

PHP y MySQL

Aplicaciones Web/Sistemas Web



Juan Pavón Mestras
Dep. Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense Madrid

Material bajo licencia Creative Commons



PHP

Breve introducción a MySQL con phpMyAdmin

Persistencia de datos

- La información se guarda para volver a utilizarla
- Mecanismos
 - Ficheros
 - Almacenamiento básico
 - Acceso secuencial o aleatorio
 - Bases de datos
 - Información estructurada
 - Relaciones
 - Búsquedas
 - Acceso concurrente
 - Control de acceso a la información
 - Integridad

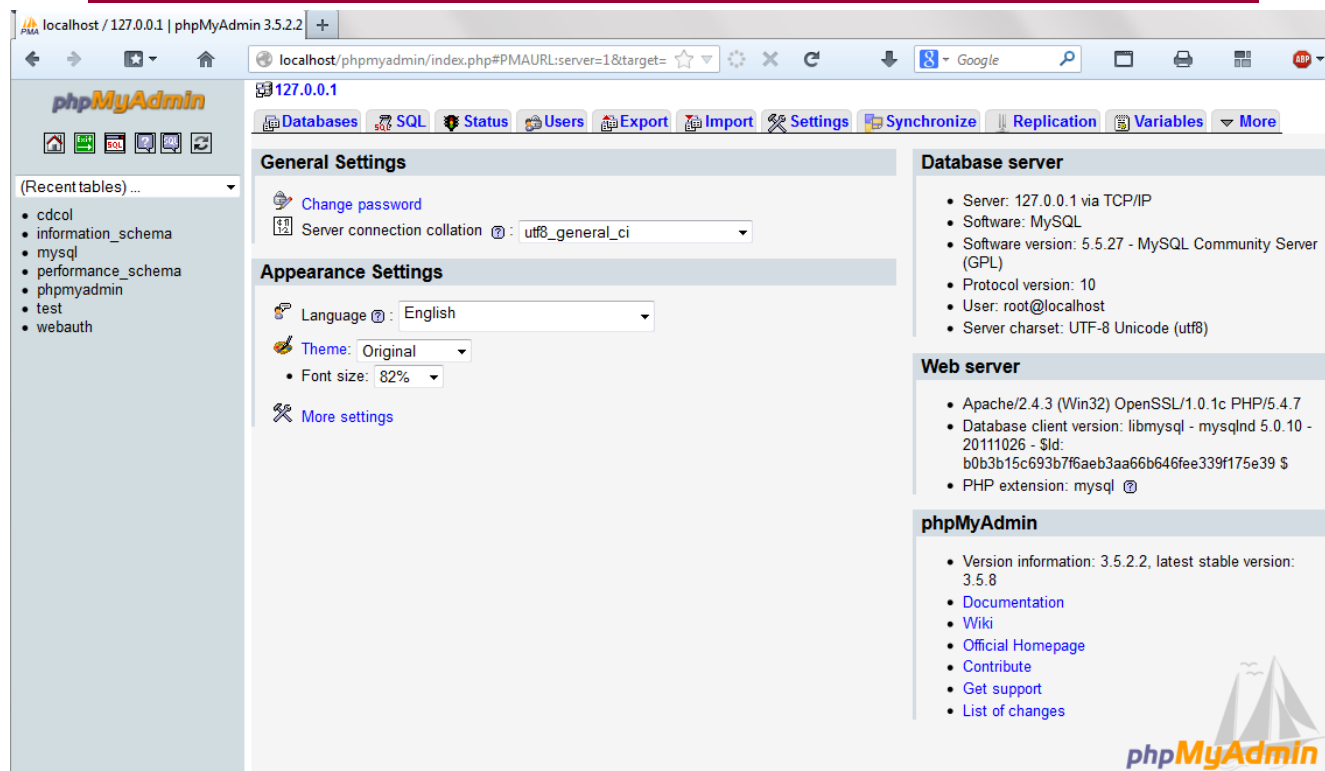
MySQL

- Sistema de Gestión de Base de Datos Relacional
 - La información se guarda en tablas
 - Una tabla es una colección de datos relacionados
 - Una tabla consta de columnas (campos) y filas (registros)
 - Las tablas se enlazan por relaciones entre columnas
- Implementa casi todo el estándar SQL (Structured Query Language)
- Código abierto
 - Actualmente de Oracle, que adquirió Sun, que tenía MySQL AB
- Escalable
 - Aplicaciones pequeñas y grandes (millones de registros)
- Transacciones, Multiusuario
- Eficiente: Multihilo, varias técnicas de hash, b-tree, etc.
- Conexión al servidor MySQL con sockets TCP/IP
 - Esto permite conectarla con casi cualquier plataforma

