

Guion para la Unidad 6: Bases de datos

Sitio: Centros - Granada
Curso: Desarrollo web en entorno servidor
Libro: Guion para la Unidad 6: Bases de datos

Imprimido por: Aguilera Aguilera, Javier
Día: miércoles, 22 de diciembre de 2021, 12:22

Tabla de contenidos

1. Bases de datos relacionales

- 1.1. Sentencias básicas MySQL
- 1.2. Herramientas de administración
- 1.3. Lenguaje de Manipulación de Datos
- 1.4. Acceso a la base de datos y CRUD
- 1.5. Ejemplo: diferencias entre bindParam y bindValue

2. Bases de datos no relacionales

1. Bases de datos relacionales

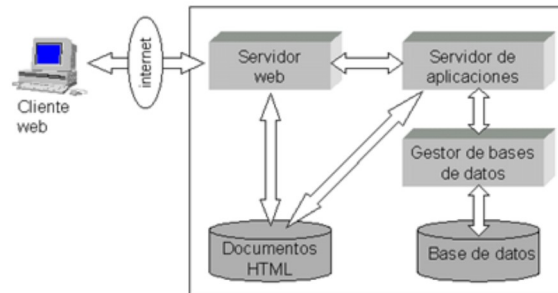
CONCEPTOS

Las bases de datos permiten almacenar de una forma estructurada y eficiente toda la información de un sitio web.

- Ventajas:
 - Proporcionan información actualizada
 - Facilitan la realización de búsquedas
 - Disminuyen los costes de mantenimiento
 - Implementan sistemas de control de acceso
 - Almacenan preferencias de los usuarios

Una de las aplicaciones más frecuentes de PHP es mostrar en una página web información extraída de la base de datos, o enviar sentencias al gestor de la base de datos para que elimine o actualice algunos registros.

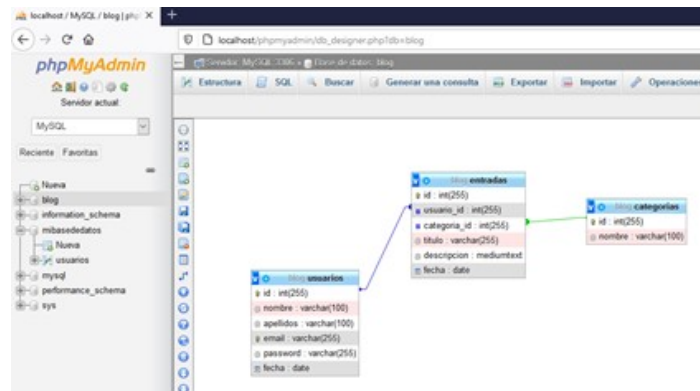
- Esquema básico de un sitio web soportado por bases de datos:



- Elementos de una BBDD: Tablas, consultas, informes y formularios

El **diseño de la BBDD** es fundamental para definir una estructura adecuada para nuestro sistema de información.

Existen numerosas herramientas para hacerlo, como por ejemplo MySQL Workbench o incluso, podemos usar directamente, el diseñador de phpMyAdmin para generar el diagrama:



Aquí tienes un video explicativo sobre las fases de diseño de una base de datos.

1.1. Sentencias básicas MySQL

Para hacer un breve repaso de las sentencias básicas las ejecutaremos directamente desde la consola de MySQL.

- Para usar la consola directamente desde Windows, localizamos la carpeta donde está instalado y escribimos:

```
mysql -h localhost -u root -p
```

Si tienes instalado WampServer dicha carpeta estará en la carpeta que tengas instalado Wamp, por ejemplo en mi caso es

`C:\wamp64\bin\mysql\mysql5.7.31\bin`. Abre el terminal de Windows localiza esta carpeta y ejecuta el comando anterior.

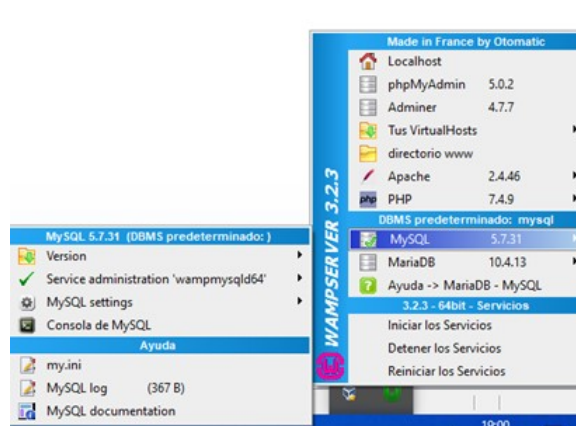
Si tienes una instalación en Xampp, busca la carpeta donde está la instalación e inícialo. Probablemente será `/opt/lampp/lampp`.

