



De Tecnologías a Soluciones

Dominar phpMyAdmin

para una administración efectiva de MySQL

Descubra el poder real de phpMyAdmin, aumente su productividad y control sobre MySQL.

Marc Delisle

[PACKT]
PUBLISHING

Dominar phpMyAdmin para una administración efectiva de MySQL

Marc Delisle



BIRMINGHAM - MUMBAI

Dominar phpMyAdmin para una administración efectiva de MySQL

Derechos reservados © 2004 Packt Publishing.

Todos los derechos están reservados. No se puede reproducir, almacenar en un sistema de recuperación o transmitir de ninguna forma o manera partes de este libro sin el previo permiso escrito del editor, a excepción del caso de breves citas incorporadas en artículos críticos o críticas.

Se ha hecho un gran esfuerzo al preparar este libro para asegurar la fidelidad de la información que se presenta. Sin embargo, la información que está en este libro se vende sin garantías, expresas o implicadas. Ni el autor, Pack Publishing o sus vendedores o distribuidores se hacen responsables de los daños que cause o pueda causar este libro directamente o indirectamente.

Packt Publishing ha prometido proveer información sobre las marcas registradas de todas las compañías y productos que se mencionan en este libro con un uso apropiado de las mayúsculas. Sin embargo, Packt Publishing no puede garantizar la fidelidad de esta información.

Primera actualización: Octubre 2004

Referencia de Producción: ES230307

Editado por Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK

ISBN 978-1-90481-198-5

www.packtpub.com

Diseño de la cubierta por www.visionwt.com

Créditos

Autor

Marc Delisle

Revisor técnico

Garvin Hicking

Alexander Turek

Programador general

Louay Fatoohi

Editor Técnico

Niranjan Jahagirdar

Presentación

Niranjan Jahagirdar

Indexadores

Vijay Tase

Ashutosh Pande

Corrector

Chris Smith

Señadora de la cubierta

Helen Wood

Traductor

Maika Vicente Navarro

Sobre el autor

Marc Delisle empezó a contribuir con phpMyAdmin en diciembre de 1998, cuando hizo la primera versión multi idioma. Ha contribuido de forma activa hasta mayo del 2001 como programador y administrador de proyectos. Ahora phpMyAdmin forma parte de su vida.

Trabaja en el Collège de Sherbrooke, Québec, Canadá, desde 1980 como programador de aplicaciones y administrador de redes. También ha dado clase sobre redes, seguridad, servidores Linux y desarrollo de aplicaciones PHP/MySQL. En una de sus clases, tuvo el placer de conocer a un usuario phpMyAdmin de Argentina.

Me siento muy agradecido con mi editor, Louay Fatoohi, quien me abordó para este libro, y me acompañó durante la producción; sus sensatos comentarios fueron muy apreciados. También quiero dar las gracias a Garvin Hicking, un miembro del equipo de desarrollo de phpMyAdmin y el revisor de este libro. Los ojos críticos de Garvin han ayudado a hacer que este libro sea más claro y completo.

Finalmente, no habría un libro sobre phpMyAdmin sin phpMyAdmin (el programa). Quiero dar las gracias a todos los contribuidores al código fuente y la documentación; el tiempo que dan al proyecto de software todavía me inspira y me motiva a continuar.

Se lo dedico a Carole, Corinne, Annie y Guillaume, con todo mi amor.

Sobre los revisores

Garvin Hicking es un programador Web alemán que trabaja para Faktor E GmbH. Crea aplicaciones Web con el uso de PHP y MySQL, en su tiempo libre disfruta trabajando en trabajos de código abierto como phpMyAdmin y Serendipity. Cuando no está en su ordenador, le gusta ir al cine con su novia y amigos, publicar en blogs y hacer fotos. Lo que más le gusta sobre su trabajo en Código abierto es que facilita la vida diaria de la gente al darles herramientas gratuitas, pero poderosas para jugar. Recibir comentarios de usuarios satisfechos es uno de los principios fundamentales de *dar y tomar* con los que le gusta vivir.

Alexander Marcus Turek nació el 2 de junio de 1985, en Dusseldorf, la capital de provincia alemana Northrhine-Westphalia. En estos momentos está estudiando *Ingeniería y administración de la información* en la Universidad de Karlsruhe, pero es originario de Mülheim an der Ruhr, la casa de su familia. Su primer contacto con la Web fue en 1998, cuando ganó un MODEM de 28,8k en *CeBit Home* en Hannover, Alemania. Unos meses más tarde, aprendió HTML y comenzó su primer proyecto Web, un archivo parche para un juego alemán llamado *Rabus' Update Site* (*Sitio de actualizaciones de Rabos*), que renombró `bugfixes.info`, cuando los dominios `.info` se hicieron disponibles. En el proceso, cambió del estático HTML a PHP para poder administrar el creciente archivo de forma más efectiva. Continuó aprendiendo PHP cuando intentó extender el portal.

Como el sistema de datos basado en flatfile se hizo muy lento cuando se hacían búsquedas en el archivo que crecía, también cambió a MySQL en el 2001. En este momento se puso en contacto con phpMyAdmin y el proyecto. Empezó revisando los archivos de idiomas porque estaban un poco desfasados y eran inconsistentes. Como se divertía haciéndolo continuó cogiendo algunos informes de errores y mandando parches. Loïc Chapeaux, uno de los co-responsables del mantenimiento de aquel momento, le añadió a la lista de programadores y le dio una cuenta CVS en marzo del 2002 de manera que pudiese fusionar los parches él mismo. Desde entonces, ha trabajado principalmente en la compatibilidad con MySQL 4.0, ha rehecho la zona de administración del servidor, desarrollado un nivel sencillo de abstracción para soportar MySQLi y ha continuado arreglando la compatibilidad; esta vez para MySQL 4.1 y 5.0. Desgraciadamente, sus estudios y phpMyAdmin requerían mucho tiempo y tuvo que dejar de trabajar en `bugfixes.info` en el 2003.

Sobre el traductor

Maika Vicente Navarro, traductora y correctora autónoma; inició su propio negocio en el 2003 que empezó con pequeñas traducciones. En estos momentos, colabora con diversos clientes en diferentes proyectos siendo el más importante Packt Publishing para quien ha traducido libros de tecnología de la información.

Tabla de contenidos

Introducción	1
Capítulo 1: Introducción a phpMyAdmin	7
PHP y MySQL: el dúo líder del código abierto	8
¿Qué es phpMyAdmin?	8
Historia	10
Resumen de las características de phpMyAdmin	13
Resumen	14
Capítulo 2: Instalación de phpMyAdmin	15
Fase de descarga	16
Instalación en un servidor remoto de Linux usando un cliente de Windows	16
Instalación en servidores locales Linux	18
Instalación en servidores locales Windows (Apache, IIS)	18
Configuración básica	19
El archivo config.inc.php	19
PmaAbsoluteUri	20
Secciones específicas del servidor	20
Extensión	21
connect_type	21
Configuración de compresión	22
Tipo de autenticación: config	22
Verificar la configuración básica	23
Configuración multi-servidor	23
Servidores definidos en el archivo de configuración	24
Servidor arbitrario	24
Instalación multi-usuario	25
Tipos de autenticación que se ofrecen	26
El control de usuario	26

Autenticación HTTP	27
Autenticación cookie	27
Seguridad	29
Protección a nivel del directorio	29
Control basado en la IP	30
Reglas	30
Orden de interpretación de las reglas	31
Restringir la lista de bases de datos	32
Proteger los datos en transmisión	32
Actualizar phpMyAdmin	33
Resumen	33
Capítulo 3: Información general de la interfaz	35
Paneles y ventanas	35
Paneles de inicio de sesión	35
Paneles de la izquierda y de la derecha	35
Página de inicio	36
Vistas	36
Ventana consulta	36
Página de arranque	36
Configuración de los títulos de la ventana	37
Configuración general de los iconos	37
Clasificación natural para los nombres de las bases de datos y las tablas	38
Selección de idioma	38
Temas	40
Configuración de los temas	40
Elección de temas	40
Panel de la izquierda	41
Lista de bases de datos y tablas	42
Modo instalación mínima	43
Atajo a las estadísticas de la tabla	44
Examinar con rapidez una tabla	45
Modo completo	45
Presentación jerárquica de tablas dentro de una base de datos	46
Elección de lista de servidores	48
Panel de la derecha	49
Página de inicio	49
Vista base de datos	50
Vista tabla	52
Vista servidor	53
Iconos para la página de inicio y menú tabulador	53
Ventana consulta	54

Cabecera y pie de página específicos de un sitio	55
Enlaces a los documentos MySQL	55
Resumen	56
Capítulo 4: Primeros Pasos	57
Creación de bases de datos	57
¿Sin permisos?	57
La creación de la primera base de datos está autorizada	58
Renombrar bases de datos	59
Crear nuestra primera tabla	60
Elegir los campos	60
Creación de la tabla	60
Elegir las claves	62
Inserción manual de datos	63
Ajustar el panel de entrada de datos para chars y varchars	65
Modo examinar	66
Enlaces a las consultas SQL	67
Barra de navegación	68
Organizar resultados	70
Marcar las filas con colores	71
Limitar la longitud de las columnas	72
Personalización del modo examinar	73
Crear una tabla adicional	73
Resumen	74
Capítulo 5: Cambiar datos	75
Modo editar	75
Tabulador para ir al próximo campo	76
Tratamiento de valores NULL	76
Aplicar una función a un valor	77
Duplicar filas de datos	78
Editar múltiples filas	79
Suprimir datos	80
Suprimir sólo una fila	80
Suprimir varias filas	81
Suprimir todas las filas de una tabla	81
Suprimir todas las filas de varias tablas	82
Suprimir tablas	82
Suprimir bases de datos	83
Resumen	83

Capítulo 6: Cambiar la estructura de la tabla	85
Añadir un campo	85
Modo vertical	86
Editar los atributos del campo	87
Campos TEXT	87
BLOB (Objeto Binario Extendido)	89
Carga de contenidos binarios	89
ENUM y SET	91
DATE, DATETIME, TIMESTAMP: pantalla emergente calendario	92
Administración de índices	93
Índices de un solo campo	94
Edición índices multi-campo y edición de índices	95
Índices FULLTEXT	96
Optimización de una tabla: EXPLAIN una consulta	97
Resumen	99
Capítulo 7: Exportar la estructura y los datos	101
Volcado de memoria, copias de seguridad y exportaciones	101
Exportación de base de datos	102
Sub-panel Exportar	103
Formato SQL	103
Opciones SQL	105
Sub-panel Guardar como archivo	109
Plantilla nombre de archivo	109
Compresión	110
Formato de datos CSV	110
CSV para datos MS Excel	112
LaTeX	112
XML	114
Native MS Excel	115
Exportación de la tabla	116
Exportación de archivos fragmentados	117
Exportaciones selectivas	117
Exportar resultados parciales de consulta	117
Exportar y recuadros	119
Guardar el archivo de exportación en el servidor	119
Resumen	120
Capítulo 8: Importar la estructura y los datos	121
Límites para la transferencia	121
Límites de tiempo	122
Otros límites	122

Importar archivos SQL	123
Importar archivos CSV	126
Diferencias entre los formatos SQL y CSV	126
Requisitos	126
Usar la interfaz LOAD	126
Opciones de LOAD	128
Carga de un directorio al servidor Web	129
Importar volcados comprimidos	130
Resumen	130
Capítulo 9: Buscar datos	131
Búsqueda en una tabla	131
Uso diario de phpMyAdmin	131
Entrar en la sub-página Buscar	131
Selección de los campos mostrados	132
Búsqueda de criterios por campo: consulta por ejemplo	133
Vista imprimir	134
Búsqueda con comodines	134
Combinar criterios	136
Aplicar una cláusula WHERE	137
Obtener resultados DISTINCT	137
Búsqueda de una base de datos completa	138
Resumen	139
Capítulo 10: Operaciones de las tablas	141
Mantenimiento de la tabla	142
Cambiar los atributos de las tablas	143
Tipo de tabla	143
Comentarios de las tablas	143
Orden de las tablas	144
Opciones de las tablas	145
Renombra, mover y copiar una tabla	145
Adjuntar datos a una tabla	146
Operaciones en múltiples tablas	147
Reparar una tabla "en uso"	147
Resumen	148
Capítulo 11: Sistema relacional	149
¿MySQL relacional?	149
InnoDB	150
Infraestructura de las tablas enlazadas	150
Posición de la infraestructura	150

Instalar la infraestructura de las tablas enlazadas	151
Instalación multi-usuario	152
Instalación de un usuario	154
Vista relación	156
Relaciones internas de phpMyAdmin	156
Definir la relación	157
Definir el campo presentación	157
Relaciones InnoDB	158
Tablas InnoDB sin la infraestructura tablas enlazadas	161
Beneficios de las relaciones definidas	162
Información de la clave foránea	162
Lista desplegable de las claves foráneas	163
Ventana de la tabla foránea que se puede examinar	163
Verificación de la integridad referencial	164
Actualización automática de los metadatos	165
Comentar columnas	165
Resumen	166
Capítulo 12: Escribir comandos SQL	167
Recuadro de consulta SQL	167
Vista base de datos	167
Vista tabla	168
Selector de campos	169
Hacer clic en el recuadro de consulta	169
Ventana de consulta	170
Opciones de la ventana de consulta	171
Historial SQL basado en JavaScript	171
Historial SQL basado en la base de datos (permanente)	171
Editar la consulta y la ventana de consulta	172
Consultas con múltiples sentencias	173
Impreso bonito (destacado de la sintaxis)	174
Validador SQL	175
Hacer que el validador esté disponible	176
Resultados del validador	176
Consultas acordes con el estándar	176
Consultas no acordes con el estándar	177
Resumen	178
Capítulo 13: Generador de consultas multi-tabla	179
Elegir tablas	180
Columna de criterios	181
Selector de campos: una columna o todas las columnas	181
Clasificar	182

Mostrar	182
Modificar la consulta	182
Criterios	183
Ajustar el número de filas de criterios	185
Ajustar el número de columnas de criterios	187
Joins automáticas	187
Ejecutar la consulta	188
Resumen	189
Capítulo 14: Marcadores	191
Crear un marcador después de ejecutar con éxito una consulta	192
Almacenar un marcador antes de mandar una consulta	193
Marcadores multi-consulta	194
Recuperar de la lista de marcadores	194
Ejecución de un marcador	195
Manipulación de un marcador	195
Marcadores públicos	195
Consulta inicial por defecto para una tabla	196
Marcadores parámetros	197
Crear un marcador parametrizado	197
Pasar un valor de parámetro a marcador	198
Ejecutar marcadores desde pma_bookmark	198
Resumen	199
Capítulo 15: Documentación del sistema	201
Vista impresión de la base de datos	201
Vista impresión selectiva de la base de datos	202
Vista impresión de la tabla	203
Diccionario de datos	203
Esquema relacional en PDF	204
Añadir una tercera tabla a nuestro modelo	204
Editar páginas PDF	205
Planear la página	205
Crear una página nueva	206
Editar una página	206
Presentar una página	207
Una nota sobre las fuentes que se usan	210
Resumen	210
Capítulo 16: Transformaciones basadas en MIME	211
Configuración de las columnas MIME	212
Tipo MIME	212
Transformaciones navegador	213
Opciones de las transformaciones	213

Requisitos para generar una imagen	214
GD2	214
Librerías JPEG y PNG	214
Memoria	214
Ejemplos de transformaciones	215
Icono al que se puede hacer clic (.jpeg o .png)	215
Enlace a una imagen	216
Formatear datos	216
Enlaces de texto	217
text/plain: link	217
text/plain: enlace de imagen	218
Preservar el formato original	218
Presentar partes de un texto	219
Aplicaciones externas	219
Ejemplo de aplicaciones externas: clasificación en la celda	220
Resumen	221
Capítulo 17: Conjunto de caracteres y colaciones	223
Archivos de idioma y UTF-8	223
Versiones de MySQL anteriores a 4.1.x	224
Codificación de datos	224
Conjunto de caracteres	225
Elegir el conjunto de caracteres efectivo	226
El impacto de cambiar	226
Importar y exportar con conjuntos de caracteres	227
MySQL 4.1.x y superior	228
Colaciones	229
Página de inicio	229
Crear una base de datos	230
Conjuntos de caracteres y colaciones disponibles	230
Conjunto de caracteres y colación efectivos	231
Vista base de datos	232
Vista tabla	233
Importar y exportar con un conjunto de caracteres	233
Vista servidor	234
Soporte Kanji	234
Resumen	235
Capítulo 18: Administración del servidor MySQL con phpMyAdmin	237
Entrar en la vista servidor	237
Administración de los usuarios y los privilegios	238
Información general del usuario	238

Añadir un usuario	239
Nombre del usuario	240
Proveedor	240
Contraseña	240
Privilegios globales	240
Límites de los recursos	240
Editar un usuario	241
Editar privilegios	241
Privilegios específicos de las bases de datos	241
Cambiar la contraseña	243
Cambiar la información de la cuenta o copiar el usuario	243
Eliminar un usuario	245
Información de la base de datos	245
Activar las estadísticas	246
Clasificar estadísticas	246
Verificación de los privilegios de las bases de datos	247
Vaciar las bases de datos seleccionadas	247
Operaciones del servidor	247
Verificación del estado del servidor	247
Página general del estado	247
Estado InnoDB	249
Variables del servidor	250
Procesos del servidor	251
Exportación completa del servidor	251
Resumen	252
Capítulo 19: Procedimiento para solucionar anomalías y soporte	253
Requisitos del sistema	253
Configuración básica	254
Resolver los errores comunes	254
Mensajes de error	254
No se puede cargar la extensión MySQL	255
MySQL Said: No se puede conectar al servidor local MySQL	255
Error # 2003: El servidor no responde	256
MySQL Said: Acceso denegado	256
Alarma: no se puede añadir información de cabecera	256
MySQL Said: Error 127, la tabla se tiene que reparar	257
La columna BLOB que se usa en la especificación de clave no tiene tamaño de clave	257
IIS: No hay un fichero de entrada especificado	257
Un error "404:página no encontrada" cuando se modifica una fila	257
Otros problemas	257
Página en blanco o caracteres raros	258
No se puede crear una base de datos	258
Problemas al importar archivos largos o cargando BLOBs largos	258
MySQL pérdida de la contraseña Root	258

Duplicar nombres de campo cuando creamos una tabla	259
La ventana de autenticación aparece más de una vez	259
phpMyAdmin cambia el tamaño de la columna	259
Ver muchas bases de datos que nos son nuestras	260
No poder almacenar un valor mayor que 127	260
Buscar soporte	260
FAQ	260
Foros de ayuda	260
Crear una cuenta SourceForge	261
Elegir el título del tema	261
Leer las respuestas	261
Rastreador de soporte	261
Rastreador de errores	261
Descripción del entorno	261
Descripción del error	262
Contribuir al proyecto	262
El código base	262
Actualización de las traducciones	262
Parches	263
Versiones futuras de phpMyAdmin	263
Resumen	263
Índice	265

Introducción

MySQL, usado por millones de programadores, es la base de datos de código abierto más popular, soporta numerosos sitios Web grandes, dinámicos y aplicaciones. MySQL ha adquirido esta amplia popularidad debido a su código abierto, su fiabilidad, su robustez y soporte para varias plataformas. La existencia de phpMyAdmin también ha ayudado con esta popularidad, la herramienta de administración estándar, industrial que hace que administrar bases de datos sea fácil para los programadores experimentados y los novatos. La poderosa interfaz gráfica que ofrece a MySQL, ha convertido a phpMyAdmin en una herramienta indispensable para MySQL y los programadores.

Este libro es un tutorial completo a phpMyAdmin, demostrando todo el potencial de esta herramienta. Muestra como configurar, activar y usar las numerosas características de phpMyAdmin, tanto básicas como avanzadas.

Que cubre este libro

Esto es una revisión rápida de los dieciocho capítulos del libro.

Capítulo 1 es una introducción a phpMyAdmin, su historia y sus características principales. *Capítulo 2* ofrece información detallada de las diferentes opciones de instalación y configuración, incluyendo instalar una copia de phpMyAdmin para múltiples usuarios y configurarla para administrar tres servidores diferentes. También se discuten temas de seguridad en este capítulo.

Capítulo 3 ofrece una información general de la interfaz gráfica de phpMyAdmin. Un examen más detallado de los diferentes paneles y ventanas aparece en los siguientes capítulos. En el *capítulo 4* vemos como crear nuestra primera base de datos y sus diferentes campos. El *capítulo 5* cubre suprimir una y varias filas, tablas y bases de datos. Este capítulo también cubre las operaciones de edición de datos, como por ejemplo tratar valores Null y aplicar funciones MySQL a los datos.

El *capítulo 6* se centra en las diferentes opciones de phpMyAdmin para cambiar la estructura de la tabla. Estas incluyen añadir tipos de campos como TEXT, BLOB, ENUM y SET, cargar datos binarios en campos BLOB y administrar índices. phpMyAdmin se puede usar para copiar datos y hacer copias de memoria inmediatas durante las fases de desarrollo y producción. El *capítulo 7* muestra como hacer estas tareas con la característica de exportación de phpMyAdmin. También se explican los diferentes formatos de datos en los que se puede exportar.

phpMyAdmin también puede importar datos que es el enfoque del *capítulo 8*. Importar datos SQL y CVS, la administración de los archivos comprimidos por phpMyAdmin, son las cosas que se cubren aquí. Además gracias a las características de fácil navegación de phpMyAdmin nos permite buscar con facilidad a través de los datos. El *capítulo 9* cubre la búsqueda en bases de datos y una tabla.

Los capítulos anteriores abarcaban principalmente los campos de las tablas. El *capítulo 10* se centra en las operaciones que afectan la tabla como un todo. Reparar y optimizar tablas, cambiar los diferentes atributos de las tablas, copias, mover tabla a otra base de datos, todo ello explicado aquí.

En el *capítulo 11* empezamos a leer sobre las características avanzadas de phpMyAdmin. Vemos como instalar la infraestructura de las tablas enlazada que es un requisito para usar las características avanzadas. Se cubren las instalaciones de un usuario y multi-usuario. También se explica con definir relaciones entre tablas.

Además nos permite realizar varias operaciones en las tablas a través de su interfaz gráfica. phpMyAdmin también nos permite ejecutar comandos SQL complejos para tareas que no se pueden ejecutar a través de su interfaz. Esta es la característica que cubre el *capítulo 12*. El *capítulo 13*, que cubre la búsqueda multi-tabla, complementa al capítulo 9 y nos muestra como buscar una tabla o una base de datos entera.

El *capítulo 14* cubre la poderosa característica de phpMyAdmin: el marcador de consultas, una de las características de las tablas enlazadas que se mencionarán en el capítulo 11. El capítulo nos muestra como registrar, manipular y pasar parámetros a marcadores.

Crear y mantener una buena información sobre la estructura de los datos es crucial, particularmente para los proyectos de equipo. phpMyAdmin nos permite hacerlo, esta característica se explica en el *capítulo 15*. Este capítulo nos muestra como generar una tabla sencilla, listas de columnas y usar diccionarios de datos para completar las listas de columnas, generar un esquema personalizado relacional para tablas en formato PDF.

phpMyAdmin puede hacer transformaciones basadas en MIME en los contenidos de la columna. La transformación de textos e imágenes se cubre en el *capítulo 16*. El *Capítulo 17* muestra como los administradores de sistema pueden usar phpMyAdmin

para administrar las cuentas de usuarios, sus privilegios y como verificar el estado del servidor.

El *capítulo 18*, el último capítulo del libro, cubre diferentes temas del procedimiento para solucionar anomalías y soporte. Cubre los mensajes de error más comunes y los problemas de configuración. El capítulo también incluye información sobre como y donde se puede obtener soporte técnico.

Que necesita para usar este libro

Necesita tener lo siguiente instalado en su máquina:

- Linux o Windows
- MySQL
- Apache o IIS
- phpMyAdmin

Convenciones

En este libro verá diferentes estilos de texto que distinguen entre diferentes tipos de información. Aquí hay algunos ejemplos de estos estilos y una explicación de su significado.

Hay tres estilos para el código. Código que aparece en un texto como sigue: "El parámetro `$cfg['PropertiesIconic']` puede tener los valores `TRUE`, `FALSE` o `'both'`."

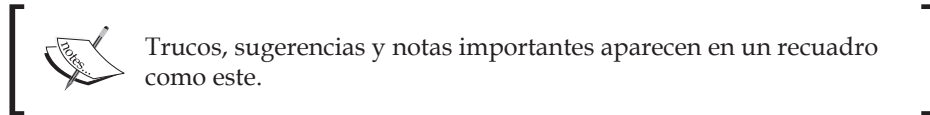
Si tenemos un bloque de código se presentará de la siguiente manera:

```
$cfg['PropertiesIconic']    = TRUE;
$cfg['ModifyDeleteAtLeft'] = TRUE;
$cfg['ModifyDeleteAtRight'] = FALSE;
```

Cuando queremos destacar una parte particular del bloque de código, las líneas importantes están en **negrita**:

```
CREATE TABLE 'autor' (
  'isbn' varchar(25) NOT NULL default '',
  'autor_id' int(11) NOT NULL default '0',
  PRIMARY KEY ('isbn'),
  KEY 'autor_id' ('autor_id')
) TYPE =MyISAM COMMENT= 'tiene la descripción del libro';
```


Las **palabras nuevas** e **importantes** se presentan en negrita. Las palabras que ve en la pantalla, en los menús o en recuadros de diálogo, por ejemplo, aparecen en el texto así: "haciendo clic en el botón **Siguiente** va a la siguiente pantalla".



Cualquier línea de comando de entrada o salida se escribe como abajo:

```
C:\packt>mysqladmin ping
mysql is alive
```

Comentarios del lector

Los comentarios de nuestros lectores son siempre bienvenidos. Déjenos saber lo que piensa de este libro, lo que le ha gustado y lo que no. Los comentarios de los lectores son importantes para desarrollar títulos que realmente le gusten.

Para enviarnos un comentario general, simplemente mándenos un correo electrónico a feedback@packtpub.com, asegúrese de mencionar el título del libro en el tema del mensaje.

Si hay un libro que necesita y que le gustaría que publicásemos, por favor mándenos una nota en Sugerir título desde <http://www.packtpub.com> o en el correo electrónico suggest@packtpub.com.

Si hay un tema en el que tenga experiencia y está interesado en escribir o contribuir en un libro, vea nuestra guía de autores en www.packtpub.com/authors.

Soporte del cliente

Ahora que es el orgulloso dueño de un libro Packt, tenemos un número de cosas para que aproveche el máximo su compra.

Descargar el código ejemplo del libro

Para descargarse un ejemplo del código o de los recursos extra de este libro, visite <http://www.packtpub.com/support> y seleccione este libro en la lista que verá. Los títulos disponibles para descargar se le mostraran en la pantalla.



Los archivos descargados contienen instrucciones sobre como usarlos.

Errata

A pesar que hemos tomado medidas para asegurarnos de la fidelidad de nuestro contenido, puede haber errores. Si encuentra un error en uno de nuestros libros; quizá un error en el texto o en el código, le agradeceríamos si nos mandase un informe sobre el mismo. Si lo hace puede ahorrar frustración al resto de lectores y ayudar a mejorar las versiones siguientes del libro.

Si encuentra una errata, puede enviar informe si visita <http://www.packtpub.com/support>, elija su libre y haga clic en el enlace **Add Errata** y escriba los detalles de su errata. Una vez se ha verificado la errata, se aceptará su envío y la errata se añadirá a la lista de erratas existentes. Las erratas existentes también se pueden ver en la misma página si hace clic en el enlace **View Errata**.

Preguntas

Puede contactar con nosotros en questions@packtpub.com si tiene un problema con algún aspecto del libro y haremos lo que podamos para ayudarle.

1

Introducción a phpMyAdmin

¡Bienvenido a la red evolucionada! En los últimos años, la red ha cambiado de forma drástica. Durante su infancia, la red era un medio usado para transmitir información **estática** ("¡Mira mi página inicial está en la red!"). Ahora, gran parte de la red porta información que se **genera dinámicamente** con programas de aplicaciones, en los que, tanto empresas como individuos confían para crear sus intrarredes y sitios Web públicos.

Debido a los claros beneficios de las bases de datos (tienen mejor acceso y una mejor estructura de la información), la mayoría de aplicaciones Web están controladas por bases de datos. La sección de entrada que se usa es el conocido navegador (que se despliega con rapidez) y que tiene un sistema de base de datos en el servicio de fondo. Las aplicaciones proveen la interfaz entre el navegador y la base de datos.

Aquellos que no usan un sitio Web controlado por una base de datos, hoy en día, no están usando este medio con su máxima capacidad. Además, podrían quedarse desfasados frente a competidores que han hecho el cambio. De este modo, no es cuestión de si *deberíamos* implementar un sitio controlado por una base de datos, sino sobre *cundo* y, aún más, *como* implementarlo.

¿Por qué usar aplicaciones Web? Porque mejoran la experiencia de los usuarios y les implica en el proceso al crear opciones como las siguientes:

- Adquisición de información de los comentarios.
- Permiten que los usuarios se comuniquen con nosotros y entre ellos a través de los foros.
- Ordenan los productos de nuestro sitio de comercio electrónico.
- Permite una edición sencilla de información basada en la red (administración del contenido).
- Diseño y mantenimiento de las bases de datos desde la red.

Hoy en día, WWW debería referirse a **onda mundial** (World-Wide Wave), una gran onda que modifica de manera profunda la manera de pensar de los programadores sobre las interfaces de usuario, la presentación de datos y, lo más importante, cómo llegan los datos al usuario y vuelven al centro de datos.

PHP y MySQL: el dúo líder del código abierto

Este capítulo describe el lugar de phpMyAdmin en el contexto de PHP/MySQL, explica la historia de phpMyAdmin y resume sus características. Veamos las soluciones que ofrecen los proveedores de servidores en estos momentos. El más dominante es la combinación PHP/MySQL.

Bien soportados por sus respectivas páginas iniciales, <http://www.php.net> y <http://www.mysql.com>, este dúo ha permitido a los programadores ofrecer numerosas aplicaciones comerciales Web de código abierto y lo más importante, ha permitido a los programadores de plantilla crear soluciones de red sólidas.

MySQL, que es el estándar SQL más compatible con ANSI-92, es un sistema de bases de datos conocido por su velocidad, robustez y su pequeña conexión a overhead, lo cual es importante en un contexto de red en el que las páginas se tienen que servir lo más rápido posible.

PHP, instalado como un módulo dentro del servidor de red, es un lenguaje de programación cuyas aplicaciones están escritas para comunicarse con MySQL con el servicio de fondo y con los navegadores en la parte frontal. Irónicamente, el significado de este acrónimo ha evolucionado junto con la red, de **Página inicial personal (Personal Home Page)** a **Página inicial profesional (Professional Home Page)**, hasta llegar a su definición recurrente de hoy en día: **PHP: Procesador de hipertexto**. Disponible en millares de dominios Web, controla su propio dominio de aplicaciones programadas con rapidez.

¿Qué es phpMyAdmin?

PhpMyAdmin es una aplicación Web que está escrita en PHP y que contiene códigos cliente en XHTML, CSS y JavaScript. Provee una interfaz Web completa para administrar las bases de datos MySQL y se reconoce en todo el mundo como la aplicación líder en este dominio.

Al ser de código abierto desde el principio de su existencia, ha disfrutado del soporte de numerosos programadores y traductores del mundo entero (se tradujo a 47 lenguas en el momento de su publicación). En estos momentos, el proyecto está en el servidor central de SourceForge.

Los proveedores de servidores de todo el mundo han mostrado su confianza en phpMyAdmin (página inicial oficial en <http://www.phpmyadmin.net>) al instalarlo en sus servidores. Además, podemos instalar nuestra propia copia de phpMyAdmin en nuestro espacio Web, siempre que en nuestro servidor haya instalada la versión mínima de PHP requerida por phpMyAdmin, que en estos momentos es PHP 4.1.0. Asimismo, el servidor Web debe tener acceso al servidor MySQL (versión 3.23.32 o superior); ya sea localmente o con una máquina remota. El popular CPanel (una aplicación de control de sitios Web) sirve de interfaz con phpMyAdmin.



The phpMyAdmin Project

[\[Home \]](#)
[\[Docs \]](#)
[\[Feedback \]](#)
[\[Stats \]](#)
[\[Awards \]](#)
[\[Downloads \]](#)
[\[Demo \]](#)
[\[Donations \]](#)
[\[Project \]](#)

Quick downloads

Latest stable version: **2.5.7-pl1**
[\[bzip2\]](#) [\[gzip\]](#) [\[zip\]](#) [\[release notes\]](#)

Latest beta release: **2.6.0-beta2**
[\[bzip2\]](#) [\[gzip\]](#) [\[zip\]](#) [\[release notes\]](#)

Latest version tested with PHP < 4.1.0 and MySQL < 3.23.32:
2.2.7-pl1 [\[bzip2\]](#) [\[gzip\]](#) [\[zip\]](#) [\[release notes\]](#)

News

[phpMyAdmin 2.5.7-pl1 is released](#) *lem9 - 2004-06-30 12:05*
 This is phpMyAdmin 2.5.7, patch level 1, fixing the vulnerability dated 2004-06-29 released on BUGTRAQ. See Documentation.html, FAQ 8.2.

