



APRENDERAPROGRAMAR.COM

CONSULTAS BÁSICAS A
BASE DE DATOS CON PHP.
mysqli_connect, select_db,
query, data_seek,
fetch_array, free_result.
EJEMPLOS. (CU00841B)

Sección: Cursos

Categoría: Tutorial básico del programador web: PHP desde cero

Fecha revisión: 2029

Resumen: Entrega nº41 del Tutorial básico "PHP desde cero".

Autor: Enrique González Gutiérrez y César Krall

USO DE PHP CON MySQL. CONSULTAS BÁSICAS

Estas próximas entregas no pretenden ser un manual de MySQL aunque sí explicaremos como realizar consultas básicas desde PHP. Veremos algunas funciones básicas para acceder a una base de datos MySQL desde PHP para recuperar información, insertar información o borrar información.



Describiremos una serie de funciones básicas que nos servirán para ilustrar cómo utilizar PHP para realizar consultas a una base de datos: añadir nuevos registros, actualizar registros, borrar registros, etc. No vamos a estudiar en profundidad las bases de datos ni el lenguaje SQL. Vamos a tratar de ser prácticos, conocer algunas funciones para acceder a bases de datos con PHP y aprender a usarlas a través de ejemplos.

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).

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

FUNCIÓN `mysqli_close` (`mysqli_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`.

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

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.

La sintaxis que vamos a emplear es:

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

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

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 = mysqli_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_ASSOC` (`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 N° 3	658954875	32	1.80
María	Manzano Cabezas	C/Arco del triunfo N° 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?

Para ello, usando la sintaxis mysqli escribiríamos esto:

```
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.

Escribe este código en un editor y guárdalo en un archivo de nombre ejemplo1.php (recuerda que antes debes haber creado la base de datos y creado datos de ejemplo como hemos visto anteriormente). Recuerda también que debes cambiar y poner tus propios valores en las funciones `mysqli_connect` y `mysqli_select_db`.

```
<html><head><meta charset="utf-8"></head>
<body>
<?php
$link = mysqli_connect("sql203.byethost7.com", "b7_10356956", "*****");
mysqli_select_db($link, " b7_10356956_mibasededatos");
$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 "- Dirección: ".$extraido['direccion']."<br/>";
echo "- Teléfono: ".$extraido['telefono']."<br/>";
echo "- Edad: ".$extraido['edad']."<br/>";
mysqli_free_result($result);
mysqli_close($link);
?>
</body>
</html>
```

El código con la sintaxis que no usa `mysqli` (no recomendado) sería:

```
<html>
  <body>
    <?php
      $link = mysql_connect("sql203.byethost7.com", "b7_10356956", "*****");
      mysql_select_db("b7_10356956_mibasededatos", $link);
      mysql_query("SET NAMES 'utf8'"); //Para que se muestren las tildes
      $result = mysql_query("SELECT * FROM agenda", $link);
      echo "Nombre: ".mysql_result($result, 0, "nombre")."<br>";
      echo "Apellidos: ".mysql_result($result, 0, "apellidos")."<br>";
      echo "Dirección: ".mysql_result($result, 0, "direccion")."<br>";
      echo "Teléfono: ".mysql_result($result, 0, "telefono")."<br>";
      echo "Edad: ".mysql_result($result, 0, "edad")."<br>";
      echo "Altura: ".mysql_result($result, 0, "altura")."<br>";
      mysql_free_result($result);
      mysql_close($link);
    ?>
  </body>
</html>
```

En este código lo que hemos hecho es usar las funciones vistas anteriormente y la base de datos y tablas que creamos en anteriores apartados del curso. Con todo ello lo que hacemos es realizar una consulta a la base de datos y mostrar los resultados por pantalla.

Interpretemos las líneas que hemos escrito:

```
$link = mysqli_connect("sql203.byethost7.com", "b7_10356956", "*****");
```

Con esta línea introducimos el identificador de conexión en una variable denominada \$link. Invocamos la función `mysqli_connect` pasándole como parámetros el nombre del hosting, el nombre de usuario y la contraseña. Estos datos debemos conocerlos (o consultarlos) previamente.

```
mysqli_select_db($link, "b7_10356956_mibasededatos");
```

Con esta línea seleccionamos la base de datos pasándole a la función como parámetros el identificador de conexión (que hemos obtenido y guardado en una variable previamente) y el nombre de base de datos.

```
$result = mysqli_query($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.