Para iniciarlo sudo `/opt/lampp/lampp startmysql` si no lo has hecho desde el entorno gráfico. A continuación para entrar en la consola:

accede a la carpeta `/opt/lampp/bin/` y ya puedes ejecutar el mismo comando para comenzar a trabajar con el usuario que tengas creado

Ejemplo : `/opt/lampp/bin/mysql -u root`

- Para utilizar la consola desde la instalación de WampServer accederemos al menú y seleccionamos MySQL o MariaDB.



Por defecto el usuario válido si no se ha creado ningún otro es root que es el que viene por defecto. Recién instalado no tiene contraseña.

- Para obtener la información sobre mi versión, usuario, puerto, etc., usaré el comando status:

```
mysql> status
-----
c:/wamp64/bin/mysql/mysql5.7.31/bin/mysql.exe Ver 14.14 Distrib 5.7.31, for Win64 (x86_64)

Connection id:          2
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.7.31 MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    latin1
Db characterset:        latin1
Client characterset:    cp850
Conn. characterset:     cp850
TCP port:               3306
Uptime:                 14 min 49 sec

Threads: 1  Questions: 5  Slow queries: 0  Opens: 108  Flush tables: 1  Open tables: 101  Qu
```

- Crear una base de datos:

```
mysql> create database mibasededatos;
Query OK, 1 row affected (0.03 sec)
```

- Mostrar las BBDD que tenemos:

```
mysql> show databases;
```

- Usar una BBDD:

```
mysql> use mibasededatos;
Database changed
```

- Ver las tablas de una BBDD:

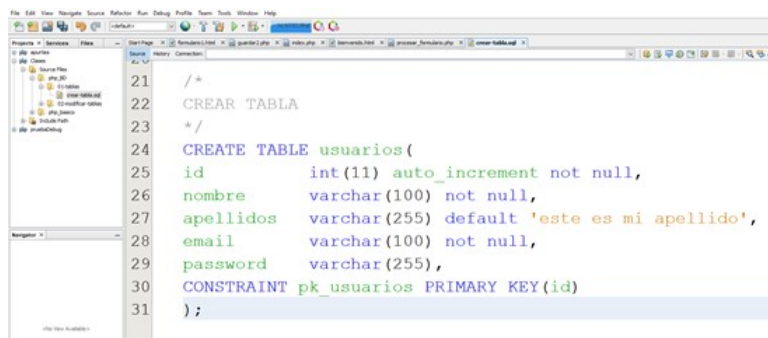
```
mysql> show tables;
```

- Borrar una BBDD:

```
mysql> drop database mibasededatos;
Query OK, 0 rows affected (0.11 sec)
```

También podemos trabajar escribiendo los comandos desde NetBeans, para ello hay que crear archivos SQL.

- Crear tablas (el nombre de la tabla en plural)



Copiamos este código en nuestra consola y ejecutamos.

- Mostrar las tablas de mi BBDD:

```
mysql> show tables;
+-----+
| Tables_in_mibasededatos |
+-----+
| usuarios                |
+-----+
1 row in set (0.00 sec)
```

- Mostrar la descripción de una tabla:

```
mysql> desc usuarios;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11) | NO | PRIMARY | NULL | 
```

- Borrar tablas

```
mysql> drop table usuarios;
```

- Algunas restricciones de integridad básicas, que nos proporcionan seguridad y permiten proteger nuestra BBDD de daños accidentales: not null, null, default, auto_increment, CONSTRAINT pk_usuarios PRIMARY KEY(id).

```
mysql> CREATE TABLE usuarios(
-> id          int(11) auto_increment not null,
-> nombre      varchar(100) not null,
-> apellidos   varchar(255) default 'los apellidos del sujeto',
-> email       varchar(100) not null,
-> password    varchar(255),
-> CONSTRAINT pk_usuarios PRIMARY KEY(id)
-> );
Query OK, 0 rows affected (0.09 sec)
```

- Algunas instrucciones para modificar tablas y columnas:

```
# RENOMBRAR UNA TABLA #
ALTER TABLE usuarios RENAME TO usuarios_renom;

# CAMBIAR NOMBRE DE UNA COLUMNA #
ALTER TABLE usuarios_renom CHANGE apellidos apellido varchar(100) null;

# MODIFICAR COLUMNA SIN CAMBIAR NOMBRE #
ALTER TABLE usuarios_renom MODIFY apellido char(40) not null;

# AÑADIR COLUMNA #
ALTER TABLE usuarios_renom ADD website varchar(100) null;

# AÑADIR RESTRICCIÓN A COLUMNA #
ALTER TABLE usuarios_renom ADD CONSTRAINT uq_email UNIQUE(email);

# BORRAR UNA COLUMNA #
ALTER TABLE usuarios_renom DROP website;
```

