Conexión, creación de bases de datos e táboas no sistema xestor de BD

As extensións PHP para o acceso a bases de datos SQL

Unha API (Interface de Programación de Aplicacións) define as clases, métodos, funcións e variables que a aplicación necesita empregar para realizar unha tarefa concreta.

As APIs poden ser procedimentais ou orientadas a obxectos. Unha API procedimental usará funcións para realizar tarefas, mentres que unha API orientada a obxectos instanciará clases e chamará aos métodos sobre os obxectos resultantes.

PHP soporta máis de 15 sistemas xestores de bases de datos e nós imos traballar con MySQL. MySQL é un xestor de bases de datos relacionais de código aberto baixo licenza GNU GPL.

No caso de aplicacións de PHP que necesiten acceder a unha base de datos, as APIs necesarias son extensións de PHP. O máis habitual é elixir entre MySQLi (extensión nativa, MySQL improved) e PDO (PHP Data Object). A extensión MySQLi pode utilizarse empregando o paradigma de Programación Orientada a Obxectos (POO) ou o procedemental. Mentres PDO ofrece un conxunto común de funcións, as extensións nativas normalmente ofrecen máis potencia (acceso a funcións específicas de cada xestor de base de datos) e nalgúns casos tamén maior velocidade.

Procedemento de acceso a bases de datos MySQL empregando MySQLi

Estos son os pasos para acceder ás bases de datos empregando MySQLi:

- Conexión ao xestor da base de datos.
- Selección da base de datos coa que se desexa traballar.
- Envío dun comando a ser executado na base de datos.
- Solicitude dun dos rexistros obtidos como resultado dunha consulta SQL.
- Desconexión da base de datos.

Conexión e desconexión

Conexión a un sistema xestor de base de datos

Para poder acceder aos datos dunha base de datos, previamente temos que establecer unha conexión co servidor MySQL. Para elo, empregaremos a función `mysqli_connect()`, que pode recibir os seguintes parámetros:

- O nome ou dirección IP do *servidor* MySQL ao que conectarse.
- Un nome de *usuario* con permisos para establecer a conexión.
- O *contrasinal* do usuario.

Para conectarse ao xestor de BD unicamente se precisan as tres primeiras.

Sintaxe da conexión á BD

```
mysqli_connect(servidor, usuario, clave)
```

Esta función devolve un obxecto que representa a conexión co xestor MySQL e que se precisará para realizar operacións futuras con dito xestor.

Exemplo de conexión á BD

```
$conexionDB = mysqli_connect('localhost', 'admin', 'contrasinalFalso');
```

Comprobación de erros

É imprescindible verificar que a conexión se estableceu correctamente antes de realizar ningunha operación sobre a base de datos, xa que, de non se establecer conexión, ningunha outra operación poderá ser realizada. No caso de que exista un erro, a función `mysqli_connect()` devolve `FALSE` en lugar do identificador da conexión. Para determinar o tipo de erro que se produciu, existen as seguintes funcións:

Comprobación de erros na conexión	
`mysqli_connect_errno()`	Devolve o número de erro xerado polo último intento de conexión, ou cero se non se produciu ningún erro.
`mysqli_connect_error()`	Devolve a mensaxe de erro xerada polo último intento de conexión, ou NULL se non se produciu ningún erro.

Exemplos de comprobación de erros na conexión

```
<?php
   $bd = @mysqli_connect('localhost', 'admin', 'contrasinalFalso');

   if (!$bd) {
     echo('Erro número '.mysqli_connect_errno().' ao establecer a
conexión: '.mysqli_connect_error().'.');
   }
?>
-----------------------------------------------------------------------
<?php
$link = mysqli_connect("localhost", "my_user", "my_password", "world");

/* check connection */
if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}
¿>
```

O símbolo `@` diante dunha función emprégase para evitar que se amosen as mensaxes por defecto de erros e advertencias de PHP e así que sexa o programador quen os xestione.

Desconexión dun sistema xestor de bases de datos

Para pechar unha conexión ao xestor de BD empregamos a función `mysqli_close()`, á que teremos que pasarlle o obxecto que obtivemos ao establecer a conexión.

Sintaxe da desconexión á BD

```
mysqli_close(obxectoConexion)
```

Esta función devolve TRUE se a desconexión foi exitosa e FALSE en caso contrario.

Exemplo de conexión á BD

```
mysqli_close($conexionBD);
```

Comprobación de erros

No caso de que se produza un erro na desconexión, a función `mysqli_close()` devolverá o valor FALSE, se isto sucede, podemos saber que tipo de erro se produciu empregando as funcións xerais de comprobación de erros de MySQLi.

Comprobación de erros na desconexión

`mysqli_errno($conexionBD)`	Devolve o número de erro xerado na base de datos, ou cero se non se produciu ningún erro.
`mysqli_error($conexionBD)`	Devolve a última mensaxe de erro xerada na base de datos, ou NULL se non se produciu ningún erro.

