



## PHP/MySQL Tutorial: How to Use PHP and MySQL?

PHP is the most popular scripting language for web development. It is free, open source and server-side (the code is executed on the server). MySQL is a Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). It is also free and open source. The combination of PHP and MySQL gives unmet options to create just about any kind of website - from small contact form to large corporate portal.

This PHP/MySQL tutorial is part of the rich collection of [web hosting tutorials](#) which SiteGround experts have prepared for you. Find out how to start your website, how to promote it, how to deal with cPanel, webmail and FTP.

### PHP/MySQL Tutorial includes:

- A short introduction on [what is PHP](#);
- [What is MySQL](#) and what are its benefits;
- [Creating MySQL database and user](#) - useful information for installing a PHP/MySQL script on your website;
- [Removing MySQL database and user](#);
- Learn how to manage MySQL databases with [phpMyAdmin](#) - [MySQL Backup/Restore](#) functionality and how to use [MySQL Access Host](#);
- [Creating a database table](#);
- [Connecting to the database](#);
- [Querying the database](#);
- [Displaying table data](#) in HTML;
- [Selecting individual records](#);
- How to use the [MySQL GUI tools](#) to [backup](#), [restore](#) and [execute queries](#) on your database;
- Information about [Zend Optimizer](#);
- Learn how to install [Pear modules](#).

### PHP Tutorial What is PHP?

PHP was at first created as a simple scripting platform called "Personal Home Page". Nowadays PHP (the short for Hypertext Preprocessor) is an alternative of Microsoft's Active Server Pages (ASP) technology.

PHP is an open source server-side language which is used for creating dynamic web pages. It can be embedded into HTML. PHP is usually used in conjunction with a [MySQL database](#) on Linux/UNIX web servers. It is probably the most popular scripting language.

PHP is a widely-used general-purpose scripting language and interpreter that is freely available. A full explanation of all the PHP tags, complete user manual and lots of tutorials can be found on the [PHP's official page](#).

[Previous](#)[Next](#)

One of the most important things about using PHP is to have a PHP specialized host. Here are some of the things we at SiteGround are proud of:

- We have long experience in providing technical support for PHP-based web sites. Thanks to it our servers are perfectly optimized to offer the best overall performance for most PHP applications.
- We offer a lot of FREE PHP tools including CMS systems, forums, galleries, blogs, shopping carts and more.
- We provide several versions of PHP (PHP 5.0, PHP 5.1, PHP 5.2) on a per-folder basis.

Our PHP hosting package is the best offer on the market - it has the lowest price for the quality and features it includes. Sign up now for our [Professional PHP Hosting!](#)

### MySQL Tutorial What is MySQL?

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).

SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. MySQL is an essential part of almost every open source PHP application. Good examples for PHP/MySQL-based scripts are [phpBB](#), [osCommerce](#), and [Joomla](#).

One of the most important things about using MySQL is to have a MySQL specialized host. Here are some of the things SiteGround is proud of:

- We have long experience in providing technical support for MySQL-based web sites. Thanks to it our servers are perfectly optimized to offer the best overall performance for most MySQL applications.
- We offer a lot of FREE MySQL tools including CMS systems, forums, galleries, blogs, shopping carts and more.
- We support both MySQL 4 and MySQL 5. We provide unlimited MySQL databases.

Our MySQL hosting package is the best offer on the market - it has the lowest price for the quality and features it includes. Sign up now for our [Professional MySQL Hosting!](#)

## MySQL Tutorial: Creating MySQL Database and User

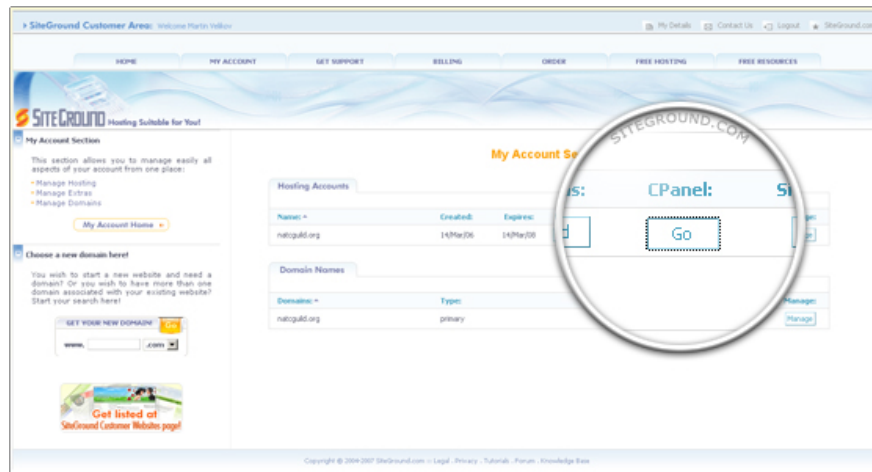
Check out the Creating and Modifying MySQL Databases video tutorial:



Creating and Modifying MySQL Databases Video Tutorial

### Creating a MySQL Database Detailed Tutorial

Creating a MySQL database with cPanel, which is included in all of SiteGround's hosting plans, can be easily accomplished. In order to create a MySQL database on your hosting account, you need to log in the cPanel. This can be done by either using the **Go** button in the **Customer Area -> My Account Account** section:



