

BASES DE DATOS - USO DE PHP CON MySQL. CONSULTAS BÁSICAS

Una cuestión a comentar es que el acceso a base de datos con PHP admite dos formas distintas: las funciones tipo mysql, por ejemplo mysql_connect, son las más antiguas, cuyo uso se desaconseja si se está trabajando con un servidor con una versión de MySQL moderna. Las funciones tipo mysqli, por ejemplo mysqli_connect, son las más modernas, cuyo uso se recomienda siempre que se esté trabajando con servidores actualizados. Las funciones mysqli pueden escribirse de dos formas: con estilo orientado a objetos o con estilo por procedimientos. Nosotros vamos a ver cómo usar estas funciones con el estilo por procedimientos, que es más sencillo inicialmente.

FUNCIÓN mysqli_connect (mysql_connect)

Esta función nos permite crear una conexión con una base de datos concreta.

La sintaxis que usaremos para esta función es:

```
$nombreConexion = mysqli_connect($hostname , $nombreUsuario , $contraseña);  
Para servidores no actualizados usar mysql_connect en lugar de mysqli_connect
```

La función devuelve una conexión almacenada en la variable \$nombreConexion, o FALSE en caso de error. Los parámetros que hemos indicado son:

\$hostname es un nombre de servidor (dato que debemos consultar en nuestra cuenta de hosting).

\$nombreUsuario es el nombre de usuario de base de datos, nombre que habremos especificado al crear la base de datos (si no lo hemos hecho, puede ser el mismo que para acceder a cPanel ó root).

\$contraseña es la contraseña de acceso para el usuario de base de datos, contraseña que habremos especificado al crear la base de datos (si no lo hemos hecho, puede ser la misma que para acceder a cPanel ó root).

FUNCIÓN mysqli_close (mysql_close)

Esta función nos permite cerrar la conexión con una base de datos anteriormente abierta. Siempre es recomendable cerrar una conexión cuando hayamos terminado de usarla. La sintaxis que usaremos es:

```
mysqli_close($nombreConexión);  
Para servidores no actualizados usar mysql_close en lugar de mysqli_close
```

La función devuelve TRUE si se ha cerrado correctamente ó FALSE en caso de error.

El identificador o nombre de conexión será aquel que hayamos obtenido previamente usando la función `mysqli_connect`.

FUNCIÓN `mysqli_select_db` (`mysqli_select_db`)

Esta función nos permite seleccionar una base de datos concreta dentro de todas las existentes en el servidor al que nos conectemos.

```
mysqli_select_db ($nombreConexión, $nombreBaseDatos);  
Para servidores no actualizados usar mysqli_select_db ($nombreBaseDatos, $nombreConexión);
```

La función devuelve TRUE si se ha realizado la selección correctamente ó FALSE en caso de error.

El nombre de conexión será aquel que hayamos obtenido previamente con la función `mysqli_connect`.

FUNCIÓN `mysqli_query` (`mysqli_query`)

Esta función nos permite ejecutar una consulta a la base de datos que especifiquemos.

La sintaxis que emplearemos es:

```
$result = mysqli_query($nombreConexion, "Consulta aquí"); ó mysqli_query($nombreConexion, $query);  
Para servidores no actualizados usar $result = mysql_query("Consulta aquí", $nombreConexion);
```

Donde "Consulta aquí" se refiere a una consulta realizada utilizando el lenguaje SQL de consultas a bases de datos, por ejemplo una consulta puede ser "SELECT * FROM agenda", que podría traducirse como "selecciona todas las filas de la tabla agenda".

Esta función devuelve TRUE o FALSE para indicar si las operaciones UPDATE, INSERT o DELETE han tenido éxito. Para la operación SELECT devuelve un nuevo identificador de resultado. UPDATE significa "actualizar", INSERT significa "insertar" y DELETE significa "BORRAR". SELECT significa "seleccionar".

El nombre de conexión será aquel que hayamos obtenido previamente con la función `mysqli_connect`.

Esta función la entenderemos mejor con el ejemplo que veremos más adelante.