phpMyAdmin

- Herramienta que ofrece una interfaz gráfica para la administración del servidor MySQL
 - Configuración del servidor y las bases de datos
 - Gestionar (crear, modificar, borrar) las bases de datos, tablas, campos, relaciones, índices, etc.
 - Consultas con SQL, y mediante ejemplos (query by example)
 - Definir usuarios y asignar permisos
 - Realizar copias de seguridad
 - Crear gráficos (PDF) del esquema de la base de datos
 - Exportar a muchos formatos (documentos de texto, hojas de cálculo)
- En XAMPP se puede invocar en **http://localhost/phpmyadmin/**
- Configuración
 - Fichero **config.inc.php** (en el directorio raíz de phpMyAdmin)
 - Pero más recomendable a través de la interfaz web de phpMyAdmin en **http://www.dominio.com/phpMyAdmin/setup**
 - Usar el password del root de mySQL, que se aplica también a ese mismo usuario en phpMyAdmin (efectivo tras rearrancar mySQL)

phpMyAdmin – Página inicial

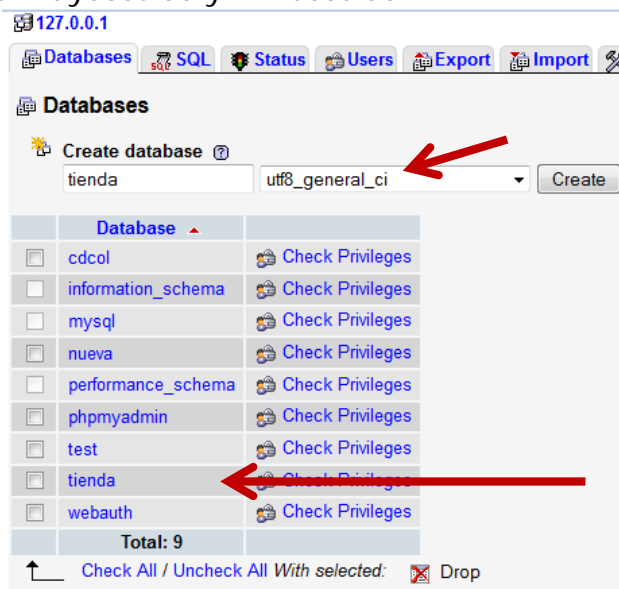


Usuarios de MySQL

- Conviene crear un nuevo usuario para cada sitio web
 - Cada sitio web tendrá sus propias bases de datos
 - El usuario root solo se debe usar para administración
- Entrar en phpMyAdmin como usuario root
 - En principio no tiene password, pero habrá que ponerlo, por seguridad
 - A continuación crear un nuevo usuario
 - por ejemplo para el sitio del ejemplo a continuación: **tienda**
 - Pestaña Users-> Add user
 - En la ventana que aparece indicar
 - Nombre de usuario: **tienda**
 - Host (local si estáis desarrollando con XAMPP en vuestro PC)
 - Password: **tienda**
 - Crear una base de datos para el nuevo usuario
 - Marcar la casilla *Create database with same name and grant all privileges*
 - No activar privilegios globales
 - Salir de la sesión como root
- Entrar con el nuevo usuario
 - Se puede trabajar con la nueva base de datos

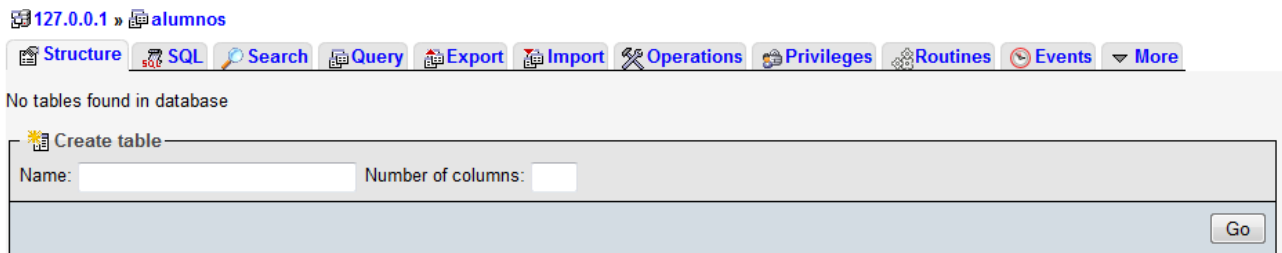
Creación de una base de datos con phpMyAdmin