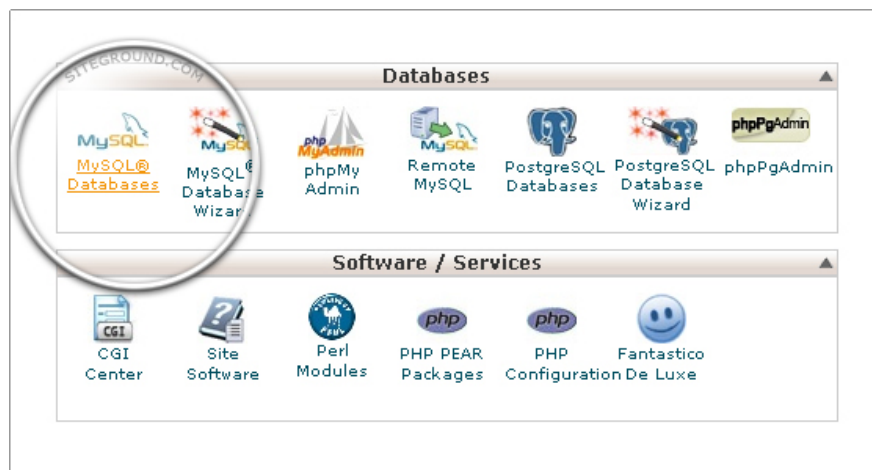
or by entering the following URL in your web browser:

<http://yourdomainname.com/cpanel>  
<http://cpanel.servername.com>



You should replace "yourdomainname.com" with your actual domain name or "servername.com" with the hostname of the SiteGround server on which your account is hosted.

Once you have logged in the cPanel, you can access the MySQL management screen by clicking on the **MySQL Databases** button as shown below.



In order to create a new database you need to enter the desired name for the database in the **New Database** field and click on the **Create Database** button as shown below:

Database field and click on the **Create Database** button as shown below:

**MySQL Databases**

MySQL Databases allow you to store lots of information in an easy to access manner. The databases themselves are not easily read by humans. MySQL databases are required by many web applications including some bulletin boards, content management systems, and others. To use a database, you'll need to create it. Only MySQL Users (different than mail or other users) that have privileges to access a database can read from or write to that database.

Video Tutorial [↓ Jump to MySQL Users](#)

**Create New Database**

New Database:

**Modify Database**

Check DB:

Repair DB:

**Current Databases**

DATABASE	SIZE	USERS	ACTIONS
----------	------	-------	---------

A confirmation screen will be displayed, informing that the database has been successfully created:

HOME

CPanel 11

**MySQL Databases**


**Create MySQL database**

Added the database **newdatabase**.

[ Go Back ]

Home ■ Trademarks ■ Help ■ Documentation ■ Log Out

You can click on the **Go Back** button to return to the database management screen.

 The MySQL database is created with your username as prefix - `username_databasename` and this is the name you have to use for your application.

## Creating a MySQL Username Detailed Tutorial

You can continue with the creation of a user who should be able to connect to the newly created database. To create a user you just need to fill in the desired user name and password as shown in the image below:

**MySQL Users**

**Add New User**

Username:  \*Seven characters max

Password:  Password Strength:

Password (Again):

**Add User To Database**

User:

Database:

Click **Submit** to save your MySQL username.

 You should always bear in mind that your cPanel username will be added as a prefix to all databases and usernames you create.

## Adding a user to a MySQL Database Detailed Tutorial

Once the database and the username are created, you can add certain privileges for the user to the database. In order to do so you should select them both from the corresponding drop down menus

database. In order to do so you should select them both from the corresponding drop-down menus under **Add User To Database** label and click on the **Submit** button:

Password:  Password Strength:   
Very Weak

Password (Again):

**Add User To Database**

User:

Database:

**Current Users**

USERS	DELETE
alvian_user	<input type="button" value="X"/>

You will be redirected to a screen where you will be prompted to choose the desired privileges. It is advisable to select **All Privileges** and click on the **Make Changes** button:

**Manage User Privileges**

User: **alvian\_user**  
Database: **alvian\_newdatabase**

**ALL PRIVILEGES**

<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> CREATE
<input checked="" type="checkbox"/> INSERT	<input checked="" type="checkbox"/> ALTER
<input checked="" type="checkbox"/> UPDATE	<input checked="" type="checkbox"/> DROP
<input checked="" type="checkbox"/> DELETE	<input checked="" type="checkbox"/> LOCK TABLES
<input checked="" type="checkbox"/> INDEX	<input checked="" type="checkbox"/> REFERENCES
<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES	

### Using the MySQL Database Wizard

All of the steps above can be accomplished using the **MySQL Database Wizard**. Click on it and you will be guided through all steps that you should perform in order to successfully create a new MySQL database and add a user to it.



Check out the Using MySQL Databases Wizard video tutorial:



[Using MySQL Databases Wizard Video Tutorial](#)

[Previous](#)

[Next](#)

If you would like to remove a privileged user from a database you can easily do so by clicking on the corresponding button next to the database user as shown below:

Check DB:

Repair DB:

**Current Databases**

DATABASE	SIZE	USERS	ACTIONS
alvian_newdatabase	0.01 MB	alvian_user	Delete Database

**MySQL Users** [↑ Jump to MySQL Databases](#)

**Add New User**

Username:  \*Seven characters max

Password:  Password Strength:

Password (Again):

A confirmation message for the removal will be displayed. This means that the username no longer has any privileges to the database.

### Removing a MySQL database

If you would like a MySQL database removed this can be easily accomplished by clicking on the **Delete Database** button under the **Actions** area:

Check DB:

Repair DB:

**Current Databases**

DATABASE	SIZE	USERS	ACTIONS
alvian_newdatabase	0.01 MB	alvian_user	Delete Database

**MySQL Users** [↑ Jump to MySQL Databases](#)

**Add New User**

Username:  \*Seven characters max

Password:  Password Strength:

Password (Again):

You will get a confirmation screen when the database is removed.

Note that this process is irreversible and all the data from this database will be lost.

