

## Capítulo 5: Manejo de Bases de Datos (MySQL)

Una de las características más importantes de PHP es su integración con diversos motores de base de datos,. El PHP está construido para generar en forma sencilla páginas web dinámicas a partir de información almacenada en bases de datos. A continuación mostramos las funciones más importantes y ejemplos típicos de uso para una base MySQL.

### 1. Conexión a la base

```
$db_link=mysql_connect(hostname, user, password);
```

Ejemplo:

```
$db=mysql_connect("localhost","root","secreto");
```

La función realiza la conexión a la base Mysql y devuelve "false" si hubo algún error en la conexión o un link a la conexión a la base en caso de que la conexión sea exitosa.

El link es un número que indica la sesión dentro del MySQL. Esta sesión se mantiene hasta que [...]. Para finalizar la conexión se debe utilizar la función mysql\_close(). Es muy importante cerrar la conexión a la base de datos una vez finalizadas las transacciones para evitar la sobrecarga en el motor de la base de datos.

### 2. Selección de la base de datos a utilizar

```
mysql_select_db(database_name, db_link);
```

Ejemplo:

```
mysql_select_db("test",$db);
```

Esta función configura cual es la base de datos que se utilizara por omisión. En este caso el link a utilizar en esta función es el link que se obtuvo al ejecutar la función mysql\_connect.

La función mysql\_select\_db devuelve el valor "false" en caso de que se encuentre algún error, como por ejemplo la inexistencia de la base de datos.

En este punto cabe aclarar que la denominación de las bases de datos de MySQL es case-sensitive, por lo que debemos mantener un standard a la hora de elegir los nombres de las distintas bases de datos.

### 3. Queries a la base de datos.

```
$result=mysql_query(query,db_link);
```

Ejemplo:

```
$result=mysql_query("update clientes set deudor='si' where apellido='Picapiedras', $db)
```

```
$query="insert into clientes (nombre, Apellido) values (Pedro, Mármol) ";  
$result=mysql_query($query,$db);
```

Nuevamente el link que se debe usar es el que se obtiene al conectarse a la base, mysql\_query devuelve falso en caso de que el query no pueda ejecutarse (error de SQL) o bien un result set en los casos que devuelva algún tipo de datos como por ejemplo en un select..

Es muy importante fijarse con que usuario se realizó la conexión a la base de datos a la hora de ejecutar el `mysql_connect`, ya que la gran mayoría de los errores producidos en esta instancia son el resultado de la falta de permisos para realizar la consulta.

## 4. Cantidad de Filas Consultadas o Modificadas

### 4.1 Filas Consultadas

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

Ejemplo:

```
$query="select nombre, telefono from contactos where edad between 20 and 25 and sexo='f';"  
$result=mysql_query($query,$db);  
$cant=mysql_num_rows($result);
```

Esta función devuelve la cantidad de filas que se obtuvieron luego de ejecutar una instrucción de consulta como por ejemplo la función `select`.

En el caso del ejemplo, en la variable `$cant` nos dirá cuantas chicas de entre 20 y 25 años tenemos en nuestra lista de contactos.

### 4.2 Filas Modificadas

```
$cantidad=mysql_affected_rows(db_link);
```

Ejemplo:

```
$cuantos=mysql_affected_rows($db);
```

Devuelve cuantos registros fueron afectados por un query con `insert`, `update`, o `delete`, notar que se le pasa el `db_link` ya que el `result_set` no tiene sentido. Si el query fue un `delete` sin clausula "where" esta función devuelve cero independientemente del número de registros eliminados de la tabla.

## 5. Obtención de registros de una consulta

### 5.1 Obtención de datos en un result set

```
$var=mysql_fetch_row(result_set);
```

Ejemplo:

```
$query="select nombre, telefono from contactos where edad between 20 and 25 and sexo='f';"  
$result=mysql_query($query,$db);$rs=mysql_fetch_row($result);
```

Toma un registro del result set y lo devuelve en un vector en el cual el elemento con índice 0 es la primer columna del registro, el elemento con índice 1 es la segunda columna, etc. Si no hay más registros por devolver devuelve `false`.