- Como root: Create database en la pestaña Database
 - Indicar un nombre para la base de datos
 - La opción "collation" indica el conjunto de reglas de comparación y ordenación del texto en la base de datos, que dependerá del idioma
 - Por ejemplo, utf8_general_ci que vale para muchos idiomas y no es sensible a mayúsculas y minúsculas

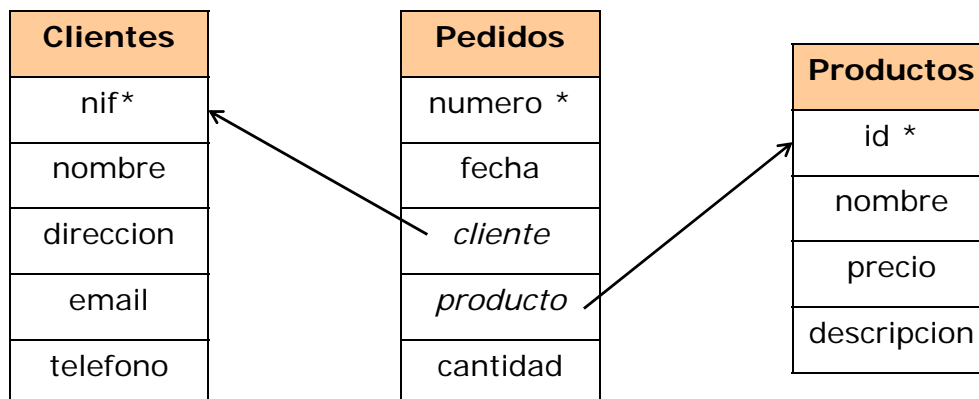


Manejo de la base de datos con phpMyAdmin

- Al seleccionar la base de datos creada aparecen las operaciones que se pueden realizar con ella
- Se pueden añadir permisos (pestaña *Privileges*) para que otros usuarios puedan usar la base de datos
- En Structure se pueden crear las tablas que definen el esquema de la base de datos



Ejemplo sencillo de base de datos: Tienda



***Clave primaria** (los objetos en esta columna son únicos y no nulos). Será indexada.

Se pueden definir también **índices para mejorar la eficiencia de las búsquedas

***Las **claves foráneas** (*foreign keys*) identifica una columna (o grupo de columnas) en una tabla que se refiere a otra columna (o grupo de columnas) en otra tabla, generalmente la clave primaria en la tabla referenciada.

Contribuyen a gestionar la integridad de la base de datos: no se puede crear un pedido de un cliente o un producto que no existan.

Las claves foráneas deberían indexarse porque se usarán para seleccionar registros con frecuencia.

Creación de tablas

- La base de datos consta de tablas
 - Cada una con una serie de columnas (campos)
 - Cada campo tendrá asociado un tipo:
 - Enteros: TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT
 - Números reales: DECIMAL, DOUBLE, FLOAT, REAL
 - Booleanos: BOOLEAN
 - Fecha: DATE, TIME, YEAR
 - Strings: VARCHAR (hasta 256 caracteres), TEXT
 - Como Storage Engine conviene usar InnoDB para poder gestionar relaciones entre tablas
 - Como Collation conviene usar utf8_general_ci

Creación de tablas

- La primera tabla es la de clientes, con cinco campos
 - nif: servirá como primary key (el nif es único)
 - nombre: de empresa o de persona (sería nombre + apellidos)
 - Se puede indexar para hacer búsquedas por este campo
 - direccion
 - email
 - telefono: como string para permitir uso de caracteres no numéricos

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I
nif	VARCHAR	10	None			<input type="checkbox"/>	PRIMARY	<input type="checkbox"/>
nombre	VARCHAR	30	None			<input type="checkbox"/>	INDEX	<input type="checkbox"/>
direccion	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>
email	VARCHAR	30	None			<input type="checkbox"/>	---	<input type="checkbox"/>
telefono	VARCHAR	15	None			<input type="checkbox"/>	---	<input type="checkbox"/>

Table comments:

Storage Engine:

Collation:

Creación de tablas

- Crear dos tablas más:
 - productos
 - pedidos

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A I
id	INT		None		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
nombre	VARCHAR	20	None			<input type="checkbox"/>	INDEX	<input type="checkbox"/>
descripcion	VARCHAR	200	None			<input type="checkbox"/>	---	<input type="checkbox"/>
precio	DECIMAL	6,2	None			<input type="checkbox"/>	---	<input type="checkbox"/>