RECUPERAR UN RESULTADO: DATA SEEK, FETCH ARRAY (MYSQL RESULT)

Una sentencia de consulta normalmente devuelve “un conjunto de resultados” que según el ejemplo de sintaxis anterior tendríamos en una variable denominada \$result. Por ejemplo, podríamos tener como resultados de una consulta lo siguiente:

nombre	apellidos	direccion	telefono	edad	altura
Manuel Jesús	López de la Rosa	C/Juan Bautista Nº 3	658954875	32	1.80
María	Manzano Cabezas	C/Arco del triunfo Nº 7	695001002	19	1.99

La cuestión que nos planteamos ahora es: ¿Cómo extraer el dato de una celda concreta? Por ejemplo el nombre de la primera fila?

```
mysqli_data_seek ($result, numeroDeFila); ó también válido $result->data_seek(numeroDeFila);  
$extraido= mysqli_fetch_array($result); ó también válido $extraido=$result->fetch_array();
```

Aquí estamos dando dos pasos: una sentencia como `mysqli_data_seek ($result, 0);` significaría “posicionarte en la fila 0 de los resultados” (la primera fila). También es válido escribir `$result->data_seek(0);` que tiene el mismo efecto. Si escribiéramos `$result->data_seek(15);` significaría “posicionarte en la fila 16 de los resultados” (tener en cuenta que se empieza a contar por cero, de ahí que 0, 1, 2, ... , 15 resulten 16 filas).

La sentencia `$extraido= mysqli_fetch_array($result);` indica que los valores existentes en la fila se introduzcan en un array cuyos índices en principio pueden ser tanto asociativos (el nombre de la columna) como numéricos (empezando por cero). Por ejemplo si la primera columna en la tabla de la base de datos es <<ciudad>> podemos usar `$extraido['ciudad']` para acceder al valor existente en la fila con la que estemos trabajando para la columna ciudad. Igualmente podríamos usar `$extraido[0]` para referirnos a la primera columna, `$extraido[1]` para referirnos a la segunda columna, `$extraido[2]` para referirnos a la tercera columna y así sucesivamente.

Sintaxis mysql: mysql_result

En servidores no actualizados se puede usar la función `mysql_result` aunque al no pertenecer a la serie de funciones `mysqli` no está recomendado su uso.

La sintaxis es: `mysql_result($result, $numeroDeFila, "identificadorDeLaColumna");`

Esta función nos devuelve el contenido de la celda en la fila y columna indicadas en forma de string (cadena de caracteres) en caso de éxito, o FALSE en caso de error.

\$result es el resultado obtenido previamente con una invocación a la función mysql_query.

numeroDeFila es un número de fila dentro del resultado obtenido teniendo en cuenta que los números de fila se cuentan empezando desde cero.

identificadorDeLaColumna puede ser el índice del campo (por ejemplo 0), el nombre del campo (por ejemplo ciudad), o el nombre de la tabla punto nombre del campo (por ejemplo agenda.ciudad)

Finalmente, se puede especificar el nombre del campo que queremos extraer dentro de la fila (esto es opcional). Esta función la entenderemos mejor con el ejemplo que veremos a continuación.

EJEMPLO DE USO DE LAS FUNCIONES ANTERIORES. ejemplo1.php

Recuerda también que debes cambiar y poner tus propios valores en las funciones mysqli_connect y mysqli_select_db.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf8" />
<title>Prueba de SELECT y MySQL</title>
</head>
<body>
<?php
// Se conecta al SGBD
if(!($link = mysqli_connect("localhost", "root", "")))
    die("Error: No se pudo conectar");
else{
    // Selecciona la base de datos
    if(!mysqli_select_db($link,"empresa"))
        die("Error: No existe la base de datos");
    else{
        $tildes = $link->query("SET NAMES 'utf8'"); //Para que se muestren las tildes
```

```
$result = mysqli_query($link, "SELECT * FROM agenda");  
  
mysqli_data_seek ($result, 0);  
  
$extraido= mysqli_fetch_array($result);  
  
echo "- Nombre: ".$extraido['nombre']."<br/>";  
  
echo "- Apellidos: ".$extraido['apellidos']."<br/>";  
  
echo "- Dir.: ".$extraido['direccion']."<br/>";  
  
echo "- Tfno: ".$extraido['telefono']."<br/>";  
  
echo "- Edad: ".$extraido['edad']."<br/>";  
  
mysqli_free_result($result);  
  
mysqli_close($link);  
  
}  
  
?  
  
