

Desarrollo de sitios web con PHP y MySQL



Tema 7: Acceso a bases de datos MySQL en
PHP

Ana María Fermoso García
afermosoga@upsa.es

Tema 7: Acceso a bases de datos MySQL en PHP

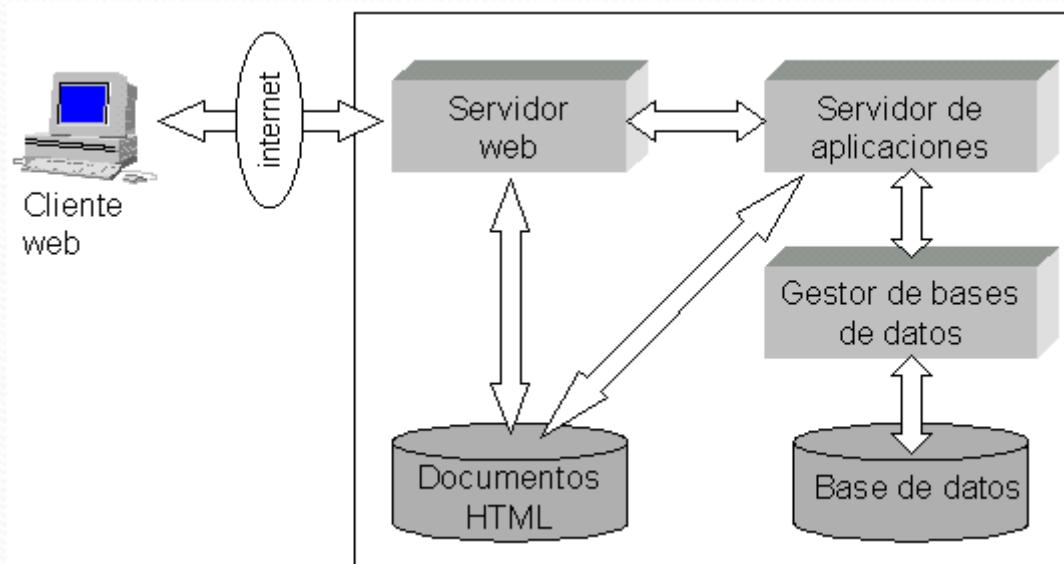
1. Bases de datos en la Web
2. Instalación y configuración de MySQL
3. MySQL
4. Herramientas de administración: phpMyAdmin
5. Lenguaje SQL
6. Funciones de PHP para el acceso a bases de datos MySQL
7. Ejercicio

Bases de datos en la Web

- n Las bases de datos permiten almacenar de una forma estructurada y eficiente toda la información de un sitio web
- n Ventajas
 - Proporcionar información actualizada
 - Facilitar la realización de búsquedas
 - Disminuir los costes de mantenimiento
 - Implementar sistemas de control de acceso
 - Almacenar preferencias de los usuarios

Bases de datos en la Web

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



Instalación y config. de MySQL

- Se puede descargar de: www.mysql.com
 - En nuestro caso se instala directamente con XAMPP

MySQL

- **Características de MySQL**
 - Modelo relacional, multiusuario
- **Tipos de datos**
 - Numéricos
 - tinyint, smallint, mediumint, int, integer, bigint
 - decimal, float, numeric
 - Fecha y hora
 - date, time, datetime, year, timestamp
 - Cadena
 - char, varchar
 - tinytext, text, mediumtext, longtext
 - tinyblob, blob, mediumblob, longblob
 - enum, set
 - Debe elegirse adecuadamente el tipo y el tamaño de cada campo

MySQL

- **Operadores**
 - Aritméticos
 - +, -, *, /
 - Comparación
 - =, !=, <=, <, >=, >, IS NULL, IS NOT NULL
 - Lógicos
 - not (!), and (&&), or (||), xor
- **Funciones**
 - Funciones de cadena
 - Funciones de comparación de cadenas
 - Funciones numéricas
 - Funciones de fecha y hora
 - Funciones de agregado

Herramientas de administración: phpMyAdmin

- **phpMyAdmin** es una herramienta para la administración del servidor de bases de datos MySQL
- Dispone de una interfaz gráfica y es de libre distribución
- Permite realizar todo tipo de operaciones sobre bases de datos:
 - crear, borrar y modificar tablas
 - consultar, insertar, modificar y eliminar datos
 - definir usuarios y asignar permisos
 - realizar copias de seguridad
 - etc
- Está escrita en php y se ejecuta desde el navegador
- Si está instalada en la carpeta `phpmyadmin`, se ejecuta escribiendo en la barra de direcciones del navegador la url
`http://localhost/phpmyadmin/`
- Puede administrar bases de datos locales y remotas
- También se instala directamente con XAMPP

phpMyAdmin

phpMyAdmin

Servidor: 127.0.0.1 > Base de datos: escuela

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutinas Eventos Disparadores Seguimiento Diseñador Columnas centrales

