

PHP and MySql - Seminar

By Petros Bournias
February 2010

Getting Started with MySQL

- An Introduction to Databases
- Logging On to MySQL
- So what's SQL?
- Creating a Database
- Creating a Table
- Inserting Data into a Table
- Viewing Stored Data
- Modifying Stored Data
- Deleting Stored Data

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Sun Microsystems, Inc.

<http://www.mysql.com/>

<http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html>

The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”).

MySQL is a relational database management system and is open source.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python).

MySQL versions vary from 3.23 to the latest versions of 5.0, 5.1, 5.2, 5.3, and their subsets. This is a very important point because it may affect your programming as well especially between major version changes.

For the majority of developers, the use and understanding of MySql will give you a better understanding of how to use PhpMyAdmin and MySql on your web servers on the Internet, as well as a better understanding of the MySql commands, and access privileges.

The MySQL site also offers test data that you can download to load into a database to practice your programming skills.

What is a database?

A **database** is a number of **tables** that are defined as **structures** used to hold data in **rows** and columns. A comparison might be an excel spreadsheet.

The following information is for those of you who wish to learn more about MySql but the main focus of programming in Php using MySql commands is the purpose of these seminars.

The organization or topology of the MySQL database management system

The MySQL database is therefore a system that helps us manage the databases that hold the tables that we design to hold our data.

Using a web server, we can create one or more databases according to whatever our hosting provider has given us based on the hosting server that we rent.

Users

Within the database that we create on our web server, we must also assign a user and therefore a user name and password, and the rights or permissions that are granted to that user.

We can have more user names that can access the database with different permissions just as we have access privileges in Elxis but the access privileges of Elxis have nothing in common with the access privileges of MySQL.

Privileges

Privileges can be assigned to databases, tables, and users.

<http://dev.mysql.com/doc/refman/5.1/en/dynindex-priv.html>

Where is our database?

MySQL is installed and has a database of its own. After that you create separate databases and tables within those databases.

Databases are created through your web server cpanel or can even be done through Php programming.

For the purposes of this seminar, Php will be used for database access.

Methods to access a MySQL database:

The MySQL console

The console is available through your web server package on your local pc or on your local Linux server otherwise you can use a tool provided by MySQL to access your database system remotely.

MySQL Console Commands

<http://dev.mysql.com/doc/refman/5.1/en/privileges-provided.html>

PhpMyAdmin

This is an easy way to view your database data whether it is locally on your pc or through cpanel or plesk on your Internet web server.

Php programming

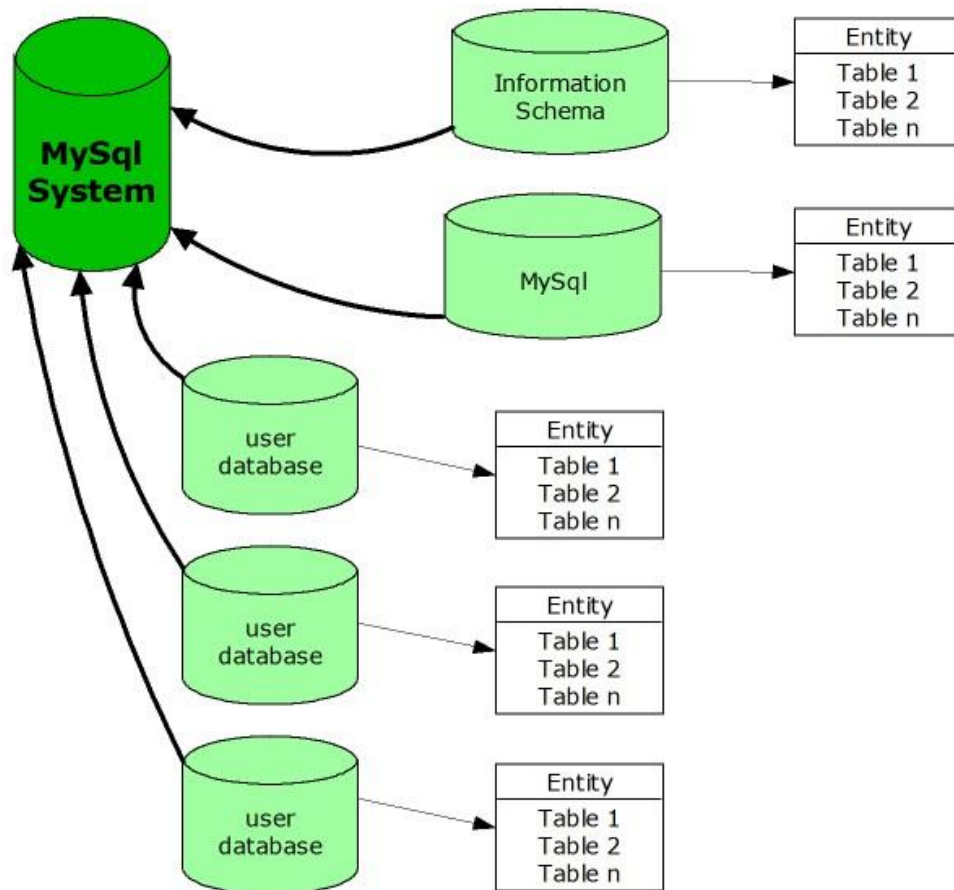
By using the connection handling of Php you can connect to your database to access your data on your local or Internet web server.

