Ambientes no propietarios

MySQL y PHP

Edwin Salvador

02 de junio de 2015

Sesión 9

Contenido I

MySQL

- Probablemente el administrador de base de datos más popular para servidores web (más de 10 millones de instalaciones).
- SQL? Structured Query Language

Author	Title	Type	Year
Mark Twain	The Adventures of Tom Sawyer	Fiction	1876
Jane Austen	Pride and Prejudice	Fiction	1811
Charles Darwin	The Origin of Species	Nonfiction	1856
Charles Dickens	The Old Curiosity Shop	Fiction	1841
William Shakespeare	Romeo and Juliet	Play	1594

- Base de datos, tabla, filas, columnas, celdas o campo.

Contenido I

Accediendo a MySQL

- Linea de comandos
- Web (phpMyAdmin)
- Lenguaje de programación (PHP)

Línea de comandos

- Windows + R: cmd - enter
- `cd C:\xampp\mysql\bin\mysql -uroot`
- Por defecto el usuario es `root` y no tiene contraseña pero esto debe ser cambiado en un ambiente de producción.
- `SHOW databases;`

Línea de comandos

Accediendo a MySQL

- En Linux será similar: `mysql -u root -p`
- Lo más seguro es encontrar Linux en un servidor remoto. Es importante conocer la linea de comandos.
- Ciertos detalles variarán acorde a la configuración del servidor. Hablar con administrador del sistema.

Interactuando con MySQL

MySQL puede presentar los siguientes prompt

MySQL prompt	Meaning
`mysql>`	Ready and waiting for a command
`->`	Waiting for the next line of a command
`'>`	Waiting for the next line of a string started with a single quote
`">`	Waiting for the next line of a string started with a double quote
`` `> ``	Waiting for the next line of a string started with a back tick
`/*>`	Waiting for the next line of a comment started with `/*`

Cancelando un comando

Se debe utilizar el comando \c para cancelar un comando que ya no se quiere ejecutar. Ejemplo:

```
esto es algo "sin significado para MySQl" \c
```

Comandos más comunes

Command	Action
ALTER	Alter a database or table
BACKUP	Back up a table
\c	Cancel input
CREATE	Create a database
DELETE	Delete a row from a table
DESCRIBE	Describe a table's columns
DROP	Delete a database or table
EXIT (CTRL-C)	Exit
GRANT	Change user privileges
HELP (\h, \?)	Display help
INSERT	Insert data
LOCK	Lock table(s)
QUIT (\q)	Same as EXIT

Command	Action
RENAME	Rename a table
SHOW	List details about an object
SOURCE	Execute a file
STATUS (\s)	Display the current status
TRUNCATE	Empty a table
UNLOCK	Unlock table(s)
UPDATE	Update an existing record
USE	Use a database

Comandos

- SQL no es sensible a mayúsculas y minúsculas (case-insensitive). Se recomienda usar mayúsculas.
- Los nombre de tablas son case-sensitive en Linux y OS X pero no en Windows. Elegir un estilo y apegarse a ese para siempre. **Se recomienda minúsculas para las tablas.**

Comandos

Base de datos

- `CREATE DATABASE publicaciones;`
- `USE publicaciones;`

Usuarios: No es recomendable otorgar acceso de usuario `root` a todos los scripts PHP. Se debe crear un usuario con el que trabajaremos.

- `GRANT PRIVILEGES ON database.object TO 'username'@'hostname' IDENTIFIED BY 'password';`

Arguments	Meaning
`*.*`	All databases and all their objects
`database``.*`	Only the database called *`database`* and all its objects
`database.object`	Only the database called *`database`* and its object called *`object`*

- `GRANT ALL ON publicaciones.* TO 'chalo'@'localhost' IDENTIFIED BY '12345'`
- `quit`
- `mysql -uchalo -p`

Comandos

Crear tablas

- `USE publicaciones;`
- ```
  CREATE TABLE clasicos (
  autor VARCHAR(128),
  titulo VARCHAR(128),
  tipo VARCHAR(16),
  anio CHAR(4)) ENGINE MyISAM;
  ```
- `DESCRIBE clasicos;`
  ```

Tipos de datos

Consultar y traer un informe de los tipos de datos que soporta MySQL y cuando deben y no deben ser utilizados. **Modificando la tabla**

- `ALTER TABLE clasicos ADD id INT UNSIGNED NOT NULL AUTO_INCREMENT KEY;`

Creado la tabla con el campo `id`

- `CREATE TABLE clasicos ( autor VARCHAR(128), titulo VARCHAR(128), tipo VARCHAR(16), anio CHAR(4), id INT UNSIGNED NOT NULL AUTO_INCREMENT KEY) ENGINE MyISAM;`

Removiendo campo

- `ALTER TABLE clasicos DROP id;`

Insertando datos

```
INSERT INTO clasicos(autor, titulo, tipo, anio) VALUES('Mark
Twain','The Adventures of Tom Sawyer','Fiction','1876');
INSERT INTO clasicos(autor, titulo, tipo, anio) VALUES('Jane
Austen','Pride and Prejudice','Fiction','1811');
INSERT INTO clasicos(autor, titulo, tipo, anio) VALUES('Charles
Darwin','The Origin of Species','Non-Fiction','1856');
INSERT INTO clasicos(autor, titulo, tipo, anio) VALUES('Charles
Dickens','The Old Curiosity Shop','Fiction','1841');
INSERT INTO clasicos(autor, titulo, tipo, anio) VALUES('William
Shakespeare','Romeo and Juliet','Play','1594');


