is used to pass in the server connection link

**PHP mysqli\_query function**

The mysqli\_query function is used to execute[SQL](https://www.guru99.com/sql.html)queries.

The function can be used to execute the following query types;

* Insert
* Select
* Update
* delete

It has the following syntax.

<?php

mysqli\_query($db\_handle,$query) ;

?>

  HERE,

* “mysqli\_query(…)” is the function that executes the SQL queries.
* “$query” is the SQL query to be executed
* “$link\_identifier” is optional, it can be used to pass in the server connection link

**PHP mysqli\_num\_rows function**

The mysqli\_num\_rows function is used to get the number of rows returned from a select query.

It has the following syntax.

<?php

mysqli\_num\_rows($result);

?>

  HERE,

* “mysqli\_num\_rows(…)” is the row count function
* “$result” is the mysqli\_query result set

**PHP mysqli\_fetch\_array function**

The mysqli\_fetch\_array function is used fetch row arrays from a query result set.

It has the following syntax.

<?php

mysqli\_fetch\_array($result);

?>

  HERE,

* “mysqli\_fetch\_array(…)” is the function for fetching row arrays
* “$result” is the result returned by the mysqli\_query function.

**PHP mysqli\_close function**

The mysqli\_close function is used to close an open database connection.

It has the following syntax.

<?php

mysqli\_close($db\_handle);

?>

  HERE,

* “mysqli\_close(…)” is the PHP function
* “$link\_identifier” is optional, it is used to pass in the server connection resource

Let’s look at practical examples that take advantage of these functions.

Creating the MySQL database This tutorial assumes knowledge of MySQL and SQL, if these terms are unfamiliar to you, refer to our MySQL and SQL tutorials.

We will create a simple database called my\_personal\_contacts with one table only.

Below are the steps to create the database and table.

* Connect to MySQL using your favorite access tool such as MySQL workbench, phpMyAdmin etc.
* Create a database named my\_person\_contacts
* Execute the script shown below to create the table and insert some dummy data

<?php

CREATE TABLE IF NOT EXISTS `my\_contacts` (

  `id` int(11) NOT NULL AUTO\_INCREMENT,

  `full\_names` varchar(255) NOT NULL,

  `gender` varchar(6) NOT NULL,

  `contact\_no` varchar(75) NOT NULL,

  `email` varchar(255) NOT NULL,

  `city` varchar(255) NOT NULL,

  `country` varchar(255) NOT NULL,

  PRIMARY KEY (`id`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO\_INCREMENT=5 ;

INSERT INTO `my\_contacts` (`id`, `full\_names`, `gender`, `contact\_no`, `email`, `city`, `country`) VALUES

(1, 'Zeus', 'Male', '111', 'zeus @ olympus . mt . co', 'Agos', 'Greece'),

(2, 'Anthena', 'Female', '123', 'anthena @ olympus . mt . co', 'Athens', 'Greece'),

(3, 'Jupiter', 'Male', '783', 'jupiter @ planet . pt . co', 'Rome', 'Italy'),

(4, 'Venus', 'Female', '987', 'venus @ planet . pt . co', 'Mars', 'Italy');

?>

  We now have a database set up that we will manipulate from PHP.

Reading records from the database We will now create a program that prints the records from the database.

<?php

$dbh = mysqli\_connect('localhost', 'root', 'melody');

//connect to MySQL server if (!$dbh)

die("Unable to connect to MySQL: " . mysqli\_error());

//if connection failed output error message

if (!mysqli\_select\_db($dbh,'my\_personal\_contacts'))

die("Unable to select database: " . mysqli\_error());

//if selection fails output error message

$sql\_stmt = "SELECT \* FROM my\_contacts";

//SQL select query

$result = mysqli\_query($dbh,$sql\_stmt);

//execute SQL statement

if (!$result)

die("Database access failed: " . mysqli\_error());

//output error message if query execution failed

$rows = mysqli\_num\_rows($result);

// get number of rows returned

if ($rows) {

while ($row = mysqli\_fetch\_array($result)) {

echo 'ID: ' . $row['id'] . '<br>';

echo 'Full Names: ' . $row['full\_names'] . '<br>';

echo 'Gender: ' . $row['gender'] . '<br>';

echo 'Contact No: ' . $row['contact\_no'] . '<br>';

echo 'Email: ' . $row['email'] . '<br>';

echo 'City: ' . $row['city'] . '<br>';

echo 'Country: ' . $row['country'] . '<br><br>';

}

}

mysqli\_close($dbh); //close the database connection

?>

  Executing the above code returns the results shown in the diagram shown below

[](https://www.guru99.com/images/2013/04/greek_gods.png)

Inserting new records

Let’s now look at an example that adds a new record into our table. the code below shows the implementation.