Los valores de los campos solicitados en el result set son devueltos en forma de array, es decir que para acceder al valor de nombre de la primera fila de nuestro ejemplo debo llamar a la variable `$rs[0]`.Ejemplo:

```
while($v=mysql_fetch_row($result)) {  
    print("Columna 0: $v[0] Columna 1: $v[1]");  
}
```

## 5.2 Obtención de datos en un vector

```
$query="select nombre, telefono from contactos where edad between 20 and 25 and sexo='f';  
$result=mysql_query($query,$db);  
$rs=mysql_fetch_array($result);
```

Ejemplo:

```
$rs=mysql_fetch_array($result);
```

Funciona de forma idéntica a `mysql_fetch_row` pero devuelve los resultados en un vector asociativo indexado por nombre de columna, por lo que para obtener los nombres de nuestra consulta es necesario llamar al vector de la forma `$rs["nombre"]`

Ejemplo:

```
while($v=mysql_fetch_array($result)) {  
    print("Columna 0: $v["nombre"]");  
}
```

## 6. Cerrar la conexión

```
mysql_close(db_link)
```

Por último, esta función nos sirve para cerrar la conexión a la base se usa.

## Otras funciones:

Función	Descripción
<code>result=mysql_create_db(database_name,db_link);</code>	Crea una base de datos con el nombre indicado, devuelve true o false según el resultado.
<code>Result=mysql_data_seek(result_set, position);</code>	Mueve el puntero interno de cada result set a la posición indicada (el primer registro es la posición 0), de esta forma la próxima llamada a <code>mysql_fetch_row</code> o <code>mysql_fetch_array</code> devolvera el registro que se acaba de apuntar. Devuelve true o false.
<code>Result=mysql_db_query(database_name, query, db_link);</code>	Realiza una consulta a una base indicada, es obviamente reemplazable por un <code>mysql_select_db</code> y un <code>mysql_query</code> aunque resulta útil si hay que consultar tablas en varias bases distintas.
<code>Result=mysql_drop_db(database_name,db_link);</code>	Dropea la base con el nombre indicado, devuelve true o false.
<code>Object=mysql_fetch_field(result_set, numero_columna);</code>	Devuelve un objeto con información sobre la columna indicada de un result set (0 es la primera columna, 1 la segunda, etc). Las propiedades que se setean en el objeto son: <ul style="list-style-type: none"><li>• <code>name</code> – nombre de la columna</li><li>• <code>table</code> – nombre de la tabla</li><li>• <code>max_length</code> – longitud maxima de la columna</li><li>• <code>not_null</code> – 1 si la columna no puede ser null</li><li>• <code>primary_key</code> - 1 si la columna es primary key</li><li>• <code>unique_key</code> - 1 si la columna es unique key</li><li>• <code>multiple_key</code> - 1 si la columna no es unique key</li><li>• <code>numeric</code> - 1 si la columna es numerica</li><li>• <code>blob</code> - 1 si la columna es un BLOB</li><li>• <code>type</code> – Tipo de la columna</li><li>• <code>unsigned</code> - 1 si la columna es sin signo</li><li>• <code>zerofill</code> - 1 si la columna es zero-filled</li></ul>
<code>object=mysql_fetch_object(result_set);</code>	Similar a <code>mysql_fetch_array</code> con la diferencia de que los resultados se devuelven en un vector en lugar de un vector asociativo. Luego se puede acceder a las distintas columnas del registro devuelto como propiedades ( <code>data_members</code> ) del objeto devuelto.
<code>int=mysql_num_fields(result_set);</code>	Devuelve el numero de columnas en un result set.

## Manejo de errores:

```
Mensaje=mysql_error();
```

Devuelve un texto con el mensaje correspondiente al error que ocurrió en caso de que se produzca un error en la base de datos.

Ejemplo:

```
$result=mysql_query($query,$db);  
if(!$result) {  
    $x=mysql_error();  
    die("Ocurrió un error en la base de datos: $x");  
}
```

## Generación dinámica de páginas a partir de una base de datos:

Repasemos el layout de nuestro web-site ejemplo:

LOGO	BANNER		
BUSCADOR			
BOTON S1	BOTON S2	BOTON S3	BOTON S4
Barra de links y Aplicaciones	CONTENIDOS		
Informacion de copyright, y pie de página			