We would like to put emphasis on the disappointment we feel when a bugreporter does not contact the authors of a software first, before posting any exploits.
[\[Read More/Comment\]](#)

[phpMyAdmin 2.6.0-beta2 is released](#) *lem9 - 2004-06-28 13:32*
 Here is phpMyAdmin 2.6.0-beta2. A few last-minute errors found their way into beta1, so beta2 fixes those, plus a few more.
[\[Read More/Comment\]](#)

[phpMyAdmin 2.6.0-beta1 is released](#) *lem9 - 2004-06-24 17:57*
 Welcome to 2.6.0-beta1! This version has reached feature freeze, and is now offered to a larger audience to get feedback.

Project information

phpMyAdmin is a tool written in PHP intended to handle the administration of MySQL over the Web. Currently it can create and drop databases, create/drop/alter tables, delete/edit/add fields, execute any SQL statement, manage keys on fields, manage

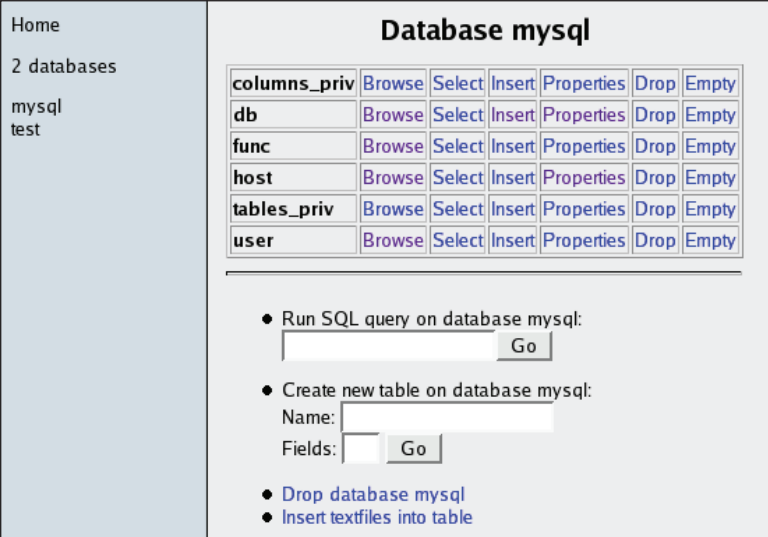
Historia

La primera versión interna (0.9.0) fue codificada por Tobias Ratschiller y tiene esta fecha 09/09/1998. Más tarde lanzó la versión 1.0.1 el 26/10/1998. Las versiones anteriores se ofrecían en el sitio de Tobias: <http://www.phpwizard.net> (este sitio ya no se asocia con él). Como Tobias comentó en las notas que le acompañaban:

 "Este trabajo está basado en MySQL-Webadmin de Meter Kuppelwieser. Su idea era crear una interfaz basada en la red para MySQL usando PHP3. A pesar de que no he usado su código fuente, he tomado prestados algunos de sus conceptos. phpMyAdmin se creó porque Peter me dijo que no iba a desarrollar más su (gran) herramienta".

Comparado con la versión de hoy en día (seis años después de la salida del original), la primera versión tenía unas características limitadas, pero que sin embargo se podía usar para crear bases de datos, editar su estructura, entrar y recuperar datos. Fíjese en la figura que sigue, el recuadro de la izquierda ya era una lista de nombres de bases de datos (aún no eran nombres de tablas) y en el recuadro derecho estaba el área de trabajo para administrar una base de datos o una tabla.

Este es el aspecto que la interfaz de bases de datos de la versión 1.3.0:



	columns_priv	db	func	host	tables_priv	user
	Browse	Browse	Browse	Browse	Browse	Browse
	Select	Select	Select	Select	Select	Select
	Insert	Insert	Insert	Insert	Insert	Insert
	Properties	Properties	Properties	Properties	Properties	Properties
	Drop	Drop	Drop	Drop	Drop	Drop
	Empty	Empty	Empty	Empty	Empty	Empty

- Run SQL query on database mysql:
- Create new table on database mysql:
Name:
Fields:
- [Drop database mysql](#)
- [Insert textfiles into table](#)

Para trabajar en una tabla, usted tenía la siguiente pantalla:

Home
2 databases
mysql
test

Database mysql - table func

Field	Type	Null	Default	Extra	Action				
name	char(64) binary	NO			Change	Drop	Primary	Index	Unique
ret	tinyint(1)	NO	0		Change	Drop	Primary	Index	Unique
dl	char(128)	NO			Change	Drop	Primary	Index	Unique
type	enum('function','aggregate')	NO	function		Change	Drop	Primary	Index	Unique

Keyname	Unique	Field	Action
PRIMARY	Yes	name	Drop

- Add new field
- View dump (schema) of table

Empecé a usar phpMyAdmin en la versión 1.2.0 (que salió el 29/11/1998) y me sentí atraído de inmediato por la idea de poder usar una aplicación Web para mantener una base de datos remota. Sin embargo, los estudiantes del Collège de Sherbrooke, donde trabajo en Québec, Canadá, son franco-parlantes; así que contacté a Tobias y le ofrecí la transformación de su código fuente subcontratando servicios para todos los mensajes en un archivo de mensajes. Aceptó la oferta y yo creé los archivos de mensajes en inglés y en francés. Entonces, el 27/12/1998, Tobias lanzó la versión 1.3.1, la primera versión multilingüística (en el interín había conseguido crear el archivo de mensajes en alemán).

En 1999 y la primera mitad de 2000, Tobias mejoró el sistema de navegación, añadió características e intercaló más archivos de lenguas. Su sitio del proyecto mantenía un foro de discusión. Así aparecieron nuevas ideas y se discutió sobre los parches. La versión 2.1.0 se lanzó el 06/08/2000, ésta fue la última versión que Tobias lanzó ya que no tuvo más tiempo para seguir con el proyecto.

Sin embargo, ya había numerosos usuarios que pedían más este producto. Muchos parches se colgaron en Internet y además no había manera de coordinarlos. Se había publicado un alerta de seguridad (y una solución), pero no se iba a lanzar una versión nueva. El 31/03/2001, Olliver Müller registró el proyecto phpMyAdmin en SourceForge.net y lanzó versión 2.2.0prel. En aquel momento se consideró la versión *no oficial*. Este reinicio del proyecto atrajo a nuevos programadores, quienes tenían la infraestructura de SourceForge (servidor CVS, foros, seguidor de errores, listas de correos) que les ayudó a que el desarrollo fuese más rápido. Yo personalmente me uní al proyecto en mayo del 2001 y empecé a fijar y mejorar el código, como ya hacían mis co-programadores.

Nos convertimos en "oficial" el 28/05/2001, cuando Tobias aceptó nuestra nueva versión como la oficial. Recuerdo esos meses de intenso desarrollo, con mejoras diarias, arreglos de errores, además de nuevas secciones de documentación. Este esfuerzo culminó el 31/08/2001 con el lanzamiento de la versión 2.2.0. Aquí hay un fragmento del archivo de anuncio para 2.2.0:



"Después de cinco meses, 5 lanzamientos beta y 4 lanzamientos de versiones candidatas, los programadores de phpMyAdmin tienen el placer de anunciarles la disponibilidad de phpMyAdmin 2.2.0 [...] el 31 de marzo de 2001, Oliver Müller (Suiza), soportado por Marc Delisle (Québec), Loïc Chapeaux (Francia) y un equipo de 8 programadores que reiniciaron el proyecto phpMyAdmin en SourceForge.net, con la autorización oficial del creador original de los paquetes. Y ahora, después de cinco meses de parches, seguimiento de errores, nuevas características y verificaciones, la versión 2.2.0 está terminada."

Esta versión tenía arreglos de seguridad, siete lenguas nuevas (con detección dinámica de lenguas), el código se había reescrito para que fuese compatible con CSS2 y XHTML 1.0, siguiendo las pautas de codificación de PEAR. La característica de los marcadores apareció en esta versión.

Durante el año que sigue, el desarrollo continuó con el lanzamiento de siete versiones menores. La última versión de la serie 2.2.x es 2.2.7-pl1, que también es la última que se ha verificado por completo bajo PHP 3. Una fecha a tener en cuenta: 03/04/2002 registramos phpmyadmin.net como el dominio oficial del proyecto.

El 11/08/2002, se lanzó la versión 2.3.0. Había habido tantas características que las páginas se estaban haciendo demasiado grandes en el plano vertical, así que esta versión fue la "gran versión dividida", presentando sub-páginas por cada tabla y grupo de características de base de datos.

El equipo comenzó con una nueva planificación para lanzar una nueva versión menor cada dos meses (2.3.1, 2.3.2...). El 23/02/2003, la versión 2.4.0 incluía un nuevo servicio de administración de servidor/usuario. Entonces, el 11/05/2005, el número de versión cambió a 2.5.0 para celebrar el nuevo sistema de transformación de celdas tipo MIME.

La versión del momento, 2.6.0, soporta la nueva extensión `mysqli` disponible en PHP 5, para un mejor resultado y una seguridad mejorada. La interfaz de esta versión se ha rediseñado para incluir nuevos iconos y un administrador de temas. Todas estas características están explicadas en este libro.

phpMyAdmin también ha ganado algunos premios, como se puede ver en la sección Premios en la página inicial del proyecto. Primero los administradores de SourceForge le dieron el premio "Proyecto del mes" en diciembre del 2002. En el documento estilo entrevista que preparamos para colgar en la página POTM SourceForge, escribí que estaba impresionado por índice de descargas de nuestro producto, que era de tres al minuto en aquel momento (desde entonces hemos alcanzado los diez por minuto en días punta).

phpMyAdmin recibió el 75% de votos de los lectores de la revista alemana *PHP Magazin* y su versión internacional, en la categoría "Mejor aplicación/herramienta PHP" en el 2003. Este premio se presentó oficialmente en la conferencia internacional de PHP del 2003, noviembre del mismo año.

Resumen de las características de phpMyAdmin

El objetivo de phpMyAdmin es ofrecer una administración completa basada en la Web de servidores y datos MySQL, progresar con MySQL y la evolución de estándares de Web. A pesar que nuestro producto no es perfecto, en estos momentos incluye las características más demandadas y muchos extras.

El equipo de desarrollo desarrolla el producto constantemente basándose en los informes de errores y las características demandadas, lanza nuevas versiones de forma regular en intervalos de unos dos meses.

phpMyAdmin ofrece características que cubren las operaciones básicas de las tablas y las bases de datos de MySQL. Además tiene un sistema relacional interno que mantiene meta datos para soportar características avanzadas. Para finalizar, los administradores del sistema pueden administrar usuarios y privilegios desde phpMyAdmin. Es importante recordar que la elección de operaciones disponibles en phpMyAdmin depende de los derechos que el usuario tiene en un servidor específico MySQL.

Las características básicas consisten en:

- Creación y Supresión de bases de datos.
- Crear, renombrar, copiar y suprimir tablas.
- Mantenimiento de la estructura de la tabla, incluyendo los índices.
- Operaciones especiales de tabla (reparación, optimización, cambiar tipo).
- Insertar, modificar y suprimir datos.
- Visualización de los datos en modo horizontal/vertical, vista Imprimir.
- Navegación y ordenación de datos.
- Carga de datos binarios.
- Búsqueda de datos (tabla o base de datos)
- Consultas siguiendo ejemplos (multi-tabla)
- Carga de lotes de datos.

- Exportación de la estructura y los datos en varios formatos con compresión.
- Instalación multiusuario y multiservidor.

Las características avanzadas incluyen:

- Comentarios a nivel de los campos.
- Teclas extranjeras (con o sin InnoDB)
- Navegar por una tabla ajena.
- Marcadores de las consultas.
- Diccionario de datos.
- Esquema relacional y diccionario PDF.
- Historial de las consultas SQL.
- Conexión a MySQL haciendo uso de la extensión tradicional `mysql` o la nueva extensión `mysql_i` (en PHP 5).
- Soporte del conjunto de caracteres de las bases de datos, tablas y campos (con MySQL 4.1).
- Transformación de los contenidos de la columna basada en el tipo MIME.
- Administración de temas para personalizar el aspecto de la interfaz.

Las características de administrador del servidor consisten en:

- Administración de usuarios y privilegios.
- Registro de los privilegios de las bases de datos.
- Estatus del servidor.
- Exportación completa del servidor.

Resumen

En este capítulo hemos hablado sobre;

- Como ha evolucionado la red para convertirse en un medio para publicar aplicaciones con facilidad.
- Porque deberíamos usar PHP/MySQL para desarrollar estas aplicaciones.
- Como se reconoce phpMyAdmin como una de las aplicaciones líder para hacer de interfaz entre MySQL con la red.
- La historia de phpMyAdmin.
- Una breve lista de sus características

2

Instalación de phpMyAdmin

Ha llegado la hora de instalar el producto (hay varias maneras de hacerlo) y darle una configuración mínima antes de usarlo por primera vez.

Tendremos que instalar phpMyAdmin por cualquiera de los siguientes motivos:

- Nuestro proveedor de servidor no ha instalado una copia central.
- Nuestro proveedor lo ha instalado pero no es la última versión.
- Estamos trabajando directamente en el servidor Web de nuestra empresa.

Algunos proveedores de servidores ofrecen un panel Web integrado con el que podemos administrar cuentas, incluyendo las cuentas MySQL y también tienen un administrador de ficheros que se puede usar para cargar contenidos Web. Dependiendo de todos estos factores, el mecanismo que usaremos para transferir phpMyAdmin a nuestro espacio Web variará. Necesitaremos información específica antes de empezar la instalación:

- El nombre del servidor Web o la dirección. Aquí asumiremos que es `www.mydomain.com`.
- La información de la cuenta de nuestro servidor Web (nombre de usuario, contraseña) que se usará para una transferencia FTP o SFTP, el inicio de sesión SSH o el inicio de sesión del panel de control Web.
- El nombre del servidor MySQL o la dirección. A menudo es `localhost` lo que quiere decir que está en la misma máquina que el servidor Web. Asumiremos que es `mysql.mydomain.com`.
- Nuestra cuenta del servidor MySQL (nombre de usuario, contraseña).

Fase de descarga

Hay varios archivos disponibles en la sección **descargas (Downloads)** de <http://www.phpmyadmin.net>. Puede que encuentre más de una versión allí; descárguese siempre la última versión estable. Sólo tenemos que descargar un archivo que incluye todos los archivos de lenguas y cargas sin importar la plataforma (navegador, servidor Web, MySQL o versión PHP).

Sin embargo, para PHP es diferente, si estamos usando un servidor que sólo soporta PHP3, tendremos que descargar un archivo con `.php3` en su nombre. En este caso, mientras seguimos las siguientes instrucciones, tendremos que transponer `.php3` cada vez que hablamos de ficheros `.php`.

Los archivos que se ofrecen tienen varias extensiones: `.zip`, `.bz2`, `.gz`. Descargue un fichero con una extensión para la que tenga el extractor correspondiente. `.zip` es el formato de archivo más universal en el mundo de Windows, a pesar que es más grande que `.gz` o `.bz2` (comunes en el mundo Linux/Unix).

Después de hacer clic en el fichero apropiado, tendremos que elegir la réplica (mirror) más cercana. El archivo empezará a descargarse y luego podemos grabarlo en nuestro ordenador.

El siguiente paso depende de la plataforma que estemos usando, las secciones que vienen detallan el proceso para algunas de las plataformas más comunes. Puede ir directamente a la sección que le interese.

Instalación en un servidor remoto de Linux usando un cliente de Windows

Usamos el explorador de archivos, hacemos clic en el archivo `phpMyAdmin-xxx` (en el que `xxx` representa la versión) que se ha cargado en la máquina de Windows, extraemos los archivos para ver todos los programas y directorios del directorio principal `phpMyAdmin-xxx` como se puede ver aquí al usar **PowerArchiver** :

Name	Modified	Size	Ratio	Packed	Path
add_message.sh	2003-11-18 07:20	630	48%	325	phpMyAdmin-2.6.0\lang\
add_message_file.sh	2003-11-18 07:20	590	45%	325	phpMyAdmin-2.6.0\lang\
afrikaans-iso-8859-1.inc.php	2004-06-30 18:25	42 344	68%	13 672	phpMyAdmin-2.6.0\lang\
afrikaans-utf-8.inc.php	2004-06-30 18:25	42 358	68%	13 681	phpMyAdmin-2.6.0\lang\
albanian-iso-8859-1.inc.php	2004-06-30 18:25	39 796	65%	14 006	phpMyAdmin-2.6.0\lang\
albanian-utf-8.inc.php	2004-06-30 18:25	40 988	66%	14 134	phpMyAdmin-2.6.0\lang\
application_octetstream_down...	2003-12-29 09:48	822	50%	411	phpMyAdmin-2.6.0\librar...
arabic-utf-8.inc.php	2004-07-07 07:12	48 923	69%	15 220	phpMyAdmin-2.6.0\lang\
arabic-windows-1256.inc.php	2004-06-30 18:25	35 554	61%	13 913	phpMyAdmin-2.6.0\lang\
arrow_ltr.gif	2004-06-16 17:45	88	02%	86	phpMyAdmin-2.6.0\theme
arrow_ltr.gif	2004-06-13 13:58	88	02%	86	phpMyAdmin-2.6.0\theme
arrow_ltr.png	2004-06-16 17:45	277	13%	240	phpMyAdmin-2.6.0\theme
arrow_ltr.png	2004-06-13 13:58	277	13%	240	phpMyAdmin-2.6.0\theme
arrow_rtl.gif	2004-06-13 13:58	91	02%	89	phpMyAdmin-2.6.0\theme
arrow_rtl.gif	2004-06-16 17:45	91	02%	89	phpMyAdmin-2.6.0\theme
arrow_rtl.png	2004-06-16 17:45	278	13%	242	phpMyAdmin-2.6.0\theme
arrow_rtl.png	2004-06-13 13:58	278	13%	242	phpMyAdmin-2.6.0\theme
asc_order.png	2004-06-13 13:58	124	03%	120	phpMyAdmin-2.6.0\theme
asc_order.png	2004-06-16 17:45	124	03%	120	phpMyAdmin-2.6.0\theme
azerbaijani-iso-8859-9.inc.php	2004-06-30 18:25	40 230	65%	14 185	phpMyAdmin-2.6.0\lang\
azerbaijani-utf-8.inc.php	2004-06-30 18:25	41 117	65%	14 363	phpMyAdmin-2.6.0\lang\

Use cualquier mecanismo que su extractor de archivos le ofrezca para guardar todos los archivos, incluyendo subdirectorios, en algún lugar de su espacio de trabajo. Aquí hemos escogido `c:\`, de modo que se crearía un directorio `c:\phpmyAdmin-xxx` cuando haya terminado la extracción.

Ahora, tenemos que abrir el fichero `config.inc.php` que está en el directorio recién creado.

Este fichero contiene caracteres especiales (fin de línea estilo Unix), de modo que debemos abrirlo con un editor de texto que entienda este formato. Si no usamos un editor de texto adecuado, el archivo que se nos mostrará tendrá líneas muy largas.

La mejor elección es un editor PHP estándar. Otra elección sería **WordPad** o **UltraEdit**, pero deberíamos tener cuidado en no añadir más caracteres (incluso líneas en blanco) al principio o al final del archivo. Ésto perturbaría la ejecución de phpMyAdmin y generaría el mensaje de error **No se puede enviar la salida de la cabecera...** Si esto sucede, vaya al capítulo 19 para ver *Procedimiento para solucionar anomalías y soporte*.

Ahora que el fichero está abierto, iremos a la sección *Configuración básica* y seguiremos las instrucciones. Guardaremos el fichero (en formato texto) y volvemos aquí.

Ahora ha llegado el momento de transferir toda la estructura del directorio `c:\phpMyAdmin` al servidor Web o a nuestro espacio Web. Usaremos nuestro programa FTP favorito o el panel de control Web para la transferencia.

El directorio exacto al que transferiremos phpMyAdmin puede variar: Puede ser nuestro directorio `public_html` u otro directorio al que normalmente transferimos documentos Web.

Cada vez que se modifique `config.inc.php` tendremos que transferirlo de nuevo a nuestro espacio Web. Esta transferencia se tendrá que hacer explícitamente con un programa de transferencia específico, o puede que sea una característica de un editor como HomeSite, Komodo o PHPEdit, en los que se puede guardar directamente a través del protocolo FTP.

Instalación en servidores locales Linux

Digamos que elegimos `phpMyAdmin-xxx` en formato `.gz` y que lo descargamos directamente de un directorio en el servidor Linux. Lo movemos al directorio raíz de nuestro servidor Web (por ejemplo `/var/www/html`) o a uno de sus subdirectorios (por ejemplo `/var/www/html/utilities`). Luego lo extraemos con el siguiente comando del entorno operativo o cualquier extractor de archivos gráfico que nuestro administrador de ventanas nos ofrezca:

```
tar -xzf phpMyAdmin-x.x.x-php.tar.gz
```

Luego nos aseguramos que los permisos y la propiedad del directorio y los archivos son apropiados para nuestro servidor Web; el usuario o grupo del servidor Web debe ser capaz de leerlos.

Más tarde usamos un editor de texto para abrir el fichero `config.inc.php` que está en el directorio recién creado.



Hay que tener cuidado para no añadir una línea en blanco al principio o al final del fichero; ya que esto perturbaría la ejecución de phpMyAdmin.

Ahora que el fichero está abierto, iremos a la sección *Configuración básica* de este capítulo.

Instalación en servidores locales Windows (Apache, IIS)

Aquí el procedimiento es similar al que se ha descrito en la sección *Instalación en un servidor remoto Linux* con el uso de cliente Windows, con la única excepción que el directorio se grabará en nuestro DocumentRoot (para Apache) o nuestro wwwroot (para IIS). Por supuesto, no hace falta que transfiramos nada más después de modificar `config.inc.php` ya que el directorio ya está en nuestro espacio Web.

Normalmente Apache funciona como un servicio de modo que tenemos que asegurarnos que el usuario bajo el cual el servicio está funcionando, tiene privilegios de lectura normales para acceder a nuestro directorio recién creado. Se aplica el mismo principio a IIS, que usa los usuarios **IUSR_machinename**. Este usuario *debe* tener permiso de lectura sobre el directorio. Puede ajustar los permisos en la pestaña **Seguridad/permisos** que está en las propiedades del directorio.

Configuración básica



Antes de continuar, podemos renombrar el directorio `phpMyAdmin-x.x.x` con un nombre más fácil de recordar como por ejemplo `phpMyAdmin`, `phpmyadmin`, `admin` o lo queramos. De este modo, nosotros o nuestros usuarios visitaremos una URL fácil de recordar para conectar `phpMyAdmin`.

El archivo `config.inc.php`

Ahora echemos un vistazo al archivo `config.inc.php`. Este archivo contiene código PHP válido que define la mayoría de los parámetros (expresados a través de las variables PHP) y que podemos cambiar para ajustar `phpMyAdmin` a nuestras necesidades. También hay comentarios PHP normales, allí podemos comentar nuestros cambios.

Si empezamos con `phpMyAdmin 2.6.0`, hay otro fichero de configuración: `layout.inc.php` ya que esta versión ofrece administración temática, este archivo contiene colores y ajustes temáticos específicos. Hay un fichero `layout.inc.php` por cada tema, está en `themes/themename`, por ejemplo, `themes/original`. Cubriremos la modificación de algunos de esos parámetros en el capítulo 4 *Primeros pasos*.

En esta sección se explican sólo los parámetros que necesitamos cambiar para hacer que `phpMyAdmin` funcione. El resto de parámetros se discutirán en los capítulos en los que se explican las características correspondientes.

El `config.inc.php` original contiene parámetros que permiten la conexión al servidor MySQL controlado por el servidor `localhost` con un nombre de usuario de `root`, sin contraseña. Esta es la configuración por defecto que producen la mayoría de procedimientos de instalación MySQL a pesar que no es muy seguro. Pero si nuestro servidor MySQL recién instalado todavía tiene la cuenta `root` por defecto, podremos conectarnos con facilidad y ver un alarma que `phpMyAdmin` da sobre esta falta de seguridad.

PmaAbsoluteUri

El primer parámetro que hay que mirar es `$cfg['PmaAbsoluteUri'] = '';` .

En la mayor parte de los casos lo dejaremos vacío ya que phpMyAdmin intentará auto detectar el valor correcto. Si más tarde navegamos por una tabla, editamos una columna y hacemos clic en **Guardar**, recibiremos un mensaje de error que nos indicará que debemos darle el valor correcto a este parámetro.

Por ejemplo, lo cambiaremos a:

```
$cfg['PmaAbsoluteUri'] = 'http://www.mydomain.com/phpMyAdmin_2.5.4/';
```

Luego tendremos que cambiar este otro parámetro cambiando FALSE por TRUE:

```
$cfg['PmaAbsoluteUri_DisableWarning'] = TRUE;
```

Con esto eliminaremos el mensaje de alarma que hemos visto al iniciar phpMyAdmin y demostraremos que nos hemos ocupado del parámetro absoluto URI.

Secciones específicas del servidor

La siguiente sección del archivo contiene configuraciones específicas de los servidores, cada una de ellas empezando así:

```
$i++;  
$cfg['Servers'][$i]['host'] = '';
```

Si examinamos sólo los parámetros normales del servidor (el resto de los parámetros se cubrirán a partir del capítulo 11), para cada sección del servidor veremos algo parecido:

```
$i++;  
$cfg['Servers'][$i]['host'] = '';  
$cfg['Servers'][$i]['port'] = '';  
$cfg['Servers'][$i]['socket'] = '';  
$cfg['Servers'][$i]['connect_type'] = 'tcp';  
$cfg['Servers'][$i]['extension'] = 'mysql';  
$cfg['Servers'][$i]['compress'] = '';  
$cfg['Servers'][$i]['controluser'] = '';  
$cfg['Servers'][$i]['controlpass'] = '';  
$cfg['Servers'][$i]['auth_type'] = '';  
$cfg['Servers'][$i]['user'] = '';  
$cfg['Servers'][$i]['password'] = '';  
$cfg['Servers'][$i]['only_db'] = '';  
$cfg['Servers'][$i]['verbose'] = '';
```

En esta sección tenemos que escribir `$cfg['Servers'][$i]['host']`, el nombre del proveedor del servidor o la dirección IP del servidor MySQL; por ejemplo `mysql.mydomain.com` o `localhost`. Si el servidor usa un puerto o una base no estándar, rellenaremos los valores correctos de `$cfg['Servers'][$i]['port']` o `$cfg['Servers'][$i]['socket']`. Vaya a la sección en `connect_type` para obtener más información sobre la base.

El nombre de servidor que aparece en la interfaz phpMyAdmin será el que escribiremos en "host" (a no ser que escribamos un valor en blanco para el siguiente parámetro). Por ejemplo :

```
$cfg['Servers'][$i]['verbose'] = 'Test server';
```

Esta característica también se puede usar para ocultar el nombre del proveedor de servidor real a los usuarios.

Extensión

El mecanismo tradicional mediante el cual PHP puede comunicarse con el servidor MySQL, disponible en PHP antes de la versión 5, es la extensión `mysql`. Esta extensión todavía está disponible en PHP 5, pero se ha desarrollado una nueva llamada `mysqli`, es mejor para PHP 5 debido a su rendimiento mejorado y su soporte de todas las funcionalidades de la familia MySQL 4.1.x. Esta extensión está diseñada para trabajar con la versión 4.1.3 o superior de MySQL.

En la versión 2.6.0 de phpMyAdmin, se ha implementado una nueva biblioteca que hace posible el uso de ambas extensiones: Se puede elegir cualquiera de las dos en un servidor particular. Indicamos en `$cfg['Servers'][$i]['extension']` que extensión queremos usar.

Otro parámetro importante (que no es específico de los servidores pero que se aplica a todas las definiciones del servidor) es a `$cfg['PersistentConnections']`. Este parámetro, cuando está configurado `TRUE`, en todos los servidores que conectemos con la extensión `mysql`, ordena a PHP que mantenga la conexión al servidor MySQL abierta. Esto acelera la interacción entre PHP y MySQL. Sin embargo, si está configurado `FALSE` por defecto en `config.inc.php`, es porque las conexiones permanentes causan un empobrecimiento de los recursos del servidor: MySQL rechaza nuevas conexiones. Por esta razón, esta opción ni siquiera está disponible en la extensión `mysqli`.

connect_type

Tanto las extensiones `mysql` y `mysqli` usan un socket (enchufe) automáticamente para conectar a MySQL si el servidor es `localhost`. Mire esta configuración:

```
$cfg['Servers'][$i]['host']           = 'localhost';
$cfg['Servers'][$i]['port']           = '';
$cfg['Servers'][$i]['socket']         = '';
$cfg['Servers'][$i]['connect_type']   = 'tcp';
$cfg['Servers'][$i]['extension']      = 'mysql';
```

El valor por defecto para `connect_type` es `'tcp'`. Sin embargo la extensión usará un `socket` porque concluye que esto es más eficiente si el proveedor del servidor es `localhost`, así que en este caso, podemos usar `tcp` o `socket` como `connect_type`. Para forzar una conexión `tcp` real podemos especificar `127.0.0.1` en lugar de `localhost` en parámetro `host`. Ya que el parámetro `socket` está vacío, la extensión intentará usar el `socket` por defecto. Si el `socket` por defecto, tal y como está definida en `php.ini`, no corresponde al `socket` real asignado al servidor MySQL, tenemos que poner el nombre del `socket` (por ejemplo `/tm/mysql.sock`) en `$cfg['Servers'][$i]['socket']`

Si el nombre del proveedor del servidor no es `localhost`, se hará una conexión `tcp`: en el puerto especial `3307`. Sin embargo, si deja el valor del puerto vacío se usará el puerto `3306` por defecto:

```
$cfg['Servers'][$i]['host']           = 'mysql.domain.com';
$cfg['Servers'][$i]['port']           = '3307';
$cfg['Servers'][$i]['socket']         = '';
$cfg['Servers'][$i]['connect_type']   = 'tcp';
$cfg['Servers'][$i]['extension']      = 'mysql';
```

Configuración de compresión

Si empieza con PHP 4.3.0 y MySQL 3.23.49, el protocolo que se usa para la comunicación entre PHP y MySQL permite un modo comprimido. El uso de este modo permite una mayor eficiencia. Para aprovechar mejor este modo, simplemente especifique:

```
$cfg['Servers'][$i]['compress']      = 'TRUE';
```

Tipo de autenticación: config

Para hacer nuestra primera prueba usaremos el tipo de autenticación `config` que es fácil de entender. Sin embargo, en la sección *Instalación multi-usuario*, veremos maneras de autenticar más poderosas y versátiles.

¡A pesar que parece que estemos conectándonos a phpMyAdmin, no lo estamos haciendo! El sistema de autenticación es una función del servidor MySQL. Sólo estamos usando phpMyAdmin (que está funcionando en el servidor Web) como una interfaz que manda nuestra información de usuario y contraseña al servidor MySQL. De hecho no nos conectamos *en* phpMyAdmin sino *a través de* phpMyAdmin.



Si usamos el tipo de autenticación `config` dejamos phpMyAdmin abierto a intrusiones, a menos que lo protejamos tal y como se explica en la sección *Seguridad* de este capítulo.

Aquí escribimos nuestro nombre de usuario y la contraseña para el servidor MySQL.

```
$cfg['Servers'][$i]['user']          = 'marc';
$cfg['Servers'][$i]['password']      = 'bingo';
```

Entonces podemos grabar los cambios que hemos hecho en `config.inc.php`.

Verificar la configuración básica

Ahora ha llegado el momento de conectar phpMyAdmin por primera vez. Con esto verificaremos lo siguiente:

- Los valores que hemos escrito en el archivo `config`.
- La configuración del componente PHP dentro del servidor Web.
- La comunicación entre la red y los servidores MySQL.

Iniciaremos nuestro navegador y señalamos el directorio donde hemos instalado phpMyAdmin, como por ejemplo en `http://www.mydomain.com/phpMyAdmin`. Si esto no funciona, lo intentaremos con `http://www.mydomain.com/phpMyAdmin/index.php` (esto querría decir que nuestro servidor Web no está configurado para interpretar `index.php` como documento de inicio por defecto).

Si todavía le da error, vaya al capítulo 19, *Procedimiento para solucionar anomalías y soporte*. Ahora deberíamos mirar la página de inicio de phpMyAdmin.

El capítulo 3 le ofrece información general de los paneles que vamos a ver ahora.