### Removing a MySQL User

If a MySQL username is no longer needed you can remove it by clicking on the corresponding **Delete** button under the **Current Users** area:

Password (Again):

**Add User To Database**

User:

Database:

**Current Users**

USERS	DELETE
alvian_user	

Again a confirmation screen will be shown upon successful removal.

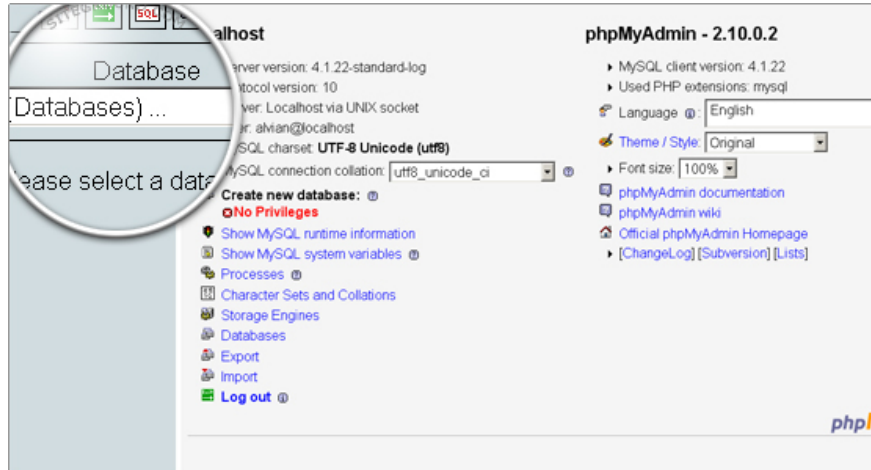
## MySQL Tutorial

### How to Access and Manage your MySQL Database - phpMyAdmin

There are two ways to access and manage your newly created database. The first one is locally through the Web based manager - **phpMyAdmin**, which is accessed from the cPanel **Databases** section.



A new page/tab of your browser will open and the phpMyAdmin will load as shown below:



Check out how to manage MySQL database through phpMyAdmin video tutorial:



[How to Manage MySQL Database Through phpMyAdmin Video Tutorial](#)

With phpMyAdmin you can create a [MySQL backup and restore](#) it.

The second database management solution is to add a [MySQL access host](#) and allow a remote MySQL connection.

## MySQL Tutorial

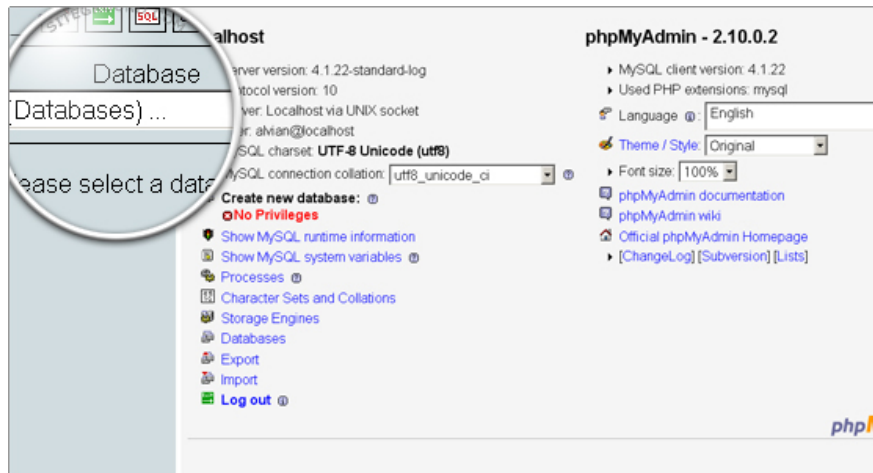
### MySQL Export: How to backup your MySQL database?

You can easily create a dump file(export/backup) of a database used by your account. In order to do so you should access the **phpMyAdmin** tool available in your cPanel:

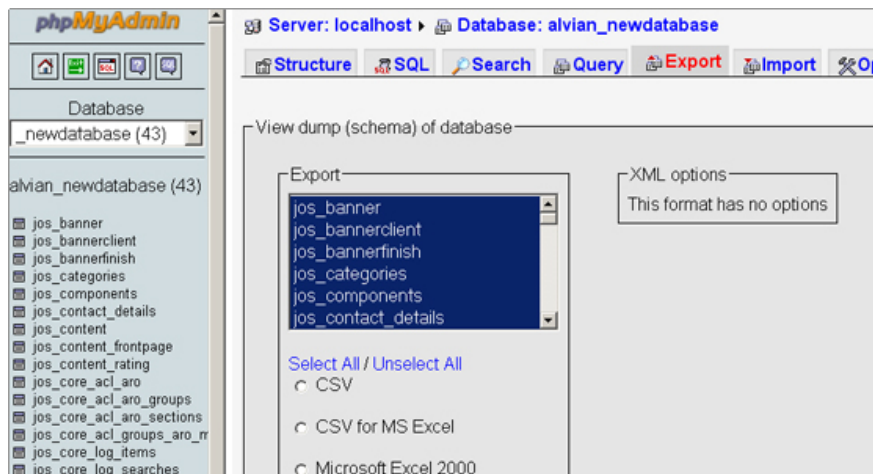


The phpMyAdmin tool will be loaded shortly.

