

PHP DB's MySQL and SQL 2 - Written by Anthony Adams

Last week we learned how to pull data from the server into a webpage. We also need to be able to push data to the database. In the case where the user fills out a form the form needs to be able to do something with the data and one way to approach this is to save that data on the DB.

The process is somewhat similar to what we did last time albeit backwards and with some different steps.

The DB Table

Make sure you have a database table made for your data. If not already created, make the table for the form.

Server: sql306.infinityfree.com » Database: epiz_34180651_schoolDB

Structure SQL Search Query Export Import Operations Routines Designer

Filters

Containing the word:

Table	Action	Rows	Type	Collation
<input type="checkbox"/> Personnel	★ Browse Structure Search Insert Empty Drop	2	MyISAM	latin1_swedish_ci
1 table	Sum	2	MyISAM	latin1_swedish_ci

Check all With selected:

Print Data dictionary

Create table

Name: Number of columns:

Server: sql306.infinityfree.com » Database: epiz_34180651_schoolDB » Table: week6

Browse Structure SQL Search Insert Export Import Operations

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A..!
first_name	VARCHAR	30	None			<input type="checkbox"/>	---	<input type="checkbox"/>
last_name	VARCHAR	30	None			<input type="checkbox"/>	---	<input type="checkbox"/>
email	VARCHAR	40	None			<input type="checkbox"/>	---	<input type="checkbox"/>
phone_number	VARCHAR	40	None			<input type="checkbox"/>	---	<input type="checkbox"/>
message	TINYTEXT	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>
ID	INT		None			<input type="checkbox"/>	PRIMARY	PRIMARY <input checked="" type="checkbox"/>

Table comments: Collation: Storage Engine:

PARTITION definition: (Expression or column list)

Partitions:

Here we enter the columns and the types of data we want to collect. Note that for the message we want to collect text.

TINYTEXT has a limit of 255 bytes/characters.

TEXT has a limit of 65,535 bytes/characters.
MEDIUMTEXT has a limit of bytes/16,777,215 characters.
LONGTEXT has a limit of bytes/4,294,967,295 characters.

We can preview the SQL query:

```
CREATE TABLE `epiz_34180651_schoolDB`.`week6` ( `first_name` VARCHAR(30) NOT NULL , `last_name` VARCHAR(30) NOT NULL , `email` VARCHAR(40) NOT NULL , `phone_number` VARCHAR(40) NOT NULL , `message` TINYTEXT NOT NULL , `ID` INT NOT NULL AUTO_INCREMENT , PRIMARY KEY (`ID`)) ENGINE = MyISAM;
```

And save it:

Server: sql306.infinityfree.com » Database: epiz_34180651_schoolDB » Table: week6

Browse Structure SQL Search Insert Export Import Operations

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	first_name	varchar(30)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	2	last_name	varchar(30)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	3	email	varchar(40)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	4	phone_number	varchar(40)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	5	message	tinytext	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	6	ID	int(11)		No	None		AUTO_INCREMENT	Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext Fulltext

Print Propose table structure Move columns Normalize

Add column(s)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	ID	0	A	No	

Create an index on columns

Now our DB is set up and ready to take the data we send it from the form.

The Form

First we must have a form

```
<form action="scripts/handler.php"
method="POST">

    <fieldset>
        <legend>Personal Info</legend>

        <label for="fname">First Name</
label>
```

```
        <br>
        <input type="text" id="fname"
name="f_name">
        <br>
        <label for="lname">Last Name</
label>
        <br>
        <input type="text" id="lname"
name="l_name">
        <br>
        <label for="email">Email</
label>
        <br>
        <input type="email" id="email"
name="e_mail">
        <br>
        <label for="phonenumber">Phone
Number</label>
        <br>
        <input type="tel" id="phonenumber"
name="phone_num">
        <br>
        <label for="msg"></label>
```

```
        <textarea
name="message_content" id="msg" cols="30"
rows="10"></textarea>

        <br>

        <input type="submit"
value="Send Data">

    </fieldset>

</form>
```

Note that in the form, every form field has a “name” attribute. That will be the attribute we use to collect the data using PHP.

The PHP

Once that is done, we write our variables and test them within the same document if you wanted to:

```
<?php

    //Note that every input has a name
attribute
    //we will use the name attribute to
```

```
pass the data to the DB
    //We will accomplish this by
collecting the form data using the name
attribute and passing it into a variable
named accordingly

    $first_name = $_POST['f_name'];
    $last_name = $_POST['l_name'];
    $email = $_POST['e_mail'];
    $phone_number =
$_POST['phone_num'];
    $message =
$_POST['message_content'];

    //we could echo them out to see if
it works

    echo $first_name . '<br>' .
$last_name . '<br>' . $email . '<br>' .
$phone_number . '<br>' . $message;

?>
```

With this set up, once we press the save button we should see:

Personal Info

First Name

Last Name

Email

Phone Number

Anthony
Maximus
email@mail.com
904-555-5555
This is a test of the message thing

The variables have returned us the data we entered into the inputs.