<?php

$dbh = mysqli\_connect('localhost', 'root', 'melody');

//connect to MySQL server if (!$dbh)

die("Unable to connect to MySQL: " . mysqli\_error());

//if connection failed output error message

if (!mysqli\_select\_db($dbh,'my\_personal\_contacts'))

die("Unable to select database: " . mysql\_error());

//if selection fails output error message

$sql\_stmt = "INSERT INTO `my\_contacts` (`full\_names`,`gender`,`contact\_no`,`email`,`city`,`country`)";

$sql\_stmt .= " VALUES('Poseidon','Mail','541',' poseidon @ sea . oc ','Troy','Ithaca')";

$result = mysqli\_query($dbh,$sql\_stmt); //execute SQL statement

if (!$result)

die("Adding record failed: " . mysqli\_error());

//output error message if query execution failed echo "Poseidon has been successfully added to your contacts list";

mysqli\_close($dbh); //close the database connection

?>

  Executing the above code outputs “Poseidon has been successfully added to your contacts list” go back to the select query example and retrieval your contacts again.

See if Poseidon has been added to your list.

Updating records Let’s now look at an example that updates a record in the database.

Let’s suppose that Poseidon has changed his contact number and email address.

<?php

$dbh = mysqli\_connect('localhost', 'root', 'melody');

//connect to MySQL server

if (!$dbh)

die("Unable to connect to MySQL: " . mysqli\_error());

//if connection failed output error message

if (!mysqli\_select\_db($dbh,'my\_personal\_contacts'))

die("Unable to select database: " . mysql\_error());

//if selection fails output error message

$sql\_stmt = "UPDATE `my\_contacts` SET `contact\_no` = '785',`email` = ' poseidon @ ocean . oc ';

//SQL select query $sql\_stmt .= " WHERE `id` = 5";

$result = mysqli\_query($dbh,$sql\_stmt);

//execute SQL statement if (!$result)

die("Deleting record failed: " . mysqli\_error());

//output error message if query execution failed

echo "ID number 5 has been successfully updated";

mysqli\_close($dbh); //close the database connection

?>

**Deleting records**

Let’s now look at an example that removes records from the database. Let’s suppose that Venus has a restraining order against us, and we must remove her contacts info from our database.

<?php

$dbh = mysqli\_connect('localhost', 'root', 'melody');

//connect to MySQL server

if (!$dbh)

die("Unable to connect to MySQL: " . mysqli\_error());

//if connection failed output error message

if (!mysqli\_select\_db($dbh,'my\_personal\_contacts'))

die("Unable to select database: " . mysqli\_error());

//if selection failes output error message $id = 4;

//Venus's ID in the database

$sql\_stmt = "DELETE FROM `my\_contacts` WHERE `id` = $id";

//SQL Delete query

$result = mysqli\_query($dbh,$sql\_stmt);

//execute SQL statement

if (!$result)

die("Deleting record failed: " . mysqli\_error());

//output error message if query execution failed

echo "ID number $id has been successfully deleted";

mysqli\_close($dbh); //close the database connection

?>

**PHP Data Access Object PDO**

The PDO is a class that allows us to manipulate different database engines such as MySQL, PostGres, MS SQL Server etc.

The code below shows the database access method using the PDO object.

*Note: the code below assumes knowledge of SQL language, arrays, exception handling and foreach loop.*

<?php

try {

$pdo = new PDO("mysql:host=localhost;dbname=my\_personal\_contacts", 'root', 'melody');

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$pdo->exec('SET NAMES "utf8"');

$sql\_stmt = "SELECT \* FROM `my\_contacts`";

$result = $pdo->query($sql\_stmt);

$result->setFetchMode(PDO::FETCH\_ASSOC);

$data = array();

foreach ($result as $row) {

$data[] = $row;

}

print\_r($data);

}

catch (PDOException $e) {

echo $e->getMessage();

}

?>

  HERE,

* “try{…catch…}” is the exception handling block
* “$pdo = new PDO("mysql…” creates an instance of the PDO object and passes the database drivers, server and database names, user id and password.
* “$pdo->setAtt…” sets the PDO error mode and exception mode attributes
* “$pdo->exec('SET NA…” sets the encoding format

ODBC ODBC is the acronym for Open Database Connectivity. It has the following basic syntax.

<?php $conn = odbc\_connect($dsn, $user\_name, $password); ?>

  HERE,

* “odbc\_connect” is the PHP built in function
* “$dsn” is the ODBC data source name.
* “$user\_name” is optional, it is used for the ODBC user name
* “$password” is optional, it is used for the ODBC password

The example used assumes you;

* Are Using Windows OS
* You have created an ODBC link to the northwind Microsoft Access database named northwind

Below is the implementation code for ODBC data access

<?php

$dbh = odbc\_connect('northwind', '', '');

if (!$dbh) {

exit("Connection Failed: " . $dbh);

}

$sql\_stmt = "SELECT \* FROM customers";

$result = odbc\_exec($dbh, $sql\_stmt);

if (!$result) {

exit("Error access records");

}

while (odbc\_fetch\_row($result)) {

$company\_name = odbc\_result($result, "CompanyName");

$contact\_name = odbc\_result($result, "ContactName");

echo "<b>Company Name (Contact Person):</b> $company\_name ($contact\_name) <br>";

}

odbc\_close($dbh);

?>

  Summary

* MySQL is an open source relational database management available on most web hosting servers
* PHP has a rich collection of built in functions that simplify working with MySQL
* PDO is the acronym for PHP Data Object; it is used to connect to different database engines using the same object
* PHP uses the odbc\_connect function to manipulate databases via ODBC