</body>  
</html>
```

```
$result = mysqli_query($link, "SELECT * FROM agenda");
```

Con esta línea seleccionamos todos los registros (filas) y campos (columnas) existentes en nuestra base de datos. Para seleccionar todos los registros escribimos SELECT *. A continuación indicamos la tabla de la cual queremos obtener dichos registros y esto lo expresamos con la sintaxis FROM agenda, siendo agenda el nombre de la tabla en la base de datos. En la llamada a la función incluimos el identificador de conexión que hemos obtenido previamente.

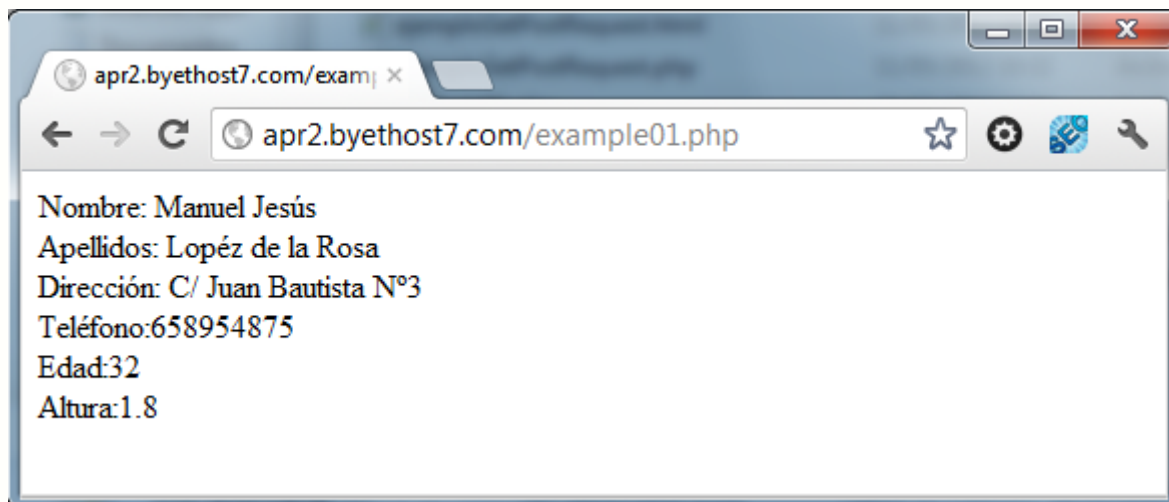
Con la sentencia mysqli_data_seek (\$result, 0); nos posicionamos en la primera fila (fila cero) de los resultados de la consulta.

Con la sentencia \$extraido= mysqli_fetch_array(\$result); introducimos en un array al que denominamos \$extraido los datos de la fila cero obtenidos de la consulta.

Finalmente mostramos por pantalla los resultados.

En la versión que no usa mysqli (no recomendada) con la sentencia mysql_result(\$result, 0, "nombre") obtenemos el campo "nombre" del primer registro (registro 0) del resultado de la consulta que hemos almacenado previamente en \$result.

El resultado de invocar el archivo que hemos creado será que se nos muestre por pantalla el nombre, apellidos, dirección, teléfono, edad y altura almacenados en la primera fila de nuestra base de datos. Algo similar a esto:



Resumiendo lo que hemos hecho, hemos usado la función `mysqli_connect()`, que abre una conexión con el servidor MySQL en el Host especificado.

Si la conexión ha tenido éxito, la función `mysqli_connect()` devuelve un identificador de dicha conexión que es almacenado en la variable `$link`. Si no tuviera éxito, devuelve 0 (FALSE).

Con `mysqli_select_db()` PHP le hemos dicho al servidor que en la conexión `$link` nos queremos conectar a la base de datos indicada.

La siguiente función `mysqli_query()`, usando el identificador de la conexión (`$link`), envía una instrucción SQL al servidor MySQL para que éste la procese. El resultado de ésta operación es almacenado en la variable `$result`.

Finalmente, elegimos la fila de resultados cero con `mysqli_data_seek($result, 0)`; y guardamos los resultados de esa fila en un array con `$extraido = mysqli_fetch_array($result)`; . Si quisiéramos mostrar los siguientes registros tendríamos que incluir instrucciones con los números 1, 2, ...

La sentencia `mysqli_free_result($result)`; libera la conexión establecida con la base de datos.

La sentencia `mysqli_close($link)`; cierra la conexión con la base de datos.

EJERCICIOS

1. Modifica el ejemplo y muestra todos los registros de la tabla Agenda
2. Utilizando la tabla <<ciudades>>:

id	ciudad	pais	habitantes	superficie	tieneMetro
1	México D.F.	México	555666	23434.34	1
2	Barcelona	España	444333	1111.11	0
3	Buenos Aires	Argentina	888111	333.33	1
4	Medellín	Colombia	999222	888.88	0
5	Lima	Perú	999111	222.22	0
6	Caracas	Venezuela	111222	111.11	1
7	Santiago	Chile	777666	222.22	1
8	Antigua	Guatemala	444222	877.33	0
9	Quito	Ecuador	333111	999.11	1
10	La Habana	Cuba	111222	333.11	0

Crea un archivo php donde se realice una consulta a la base de datos para extraer todos los datos de la tabla y a continuación muestra por pantalla sólo los valores de país y ciudad de cada fila. Ejemplo de resultados a obtener:

Los resultados de la consulta a la base de datos son los siguientes:

País: México --- Ciudad: México D.F.

País: España --- Ciudad: Barcelona

País: Argentina --- Ciudad: Buenos Air

País: Cuba --- Ciudad: La Habana