Configuración multi-servidor

El archivo original `config.inc.php` tiene tres secciones específicas para los servidores, lo que permite a una sola copia de phpMyAdmin administrar tres servidores distintos, pero podemos añadir secciones para administrar más. Veamos como podemos configurar más servidores.

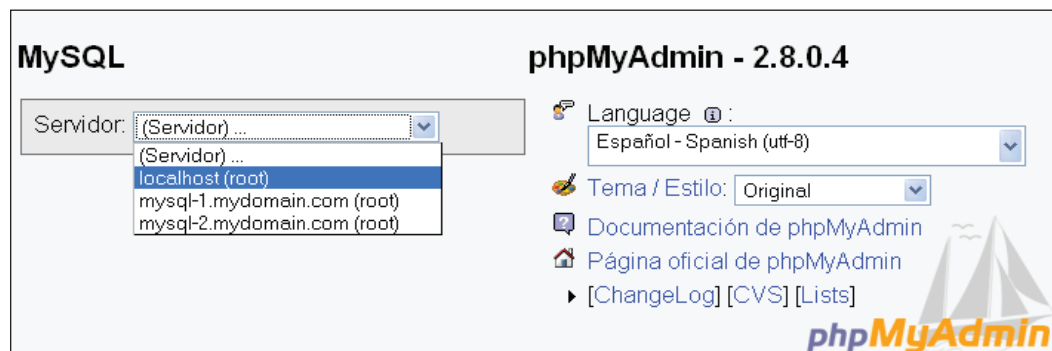
Servidores definidos en el archivo de configuración

En las secciones específicas de los servidores del archivo `config.inc.php`, vemos líneas que se remiten a `$cfg['Servers'][$i]` por cada servidor. Aquí la variable `$i` se usa para configurar más servidores y para que uno pueda cortar y pegar secciones enteras del fichero de configuración con facilidad.

Luego, al final de las secciones, la siguiente línea controla lo que sucede durante el inicio:

```
$cfg['ServerDefault'] = 1;
```

El valor por defecto, 1, significa que phpMyAdmin se conectará al primer servidor definido. Podemos especificar cualquier número para las secciones específicas de los servidores correspondientes. También podemos escribir el valor 0, que significa que no hay un servidor por defecto, en ese caso phpMyAdmin se presentará con una elección de servidores:



Servidor arbitrario

Se puede usar otro mecanismo si queremos poder conectar a un servidor MySQL sin definir. Primero tenemos que configurar el siguiente parámetro :

```
$cfg['AllowArbitraryServer'] = TRUE;
```

Luego, tenemos que añadir el tipo de autenticación `cookie`, que se explica en la siguiente sección. Podremos elegir servidor y escribir el nombre de usuario y la contraseña.



Este mecanismo probablemente debería usarse en conjunto con un mecanismo de seguridad reforzado (vea la sección *Seguridad*) porque cualquier servidor MySQL al que se pueda acceder desde nuestro servidor Web se podría conectar.

Como pude ver aquí, aún podemos elegir entre uno de los servidores definidos en Elección de servidor o podemos escribir un nombre arbitrario de servidor, de usuario y de contraseña:

Bienvenido a phpMyAdmin 2.8.0.4

Language 

Español - Spanish (utf-8)

Iniciar sesión

Servidor

Usuario: root

Contraseña:

Elección del servidor: localhost

Continuar

Instalación multi-usuario

Puede ser que queramos que una copia de phpMyAdmin sea usada por un grupo de personas, cada una de ellas con su propio nombre de usuario y su contraseña, y ver sólo las bases de datos a las que tienen acceso. O puede que prefiramos que nuestro nombre de usuario y nuestra contraseña aparezcan en un texto claro en `config.inc.php`.

Tipos de autenticación que se ofrecen

En lugar de confiar en la transmisión del par nombre de usuario/contraseña almacenados en `config.inc.php`, phpMyAdmin se comunicará con el navegador y obtendrá datos de autenticación del mismo. Esto habilita un **inicio de sesión verdadero** para todos los usuarios que están definidos en un servidor MySQL específico, sin tener que definirlos en el fichero de configuración. Se ofrecen dos modos para permitir un inicio de sesión controlado a MySQL a través de phpMyAdmin: `http` y `cookie`. Tendremos que elegir el que más se ajuste a nuestra situación y entorno específicos (veremos esto con más detalle en un momento).

Ambos modos requieren que primero definamos al usuario de control.

El control de usuario

Para poder usar tipos de autenticación de todo tipo de usuario MySQL (en MySQL, los privilegios del usuario se expresan de varias maneras), debemos definir un control usuario y contraseña en la sección específica de servidores del servidor. Si no definimos uno, los usuarios que se han definido en MySQL con una sintaxis de `'user'@'hostname'` o `'user'@'%'` podrán iniciar la sesión, el resto no.

El control de usuario es un usuario especial (el nombre que se suele escoger es **pma**, una abreviatura familiar de phpMyAdmin) que tiene derecho a leer algunos campos de la base de datos especial **mysql** (que contiene todas las definiciones de usuario). phpMyAdmin manda consultas con este control de usuario especial sólo para las necesidades específicas de autenticación, no para una operación formal. Los comandos para crear el control de usuario están disponibles `Documentation.html` de phpMyAdmin, puede que varíen de versión a versión. Estos documentos contienen los comandos más recurrentes.

Hay otra razón para definir el control de usuario: Para poder usar las características avanzadas de phpMyAdmin.

Cuando hayamos definido nuestro control de usuarios, rellenaremos los parámetros como en el siguiente ejemplo:

```
$cfg['Servers'][$i]['controluser']    = 'pma';  
$cfg['Servers'][$i]['controlpass']   = 'bingo';
```



Uso la contraseña bingo cuando enseño phpMyAdmin; se recomienda no usar la misma contraseña para su propia instalación.

Autenticación HTTP

Este modo, `http`, es el modo tradicional que HTTP ofrece, en el que el navegador pide un nombre de usuario y una contraseña, los manda a phpMyAdmin y los sigue mandando hasta que todas las ventanas del navegador están cerradas.

Para habilitar este modo, simplemente use la siguiente línea :

```
$cfg['Servers'][$i]['auth_type'] = 'http';
```

Este modo tiene algunas limitaciones:

- Dependiendo de la versión de PHP puede que no soporte la autenticación HTTP. Funciona cuando PHP se ejecuta como módulo bajo Apache; para el resto de casos deberíamos consultar los documentos PHP de nuestra versión.
- Si queremos proteger el directorio phpMyAdmin con un fichero `.htaccess` (vea la sección *Seguridad* de este capítulo), ésto interferirá con la autenticación de HTTP. No podemos usarlos a la vez.

No es una desconexión de verdad, tendremos que cerrar todas las ventanas del servidor para poder conectarnos otra vez con el mismo nombre de usuario. Incluso teniendo en cuenta estas limitaciones, este modo puede ser una buena elección por las siguientes razones:

- No hace falta que el navegador acepte cookies.
- Algunos navegadores (como Mozilla) pueden almacenar la información de autenticación de forma encriptada.
- Es más rápido que procesar cookies.

Autenticación cookie

El otro modo `cookie`, presenta un panel de inicio de sesión (vea la siguiente figura) que proviene de phpMyAdmin. Se puede personalizar ya que tenemos el código fuente de la aplicación. Sin embargo, como ya habrá adivinado, para la autenticación cookie, el navegador debe aceptar las cookies que provienen del servidor Web.

Este modo almacena el nombre de usuario que se ha escrito en la pantalla de inicio de sesión en una cookie permanente de nuestro navegador. La contraseña se almacena como una cookie temporal. En una configuración multi-servidor, la pareja nombre de usuario/contraseña que corresponden a cada servidor se almacenan por separado. Para proteger el secreto nombre de usuario/contraseña contra algunos métodos de ataque que tienen como contenido el contenido de la cookie, éstas se encriptan con el mecanismo Blowfish. Así que para usar este modo tenemos que definir una contraseña secreta (una vez) en `config.inc.php`, que se usará para encriptar de forma segura todas las contraseñas que se almacenen como cookies en la instalación de este phpMyAdmin.

Ésto se hace escribiendo una contraseña secreta aquí :

```
$cfg['blowfish_secret'] = 'SantaLivesInCanada';
```

Luego, para cada sección específica del servidor use lo siguiente:

```
$cfg['Servers'][$i]['auth_type'] = 'cookie';
```

La próxima vez que iniciemos phpMyAdmin veremos el panel de inicio de sesión.

Por defecto, phpMyAdmin muestra (en el panel de inicio de sesión) el último nombre de usuario que se ha conectado con éxito a este servidor particular, tal y como se ha recuperado de la cookie permanente. Si este comportamiento no es aceptable (preferimos que cualquier persona que se conecte a la misma estación de trabajo no vea al usuario anterior), podemos configurar el siguiente parámetro como FALSE:

```
$ cfg['LoginCookieRecall'] = FALSE;
```

Se añadió una característica de seguridad phpMyAdmin 2.6.0: La contraseña que se ha escrito tiene un tiempo límite de validez. Esta característica ayuda a proteger la sesión que se está utilizando. Después de conectar con éxito, nuestra contraseña se almacena en una cookie junto con un temporizador. Cada acción en phpMyAdmin reinicia el temporizador. Si nos mantenemos *inactivos* un cierto número de segundos, definidos en `$cfg['LoginCookieValidity']`, nos desconectan y tenemos que iniciar la sesión otra vez. Por defecto son 1800 segundos. El modo de autenticación `cookie` es superior al `http` en términos de las funcionalidades que ofrece. Ofrece un inicio y finalización de sesión verdaderos, se puede usar con PHP en ejecución en cualquier tipo de servidor Web.

Seguridad

La seguridad se puede examinar a varios niveles:

- Nivel de protección del directorio para phpMyAdmin.
- Control de acceso basado en la IP.
- Las bases de datos que un usuario legítimo puede ver.
- Protección de datos en transmisión.

Protección a nivel del directorio

Imagine que una persona no autorizada está intentando ejecutar nuestra copia de phpMyAdmin. Si hemos usado un tipo de autenticación `config`, cualquier persona que conozca la URL de nuestro phpMyAdmin tendrá los mismos derechos efectivos sobre nuestros datos que nosotros. En este caso, deberíamos usar el mecanismo de protección de directorio que ofrece nuestro servidor Web (por ejemplo `htaccess`) para añadir un nivel de protección.

Si usamos tipos de autenticación `http` o `cookie`, nuestros datos estarán bastante seguros, pero deberíamos tomar las precauciones normales con nuestra contraseña (incluyendo su cambio periódico).

El directorio donde está instalado phpMyAdmin contiene datos delicados. No sólo se debe proteger el archivo de configuración sino que también se tienen que proteger todos los programas almacenados para que no se puedan cambiar. Deberíamos asegurarnos que, aparte de nosotros, sólo el usuario efectivo del servidor Web tiene acceso de lectura a los ficheros que están en este directorio y que sólo nosotros podemos *escribirlos*.



Los programas de phpMyAdmin nunca deben modificar nada dentro de este directorio, excepto cuando usamos la característica **Guardar fichero exportado en el archivo**, que se explica en el capítulo 7.

Otro ataque posible puede venir de otros programadores que tengan una cuenta en el mismo servidor Web que nosotros. En este tipo de ataque, alguien puede intentar abrir nuestro fichero `config.inc.php`. Como este archivo se puede leer en el servidor Web, cualquiera puede intentar incluir nuestro fichero desde sus programas PHP. Para protegernos de esto, deberíamos asegurarnos con el proveedor que el modo seguro PHP está activo.

Control basado en la IP

Se puede añadir un nivel de protección adicional que verifique la dirección del Protocolo de Internet (IP) de la máquina desde la que se recibe la petición para usar phpMyAdmin.

Para llegar a este nivel de protección, construiremos reglas que permitirán o denegarán el acceso y especificaremos el orden con el que estas reglas se aplicarán.

Reglas

El formato de una regla es:

```
<'allow' | 'deny'> <username> [from] <source>
```

para ser opcional. Aquí hay algunos ejemplos:

```
allow Bob from 1.2.3.4
```

Se le permite acceso al usuario Bob desde la dirección IP 1.2.3.4.

```
allow Bob from 1.2.3/24
```

Se permite el acceso al usuario Bob desde cualquier dirección que corresponda con la red 1.2.3 (es una correspondencia IP CIDR).

```
deny Alice from 4.5/16
```

El usuario Alice no puede acceder cuando usa la red 4.5.

```
allow Melanie from all
```

El usuario Melanie puede iniciar la sesión desde cualquier dirección.

```
allow Julie from localhost
```

Equivalente a 127.0.0.1

```
Deny % from all
```

All se puede usar como un equivalente a 0.0.0.0/0, lo que quiere decir cualquier proveedor.

Normalmente tendremos numerosas reglas. Digamos que deseamos tener dos reglas seguidas:

```
allow Marc from 45.34.23.12
allow Melanie from all
```

Tenemos que ponerlas en config.inc.php (en la sección relacionada específica para el servidor) como sigue :

```
$cfg['Servers'][$i]['AllowDeny']['rules'] =
    array('allow Marc from 45.34.23.12,
        'allow Melanie from all');
```

Cuando definimos una regla sencilla o múltiple, se usa un conjunto() PHP, y debemos seguir su sintaxis, cerrando cada regla con una comilla y separando cada regla con una coma.

El siguiente parámetro explica el orden con que se interpretan las reglas.

Orden de interpretación de las reglas

Por defecto este parámetro está vacío :

```
$cfg['Servers'][$i]['AllowDeny']['order'] = '';
```

Eso quiere decir que *no* se hace una verificación basada en la IP.

Suponga que queremos permitir acceso por defecto pero denegar acceso sólo a algunas parejas nombre de usuario/IP. Deberíamos usar:

```
$cfg['Servers'][$i]['AllowDeny']['order'] = 'deny,allow';
```

En este caso, todas las reglas deny se aplicarán primero seguidas por las reglas allow. Si este caso no se menciona en las reglas, se permite el acceso. Si queremos ser más restrictivos desautorizaremos por defecto. Podemos usar:

```
$cfg['Servers'][$i]['AllowDeny']['order'] = 'allow,deny';
```

Esta vez, todas las reglas allow se aplican primero seguidas por las reglas deny. Si este caso no se menciona en las reglas, se deniega el acceso.

La tercera manera (la más restrictiva) de especificar el orden de las reglas es:

```
$cfg['Servers'][$i]['AllowDeny']['order'] = 'explicit';
```

Las reglas deny se aplican antes que las reglas allow, pero para que se acepte un nombre de usuario/dirección IP *deben estar en la lista* de las reglas allow y no en las deny.

Restringir la lista de bases de datos

Algunas veces es útil evitar mostrar todas las bases de datos a las que el usuario tiene acceso en el panel de la izquierda. En este caso, se puede usar el parámetro `only-db`. Puede tener un nombre de base de datos o una lista de nombres de bases de datos. Sólo se verán esas bases de datos en el panel de la izquierda :

```
$cfg['Servers'][$i]['only_db']          = 'payroll';  
$cfg['Servers'][$i]['only_db']          = array('payroll', 'hr');
```

El nombre de la base de datos puede tener caracteres comodín MySQL como por ejemplo `_` y `%`.

Este parámetro se aplica a todos los usuarios de esta configuración específica del servidor.



Este mecanismo no reemplaza el sistema de privilegios MySQL. Los derechos del usuario en otras bases de datos todavía se aplican, pero no pueden usar el panel izquierdo de phpMyAdmin para navegar a sus otras bases de datos o tablas.

Proteger los datos en transmisión

El protocolo HTTP no es inherentemente inmune a los bombardeos de la red (grabar datos delicados fuera de la conexión), así que si queremos proteger, no sólo nuestro nombre de usuario y nuestra contraseña, sino también todos los datos que viajan entre nuestro servidor Web y el navegador, tenemos que usar HTTPS.

Para hacerlo, asumiendo que nuestro servidor Web soporte HTTPS, sólo tenemos que iniciar phpMyAdmin escribiendo `https` en lugar de `http` en la URL, como aquí:

```
https://www.mydomain.com/phpMyAdmin
```

Si estamos usando la auto-detección `PmaAbsoluteUri`:

```
$cfg['PmaAbsoluteUri'] = '';
```

phpMyAdmin verá que estamos usando HTTPS en la URL y reaccionará de acuerdo con la información. Si no, debemos poner la parte `https` en el parámetro, como sigue:

```
$cfg['PmaAbsoluteUri'] = 'https://www.mydomain.com/phpMyAdmin';
```

Actualizar phpMyAdmin

De versión a versión, aparecen nuevos parámetros en `config.inc.php`. Si no queremos modificar sus valores por defecto, podremos utilizar los antiguos archivos de configuración, de esa mantendremos intactos las opciones por defecto en los nuevos parámetros. Por supuesto, si una nueva característica tiene un parámetro que tenemos que cambiar, sería más fácil comenzar con el nuevo fichero.

Se recomienda instalar la nueva versión en un directorio diferente, renombre `config.inc.php` a `config.inc.new.php` para tener una referencia y copie el antiguo `config.inc.php` en este directorio. Podemos hacer los ajustes necesarios (como por ejemplo `$cfg['PmaAbsoluteUri']`) y verificar la nueva instalación.

Se debe tener un cuidado especial al propagar los cambios que hayamos hecho al archivo `layout.inc.php` ya que depende de los temas que se hayan usado. Quizá tengamos que copiar nuestros subdirectorios de temas personalizados si hemos añadido nuestros temas a la estructura.

Resumen

Este capítulo explica:

- Las razones comunes para instalar phpMyAdmin.
- Los pasos para descargarlo desde el sitio principal, las configuraciones básicas y cargarlo en nuestro servidor Web.
- Usar una copia única de phpMyAdmin para administrar servidores múltiples.
- Usar tipos de autenticación para cubrir las necesidades de un grupo de usuarios mientras protegemos los datos.
- Asegurar la instalación de nuestro phpMyAdmin.
- Actualizar phpMyAdmin.

3

Información general de la interfaz

Paneles y ventanas

La interfaz de phpMyAdmin está compuesta de varios paneles y ventanas. Cada panel tiene una función específica, no es posible ver todos los paneles a la vez. Primero le daremos un poco de información general sobre cada panel y luego le ofreceremos información más detallada en este capítulo.

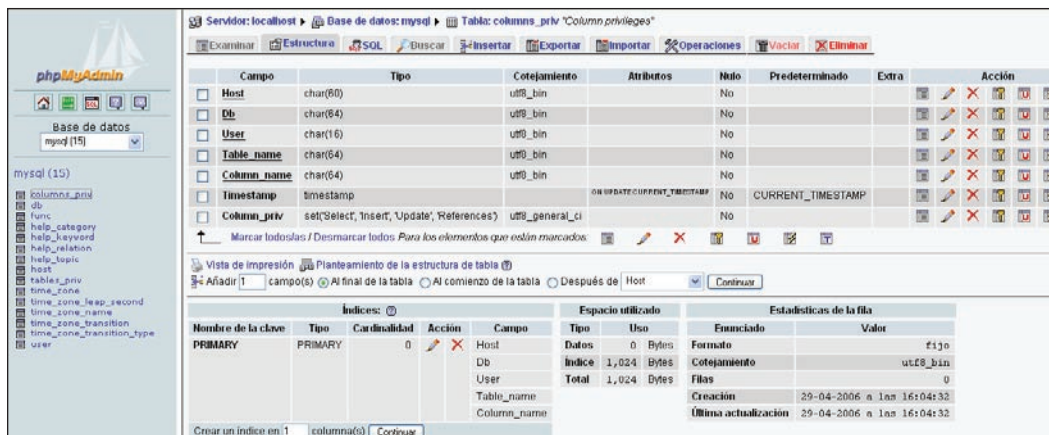
Paneles de inicio de sesión

El panel de inicio de sesión que aparece depende del tipo de autenticación que se haya escogido. Para el tipo `http`, tendrá el aspecto de la pantalla emergente de nuestro navegador HTTP. Para el tipo `cookie`, se mostrará el panel de inicio de sesión específico de phpMyAdmin (se ha visto en el capítulo 2). Por defecto, en este panel aparece un diálogo para elegir `Servidor` y un selector de `Idioma` en este panel.

Sin embargo, si usamos el tipo de autenticación `config`, no se muestra un panel de inicio de sesión, la interfaz que primero se muestra tiene los paneles de la izquierda y la derecha.

Paneles de la izquierda y de la derecha

Estos paneles van juntos y se muestran durante la mayor parte de nuestra sesión de trabajo con phpMyAdmin. El **panel de la izquierda** es nuestra guía de navegación a través de las bases de datos y las tablas. El **panel de la derecha** es la zona de trabajo principal donde se administran los datos y aparecen los resultados. Su aspecto exacto depende de las elecciones que se han hecho en el panel de la izquierda y de la secuencia exacta de operaciones que se han llevado a cabo.



Página de inicio

El panel de la derecha puede tener el aspecto de una página de **inicio**, que contiene varios enlaces a las operaciones de MySQL o información de phpMyAdmin, elección de **Idioma** y seguramente elección de temas.

Vistas

En el panel de la derecha podemos elegir la vista base de datos, en la que podemos hacer algunas acciones serias sobre una base de datos específica, o la vista tabla, en la que podemos acceder a las numerosas funciones para administrar una tabla. Un administrador del sistema también puede acceder a la vista de servidor. Todas estas vistas tienen un menú arriba, que tiene la forma de tabuladores que llevan a diferentes sub-páginas usadas para presentar la información reagrupada por funciones comunes (estructura de la tabla, privilegios, etc.).

Ventana consulta

Es una ventana distintiva que se puede abrir desde el panel de la izquierda o del de la derecha. Su objetivo principal es facilitar el trabajo en las búsquedas y ver los resultados en el panel de la derecha.

Página de arranque

Cuando iniciemos phpMyAdmin veremos lo siguiente (dependiendo del tipo de autenticación que se haya especificado en `config.inc.php` y de si tiene un servidor definido):

- Uno de los mensajes de inicio de sesión.
- El panel izquierdo y derecho, con la página de inicio en el panel de la izquierda.

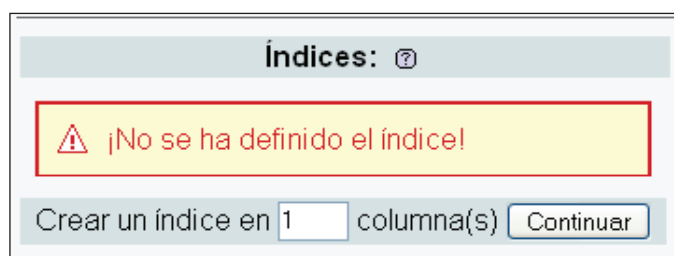
Configuración de los títulos de la ventana

Cuando se muestran los paneles de la izquierda y de la derecha, el título de la ventana cambia para reflejar *que* servidor MySQL, base de datos y tabla están activos. phpMyAdmin también muestra información sobre el proveedor del servidor Web si `$cfg['ShowHttpHostTitle']` está configurado `TRUE`. Cuando se muestra depende de otra configuración, `$cfg['SetHttpHostTitle']`. Si la configuración está vacía (como por defecto) el nombre verdadero del proveedor del servidor Web aparece en el título. Podemos poner otra cadena aquí, como por ejemplo 'mi servidor Web', y esto es lo que se mostrará en lugar del nombre verdadero del proveedor.

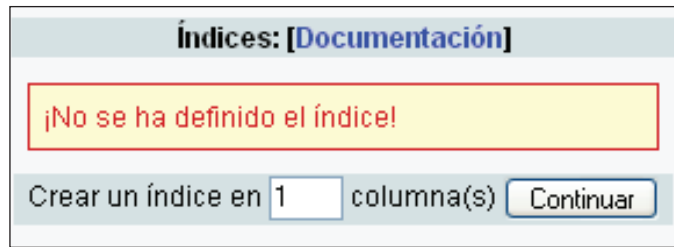
Ver el nombre del proveedor del servidor Web puede ser muy práctico cuando tenemos muchas ventanas phpMyAdmin abiertas, o sea, estar conectados a más de un servidor Web. Por supuesto, cada ventana phpMyAdmin puede dar acceso a numerosos servidores MySQL.

Configuración general de los iconos

Cuando aparecen varios mensajes de alarma, error o información, éstos pueden ir acompañados de un icono si `$cfg['ErrorIconic']` está configurado `TRUE`. Otro parámetro, `$cfg['ReplaceHelpImg']`, cuando se configura `TRUE`, muestra un pequeño icono que pone una pregunta en todos los sitios donde hay documentos disponibles para un tema específico. Estos dos parámetros se configuran `TRUE` por defecto, de modo que vemos:



Se pueden configurar independientemente `FALSE`. Si los configuramos `FALSE` veremos:



Clasificación natural para los nombres de las bases de datos y las tablas

Normalmente los ordenadores clasifican los objetos por orden alfabético, lo que da el siguiente resultado para una lista de tablas:

```
Table1
Table10
Table2
Table3
```

Ahora phpMyAdmin ha implementado el 'orden natural de clasificación' por defecto, como se especifica en `$cfg['NaturalOrder']` si es `TRUE`. Así pues, las listas de base de datos y tablas en los paneles de la izquierda y de la derecha están clasificadas así:

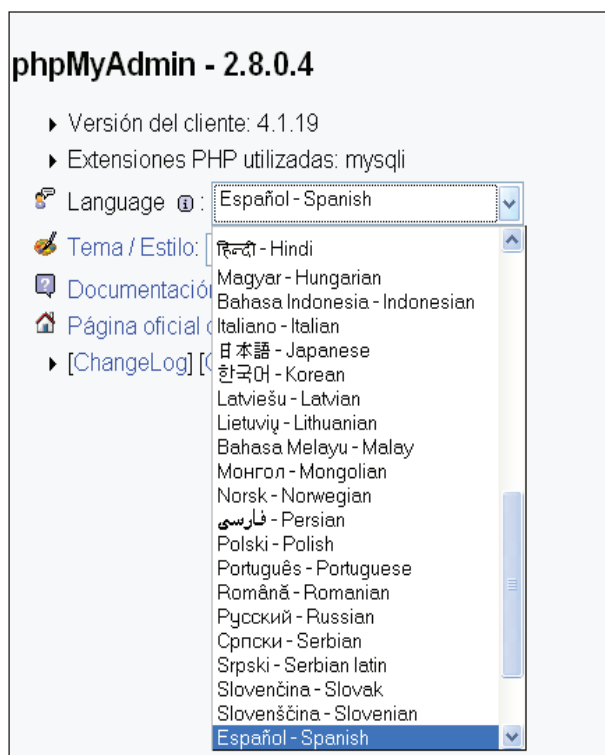
```
Table1
Table2
Table3
Table10
```

Selección de idioma

Aparece un selector de Idioma en el panel de inicio de sesión (si lo hay) y en la página de inicio. El procedimiento por defecto de phpMyAdmin es usar el idioma que está definido en las preferencias de nuestro navegador, si hay un archivo de idioma que corresponda para esta versión.

El idioma que se usa por defecto en el caso que el programa no pueda detectar, está definido en `config.inc.php` en el parámetro con `'en-iso-8859-1'`. Este valor se puede cambiar. Los posibles valores de los nombres de idioma están definidos en el programa `libraries/select_lang.lib.php` como un conjunto.

Incluso si el idioma por defecto está configurado, cada usuario (esto es especialmente verdad en una instalación multi-usuario) puede elegir su idioma preferido en el selector.



Siempre que sea posible, la elección del usuario se recordará en una cookie.

El pequeño icono de información que está bajo **Idioma** da acceso a la página de traductores de phpmyAdmin, que tiene una lista, por idiomas, del traductor oficial y de su información de contacto. De esta manera podemos contactar con el traductor para las correcciones y ofrecerle ayuda con los mensajes sin traducir. También podemos forzar un sólo idioma si configuramos el parámetro `$cfg['Lang']` con un valor, como por ejemplo `'en-iso-8859-1'`.

En la página **inicio**, también podemos ver un selector MySQL **Charset** o información **Charset** de MySQL (no está en un selector). Puede ir al capítulo 17, *Conjunto de caracteres y colaciones*, para tener más detalles sobre este tema.

Temas

Hay disponible un sistema de temas en phpMyAdmin que empieza con la versión 2.6.0. Los parámetros de color y los diferentes iconos están en una estructura bajo el subdirectorio `temas`. Por cada tema disponible hay un subdirectorio con el nombre del tema. Contiene:

- `Layout.inc.php` para los parámetros del tema.
- Directorio `css` con varios programas CSS.
- Directorio `img` que contiene los iconos.
- `Screen.png`, una muestra de pantalla de este tema.

Configuración de los temas

En `config.inc.php`, el parámetro `$cfg['ThemePath']` contiene 'temas' por defecto, lo que indica en *que* subdirectorio está localizada la estructura necesaria. Puede cambiarse para apuntar a otro subdirectorio en el que los temas específicos de nuestra compañía phpMyAdmin están localizados.

El tema elegido por defecto está especificado en `$cfg['ThemeDefault']`, y está configurado como 'original'. Si no hay una elección de temas disponible para los usuarios, se usará este tema.



El subdirectorio `original` no debe borrarse nunca; phpMyAdmin depende de él para las operaciones normales.

Elección de temas

En la página de **inicio** podemos ofrecer un selector de temas a nuestro usuarios: Configurar `$cfg['ThemeManager']` a `TRUE` (por defecto) muestra el selector:



Para ayudarnos a elegir el tema ideal, el icono paleta de colores que está al lado de **tema / estilo** nos enseña muestras de pantallas de los temas disponibles. Podemos hacer clic y ponerlo bajo el tema que queremos. El tema elegido se recuerda en una cookie.

Panel de la izquierda

El panel de la izquierda tiene los siguientes elementos:

- El logo (si `$cfg['LeftDisplayLogo']` está configurado `TRUE`), se puede hacer clic en él y se puede enlazar a <http://www.phpMyAdmin.net>.
- La lista de servidores (si `$cfg['LeftDisplayServers'] = TRUE;`).
- El enlace o icono a la **página de inicio** (le lleva a la página de inicio de phpMyAdmin).
- El enlace o icono **Salir**.
- Un enlace o icono que lleve a la ventana **Consulta**.
- Iconos para mostrar los documentos de phpMyAdmin y MySQL.
- La elección de bases de datos y tablas, con una estadística sobre el número de tablas por base de datos.

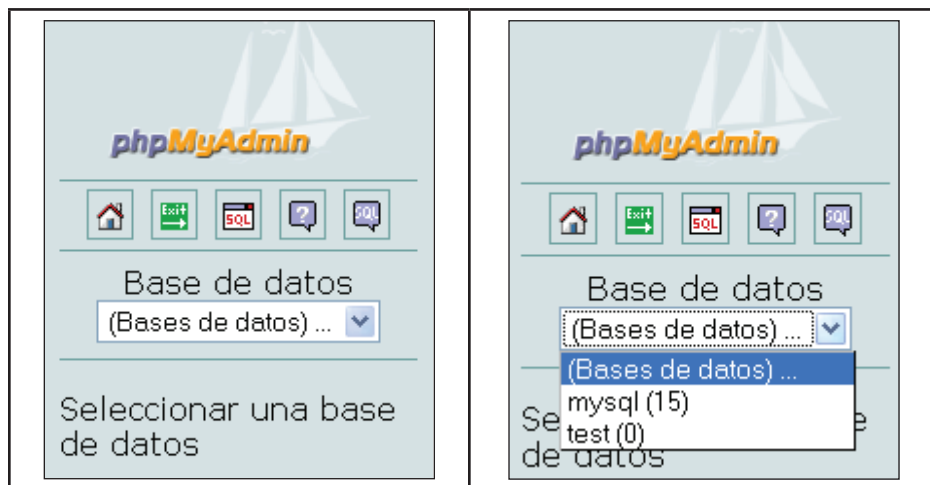
Si `$cfg['MainPageIconic']` está configurado `TRUE` (por defecto), vemos los *iconos*. Si está configurado `FALSE`, entonces vemos los enlaces **página de inicio**, **terminar sesión** y **consulta**.

Se puede ajustar el tamaño del panel de la izquierda haciendo clic y mover la línea vertical de separación en la dirección preferida, para revelar más datos en el caso que los nombres de las bases de datos o de las tablas sean demasiado largos para el tamaño por defecto del panel de la izquierda.

Podemos personalizar el aspecto de este panel: todos los parámetros están en `themes/themename/layout.inc.php`, excepto cuando se diga lo contrario. `$cfg['LeftWidth']` contiene la amplitud por defecto del marco de la izquierda, está en píxeles. El color de fondo está definido en `$cfg['LeftBgColor']` con un valor por defecto de `'#D0DCE0'`. El parámetro `$cfg['LeftPointerColor']`, con un valor por defecto de `'#CCFFCC'`, define el color del puntero (el puntero aparece cuando usamos el modo completo que se discutirá en breve). Para activar el puntero de la izquierda para cualquier tema usado, existe una configuración maestra en `config.inc.php` `$cfg['LeftPointerEnable']`. Su valor por defecto es `TRUE`.