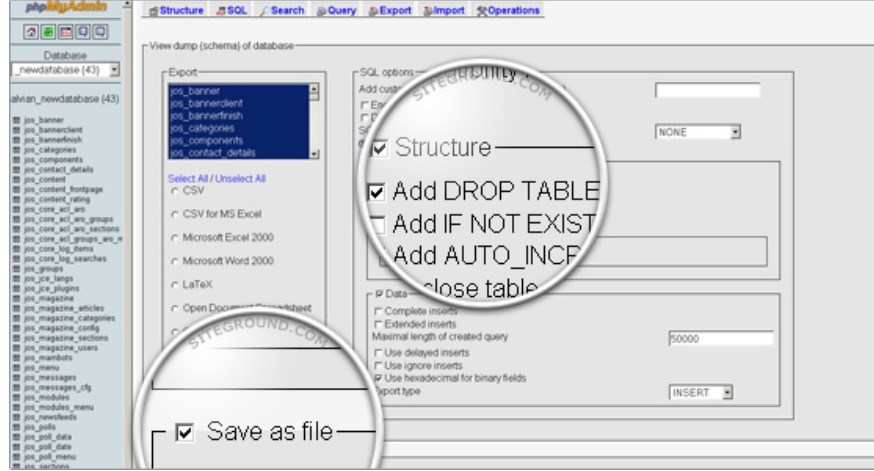
You can select the database that you would like to backup from the drop-down menu called **Database** (located in the upper left corner of the page).



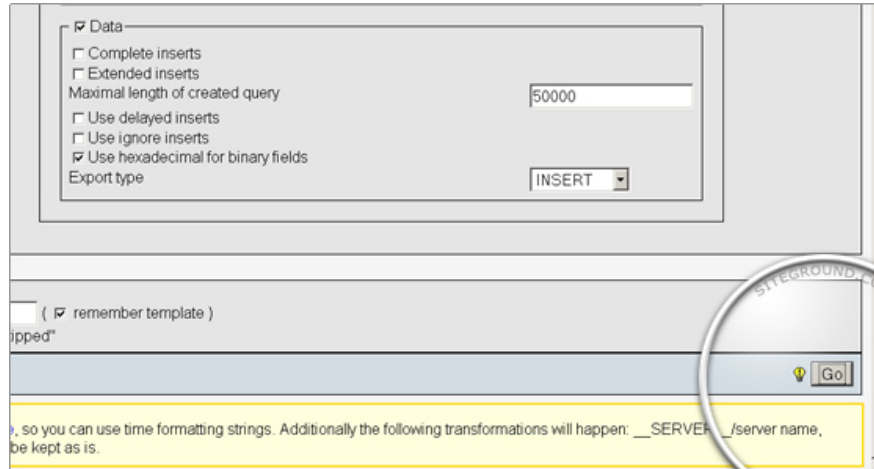
A new page will be loaded in phpMyAdmin showing the selected database. In order to proceed with the backup click on the **Export** tab:



The options that you should select apart from the default ones are **Save as file** (which will save the file locally to your computer in an .sql format) and **Add DROP TABLE** (which will add the drop table functionality if the table already exists in the database backup) as shown below:



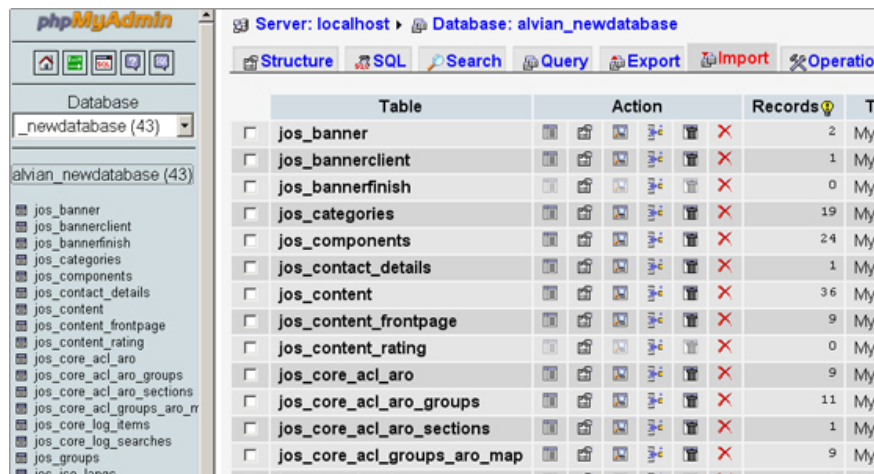
Click on the **Go** button to start the export/backup procedure for your database:



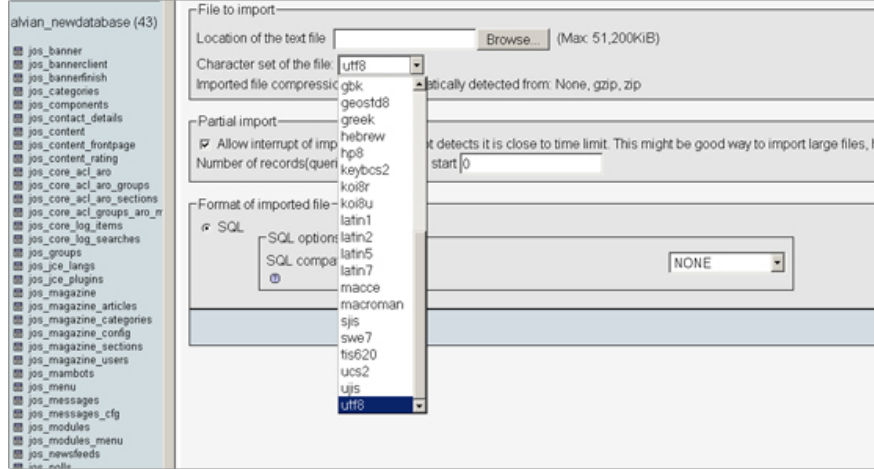
A download window will pop up prompting for the exact place where you would like to save the file on your local computer. It is possible that the download starts automatically. This depends on your browser's settings.

## MySQL Import: How to restore your MySQL database from a backup

To restore (import) a database via **phpMyAdmin**, first choose the database you'll be importing data into. This can be done from the drop-down menu on the left. Then click the **Import** tab:



You have the option of importing an .sql file. Use the **Browse** button to find it on your local computer. Note that you are given the option to choose the character set of the file from the drop down-menu just below the upload box. If you are not certain about the character set your database is using just leave the default one.



