Mar 01 2015
CheckoutCrypto Canada

Introduction to MySQL with PHP

By: Grant Hutchinson

Copyright 2015 CheckoutCrypto GNU Public License v3

Mar 01 2015
CheckoutCrypto Canada

Index

Reference Documentation:
http://dev.mysql.com/doc/refman/5.6/en/sql-syntax.html

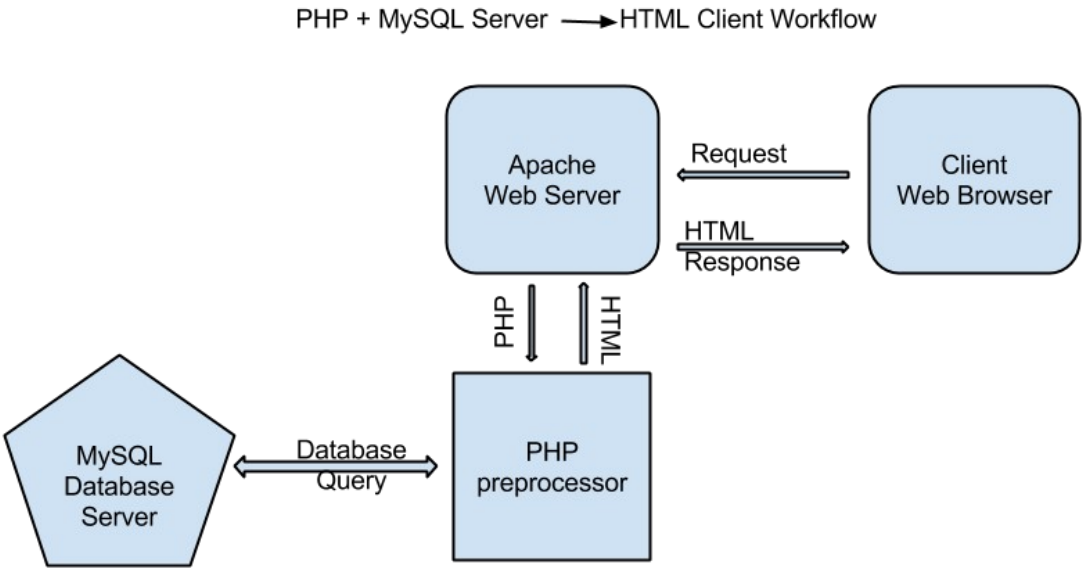
Copyright 2015 CheckoutCrypto GNU Public License v3

Mar 01 2015
CheckoutCrypto Canada

MySQL

MySQL is one of the most popular, open-source, database management system(DBMS). Uses the SQL acronym which stands for Structured Query Language. Used by a variety software and application systems e.g. LAMP Drupal, every major website.

Prerequisites: PHP knowledge, apache server with php5 installed and configured



PHPMYADMIN

```
sudo apt-get install phpmyadmin
```

enter your mysql root password when asked during install.
Visit http://127.0.0.1/phpmyadmin login with root mysql account
Begin by creating a new database, call it **testdb**

Copyright 2015 CheckoutCrypto GNU Public License v3

Connect

First we need to seperate our work into to **four** .php files on your apache server:

- configuration file(config.php), for all our MySQL account information.
- MySQL class file(dbconnect.php), for all connections, queries, to MySQL
- implementation file(sql_test.php),

This makes MySQL queries easier to: organize, test, build, update, maintain.

Config – config.php

```
<?php
 DEFINE('dbAddress','127.0.0.1');  // dblocation
 DEFINE('dbUser','');        // dbuser
 DEFINE('dbPass','');  // dbpass
 DEFINE('dbName',''); // dbname
?>
```

Prepared MySQL Library Object – dbconnect.php

```
<?php
class dbconnect
{

	function sqlConn(){
		include_once('config.php');
		$db = new PDO("mysql".":host=".dbAddress.";dbname=".dbName, dbUser, dbPass);
		return $db;
	}
	function addTables(){
		/// To be filled with query
	}
	function addRow(){
		/// To be filled with query
	}
	function displayRow($qry){
		/// To be filled  with query
	}
	function updateRow($qry){
		/// To be filled  with query
	}
	function removeRow($qry){
		/// To be filled  with query
	}
}

?>
```

Copyright 2015 CheckoutCrypto GNU Public License v3

Implementation – sql_test.php

```
<?php
require('dbconnect.php');

$db = new dbconnect();
/// $db->addRow();  etc
/// we'll add our test function calls  here
?>
```

