In this tutorial I will show you how to **store** images in a mysql database. I strive to make this tutorial as simple as possible. The example scripts that I provide here are intended to show you how to simply get a single image into your database. It will be up to you to write other scripts using php loops -- and functions like:

* is\_dir()
* is\_file()

**Creating a Test Mysql Database**

Since it is impossible for me know if you already have a mysql database set up, I will include a fictitious database and table for this tutorial. The database will be called**test\_imgs**. It will contain a single table called **pictures**.

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **pics** | **ext** | **gender** |
| **1** | **retergfghfghterhgcfnt hy6e5ghjwtjthj** | **jpg** | **female** |

1. "id" is primary, key, not null, and auto\_increment
2. "pics" blob not null
3. "ext" is varchar(4) not null
4. "gender" is varchar(7) not null
5. I will assume English mysql defaults for everything else.
6. I am calling the directory "/test/."
7. Practice script "test.php"

**Example Php Database Function**

Before I can put anything in my database, I will need to connect to it. I have decided to write a function for this task.

function database\_connect($db\_host, $db\_user, $db\_pw, $db\_name, &$db\_selected, &$connection){  
  
$connection = mysql\_connect($db\_host, $db\_user, $db\_pw);  
  
  
  
if (!$connection){  
  
die("Could not Connect to $db\_host");  
  
}  
  
$db\_selected = mysql\_select\_db($db\_name, $connection);  
  
if (!$db\_selected){  
  
die( "Could not select database $db\_name");  
  
}  
  
}

**Explanation of Database Function**

I named my function "database connect" for obvious reasons, then passed the following arguments:

1. $db\_host -- Database Host
2. $db\_user -- Database User Name
3. $db\_pw -- Database Password
4. $db\_name -- Database Name
5. &$db\_selected -- Passed by reference
6. &$connection -- Passed by reference

Arguments 1 through 4 need no explanation. Five and six were passed by reference so that, if I need to, I can used them outside the function without declaring them "global," or returning them. "$db\_selected" will be of type "bool," which provides a great way to see if the database you are trying to connect to was selected. I suggest that you comment out the "die()" functions when you are not troubleshooting your code.

*Just to make note of it; php does not require that the programmer declare a type for functions or variables. In a programming language like C, the coder would have to declare the above function "VOID" because it does not return anything.*

[Bookmark This Tutorial](javascript:CreateBookmarkLink();)

**Php Script to Load Images in Mysql Database**

Now lets get down to the task of *storing images in a mysql database.*

$handle = fopen("testpic.jpg", "rb");  
$img = fread($handle, filesize('testpic.jpg'));  
fclose($handle);  
//die($img);  
  
$img = base64\_encode($img);  
  
database\_connect  
('localhost', 'root', 'admin', 'img\_test', &$db\_selected, &$connection);  
  
$sql = "insert into pictures values(null,'$img','jpg','female')";  
  
mysql\_query($sql) or die('Bad Query at 12');  
  
echo "Success! You have inserted your picture!";  
?>

**Explanation of Image Script**

Notice that I used "fopen(), fread(), and fclose()" to get the contents of "testpic.php" and**store** it in a variable -- $img.

Next, I take $img and run it through "base64\_encode()."( If you need an explanation of base64\_encode and images, I suggest you Google it. Just know that it works when storing images -- 'That's enough for right now, I think.)

Lastly, after a little SQL magic, I stick the encoded data from my test picture into the database. Viola! Now I have a jpeg image stored in mysql. Make note that I made a point to record the file extension along with the picture itself. I will be using that information when I am ready to [get images out of my sql database](http://mrarrowhead.com/php_tutorials_mah/retreive_images_mysql_php.php).

**How to get images out of mysql database with php and use them on my webpages**

**Author: D.Shaun Morgan**  
  
**Versions and Skill Level**

* PHP Version - PHP 5x
* Mysql Version - 5
* Reader skill level - Intermediate

**Tutorial Outline:**

1. [Creating a Test Mysql Database](http://mrarrowhead.com/index.php?page=retreive_images_mysql_php.php#testdb)
2. [Php Script to Load Images in Mysql Database](http://mrarrowhead.com/index.php?page=retreive_images_mysql_php.php#scriptin)
3. [Php Script to retrieve images from mysql database](http://mrarrowhead.com/index.php?page=retreive_images_mysql_php.php#scriptout)
4. [Example of calling php script with img HTML tag](http://mrarrowhead.com/index.php?page=retreive_images_mysql_php.php#html)
5. [Explanation of Image Script](http://mrarrowhead.com/index.php?page=retreive_images_mysql_php.php#exc)