In order to start the restore click on the **Go** button at the bottom-right. A notification will be displayed upon a successful database import.

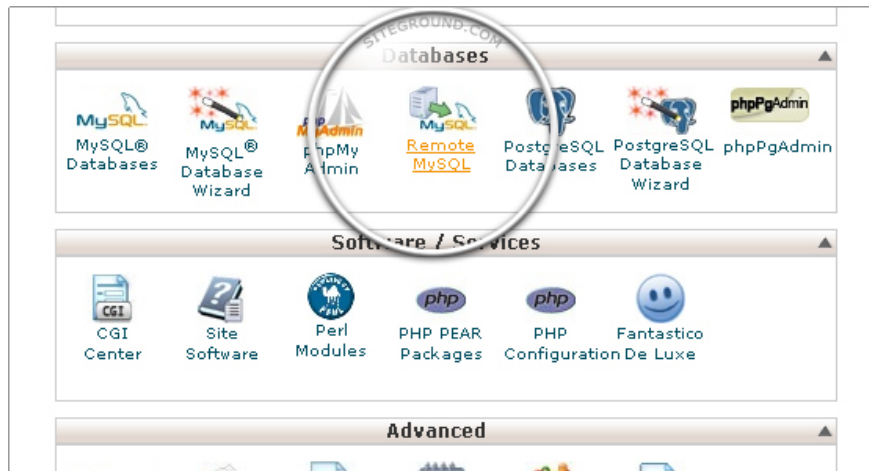
[Previous](#)

[Next](#)

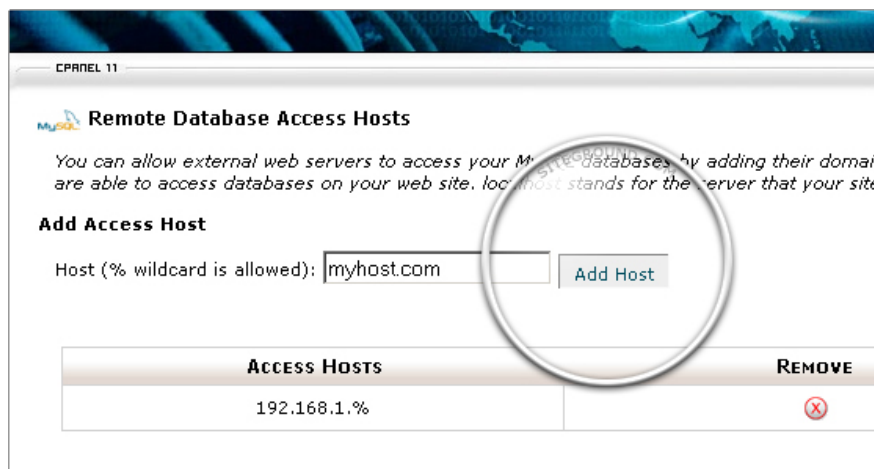
## MySQL Tutorial Adding MySQL access hosts

You can allow access to MySQL databases from an external location by adding its domain name or IP address to the list of hosts that are allowed to access your databases remotely. Additionally, if you would like to manage your database using an application for remote database management, the IP address from which the connection is established should be added to the list of allowed hosts.

In order to add allowed hosts you should access the tool via **cPanel -> Remote MySQL**:



In the **Host** field you should enter the desired domain name or IP address:



Click on the **Add host** button.

A confirmation message will inform that the host is successfully added. Next time you access **Remote MySQL** the newly added host will be listed under **Access Hosts**.

## PHP/MySQL Tutorial - Creating a database table

Before you can do anything with your database, you should create a table and set the corresponding fields in it.

Creating a table in PHPMyAdmin is simple. Just type the name, select the number of fields and click the **Go** button. You will then be taken to a setup screen where you should create the fields for the database.



The screenshot shows a web form with two input fields. The first field is labeled 'Name:' and contains the text 'new'. The second field is labeled 'Number of fields:' and contains the number '33'. Below these fields is a 'Go' button.

Alternatively, you can run a MySQL query, which will create the table. The format is:

```
CREATE TABLE tablename (
```

```
Fields
```

```
)
```

The fields are defined as follows:

```
fieldname type(length) extra info,
```

The fields are separated by comma.

For example, if you wish to create a table called **Members** with 3 fields in it - FirstName, LastName and Age, you should execute the following query:

```
CREATE TABLE Members
(
  FirstName varchar(15),
  LastName varchar(15),
  Age int
);
```

More information on how to connect to the database and how to query it can be found in the following tutorials:

[Step 2: Connect to the database](#)

[Step 3: Query the database](#)

[Step 4: Display table data](#)

[Step 5: Select individual records](#)

## PHP/MySQL Tutorial - PHP MySQL Connection How to Connect to Your Database

» Step 2: You should establish a connection to the MySQL database. This is an extremely important step because if your script cannot connect to its database, your queries to the database will fail.

A good practice when using databases is to set the username, the password and the database name values at the beginning of the script code. If you need to change them later, it will be an easy task.

```
$username="your_username";
$password="your_password";
$database="your_database";
```

You should replace "your\_username", "your\_password" and "your\_database" with the MySQL username, password and database that will be used by your script.

At this point you may be wondering if it is a security risk to keep your password in the file. You don't need to worry because the PHP source code is processed by the server before being sent to the browser. So the visitor will not see the script's code in the page source.

Next you should connect your PHP script to the database. This can be done with the **mysql\_connect** PHP function:

```
mysql_connect(localhost,$username,$password);
```

This line tells PHP to connect to the MySQL database server at '**localhost**' (localhost is the MySQL server which usually runs on the same physical server as your script).