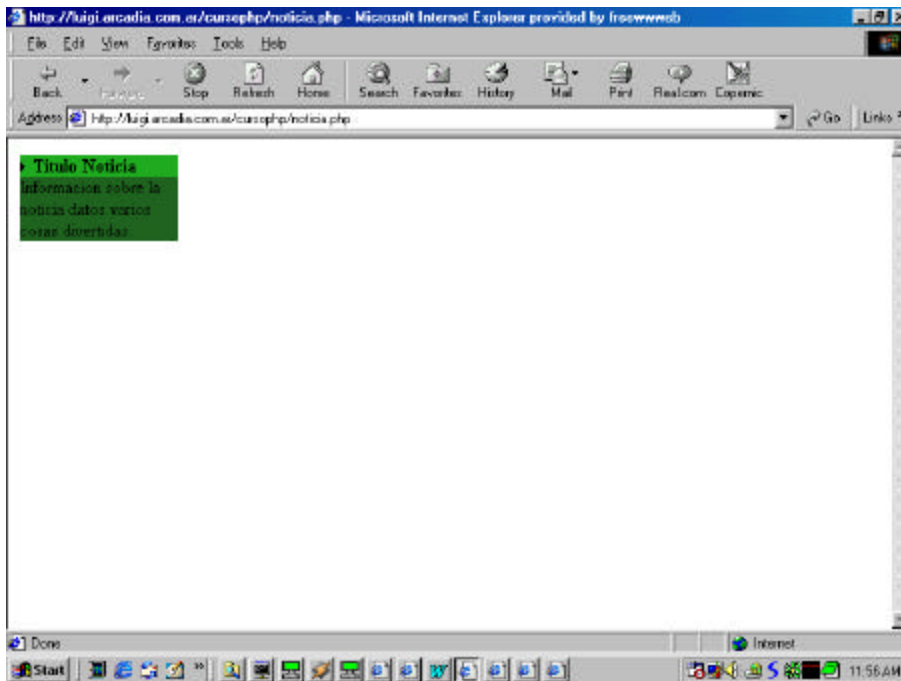
Supongamos que a la barra de links y aplicaciones a la derecha se decide agregarle un módulo “noticia de la semana” el layout queda entonces de la siguiente manera:

LOGO	BANNER		
BUSCADOR			
BOTON S1	BOTON S2	BOTON S3	BOTON S4
Noticia de la semana.	CONTENIDOS		
Barra de links y aplicaciones.			
Información de copyrigh, y pie de página			

Se define que el módulo php que mostrará la noticia de la semana se denominará noticia.php

Ejercicio: Modificar el archivo layout1.php para reflejar el cambio.

A continuación una tarea habitual es que el equipo de diseño realice un archivo html (o php) con un ejemplo de cómo debe quedar para que sea visualmente agradable el módulo noticia de la semana, por ejemplo puede verse de la siguiente manera:



El código HTML del módulo noticia.php es el siguiente:

```
<table bgcolor="#22AE22" width="140" border="0" cellspacing="0" cellpadding="0">
<tr><td>

<b>Título Noticia</b>
</td></tr>
<tr><td bgcolor="#226622">
Información sobre la noticia datos varios cosas divertidas.
</td>
</tr>
</table>
```

Lo que debemos hacer es reemplazar el título de la noticia y el texto de la misma por valores que obtenemos de la base de datos. Para ello decidimos crear una tabla llamada "noticia" en donde almacenaremos la noticia de la semana, además tenemos que construir un cargador que permita insertar el título y texto de la noticia en la base.

Definimos la tabla noticia en MySQL de la siguiente manera:

```
CREATE TABLE noticia(
  título varchar(40),
  texto text,
  fecha datetime
);
```

La fecha la vamos a usar para poder insertar varias noticias y que en la página se muestre siempre la última (la que tiene fecha más reciente), el historial de noticias de la semana puede usarse eventualmente en otra página para mostrar un listado de los títulos de la noticia de la semana, etc.

El formulario cargador que vamos a utilizar usa las técnicas que vimos en el capítulo de formularios y le agregamos el manejo de la base de datos para que realice el insert en la tabla de noticias.

```
<?
//Este código se ejecuta cuando se presiono el botón de submit
if(isset($proc)) {
  $dab=mysql_connect("localhost","root","seldon");
  mysql_select_db("curso",$dab);
  $query="INSERT into noticia(título,texto,fecha) values('$título','$texto',now()
)";
  $res=mysql_query($query,$dab);
  if(!$res) {
    $x=mysql_error();
    print("Se produjo un error al insertar: $x");
  }
}
?>
<form action="<?print("$PHP_SELF");?>" method="post">
Título de la noticia: <input type="text" maxlength="40" name="título"> <BR>
Texto de la noticia: <textarea rows="10" cols="80" name="texto"></textarea><BR>
<input type="submit" name="proc">
</form>
```

Luego podemos modificar el módulo noticia.php para que obtenga el título y texto de la última noticia y sea esto lo que se muestre al usar el módulo.