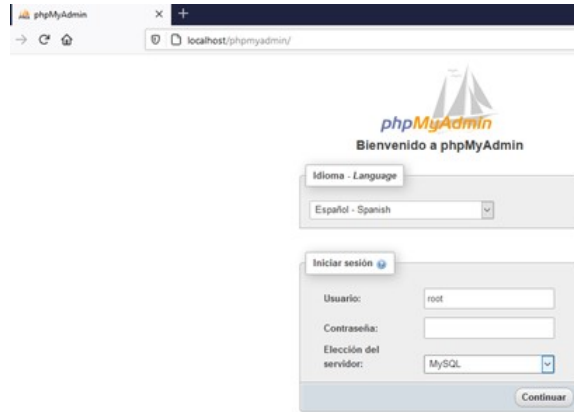

1.2. Herramientas de administración

(Si has borrado tablas o columnas en el apartado anterior, vuelve a crearlas para verlas con las herramientas visuales que se presenta a continuación)

1. phpMyAdmin

También podemos gestionar la BBDD mediante herramientas de administración visuales como por ejemplo phpMyAdmin.

En casi todos los servidores de aplicaciones de web locales tenemos esta herramienta. Lo podemos localizar, por ejemplo, en el menú de WampServer. Al iniciarlo nos solicitará las credenciales (root sin contraseña si no hemos realizado cambios durante la instalación). Seguidamente nos da la oportunidad de elegir MariaDB o MySQL.



Una vez dentro podrás ver tanto la estructura de la BBDD como de cada una de las tablas que hemos creado en la consola anteriormente.

También podemos ver los datos , insertar, modificar, eliminar, exportar o ejecutar SQL desde aquí sin necesidad de escribir comandos.

Recuerda que phpMyAdmin está en el directorio de trabajo de Apache.

2. MySQL integrado en el IDE.

1.3. Lenguaje de Manipulación de Datos

Este es un breve resumen de DML

1.4. Acceso a la base de datos y CRUD

En PHP hay drivers para manejar los sistemas gestores de BBDD más extendidos. También hay varias extensiones que proveen una capa de abstracción sobre la base de datos entre las que destaca PHP Data Objects (PDO).

PDO permite manejar diferentes bases de datos, con una interfaz común. Tiene la ventaja de que si se cambia la BBDD no hay que modificar el código.

El primer paso que debemos hacer es conectar desde PHP con la base de datos, para lo que disponemos de la librería **MySQLi**.

Esta librería se puede usar tanto con funciones clásicas de PHP como con POO.

- **Acceso a la BBDD mediante funciones clásicas de PHP** (no recomendado)
- **Acceso a la base de datos orientado a objetos**. Recuerda la extensión PDO es la biblioteca recomendada para acceder a bases de datos con PHP.

En estos documentos también encontrarás cómo hacer CRUD con PHP y MySQL con tablas relacionales.

CRUD es el acrónimo de las operaciones Create, Read, Update y Delete.

1.5. Ejemplo: diferencias entre bindParam y bindValue

```
// Ejemplo 1: Consulta
echo 'EJEMPLOS de las DIFERENCIAS entre bindParam y bindValue <br><br>';
// Ejemplo con bindParam:
echo 'Consulta realizada con bindParam: <br>';
$nom = 'Susana';
$s = $dbh->prepare('SELECT * FROM usuarios WHERE nombre = :nombre');
$s->setFetchMode(PDO::FETCH_ASSOC); // Especificamos el fetch mode antes de llamar a fetch()
$s->bindParam(':nombre', $nom);
$nom = 'Daniel';
$s->execute(); // se ejecutó con el valor WHERE nombre = 'Daniel'
// Mostramos los resultados

while ($row = $s->fetch()){
    echo "Nombre: {$row["nombre"]} <br>";
    echo "Clave: {$row["clave"]} <br>";
    echo "Rol:  {$row["rol"]} <br><br>";
}

// El mismo ejemplo con bindValue:
echo 'Consulta realizada con binValue: <br>';
$nom = 'Susana';
$s = $dbh->prepare('SELECT * FROM usuarios WHERE nombre = :nombre');
// $s->setFetchMode(PDO::FETCH_ASSOC);
$s->bindValue(':nombre', $nom);
$nom = 'Daniel';
$s->execute(); // se ejecutó con el valor WHERE nombre = 'Susana'
// Mostramos los resultados

while($row = $s->fetch(PDO::FETCH_OBJ)){
    echo "Nombre: " . $row->nombre . "<br>";
    echo "Clave: " . $row->clave . "<br>";
    echo "Rol: " . $row->rol . "<br>";
}
```

2. Bases de datos no relacionales

Las bases de datos no-SQL no siguen el modelo relacional.

