

INTRODUCCIÓN A LA PROGRAMACIÓN CON PHP

PRÁCTICA Nº 6

MySQL

¿Qué es una base de datos relacional?

Una base de datos es una estructura ordenada de información.

Ejemplo: lista de la compra, información de una empresa.

Para poder acceder a éstos datos y al equipo que los contiene se necesita un gestor/servidor de bases de datos como por ejemplo *MySQL*.

¿Qué es una Base de Datos Relacional?

Una **base de datos relacional**, es una aplicación capaz de almacenar grandes cantidades de información y de ofrecerla de manera rápida con capacidad para combinar y ordenar los datos, ajustándose a los requerimientos del usuario.

¿Qué es una Base de Datos Relacional?

SQL es el acrónimo de *Structured Query Language*, es decir Lenguaje estructurado de consultas.

Este es un estándar que utilizan las bases de datos relacionales para insertar, actualizar, eliminar y recuperar información.

Existen muchas bases de datos relacionales, las más difundidas son MySQL, Oracle, SQL Server y PostgreSQL. Cada una de ellas es compatible con el estándar SQL, aunque cada una de ellas tiene un juego de instrucciones propias.

¿Qué es una Base de Datos Relacional?

La estructura interna de una base de datos relacional está basada en tablas y a su vez en filas o registros, por ejemplo:

Tabla: **Productos**

id	producto	precio
1	Pan	0.45
2	Leche	0.80
3	Aceite	1.40

Los datos de la tabla se leen por filas, por ejemplo sabemos que el producto pan tiene un precio de 0.45.

Tipos de datos en MySQL

Igual que veíamos en PHP y en otros lenguajes de programación en MySQL hay varios tipos de datos:

TINYINT, SMALLINT, MEDIUMINT, INT y BIGINT

Números enteros. Se diferencian en el rango, es decir, en el mayor número representable en un campo.

Tipos de datos	Bytes	Valor mínimo	Valor máximo
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INT o INTEGER	4	-2147483648	2147483647
BIGINT	8	-9223372036854775808	9223372036854775807

Tipos de datos en MySQL

FLOAT y **DOUBLE** Número de punto flotante (decimal).

DATE Un campo de fecha con formato AAAA-MM-DD.

DATETIME Campo de fecha y hora.

Su formato es AAAA-MM-DD HH:MM:SS.

Referencia donde encontrar más información:

<https://disenowebakus.net/tipos-de-datos-mysql.php>

Tipos de datos en MySQL

CHAR Campo alfanumérico.

VARCHAR Campo alfanumérico de longitud variable.

BLOB y **TEXT** Campos alfanuméricos largos.

MEDIUMBLOB y **MEDIUMTEXT** Análogo a los campos BLOB y TEXT. Longitud máxima 16777215 caracteres.

LONGBLOB y **LONGTEXT** Análogo a los campos BLOB y TEXT. Longitud máxima 4294967295 caracteres.

Tipos de datos en MySQL

En todos los tipos de datos (menos en las fechas) es necesario especificar el tamaño del campo.

Para datos numéricos es necesario especificar si es con signo o sin signo, la diferencia de esto último es que los campos sin signo **UNSIGNED** tienen el doble de rango que los que usan signo **SIGNED**

Manejo de Base de Datos

FUNCIÓN: `mysqli_connect` (`mysqli_connect`)

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

La sintaxis de esta función es:

```
$nombreConexion = mysqli_connect($hostname ,  
                                $nombreUsuario , $contraseña);
```

Para servidores no actualizados (versiones anteriores a PHP 5) usar `mysql_connect` en lugar de `mysqli_connect`

Manejo de Base de Datos

La función devuelve una conexión almacenada en la variable `$nombreConexion`, o `FALSE` en caso de error.

Los parámetros 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 root).

\$contraseña es la contraseña de acceso para el usuario de base de datos, contraseña que habremos especificado al crear la base de datos.

Manejo de Base de Datos

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 es:

```
mysqli_select_db ($nombreConexión,  
$nombreBaseDatos);
```

Para servidores no actualizados debe usar
**mysql_select_db (\$nombreBaseDatos,
\$nombreConexión);**

Manejo de Base de Datos

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`.