Lista de bases de datos y tablas

Los siguientes ejemplos muestran que no se ha elegido ninguna base de datos en el menú desplegable.



También es posible que aparezca la siguiente pantalla:



Esto significa que nuestros derechos MySQL no nos permiten ver las bases de datos existentes.

[ Un servidor MySQL siempre tiene una base de datos como *mínimo* (llamada **mysql**), pero en este caso, puede que no tengamos derecho a verla.]

Puede que tengamos derecho a crear una, como se explica en el capítulo 4.

Modo instalación mínima

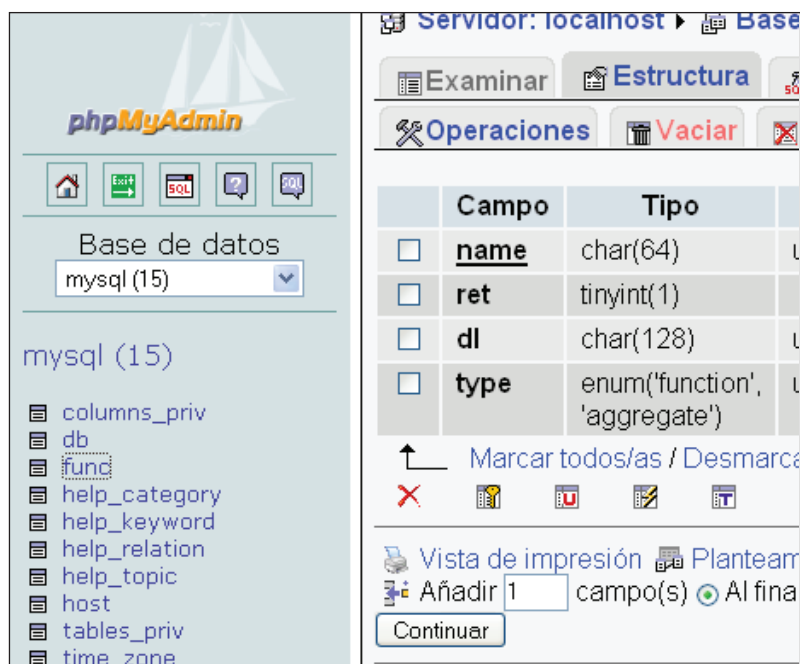
El panel de la izquierda puede tener dos formas: El modo instalación mínima o el modo completo. El modo instalación mínima se usa por defecto, definido por un valor `TRUE` en `$cfg['LeftFrameLight']`. El modo instalación mínima muestra una lista desplegable de las bases de datos disponibles, y sólo se muestran las tablas de las bases de datos elegidas (aquí hemos elegido la base de datos `mysql`):



Al hacer clic en el nombre de la base de datos se abre el panel de la derecha en la vista base de datos, al hacer clic en el nombre de la tabla se abre el panel de la derecha en la vista tabla (vea la sección *Panel de la derecha* para más información).

Atajo a las estadísticas de la tabla

Mover el cursor sobre el nombre de una tabla muestra comentarios sobre la tabla (si los hay) y el número de columnas que tiene:

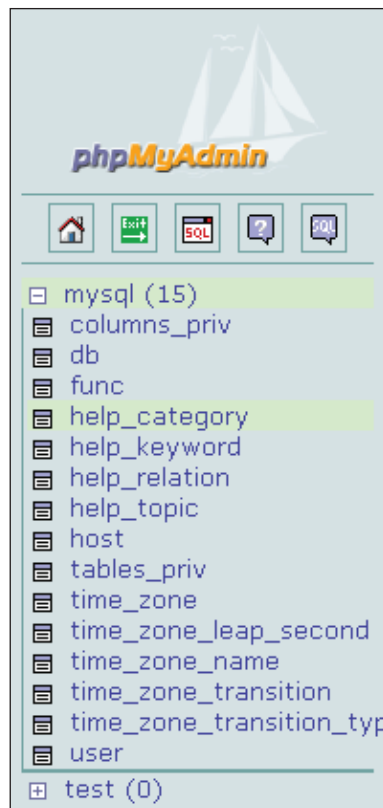


Examinar con rapidez una tabla

El pequeño icono que está debajo de los nombres de todas las tablas es una manera rápida de navegar por las columnas de las tablas. Abre el panel de la derecha en la vista tabla, examinando la primera página de datos de esa tabla.

Modo completo

Los anteriores ejemplos se han mostrado en el modo instalación mínima, pero si configuramos el parámetro `$cfg['LeftFrameLight'] FALSE`, obtendremos un diseño completo de nuestras bases de datos y tablas, haciendo uso de menús expandibles (si nuestro navegador los soporta):

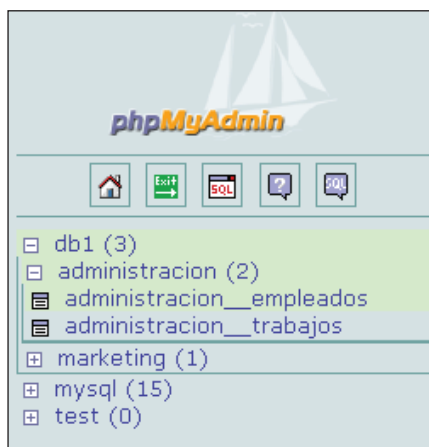


El número de tablas por base de datos se muestra entre paréntesis. El modo completo no se selecciona por defecto; puede incrementar el tráfico de la red y la carga del servidor si nuestros derechos actuales nos dan acceso a más bases de datos y tablas. Los enlaces se tienen que generar en el panel de la izquierda para habilitar el acceso a la tabla y acceso rápido a todas las tablas y además el servidor tiene que contar el número de columnas de todas las tablas.

Presentación jerárquica de tablas dentro de una base de datos

La estructura de datos MySQL está basada en dos niveles: Bases de datos y tablas. Esto no permite subdivisiones de tablas por proyecto, una característica que los usuarios de MySQL piden a menudo. Dependen en tener múltiples bases de datos, cosa que su proveedor no siempre les permite. Para ayudarles en esto, phpMyAdmin introduce la característica **Niveles jerárquicos**, basada en el nombre de la tabla. En estos momentos, esta característica sólo está disponible en el modo completo.

Digamos que tenemos acceso a la base de datos **db1** y a dos proyectos, **marketing** y **administración**. Si usamos un separador especial (por defecto subrayado doble) entre el nombre del proyecto y el nombre de la tabla, podemos conseguir un efecto visual interesante:

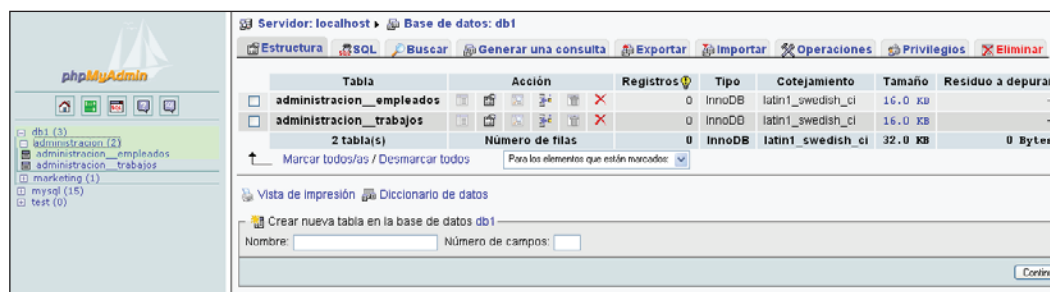


Esta característica se convierte en parámetro con `$cfg['LeftFrameTableSeparator']` (configurado en `'__'`) para elegir los caracteres que marcaran el cambio de cada nivel y `$cfg['LeftFrameTableLevel']` (configurado en `'1'`) para el número de sub-niveles.



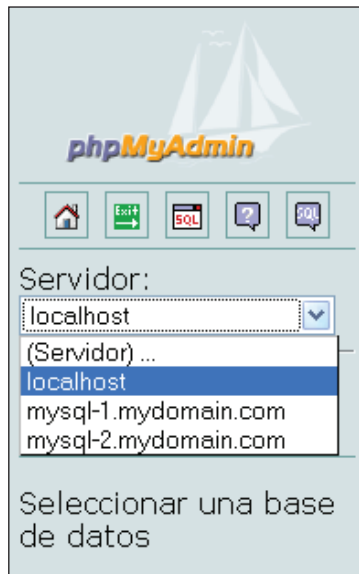
La característica nivel jerárquico tiene por objetivo mejorar el aspecto del panel de la izquierda. La manera adecuada de hacer referencia a las tablas con sentencias MySQL no cambia: por ejemplo, `db1.payroll__jobs`.

Si empezamos con phpMyAdmin 2.6.0, un clic en el panel de la izquierda en el *nombre del proyecto* (en este caso departamento contable) abre este proyecto en el panel de la derecha, mostrando sólo las tablas de este proyecto.

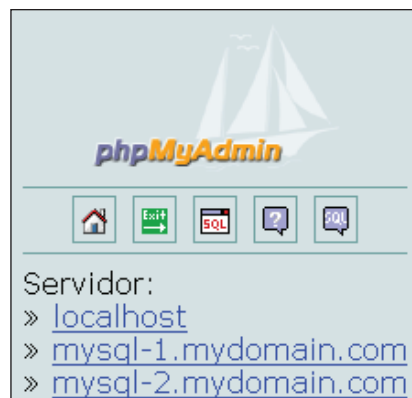


Elección de lista de servidores

Si tenemos que administrar múltiples servidores desde la misma ventana phpMyAdmin y necesitamos cambiar entre servidores a menudo, puede que sea útil ver la lista de servidores en el panel de la izquierda:



Para ello el parámetro `$cfg['LeftDisplayServers']` debe estar configurado en `TRUE`. La lista de servidores puede tener dos formas: Desplegable o enlaces. En que forma aparece depende de `$cfg['DisplayServersList']`. Por defecto, este parámetro está configurado `FALSE`, de modo que vemos una lista desplegable de los servidores. Configurar `$cfg['DisplayServersList']` en `TRUE` una lista de estilo enlaces:

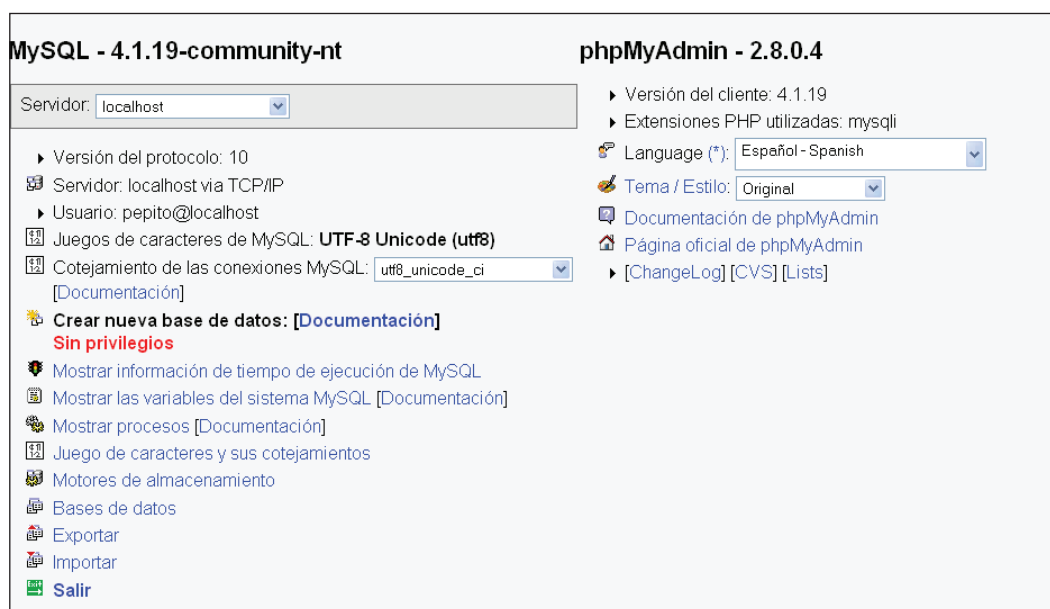


Panel de la derecha

El panel de la derecha es la zona principal de trabajo, las posibles vistas se explican en las secciones que siguen. Su aspecto se puede personalizar. El color de fondo está definido en `$cfg['RightBgColor']` con un valor por defecto de `#F5F5F5`. También podemos poner una imagen de fondo si configuramos el URI de la imagen que queremos (por ejemplo `http://www.mydomain.com/images/clouds.jpg`) en `$cfg['RightBgImage']`.

Página de inicio

La página de inicio puede tener un número variable de enlaces que dependen del modo de inicio de sesión y los derechos de usuario. Un usuario normal puede que la vea así:



El enlace **Inicio** del panel de la izquierda se usa para mostrar esta página. Muestra la versión de phpMyAdmin y MySQL, el nombre del servidor MySQL y el usuario que está conectado. También vemos que este usuario no tiene el privilegio de crear una base de datos. Vemos algunos enlaces que están relacionados con MySQL y phpMyAdmin. El enlace **Salir** puede que no aparezca si nos hemos conectado de forma automática, como se indicaba en el fichero de configuración.

En este ejemplo, un usuario normal no tiene permiso para cambiar su contraseña desde la interfaz. Para permitir el cambio de contraseña, configuramos `$cfg['ShowChgPassword'] = TRUE`. Los usuarios con privilegios tienen más opciones en la página de inicio. Siempre pueden crear bases de datos y tienen más enlaces para administrar el servidor (vista servidor):



Vista base de datos

phpMyAdmin entra en esta vista (que vemos en la muestra de pantalla que sigue) cada vez que hacemos clic en un nombre de base de datos del recuadro de la izquierda, o si escribimos el comando `USE` seguido de un nombre de base de datos en la caja SQL.

Es aquí donde podemos obtener una vista general de la base de datos: Las tablas existentes, el enlace para crear una tabla, los tabuladores de la vista base de datos para las sub-páginas y algunas operaciones especiales que podemos hacer en las bases de datos para generar documentos y estadísticas. Hay una casilla de control bajo cada tabla para hacer operaciones globales en esas tablas (está cubierto en el capítulo 10). La tabla se escoge con la casilla de control o haciendo clic en cualquier parte del fondo de pantalla. También podemos ver el tamaño de todas las tablas si `$cfg['ShowStats']` está configurado `TRUE`. Este parámetro también controla el aspecto de las estadísticas específicas de una tabla en la vista tabla.

En esta pantalla, vemos un mensaje de error. Podemos ignorarlo con seguridad por ahora ya que en la sección *Sistema relacional* (capítulo 11) le mostraremos la configuración inicial para las características especiales de tablas enlazadas.

La pantalla inicial que aparece aquí es la sub-página **estructura** de la base de datos. Puede ser que queramos ver una sub-página inicial diferente cuando entremos en la vista base de datos. Esto se puede controlar. Con el parámetro `$cfg['DefaultTabDatabase']`, las elecciones disponibles aparecen en configuración en forma de comentarios.

Servidor: localhost Base de datos: mysql

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios

Eliminar

	Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
☐	columns_priv		0	MyISAM	utf8_bin	1.0 KB	-
☐	db		2	MyISAM	utf8_bin	4.8 KB	-
☐	func		0	MyISAM	utf8_bin	1.0 KB	-
☐	help_category		34	MyISAM	utf8_general_ci	4.0 KB	-
☐	help_keyword		357	MyISAM	utf8_general_ci	19.3 KB	-
☐	help_relation		641	MyISAM	utf8_general_ci	16.6 KB	-
☐	help_topic		415	MyISAM	utf8_general_ci	208.5 KB	-
☐	host		0	MyISAM	utf8_bin	1.0 KB	-
☐	tables_priv		0	MyISAM	utf8_bin	1.0 KB	-
☐	time_zone		0	MyISAM	utf8_general_ci	1.0 KB	-
☐	time_zone_leap_second		0	MyISAM	utf8_general_ci	1.0 KB	-
☐	time_zone_name		0	MyISAM	utf8_general_ci	1.0 KB	-
☐	time_zone_transition		0	MyISAM	utf8_general_ci	1.0 KB	-
☐	time_zone_transition_type		0	MyISAM	utf8_general_ci	1.0 KB	-
☐	user		3	MyISAM	utf8_bin	2.2 KB	-
	15 tabla(s)	Número de filas	1,452	InnoDB	latin1_swedish_ci	264.4 KB	0 Bytes

Marcar todas/as / Desmarcar todos Para los elementos que están marcados:

El número de registros se obtiene gracias un rápido método: No haciendo un `SELECT COUNT (*) FROM NOMBRE_DE_TABLA`. Este rápido método normalmente es preciso, a excepción de las tablas InnoDB, para las cuales devuelve un número aproximado de registros. Hay un parámetro disponible para obtener el número correcto de registro, incluso para InnoDB, el `$cfg['MaxExactCount']`. Si el número aproximado de registros es más bajo que el valor de este parámetro, por defecto 20000; se usará el método `SELECT COUNT (*)` que es más lento.



No ponga un valor demasiado alto para este parámetro. Podría obtener resultados correctos, pero sólo después de esperar unos minutos, si hay miles de miles registros en sus tabla InnoDB.

Vista tabla

Es una vista que se usa habitualmente y que da acceso a todas las sub-páginas específicas de las tablas. Normalmente, la pantalla inicial es la pantalla **Estructura** de tabla, que muestra (tenga en cuenta la parte superior) todos los campos e índices. Tenga en cuenta que la cabecera de esta pantalla siempre muestra el nombre de la base de datos y la tabla que se están usando en ese momento. También podemos ver los comentarios escritos de la tabla:

Servidor: localhost ▶ Base de datos: mysql ▶ Tabla: columns_priv "Column privileges"

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
☐ Host	char(60)	utf8_bin		No			[icon]
☐ Db	char(64)	utf8_bin		No			[icon]
☐ User	char(16)	utf8_bin		No			[icon]
☐ Table_name	char(64)	utf8_bin		No			[icon]
☐ Column_name	char(64)	utf8_bin		No			[icon]
☐ Timestamp	timestamp		ON UPDATE CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP		[icon]
☐ Column_priv	set('Select', 'Insert', 'Update', 'References')	utf8_general_ci		No			[icon]

Marcar todos/as / Desmarcar todos Para los elementos que están marcados: [icon] [icon] [icon] [icon] [icon] [icon]

Vista de impresión Planteamiento de la estructura de tabla

Añadir 1 campo(s) ☐ Al final de la tabla ☐ Al comienzo de la tabla ☐ Después de Host Continuar

Índices				Espacio utilizado		Estadísticas de la fila	
Nombre de la clave	Tipo	Cardinalidad	Acción	Tipo	Uso	Enunciado	Valor
PRIMARY	PRIMARY	0	[icon] [icon]	Datos	0 Bytes	Formato	fijo
			Host	Índice	1,024 Bytes	Cotejamiento	utf8_bin
			Db	Total	1,024 Bytes	Filas	0
			User			Creación	29-04-2006 a las 16:04:32
			Table_name			Última actualización	29-04-2006 a las 16:04:32
			Column_name				

Crear un índice en 1 columna(s) Continuar

La parte inferior de la página **Estructura** nos permite ejecutar consultas e insertar datos desde un fichero fuente exterior. Todas estas funciones se discutirán en los siguientes capítulos.

Servidor: localhost ▶ Base de datos: mysql ▶ Tabla: columns_priv "Column privileges"

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Ejecutar la(s) consulta(s) SQL en la base de datos mysql:

```
SELECT * FROM `columns_priv` WHERE 1
```

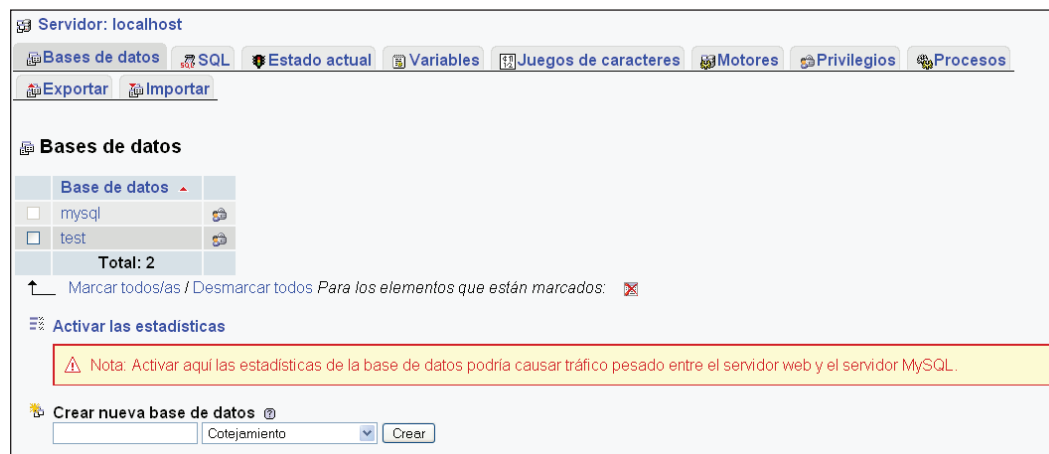
Campos
Host
Db
User
Table_name
Column_name
<<

☒ Mostrar esta consulta otra vez Continuar

El parámetro `cfg['DefaultTabTable']` define la sub-página inicial de la vista tabla. Algunos usuarios prefieren no ver la estructura, porque cuando están trabajando, ejecutan consultas guardadas o entran en la sub-página **Buscar** (se explicará en el capítulo 9) de forma rutinaria.

Vista servidor

Esta vista se muestra cada vez que elegimos una opción relacionada con MySQL desde la página inicial: Por ejemplo, **base de datos** o **ver información del tiempo de ejecución de MySQL**. Por supuesto, un usuario con privilegios verá más opciones en la vista servidor. El panel vista servidor se creó para agrupar la administración de todas sub-páginas del servidor y permitir una fácil navegación entre ellos.



La página servidor por defecto está controlada por el parámetro `$cfg['DefaultTabServer']`. Este parámetro también define la página inicial de inicio. Para las instalaciones multi-usuario, se recomienda mantener el valor por defecto (`main.php`) que muestra la página de inicio tradicional. Podríamos elegir mostrar las estadísticas del servidor en lugar de cambiar este parámetro a `server_status.php`, o ver la lista de usuarios con `server_privileges.php`. Hay otras elecciones posibles que están explicadas en el archivo de configuración y las páginas de administración del servidor están explicadas en el capítulo 18.

Iconos para la página de inicio y menú tabulador

Un parámetro de configuración, `$cfg['MainPageIconic']`, controla la aparición de los iconos en diferentes lugares del panel de la derecha:

- En la página inicial.
- En la parte superior de la página, cuando se hace la lista **servidor**, **base de datos** y **tabla**.
- En los tabuladores del menú en las vistas base de datos, tabla y servidor.

Este parámetro está configurado TRUE por defecto, produciendo, por ejemplo:



Ventana consulta

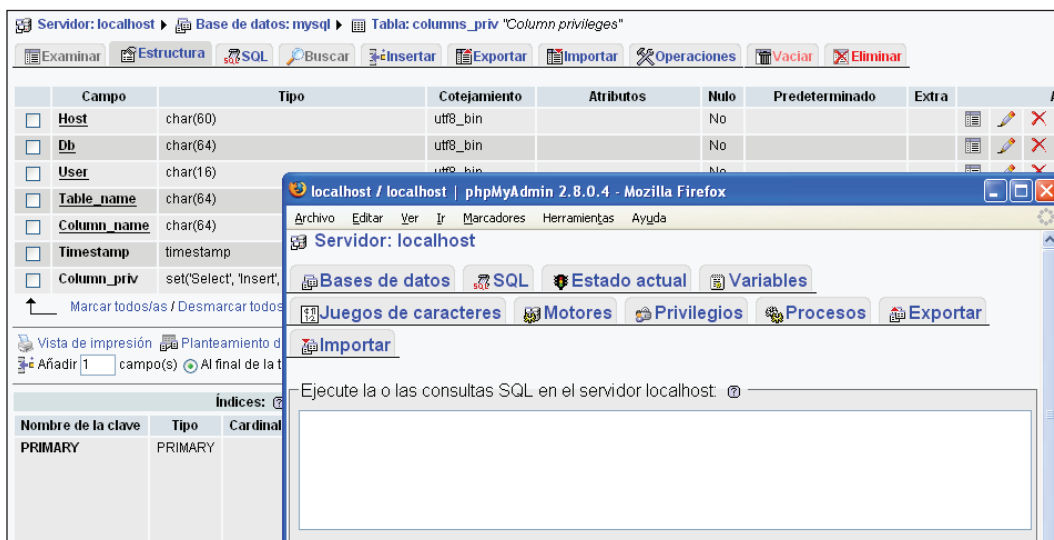
A menudo es conveniente tener una *ventana distintiva* en la que podamos escribir y refinar consultas, y que esté sincronizada con el panel de la derecha. Esta ventana se llama **ventana consulta**. Podemos abrir esta ventana si usamos el pequeño icono SQL o el enlace **ventana consulta** de los iconos del panel de la izquierda o de la zona de enlaces.

Este enlace o icono se muestran si `$cfg['QueryFrame']` está configurado TRUE. El TRUE de `$cfg['QueryFrameJS']` le dice a phpMyAdmin que debe abrir una ventana diferente y que debe actualizarla usando comandos de JavaScript; por supuesto, esto sólo funciona en un navegador que permite JavaScript. Si está configurado FALSE, hacer clic en la **ventana consulta** sólo abrirá el panel de la derecha y mostrará la sub-página SQL normal.



Sólo se puede aprovechar toda la utilidad de la **ventana consulta** con el modo ventana diferente.

La **ventana consulta** tiene sub-páginas que aparecen sobre el panel de la derecha:



Podemos elegir las dimensiones de esta ventana (en píxeles) con `$cfg['QueryWindowWidth']` y `$cfg['QueryWindowHeight']`. El capítulo 12 explica la ventana consulta con más detalle, incluyendo las características SQL de historial de consultas, disponibles.

Cabecera y pie de página específicos de un sitio

Algunos usuarios quieren presentar el logo de la compañía, un enlace al equipo de soporte técnico u otra información en la interfaz de phpMyAdmin. En el directorio principal phpMyAdmin hay dos programas disponibles para este propósito: `config.header.inc.php` y `config.footer.inc.php`. Podemos añadir nuestro código PHP o XHTML en estos programas, y aparecerá al principio (para la cabecera) o al terminar la página (pie de página).

- En la cookie de la página de inicio de sesión.
- En el panel de la derecha.

Enlaces a los documentos MySQL

phpMyAdmin presenta enlaces a los documentos de MySQL en varios lugares de su interfaz. Estos enlaces hacen referencia al punto exacto donde se puede aprender un comando MySQL en la documentación oficial de MySQL. Podemos personalizar la posición, el idioma y el tipo de manual al que hacen referencia, con los siguientes parámetros de configuración:

```
$cfg['MySQLManualBase'] = 'http://www.mysql.com/doc/en';  
$cfg['MySQLManualType'] = 'searchable';
```

Puede mirar en <http://www.mysql.com/documentation> para ver en que idiomas está disponible el manual y cambiar los parámetros de acuerdo con esta información. Para el tipo de manual, los posibles valores más actualizados están explicados como comentarios en `config.inc.php`. Los usuarios que prefieran tener una copia de estos documentos en un servidor local deberán especificar un enlace local aquí.

El parámetro `$cfg['ReplaceHelpImg']` controla la presentación de los enlaces. Su valor por defecto de `TRUE` hace que phpMyAdmin presente pequeños iconos pregunta y `FALSE` muestra los enlaces **documentación**.

Resumen

En este capítulo hemos cubierto:

- El sistema de selección de idioma.
- El propósito de los paneles de la izquierda y de la derecha.
- Los contenidos del panel de la izquierda, incluyendo el modo instalación mínima y el modo completo.
- Los contenidos del panel de la derecha, con sus diferentes vistas que dependen del contexto.
- La **ventana consulta**.
- Personalización de los enlaces a los documentos MySQL.

4

Primeros Pasos

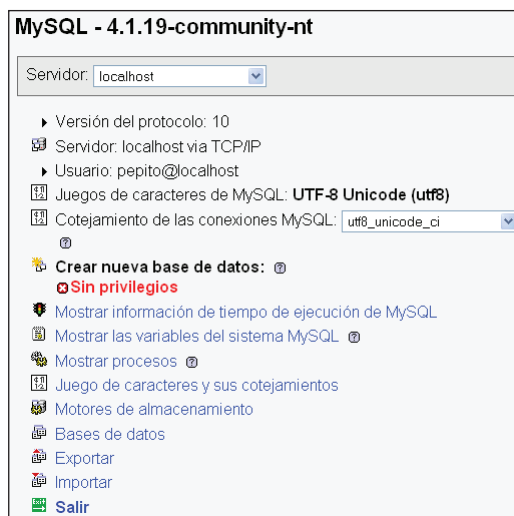
Creación de bases de datos

Como ya hemos visto el diseño general del panel de phpMyAdmin, estamos preparados para: Crear una base de datos, nuestra primera tabla, insertar algunos datos y examinarla. Antes de crear una tabla, debemos asegurarnos que tenemos una base de datos para la que el administrador del servidor MySQL nos ha dado permisos `CREATE`. Existen varias posibilidades:

- El administrador ya ha creado una base de datos, podemos ver su nombre en el panel de la izquierda; no tenemos permiso para crear una base de datos adicional.
- Tenemos permiso para crear bases de datos desde phpMyAdmin.
- Estamos en un servidor compartido y el proveedor ha instalado una interfaz Web general (por ejemplo, Cpanel) para crear bases de datos y cuentas MySQL.

¿Sin permisos?

En este caso, la página de **inicio** tiene el siguiente aspecto:



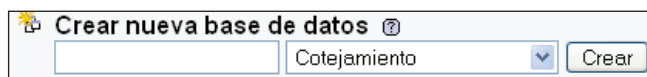
Esto quiere decir que tenemos que trabajar con las bases de datos que nos han creado, o pedirle al administrador del servidor MySQL que nos de los permisos CREATE necesarios.



En el caso que usted sea el administrador de MySQL, por favor vaya al capítulo 18, *Administración del servidor MySQL con phpMyAdmin*.

La creación de la primera base de datos está autorizada

Si phpMyAdmin detecta que tenemos permiso para crear una base de datos, la página de inicio tiene el mismo aspecto que la siguiente figura:



Si nos restringen a usar un prefijo, el prefijo se nos sugerirá en el campo de entrada (una elección popular para este prefijo es el nombre de usuario).

Asumiremos que tenemos permiso para crear una base de datos llamada dbbook. Escribimos **dbbook** en el campo de entrada y hacemos clic en Crear. Una vez que se ha creado la base de datos, veremos la siguiente pantalla:



Debe darse cuenta de lo siguiente:

- El título principal del panel de la derecha ha cambiado para reflejar el hecho que ahora estamos en una base de datos.
- Aparece un mensaje de confirmación relacionado con la creación de la base de datos.

El panel de la izquierda se ha actualizado; ahora vemos **dbbook (-)**. Aquí, el nombre indica que la base de datos dbbook se ha creado y el símbolo indica que no tiene tablas.

- Por defecto, la consulta SQL que phpMyAdmin ha enviado para crear la base de datos, se muestra en color.

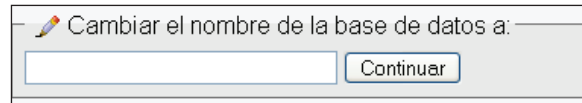
[ phpMyAdmin siempre muestra la consulta que ha generado porque `$cfg['ShowSQL']` está configurado TRUE. Ver las consultas que se han generado puede ser una buena manera de aprender SQL.]

Es importante examinar los comentarios de phpMyAdmin para asegurarnos de la validez de las operaciones que hacemos a través de la interfaz. De este modo, podemos detectar errores como por ejemplo errores tipográficos en los nombres o la creación de una tabla en la base de datos incorrecta.

Renombrar bases de datos

Empezamos con phpMyAdmin 2.8.0.4, veremos un diálogo **Renombrar base de datos** en la vista base de datos. A pesar que esta acción no está directamente soportada por MySQL, phpMyAdmin lo hace de forma indirecta al crear una base

de datos nueva, renombrando cada tabla; así pues mandándolas a la nueva base de datos; y al final elimina la base de datos original.



Cambiar el nombre de la base de datos a:

Crear nuestra primera tabla

Ahora que tenemos una base de datos nueva, tenemos que crear una tabla en ella. La tabla de ejemplo que usaremos, es la familiar tabla **libros**.