SELECT * FROM classics;
```

Renombrando tablas

- `ALTER TABLE clasicos RENAME pre1900;`
- `ALTER TABLE pre1900 RENAME clasicos;`

Contenido I

phpMyAdmin

- Iniciar servicio de MySQL en XAMPP
- `localhost/phpmyadmin`
- Listado de BDD (izquierda)
- Seleccionar publicaciones
- Tabla clasicos
- Crear BDD, añadir tablas, contenido, indices.

Contenido I

MySQL y PHP

- Podemos acceder a MySQL desde PHP y presentar los datos de las tablas en manera de formularios, listas, etc.
- El proceso para usar MySQL a través de PHP es:
  - Conectarse a MySQL y seleccionar una BDD.
  - Construir un string de consulta
  - Realizar la consulta
  - Tomar los resultados y presentarlos la página
  - Repetir del 2 al 4 hasta que todos los datos necesarios han sido recibidos
  - Desconectarse de MySQL

Conectar a MySQL

- Creamos la misma estructura de archivos que contiene el .htaccess.
- En el archivo config.php declaramos variables para el host, bdd, usuario, contraseña.
- Creamos un archivo query.php y nos conectamos a MySQL escribiendo:

```
require_once 'login.php';
$conn = new mysqli($host, $usuario, $cont, $db);
if ($conn->connect_error) die($conn->connect_error);
```

- El if verifica la conexión, si existe error "mata" el sistema, presenta el mensaje y sale.
- EL símbolo -> indica que connect_error es un método de `$conn`.
- La función `die` es solo para ambiente de desarrollo. En producción debemos presentar el mensaje adecuado.

Consultando la BDD

```php
$query = "SELECT * FROM clasicos";
$result = $conn->query($query);
if (!$result) die($conn->error);
```

Obteniendo los resultados: En el archivo "query.php" con el contenido:

```php
$rows = $result->num_rows;

for ($j = 0 ; $j < $rows ; ++$j)
{
  $result->data_seek($j);
  $row = $result->fetch_array(MYSQLI_ASSOC); //MYSQLI_NUM,
      MYSQLI_BOTH

  echo 'Author: ' . $row['autor'] . '<br>';
  echo 'Title: ' . $row['titulo'] . '<br>';
  echo 'Category: ' . $row['categoria'] . '<br>';
  echo 'Year: ' . $row['anio'] . '<br>';
}
$result->close();
$conn->close();
```

Contenido I

Seguridad

Contenido I

Ejercicio 1

- A través de phpMyAdmin añadir un campo `id` a la tabla `clasicos`. `INT UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY FIRST`
- Presentar todos las entradas de la tabla "clasicos" en una tabla HTML.
- Crear un formulario HTML que permita ingresar nuevos libros a la tabla "clasicos". **El campo id siempre debe ser `NULL` al ingresar un dato ya que MySQL le asignará un número automáticamente.**
- Añadir un campo a la tabla HTML que contenga un botón "Eliminar" que elimine el libro correspondiente.
- Cada uno de los botones eliminar deben ser un formulario que envía los datos mediante el post.
- Para saber que libro eliminar debemos usar un elemento HTML "**`hidden`**" que contenga el `id` del libro.
- Debido a que tenemos varios formularios en la página, debemos verificar el formulario que se ha enviado por POST. **Pista:** Se lo puede hacer verificando que botón se ha presionado.

Ejercicio 2

Escribir un script PHP que

- Se conecte a MySQL.
- Cree una tabla `Empleado` que contenga:
  - id
  - nombre
  - apellido
  - fecha de nacimiento
  - dirección
  - telefono
  - estado civil
  - departamento
- Debe tener un formulario HTML que permita ingresar la información de los empleados a la BDD.
- Debe presentar los datos en una tabla HTML.
- Debe permitir borrar un empleado de la lista.
- Debe permitir modificar la información de un empleado.

Ejercicio 2

- Para el ejercicio 2 también se contará con varios formularios en la página: ingresar, eliminar, modificar.
- Se debe controlar que formulario ha sido enviado para realizar la acción correspondiente. **Podemos utilizar el mismo formulario para ingresar datos y para modificar los datos.**
- Similar al ejercicio 1 podemos primero verificar si se desea:
  - Eliminar: en este caso debemos verificar nuestro elemento `hidden` con el `id` del empleado y ejecutar el comando SQL correspondiente para eliminar el registro.
  - Modificar: en este caso debemos verificar nuestro elemento `hidden` con el `id` del empleado. Una vez que tenemos el `id` debemos hacer un `SELECT` para obtener los datos del empleado y cargarlos en el formulario. Cuando presionemos el botón enviar debe realizar el proceso similar al de ingresar pero en lugar de hacer un `INSERT` debe hacer un `UPDATE`.
  - Ingresar: En este caso simplemente obtenemos los datos enviados en el formulario, los validamos y ejecutamos `INSERT`.