Explanation

```
$db = new PDO(driver.":host=".dbAddress.";dbname=".dbName.", dbUser, dbPass);
```

This is our prepared connection to MySQL through our config's global variables.

```
$db  = $this->sqlConn();
```

We return this new database object to each function that needs it to perform MySql functions such as queries.

```
$stmt = $db->prepare("INSERT INTO some_table (somefield, somevar2, something3)");
```

prepare takes a query, complete with field variables, that can be binded following it's invocation.

```
$stmt->bindValue('somefield', 'anything', PDO:PARAM_STR);
$stmt->bindValue('somevar2', 'somevalue', PDO:PARAM_STR);
$stmt->bindValue('something', 'whatever', PDO:PARAM_STR);
```

Values are binded to variables within the prepare() query. The syntax bindValue($field, $value, $datatype);

```
$stmt->execute();
```

Perform the query.

A deprecated method(not recommended):

```
mysql_connect(dbAddress, dbUser, dbPass) or die(mysql_error());
```

This opens the connection to mysql, to any given server IP, using a user and password. If no connection is made, it will throw mysql_error() - prints the error to the window

```
mysql_select_db(dbName) or die("couldn't select db: " . mysql_error());
```

This sets the selected database on the mysql server we connect to.

```
mysql_query($qry)
```

This querys the mysql connection, at the selected database, with the selected query.

```
mysql_close();
```

This closes the mysql connection.

Create Tables

syntax: CREATE table sometable (somefieldname datatype(fieldsize));
explanation: some datatypes include:
Strings
varchar, char, binary, blob, text, enum, set
Integers
integer, int, smallint, tinyint, mediumint, bigint, decimal, numeric, float, double,

Copyright 2015 CheckoutCrypto GNU Public License v3

Timestamps
date, time, datetime, timestamp, year,
reminder: always open and close your sql connections

A better example:

```
$someqry = “CREATE table Contacts  (id INT NULL AUTO_INCREMENT, PRIMARY KEY(id), UserID
varchar(50),ContactName varchar(100), ContactEmail varchar(100), ContactPhone varchar (20))";
```

To add an **auto_increment** field, we add **id INT NULL AUTO_INCREMENT.**
To set the primary key we use **primary key**(fieldname);

To apply this to our working MySQL model above, open the dbconnect.php library you created earler, add the following within the function addTables(){ }

```
$db  = $this->sqlConn();

$stmt = $db->prepare("CREATE table Contacts  (id INT NULL AUTO_INCREMENT, PRIMARY KEY(id), UserID
varchar(50),ContactName varchar(100), ContactEmail varchar(100), ContactPhone varchar (20))");

$stmt->execute();
```

Finally, amend your implementation file(sql_test.php) to look like this:

```
<?php
require('dbconnect.php');

$db = new dbconnect();
$db->addTables();
?>
```

Try it

Visit http://127.0.0.1/yoursite/sql_test.php then check the tables were added by visiting your phpmyadmin mysql administration page(assuming you installed it, as per instructions) at http://127.0.0.1/phpmyadmin open the database you created called **testdb.** Look at the left hand side see if you see any table listed named “Contacts” .

Congratulations, you just created your first MySQL database table.

Now, it's time to fill it with some actual data.

Copyright 2015 CheckoutCrypto GNU Public License v3

Insert into table

syntax: INSERT into SomeTable values ('col1value', 'col2value', 'col3value')
explanation: an insert query must contain values for all the fields in a table.
reminder: always open and close your sql connections

To place data in your newly created tables, change your **dbconnect.php** in the function addRow(){ } to reflect the following:

```
function addRow($userid, $contactName, $contactPhone, $contactEmail){
		$db  = $this->sqlConn();
		$stmt = $db->prepare("INSERT into Contacts (userid, name, email, phone)");
		$stmt->bindValue(':userid', $userid, PDO::PARAM_INT);
		$stmt->bindValue(':name', $contactName, PDO::PARAM_STR);
		$stmt->bindValue(':email', $contactEmail, PDO::PARAM_STR);
		$stmt->bindValue(':phone', $contactPhone, PDO::PARAM_STR);
		$stmt->execute();
}
```

Finally, create a new **sql_test_add.php** or simply modify your **sql_test.php** to look like the following:

```
<?php
require('dbconnect.php');

$db = new dbconnect();
$db->addRow(1, 'Joe', '432 322 2234', 'joe@mailinator.com');
?>
```

Try it!

Visit http://127.0.0.1/site/sql_test_add.php to add a new row to your database table called “Contacts”.