Noticia.php modificado queda de la forma:

```
<?
//Obtenemos los datos de la noticia a buscar
$dab=mysql_connect("localhost","root","seldon");
if(!$dab) {$título="No hay noticia";$texto="No hay noticia";} else {
  mysql_select_db("curso",$dab);
  $query="select título,texto from noticia order by fecha desc";
  $res=mysql_query($query,$dab);
  if(!$res) {$título="No hay noticia";$texto="No hay texto";} else {
    $r=mysql_fetch_row($res);
    $título=$r[0];
    $texto=$r[1];
  }
}
?>
<table bgcolor="#22AE22" width="140" border="0" cellspacing="0" cellpadding="0">
<tr><td>

<b><?print("$título");?></b>
</td></tr>
<tr><td bgcolor="#226622">
<?print("$texto");?>
</td>
</tr>
</table>
```

Como podemos ver probando el módulo o usando el layout modificado con la inclusión de este módulo a partir de este momento la noticia de la semana que se muestra al ingresar a la página es la última noticia ingresada en la base, usando el cargador de noticias puede modificarse la misma cargando una nueva noticia y la próxima vez que se acceda a la base automáticamente se ve la nueva noticia. Este es el principio básico de un web-site dinámico y de la generación dinámica y en tiempo real de páginas web.

## Almacenamiento de objetos binarios en MySQL:

Un tema interesante en Mysql es usar la base de datos para guardar además de datos de tipo texto datos binarios como por ejemplo código html o imágenes. De esta forma se pueden almacenar completamente en la base de datos todos los elementos necesarios para construir un web-site.

```
CREATE TABLE binary_data (
id INT(4) NOT NULL AUTO_INCREMENT PRIMARY KEY,
description CHAR(50),
bin_data LONGBLOB,
filename CHAR(50),
filesize CHAR(50),
filetype CHAR(50)
);
```

El siguiente script puede usarse para insertar objetos binarios en la base de datos desde un browser. Notar que se usa el tag input type="file" de un form html para subir un archivo.



```

<HTML>
<HEAD><TITLE>Store binary data into SQL Database</TITLE></HEAD>
<BODY>

<?php
if ($submit) {
//codigo que se ejecuta si se presiono el botón submit
MYSQL_CONNECT( "localhost", "root", "password");
mysql_select_db( "binary_data");
$data = addslashes(fread(fopen($form_data, "r"), filesize($form_data)));
$result=MYSQL_QUERY( "INSERT INTO binary_data
(description,bin_data,filename,filesize,filetype) ".
"VALUES
('$form_description','$data','$form_data_name','$form_data_size','$form_d
ata_type')");
$id= mysql_insert_id();
print "<p>Database ID: <b>$id</b>";
MYSQL_CLOSE();
} else {
// sino mostrar el formulario para nuevos datos:
?>
<form method="post" action=" <?php echo $PHP_SELF; ?>"
enctype="multipart/form-data">
File Description:<br>
<input type="text" name="form_description" size="40">
<INPUT TYPE="hidden" name="MAX_FILE_SIZE" value="1000000">
<br>File to upload/store in database:<br>
<input type="file" name="form_data" size="40">
<p><input type="submit" name="submit" value="submit">
</form>
<?php
}
?>
</BODY>
</HTML>

```

Notar el uso de la variable predefinida \$PHP\_SELF que tiene el nombre del script de esta forma el formulario se llama a si mismo independientemente del nombre que se le ponga al archivo.

El siguiente script (getdata.php) puede usarse para recuperar los datos desde la base de datos, notar que el script espera recibir la variable \$id con el id del registro a recuperar de la tabla.

```

<?php
if($id) {
@MYSQL_CONNECT( "localhost", "root", "password");
@mysql_select_db( "binary_data");
$query = "select bin_data,filetype from binary_data where id=$id";
$result = @MYSQL_QUERY($query);
$data = @MYSQL_RESULT($result,0, "bin_data");
$type = @MYSQL_RESULT($result,0, "filetype");
Header( "Content-type: $type");
echo $data;
};
?>

```

Para usar una imagen que se obtiene de la base de datos se puede usar:

```

```

Notar como se le pasa la variable id al script para saber cual es el registro a recuperar de la base.

Usando este conjunto de scripts es sencillo:

- Armar un cargador que permita insertar en la base de datos archivos binarios (imágenes, código html, etc)
- Utilizar los datos almacenados en la base para insertarlos en una página web usando el script getdata o una variante del mismo.