En estas bases de datos los modelos de datos son, en general, más flexibles y es más fácil realizar cambios en el esquema a lo largo del proceso de desarrollo.

Veamos un ejemplo con MongoDB. (podéis profundizar sobre MongoDB en el curso <https://openwebinars.net/academia/portada/mongodb/>).

En MongoDB el elemento básico de almacenamiento es el documento. Los documentos se agrupan dentro de colecciones. Una base de datos cuenta con una o más colecciones. (los documentos serían filas y las colecciones tablas, pero aquí no todos los documentos dentro de una colección tienen que tener la misma estructura).

Los documentos en MongoDB siguen un formato similar a JSON

Para trabajar hay que descargar:

- MongoDB Community Server, la versión libre <https://docs.mongodb.com/manual/administration/install-community/>
- MongoDB Compass (aplicación para manejar la base de datos con interfaz gráfica) : <https://www.mongodb.com/products/compass>
- El driver de PHP para MongoDB. Para conectarse desde PHP tenemos que instalar el driver utilizando `composer`:

```
composer require mongodb/mongodb
```

Se recomienda intentar crear una base de datos y una colección para ver cómo funciona.

Para el ejemplo siguiente se ha usado la base de datos llamada libroservidor y una colección llamada usuarios. En dicho ejemplo se inserta documentos en la colección usuarios usando los métodos `insertOne()` e `insertMany()`

```
<?php
require 'vendor/autoload.php';
$cliente = new MongoDB\Client("mongodb://localhost:27017");
$db = $cliente->libroservidor;

try {
    $res = $db->usuarios->insertOne( [ 'nombre' => 'Ana', 'clave' => '1234', 'saldo' => 1000 ] )
    echo "Id del último registro: " . $res->getInsertedId() . "<br>";
    $res = $db->usuarios->insertMany( [
        [ 'nombre' => 'Paco', 'clave' => '1234', 'saldo' => 100],
        [ 'nombre' => 'Nuria', 'clave' => '1234', 'saldo' => 30],
    ] );
    echo "Documentos insertados: " . $res->getInsertedCount() . "<br>";
    print_r($res->getInsertedIds());
} catch (Exception $e) {
    print ($e);
}
```

Para las consultas se usan los métodos `find()` y `findOne()`. Ejemplo:

```
<?php
require 'vendor/autoload.php';
try {
    $cliente = new MongoDB\Client("mongodb://localhost:27017");
    $db = $cliente->libroservidor;
    // devuelve todos los usuarios
    echo "Todos los usuarios" . "<br>";
    $usuarios = $db->usuarios->find();
    foreach($usuarios as $usuario){
        var_dump($usuario);
    }
    // usuarios con nombre Ana
    echo "Usuarios con nombre 'Ana'" . "<br>";
    $usuarios = $db->usuarios->find(['nombre' => 'Ana']);
    foreach($usuarios as $usuario){
        var_dump($usuario);
    }
    // solo devuelve el primero que encuentre
    echo "Usuario con nombre 'Ana'" . "<br>";
    $ana = $db->usuarios->findOne(['nombre' => 'Ana']);
    var_dump($ana);
} catch (Exception $e) {
    print ($e);
}
```

Para actualizar update() y updateOne():

```
<?php
require 'vendor/autoload.php';
try {
    $cliente = new MongoDB\Client("mongodb://localhost:27017");
    $bd = $cliente->libroservidor;
    /* pone a 7000 el saldo del usuario con nombre 'Ana'*/
    $updateResult = $bd->usuarios->updateOne(
        [ 'nombre' => 'Ana' ],
        [ '$set' => [ 'saldo' => '7000' ] ]
    );

} catch (Exception $e) {
    print ($e);
}
```

Por último, para borrar se usan los métodos delete() y deleteOne()

```
<?php
require 'vendor/autoload.php';
try {
    $cliente = new MongoDB\Client("mongodb://localhost:27017");
    $bd = $cliente->libroservidor;
    /* pone a 7000 el saldo del usuario con nombre 'Ana'*/
    $updateResult = $bd->usuarios->deleteOne(
        [ 'nombre' => 'Paco' ] );
    echo "Documentos restantes después de borrar: " . $bd->usuarios->count();

} catch (Exception $e) {
    print ($e);
}
```