**See Also:**

1. [How to store images in MySql database](http://mrarrowhead.com/php_tutorials_mah/store_images_mysql_php.php)

[Bookmark This Tutorial](javascript:CreateBookmarkLink();)

This tutorial expands on another database/image how-to [How to Store Images in Mysql Database](http://mrarrowhead.com/php_tutorials_mah/store_images_mysql_php.php). In the "Images In" tutorial I showed you the basic steps needed to open a file (your image file) and read it into a variable ($img), convert that file using php's base64\_encode() function, and stick that file into a Mysql database. Now, you will learn how to bring those images back out of your Mysql database, and output them to a web browser.

**Retrieving Images From Mysql Database With php**

To begin, do the following:

1. Create a mysql database with a table matching (Database Example)
2. Create a directory on your server to place files in (test)
3. Create a file named "test01.php" and place example 1 in it (below)
4. Create a file named "test02.php" and place example 2 php script in it (below)
5. Create a file named "test03.php" and place example 3 HTML in it (below)
6. Copy the following test images and save them in your test folder

     
  
**Be certain that you have changed the database username, password *et cetera* so the test scripts can access your database server.**   
  
**>>NOTE>>** *If you were looking for a tutorial on how to use php to store and retrieve bmp images, please read on... YOU WILL BE LOOKING FOR A LONG TIME! Bmp images are so large that they find little use in web development. Thus, there is very little support for it in php. You will do well to just gear your website toward jpg, gif or png. IF you are insisting on using bmp, you will do better to store them in a folder on your web server with unique names, and record their locations in your database. Storing the location of your pics in mysql database is covered here at mrarrowhead.com. Use the search function to find it.*

**Database example**

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **pics** | **ext** | **gender** |
| **1** | **retergfghfghterhgcfnt hy6e5ghjwtjthj** | **jpg** | **female** |

1. Database name is "test\_imgs"
2. Table name is pictures
3. "id" is primary, key, not null, and auto\_increment
4. "pics" blob not null
5. "ext" is varchar(4) not null
6. "gender" is varchar(7) not null
7. I will assume English mysql defaults for everything else.
8. Practice script "test.php"

**Example 1**

<?php  
$host = '';   
  
$user = '';   
  
//$pw = '';   
  
//$db = '';   
  
$testpic = 'testpic.jpg';  
$ext = 'jpg';   
  
//$testpic = 'testpic.gif';  
//$ext = 'gif';   
  
//$testpic = 'testpic.png';  
//$ext = 'png';  
  
  
  
mysql\_connect($host,$user,$pw);   
  
mysql\_select\_db($db);   
  
$handle = fopen($testpic, "rb");  
$img = fread($handle, filesize($testpic));  
fclose($handle);  
  
  
$img = base64\_encode($img);   
  
$sql = "insert into pictures values(null,'$img','$ext','female');";   
  
mysql\_query($sql) or die('Bad Query at 12'.mysql\_error());   
  
echo "Success! You have inserted your picture!";  
?>   
  
< href="test02.php">Next</a>

**Example 2**

<?php  
$host = '';  
$user = '';  
$pw = '';  
$db = '';  
  
  
mysql\_connect($host,$user,$pw);   
  
mysql\_select\_db($db);   
  
$sql = "select pic, ext from pictures where id='1'";   
  
$result = mysql\_query($sql) or die('Bad query at 12!'.mysql\_error());   
  
while($row = mysql\_fetch\_array($result,MYSQL\_ASSOC)){  
  
$db\_img = $row['pic'];  
$type = $row['ext'];  
  
  
}   
  
$db\_img = base64\_decode($db\_img); //print\_r($db\_img );  
  
$db\_img = imagecreatefromstring($db\_img);  
if ($db\_img !== false) {   
switch ($type) {  
case "jpg":  
header("Content-Type: image/jpeg");  
imagejpeg($db\_img);  
break;  
case "gif":  
header("Content-Type: image/gif");  
imagegif($db\_img);  
break;  
case "png":  
header("Content-Type: image/png");  
imagepng($db\_img);  
break;  
}  
  
  
}  
imagedestroy($db\_img);  
?>

**Example 3**

<img src="test02.php"/>

**Tutorial Conclusion and Explanation**

After building the database, and uploading all the example files and images to your server, simply navigate to the test01.php script on your server. By default the test script should open testpic.jpg and send it to your mysql server with an id of "1". Click the next link and go to test\_02.php to view your image. Then type test03.php into your browser's nav bar.