Elegir los campos

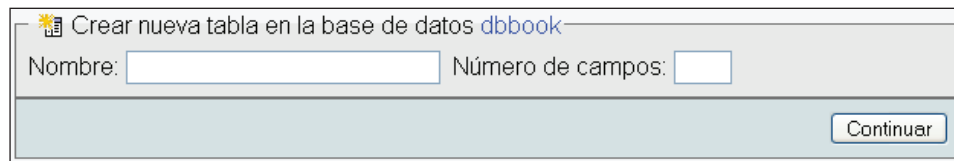
Antes de crear una tabla, deberemos pensar en que información queremos almacenar. Normalmente esto se hace con el uso de diseño de la base de datos, diagrama de relación entre las entidades y normalización de la tabla. En nuestro caso, un análisis sencillo nos lleva a guardar los siguientes datos relacionados con el libro:

- Número de libro estándar internacional (ISBN).
- Título del libro.
- Número de páginas.
- Identificación del autor.

Por ahora, no es importante tener la lista completa de campos (o columnas) de nuestra tabla **libros** ; la modificaremos al hacer una aplicación prototipo que refinaremos más tarde. Al final del capítulo, añadiremos una segunda tabla, **autores**, que tendrá información sobre cada autor.

Creación de la tabla

Hemos elegido el nombre de nuestra tabla y sabemos cuantos campos tiene. Escribimos esta información en el diálogo **Crear tabla nueva** y hacemos clic en **Continuar** para empezar a crear la tabla:



Crear nueva tabla en la base de datos dbbook

Nombre: Número de campos:

Luego vemos el panel para especificar la información de los campos. Como hemos pedido cuatro campos, veremos cuatro filas, cada fila se refiere a la información específica de un campo:

Servidor: localhost ▶ Base de datos: dbbook ▶ Tabla: libros

Campo	Tipo ^⑦	Longitud/Valores ^{*1}	Cotejamiento	Atributos	Nulo	Predeterminado ²
	VARCHAR				not null	
	VARCHAR				not null	
	VARCHAR				not null	
	VARCHAR				not null	

Comentarios de la tabla:

Motor de almacenamiento:

Cotejamiento:

O Añadir 1 campo(s)



Puede que en este momento aparezca otra columna, **Colación**. Por favor vaya al capítulo 17, *Conjunto de caracteres y colaciones*.

Los documentos MySQL explican cuáles son los caracteres válidos para los nombres de tabla y campo (si buscamos **Nombres legales**). Esto puede variar dependiendo de la versión MySQL. Normalmente, cualquier carácter que se permite en un nombre de archivo (exceptuando el punto y la coma) es válido para un nombre de tabla, el tamaño del nombre no debe exceder los 64 caracteres. Este límite de 64 caracteres también existe para los nombres de campos; sin embargo podemos usar cualquier carácter.

Escribimos el nombre de nuestros campos en la columna **Campo**. Cada campo tiene un tipo, el tipo **VARCHAR** (carácter variable) que aparece por defecto porque es el que se usa más comúnmente. El tipo VARCHAR se usa normalmente cuando el contenido del campo es alfanumérico, ya que los contenidos sólo ocuparán el espacio que necesitan. Este tipo necesita una longitud máxima que tenemos que especificar. Si nos olvidamos, un pequeño mensaje emergente nos lo recordará cuando grabemos. Para el número de páginas y la identificación del autor, hemos elegido el tipo **INT** (número entero), que se ve en la siguiente muestra de pantalla:

Campo	Tipo ^⑦	Longitud/Valores ^{*1}
isbn	VARCHAR	25
título	VARCHAR	100
num_págs	INT	
autor_id	INT	

Hay otros campos atributos, pero los dejaremos vacíos en este breve ejemplo. Debe mirar los **campos**: hay 4 diálogos en la parte superior de la pantalla. Esto no es tan sólo para recordarnos cuantos campos hemos pedido hacer. De hecho, podemos usarlo para cambiar el número de campos que queremos (normalmente para aumentarlo) y hacer clic en **continuar**. El número de filas cambiará de acuerdo con el número de campos, manteniendo intacta la información que ya se ha escrito sobre los cuatro primeros campos. Antes de guardar la página, definamos las claves.

Elegir las claves

Normalmente, una tabla debería tener una clave primaria (un campo con un contenido único que representa cada fila). Se recomienda tener una clave primaria para identificar las filas, para un mejor rendimiento y para hacer posibles relaciones cruzadas entre tablas. Un valor es ISBN, ahora seleccionamos **Primaria** para el campo **isbn**. Como `$cfg['PropertiesIconic']` está configurado TRUE por defecto, vemos iconos que indican las diferentes posibilidades de los índices. Si movemos el ratón sobre los botones de selección veremos: **primaria**, **índice**, **único** y **texto completo**.

 La administración del índice (también llamado Clave) se puede hacer al crear la tabla, o más tarde en la sub-página **Estructura** o la vista tabla.

Para mejorar la velocidad de las consultas haremos un ID del autor, deberíamos añadir un índice en este campo. Nuestra pantalla tiene este aspecto:

Campo	Tipo	Longitud/valores	Cotización	Atributos	Índice	Predefinido	Extra	Comentarios
ISBN	VARCHAR	25			not null			
Título	VARCHAR	100			not null			
Número año	INT				not null			
Subj. ID	INT				not null			

Comentarios de la tabla:

Motor de almacenamiento:

Cotización:

En este punto podríamos cambiar el tipo de tabla haciendo uso del menú desplegable **Tipo**, pero por ahora aceptaremos el tipo por defecto.

Ahora ya estamos preparados para crear la tabla haciendo clic en **Guardar**. Si todo va bien, la siguiente pantalla confirma que la tabla se ha creado; ahora estamos en la sub-página **Estructura** de la vista tabla.

Si olvidamos especificar un valor en la columna **Longitud** para CHAR y VARCHAR, phpMyAdmin nos lo recordará antes de crear la tabla.

De los diferentes tabuladores que llevan a otras sub-páginas, algunos no están activos porque no tendría sentido examinar todas las tablas si no tienen filas. Sin embargo, se podría exportar, ya que podemos exportar la estructura de una tabla a pesar que no tenga datos.

Servidor: localhost ▶ Base de datos: dbbook ▶ Tabla: libros

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Tabla libros se creó.

consulta SQL:

```
CREATE TABLE `libros` (
  `ISBN` VARCHAR(25) NOT NULL ,
  `Título` VARCHAR(100) NOT NULL ,
  `número_pág` INT NOT NULL ,
  `autor_id` INT NOT NULL ,
  PRIMARY KEY (`ISBN`),
  INDEX (`autor_id`)
) ENGINE = INNODB;
```

[Editar] [Crear código PHP]

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
☐	ISBN	varchar(25)	latin1_swedish_ci		No			      
☐	Título	varchar(100)	latin1_swedish_ci		No			      
☐	número_pág	int(11)			No	0		      
☐	autor_id	int(11)			No	0		      

↑ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:     

Vista de impresión Planteamiento de la estructura de tabla

Añadir 1 campo(s) ☐ Al final de la tabla ☐ Al comienzo de la tabla ☐ Después de ISBN Continuar

Índices: 2					Espacio utilizado		Estadísticas de la fila	
Nombre de la clave	Tipo	Cardinalidad	Acción	Campo	Tipo	Uso	Enunciado	Valor
PRIMARY	PRIMARY	0		ISBN	Datos	0 Bytes		
autor_id	INDEX	0		autor_id	Índice	0 Bytes		
Crear un índice en 1 columna(s) Continuar					Total	0 Bytes		

Inserción manual de datos

Ahora que tenemos una tabla, escribamos algunos datos de forma manual. Antes de hacerlo, aquí hay algunas referencias útiles sobre como manipular datos que están en este libro:

- El capítulo 5 explica como cambiar los datos.
- El capítulo 8 explica como importar datos desde archivos existentes.
- El capítulo 10 explica como copiar datos de otras tablas.

El capítulo 11 explica el sistema relacional (en nuestro caso, queremos enlazar la tabla **autores**).

Por ahora, haga clic en el enlace **Insertar**, que nos llevará al panel de escritura de datos (o su edición) que se muestra en la pantalla que sigue.

Podemos escribir los siguientes datos sobre estos dos libros:

- ISBN: 1-234567-89-0, título: Cien años de cine (volumen 1), 600 páginas, ID del autor: 1
- ISBN: 1-234567-22-0, título: Recuerdos futuros, 200 páginas, ID del autor: 2

Servidor: localhost Base de datos: dbbook Tabla: libros "InnoDB free: 4096 kB"

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Campo	Tipo	Función	Nulo	Valor
ISBN	varchar(25)			
Título	varchar(100)			
número_pág	int(11)			0
autor_id	int(11)			0

☒ Ignorar

Campo	Tipo	Función	Nulo	Valor
ISBN	varchar(25)			
Título	varchar(100)			
número_pág	int(11)			0
autor_id	int(11)			0

Insertar como una nueva fila y luego Volver

Continuar Reiniciar

Esta pantalla tiene espacio para escribir información en dos columnas: Dos libros. Esto pasa porque el valor por defecto de `$cfg['InsertRows']` es 2. Por defecto, el recuadro **Ignorar** está marcado, lo que quiere decir que el segundo grupo de campos se ignorará. Pero tan pronto como escribimos información en un campo de este grupo y salimos del campo, el recuadro **Ignorar** deja de estar marcado.

Empezamos escribiendo datos en las columnas primera y segunda. La columna **Valor** responde a la longitud máxima de los campos de caracteres. Si queremos escribir datos para más libros después de esos dos, seleccionamos **Insertar otra columna nueva**. Luego hacemos clic en **Continuar** para escribir datos:

Campo	Tipo	Función	Nulo	Valor
ISBN	varchar(25)		☐	1-234567-89-0
Título	varchar(100)		☐	Cien años de cine (volumen 1)
número_pág	int(11)		☐	0
autor_id	int(11)		☐	0

☐ Ignorar

Campo	Tipo	Función	Nulo	Valor
ISBN	varchar(25)		☐	1-234567-22-0
Título	varchar(100)		☐	Recuerdos futuros
número_pág	int(11)		☐	0
autor_id	int(11)		☐	0

y luego

Ajustar el panel de entrada de datos para chars y varchars

Por defecto, phpMyAdmin presenta el campo de entrada de datos en una línea para los tipos CHAR y VARCHAR. Esto se puede controlar si configuramos `$cfg['CharEditing']` en 'input'. Algunas veces queremos insertar caídas de líneas (nuevas líneas) en el campo (esta inserción se puede hacer de forma manual con la tecla *Intro*, o copiando y pegando líneas de texto de otra fuente que esté en la pantalla). Esto se puede controlar si configuramos `$cfg['CharEditing']` en 'textarea'. Esta es una configuración global que se aplicará a todos los campos, todas las tablas y todos los usuarios de esta copia de phpMyAdmin.

Podemos ajustar el número de columnas y filas en esta área de texto con:

```
$cfg['CharTextareaCols'] = 40;
$cfg['CharTextareaRows'] = 2;
```

Aquí el 2 del `$cfg['CharTextareaRows']` quiere decir que deberíamos ver al menos dos líneas antes que el navegador muestre una barra vertical de desplazamiento. Esta configuración se aplica a todos los campos CHAR y VARCHAR, usarlos generaría una pantalla **Insertar** diferente, como la que sigue:

Valor

1-234567-89_0

Cien años de cine (volumen 1)

0

0



Con este modo de escritura, la longitud máxima de cada campo ya no se aplica visualmente, pero MySQL lo reforzaría a la hora de INSERT.

Modo examinar

Hay muchas maneras de entrar en este modo. De hecho se usa cada vez que se presentan los resultados de las consultas. Podemos entrar a este modo manualmente usando el icono de guardar rápido del panel de la izquierda o yendo a la vista tabla específica y haciendo clic en **Examinar**:

Servidor: localhost
Base de datos: dbbook
Tabla: libros "InnoDB free: 4096 kB"

Examinar
Estructura
SQL
Buscar
Insertar
Exportar
Importar
Operaciones
Vaciar
Eliminar

Mostrando registros 0 - 1 (2 total, La consulta tardó 0.0141 seg)

consulta SQL:

```
SELECT *
FROM 'libros'
LIMIT 0 , 30
```

[Editar]
[Explicar el SQL]
[Crear código PHP]
[Actualizar]

Mostrar: 30
filas empezando de 0

en modo horizontal
y repetir los encabezados cada 100 celdas

Organizar según la clave: Ninguna
Continuar

	ISBN	Título	número_pág	autor_id
☐	1-234567-22-0	Recuerdos futuros	0	0
☐	1-234567-89-0	Cien años de cine (volumen 1)	0	0

↑
Marcar todos/as / Desmarcar todos
Para los elementos que están marcados:

Mostrar: 30
filas empezando de 0

en modo horizontal
y repetir los encabezados cada 100 celdas

Insertar nueva fila
Vista de impresión
Previsualización para imprimir (documento completo)
Exportar

El recuadro que está al lado de cada fila de resultados y el menú **Amplitud seleccionada** se explicarán en el capítulo 5.

Enlaces a las consultas SQL

En los resultados **Examinar**, la parte que se presenta primero es la consulta en sí junto con algunos enlaces. Los enlaces que se muestran varían, dependiendo de nuestras acciones y nuestros parámetros de configuración.

Mostrando registros 0 - 1 (2 total, La consulta tardó 0.0355 seg)

consulta SQL:

```
SELECT *
FROM `libros`
LIMIT 0 , 30
```

[Editar] [Explicar el SQL] [Crear código PHP] [Actualizar]

El enlace **Editar** aparece si `cfg['SQLQuery']['Edit']` está configurado TRUE. Su propósito es abrir la **ventana de consultas**, poder editar esa consulta (vaya al capítulo 12, Escribir comandos SQL).

El enlace **SQL** aparece si `cfg['SQLQuery']['Explain']` está configurado TRUE. En el capítulo 6, Cambiar la estructura de la tabla, veremos para que se puede usar este enlace.

Se puede hacer clic en el enlace **Crear código PHP** para reformatear la sintaxis en la sintaxis que se espera de un programa PHP. Se puede copiar y pegar directamente el lugar donde necesitamos la consulta, en el programa PHP con el que estamos trabajando. Debe darse cuenta que después de hacer clic a este enlace cambiará a **Sin código PHP**, lo que le devolverá a la presentación normal de una consulta. Este enlace está disponible si `cfg['SQLQuery']['ShowAsPHP']` está configurado TRUE.

Crear código PHP

consulta SQL:

```
$sql = 'SELECT * FROM `libros` LIMIT 0, 30 ';
```

[Editar] [Explicar el SQL] [Sin código PHP] [Ejecutar la consulta] [Actualizar]

Actualizar se usa para ejecutar la misma consulta otra vez. Los resultados pueden cambiar, ya que el servidor MySQL es un servidor multi-usuario, y otros usuarios pueden modificar la(s) misma(s) tabla(s). Este enlace aparece si `cfg['SQLQuery']['Refresh']` está configurado TRUE.

El valor por defecto de estos cuatros parámetros es TRUE en `config.inc.php`.

Barra de navegación

Esta barra se muestra en la parte superior e inferior de los resultados. Las cabeceras de las columnas se pueden repetir entre los resultados a ciertos intervalos, dependiendo del valor escrito en **encabezado cada... celdas**.

En el ejemplo anterior la barra era sencilla:

Mostrar: 30

filas empezando de 0

en modo horizontal
y repetir los encabezados cada 100 celdas

La barra nos permite navegar de página a página, presentando un número arbitrario de registros (o columnas), comenzando en un punto de los resultados. Ya que hemos entrado en el modo **Examinar** al hacer clic en examinar, la consulta básica que ha generado los resultados incluye toda la tabla. Sin embargo este no es siempre el caso.

Debe darse cuenta que estamos en el registro número 0 y estamos viendo los registros en modo **horizontal**.

Veamos otro ejemplo, ahora con una tabla recién creada: La tabla **user**. Esta tabla contiene cuatro campos: `help_category_id`, `name`, `parent_category_id` y `url`. En esta tabla hay muchas filas (aquí 32109), la barra de navegación se adapta sola:

Mostrando registros 2 - 3 (5 total, La consulta tardó 0.0006 seg)

consulta SQL:

```
SELECT *
FROM 'user'
LIMIT 2, 2
```

[Editar] [Explicar el SQL] [Crear código PHP] [Actualizar]

<<

<

Mostrar: 2

filas empezando de 4

>

>>

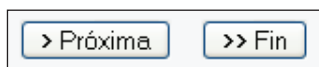
Número de página: 2

en modo horizontal
y repetir los encabezados cada 100 celdas

Organizar según la clave: Ninguna
Continuar

	Host	User	Password	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv	Reload_priv
☐	%			N	N	N	N	N	N	N
☐	%	marc	*8BA9FA710...	N	N	N	N	N	N	N

Ahora aparecen botones marcados <<,<,> y >> para acceder de forma fácil a la primera página, la página anterior, la página siguiente y la última página de los resultados. Los símbolos se muestran de esta manera porque la configuración por defecto de `$cfg['NavigationBarIconic']` es `TRUE`. Un `FALSE` aquí produciría unas etiquetas diferentes:



También hay un menú desplegable **Número de página**, para ir directamente a una de las páginas que está cerca de la página abierta. Como puede haber miles y miles páginas, este menú es pequeño, sólo tiene unos pocos números de páginas antes y después de la página abierta. Si selecciona el modo **vertical** en la tabla **user**, se genera la siguiente pantalla:

Mostrando registros 2 - 3 (34 total, La consulta tardó 0.0004 seg)

consulta SQL:

```
SELECT *
FROM 'help_category'
LIMIT 2, 2
```

[Editar] [Explicar el SQL] [Crear código PHP] [Actualizar]

Empezar << Previo < Mostrar: 2 filas empezando de 4 > Próxima >> Fin Número de página: 2

en modo vertical y repetir los encabezados cada 100 celdas

Organizar según la clave: Ninguna Continuar

help_category_id	3	4
name	WKT	Numeric Fu...
parent_category_id	29	33
url		

Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Por diseño, phpMyAdmin siempre intenta dar resultados con rapidez, uno puede llegar a este resultado si añade una cláusula `LIMIT` en `SELECT`. Si ya hay una cláusula `LIMIT` en la consulta original, phpMyAdmin la respetará. El límite por defecto es 30 filas, que está configurado en `$cfg['MaxRows']`. Con múltiples usuarios en el servidor, esto le ayuda a mantener la carga al mínimo.

Hay otro botón disponible en la barra de navegación, pero debe activarse configurando `$cfg['ShowAll']` en `TRUE`. Sería muy tentador para los usuarios usar este botón a menudo. Así, que en una instalación de phpMyAdmin, se recomienda que esté desactivado (`FALSE`). Cuando está activado, la barra de navegación aumenta como en esta figura:

> Próxima

>> Fin

Número de página:

1

▼

Mostrar todo

Si hace clic en el botón **Mostrar todos**, se extraen todas las filas de los resultados, que puede llegar al tiempo límite de ejecución de PHP, al límite de la memoria del servidor o del navegador


 Si escribimos un número grande en el diálogo **Mostrar...filas**, se encontrarán los mismos resultados (y encontraremos los mismos problemas potenciales).

Organizar resultados

En SQL, nunca podemos estar seguros en que orden se extraen los resultados a no ser que organicemos los datos explícitamente. Alguna implementación en la máquina de extracción puede mostrar resultados en el mismo orden con el que los datos se escribieron, con la clave primaria; pero una manera segura de obtener resultados en orden es organizarlos explícitamente.

Una manera obvia de organizarlos es con claves. El dialogo **Organizar** muestra las claves que ya están definidas. Aquí vemos una clave llamada **PRIMARIA**, el nombre que hemos dado a nuestra clave primaria cuando hemos marcado **Primaria** para este campo cuando lo creamos.

Mostrar: 30

filas empezando de 0

en modo horizontal

▼

y repetir los encabezados cada 100

celdas

Organizar según la clave:

Ninguna

Continuar

←T→

IS

PRIMARY (Ascendente)

PRIMARY (Descendente)

autor_id (Ascendente)

autor_id (Descendente)

Ninguna

1-2345

0

0

1-2345

(volumen 1)

0

0

↑

Marcar todos/as

/

Desmarcar todos

Para los elementos que están marcados:

Mostrar: 30

filas empezando de 0

en modo horizontal

▼

y repetir los encabezados cada 100

celdas

Esta puede que sea la única manera de organizar entre múltiples campos a la vez (para índices multi-campo).

Si elegimos organizar por **autor_id** (descendente), vemos:

Mostrar : 30 filas empezando de 0

en modo horizontal y repetir los encabezados cada 100 celdas

Organizar según la clave: **autor_id (Descendente)** Continuar

	ISBN	Título	número_pág	autor_id
☐	1-234567-22-0	Recuerdos futuros	0	0
☐	1-234567-89-0	Cien años de cine (volumen 1)	0	0

Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Mostrar : 30 filas empezando de 0

en modo horizontal y repetir los encabezados cada 100 celdas

Ahora vemos un pequeño triángulo rojo hacia abajo en la cabecera **autor_id**. Esto quiere decir que el orden de organización es 'descendente'. Si movemos el cursor del ratón sobre la cabecera **autor_id** hace que el triángulo cambie de dirección, para indicar que pasará si hacemos clic en la cabecera: Una organización ascendente de **autor_id**.

De hecho, todas las cabeceras de las columnas se pueden marcar para ordenar esta columna, aunque no formen parte de un índice. Podemos confirmarlo si miramos la consulta SQL que está en la parte superior de la pantalla, debería tener una cláusula ORDER BY.

La organización inicial por defecto está definida en `$cfg['Order']` con ASC para ascendente, DESC para descendente y SMART que se refiere a campos tipo DATE, TIME, DATETIME y TIMESTAMP que se organizan en orden descendente y otros tipos de campos en orden ascendente.

Marcar las filas con colores

Cuando movemos el ratón entre las filas, el color de fondo de las filas puede cambiar al color definido en `$cfg['BrowsePointerColor']`. Este parámetro se puede encontrar en `themes/themename/layout.inc.php`. Para activar este puntero del navegador para todos los temas `$cfg['BrowsePointerEnable']` debe estar configurado TRUE (por defecto) en `config.inc.php`.

Puede ser interesante marcar visualmente algunas filas para destacar su importancia para hacer una comparación personal de datos o cuando se están mostrando datos a otras personas. Se destaca al Hacer clic en la fila. Hacer clic quita la marca de la fila. El color elegido se define con `$cfg['BrowseMarkerColor']` (vaya a `themes/themename/layout.inc.php`). Esta característica debe estar activada configurando

`$cfg['BrowseMarkerEnable']` en `TRUE`, esta vez en `config.inc.php`: con esto se configura esta característica para todos los temas. Podemos marcar más de una fila. Si marcamos una fila, activamos el recuadro de esta fila.

Mostrar : 30 filas empezando de 0

en modo horizontal y repetir los encabezados cada 100 celdas

Organizar según la clave: autor_id (Descendente) Continuar

	ISBN	Título	número_pág	autor_id
☐  	1-234567-22-0	Recuerdos futuros	0	0
☐  	1-234567-89-0	Cien años de cine (volumen 1)	0	0

↑ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:   

Mostrar : 30 filas empezando de 0

en modo horizontal y repetir los encabezados cada 100 celdas

Limitar la longitud de las columnas

En los ejemplos anteriores, siempre teníamos contenidos completos para cada columna porque su número de caracteres estaba dentro del límite definido por `$cfg['LimitChars']`. Hay un límite obligatorio en todos los campos no numéricos. Si este límite fuese más bajo (digamos 10), la presentación sería como la siguiente:

	ISBN	Título	número_pág	autor_id
☐  	1-234567-2...	Recuerdos ...	0	0
☐  	1-234567-8...	Cien años...	0	0

↑ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:   

Esto nos ayuda a ver más columnas al mismo tiempo (con el coste de ver menos de cada columna).

Para ver los textos completos, podemos hacer clic en la T que está al lado de la cabecera de la columna, sirve para conmutar el modo texto completo y el modo texto parcial:

	ISBN	Título	número_pág	autor_id
☐  	1-234567-2...	Recuerdos ...	0	0
☐  	1-234567-8...	Cien años...	0	0

↑ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:   

Personalización del modo examinar

Aquí hay más parámetros que controlan el aspecto de los resultados. Estos parámetros, excepto `$cfg['RepeatCells']`; están en `themes/themename/layout.inc.php`.

- `$cfg['Border']`: las tablas HTML que se presentan en los resultados no tienen bordes porque por defecto este parámetro está configurado 0, pero podemos poner un número más alto (1 o 2) para añadir bordes a las tablas. `$cfg['ThBgColor']`: Las cabeceras mencionadas tienen #D3DCE3 como color de fondo por defecto.
- `$cfg['BgcolorOne']`: `$cfg['BgcolorTwo']`: Cuando se muestra una fila de resultados, se usan dos colores de fondo por defecto de forma alternativa, son #CCCCCC y #DDDDDD.
- `$cfg['RepeatCells']`: Cuando se muestran demasiadas filas, podemos perder el significado de cada columna; por defecto, cada 100 celdas, se muestran las cabeceras de las columnas.

Crear una tabla adicional

En nuestro sencillo diseño, sabemos que necesitamos otra tabla: La tabla **autores**. La tabla **autores** tendrá:

- Identificación del autor.
- El nombre completo del autor.
- Número de teléfono.

Para crear esta tabla debemos volver a la vista base de datos. En el panel de la izquierda, haremos clic en **dbbook**, y pediremos que nos cree otra tabla con tres campos:

The screenshot shows the dbbook interface for a database named 'dbbook' on 'localhost'. The top navigation bar includes buttons for 'Estructura', 'SQL', 'Buscar', 'Generar una consulta', 'Exportar', 'Importar', 'Operaciones', 'Privilegios', and 'Eliminar'. Below this, a table lists the existing table 'libros' with 2 records, InnoDB type, latin1_swedish_ci collation, and a size of 32.0 KB. Below the table list, there are links for 'Vista de impresión' and 'Diccionario de datos'. At the bottom, there is a section 'Crear nueva tabla en la base de datos dbbook' with input fields for 'Nombre: autor' and 'Número de campos: 3', and a 'Continuar' button.

Si usamos las mismas técnicas que hemos usado para crear la primera tabla, obtendremos:



The screenshot shows a database management interface with the following breadcrumb: Servidor: localhost > Base de datos: dbbook > Tabla: autor. Below this is a table configuration grid with the following columns: Campo, Tipo, Longitud/Valores¹, Cotejamiento, Atributos, Nulo, and Predeterminado². The table has three rows: 'autor_id' with type INT, 'nombre_autor' with type VARCHAR and length 30, and 'telefono' with type VARCHAR and length 30. All 'Nulo' values are set to 'not null'.

Campo	Tipo	Longitud/Valores ¹	Cotejamiento	Atributos	Nulo	Predeterminado ²
autor_id	INT				not null	
nombre_autor	VARCHAR	30			not null	
telefono	VARCHAR	30			not null	

Aquí usamos el mismo nombre de campo (**autor_id**, que es nuestra clave primaria en esta tabla nueva), para que nuestro diseño sea más consistente. Después de guardar la estructura de la tabla, escribimos algunos datos para los autores 1 y 2. ¡Use su imaginación para hacerlo!

Resumen

En este capítulo, hemos:

- Explicado como crear una base de datos y algunas tablas.
- Visto como introducir datos manualmente en las tablas.
- Confirmado la presencia de datos al familiarizarnos con el modo guardar, incluyendo los enlaces de consultas SQL, la barra de navegación, las opciones de organización y marcado de filas.

5

Cambiar datos

Los datos no son estáticos, a menudo cambian. Este capítulo se centra en la edición y supresión de datos y sus estructuras de soporte: Tablas y bases de datos.

Modo editar

Cuando examinamos una tabla o vemos los resultados de una búsqueda en una tabla sencilla de consultas, aparece un pequeño icono a la izquierda o a la derecha de cada fila de tablas:



La fila se puede editar con el icono en forma de lápiz y borrar con el icono en forma de X. La forma y posición exactas de estos controles están gobernadas por:

```
$cfg['PropertiesIconic']      = TRUE;  
$cfg['ModifyDeleteAtLeft']   = TRUE;  
$cfg['ModifyDeleteAtRight'] = FALSE;
```

Podemos decidir si queremos verlos en la izquierda, en la derecha o en ambos lados. El parámetro `$cfg['PropertiesIconic']` puede tener los siguientes valores: TRUE, FALSE o 'both'. TRUE presenta los iconos tal y como se han visto en la imagen anterior, FALSE presenta **Editar** o **Suprimir** (o su equivalente traducido) como enlaces y 'both' presenta el icono y el texto.

El pequeño recuadro que está al lado de todas las filas está explicado en las secciones *Editar múltiples filas* y *Suprimir varias filas* que aparecen más tarde en este capítulo.

Si hace clic en el icono o enlace **Editar** le muestra el siguiente panel que es parecido al panel de entrada de datos (a excepción de la parte inferior):

Campo	Tipo	Función	Nulo	Valor
ISBN	varchar(25)		☐	1-234567-22-0
Título	varchar(100)		☐	Recuerdos futuros
número_pág	int(11)		☐	0
autor_id	int(11)		☐	0

En este panel podemos cambiar datos tecleando directamente (o cortar y pegar con los mecanismos operativos normales del sistema). También podemos volver a los contenidos originales si usamos el botón **Restablecer**.

Por defecto, los botones de radio inferiores están configurados para **Guardar** (lo que quiere decir que podemos cambiar *esa* fila) y **Volver a la página anterior** (de modo que podemos seguir editando otra fila en la página anterior de resultados). Puede que queramos quedarnos en la página abierta después de hacer clic en **Continuar** (queremos guardar y continuar editando) de ese modo elegimos **Volver** a esta página. Si queremos insertar otra fila nueva después de guardar la fila actual, sólo tenemos que elegir **Insertar otra fila nueva** antes de guardar. La elección **Insertar como una fila nueva** se explica en la sección *Duplicar filas de datos*, en este capítulo.

Tabulador para ir al próximo campo

La gente que prefiere usar el teclado puede usar la tecla *Tabulador* para ir al siguiente campo. Normalmente, el cursor va de la izquierda a la derecha, de arriba abajo, de manera que irá a los campos de la columna **Función** (más en este momento). Sin embargo, para facilitar la navegación en phpMyAdmin, el orden natural de navegación se ha alterado; la tecla *Tabulación* primero pasa por cada campo en la columna **Valor** y luego por cada columna **Función**.

Tratamiento de valores NULL

Si la estructura de la tabla permite un valor NULL dentro de un campo, aparece un pequeño recuadro en la columna **Nulo** del campo. Si lo marcamos ponemos un valor NULL en el campo. Se ha añadido un mecanismo especial a phpMyAdmin para asegurarse que si se escriben datos en la columna **Valor** para este campo, el recuadro **Nulo** se vacíe automáticamente (esto es posible para los navegadores que tienen activado JavaScript).

Aquí hemos modificado la estructura (como se explica en el capítulo 6) en el campo **teléfono** de la tabla **autores** para permitir un valor NULL: Aquí el recuadro **Nulo** no está marcado:

Campo	Tipo	Función	Nulo	Valor
autor_id	int(11)		☐	1
nombre_autor	varchar(30)		☐	John Smith
teléfono	varchar(30)		☐	+01 445 789-1234

Los datos se borran cuando se marca el recuadro **Nulo** como se muestra en la siguiente pantalla:

Campo	Tipo	Función	Nulo	Valor
autor_id	int(11)		☐	1
nombre_autor	varchar(30)		☐	John Smith
teléfono	varchar(30)		☒	

El panel **Editar** aparecerá así si la fila se vuelve a poner en la pantalla.

Aplicar una función a un valor

El lenguaje MySQL ofrece algunas funciones que podemos aplicar antes de guardar, algunas de estas funciones aparecen en el menú desplegable que está debajo de cada campo, si `$cfg['ShowFunctionFields']` está configurado `TRUE`.

La lista función está definida en el conjunto `$cfg['Functions']`. Las funciones más usadas comúnmente para ciertos tipos de datos se presentan primero en la lista. Algunas restricciones están definidas en los conjuntos `$cfg['RestrictColumnTypes']` y `$cfg['RestrictFunctions']` que controlan *que* funciones se presentan primero.

Aquí hay definiciones que restringen los nombres de las funciones que se presentan en el tipo VARCHAR:

```
$cfg['RestrictColumnTypes'] = array(
    'VARCHAR' => 'FUNC_CHAR', [...]
```

```
$cfg['RestrictFunctions'] = array(
    'FUNC_CHAR' => array(
        'ASCII',
        'CHAR',
        'SONDEX',
        'LCASE',
        'UCASE',
        'PASSWORD',
        'MD5',
        'ENCRYPT',
        'LAST_INSERT_ID',
        'USER',
        'CONCAT'
    ), [...]
```

Como se muestra en la siguiente pantalla, aplicamos la función UCASE al título cuando guardamos *esta fila*:

Campo	Tipo	Función	Nulo	Valor
ISBN	varchar(25)			1-234567-22-0
Título	varchar(100)			Recuerdos futuros
número_pág	int(11)			0
autor_id	int(11)			0

Grabar

ASCII
 CHAR
 SOUNDEX
 LCASE
UCASE
 PASSWORD
 OLD_PASSWORD
 MD5
 SHA1
 ENCRYPT

Continuar
Reiniciar

Esta característica se puede desactivar si configuramos `$cfg['ShowFunctionFields']` en `FALSE` para ganar algo de espacio en la pantalla (para poder ver más parte de los datos).