Table comments:

Storage Engine:

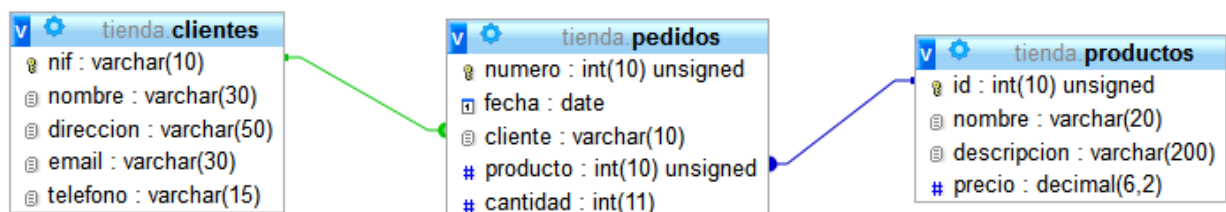
Collation:

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A I
numero	INT		None		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
fecha	DATE		None			<input type="checkbox"/>	---	<input type="checkbox"/>
cliente	VARCHAR	10	None			<input type="checkbox"/>	INDEX	<input type="checkbox"/>
producto	INT		None		UNSIGNED	<input type="checkbox"/>	INDEX	<input type="checkbox"/>
cantidad	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>

Definir relaciones entre tablas

- Usar la pestaña Designer para ver gráficamente las tablas
 - Se pueden recolocar las tablas
 - Para añadir una relación seleccionar el botón Create relation
 - Seleccionar la primary key de la tabla clientes: nif
 - Seleccionar la foreign key en la tabla pedidos: cliente
 - Aparece una ventana para seleccionar qué hacer para preservar la integridad de las referencias, con las siguientes operaciones:
 - DELETE: seleccionar RESTRICT
 - UPDATE: seleccionar CASCADE
 - La restricción más adecuada en la mayoría de los casos es evitar realizar borrados en cascada y actualizar en cascada
 - La relación queda establecida y aparece en el gráfico
 - Para salvar el diagrama, usar el botón Save



Definir relaciones entre tablas

- Si falla el establecimiento de una relación, comprobar que:
 - Ambas usan tablas InnoDB como storage engine
 - No hay registros en las tablas
 - Si los hubiera hay que comprobar que concuerdan para mantener la integridad de las referencias
 - El campo en la primera tabla es una clave primaria
 - El campo correspondiente en la segunda tabla está indexado
 - Ambos campos tienen exactamente los mismos tipos de datos
 - Si son numéricos, ambos deben ser unsigned (o ninguno)
- Una vez que se haya establecido la relación no será posible introducir registros en la segunda tabla a menos que se correspondan en el campo de la relación con el de la primera tabla
 - ¿Qué ocurre en la segunda tabla cuando se borra un registro de la primera tabla?

Introducir datos en tablas

- Seleccionar la tabla
 - Seleccionar la pestaña Insert
 - Introducir datos para los campos correspondientes en Value
- También se pueden introducir a través del programa PHP que recupera la información que un usuario haya introducido en un formulario de una página web
- En la pestaña Browse se pueden ver los registros de la base de datos y modificar campos de los mismos

Backup de la base de datos

■ Export

- Conveniente de forma regular
 - Especialmente si se hacen muchos cambios
- Opciones (seleccionar *Custom* en *Export Method*)
 - Qué se guarda
 - El servidor completo
 - Una base de datos entera
 - Una tabla
 - Estructura o datos, o ambos
 - Compresión: ninguna, zipped, gzipped, bziped
 - Formato
 - SQL, CSV, Word, Latex, Excel, OpenDoc, PDF, XML, JSON, etc.

■ El proceso inverso es posible con **Import**

Ejercicios con phpMyAdmin

- Instalar phpMyAdmin
- Crear la base de datos tienda con las tablas clientes, productos y pedidos tal como se han definido previamente
- Insertar elementos en las tres tablas, primero en clientes y productos y luego en pedidos
 - Comprueba qué ocurre si se intenta introducir un pedido para un cliente que no existe
- Intenta eliminar un cliente que tiene algún pedido
 - Observa el efecto de haber definido la política DELETE: RESTRICT cuando se estableció la FOREIGN KEY
- Intenta cambiar el nombre de un producto que tiene algún pedido
 - Observa el efecto de haber definido la política UPDATE: CASCADE cuando se estableció la FOREIGN KEY