**Notice that on the test03.php page, I inserted test\_02.php in the img tag.**  
  
<img src="test02.php" alt=""/>

This works great for creating on the fly images with your html. Using the img tag allows me to call a php script without using an include, or iframe. It also sets up the right header for my image to display. If you notice on test02.php, I had to send an header before I used the image function to output my image to browser. The php header function is not necessary if you use the test03.php method. Comment the header functions out of test02.php, then run both test02.php and test03.php. test03.php will give an image, and the former will give a mess.

Repeat the steps above for the .gif and .png images. I suggest you practice with some loops, if statements, and a form so that your php script can handle all three image situations based on a user's choice.   
  
*NOTE:  
php also supports wbmp for wireless. You can find the appropriate functions in the php manual.*

image\_data (LONGBLOB)  
image\_type (VARCHAR(50))  
image\_description (VARCHAR(255))

Third, paste the following code in upload.php:

$imageData = file\_get\_contents($\_FILES['image\_name']['tmp\_name']);  
$imageType = file\_get\_contents($\_FILES['image\_name']['type']);  
$imageStore = new ImageStore();  
$imageStore -> image\_data = $imageData;  
$imageStore -> image\_type = $imageType;  
$imageStore -> image\_description = "an image";  
if ($imageStore -> Save())  
{  
 echo "image successfully saved";  
}

and saving an image to a database is as simple as that. You’ve now stored the uploaded picture in a database table.

Retrieving the image is also pretty easy:

header("Content-type: ".$imageStore->image\_type);  
echo $imageStore -> image\_data;

People have different opinions on saving images in database. Some says, "Why bother database if we can handle this by saving images to disk". I am agree with this;).

Most of the requirements can be fulfilled by saving the images to disk. This reduces the unnecessary load on MySQL.

Here is the small code to save the image in MySQL with the help of PHP.

Images are saved in MySQL as BINARY data. BINARY data can not be saved in varchar or char data types, for this purpose we need a data type which can handle binary data. [BLOB](http://dev.mysql.com/doc/refman/5.0/en/blob.html) columns are treated as binary strings (byte strings). The following table is fulfilling our requirements for a simple test with a BLOB field.

**CREATE TABLE** `images` (

`id` **INT**(11) **NOT NULL** **AUTO\_INCREMENT**,

`image` **MEDIUMBLOB** **NOT NULL**,

**PRIMARY KEY** (`id`)

) ENGINE=MyISAM;

Now we have a MySQL table ready to store the image. Next steps are easy i.e.

1. Read the image
2. Encode the image data
3. Save binary data in DB

These 3 steps are performed with the following PHP code

$image = [chunk\_split](http://www.php.net/chunk_split)([base64\_encode](http://www.php.net/base64_encode)([file\_get\_contents](http://www.php.net/file_get_contents)("image.jpg")));

$query = "INSERT INTO images (image) VALUES('$image')";

[mysql\_query](http://www.php.net/mysql_query)($query) or [die](http://www.php.net/die)([mysql\_error](http://www.php.net/mysql_error)());

[echo](http://www.php.net/echo) "Image id is ".[mysql\_insert\_id](http://www.php.net/mysql_insert_id)();

Now we have saved the image in database successfully. The next step is to display the image.

*// showimage.php*

[header](http://www.php.net/header)('Content-type: image/jpeg');

$query = "SELECT image from images where id=1";

$rs = [mysql\_fetch\_array](http://www.php.net/mysql_fetch_array)([mysql\_query](http://www.php.net/mysql_query)($query));

[echo](http://www.php.net/echo) [base64\_decode](http://www.php.net/base64_decode)($rs["image"]);

Ohhh!!! You wanted to display the image in HTML page. No problem, call this file as a image in <img> .e.g.

[**<html>**](http://december.com/html/4/element/html.html)

[**<head>**](http://december.com/html/4/element/head.html)

[**<title>**](http://december.com/html/4/element/title.html)Image Test**</title>**

**</head>**

[**<body>**](http://december.com/html/4/element/body.html)

[**<h1>**](http://december.com/html/4/element/h1.html)Displaying image from database**</h1>**

[**<img**](http://december.com/html/4/element/img.html) src="showimage.php" /**>**

**</body>**

**</html>**