Duplicar filas de datos

Durante el mantenimiento de datos (para duplicar de forma permanente o para hacer test), a menudo tenemos que generar una copia de una fila. Si esto se hace en la misma tabla, debemos respetar las reglas de claves únicas.

Aquí necesitamos un ejemplo. Nuestro autor ha escrito el volumen 2 del libro sobre el cine y los únicos campos que tienen que cambiarse son el número ISBN y el título. Ponemos la fila existente en la pantalla, cambiamos estos dos campos y elegimos **Insertar como una fila nueva**, como vemos en la pantalla:

Campo	Tipo	Función	Nulo	Valor
ISBN	varchar(25)			1-234567-89-0
Título	varchar(100)			Cien años de cine (volumen 2)
número_pág	int(11)			0
autor_id	int(11)			0

Insertar como una nueva fila
y luego
Volver

Continuar
Reiniciar

Cuando hacemos clic en **Continuar**, se crea otra fila con la información cambiada y deja la fila original sin cambiar:

→T←			ISBN	Título	número_pág	autor_id
☐			1-234567-22-0	Recuerdos futuros	200	2
☐			1-234567-89-0	Cien años de cine (volumen 1)	600	1
☐			1-234567-90-0	Cien años de cine (volumen 2)	600	1

[Marcar todos/as](#) / [Desmarcar todos](#) Para los elementos que están marcados:

Editar múltiples filas

Si empezamos con phpMyAdmin 2.5.5, la característica de edición multi-filas nos permite usar recuadros en las filas si queremos editar; podemos usar el enlace **Editar** (o el icono en forma de lápiz) en el menú **Anchura seleccionada**. Los enlaces **Marcar Todos/Desmarcar todos** también se pueden usar para marcar o dejar de marcar los recuadros. También podemos hacer clic en cualquier parte de la fila de datos para activar el recuadro correspondiente.

Esto hace que aparezca el panel Editor que contiene todas las filas elegidas, el proceso de edición continuará mientras se vean, comparen o cambien los datos de estas filas.

Servidor: localhost ▶ Base de datos: dbbook ▶ Tabla: autor "InnoDB free: 4096 kB"

[Examinar](#) [Estructura](#) [SQL](#) [Buscar](#) [Insertar](#) [Exportar](#) [Importar](#) [Operaciones](#) [Vaciar](#) [Eliminar](#)

Campo	Tipo	Función	Nulo	Valor
autor_id	int(11)		☐	1
nombre_autor	varchar(30)		☐	John Smith
teléfono	varchar(30)		☐	111-1111

☐ Ignorar

Campo	Tipo	Función	Nulo	Valor
autor_id	int(11)		☐	2
nombre_autor	varchar(30)		☐	Maria Sunshine
teléfono	varchar(30)		☐	222-2222

Insertar como una nueva fila y luego Volver



Cuando marcamos algunas filas con los recuadros, también podemos hacer dos acciones en ellas: Suprimir (vea la sección *Suprimir varias filas* de este capítulo) y exportar (vea el capítulo 7).

Suprimir datos

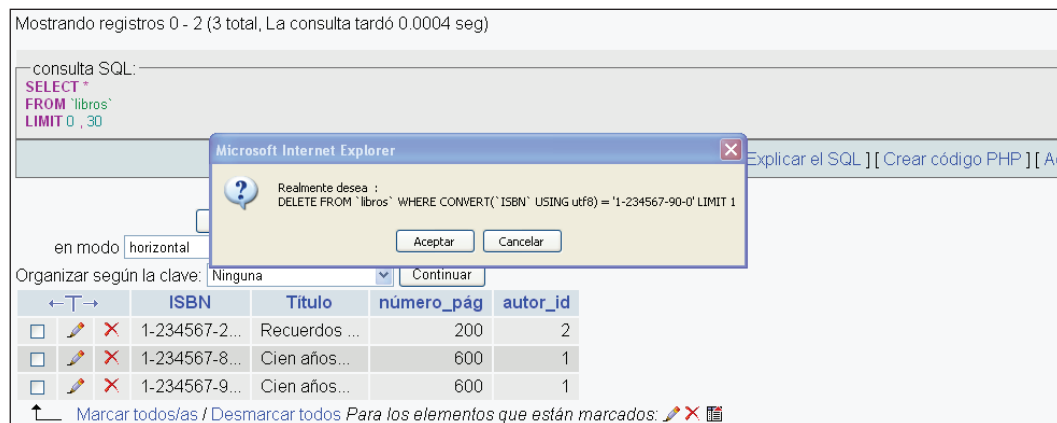
La interfaz de phpMyAdmin nos permite suprimir lo siguiente:

- Una fila de datos.
- Varias filas de datos.
- Todas las filas de una tabla.
- Todas las filas de varias tablas.

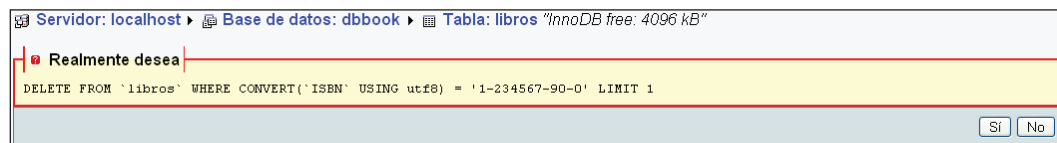
Suprimir sólo una fila

Podemos usar el pequeño icono en forma de X, que está al lado de cada fila, para borrar la fila. Si el valor de `$cfg['Confirm']` está configurado en `TRUE`, cada sentencia `DELETE` MySQL tiene que confirmarse antes de ejecutarse. Esto se hace por defecto ya que no es prudente permitir borrar una fila con tan sólo un clic.

La forma de la confirmación varía dependiendo de si el servidor puede ejecutar o no JavaScript. Una confirmación emergente basada en JavaScript tendrá el aspecto de esta pantalla:



Si JavaScript se ha desactivado, aparecerá un panel distintivo:



La sentencia DELETE usará la información que necesite para asegurar la supresión de la fila seleccionada. En nuestro caso, se ha definido una clave primaria y se ha usado en la cláusula WHERE. Si no hay una clave primaria, se generará una cláusula WHERE basada en el valor de cada campo. La cláusula WHERE incluso puede evitar la ejecución correcta de la operación DELETE, especialmente si hay campos TEXT o BLOB, porque la transacción HTTP que se usaba para enviar la consulta al servidor Web puede que tenga una longitud limitada por el navegador o el servidor.

Suprimir varias filas

Una característica que se añadió a phpMyAdmin en la versión 2.5.4 es la supresión de multi-fila. Digamos que vamos a examinar una página de filas y decidimos destruir algunas filas. En lugar de borrarlas una a una con el enlace o icono **Suprimir**; y porque a menudo la decisión de suprimir se hace mientras se examina un grupo de filas; hay recuadros al lado de cada fila en la vista tabla:

→T←			ISBN	Título	número_pág	autor_id
☒			1-234567-22-0	Recuerdos futuros	200	2
☐			1-234567-89-0	Cien años de cine (volumen 1)	600	1
☒			1-234567-90-0	Cien años de cine (volumen 2)	600	1
↶ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:						

Estás se usan con el icono en forma de X **seleccionado**. Aparece una pantalla de confirmación con una lista de todas las filas que se van a suprimir. También podemos hacer clic en cualquier parte de la fila de datos para activar el recuadro correspondiente.

Suprimir todas las filas de una tabla

Para borrar por completo todas las filas de una tabla (dejando su estructura intacta), tenemos que ir a la vista base de datos y hacer clic en el nombre de base de datos del panel de la izquierda. Luego haremos clic en el icono papelera que está en la misma línea que la tabla que queremos vaciar:

Servidor: localhost ▶ Base de datos: dbbook							
Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegio							
	Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
☐	autor		0	InnoDB	latin1_swedish_ci	16.0 KB	-
☒	libros		3	InnoDB	latin1_swedish_ci	48.0 KB	-
	2 tabla(s)	Número de fila: vaciar	3	InnoDB	latin1_swedish_ci	64.0 KB	0 Bytes
☐ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:							

Tendremos un mensaje que confirmará TRUNCATE (la sentencia MySQL que se usa para vaciar una tabla de forma rápida). Vaciar una tabla también se puede hacer en la vista tabla, con el enlace **Vaciar** que está en el menú superior:



 Suprimir datos, ya sea fila por fila o vaciando la tabla, es una acción permanente. No se puede recuperar a excepción con la ayuda de nuestras copias de seguridad.

Suprimir todas las filas de varias tablas

La pantalla de antes muestra un recuadro a la izquierda de cada tabla. Podemos elegir algunas tablas, luego en el menú **Para todos los elementos que están marcados**, elija la operación **Vaciar** como se muestra en la siguiente pantalla :

Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
☒ autor		2	InnoDB	latin1_swedish_ci	16.0 KB	-
☒ libros		2	InnoDB	latin1_swedish_ci	32.0 KB	-
2 tabla(s)	Número de filas	4	InnoDB	latin1_swedish_ci	48.0 KB	0 Bytes
↑ Marcar todos/as / Desmarcar todos Vista de impresión Diccionario de datos Crear nueva tabla en la base de datos dbbo Nombre: Número Para los elementos que están marcados: - Vaciar - Eliminar - Vista de impresión - Revisar la tabla - Optimizar la tabla - Reparar la tabla - Analizar la tabla						

Por supuesto, ¡no debe tomar esta decisión a la ligera!

Suprimir tablas

Suprimir una tabla borra los datos y la estructura de la tabla. Podemos suprimir tablas si usamos el enlace **Eliminar** de la vista tabla.



En la vista base de datos, podemos suprimir una tabla específica si usamos el icono en forma de X de esa tabla. También existe el mismo mecanismo para borrar más de una tabla (con el menú desplegable y la acción **Eliminar**).

 Las acciones Vaciar y Eliminar están marcadas en rojo para indicar mejor el peligro inherente de estas acciones en los datos.

Suprimir bases de datos

Podemos suprimir toda una base de datos ; incluyendo todas sus tablas; si hacemos uso del enlace Eliminar de la vista base de datos.



Por defecto `$cfg['AllowUserDropDatabase']` está configurado en `FALSE`, de modo que este enlace está oculto a los usuarios sin permisos hasta que esta configuración se cambia manualmente a `TRUE`.

Para que nos los pensemos dos veces, aparece un mensaje especial antes que se suprima la base de datos: **¡Está a punto de DESTRUIR una base de datos!**

Resumen

En este capítulo hemos examinado los siguientes conceptos:

- Editar datos, incluyendo el campo nulo y el uso de la tecla *Tabulación*.
- Aplicar una función a un valor.
- Duplicar una fila de datos.
- Suprimir datos, tablas y bases de datos

6

Cambiar la estructura de la tabla

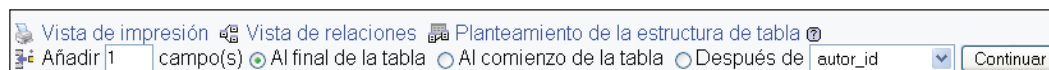
Este capítulo explora la edición de la definición de la tabla y el uso de tipos de columnas especiales. Cuando se desarrolla una aplicación, los requisitos cambian a menudo a causa del cambio de necesidades. Los programadores deben arreglar estos cambios a través de una edición de la estructura de la tabla sensata.

Añadir un campo

Imagine que necesitamos un campo nuevo para almacenar el idioma del libro, y por defecto, los libros cuyos datos tenemos guardados están escritos en inglés. Decidimos que el campo se llamará **idioma**, será un código de dos caracteres (en por defecto).

En la sub-página **Estructura** de la vista página para la tabla **libros**, podemos ver el diálogo **Añadir nuevo campo**. Aquí, especificamos cuantos campos queremos y donde van los nuevos campos.

Las posiciones de los nuevos campos de la tabla, sólo importan desde el punto de vista del programador; normalmente nosotros agrupamos los campos de forma lógica para encontrarlos con más facilidad en la lista de campos. La posición exacta de los campos no afecta a los resultados (salida de datos de las consultas), porque los resultados se pueden ajustar sin tener en cuenta la estructura de la tabla. Normalmente, los campos más importantes (incluyendo las claves) están al principio de tabla, pero es cuestión de preferencia personal. Elegimos poner el nuevo campo **Al final de la tabla**, así que marcamos el botón de radio y hacer clic en **Continuar**:



Vista de impresión Vista de relaciones Planteamiento de la estructura de tabla

Añadir 1 campo(s) ☒ Al final de la tabla ☐ Al comienzo de la tabla ☐ Después de autor_id Continuar

Otras elecciones posibles podrían ser **Al principio de la tabla** o **Después de...** (con el que podríamos elegir donde deber ir el nuevo campo en el menú desplegable).

Vemos que aparece el panel familiar para los nuevos campos, en el se ve repetido el número de campos que hemos pedido. Lo rellenamos y esta vez ponemos un valor por defecto, **en**. Luego hacemos clic en **Guardar**.

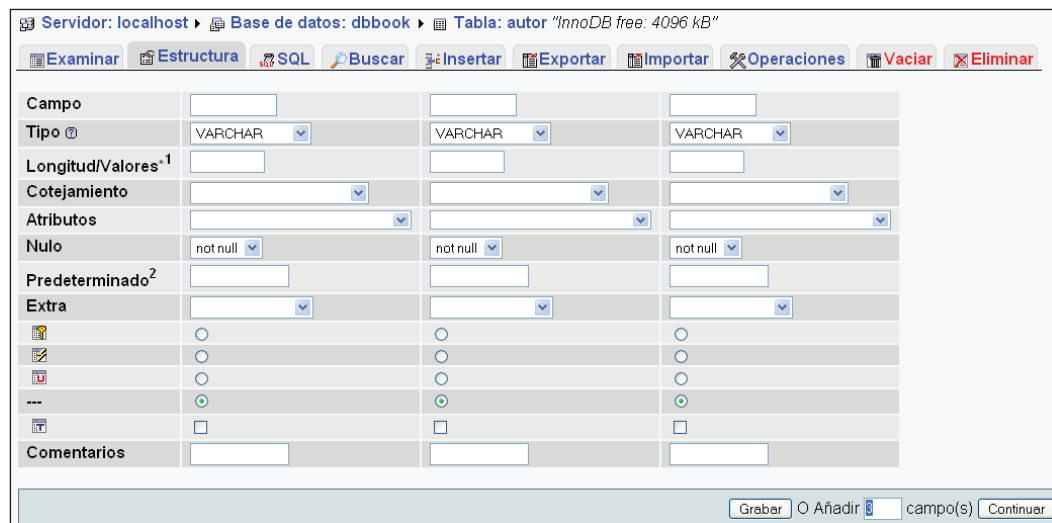


Este panel aparece en modo horizontal ya que es la configuración por defecto de `$cfg['DefaultPropDisplay']`.

Modo vertical

Si configuramos `$['DefaultPropDisplay']` en 'vertical', el panel (junto con el panel para editar la estructura de la tabla) que añade campos nuevos tendrá una presentación vertical. Las ventajas de trabajar en modo vertical son obvias, especialmente cuando hay más de una elección por campo, tal y como se explica en el capítulo 16, *Transformaciones basadas en MIME*.

Veamos como se presenta el panel si estamos en el modo vertical y pedimos tres campos nuevos:



Editar los atributos del campo

En la sub-página **Estructura** podemos hacer más cambios a nuestra tabla. Para este ejemplo, hemos configurado `$cfg['PropertiesIconic']` en `'both'` para ver los iconos junto con su explicación textual:

Servidor: localhost ▶ Base de datos: dbbook ▶ Tabla: libros "InnoDB free: 4096 kB"

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
☐	ISBN	varchar(25)	latin1_swedish_ci		No			      
☐	Título	varchar(100)	latin1_swedish_ci		No			      
☐	número_pág	int(11)			No	0		      
☐	autor_id	int(11)			No	0		      
☐	idioma	char(3)	latin1_swedish_ci		No	en		      

↑ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:       

Este panel no permite cualquier cambio a los campos. Específicamente permite:

- Cambiar la estructura en un campo haciendo uso del enlace **Cambiar** de un campo específico.
- Borrar el campo: **Eliminar**.
- Añadir un campo en la clave **Primaria** que ya existe.
- Configurar un **índice** no único o un índice **Único** en un campo.
- Configurar un índice de **texto completo** (se ofrece sólo si el tipo de campo lo permite).

Hay enlaces rápidos que pueden ser útiles en situaciones específicas. Tenga en cuenta que reemplazan el índice completo del panel de administración o el índice completo del panel estructura. Ambos se explican en este capítulo.

También podemos usar los recuadros para elegir campos y con los iconos **Para los elementos que están marcados**, editar los campos o suprimir múltiples campos con **Eliminar**. La opción **Marcar todos/Desmarcar todos** permite marcar o desmarcar con facilidad todos los recuadros.

Campos TEXT

Ahora vamos a explicar como se usa el tipo de campo TEXT y los valores de configuración relevantes que hay que ajustar para que phpMyAdmin funcione mejor.

Primero añadimos un campo TEXT llamado **descripción**:

Campo	Tipo ¹	Longitud/Valores ¹
descripción	VARCHAR	
	VARCHAR	
	TINYINT	
	TEXT	
	DATE	
	SMALLINT	
	MEDIUMINT	
	INT	
	BIGINT	
	FLOAT	
	DOUBLE	
	DECIMAL	
	DATETIME	
	TIMESTAMP	
	TIME	
	YEAR	

¹ Si el tipo es un número, ingrese el número de caracteres. Si alguna vez necesita un tipo de dato que no está en esta lista, ingrese el nombre del tipo de dato en la columna "Tipo".

² Para valores de texto, ingrese el número de caracteres. Para valores de fecha y hora, ingrese el formato de la fecha y hora.

Hay tres parámetros que controlan el formato de la zona del texto que se presenta en el modo **Insertar** o **Editar** para los campos TEXT. Primero, el número de columnas y filas de cada campo está definido por :

```
$cfg['TextareaCols'] = 40;
$cfg['TextareaRows'] = 7;
```

Por defecto, esto le da el siguiente espacio para trabajar en un campo TEXT:

descripción	text	
-------------	------	--

La configuración sólo impone un límite visual a la zona de texto, se crea una barra de desplazamiento vertical si es necesario.

A pesar que las columnas MEDIUMTEXT, TEXT y LONGTEXT tienen capacidad para más de 32k de datos, los navegadores de hoy en día no siempre pueden editarlas con los mecanismos que HTML ofrece: Una zona de texto. De hecho, los experimentos han convencido al equipo de desarrollo de phpMyAdmin que el producto debe presentar un mensaje de advertencia cuando el contenido tiene más de 32 k, advirtiéndolo a los usuarios que quizá no se pueda editar.

Para los campos LONGTEXT cuando la configuración de `$cfg['LongtextDoubleTextarea']` es TRUE, el espacio disponible para editar se dobla.

BLOB (Objeto Binario Extendido)

Los campos BLOB normalmente se usan para guardar algunos datos binarios (imagen, sonido), a pesar que la documentación de MySQL presupone que los campos TEXT se pueden usar con ese mismo propósito. Sin embargo la intención de phpMyAdmin es trabajar con campos BLOB para guardar todos los datos binarios.

Veremos en el capítulo 16, *Transformaciones basadas en MIME* que son mecanismos especiales disponibles para ir más lejos con los campos BLOB, incluso poder ver directamente algunas imágenes desde phpMyAdmin.

Primero añadimos un campo BLOB, **foto_cubierta**, a nuestra tabla **libros**:

Campo	Tipo	Longitud/Valores ¹
foto_cubierta	BLOB	

Si ahora examinamos la tabla, podemos ver la información de longitud del campo **[BLOB - 0 Bytes] por cada campo BLOB**:

	ISBN	Título	numero_pág	autor_id	idioma	descripción	foto_cubierta
	1-234567-22-0	Recuerdos futuros	200	2	en		[BLOB - 0 Bytes]
	1-234567-89-0	Cien años de cine (volumen 1)	600	1	en		[BLOB - 0 Bytes]
	1-234567-90-0	Cien años de cine (volumen 2)	600	1	en		[BLOB - 0 Bytes]

Esto es porque la directiva de configuración `$cfg['ShowBlob']` está configurado en FALSE por defecto, de modo que los contenidos de BLOB no se pueden ver en los modos Buscar y Editar (y la ventana de alarma **Binario; no editar**). Es un comportamiento intencionado ya que normalmente no podemos hacer nada con datos binarios representados en texto normal.

Carga de contenidos binarios

Si ahora editamos una fila, vemos un alarma y un botón **Buscar...** A pesar que no esta permitido editar, podemos cargar con facilidad un archivo texto o binario en esta columna BLOB.

Escojamos un archivo imagen con el botón **Buscar**; por ejemplo, el archivo `logo_left.png` que está en el directorio `phpMyAdmin/themes/original/img`; y hacemos clic en Continuar:

foto_cubierta	blob		Binario - ¡no editar! (0 Bytes) C:\Archivos de Examinar... (Tamaño máximo: 65,536Bytes)
---------------	------	----------------------	---

Tenemos que tener en cuenta que hay límites de tamaño en la carga. Primero, el tamaño del campo `BLOB` está limitado a 64k, así pues `phpMyAdmin` nos recuerda este límite con el **Max**: Alarma 65.536 bytes. También, puede haber límites inherentes en PHP; ver el capítulo 8, *Importar la estructura y datos*; que se tienen que tener en cuenta en este valor tamaño máximo. Ahora hemos cargado una imagen en este campo, en una fila específica.

			1-234567-22-0	Recuerdos futuros	200	2	en	[BLOB - 4.3 KB]
---	---	---	---------------	-------------------	-----	---	----	-----------------

Si `$cfg['ShowBlob']` está configurado `TRUE`, vemos el siguiente campo BLOB:

◆PNG\r\nZ\r\n0\0...

Para ver una imagen desde phpMyAdmin, por favor vaya al capítulo 16, *Transformaciones basadas en MIME*.

El parámetro `$cfg['ProtectBinary']` controla lo que se puede hacer mientras se editan los campos binarios (BLOBs y cualquier otro campo con el atributo `binary`).

El valor por defecto 'blob' nos permite proteger los campos BLOB de ser editados, lo que nos permite editar otros campos marcados binary por MySQL. Un valor 'all' protegería todos los campos binarios de ser editados. Un valor FALSE, no protegería nada lo que nos permitiría editar todos los campos. Si intentamos la última elección, veremos el siguiente panel de edición para esta fila.

¡Lo más seguro es que no sea nuestro editor de imagen favorito! De hecho, los datos pueden corromperse incluso si guardamos este archivo sin tocar el campo `BLOB`. Pero la configuración para borrar `ProtectBinary` existe porque algunos usuarios ponen texto en sus `BLOB` y se les tiene que dar permiso para modificarlos.

 Los tipos de dato BLOB de MySQL en realidad son parecidos a sus tipos de dato TEXT, con la única diferencia de ser tratados como sensibles a la hora de la clasificación y la comparación. Por eso phpMyAdmin se puede configurar para permitir la edición de los campos BLOB.

ENUM y SET

Ambos tipos de campos tienen por objetivo representar una lista de valores probables; la diferencia es que el usuario puede elegir tan sólo un valor de una lista definida de valores con `ENUM` y más de un valor con `SET`. Con `SET`, los valores múltiples van en una celda; tener valores múltiples no implica la creación de más de una fila de datos.

Añadimos un campo llamado **género** y lo definimos como `ENUM`. Por ahora, elegimos poner códigos cortos en la lista de valores y uno de ellos (F) es el valor por defecto:

Campo	Tipo	Longitud/Valores ¹	Cotejamiento	Atributos	Nulo	Predeterminado ²
género	ENUM	'F','Ni','N'			not null	F

En la lista de valores, tenemos que blindar cada valor con comillas, al contrario que el campo valor por defecto. En nuestro diseño, sabemos que esos valores quieren decir *Fantasía*, *Niños* y *Novela*, pero por ahora queremos ver como se comporta la interfaz con códigos cortos. Ahora en el panel **insertar** vemos una interfaz buzón radio.

foto_cubierta	blob			Binario - ¡no editar! (4.3 KB)	Examinar...	(Tamaño máximo: 65,536Bytes)
género	enum	--		☒ F ☐ Ni ☐ N		

Si decidimos poner códigos más descriptivos, podemos volver al modo Estructura y cambiar la definición del campo **género**. En el siguiente ejemplo, no vemos la lista de valores completa porque el campo no es lo bastante largo, pero lo que hemos escrito es **'Fantasía'**, **'Niños'**, **'Novela'**. También tenemos que cambiar el valor por defecto a uno de los valores posibles, para evitar que nos salga un mensaje de error mientras guardamos la modificación de la estructura del archivo.

Campo	Tipo	Longitud/Valores ¹	Cotejamiento	Atributos	Nulo	Predeterminado ²
género	ENUM	'as','Novela'	latin1_swedish_ci		not null	Fantasía

Con la lista de valores modificada, el panel **Insertar** tiene este aspecto:

género	enum	--		Fantasía		
Insertar como una nueva fila y luego Volver Continuar Reiniciar						

Observe que los botones radio previos han cambiado a una lista de selección porque los posibles valores son más largos.

Si queremos elegir más de un valor posible, tenemos que cambiar el tipo de campo a SET. Se puede usar la misma lista de valores, pero ahora, usaremos el selector de valores múltiples de nuestro navegador (normalmente hacer clic en control) para seleccionar más de un valor:

género	set	--	Fantasia Niños Novela
--------	-----	----	--

 Para el ejemplo previo, guardaremos sólo los códigos de **género** en la tabla **libros** en una estructura de tabla normal y confiaremos en otra tabla para guardar la descripción de cada código. En este caso no usaremos SET o ENUM.

DATE, DATETIME, TIMESTAMP: Pantalla emergente calendario

Podemos usar un campo normal de caracteres para almacenar información de fecha o tiempo, pero DATE, DATETIME y TIMESTAMP son más eficientes para este propósito. MySQL verifica los contenidos para asegurarse que la fecha y el tiempo son válidos. Como un beneficio añadido, phpMyAdmin ofrece una ventana emergente calendario para escribir datos con más facilidad.

Empezaremos añadiendo un campo DATE, **fecha-publicación**, a nuestra tabla **libros**. Si vamos al modo **Insertar**, deberíamos ver el campo nuevo donde podemos escribir una fecha. También hay disponible un icono **Calendario**.

fecha_publicación	date		0000-00-00
☒ Ignorar			

Este icono hace que aparezca una ventana emergente que está sincronizada con el campo DATE. Si ya hay un valor en este campo, la ventana emergente muestra lo mismo. En nuestro caso el valor por defecto 00-00-0000 era el que se veía en el campo, de modo que el calendario muestra la fecha actual:

Título	varchar(100)		
número_pág	int(11)		
autor_id	int(11)		
idioma	enum		
descripción	text		
foto-cubierta	blob		Binario - ¡no editar! (0 Bytes) Examinar... (Tamaño máximo: 65,536Bytes)
género	set	--	

Los pequeños símbolos al lado de cada cabecera año y mes, nos permite navegar con facilidad por los meses y los años, con un sencillo clic podemos transportar la fecha que queremos a nuestro campo **fecha_publicación**.

Para los campos DATETIME o TIMESTAMP, la ventana emergente ofrece la edición de la parte del tiempo:

« May »			
Dom	Lun	Mar	Mie
	1	2	3
7	8	9	10
14	15	16	17
21	22	23	24
28	29	30	31
00	00	00	
Continuar			

Administración de índices

phpMyAdmin tiene un gran número de opciones para administrar índices, opciones que cubriremos en esta sección.

Índices de un solo campo

Ya hemos visto como el panel Estructura ofrece una manera rápida de crear un índice en un solo campo, gracias a los enlaces rápidos **Primaria**, **Índice** y **Único**. Bajo la lista de los campos, hay una sección de la interfaz que se usa para administrar índices:

Índices: ⓘ				
Nombre de la clave	Tipo	Cardinalidad	Acción	Campo
PRIMARY	PRIMARY	3	 	ISBN
autor_id	INDEX	3	 	autor_id
Crear un índice en 1 columna(s) Continuar				

Esta sección tiene enlaces para editar o suprimir cada índice. Aquí, la parte **campo** sólo pone un campo en la lista por índice, además vemos que todo el campo participa en el índice.

Ahora añadiremos un índice en el título. Sin embargo, queremos restringir la longitud de este índice para reducir el espacio en disco usado por la estructura índice. La opción más apropiada es **Crear un índice en una columna** y luego hacer clic en **Continuar**. En la siguiente pantalla, especificamos los detalles del índice como podemos ver aquí:

Servidor: localhost ▶ Base de datos: dbbook ▶ Tabla: libros "InnoDB free: 4096 kB"

Crear un nuevo índice

Nombre del índice : Tipo de índice : INDEX ⓘ

 ("PRIMARY" exclusivamente debe ser el nombre de una clave primaria!)

Campo	Tamaño
Título [varchar(100)]	30

☐ Añadir al índice columna(s)

Aquí ve como tiene que rellenar este panel:

- **Nombre del índice:** Ponemos un nombre que nos hemos inventado.
- **Tipo de índice:** Podemos elegir entre **ÍNDICE** o **ÚNICO**.
- **Campo:** Elegimos el campo que se usa para hacer el índice, en este caso el campo **título**.
- **Tamaño:** Escribimos 30 en lugar de 100 (la longitud completa del campo) para ahorrar espacio.

Después de guardar este panel, en la pantalla de muestra que sigue podemos confirmar que se ha creado el índice y que no cubre toda la longitud del campo título:

Índices: ⓘ					
Nombre de la clave	Tipo	Cardinalidad	Acción		Campo
PRIMARY	PRIMARY	2			ISBN
autor_id	INDEX	2			autor_id
por_titulo	INDEX	2			Título 30
Crear un índice en 1 columna(s) Continuar					

Edición índices multi-campo y edición de índices

En el siguiente ejemplo, asumiremos que en una aplicación futura necesitaremos encontrar los libros escritos por un autor específico en un idioma específico. Por lo tanto tiene sentido expandir nuestro índice **autor_id**, añadiendo el campo **idioma**.

Hacemos clic en el enlace **Editar** (pequeño lápiz) en la línea que contiene el índice **autor_id** y elegimos añadir una columna a este índice como se ve en la siguiente muestra de pantalla:

Servidor: localhost ▶ Base de datos: dbbook ▶ Tabla: libros "InnoDB"

Modificar un índice

Nombre del índice: Tipo de índice: INDEX ⓘ

⚠ ("PRIMARY" exclusivamente debe ser el nombre de una clave primaria!)

Campo	Tamaño
autor_id [int(11)]	

☐ Añadir al índice columna(s)

Elegimos el campo **idioma** en el siguiente panel. Esta vez no tenemos que poner un tamaño ya que usará todo el campo en el índice:

Modificar un índice

Nombre del índice : Tipo de índice : INDEX

⚠ ("PRIMARY" exclusivamente debe ser el nombre de una clave primaria!)

Campo	Tamaño
autor_id [int(11)]	
- Ignorar -	
- Ignorar -	
ISBN [varchar(25)]	
Título [varchar(100)]	
número_pág [int(11)]	
autor_id [int(11)]	
idioma [char(2)]	
descripción [text]	

Eliminar O Añadir al índice 1 columna(s) Continuar

Para tener mejor información, deberíamos cambiar el nombre de la clave (**autor_id** sería apropiado). Guardamos la modificación de este índice y volvemos a:

Índices: ⓘ					
Nombre de la clave	Tipo	Cardinalidad	Acción	Campo	
PRIMARY	PRIMARY	3		ISBN	
autor_id Idioma	INDEX	3		autor_id	Idioma
por_título	INDEX	3		Título	30
Crear un índice en 1 columna(s) Continuar					

Índices FULLTEXT

Este tipo especial de índice permite búsquedas de textos completos. Está soportado en las tablas tipo MyISAM para los campos VARCHAR y TEXT. Podemos usar el enlace rápido **Texto completo** que está en la lista de los campos o ir al panel de administración de índices y elegir **Texto completo** en el menú desplegable:

Texto completo

Queremos un índice de **Texto completo** en el campo **descripción**, de forma que podamos encontrar un libro por las palabras que aparecen en su descripción. Después de crear el índice, la lista de índices tiene este aspecto:

Índices: 0					
Nombre de la clave	Tipo	Cardinalidad	Acción		Campo
PRIMARY	PRIMARY	0	Editar	Eliminar	ISBN
por_titulo	INDEX	Ninguna	Editar	Eliminar	Titulo 30
autor_idioma	INDEX	Ninguna	Editar	Eliminar	autor_id idioma
descripción	FULLTEXT	Ninguna	Editar	Eliminar	descripción
Crear un índice en 1 columna(s) Continuar					

Puede que veamos un 1 como longitud del campo para el índice recién creado, dependiendo de la versión de MySQL. De hecho, MySQL no soporta la longitud de un índice FULLTEXT: El índice siempre aparece en todo el campo, pero *este* 1 será el valor que informa MySQL.