Filtros

Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
alumnos	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	latin1_spanish_ci	32.0 KB	-
asignaturas	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	latin1_spanish_ci	16.0 KB	-
matriculas	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	latin1_spanish_ci	32.0 KB	-
3 tablas	Número de filas	11	InnoDB	utf8mb4_general_ci	80.0 KB	0 B

↑ Seleccionar todo Para los elementos que están marcados:

Imprimir Diccionario de datos

Crear tabla

Nombre: Número de columnas: 4

Continuar

Lenguaje SQL

- SQL (*Structured Query Language*) es el lenguaje que se utiliza para comunicarse con la base de datos
- Procedimiento de comunicación con la base de datos:



Lenguaje SQL

- Las instrucciones más habituales son SELECT, INSERT, UPDATE, DELETE
- Veamos su sintaxis básica y algunos ejemplos de uso
- Para ello utilizaremos una tabla **noticias** con cinco campos: un identificador único de la noticia, el título de la noticia, el texto de la noticia, la categoría de la noticia y la fecha de publicación de la noticia

noticias
id
título
texto
categoría
fecha

Lenguaje SQL

- SELECT

Sintaxis:

```
SELECT expresión FROM tabla  
[WHERE condición]  
[ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC] , ...]  
[LIMIT [offset,] row_count | row_count OFFSET offset]
```

Ejemplo:

```
SELECT * from noticias WHERE fecha=CURDATE() LIMIT 10 ORDER BY fecha  
DESC
```

Obtiene las noticias del día con un tope máximo de 10, ordenadas de la más reciente a la más antigua

Lenguaje SQL

- INSERT

Sintaxis:

```
INSERT [INTO] nombre_tabla [(nombre_columna,...)]  
VALUES ((expresión | DEFAULT),...), (...),...  
INSERT [INTO] nombre_tabla  
SET nombre_columna=(expresión | DEFAULT), ...
```

Ejemplo:

```
INSERT INTO noticias (id, titulo, texto, categoria, fecha) VALUES  
    (37, "Nueva promoción en Nervión", "145 viviendas de lujo en  
    urbanización ajardinada situadas en un entorno privilegiado",  
    "promociones", CURDATE())
```

Inserta una noticia con los valores indicados

Lenguaje SQL

- UPDATE

Sintaxis:

```
UPDATE nombre_tabla  
SET nombre_columna1=expr1 [, nombre_columna2=expr2 ...]  
[WHERE condición]  
[ORDER BY ...]  
[LIMIT row_count]
```

Ejemplo:

```
UPDATE noticias SET categoria = "ofertas" WHERE id=37
```

Modifica la categoría de la noticia con id=37 de la tabla

Lenguaje SQL

- DELETE

Sintaxis:

```
DELETE FROM nombre_tabla  
[WHERE condición]  
[ORDER BY ...]  
[LIMIT row_count]
```

Ejemplo:

```
DELETE FROM noticias WHERE fecha < CURDATE() -10
```

Borra las noticias con más de 10 días de antigüedad

Funciones de PHP para el acceso a bases de datos MySQL

- Los pasos para acceder desde PHP a una base de datos son los siguientes:
 - Conectar con el servidor de bases de datos
 - Seleccionar una base de datos
 - Enviar la instrucción SQL a la base de datos
 - Obtener y procesar los resultados
 - Cerrar la conexión con el servidor de bases de datos

Acceso a bases de datos MySQL

- Las funciones concretas de MySQL que realizan estas operaciones son:
 - Conectar con el servidor de bases de datos:
 - **mysqli_connect()**
 - Seleccionar una base de datos:
 - **mysqli_select_db()**
 - Enviar la instrucción SQL a la base de datos:
 - **mysqli_query()**
 - Obtener y procesar los resultados:
 - **mysqli_num_rows()** y **mysqli_fetch_array()**
 - Cerrar la conexión con el servidor de bases de datos:
 - **mysqli_close()**

Acceso a bases de datos MySQL

- Conectar con el servidor de bases de datos: **mysqli_connect()**
 - Devuelve un identificador de la conexión en caso de éxito y false en caso contrario
- Sintaxis:

```
$conexion = mysqli_connect (servidor, username, password)
    or die      (mysqli_error())
```

- Ejemplo:

```
$conexion = mysqli_connect ('localhost', "cursophp", "")
    or die ("No se puede conectar con el servidor");
$conexion = mysqli_connect ("localhost", "cursophp-ad", "php.hph")
    or die ("No se puede conectar con el servidor");
```

Acceso a bases de datos MySQL

- Seleccionar una base de datos: **mysqli_select_db()**
 - Devuelve true en caso de éxito y false en caso contrario

Sintaxis:

```
mysqli_select_db($conexion, "nom_bd") or die  
    (mysqli_error());
```

- Ejemplo:

```
mysqli_select_db ($conexion, "academia")  
or die ("No se puede seleccionar la base de datos");
```

Acceso a bases de datos MySQL

- Enviar la instrucción SQL a la base de datos: **mysqli_query()**
 - Devuelve un identificador o true (dependiendo de la instrucción) si la instrucción se ejecuta correctamente y false en caso contrario
- Sintaxis:

```
$consulta = mysqli_query ($conexion, string instrucción);
```