Database and table naming

Use Latin characters and use an underscore or underline. No spaces in between.

Be aware that not all characters can be used for proper naming.

MySQL Database System



How do we access a database?

You must login to your database using the database name, a user name, and a user password.

Logging On to MySQL

The following example is the set of commands to link to a database using the server name, user name, and password. The link is in a function to allow it to be used inside Php.

The database name is test. The server name is localhost which is common for most servers today. \$link will give us a success or failure result for testing that our link works.

A simple function to connect to the database:

```
function test_db_connect() {  
    // to connect to the live server on the net  
    $link = mysql_connect ("localhost","user","password");  
    if ($link && mysql_select_db ("test "))  
        return ($link);  
    return (false);  
}
```

The above SQL can also be used without it being a function but in case you wish to open and close a database many times it is easier to use as a function.

So what's SQL?

SQL is basically a set of commands that create a result in an array.

In this example we are selecting 4 fields a table where the field is equal to the variables that we have set such as in a search.

\$result is the resource variable that we named to tell us if the query was a success or failure.

The select in bold red is the command that we need to tell MySql to retrieve data.

The MySql command is shown in orange.

The actual SQL is everything within the double quotes, shown in blue.

Exit() is a Php command, it can have an integer or a string for a message.

```
$result = mysql_query("select onoma, epitheto, email, hora from table_name  
WHERE (onoma = '$onoma' and epitheto = '$epitheto' and email = '$email'  
and hora = '$hora' )") or exit ("Δεν είναι δυνατή η σύνδεση με τον διακομιστή,  
παρακαλώ ξανά δοκιμάστε σε άλλη στιγμή.  
<br /> Cannot connect to server, Please try again at another time.<br /> \n");
```

```
// row is the array that holds the data that meets the above request.
```

```
$row = mysql_num_rows($result);
```

SQL can be broken into multiple lines or written on one long continuous line and ends with a semi-colon.

You will see how we use \$row to extract data below.

In accessing MySql database data, we can use the notation table.field to be able to access multiple tables that may also have similar field names.

e.g. table1.field1, table2.field1, table3.field1, table4.field3

Creating a Database

A database is generally created by us or in the case of Elxis by the program during installation. In the case of Elxis, it is done via Php. If we are doing it manually, we can use Cpanel on the Internet or PhpMyAdmin for our local server.

For the purposes of foreign characters, it is best if you create your databases and tables with the character set to utf8_general_ci with the same for collation.

Use Latin characters to name your database. The underline is to separate between the logon name and the database as shown in this example.

e.g. your web logon is **dante** and you want to name your database **stars**.

e.g. your database will most likely be shown as **dante_stars**

You may be restricted to a certain number of databases that you create on your server on the Internet. That means that you must use them wisely.

Creating a Table

Tables are generally created by us or in the case of Elxis by the program during installation. In the case of Elxis, it is done via Php.