Optimización de una tabla: EXPLAIN una consulta

En esta sección queremos obtener información sobre el índice que MySQL usa para una consulta específica y el impacto en el funcionamiento si no se ha definido un índice.

Asumamos que queremos usar la siguiente consulta.

```
SELECT *
FROM `books`
WHERE author_id = 2 AND language = 'es'
```

Queremos saber que libros escritos por el autor 2 están en el idioma **es**, nuestro código para español.

Para escribir esta consulta, usamos el enlace SQL desde el menú base de datos, el menú tabla o la ventana de consulta SQL. Escribimos esta consulta en el recuadro consulta y hacemos clic en **Continuar**. Ahora no es importante si la consulta encuentra resultados.

MySQL ha devuelto un valor vacío (i.e., cero columnas). (La consulta tardó 0.0004 seg)

consulta SQL:


```
SELECT *
FROM `libros`
WHERE author_id = 2
AND idioma = 'es'
LIMIT 0 , 30
```

[\[Editar \]](#)
[\[Explicar el SQL \]](#)
[\[Crear código PHP \]](#)
[\[Actualizar \]](#)

Veamos los enlaces: [\[Editar\]](#) [\[Explicar SQL\]](#) [\[Crear código PHP\]](#) [\[Actualizar\]](#)

Ahora usaremos el enlace [\[Explicar SQL\]](#) para obtener información sobre que índice (si se ha usado alguno) se ha usado para esta consulta:

consulta SQL

consulta SQL:

```
EXPLAIN SELECT *  
FROM `libros`  
WHERE autor_id =2  
AND idioma = 'en'
```

[\[Editar\]](#) [\[Salir de la explicación del SQL\]](#) [\[Crear código PHP\]](#)

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	libros	ref	autor_idio...	autor_idio...	5	const,cons...	1	Using wher...

Podemos ver que el comando `EXPLAIN` se ha pasado a MySQL, lo que nos dice que la `clave_posible` usada es **autor_idioma**. Así pues sabemos que este índice se usará para este tipo de consulta. Si este índice no hubiese existido, el resultado hubiese sido bastante diferente:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	autor	ALL	NULL	NULL	NULL	NULL	1	

Aquí, la **clave_posible** (NULL) y el **tipo** (ALL) quieren decir que no se ha usado ningún índice y que todas las filas necesitan ser examinadas para encontrar los datos deseados. Esto podría tener un impacto serio en el funcionamiento, dependiendo del número total de filas. Podemos asegurarnos del impacto exacto si examinamos el tiempo de la consulta que phpMyAdmin presenta en cada página de resultados y comparar entre tener índice o no tenerlo:

Mostrando registros 0 - 0 (1 total, La consulta tardó 0.0005 seg)

Sin embargo, la diferencia tiempo puede ser mínima si sólo tenemos datos de test limitados comparados a la producción de una tabla real.

Resumen

En este capítulo hemos cubierto:

- Como añadir campos, incluyendo tipos especiales como TEXT, BLOB, ENUM y SET.
- Como usar una pantalla emergente calendario para los campos DATE, DATETIME y TIMESTAMP.
- Como cargar datos binarios en un campo BLOB.
- Como administrar índices (multi-campo y texto completo).
- Como obtener información de MySQL sobre que índices se usan en una consulta concreta.

7

Exportar la estructura y los datos

Tener una buena copia de seguridad es crucial para un proyecto: Está constituida por copias de seguridad actualizadas y copias instantáneas hechas durante las fases de desarrollo y producción. La característica de exportación de phpMyAdmin puede generar una copia de seguridad y se puede usar para mandar datos a otras aplicaciones.

Volcado de memoria, copias de seguridad y exportaciones

Primero aclaremos algunos términos del vocabulario. En los documentos MySQL, encontrará el término **copia de la memoria**, en otras aplicaciones, el término **copia de seguridad** o **exportación**. Todos estos términos tienen el mismo significado en el contexto de phpMyAdmin.

MySQL incluye una utilidad línea de comando **mysqldump** que se puede usar para generar archivos de exportación. Pero no todos los proveedores de servidores ofrecen el acceso al entorno operativo necesario para usar las utilidades de la línea de comando. También, acceder a la característica de exportación que está en la interfaz Web es más conveniente. Por eso phpMyAdmin (desde la versión 1.2.0) ofrece la característica de **Exportación** con más formatos de exportación que **mysqldump**. En este capítulo nos centraremos en la característica de exportación de phpMyAdmin.

También existe un modo de exportación de todo el servidor. Este modo solo está disponible para los usuarios MySQL con permisos y se discute en el capítulo 18.

Antes de empezar una exportación, debemos tener una imagen clara del objetivo de la exportación, las siguientes preguntas pueden ayudarle:

- ¿Necesitamos la base de datos completa o sólo unas tablas?
- ¿Necesitamos solo la estructura, solo los datos o ambos?
- ¿Qué utilidad se usará para importar los datos (phpMyAdmin no puede importar todos los formatos de exportación)?
- ¿Queremos solo un subconjunto de los datos?
- ¿Cuál es el tamaño de la exportación que se quiere hacer? ¿Qué velocidad tiene el enlace entre nosotros y el servidor?

Exportación de base de datos

En la vista base de datos, haga clic en el enlace **Exportar**. El panel de exportación por defecto tiene este aspecto:

Ver el volcado esquema de la base de datos

Exportar

autor

libros

Seleccionar todo / Deseleccionar todo

☒ SQL

☐ LaTeX

☐ PDF

☐ Microsoft Excel 2000

☐ Microsoft Word 2000

☐ CSV para datos de MS Excel

☐ Datos CSV

Opciones SQL?

Añadir su propio comentario en el encabezado (ñ segmenta las oraciones):

☐ Incluir lo exportado en una transacción

☐ Deshabilitar la revisión de las llaves extranjeras (foreign keys)

Compatibilidad con exportación SQL: NONE ?

☒ Estructura

☐ Añadir DROP TABLE

☐ Añadir IF NOT EXISTS

☒ Añadir el valor AUTO_INCREMENT

☒ Usar "backquotes" con tablas y nombres de campo

Añadir en los comentarios:

☐ Fechas de creación/actualización/revisión

☒ Datos

☐ Completar los INSERTS

☐ INSERTs extendidos

Longitud máxima de la consulta creada: 50000

☐ Usar "inserts" con retraso

☐ Usar la opción ignore inserts

☒ Usar hexadecimal para campos binarios

Tipo de exportación: INSERT

☐ Enviar (genera un archivo descargable)

Plantilla del nombre del archivo ⁽¹⁾: _DB_ (☒ recordar la plantilla)

Compresión: ☒ Ninguna ☐ "comprimido con zip" ☐ "Comprimido con gzip"

Continuar

Los valores seleccionados aquí por defecto dependen de `config.inc.php`, más específicamente del conjunto de parámetros `$cfg['Export']`. Por ejemplo, el parámetro `$cfg['Export']['format']` está configurado en `'sql'`, lo que hace que el modo de exportación elegido por defecto sea `SQL`. El panel de exportación abarca tres sub-panes. El panel superior **Exportar** y el panel inferior Guardar como fichero siempre están aquí, el tercer panel varía (usa técnicas de menú dinámicas), para mostrar las opciones del modo de exportación elegido (aquí `SQL`).

Sub-panel Exportar

Este sub-panel contiene un selector de tablas en el que podemos elegir la(s) tabla(s) y el formato que queremos. El formato `SQL` es útil para nuestras necesidades ya que crea comandos `SQL` estándar que funcionan en cualquier servidor `SQL`. Otros posibles formatos incluyen **LaTeX**, **Comma-Separate Values (CSV)** y **XML**. Otro formato, **Native MS Excel**, está disponible después de instalar más programas y configurarlos (ver la sección *Native MS Excel* de este capítulo).



Incluso si podemos exportar desde phpMyAdmin en todos estos formatos, sólo los formatos `SQL` y `CSV` se pueden importar con la versión actual de phpMyAdmin. Use sólo estos dos formatos para hacer copias de seguridad.

Ahora discutiremos los formatos (y las opciones disponibles una vez que los hemos elegido) que se pueden elegir en el sub-panel **Exportar**.

Formato SQL

Empezaremos haciendo clic en **Seleccionar todos**; queremos todas las tablas. Sabemos que las tablas son pequeñas así que la pantalla de exportación que aparecerá no será muy grande. Luego haremos clic en **Continuar** que produce los siguientes datos de salida:

```
-- phpMyAdmin SQL Dump
-- version 2.8.0.4
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Tiempo de generación: Jul 05, 2006 a 01:10 PM
-- Versión del servidor: 4.0.18
-- Versión de PHP: 5.0.5-dev
--
-- Base de datos: 'dbbook'
--
```

```
-- -----  
  
--  
-- Estructura de la tabla para la tabla 'autor'  
--  
  
CREATE TABLE 'autor' (  
  'autor_id' int(11) NOT NULL default '0',  
  'nombre_autor' varchar(30) NOT NULL default '',  
  'teléfono' varchar(30) default NULL,  
  PRIMARY KEY `('autor_id')  
) TIPO=MyISAM;  
  
--  
-- Copia en la memoria de los datos de la tabla 'autor'  
--  
  
INSERT INTO 'autor' VALUES (1, 'John Smith', '111-1111');  
INSERT INTO 'autor' VALUES (2, 'Maria Sunshine', '222-2222');  
  
-- -----  
  
--  
-- Estructura de la tabla para la tabla 'libros'  
--  
  
CREATE TABLE 'libros' (  
  'isbn' varchar(25) NOT NULL default '',  
  'título' varchar(100) NOT NULL default '',  
  'número_pag' int(11) NOT NULL default '0',  
  'autor_id' int(11) NOT NULL default '0',  
  'idioma' char(2) NOT NULL default 'en',  
  'descripción' text NOT NULL,  
  'foto_cubierta' blob NOT NULL,  
  'género' set ('Fantasía', 'Niño', 'Novela') NOT NULL default  
'Fantasía',  
  'fecha_publicación' date NOT NULL default '00/00/0000',  
  PRIMARY KEY ('isbn'),  
  KEY 'por_título' ('título (30)'),  
  FULLTEXT KEY 'descripción' ('descripción')  
) TIPO=MyISAM;  
  
--  
-- Copia en la memoria de los datos de la tabla 'libros'  
--  
  
INSERT INTO 'libros' VALUES ('1-23567-89-0', 'Cien años de cine  
(volumen 1)', 600, 1, 'en', '', '', '', '00/00/0000');  
INSERT INTO 'libros' VALUES ('1-23567-22-0', 'Recuerdos futuros', 200,  
2, 'en', '', '0x8950e004954e44ae426082', '', '00/00/0000');
```

```
INSERT INTO 'libros' VALUES ('1-23567-90-0', 'Cien años de cine
(volumen 1)', 600, 2, 'en', '', '', '', '00/00/0000');
```

En este ejemplo de exportación, los datos del segundo libro (empezando con 0x8950) se han truncado para ser breves. De hecho, contendría la representación completa hexadecimal del campo `foto_cubierta` de este libro.

La primera parte de la exportación abarca comentarios (empieza con los caracteres `--`) que detallan la utilidad (y la versión) que ha creado el fichero, los datos y otra información del entorno. Luego veremos las consultas `CREATE` e `INSERT` para cada tabla.

Desde la versión 2.6.0, phpMyAdmin genera comentarios compatibles ANSI en el fichero de exportación; los comentarios empiezan con `--`. Esto ayuda a importar el fichero en sistemas compatibles ANSI SQL. En las versiones previas, se usaba el carácter específico `#` MySQL.

Opciones SQL

Las opciones de SQL se usan para definir exactamente que datos contendrá la exportación. Puede que queramos ver la estructura, los datos o ambos. Seleccionar **Estructura** genera una sección con consultas `CREATE` y seleccionar Datos produce consultas `INSERT`:

Opciones SQL?

Añadir su propio comentario en el encabezado (ñ segmenta las oraciones):

☐ Incluir lo exportado en una transacción

☐ Deshabilitar la revisión de las llaves extranjeras (foreign keys)

Compatibilidad con exportación SQL: NONE

☒ Estructura

☐ Añadir DROP TABLE

☐ Añadir IF NOT EXISTS

☒ Añadir el valor AUTO_INCREMENT

☒ Usar "backquotes" con tablas y nombres de campo

Añadir en los comentarios:

☐ Fechas de creación/actualización/revisión

☒ Datos

☐ Completar los INSERTS

☐ INSERTs extendidos

Longitud máxima de la consulta creada: 50000

☐ Usar "inserts" con retraso

☐ Usar la opción ignore inserts

☒ Usar hexadecimal para campos binarios

Tipo de exportación: INSERT

Las opciones en la sección **Estructura** son:

- Añadir un comentario personalizado en la cabecera: Podemos añadir nuestros propios comentarios en esta exportación; por ejemplo 'Copia de seguridad mensual'; que se verá en las cabeceras de exportación (después del número de versión de PHP). Si los comentarios tienen más de una línea, debemos usar el carácter especial `\n` para separar cada línea.
- Cerrar la exportación en una transacción: Desde la versión MySQL 4.0.11, podemos usar la sentencia `START TRANSACTION`. Este comando, combinado con `SET AUTOCOMMIT=0` en el principio y `COMMIT` en el final, le pide a MySQL que ejecute la importación; cuando reimportemos este fichero; en una transacción, de ese modo se asegura que los cambios se hacen en conjunto
- Desactivar la verificación de las claves foráneas: En el fichero de exportación podemos añadir sentencias `DROP TABLE`. Sin embargo, normalmente, una tabla no se puede vaciar si está referenciada en una restricción de clave foránea. Esta opción invalida la verificación al añadir `SET FOREIGN_KEY_CHECKS=0` al fichero de exportación.
- Añadir `DROP TABLE` : Añade una sentencia `DROP TABLE IF EXISTS` antes de cada sentencia `CREATE TABLE`, por ejemplo `DROP TABLE IF EXISTS 'autor' ;`. De esta manera, podemos asegurarnos que el fichero de exportación se puede ejecutar en una base de datos en la que ya exista la misma tabla, actualizando su estructura pero destruyendo los contenidos anteriores de la tabla.
- Añadir `IF NOT EXISTS`: Añade el modificador `IF NOT EXISTS` a las sentencias `CREATE TABLE`, lo que impide que haya un error si la tabla ya existe.
- Añadir el valor `AUTO-INCREMENT`: Pone información auto-increment de las tablas que están en la exportación, lo que asegura que las filas que se han insertado en las tablas recibirán el siguiente valor auto-increment ID correcto.
- Cerrar los nombres de tabla y campos con comillas especiales: Las comillas especiales son una manera normal de proteger los nombres de tabla y campo que tiene caracteres especiales. En muchos casos es útil usarlas, excepto si el servidor destino (al que se importará el fichero exportado) funciona con una versión MySQL más antigua que 3.23.6, que no soporta las comillas especiales.
- Añadir en los comentarios: Pone información (en forma de comentarios SQL) que no se puede importar directamente, pero que sin embargo representa información de la tabla que los humanos pueden leer. La cantidad de información de aquí varía, dependiendo de la configuración del sistema relacional (ver el capítulo 11). De hecho, con un sistema relacional activado, tendremos las siguientes elecciones:

Añadir en los comentarios:

- ☐ Fechas de creación/actualización/revisión
- ☐ Relaciones
- ☐ MIME-type

Si seleccionamos todas estas elecciones, se producirá esta exportación de la estructura más completa:

```
CREATE TABLE 'libros' (
  'isbn' varchar(25) NOT NULL default '',
  'título' varchar(100) NOT NULL default '',
  'número_pag' int(11) NOT NULL default '0',
  'autor_id' int(11) NOT NULL default '0',
  'idioma' char(2) NOT NULL default 'en',
  'descripción' text NOT NULL,
  'foto_cubierta' mediumblob NO NULL,
  'género' set ('Fantasía', 'Niño', 'Novela') NOT NULL default
'Fantasía',
  'fecha_publicación' date NOT NULL default '00/00/0000',
  PRIMARY KEY ('isbn'),
  KEY 'por_título' ('título'),
  KEY 'autor_id' ('autor_id')
) TIPO=MyISAM COMMENT= 'Contiene la descripción del libro';

--
-- COMENTARIOS PARA LA TABLA 'libros':
--   'isbn'
--     'número de libro'
--   'número_pag'
--     'aproximado'
--   'autor_id'
--     'ver tabla autores'
--
--
-- TIPOS MIME PARA LA TABLA 'libros':
--   'fot0_cubierta'
--     'image_jpeg'
--   'fecha_publicación'
--     'text_plain'
--   'descripción'
--     'text_plain'
--
-- RELACIONES PARA LA TABLA 'libros':
--   'autor_id'
--     'autor' -> 'autor_id'
```


Las opciones disponibles en la sección **Datos** son:

- Inserciones completas: Genera la siguiente exportación para la tabla autor:

```
INSERT INTO 'autor' ('autor_id', 'nombre_autor', 'teléfono') VALUES
(1, 'John Smith', '+01 445 789-1234');
INSERT INTO 'autor' ('autor_id', 'nombre_autor', 'teléfono') VALUES
(2, 'Maria Sunshine', '+01 455 444-4683');
```

Tenga en cuenta que todos los nombres de columna están presentes en cada sentencia. El fichero final es más grande pero será más portátil en diferentes sistemas SQL, con el beneficio añadido de estar mejor documentado.

- Inserciones extendidas: Guarda todos los datos en una sola sentencia INSERT:

```
INSERT INTO 'autor' VALUES (1, 'John Smith', '+01 445 789-1234');
(2, 'Maria Sunshine', '+01 455 444-5683);
```

Este método de insertar datos es más rápido que usar INSERT múltiples, pero es inconveniente porque hace que leer el fichero final más difícil. **Inserciones extendidas** también produce un fichero más pequeño, pero debe tener en cuenta por adelantado de no se puede ejecutar cada línea de este fichero porque no todas las líneas tienen una sentencia INSERT. Si no puede importar el fichero completo en una operación, no puede partir el fichero con un editor de texto e importarlo codificado.

- Usar inserciones suprimidas: Añade un modificador DELAYED a las sentencias INSERT. Esto acelera la operación INSERT porque se pone en cola en el servidor, que se ejecutará cuando la tabla no esté en uso. Por favor tenga en cuenta que es una extensión MySQL no estándar sólo está disponible para las tablas MyISAM e ISAM.
- Usar ignorar inserciones: Normalmente, durante la importación, no podemos insertar valores duplicados para las claves; esto abortaría la operación insertar. Esta opción añade el modificador IGNORE a las sentencias INSERT y UPDATE, de forma que se salta las filas que generan errores de duplicación de claves.
- Usar campos binarios hexadecimales: Un campo con el atributo BINARY puede tener (o no) contenidos binarios. Esta opción hace que phpMyAdmin codifique los contenidos de estos campos en formato 0x. Desmarque esta opción si los campos están marcados BINARY pero están en texto normal, como la tabla mysql.user.
- Tipo de exportación: Las elecciones son INSERTAR, ACTUALIZAR y SUSTITUIR. El más conocido de estos tipos es el que aparece por defecto INSERTAR; si usa sentencias INSERT para importar los datos. Sin embargo, durante la importación, estaremos en una situación en la que la tabla ya existe, tiene datos de valor y sólo queremos actualizar los campos que están

en la tabla *actual* que estamos exportando. ACTUALIZAR genera sentencias como `UPDATE 'autor' SET 'auto_id' = 1, 'nombre_autor' = 'John Smith', 'teléfono' = '111-1111' WHERE 'autor_id' = '1';`—. Sólo actualiza una fila cuando se encuentra la misma clave primaria o una clave única. La tercera posibilidad, SUSTITUIR, produce sentencias como `REPLACE INTO 'autor' VALOR (1, 'John Smith', '111-1111');` que actúan como un `INSERT` para las nuevas filas y actualiza las filas existentes, basado en las claves primaria o única.

Sub-panel Guardar como archivo

En los anteriores ejemplos, los resultados de la operación de exportación se mostraban en la pantalla, por supuesto no se comprimían los datos. Podemos elegir entre transmitir los ficheros por HTTP si marcamos el recuadro **Guardar como fichero**. Esto activa el diálogo Guardar del navegador, que al final guarda el fichero en nuestra extensión local:

☐ Enviar (genera un archivo descargable)

Plantilla del nombre del archivo ⁽¹⁾: (☒ recordar la plantilla)

Compresión: ☒ Ninguna ☐ "Comprimido con zip" ☐ "Comprimido con gzip"

Plantilla nombre de archivo

El nombre del fichero propuesto obedecerá a la **Plantilla nombre de fichero**. En esta plantilla, podemos usar los marcadores de posición `__DB__` y `__TABLE__`, que se cambiarán por la base de datos o el nombre de tabla actuales (en una exportación de una sola tabla). Tenga en cuenta que hay dos caracteres subrayados antes y después de las palabras. También podemos usar caracteres especiales de la función `strftime` de PHP; esto es útil para generar un fichero de exportación basado en la hora y la fecha actuales. Para terminar, podemos poner otras cadenas de caracteres (no forman parte de los caracteres especiales `strftime`), que se usarán literalmente. La extensión del fichero es general, teniendo en cuenta el tipo de exportación. En este caso será `.sql`. Aquí hay unos ejemplos de la plantilla:

- `__DB__` generaría `dbbook.sql`
- `__DB__-%Y%m%d` genera `dbbook-20031206.sql`

La opción **recordar plantilla** cuando está activada, almacena las configuraciones escritas de la plantilla en las cookies (para las exportaciones de base de datos, tabla del servidor) y las saca la próxima vez que usemos el mismo tipo de exportación.

Compresión

Para ahorrar tiempo de transmisión y obtener un fichero de exportación más pequeño. phpMyAdmin puede comprimir en los formatos zip, gzip o bzip2. phpMyAdmin tiene soporte nativo para el formato zip pero los formatos gzip y bzip sólo funcionarán si el servidor PHP se ha compilado con la opción de configuración `--with-zlib` o `--with-bz2`, respectivamente. Los siguientes parámetros controlan que elecciones de compresión se presentan en el panel:

```
$cfg['ZipDump']           = TRUE;
$cfg['GZipDump']          = TRUE;
$cfg['BZipDump']          = FALSE;
```

Un administrador de sistema que instalase phpMyAdmin para un número de usuarios podría configurar todos estos parámetros en `FALSE` para evitar overhead potencial que pasaría si muchos usuarios comprimiesen sus exportaciones al mismo tiempo. Esta situación normalmente causa más overhead que si todos los usuarios estuviesen transmitiendo sus archivos sin comprimir al mismo tiempo.

En las versiones antiguas de phpMyAdmin, el archivo de compresión se construía en la memoria del servidor. Algunos de los problemas que esto causaba eran:

- La generación de los ficheros dependía de los límites de memoria asignados por los programas PHP que estaban en funcionamiento.
- Durante el tiempo cuando se generaba y comprimía el archivo, no se hacía ninguna transmisión, de modo que los usuarios tendían a pensar que la operación no funcionaba y que algo había fallado.
- La compresión de archivos largos era imposible de conseguir.

El parámetro `$cfg['CompressOnFly']` (que está configurado `TRUE` por defecto) se añadió para generar un archivo comprimido que tuviese más cabeceras (para los formatos gzip y bzip2). Ahora, la transmisión empieza casi de inmediato. El archivo se envía en pequeñas codificaciones; el proceso entero consume mucha menos memoria.

Formato de datos CSV

Muchos programas entienden este formato, lo encontrará útil para intercambiar datos. Tenga en cuenta que es un formato de datos, no hay estructura SQL aquí.

Ver el volcado esquema de la base de datos

Exportar

autor

libros

Seleccionar todo / Deseleccionar todo

- ☐ SQL
- ☐ LaTeX
- ☐ PDF
- ☐ Microsoft Excel 2000
- ☐ Microsoft Word 2000
- ☐ CSV para datos de MS Excel
- ☒ Datos CSV

Opciones CSV

Campos terminados en

Campos encerrados por

Caracter de escape

Líneas terminadas en

Reemplazar NULL con

☐ Poner los nombres de campo en la primera hilera

Las opciones disponibles son:

- Campos terminados por: Ponemos una coma aquí lo que quiere decir que se pondrá una coma detrás de cada campo.
- Campos cerrados por: Ponemos un carácter de cierre aquí (como por ejemplo la comilla) para asegurarnos que el campo que contiene un carácter de cierre (coma) no se pone en dos campos.
- Campos escapados por: Si el generador de la exportación encuentra el carácter **Campos cerrados por** dentro de un campo, el carácter del Campo escapados por se pondrá delante para protegerlo. Por ejemplo, "John\ "El Grande\" Smith".
- Líneas terminadas por: Esto decide que carácter termina cada línea. Deberíamos usar el delimitador de línea correcto, dependiendo del sistema operativo con el manipularemos el archivo de exportación resultante. Aquí elegimos \n para una línea nueva estilo UNIX.
- Sustituir NULL por: Si el sistema encuentra un valor NULL en un campo, esto decide que cadena lo sustituirá en el archivo de exportación.
- Poner los nombres del campo en la primera fila: Esto da información sobre el significado de cada campo. Algunos programas usarán esta información para nombrar la columna.

Para terminar seleccionamos la tabla **autor**.

El resultado es:

```
"autor_id", "nombre_autor", "teléfono"
"1", "John Smith", "+01 445 789-1234"
"2", "Maria Sunshine", "+01 455 444-5683"
```

CSV para datos MS Excel

Esta opción puede ser útil para las versiones antiguas de Excel, pero ya no es necesaria. Los datos exportados con la opción CSV se pueden importar en las versiones recientes de Excel en Windows y Macintosh.

LaTeX

LaTeX es una lengua de composición de textos. phpMyAdmin puede generar un archivo `.text` que representa la estructura de los datos y/o los datos en un formato tabular lateral. Tenga en cuenta que este archivo no se puede ver directamente, debe procesarse y convertirse para verlo en el medio final.

Opciones LaTeX

☒ Incluir el subtítulo de la tabla

☒ Estructura

Subtitulado de la tabla: Estructura de la tabla __TABLE__

Continuación del subtítulo de la tabla: Estructura de la tabla __TABLE__

Clave de la etiqueta: tab:__TABLE__-structure

☒ Relaciones

☒ Comentarios

☒ MIME-type

☒ Datos

☒ Nombre de las columnas

Subtitulado de la tabla: Contenido de la tabla __TABLE__

Continuación del subtítulo de la tabla: Contenido de la tabla __TABLE__

Clave de la etiqueta: tab:__TABLE__-data

Reemplazar NULL con: \textit{NULL}

Las opciones disponibles son:

- Incluir la leyenda de las tablas: Presenta las leyendas en los datos de salida tabulares.
- Estructura y Datos : La elección familiar para pedir estructura, datos o ambos.
- Leyenda de la tabla : La leyenda que aparece en la primera página.
- Leyenda de la tabla (continuación): La leyenda que aparece después de la página número uno.

- Relaciones, comentarios, tipo MIME: Otra información de la estructura que queremos que sean parte de los datos de salida. Estas elecciones están disponibles en la infraestructura relacional está puesta (vea el capítulo 11).

El archivo LaTeX generado para los datos de la tabla **autor** tiene este aspecto:

```
% phpMyAdmin LaTeX Dump
% versión 2.8.0.4
% http://www.phpmyadmin.net
%
% Host: localhost
% Tiempo de generación: Jul 05, 2006 a 01:10 PM
% Versión del servidor: 4.0.18
% Versión de PHP: 5.0.5-dev
%
% Base de datos: 'dbbook'
%
%
% Estructura: autor
%
\begin{longtable}{|l|c|c|c|}
\caption{Structure of table authors} \label{tab:authors-structure} \\
\hline\multicolumn{1}{|c|}{\textbf{Field}} & \multicolumn{1}{|c|}{\textbf{Type}} & \multicolumn{1}{|c|}{\textbf{Null}} & \multicolumn{1}{|c|}{\textbf{Default}} \\ \hline \hline
\endfirsthead
\caption{Structure of table authors (continued)} \\
\hline\multicolumn{1}{|c|}{\textbf{Field}} & \multicolumn{1}{|c|}{\textbf{Type}} & \multicolumn{1}{|c|}{\textbf{Null}} & \multicolumn{1}{|c|}{\textbf{Default}} \\ \hline \hline
\endhead
\endfoot
\textbf{\textit{author\_id}} & int(11) & No & 0 \\ \hline
\textbf{\textit{author\_name}} & varchar(30) & No & \\ \hline
\textbf{\textit{phone}} & varchar(30) & Yes & NULL \\ \hline
\end{longtable}

%
% Data: autor
%
\begin{longtable}{|l|l|l|}
\hline \endhead \hline \endfoot \hline
\caption{Content of table authors} \label{tab:authors-data} \\ \hline
\multicolumn{1}{|c|}{\textbf{\textit{author\_id}}} & \multicolumn{1}{|c|}{\textbf{\textit{author\_name}}} & \multicolumn{1}{|c|}{\textbf{\textit{phone}}} \\ \hline \hline
\endfirsthead
\caption{Content of table authors (continued)} \\ \hline
\multicolumn{1}{|c|}{\textbf{\textit{author\_id}}} & \multicolumn{1}{|c|}{\textbf{\textit{author\_name}}} & \multicolumn{1}{|c|}{\textbf{\textit{phone}}} \\ \hline
```

```
\hline \endhead \endfoot
1 & John Smith & 111-1111 \\ \hline
2 & John Smith & 222-2222 \\ \hline
\end{longtable}
```

XML

Este formato es muy popular hoy en día para intercambiar datos. Si elige **XML** en la interfaz Exportar no produce una elección de opciones. Lo que sigue son los datos de salida para la tabla autor:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!--
-
- phpMyAdmin XML Dump
- version 2.5.5-rc2
- http://www.phpmyadmin.net
-
- Host: localhost
- Tiempo de generación: En 22, 2004 a 03:21 PM
- Versión del servidor: 4.0.11
- Versión de PHP: 4.3.1
-->
<!--
- Base de datos: 'dbbook'
-->
<dbbook>
  <!-- Table authors -->
  <autor>
    <autor_id>1</autor_id>
    <nombre_autor>John Smith</nombre_autor>
    <teléfono >111-2222</teléfono>
    <código país></código país>
  </autor>
  <autor>
    <autor_id>2</autor_id>
    <nombre_autor >Maria Sunshine</nombre_autor>
  <teléfono>111-2222</teléfono>
    <código país></código país>
  </autor>
  <autor>
    <autor_id>101</autor_id>
    <nombre_autor >Melanie Smith</nombre_autor>
    <teléfono>111-2222</teléfono>
    <código país></código país>
  </autor>
```