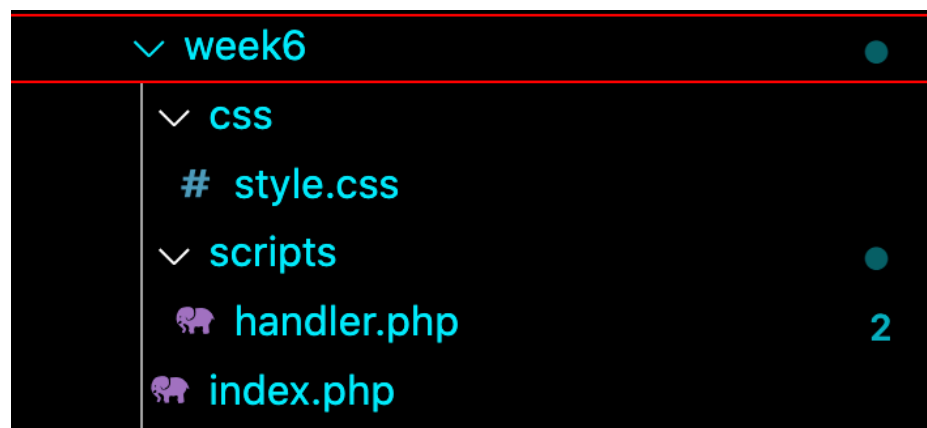
At this point we could move the PHP to an

external PHP file, that way we can run the form in an HTML file, or a PHP file as HTML. However you want to do it really does not matter because:

```
<form action="scripts/handler.php"
method="POST">
```

The opening tag has the file name of where the PHP code is to deal with the form data.

So we can make a file structure like this:



Now we can finish the PHP:

```
<?php
//Handler for the form data
```

```
//Collect the form data and pass the data into
variables
//$variable    <-    [collected data]
$first_name = $_POST['f_name'];
$last_name = $_POST['l_name'];
$email = $_POST['e_mail'];
$phone_number = $_POST['phone_num'];
$message = $_POST['message_content'];

//store the connection cred in variables
(optional)
$hostname = 'sql306.infinityfree.com';
$username = 'epiz_34180651';
$pass = 'Z30WgZ6WBzf';
$dbname = 'epiz_34180651_schoolDB';

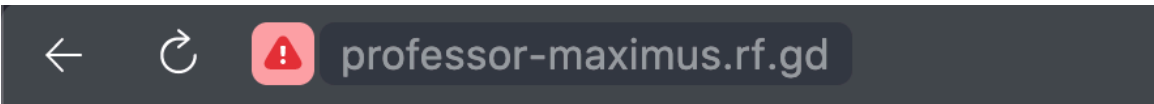
//connect to the DB *Note that this could be
done without the variables instead you would just
put the values above right in the connection
function.
$con = mysqli_connect(
    $hostname,
    $username,
    $pass,
    $dbname
);

//if the connection fails stop it all and throw
a code
if($con->connect_error){
    die('Connection Failed');
}else{
    //make a variable and load the SQL
connection and SQL query into it we are preparing
```

```
the data collected to be sent to the database here
    $stmt = $con->prepare("insert into
week6(first_name, last_name, email, phone_number,
message) values(?, ?, ?, ?, ?)");
    //below we bind variables to the table
columns that the corresponding data will be entered
into //the the "sssss" below is data type for
string - each var is a string here. if one was an
int use i or whatever.
    $stmt -> bind_param("sssss", $first_name,
$last_name, $email, $phone_number, $message);
    //execute the query
    $stmt -> execute();
    //return msg
    echo 'Form sent!';
    //close all connections
    $stmt -> close();
    $con -> close();
}

//Note $stmt just means "statement" nothing
more nothing less its just a regular var
//Note we do not use a closing PHP tag at the
end. Because this file only contains PHP so we
don't need to close it.
```

Then the form is sent the message will appear



Form sent!

[Back](#)

You could add to this a back button, just remember to relatively direct the link. Seeing as this php file is stored in a subfolder the href link will have to be ../index.php to get back to the index page.

Show all | Number of rows: 25 | Filter rows:

Options

	ID	first_name	last_name	email	phone_number	message
<input type="checkbox"/> Edit Copy Delete	5	Anthony	Maximus	email@mail.com	222-444-9999	Some form message blah blah blah blah blah blahbla...

↑ Check all With selected: Edit Copy Delete Export

Check DB for entries

*note if you build this and it does not work right away, don't get discouraged. 1 typo can make

the whole thing not work. This happens all the time.

Operater used:

-> - arrow operator is used to point to a property of an object

Statements used:

Prepare

<https://www.php.net/manual/en/https://php.net/manual/en/mysqli.prepare.php>

bind_param

<https://php.net/manual/en/mysqli-stmt.bind-param.php>

execute

<https://php.net/manual/en/mysqli-stmt.execute.php>

close

<https://php.net/manual/en/mysqli-stmt.close.php>