



CALLING FUNCTIONS WITH PHP

The simplest way to use stored routines with PHP is to use functions. In the stored procedure tutorials we discussed the problems associated with functions in MySQL with regard to using select statements. Once this restriction is lifted functions will become much more useful but we can still use them in their current form. If you have previous experience of developing PHP then using functions will not be too dissimilar to methods you may have already used.

In this section we will be creating functions and procedures against the tables contained in the setup.myp script. If you have previously downloaded this script it may be worth downloading this again as there have been some minor amendments to the tables.

```
drop database if exists pers
\g
create database pers
\g
use pers
\g
create table emps(emp_id int NOT NULL,
                  emp_name varchar(30),
                  dept_id int,
                  dept_name varchar(30),
                  salary decimal(5,2),
                  primary key(emp_id))
\g
insert into emps (emp_id,emp_name,dept_id,salary)
values (1,'Roger',1,2000.00),(2,'John',2,2500.00),(3,'Alan',1,2100.00)
\g
select * from emps
\g
create table dept (dept_id int NOT NULL,
                  description varchar(30),
                  primary key(dept_id))
\g
insert into dept (dept_id,description)
values (1,'Information Technology'),(2,'Sales')
\g
select * from dept
\g
```

 [setup.myp](#)

Lets start by creating a simple function that we can use on our PHP pages. We won't make this too complicated so that if we have problems we can be sure that the function is not the problem.

-- MySQL --

```
create function helloworld() returns varchar(20)
return "Hello World";
//
Query OK, 0 rows affected (0.00 sec)
```

 [helloworld1.myp](#)

Lets just check that the function works at the MySQL command line before we start using it with PHP.

-- MySQL --

```
select helloworld() //
+-----+
| helloworld() |
+-----+
| Hello World |
+-----+
1 row in set (0.02 sec)
```

OK so now lets look at how we call the function from PHP. We will assume some knowledge of PHP so apologies if we skip over sections of PHP code and focus on the important parts as far as MySQL is concerned. To start lets build a small section of code to connect to the MySQL server from PHP. To save space and to make sure errors are kept at a minimum we will be writing straight PHP we won't for the moment include HTML.

-- PHP --

```
<?php

$link = mysqli_connect("localhost","root","xxxxxxx");

if (mysqli_connect_errno()) {
echo "error";
exit();
} else { echo "no error"; }
?>
```

 [php_connection.phx](#)

Note : the source file has a .phx extension. This is so that the source can be viewed in the browser. If you wish to use the source files save them as PHP so your webserver interprets the PHP correctly.

This small PHP program simply tests that you have successfully setup MySQL, PHP and your webserver. Make sure you set the correct values for the server name, user and password in the following line.

```
$link = mysqli_connect("localhost","root","xxxxxxx");
```

If you load the page in your browser you will see **no error** if PHP successfully connected to your MySQL installation. If you see **error** then this means that there is a problem either with your **mysqli_connect** statement or that your MySQL instance is not correctly started or installed.

If you are having problems check the following

- * Your PHP installation is working
- * You can connect to MySQL via the command line
- * Your Webserver is operational
- * The details in the mysqli_connect string are correct

It's worth making sure you can get the **no error** message to display before continuing so that it's easier to eliminate problems in future PHP pages.

Once we have been able to connect to MySQL from PHP we can start to call our functions. The first thing to do is select the database we want to use, in the case of our example this is **PERS** the PHP commands

to do this is as follows.

```
-- PHP --
```

```
mysqli_select_db ($link,"pers");
```

This code is the equivalent of typing **use pers;** at the MySQL command line. We simply pass in **\$link** which is the connection we have already established with the **mysqli_connect** command. We then pass in the name of the database we wish to use, in this case **pers**. Next we want to call the function, this is fairly simple and similar to selecting rows from a table in PHP. This is done using the **mysqli_query** command like so.

```
-- PHP --
```

```
$result = mysqli_query($link,"select helloworld()")
```

Again we pass in the **\$link** object to tell PHP to use the connection we have already established. We then pass in the select statement we want to run, as we are calling a function and not a select against a table this is simply **select** plus the name of the function, in this case **helloworld()**. The result of the call is placed into a variable called **\$result**, this will be used to display the results of the select.

At this stage we have a number of options to retrieve the data returned we can use one of three methods, we can use **MYSQLI_NUM** to reference the results by number, **MYSQLI_ASSOC** to reference the results by name and finally **MYSQLI_BOTH** which will allow to reference them by both name and number. Which you use is up to you, we will be using **MYSQLI_NUM** in this example. We extract the results using the **mysqli_fetch_array** command like so.

```
-- PHP --
```

```
$row = mysqli_fetch_array($result, MYSQLI_NUM);
```

The array of results is fetched into a variable called **\$row**. This time we don't need to pass in the connection as we have the results stored in a variable called **\$result**, this is passed into **mysqli_fetch_array** along with the **MYSQLI_NUM** option. We can now use **\$row** to display the value returned from the function, as we used **MYSQLI_NUM** we simply use the number to return the value like so.

```
-- PHP --
```

```
echo $row[0];
```

This will simply display the value at position 0 in the array. Position 0 is the first column returned by the select statement, as our call to the function was a simple select position 0 will hold the return value of our function.

Lets add all of these elements together to create a page which displays the result of our function. One extra section of code we will add is an if around the call to **mysqli_query** so that we can trap an error if the function does not exist or fails to fire.

```
-- PHP --
```

```
<?php

$link = mysqli_connect("localhost","root","xxxxxx");

if (mysqli_connect_errno()) {
    echo "error";
    exit();
}

mysqli_select_db ($link,"pers");

if ($result = mysqli_query($link,"select helloworld())) {

    $row = mysqli_fetch_array($result, MYSQLI_NUM);

    echo $row[0];

} else { echo "problem :( "; }

?>
```

[php_helloworld.php](#)

Now load the PHP page in your browser, if all goes to plan you should see the words "Hello World" displayed. Not particularly visually stunning but it means we have been able to call a MySQL function via PHP. If you don't see the "Hello World" run through the checks to verify that you web server, PHP and MySQL Database are configured correctly.

[< Introduction Calling Procedures From PHP >](#)



Last Update : 10/05/2005 22:24 GMT