After the connection is established you should select the database you wish to use. This should be a database to which your username has access to. This can be completed through the following command:

```
@mysql_select_db($database) or die( "Unable to select database");
```

It tells PHP to select the database stored in the variable \$database (in our case it will select the database "your\_database"). If the script cannot connect it will stop executing and will show the error message:

Unable to select database

The 'or die' part is useful as it provides debugging functionality. However, it is not essential.

Another important PHP function is:

```
mysql_close();
```

This is a very important function as it closes the connection to the database server. Your script will still run if you do not include this function. And too many open MySQL connections can cause problems for your account. This is a good practice to close the MySQL connection once all the queries are executed.

You have connected to the server and selected the database you want to work with. You can start querying the database now.

[Step 3: Query the database](#)

[Step 4: Display table data](#)

[Step 5: Select individual records](#)

[Previous](#)

[Next](#)

## PHP/MySQL Tutorial - PHP MySQL Connection How to Query the Database

» Step 3: There are at least two ways to query a database. One is to **enter the command in PHP**.

Another way is **to define the command as a variable**.

In this part of the tutorial we will show the first way. The command will look like this:

```
mysql_query($query);
```

The command can be repeated over and over again in the source code. All you need to do is to change the variable.

Here is the complete code that should be used to create a MySQL table in PHP:

```
<?php
$user="username";
$password="password";
$database="database";
mysql_connect(localhost,$user,$password);
@mysql_select_db($database) or die( "Unable to select database");
$query="CREATE TABLE tablename(id int(6) NOT NULL auto_increment,first varchar(15) NOT NULL,last
varchar(15) NOT NULL,field1-name varchar(20) NOT NULL,field2-name varchar(20) NOT NULL,field3-
name varchar(20) NOT NULL,field4-name varchar(30) NOT NULL,field5-name varchar(30) NOT
NULL,PRIMARY KEY (id),UNIQUE id (id),KEY id_2 (id))";
mysql_query($query);
mysql_close();
?>
```

Enter your database, MySQL username and MySQL password in the appropriate positions on the first three lines above.

The next query should fill in the table. Here is a sample one:

```
$query = "INSERT INTO tablename VALUES
(',$field1-name',$field2-name',$field3-name',$field4-name',$field5-name)";
```

You can't insert more values than the number of fields you have created with the first query.

[Step 4: Display table data](#)

[Step 5: Select individual records](#)

[Previous](#)

[Next](#)

## PHP/MySQL Tutorial - PHP MySQL Connection How to Display MySQL Table Data?

» Step 4: After you have created the table and entered the data, you will probably need to display it. This is usually done using basic HTML code which invokes a PHP script.

We will start the example from the beginning. We will populate a new database table with data.

The following HTML code will collect the data from the text boxes and pass it to the PHP script:

```
<form action="insert.php" method="post">
Value1: <input type="text" name="field1-name"><br>
Value2: <input type="text" name="field2-name"><br>
Value3: <input type="text" name="field3-name"><br>
Value4: <input type="text" name="field4-name"><br>
Value5: <input type="text" name="field5-name"><br>
<input type="Submit">
</form>
```

The next thing you need is a new PHP script which will enter the data in the database.

```
<?php
$username="username";
$password="password";
$database="your_database";

$field1-name=$_POST['Value1'];
$field2-name=$_POST['Value2'];
$field3-name=$_POST['Value3'];
$field4-name=$_POST['Value4'];
$field5-name=$_POST['Value5'];

mysql_connect(localhost,$username,$password);
@mysql_select_db($database) or die( "Unable to select database");

$query = "INSERT INTO tablename VALUES
(',$field1-name',$field2-name,$field3-name,$field4-name,$field5-name)";

mysql_query($query);

mysql_close();
?>
```

This script should be saved as **insert.php** so that it can be called by the HTML form.

Now that you have at least one record in your database, you may want to know how you can output this data using PHP.

The first command you will need to use is the SELECT FROM MySQL statement:

```
SELECT * FROM tablename
```

This is a basic MySQL query which will tell the script to select all the records from the tablename table. After the query execution the result will be assigned to a variable:

```
$query="SELECT * FROM tablename";
$result=mysql_query($query);
```

The whole content of the table is now included in a PHP array with the name \$result. Before you can output this data you should change each piece into a separate variable. There are two stages.

The first one is counting the rows. Before you can go through the data in your result variable, you should know the number of the database rows. You could, of course, just type this into your code but it is not a very good solution as the script code will have to be changed every time a new row is added. Instead you can use the command:

```
$num=mysql_numrows($result);
```

The \$num value will be the number of rows stored in \$result. This will be used in a loop to get all the data and display it on the screen.

The second stage is to set up the loop. It will take each row of the result and print the data stored there. In the code below, \$i is the number of times the loop runs. In this way all the records are displayed.

```
$i=0;
while ($i < $num) {
```

CODE

```
$i++;
}
```

This is a basic PHP loop and will execute the code the correct number of times. Each time \$i will be incremented by one. This is useful, as \$i will tell the script which line of the results should be read. As the first line in MySQL output is 0, this will work correctly.

The final part of the output script is to assign each piece of data to its own variable:

```
$variable=mysql_result($result,$i,"fieldname");
```

So to take each individual piece of data in our database we would use the following:

```
$field1-name=mysql_result($result,$i,"field1-name");  
$field2-name=mysql_result($result,$i,"field2-name");  
$field3-name=mysql_result($result,$i,"field3-name");  
$field4-name=mysql_result($result,$i,"field4-name");  
$field5-name=mysql_result($result,$i,"field5-name");
```

You do not need to get the ID field because there is no use for it in the output page.