- Ejemplo:

```
$consulta = mysqli_query ($conexion, "select * from noticias")  
or die ("Fallo en la consulta");
```

Acceso a bases de datos MySQL

- Obtener y procesar los resultados: **mysqli_num_rows()**, **mysqli_fetch_array()**
 - En el caso de que la instrucción enviada produzca unos resultados, **mysqli_query()** devuelve las filas de la tabla afectadas por la instrucción
 - **mysqli_num_rows()** devuelve el número de filas afectadas
 - Para obtener las distintas filas del resultado se utiliza la función **mysqli_fetch_array()**, que obtiene una fila del resultado en un array asociativo cada vez que se invoca
- **Sintaxis:**

```
$nfilas = mysqli_num_rows ($consulta);  
$fila = mysqli_fetch_array ($consulta);
```

Cuando el resultado se desea procesar como una matriz asociativa

```
$fila = mysqli_fetch_array ($consulta, MYSQL_ASSOC);  
foreach ($fila as $nom_colum=>$valor_colum)  
{  
    procesar $nom_colum,$valor_colum  
}
```

Acceso a bases de datos MySQL

- Ejemplo:

noticias				
	Título	Texto	Categoría	Fecha
1	Título 1	Texto 1	ofertas	05/02/2004
2	Título 2	Texto 2	promociones	05/02/2004
3	Título 3	Texto 3	promociones	04/02/2004
4	Título 4	Texto 4	costas	01/02/2004
5	Título 5	Texto 5	promociones	31/01/2004

Instrucción:

```
select * from noticias where categoria="promociones"
```

Acceso a bases de datos MySQL

- Ejemplo:

noticias				
	Título	Texto	Categoría	Fecha
1	Título 1	Texto 1	ofertas	05/02/2004
2	Título 2	Texto 2	promociones	05/02/2004
3	Título 3	Texto 3	promociones	04/02/2004
4	Título 4	Texto 4	costas	01/02/2004
5	Título 5	Texto 5	promociones	31/01/2004

← \$consulta
\$nfilas=3

Instrucción:

```
select * from noticias where categoria="promociones"
```

Acceso a bases de datos MySQL

- Obtención de las filas:

```
$nfilas = mysqli_num_rows ($consulta);
if ($nfilas > 0)
{
    for ($i=0; $i<$nfilas; $i++)
    {
        $fila = mysqli_fetch_array ($consulta);
        procesar fila i-ésima de los resultados
    }
}
```

Acceso a bases de datos MySQL

- Obtener los resultados: **mysqli_num_rows()**, **mysqli_fetch_array()**
 - Para acceder a un campo determinado de una fila se usa la siguiente sintaxis:

```
$fila["nombre_campo"] // por ser un array asociativo  
$fila[$i]           // $i=índice del campo desde 0
```

- Ejemplo:

```
for ($i=0; $i<$nfilas; $i++)  
{  
    $fila = mysqli_fetch_array ($consulta);  
    print "Título: " . $fila["titulo"];  
    print "Fecha: " . $fila["fecha"];  
}
```

Acceso a bases de datos MySQL

- Cerrar la conexión con el servidor de bases de datos:
mysqli_close()
- Sintaxis:

```
mysqli_close ($conexion);
```

- Ejemplo

```
mysqli_close ($conexion);
```

Ejercicio

Se trata de implementar el siguiente modelo de base de datos y a continuación implementar las funcionalidades que se plantean:

ALUMNOS (expediente, nombre, usuario, clave, f_nac, origen, email)

ASIGNATURAS (cod_asig, nom_asig, creditos, curso)

MATRICULAS (expediente, cod_asig, calificacion)

- **Alta de un alumno**, previa validación de datos
- **Acceso de un alumno con usuario y clave** y si es correcto, **mostrar datos del alumno** y dar paso a opciones de *modificar, eliminar, ver calificaciones o matricular alumno*.
Desde la misma pantalla se puede acceder a Alta alumno y en este caso no se introduce usuario y contraseña
- **Modificar un alumno**
- **Eliminar alumno**
- **Ver calificaciones** mostrar una tabla con el nombre de asignatura y nota, para cada una de las asignaturas en que está matriculado el alumno
- **Matricular un alumno** en asignatura en que previamente no estuviera matriculado, mostrando mediante lista despegable, solo estas asignaturas como posibles para su matrícula y calificación

Formato tablas ejercicio “Escuela”:

Alumnos

Nombre	Tipo
expediente	char(5)
nombre	varchar(20)
usuario	varchar(10)
clave	varchar(8)
f_nac	date
origen	varchar(20)
email	varchar(40)
observaciones	text

Asignaturas

Nombre	Tipo
cod_asig	int(10)
nom_asig	varchar(20)
creditos	int(10)
curso	int(10)

Matrículas

Nombre	Tipo
expediente	varchar(5)
cod_asig	int(10)
calificacion	decimal(10,2)