Check that a new row was added by logging into your phpmyadmin at http://127.0.0.1/phpmyadmin then select the “Contacts” table from the right hand side.

Select from table

syntax: “SELECT somefield, somethingelse, someotherfield from SomeTable where field=:var”
explanation: an insert query must contain values for all the fields in a table.
reminder: always open and close your sql connections

To place data in your newly created tables, change your **dbconnect.php** in the function displayRow() { } to reflect the following:

Copyright 2015 CheckoutCrypto GNU Public License v3

```
function displayRow($username){
try {
		$contact = array();
		$db  = $this->sqlConn();
		$stmt = $db->prepare("SELECT ContactName, ContactEmail, ContactPhone FROM Contacts WHERE
ContactName=:username");
		$stmt->bindValue(':username', $username, PDO::PARAM_STR);
		$stmt->execute();

		$rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
		if(is_array($rows) AND count($rows) == 1) {
			$row = $rows[0];
			if(isset($row['ContactName'])){
				$contact['name'] = $row['ContactName'];
				$contact['email'] = $row['ContactEmail'];
				$contact['phone'] = $row['ContactPhone'];
			}
		}
} catch (exception $e) {
	echo $e;
}
		return $contact;
}
```

Finally, create a new **sql_test_display.php** or simply modify your **sql_test.php** to look like the following:

```
<?php
require('dbconnect.php');

$db = new dbconnect();
$user = $db->displayRow('joe');
var_dump($user);
?>
```

Try it!

Visit http://127.0.0.1/yoursite/sql_test_display.php to display the row you previously inserted into the database table.

Update from table

syntax: "UPDATE SomeTable SET somefield='somevalue', someotherfield='somevalue' "
explanation: an insert query must contain values for all the fields in a table.
reminder: always open and close your sql connections

Copyright 2015 CheckoutCrypto GNU Public License v3

To place data in your newly created tables, change your **dbconnect.php** in the function displayRow() { } to reflect the following:

```
function updateRow($userid, $contactName, $contactPhone, $contactEmail){
	$db  = $this->sqlConn();

	$stmt = $db->prepare("UPDATE Contacts set ContactName = :name, ContactEmail = :email, ContactPhone = 
:phone WHERE UserID = :userid");
	$stmt->bindValue(':userid', $userid , PDO::PARAM_INT );
	$stmt->bindValue(':name', $contactName ,  PDO::PARAM_STR);
	$stmt->bindValue(':phone', $contactPhone ,  PDO::PARAM_STR);
	$stmt->bindValue(':email', $contactEmail ,  PDO::PARAM_STR);
	$stmt->execute();
}
```

Finally, create a new **sql_test_update.php** or simply modify your **sql_test.php** to look like the following:

```
<?php
require('dbconnect.php');

$db = new dbconnect();
$db->updateRow(1, 'Joe', '422 342 2614', 'someguy@mailinator.com');
?>
```

Try it!

Visit http://127.0.0.1/yoursite/sql_test_update.php to display the row you previously inserted into the database table.

Check that the current row was update by logging into your phpmyadmin at http://127.0.0.1/phpmyadmin then select the “Contacts” table from the right hand side.

Or

Amend your implementation file(**sql_test_update.php** or **sql_test.php**) with the following:

```
<?php
require('dbconnect.php');

$db = new dbconnect();
$user = $db->displayRow('joe');
var_dump($user);
?>
```

Copyright 2015 CheckoutCrypto GNU Public License v3

Mar 01 2015
CheckoutCrypto Canada

Delete from table

syntax: "DELETE from SomeTable where somefield = :somevalue"
explanation: an insert query must contain values for all the fields in a table.
reminder: always open and close your sql connections

To place data in your newly created tables, add the following to your **dbconnect.php** in the function removeRow(){ }

```
function removeRow($username){
        $db  = $this->sqlConn();
        $stmt = $db->prepare("DELETE FROM Contacts WHERE ContactName =:name");
        $stmt->execute(array(':name' => $username));
}
```

Finally, create a new **sql_test_del.php** or simply modify your **sql_test.php** to look like the following:

```
<?php
require('dbconnect.php');

$db = new dbconnect();
$db->removeRow('Joe');
?>
```

Try it!

Visit http://127.0.0.1/yoursite/sql_test_del.php to delete a row from your database table with a ContactName of Joe

Check that a row was removed by logging into your phpmyadmin at http://127.0.0.1/phpmyadmin then select the "Contacts" table from the right hand side.

Copyright 2015 CheckoutCrypto GNU Public License v3