Manejo de Base de Datos

FUNCIÓN `mysqli_query` (`mysqli_query`)

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

La sintaxis es:

```
$result = mysqli_query($nombreConexion,  
                        "Consulta aquí"); ó  
mysqli_query($nombreConexion, $query);
```

Para servidores no actualizados deben usar

```
$result = mysqli_query("Consulta aquí",  
$nombreConexion);
```

Manejo de Base de Datos

Donde "Consulta aquí" se refiere a una consulta realizada utilizando el lenguaje SQL de consultas a bases de datos.

Ejemplo: una consulta puede ser:

`"SELECT * FROM agenda",`

que podría traducirse como “selecciona todas las filas de la tabla agenda”.

Manejo de Base de Datos

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”.

DELETE significa “BORRAR”.

SELECT significa “seleccionar”.

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

Manejo de Base de Datos

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 es:

mysqli_close(\$nombreConexión);

Para servidores no actualizados **usar**
mysql_close

Manejo de Base de Datos

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`.

Manejo de Base de Datos

RECUPERAR UN RESULTADO:

DATA_SEEK, FETCH_ARRAY (MYSQL_RESULT)

Una sentencia de consulta normalmente devuelve “un conjunto de resultados” en una variable denominada \$result.

Manejo de Base de Datos

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

¿Cómo extraer el dato de una celda concreta? Por ejemplo ¿el nombre de la primera fila?

Manejo de Base de Datos

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();
```

Manejo de Base de Datos

Se están realizando 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).

Manejo de Base de Datos

- 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).

Manejo de Base de Datos

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.

Manejo de Base de Datos

FUNCIÓN:

Sintaxis:

array mysqli_fetch_assoc (\$result)

Retorna un array asociativo correspondiente a la fila obtenida o NULL si no hubiera más filas.

Nota: Los nombres de los campos devueltos por esta función son sensibles a mayúsculas y minúsculas.

Parámetro **\$result**: un conjunto de identificadores de resultados devuelto por [mysqli_query\(\)](#).

Manejo de Base de Datos

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");
```

Manejo de Base de Datos

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`.

Manejo de Base de Datos

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).

Manejo de Base de Datos

or die()

Captura el error.

string mysqli_error (\$link)

Devuelve el último mensaje de error para la llamada más reciente a una función de MySQLi que puede haberse ejecutado correctamente o haber fallado.

Parámetro: \$link : un identificador de enlace devuelto por [mysqli_connect\(\)](#)

Valores devueltos: una cadena que describe el error. Una cadena vacía si no ha ocurrido error alguno.

Manejo de Base de Datos

FUNCIÓN:

int mysqli_num_rows (\$result)

Retorna el número de filas del resultado.

Parámetro: **\$result** es un conjunto de identificadores de resultados devuelto por [mysqli_query\(\)](#).

Este comando es valido para las sentencias SELECT.

Manejo de Base de Datos

La sentencia ***SELECT*** nos permite consultar los datos almacenados en una tabla de la base de datos.

LIMIT :

La cláusula **LIMIT** puede ser usada para acotar el número de filas devueltas por la sentencia **SELECT**.

Manejo de Base de Datos

LIMIT tiene dos argumentos, el primero es el registro por el que empezar los resultados y el segundo el número de resultados

Si el segundo argumento supera la cantidad de registros de la tabla, se limita hasta el último registro.

Manejo de Base de Datos

Ejemplo:

select * from libros limit 0,4;

Muestra los primeros 4 registros, 0,1,2 y 3.

Si tipeamos:

select * from libros limit 5,4;

recuperamos 4 registros, desde el 5 al 8.

Si se coloca un solo argumento, indica el máximo número de registros a retornar, comenzando desde 0.

Ejemplo:

- **select * from libros limit 8;** Muestra los primeros 8 registros.

Bibliografía:

Información extraída de:

<http://ar.php.net/manual/es/langref.php>

http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=612:php-consultas-mysql-mysqliconnect-selectdb-query-fetcharray-freeresult-close-ejemplos-cu00841b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193