Use Latin characters to name your tables. The underline is to separate between the logon name and the database as shown in this example.

e.g. your web logon is **dante** and you want to name your database **stars**.

e.g. your database will most likely be shown as dante_stars

```
CREATE TABLE IF NOT EXISTS `enolib` (  
  `id` int(8) unsigned NOT NULL AUTO_INCREMENT,  
  `dateadded` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
  `leader` varchar(24) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `recdate` varchar(8) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `title` varchar(1000) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `published` varchar(200) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `subject` varchar(200) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `author` varchar(200) COLLATE utf8_unicode_ci DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
```

The most common way to enter data into a table is through a form. In the case of Elxis, the entire program can be theorized as a form because we enter data to create titles, headings, content in content items to generate web pages. All of the data that we enter creates our web site and is recorded into the database that we created when the program was installed on our server.

```
$query = "insert into $table_name (onoma, epitheto, hora, email, msgtxt)
values ('$onoma', '$epitheto', '$hora', '$email', '$msgtxt')";
```

```
INSERT INTO `table_name` (`bid`, `onoma`, `epitheto`, `dieythinsi`, `poli`, `tk`, `hora`, `email`, `msgtxt`,
`dateadded`) VALUES
(1, 'Αντώνιος', 'Μπουρνιάς', ',', ',', ',', 'Ελλάδα', 'pab@linux.cxm', 'Κανένα σχόλιο.', '2010-01-22 12:07:10'),
(2, 'Νικόλαος', 'Μπαρμπάτση', ',', ',', ',', 'Ελλάδα', 'bdhs@linux.cxm', 'Κανένα σχόλιο.', '2010-01-22 15:12:26');
```

Extracting data from our array and displaying it is simply a matter of taking the \$row array and using either specific field names or column names that you assigned to have it display the data from the array.

```
while ( $row = mysql_fetch_array($result, MYSQL_ASSOC) ) {
    $bgcolor = ($i++ & 1) ? '#FFFFFF' : '#FFFFFFF';
    print("<tr bgcolor='$bgcolor">");
    $seq_cnt++;
    print ("<td>$seq_cnt</td>\n");
    printf("[%010s]\n", $s); // zero-padding works on strings too
    printf("<td class='td'>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</td>\n", $row["bid"]);
    printf("<td class='td'>%s</td>\n", $row["onoma"]);
    printf("<td class='td'>%s</td>\n", $row["epitheto"]);
    printf("<td class='td'>%s</td>\n", $row["hora"]);
    echo "</tr>";
}
```

Modifying, Changing, and Updating Stored Data Records

To err is human, and so we usually create a way for users to make changes and update data. There are two parts to doing this and I have broken it into 2 pieces but normally it can be done all together. The first is to clean the posted variables and the second is to update the row columns of the table.

```
// this is one way of cleaning the post variables
$onoma      = mysql_real_escape_string($_POST["onoma"]);
$epitheto   = mysql_real_escape_string($_POST["epitheto"]);
$hora       = mysql_real_escape_string($_POST["hora"]);
$msgtxt     = mysql_real_escape_string($_POST["msgtxt"]);
```

```
// this updates the table fields
$update = "update $table_name set
          onoma      = '$onoma',
          epitheto   = '$epitheto',
          hora       = '$hora',
          msgtxt     = '$msgtxt'
WHERE bid = '$bid' ";
```

```
// this creates a variable that tells us if our result was a success or failure
$result2 = mysql_query($update) or die(mysql_error());
```

```
// the commit is the equivalent to making sure that the data is written to the drive.
mysql_query("COMMIT");
```

Deleting Stored Data Records

If to err is human, then we must also have a way of **deleting records**.

Be extremely careful when specifying what you are deleting and where or in other words which database, table, and records.

The following is an example using two tables, a where clause, with a combined condition. Is just the plain sql that can be used directly in PhpMyAdmin as can other sql.

```
DELETE t1 FROM tbl_name t1, tbl_name t2 WHERE t1.userID=t2.userID AND
t1.eventID=t2.eventID AND t1.ueventID < t2.ueventID
```

To use the delete statement in Php, simply wrap it as a string just the way that the update was done. To make it easier to read, put some spaces in before and after the = signs or even break into more lines like this:

```
DELETE t1 FROM tbl_name t1, tbl_name t2
WHERE t1.userID = t2.userID
AND t1.eventID = t2.eventID
AND t1.ueventID < t2.ueventID
```

You can use parenthesis to make your logic clearer.

Additional Resources

Tools

Be aware that some tools can be used both locally and remotely. Many of these basically do the same thing, they manipulate the database.

MySQL Workbench

<http://wb.mysql.com/>

<http://www.smashingmagazine.com/2009/03/25/mysql-admin-and-development-tools-round-up/>

MySQL Web Seminars

<http://www.mysql.com/news-and-events/web-seminars/>