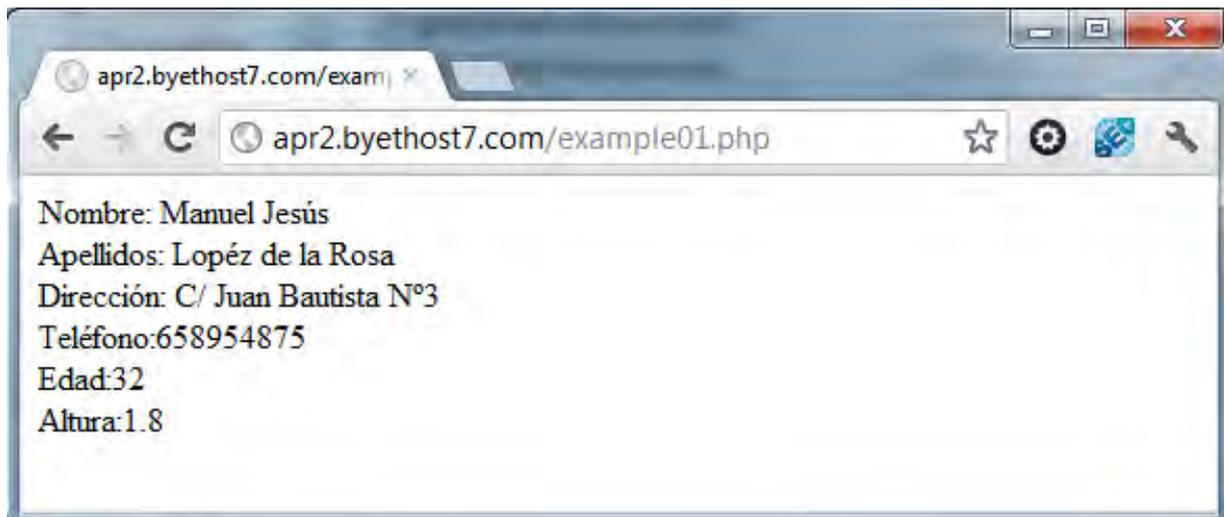
Con la sentencia `mysqli_data_seek ($result, 0)`; nos posicionamos en la primera fila (fila cero) de los resultados de la consulta.

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 (en este ejemplo la máquina donde está alojada el servidor MySQL es `sql203.byethost7.com`, pero tú tendrás que introducir tu propio dato). También hemos especificado un usuario (`b7_10356956` en este ejemplo, pero tú tendrás que introducir tu propio dato), y un password para el usuario indicado (tendrás que escribir tu password de usuario de base de datos).

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.

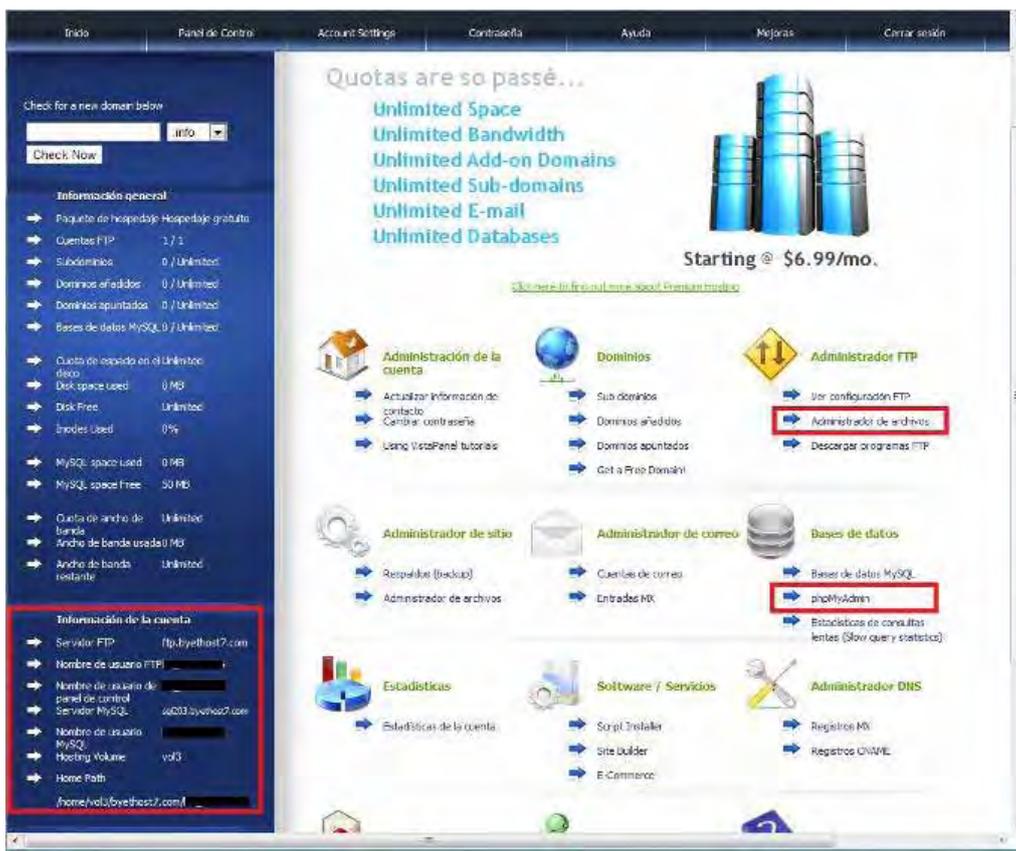
La siguiente función `mysqli_query()`, usando el identificador de la conexión (`$link`), envía una instrucción SQL al servidor MySQL para que éste la procese. El resultado de ésta operación es almacenado en la variable `$result`.

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.

*Nota: Si tenemos algún problema en localizar la información de nuestro proveedor (nombre de base de datos, usuario, etc.). recordar que dicha información normalmente está disponible en el panel de control del servidor (cpanel o similar) tal y como vimos en entregas anteriores.



En el panel de control que estamos usando nosotros para este ejemplo, si bajamos con el scroll hacia abajo, después de la información ftp que nos aparece marcada con un recuadro rojo, aparecerá la información relativa al servidor de base de datos. Si estás utilizando otro panel de control, tendrás que buscar el lugar específico donde figura esta información.

EJERCICIO

Utilizando la tabla <<ciudades>> que hemos creado para el ejercicio de la entrega anterior del curso, cuyo contenido era:

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 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 Aires

...

...

País: Cuba --- Ciudad: La Habana

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros [aprenderaprogramar.com](http://www.aprenderaprogramar.com).

Próxima entrega: CU00842B

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193