You can now write a full script to output the data. In this script the data is not formatted when it is printed:

```
<?php  
$username="username";  
$password="password";  
$database="your_database";  
  
mysql_connect(localhost,$username,$password);  
@mysql_select_db($database) or die( "Unable to select database");  
$query="SELECT * FROM tablename";  
$result=mysql_query($query);  
  
$num=mysql_numrows($result);  
  
mysql_close();  
  
echo "<b><center>Database Output</center></b><br><br>";  
  
$i=0;  
while ($i < $num) {  
  
$field1-name=mysql_result($result,$i,"field1-name");  
$field2-name=mysql_result($result,$i,"field2-name");  
$field3-name=mysql_result($result,$i,"field3-name");  
$field4-name=mysql_result($result,$i,"field4-name");  
$field5-name=mysql_result($result,$i,"field5-name");  
  
echo "<b>$field1-name  
$field2-name</b><br>$field3-name<br>$field4-name<br>$field5-name<hr><br>";  
  
$i++;  
}  
  
?>
```

This outputs a list of all the values stored in the database. This will give you a very basic output. It is not useful for a working website. Instead, it would be better if you could format it into a table and display the information in it. Doing the formatting is not complicated. All you need to do is use HTML to print the result by including the variables in the correct spaces. The easiest way to do this is by closing your PHP tag and entering HTML normally. When you reach a variable position, include it as follows:

```
<? echo $variablename; ?>
```

in the correct position in your code.

You can also use the PHP loop to repeat the appropriate code and include it as part of a larger table. The final output is:

```
<html>  
<body>  
<?php  
$username="username";  
$password="password";  
$database="your_database";  
  
mysql_connect(localhost,$username,$password);  
@mysql_select_db($database) or die( "Unable to select database");  
$query="SELECT * FROM tablename";  
$result=mysql_query($query);  
  
$num=mysql_numrows($result);  
  
mysql_close();
```

```

?>
<table border="0" cellspacing="2" cellpadding="2">
<tr>
<th><font face="Arial, Helvetica, sans-serif">Value1</font></th>
<th><font face="Arial, Helvetica, sans-serif">Value2</font></th>
<th><font face="Arial, Helvetica, sans-serif">Value3</font></th>
<th><font face="Arial, Helvetica, sans-serif">Value4</font></th>
<th><font face="Arial, Helvetica, sans-serif">Value5</font></th>
</tr>

<?php
$i=0;
while ($i < $num) {

$f1=mysql_result($result,$i,"field1");
$f2=mysql_result($result,$i,"field2");
$f3=mysql_result($result,$i,"field3");
$f4=mysql_result($result,$i,"field4");
$f5=mysql_result($result,$i,"field5");
?>

<tr>
<td><font face="Arial, Helvetica, sans-serif"><?php echo $f1; ?></font></td>
<td><font face="Arial, Helvetica, sans-serif"><?php echo $f2; ?></font></td>
<td><font face="Arial, Helvetica, sans-serif"><?php echo $f3; ?></font></td>
<td><font face="Arial, Helvetica, sans-serif"><?php echo $f4; ?></font></td>
<td><font face="Arial, Helvetica, sans-serif"><?php echo $f5; ?></font></td>
</tr>

<?php
$i++;
}
?>
</body>
</html>

```

This code will print out table content and add an extra row for each record in the database, formatting the data as it is printed.

#### [Step 5: Select individual records](#)

[Previous](#)

[Next](#)

## PHP/MySQL Tutorial - PHP MySQL Connection How to Select Individual Records

» Step 5: As well as showing the whole database, PHP can be used to select individual records, or records which match certain criteria. To do this you must use a variation of the SELECT query. To display the whole table you used the query:

```
SELECT * FROM tablename
```

If we just want to select records which have value=1 in the field1-name row we would use the following query:

```
SELECT * FROM tablename WHERE field1-name='1'
```

In the same way you could select records based on any field in the database. You can also search in more fields by adding more:

```
field='value'
```

sections into the query.

For further reference you can visit the official websites of [PHP](#) and [MySQL](#).

[Previous](#)

[Next](#)

One of the most important things about using PHP and MySQL is to have a PHP/MySQL specialized host. Here are some of the things we at SiteGround are proud of:

- We have long experience in providing technical support for PHP/MySQL-based web sites. Thanks to it our servers are perfectly optimized to offer the best overall performance for most PHP/MySQL applications.
- We offer FREE installation of many PHP/MySQL applications including CMS systems, forums, galleries, blogs, shopping carts and more.
- We provide several versions of PHP (PHP 5.0, PHP 5.1, PHP 5.2) on a per-folder basis.
- We support both MySQL 4 and MySQL 5. We provide unlimited MySQL databases.
- We develop comprehensive tutorials for the most popular PHP/MySQL scripts and a great

we develop comprehensive tutorials for the most popular PHP/MySQL scripts and a great variety of articles in our Knowledge Base.

Our PHP hosting package is the best offer on the market - it has the lowest price for the quality and features it includes. Sign up now for our [Professional PHP Hosting!](#)

## Zend Optimizer Tutorial

What is Zend Optimizer?

Zend Optimizer is a free application that allows PHP to run files encoded by Zend Guard. Zend Optimizer greatly enhances the performance of PHP applications.

The Zend Optimizer goes over the code generated by the standard Zend run-time compiler and optimizes it for faster execution. The standard Zend run-time compiler used by PHP is indeed very fast, generating code that is usually 2 to 10 times faster. But an application that uses Zend Optimizer can execute scripts another 40% to 100% faster.

The latest stable version of Zend Optimizer is installed on all SiteGround servers as part of our [Professional Zend Optimizer Hosting](#).