Exemplo de comprobación de erros na conexión

```
<?php
   if (!mysqli_close($conexionBD)) {
         echo('Erro número ' . mysqli_errno($conexionBD) . ' ao pechar a
    conexión: ' . mysqli_error($conexionBD) . '.');
   }
?>
```

Selección dunha base de datos

A conexión a unha base de datos existente pode facerse na propia conexión ao xestor de Bases de Datos ou nunha sentenza independente.

Sintaxe da conexión a una BD existente na conexión

```
mysqli_connect(servidor, usuario, clave, nomeBD)
```

Sintaxe da selección de BD unha vez conectado

```
mysqli_select_db(conexionBD, nomeBD);
```

Devolve TRUE se se seleccionou correctamente e FALSE en caso de erro.

A función `mysqli_select_db()` tamén a podemos usar para cambiar a base de datos coa que imos traballar.

Exemplo de selección de BD

```
mysqli_select_db($conexionBD, 'empregados');
```

Comprobación de erros

No caso de que se produza un erro na desconexión, a función `mysqli_select_db()` devolverá o valor FALSE, se isto sucede, podemos saber que tipo de erro se produciu empregando as funcións xerais de comprobación de erros de MySQLi expostas anteriormente.

Realización de operacións

Para a realización de calquera operación SQL no sistema xestor, empregarase a función `mysqli_query()`, á que haberá que pasarlle a conexión á BD e unha cadea coa sentenza SQL da operación a realizar.

Sintaxe da realización de operacións

```
mysqli_query(conexionBD, operacion)
```

Devolve FALSE en caso de erro ou un obxecto co resultado se non houbo erro.

Nos sucesivos apartados se explica como empregar esta sentenza para crear bases de datos e táboas, pero o seu uso sería o mesmo para outras operacións en SQL simplemente empregando a consulta SQL apropiada: eliminación de bases de datos (`DROP DATABASE nomeBD`), eliminación de táboas (`DROP TABLE nomeTaboa`), etc.

Creación dunha nova base de datos

Para crear unha nova BD no sistema xestor, debemos empregar a sentenza SQL `CREATE DATABASE`, como se pode ver no seguinte exemplo:

Exemplo de creación dunha nova BD

```
if (mysqli_query($conexionBD, 'CREATE DATABASE novaBD') === TRUE) {
    echo('Creouse correctamente a BD chamada novaBD.');
} else {
    echo('Erro na creación da BD: ' . mysqli_error($conexionBD));
}
```

No que se pode ver que se comproba se se produciu algún erro na operación. Para obter máis detalles dese posible erro, empréganse as funcións xerais de comprobación de erros de MySQLi expostas anteriormente.

É importante salientar que para poder crear una nova BD debe empregarse un usuario que teña privilexios de creación (`CREATE`).

Creación de táboas nunha base de datos

Para crear unha nova táboa na base de datos empregamos a sentenza SQL `CREATE TABLE` indicando, segundo a sintaxe SQL, as columnas que conterá e os seus tipos de datos.

Exemplo de creación dunha nova táboa

```
$sentenzaSQL = 'CREATE TABLE alumnos (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
nome VARCHAR(30) NOT NULL,
apelido1 VARCHAR(30) NOT NULL,
apelido2 VARCHAR(30) NOT NULL,
email VARCHAR(50)
```

```
)';

if (mysqli_query($conexionBD, $sentenzaSQL)) {
    echo('Táboa alumnos creada con éxito.');
} else {
    echo('Erro na creación da táboa: ' . mysqli_error($conexionBD));
}
```

Tarefas

Tarefa 1. Conexión e desconexión a unha base de datos de exemplo

Esta tarefa consiste na implementación dun pequeno programa en PHP que se conecte ao servidor MySQL local e escolla a base de datos chamada `daw`. Posteriormente debe desconectarse. Deben facerse comprobacións de erros cando sexa preciso.

Pódese probar que a detección de erros é correcta conectándose cun usuario ou a unha BD que non existe.

Tarefa 2. Creación dunha nova base de datos no sistema xestor de BD.

Esta tarefa consiste na implementación dun pequeno programa en PHP que se conecte ao servidor MySQL local e cree unha nova base de datos chamada `daw2`. Han de facerse comprobacións de erros cando sexa preciso.

Pódese probar que a detección de erros é correcta tratando de crear unha BD que xa existe.

Tarefa 3. Creación de táboas nunha base de datos existente

Esta tarefa consiste na implementación dun pequeno programa en PHP que se conecte ao servidor MySQL local e cree unha nova táboa (chamada `unidades`) na base de datos chamada `daw`, cos seguintes campos:

- Número da unidade (que será un enteiro sen signo e a súa clave).
- Nome da unidade (que será unha cadea de texto de 50 carácteres e será obrigatorio).
- Descrición da unidade (que será unha cadea de texto de 100 carácteres opcional).

  Deben facerse comprobacións de erros cando sexa preciso.

Pódese probar que a detección de erros é correcta tratando de crear unha táboa que xa existe.