### Zend Guard Compatibility:

The encoded files must be compiled with a compatible version of Zend Guard. When you are not sure about the Zend Guard version, always use the latest version of Zend Optimizer. If the versions of Zend Guard and Zend Optimizer are incompatible, the Optimizer will fail to run the encoded files and will display an error message.

To ensure that Zend Optimizer is properly running, create a [phpinfo.php file](#) with the following code:

```
<?php
phpinfo();
?>
```

Once this is done access the file via web browser. The part associated with Zend Optimizer will look like this:

```
This program makes use of the Zend Scripting Language Engine:
Zend Engine vX.Y.Z, Copyright (c) 1998-2007 Zend Technologies.>with Zend
Optimizer vX.Y.Z, Copyright (c) 1998-2007, by Zend Technologies
```

The Zend Optimizer is capable of performing around 20 different optimizations, some of which are quite complex. One of the optimizations Zend Optimizer does is to change post-incrementing to pre-incrementing, where possible, since pre-incrementing is the faster operation of the two. You can adjust your Zend Optimizer to perform only the desired optimizations. Each optimization can be turned on or off by setting the corresponding option.

[Previous](#)

[Next](#)

One of the most important things about using Zend Optimizer is to have a Zend Optimizer specialized host. Here are some of the things we at SiteGround are proud of:

- We have long experience in providing technical support for web sites which use Zend Optimizer.
- We offer FREE installation of many applications which use Zend Optimizer - CMS systems, forums, galleries, blogs, shopping carts and more.
- We provide several versions of PHP (PHP 5.0, PHP 5.1, PHP 5.2) on a per-folder basis.

Our Zend Optimizer hosting package is the best offer on the market - it has the lowest price for the quality and features it includes. Sign up now for our Professional [Zend Optimizer Hosting!](#)

## Pear Modules

By default all SiteGround servers have the following Pear packages installed:

- Archive\_Tar
- Auth\_SASL
- Console\_Getopt
- DB
- HTML\_Template\_IT
- HTTP\_Request
- Log

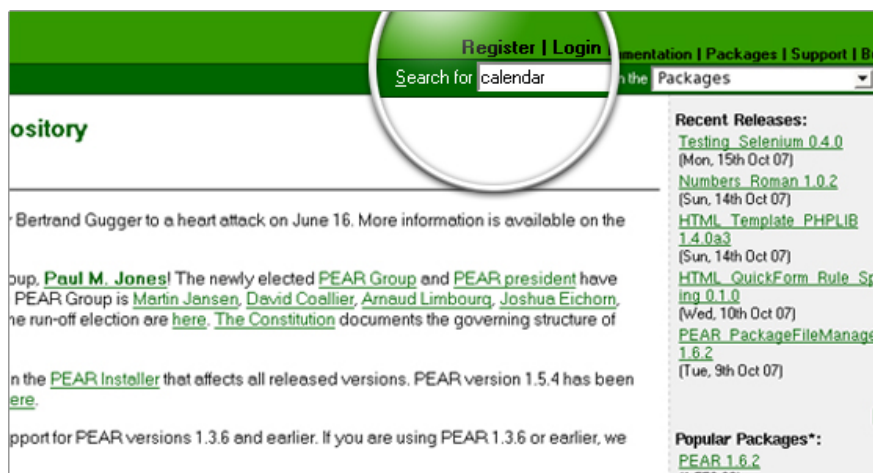
- MDB2
- MDB2\_Driver\_mysql
- MDB2\_Driver\_mysqli
- MDB2\_Driver\_pgsql
- Mail
- Mail\_Mime
- Mail\_mimeDecode
- Net\_DIME
- Net\_Socket
- Net\_URL
- Net\_URL2
- Net\_UserAgent\_Detect
- PEAR
- PEAR\_Frontend\_Web
- SOAP
- Structures\_Graph
- XML\_RPC

The latest stable versions of Pear modules are installed on all SiteGround servers as a part of our Professional [Pear Modules Hosting](#).

## How to install an additional Pear package?

You can easily install any additional Pear package. In this tutorial we will demonstrate how to install the Calendar Pear package.

» Step 1. First go to <http://pear.php.net/> and search for Calendar:



» Step 2. The search will find the Calendar package with its Description: "A package for building Calendar data structures". Follow the link to the [Calendar packet home page](#).

» Step 3. There click on the link to the Download section and choose to download the latest version;

» Step 4. Once you have the package downloaded locally, you have to extract it. For this purpose use any popular archiver like Tar or Rar.

» Step 5. Now you should have a folder called Calendar-X.X.X. In order to simplify things we will rename it to **Calendar**.

» Step 6. Next connect to your account by [FTP](#) and create a directory called **pear** right in the home directory (it does not have to be in the webroot). Upload the **Calendar** directory inside the **pear** directory. You should upload any other Pear packages there.

» Step 7. Next tell PHP to include the local pear directory. This can be done by [creating a file](#) called php.ini and pasting the following source code into it:

```
include_path = ".:usr/lib/php:/usr/local/lib/php:/home/my_cPanel_username/pear";
```

Make sure you change my\_cPanel\_username to your actual cPanel username. Also the first two paths (/usr/lib/php:/usr/local/lib/php) should remain so that the default PHP libraries are loaded.

This **php.ini** file must be copied in each directory where pear packages are needed.

From this point on the Calendar Pear package will be available with the fixed path:

```
include 'Calendar/Calendar.php';
```

Of course, you could just place the Calendar.php in a directory and include it directly. However, this way you will end up with many hardcoded paths and confusion will be increased in case you plan to use more Pear packages.

Following the above steps you can install other Pear packages.

Please feel free to contact us if you have any questions or recommendations about this tutorial at:  
[tutorials 'at' siteground.com](mailto:tutorials@siteground.com)

[Previous](#)