```
<autor>
  <autor_id>100</autor_id>
  <nombre_autor>Paul Smith</nombre_autor>
  <teléfono>111-2222</teléfono>
  <código país ></código país
</autor>
</dbbook>
```

Native MS Excel

Desde la versión 2.6.0, phpMyAdmin ofrece un módulo *experimental* para exportar directamente en formato `.xls`, el formato nativo de hoja de cálculo que **MS Excel** y **OpenOffice Calc** entienden. Cuando este soporte está activado (más en este momento) vemos una nueva elección de exportación:

The screenshot shows the 'Ver el volcado esquema de la base de datos' window in phpMyAdmin. The 'Exportar' tab is active, displaying a list of tables with 'autor' and 'libros' selected. Below the table list are links for 'Seleccionar todo / Deseleccionar todo' and a list of export formats: SQL, LaTeX, PDF, Microsoft Excel 2000, Microsoft Word 2000, CSV para datos de MS Excel (which is selected), and Datos CSV. To the right, the 'Opciones de Excel' section contains a text input for 'Reemplazar NULL con' set to 'NULL', a dropdown for 'Edición Excel' set to 'Windows', and a checked checkbox for 'Poner los nombres de campo en la primera hilera'.

Opcionalmente podemos poner los nombres de nuestros campos en la primera fila de la hoja de cálculo, con **Poner los nombres de campo en la primera fila**.

Esta funcionalidad se basa en el módulo PEAR `Spreadsheet_Excel_Writer`, que en estos momentos está en la versión 0.8 y genera archivos de formato Excel 5.0. Este modulo está documentado en http://pear.php.net/package/Spreadsheet_Excel_Writer, pero la instalación completa en el contexto de phpMyAdmin está documentada aquí.

1. Asegúrese que el servidor PHP tiene soporte PEAR (el comando `pear` fallará si no tenemos soporte PEAR). PEAR está documentado en <http://pear.php.net>.
2. Si tenemos PHP en el modo seguro, tenemos que asegurarnos que tenemos permiso para incluir los módulos PEAR. Asumiendo que los módulos estén bajo `/usr/local/lib/php`, debemos tener la línea `safe_mode_include_dir = /usr/local/lib/php` in `php.ini`.
3. Cuando instalamos el módulo con `pear -d preferred_state=beta install -a Spreadsheet_Excel_Writer` (ya que el módulo en estos momentos está en beta). Este comando extrae los módulos necesarios de Internet y los instala en nuestra infraestructura PEAR.
4. Necesitamos un directorio temporal; que esté bajo el directorio principal de phpMyAdmin; para generar `.xls`. Se puede crear en un sistema Linux con `mkdir tmp ; chmod o+rw tmp`.
5. Configuramos el parámetro `$cfg['TempDir']` en `config.inc.php` a `'./tmp'`.

Ahora deberíamos poder ver la nueva elección de exportación Native MS Excel.

Exportación de la tabla

En la vista tabla , el enlace **Exportar** muestra el sub-panel de exportación de una tabla específica. Es similar a la exportación de la base de datos, a excepción que no hay un selector de tabla. Sin embargo, hay una sección adicional para exportaciones fraccionadas antes del sub-panel Guardar como.

Exportación de archivos fragmentados

El diálogo **Copiar 3 filas** que empieza en el registro #0 nos permite fraccionar el archivo en codificaciones. Podemos experimentar con varios valores para el número de las filas, dependiendo del tamaño exacto de la fila, para ver cuantas filas se pueden poner en un archivo de exportación antes de llegar a los límites de la memoria o el tiempo de ejecución del servidor. Podemos usar nombres como por ejemplo libro00.sql y libro01.sql para nuestros archivos de exportación.

Exportaciones selectivas

En varios lugares de la interfaz de phpMyAdmin, podemos exportar los resultados que vemos o elegir *que* filas queremos exportar.

Exportar resultados parciales de consulta

Cuando los resultados se muestran en phpMyAdmin, aparece un enlace **Exportar** en la parte inferior de la página.

Exportar la estructura y los datos

Mostrando registros 0 - 0 (1 total, La consulta tardó 0.0492 seg)

consulta SQL:

```
SELECT *  
FROM "libros"  
WHERE "autor_id" =2  
LIMIT 0 , 30
```

[Editar] [Explicar el SQL] [Crear código PHP]

Mostrar: 30 filas empezando de 0
en modo horizontal y repetir los encabezados cada 100 celdas

	ISBN	Título	número_pág	autor_id	idioma	descripción	foto-cubierta	género	fecha_publicación
☐	1-234567-2...	Recuerdos ...	200	2	en		[BLOB - 0 Bytes]	F	0000-00-00...

Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Mostrar: 30 filas empezando de 0
en modo horizontal y repetir los encabezados cada 100 celdas

Insertar nueva fila Vista de impresión Previsualización para imprimir (documento completo) Exportar

Si hacemos clic en este enlace, se presenta un panel de exportación especial que tiene la consulta en la parte superior junto con el resto de opciones de exportación de la tabla:

consulta SQL:

```
SELECT *  
FROM "libros"  
WHERE "autor_id" =2
```

[Editar] [Explicar el SQL] [Crear código PHP]

Mostrar el listado esquema de la tabla

Exportar

- ☒ SQL
- ☐ LaTeX
- ☐ PDF
- ☐ Microsoft Excel 2000
- ☐ Microsoft Word 2000
- ☐ CSV para datos de MS Excel
- ☐ Datos CSV

Opciones SQL

Añadir o propio como texto en el encabezado (la segmenta las oraciones):

☐ Incluir lo exportado en una transacción

☐ Deshabilitar la restitución de los libros extranjeros (foreign keys)

Compatibilidad con exportación SQL: NONE

☒ Estructura

- ☐ Añadir DROP TABLE
- ☐ Añadir IF NOT EXISTS
- ☒ Añadir el valor AUTO_INCREMENT
- ☒ Usar "backquotes" con tablas y nombres de campo

Añadir en los comentarios:

- ☐ Fechas de creación/actualización/eliminación
- ☐ Relaciones
- ☐ MIME-type

☒ Datos

- ☒ Completar los INSERTS
- ☒ INSERTS extendidos

Longitud máxima de la consulta creada: 50000

- ☐ Usar "insert" con retablos
- ☐ Usar la opción ignore insert
- ☒ Usar hexadecimal para campos binarios

Tipo de exportación: INSERT

Volcar 1 filas empezando por la fila 0



Los resultados de consultas de una sola tabla se pueden exportar en todos los formatos disponibles, mientras que los resultados de consultas de múltiples tablas sólo se pueden exportar en los formatos CSV, XML y LaTeX.

Exportar y recuadros

Cada vez que vemos resultados (cuando navegamos o buscamos, por ejemplo), podemos marcar el recuadro que está al lado de la fila que queremos y usar **Para los elementos seleccionados**: Con el icono exportar para generar una exportación parcial de sólo esas filas.

← T →	isbn	título	número_pag	autor_id	idioma	descripción	foto_cubierta	género	fecha_publicació
☒	1-234567-22-0	Recuerdos de futuro	200	2	en		[BLOB - 0 Bytes]	F	0000-00-00
☐	1-234567-89-0	Cien años de cine (volumen 1)	600	1	en		[BLOB - 0 Bytes]	F	0000-00-00
☒	1-234567-90-0	Cien años de cine (volumen 2)	600	1	en		[BLOB - 0 Bytes]	F	0000-00-00

Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Guardar el archivo de exportación en el servidor

En lugar de transmitir el archivo de exportación por la red con HTTP, es posible guardarlo directamente en el sistema de archivos del servidor Web. Esto puede ser más rápido y menos sensible a los límites del tiempo de ejecución porque la transferencia del servidor al cliente se deriva. Al final, se puede usar un protocolo de transferencia de archivos como FTP o SFTP para recuperar el archivo ya que dejarlo en la misma máquina no nos ofrece una buena protección de copia de seguridad.

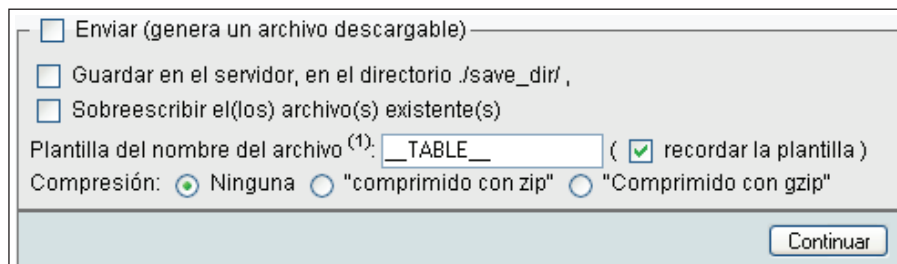
Se tiene que crear un directorio especial en el servidor Web antes de guardar un fichero de exportación en él. Normalmente es un sub-directorio del directorio principal phpMyAdmin. Usaremos `save_dir` como ejemplo. Este directorio debe tener permisos especiales. Primero, el servidor Web debe tener permisos de escritura para este directorio. También, si el componente PHP de este servidor Web funciona en modo seguro, el dueño de los programas phpMyAdmin debe ser el mismo dueño de `save_dir`.

En un sistema Linux; asumiendo el servidor Web funciona como un usuario de **Apache** y los programas los tiene el usuario Marc; los siguientes comandos deberían funcionar:

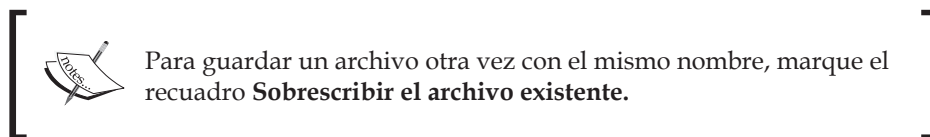
```
# mkdir save_dir
# chown marc.apache save_dir
# chmod g=rwx save_dir
```

También tenemos que definir el nombre del directorio './save_dir' en `$cfg['SaveDir']`. Aquí estamos usando una ruta relativa para el directorio phpMyAdmin, pero una ruta absoluta también funcionaría.

La sección **Guardar como archivo** aparecerá en la nueva sección Guardar en el servidor:



Después de hacer clic en **Guardar**, aparecerá un mensaje de confirmación o un mensaje de error (si el servidor Web no tiene los permisos requeridos para guardar el archivo).



Resumen

En este capítulo:

- Hemos examinado las diferentes maneras de activar una exportación: Desde la vista base de datos, desde la vista tabla o desde una página de resultados.
- Hemos hecho una lista de los diferentes formatos de exportación disponibles, sus opciones, la posibilidad de comprimir un archivo de exportación y los diferentes lugares donde lo podemos enviar.

8

Importar la estructura y los datos

En este capítulo aprenderemos a recuperar los datos exportados que hemos creado para hacer una copia de seguridad o para hacer una transferencia. Los datos exportados también pueden provenir de autores de otras aplicaciones y pueden tener toda la estructura de fundación de estas aplicaciones y algunos datos de ejemplo.

La versión actual de phpMyAdmin puede importar directamente archivos que contengan sentencias MySQL (normalmente tienen un sufijo `.sql`, pero no es necesario). También hay una interfaz en la sentencia MySQL `LOAD DATA INFILE`, que nos permite cargar archivos de texto que tengan datos, normalmente son CSV (comma-separated values), a pesar que el separador puede ser otro carácter. La carga del campo binario que se ha cubierto en el capítulo 6 se puede clasificar dentro de la familia importar.



Importar y cargar son sinónimos en nuestro contexto.

En la versión 2.5.5 de phpMyAdmin, no hay un menú **Importar** en la vista **base de datos** o en la **vista tabla** que contenga todos los enlaces posibles de importación, pero hay un menú **Importar archivos** disponibles dentro de la **ventana de consulta**.

Límites para la transferencia

Cuando importamos, el archivo fuente normalmente está en nuestra máquina cliente, de modo que debe llegar al servidor vía HTTP. Esta transferencia necesita tiempo y usa recursos que pueden estar limitados en el servidor Web por la configuración de PHP.

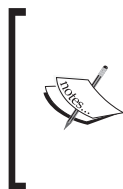
En lugar de usar HTTP, podemos cargar nuestro archivo al servidor usando un protocolo como FTP, como la sección *Carga de un directorio en un servidor Web* describe. Este método evita los límites de carga del servidor PHP.

Límites de tiempo

Primero tengamos en cuenta el límite de tiempo. En `config.inc.php`, la directiva de configuración `$cfg['ExecTimeLimit']` asigna, por defecto, un tiempo de ejecución máximo de 300 segundos (cinco minutos) para *cualquier* programa de phpMyAdmin, incluyendo los procesos de datos cuando se ha cargado el archivo. Un valor de 0 quita el límite y en teoría nos da tiempo infinito para completar la operación de importe. Si el servidor PHP está funcionando en modo seguro, modificar `$cfg['ExecTimeLimit']` no tendrá ningún efecto, porque los límites que están en `php.ini` o en una configuración relacionadas con el usuario (como por ejemplo `.htaccess` o una configuración de proveedor virtual) tienen preferencia sobre este parámetro.

Por supuesto, el tiempo que tiene en efectivo depende de dos factores clave:

- La carga del servidor Web.
- La carga del servidor MySQL.



El tiempo que el archivo toma mientras viaja entre el cliente y el servidor no se cuenta como tiempo de ejecución, porque el programa de PHP sólo empieza a ejecutarse cuando el archivo se ha recibido en el servidor. De modo que el parámetro `$cfg['ExecTimeLimit']` sólo tiene impacto en el tiempo que se usa para procesar los datos (como por ejemplo la descompresión o enviarlo al servidor MySQL).

Otros límites

Si usamos el archivo `php.ini` o la configuración de proveedor virtual del servidor Web, el administrador del sistema puede controlar las cargas del servidor.

El parámetro `upload_max_filesize` especifica el límite superior o el tamaño máximo del archivo que se puede cargar vía HTTP. Esto es obvio, pero hay otro parámetro menos obvio `post_max_size`. Ya que la carga de HTTP se hace a través del método POST, este parámetro puede limitar nuestras transferencias.

El parámetro `memory_limit` sirve para evitar que los procesos del servidor Web hijo tomen mucha memoria del servidor Web; phpMyAdmin también funciona como un proceso hijo. Así pues, la gestión de cargas de fichero normales, en especial las copias de memoria comprimidas, pueden verse comprometida si ponemos un valor

pequeño para este parámetro. Aquí, se recomienda un valor no preferido; depende del tamaño de los datos cargados.

Para terminar, las cargas de los archivos deben estar permitidas configurando `file_uploads` en On. Sino, phpMyAdmin ni siquiera mostrará el diálogo **Posición del archivo texto**. Sería inútil presentar este diálogo ya que la conexión sería rechazada más tarde por el servidor PHP.

Importar archivos SQL¹

Cualquier archivo que contenga sentencias MySQL se puede importar con este mecanismo. El diálogo está disponible en la vista base de datos o en la vista tabla, a través de la sub-página SQL, o en la **ventana de consulta**. La parte importante a tener en cuenta es el diálogo **Posición del archivo texto** :



No hay una relación entre la tabla que se ha seleccionado (aquí **autor**) y los contenidos actuales que se importarán del archivo SQL. Todos los contenidos del archivo SQL se importarán y estos contenidos serán los que determinen que tabla o base de datos se verán afectadas. Sin embargo, si el archivo importado no tiene sentencias SQL para seleccionar una base de datos, todas las sentencias del archivo importado se ejecutarán en la base de datos seleccionada.

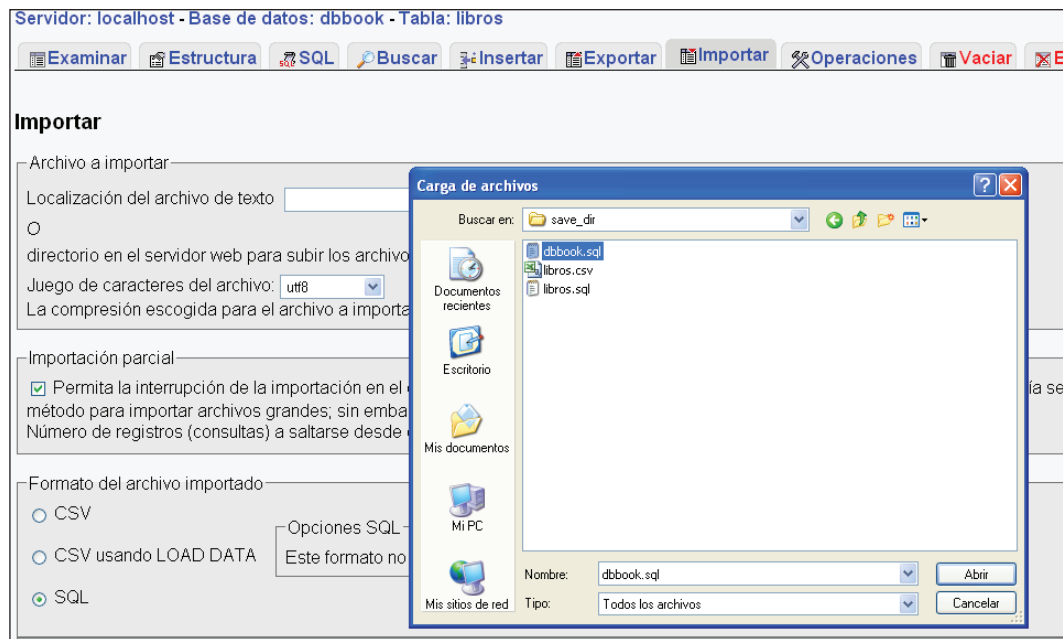
Hagamos un ejercicio de importación. Primero nos aseguramos que tenemos una exportación actual de la tabla **libros** (como se explicó en el capítulo 7). Este archivo de exportación debe contener la estructura y los datos. Luego **vaciamos** la tabla **libros** (¡sí, de verdad!). También podemos renombrarla (vea el capítulo 10 para ver como se hace).

1. En las versiones superiores de phpMyAdmin existe una pestaña Importar, es aquí donde se encuentran todos diálogos Importar.

Ahora podemos importar el archivo. Deberíamos estar en la página donde podemos ver el diálogo **Posición del archivo texto**. Sólo tenemos que hacer clic en el botón **Examinar** para elegir nuestro archivo.

También podemos especificar que método de compresión se ha aplicado al archivo o dejar que phpMyAdmin lo auto detecte. En este caso, usamos la elección por defecto **Auto detectar**.

Para empezar la importación hacemos clic en **Continuar**:



Podemos confirmar los resultados si verificamos la sentencia `CREATE TABLE` y el número de filas afectadas por cada sentencia `INSERT`. También podemos examinar nuestra tabla recién creada para confirmar el éxito de la operación de importación.

La importación se ejecutó exitosamente, se ejecutaron 3 consultas.

consulta SQL:

```
-- phpMyAdmin SQL Dump
-- version 2.8.0.4
-- http://www.phpmyadmin.net
--
-- Servidor: localhost:3306
-- Tiempo de generación: 05-06-2006 a las 22:21:08
-- Versión del servidor: 4.1.19
-- Versión de PHP: 5.1.4
--
-- Base de datos: `dbbook`
--
-----
--
-- Estructura de tabla para la tabla `libros`
--
DROP TABLE IF EXISTS `libros`; # MySQL ha devuelto un valor vacío (i.e., cero columnas).
CREATE TABLE `libros` (
  `isbn` VARCHAR(25) NOT NULL DEFAULT '',
  `título` VARCHAR(100) NOT NULL DEFAULT '',
  `número_pag` INT(11) NOT NULL DEFAULT '0',
  `autor_id` INT(11) NOT NULL DEFAULT '0',
  `idioma` CHAR(2) NOT NULL DEFAULT 'en',
  `descripción` TEXT NOT NULL ,
  `foto_cubierta` BLOB NOT NULL ,
  `género` SET('F','N','N') NOT NULL DEFAULT 'F',
  `fecha_publicación` DATE NOT NULL DEFAULT '0000-00-00',
  PRIMARY KEY (`isbn`),
  KEY `por_título` (`título` (30)),
  KEY `autor_idioma` (`autor_id`, `idioma`)
) ENGINE = INNODB DEFAULT CHARSET = latin1; # MySQL ha devuelto un valor vacío (i.e., cero columnas).
--
-- Volcar la base de datos para la tabla `libros`
--
INSERT INTO `libros` (`isbn`, `título`, `número_pag`, `autor_id`, `idioma`, `descripción`, `foto_cubierta`, `género`,
`fecha_publicación`)
VALUES (
  '1-234567-22-0', 'Recuerdos de futuro', 200, 2, 'en', '', '', 'F', '0000-00-00'
), (
  '1-234567-89-0', 'Cien años de cine (volumen 1)', 600, 1, 'en', '', '', 'F', '0000-00-00'
), (
  '1-234567-90-0', 'Cien años de cine (volumen 2)', 600, 1, 'en', '', '', 'F', '0000-00-00'
); # Filas afectadas: 3
```

El archivo se puede importar en una base de datos diferente o incluso en el servidor MySQL para verificar.

Importar archivos CSV

En esta sección, examinaremos como importar archivos CSV .

Diferencias entre los formatos SQL y CSV

Hay algunas diferencias entre estos dos formatos. El archivo CSV *solo* tiene datos, de modo que ya debemos tener una tabla existente en su lugar. Esta tabla no necesita tener la misma estructura que la tabla original (de donde provienen los datos); el diálogo **Nombre de columna** nos permite elegir que columnas se ven afectadas en la tabla destino.

Otra diferencia es que phpMyAdmin se basa en los mecanismos `LOAD DATA INFILE` o `LOAD DATA LOCAL INFILE` para hacer la importación en lugar de procesar los datos internamente. Estas sentencias son la manera más rápida de importar texto en MySQL. Hacen que MySQL empiece una operación de lectura del archivo que está en el servidor MySQL (`LOAD DATA INFILE`) o de otro lugar (`LOAD DATA LOCAL INFILE`), que, en este contexto, siempre es el sistema de archivos del servidor Web. Si el servidor MySQL está en un ordenador diferente al del servidor Web, no podremos usar el mecanismo `LOAD DATA INFILE`.

Requisitos

Confiar en el servidor MySQL tiene algunas consecuencias. Para usar `LOAD DATA INFILE` es necesario que el usuario conectado tenga un permiso `FILE` global. Además, los procesos del servidor MySQL deben leer el archivo.



El capítulo 18 explica la administración de permisos de la interfaz phpMyAdmin para los administradores de sistema.

Si usamos el modificador `LOCAL` en `LOAD DATA LOCAL INFILE`, debemos tener permiso del servidor MySQL y de la librería cliente MySQL que usa PHP.

Ambos métodos `LOAD` están disponibles en la interfaz **CARGAR** de phpMyAdmin, que intenta elegir la mejor opción por defecto.

Usar la interfaz LOAD

Empezaremos generando un archivo de exportación `autor.csv`, esta vez de la tabla **autor**. En la exportación CSV usamos los valores por defecto. Luego podemos **Eliminar** la tabla autor (aún necesitamos la estructura de la tabla).

Para encontrar la página diálogo de importación tenemos que estar en la vista tabla de la tabla **autor**.

El **Insertar datos de un archivo texto** en una tabla se encuentra en varias páginas:

- La página **Estructura**.
- La página **Importar**.
- La pestaña **Importar archivos** en la **ventana de consulta**.

Elegimos empezar en la página **SQL**, el enlace **Insertar datos de un archivo texto** en una tabla nos muestra el siguiente diálogo:

Archivo a importar

Localización del archivo de texto Examinar... (Tamaño máximo: 2,048KB)

Juego de caracteres del archivo: utf8

La compresión escogida para el archivo a importar se detectará automáticamente de: Ninguna, gzip, zip

Importación parcial

☒ Permita la interrupción de la importación en el caso de que el script detecte que se ha acercado a su límite de tiempo. Esto podría ser un buen método para importar archivos grandes; sin embargo, puede dañar las transacciones.

Número de registros (consultas) a saltarse desde el inicio

Formato del archivo importado

☐ CSV

☒ CSV usando LOAD DATA

☐ SQL

Las opciones para la importación CSV usando LOAD DATA

☐ Reemplazar los datos de la tabla con los del archivo

☐ Ignore las filas duplicadas

Campos terminados en

Campos encerrados por

Caracter de escape

Líneas terminadas en

Nombre de las columnas

☐ Use la palabra clave LOCAL

Continuar

Hay muchas opciones porque los datos pueden provenir de varios lugares; la manera de codificar los campos e incluso el carácter de fin de línea varían dependiendo del programa que se ha usado para genera el archivo o el sistema operativo fuente.

Primero vemos la pregunta familiar **Posición del archivo texto**. Elegimos nuestro archivo `autor.csv` y por ahora sólo hacer clic en **Enviar**, exploraremos otras opciones más tarde.

Si todo funciona, vemos la pantalla de confirmación:

La importación se ejecutó exitosamente, se ejecutaron 1 consultas.

consulta SQL:

LOAD DATA INFILE 'C:\WINDOWS\php43.tmp' INTO TABLE `libros` FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\W' LINES TERMINATED BY '\n' # Filas afectadas: 3

[[Editar](#)] [[Crear código PHP](#)]

Esta pantalla muestra la sentencia exacta **LOAD DATA LOCAL INFILE** que se ha usado. Tenga en cuenta que el archivo que se ha usado era /tmp/phpU0HGGG y no autor.csv, porque PHP, guardado como un archivo temporal, leído y cargado por MySQL.

Opciones de LOAD

Por defecto, la operación **LOAD** no modificará los datos existentes (basados en claves primaria o única), pero la opción **Sustituir** le ordena a phpMyAdmin añadir una palabra clave **REPLACE** al comando generado, que modificará los datos y las claves que sean las mismas.

Podemos especificar el carácter que termina cada campo, el carácter que cierra los datos, y el carácter que escapa al carácter de cierre. Normalmente es \. Por ejemplo, con un carácter final doble comilla, si el campo de datos tiene comillas dobles, debe ponerse así: **"algunos datos \"otros datos"**.

Para las **Líneas terminadas por**, la elección usar es \n para los sistemas basados en UNIX, \r\n para DOS o los sistemas Windows y \r para los sistemas basados en Mac. Si tenemos dudas, podemos usar un editor hexadecimal en nuestro ordenador cliente (no forma parte de phpMyAdmin) para examinar los códigos exactos.

Por defecto, phpMyAdmin espera un archivo CSV con el mismo número de campos y el mismo orden de campos que la tabla destino, pero esto se puede cambiar si escribimos una lista separada por comas de los nombres de columna en **Nombre de columna**, respetando el formato del archivo fuente. Por ejemplo, digamos que nuestro archivo fuente contiene información sobre el ID del autor y su nombre:

```
"1", "John Smith"
"2", "Maria Sunshine"
```

Tenemos que poner los nombres **autor_id**, **nombre_autor** en **Nombre de columna** para que cuadre con el archivo fuente.

Para terminar, debemos elegir el formato **LOAD**, como se ha explicado antes.

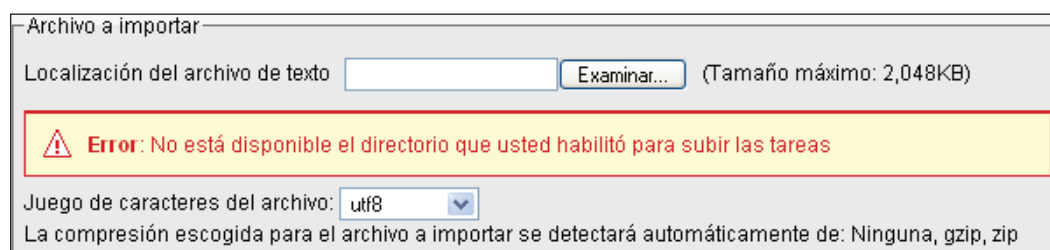
Si el archivo es muy grande, hay maneras para resolver la situación. Si todavía tenemos problemas para acceder a los datos originales, podemos usar phpMyAdmin para generar archivos de exportación CSV más pequeños, eligiendo **Copiar n fila en el registro # n**. Si esto no es posible, tendremos que usar un editor de texto para fragmentar el archivo en secciones más pequeñas. Otra posibilidad es usar el mecanismo UploadDir.

Carga de un directorio al servidor Web

Para solucionar casos en las cargas estén desactivas por completo por la configuración PHP del servidor Web o en los que los límites de carga sean muy pequeños, phpMyAdmin puede leer archivos de carga de un servidor especial que está en el sistema de archivos del servidor Web. Este mecanismo solo se puede usar con importaciones SQL y CSV.

Primero especificamos el nombre del directorio elegido en el parámetro `$cfg['UploadDir'];` por ejemplo: `'./upload'`.

Ahora, volvamos a la sub-página SQL y veamos que pasa:



Este error es de esperar ya que el directorio no existe. Se supone que se ha creado dentro del directorio de instalación phpMyAdmin actual. El mensaje también puede indicar que el directorio existe pero que el servidor Web no puede leerlo (en modo seguro PHP; el dueño del directorio y el dueño de los programas instalados de phpMyAdmin debe ser el mismo).

Si usamos un cliente SFTP o FTP, creamos el directorio necesario y podemos cargar un archivo aquí (esquivando las interrupciones de PHP o los límites máximos de carga), por ejemplo `libros.sql`. Por favor tenga en cuenta que el archivo debe tener permisos para que el servidor lo lea. En la mayoría de los casos lo más fácil es permitir que todo el mundo lea el archivo.

Si actualizamos la sub-página SQL vemos lo siguiente:

Archivo a importar

Localización del archivo de texto (Tamaño máximo: 2,048KB)

☐

directorio en el servidor web para subir los archivos :

Juego de caracteres del archivo:

La compresión escogida para el archivo a importar se detectará automáticamente de: Ninguna, gzip, zip

Hacer clic en **Continuar** ejecutará el archivo.

Importar volcados comprimidos

phpMyAdmin puede descomprimir archivos cargados. Podemos elegir el modo de compresión directamente cuando elegimos el archivo o simplemente usar el modo **Auto detectar** que normalmente debería funcionar. Los formatos que el programa puede descomprimir varían dependiendo de la versión phpMyAdmin y *que* extensiones están disponibles en el componente PHP del servidor Web.

Esta característica también está disponible para los archivos que están en el directorio de carga. Los archivos deben tener extensiones como `.bz2`, `.gz`, `.sql.bz2` o `.sql.gz`.

 Usar extensiones dobles (`.sql.bz2`) es una manera mejor para indicar que un archivo `.sql` se ha producido y se ha comprimido; veremos todos los pasos que generan este archivo.

Resumen

En este capítulo hemos aprendido:

- Las diferentes opciones de phpMyAdmin que nos permiten importar datos.
- Los diferentes mecanismos implicados en la importación de archivos SQL y CSV.
- Los límites que podemos alcanzar cuando intentamos transferir.
- Algunas maneras de esquivar los límites.

9

Buscar datos

Aquí presentamos un mecanismo que se puede usar para encontrar los datos que estamos buscando en lugar de examinar las tablas página por página y ordenarlas. Este capítulo cubre las búsquedas de una tabla y de bases de datos enteras. El capítulo 13 complementa este capítulo y presenta consultas multi-tabla por ejemplo.

Búsqueda en una tabla

Esta sección describe la sub-página de **Búsqueda** en la que está disponible la búsqueda de una tabla.

Uso diario de phpMyAdmin

El uso principal de esta herramienta para algunos usuarios es el **Modo buscar** para encontrar y actualizar datos. Para eso, el equipo de phpMyAdmin ha hecho posible la definición de que página es la página de arranque en la vista de tabla con el parámetro `$cfg['DefaultTabTable']`. Si lo configuramos en `'tbl_select.php'` definiremos la sub-página por defecto de Buscar.

Con este modo, los programadores de aplicaciones pueden buscar datos de maneras que la interfaz que están construyendo no esperaba, ajustando y a veces reparando datos.

Entrar en la sub-página Buscar

Se puede llegar a la sub-página **Buscar** haciendo clic en el enlace **Buscar** de la vista tabla. Esto es lo que se ha hecho aquí para la tabla **libros**:

Seleccionar campos (al menos uno):

- ☒ isbn
- ☒ título
- ☒ número_pag
- ☒ autor_id
- ☒ idioma
- ☒ descripción
- ☒ foto_cubierta
- ☒ género
- ☒ fecha_publicación

☐ DISTINCT

registros por página:

Mostrar en este orden:

☒ Ascendente ☐ Descendente

Insertar las condiciones de búsqueda (cuerpo de la cláusula "where"):

[Continuar](#)

O Hacer una consulta (comodín: "%")

Campo	Tipo	Cotejamiento	Operador	Valor
isbn	varchar(25)	latin1_swedish_ci	LIKE	
título	varchar(100)	latin1_swedish_ci	LIKE	
número_pag	int(11)		=	
autor_id	int(11)		=	
idioma	char(2)	latin1_swedish_ci	LIKE	
descripción	text	latin1_swedish_ci	LIKE	
foto_cubierta	blob		LIKE	
género	set('F', 'Ni', 'N')	latin1_swedish_ci	LIKE	
fecha_publicación	date		=	

[Continuar](#)

Selección de los campos mostrados

El primer panel facilita una selección de los campos que se muestran en los resultados:

Seleccionar campos (al menos uno):

isbn
título
número_pag
autor_id
idioma
descripción
foto_cubierta
género
fecha_publicación

☐ DISTINCT

Por defecto, todos los campos están seleccionados, pero podemos controlar los campos haciendo clic para elegir los necesarios.

Aquí están los campos que nos interesan en este ejemplo:

También podemos especificar el número de filas por página en el recuadro de texto que está justo debajo del campo de selección. El recuadro **Añadir condiciones de búsqueda** se explicará en la sección *Aplicar una cláusula WHERE*, más tarde en este capítulo.

Búsqueda de criterios por campo: Consulta por ejemplo

El uso principal del panel **Buscar** es escribir criterios para algunos campos y recuperar sólo los datos en los que estamos interesados. Esto se llama **Consulta por ejemplo** porque le hemos dado un ejemplo de lo que estamos buscando. Nuestra primera recuperación está relacionada con encontrar el libro con ISBN 1-234567-89-0. Sólo tenemos que poner este valor en el recuadro **isbn** y elegir el operador **=**.

○ Hacer una consulta (comodin: "%")

Campo	Tipo	Cotejamiento	Operador	Valor
isbn	varchar(25)	latin1_swedish_ci	LIKE	1-234567-89-0
título	varchar(100)	latin1_swedish_ci	LIKE LIKE %...% NOT LIKE	
número_pág	int(11)		=	
autor_id	int(11)		!= REGEXP NOT REGEXP	
idioma	char(2)	latin1_swedish_ci	LIKE	
descripción	text	latin1_swedish_ci	LIKE	
foto_cubierta	blob		LIKE	
género	set('F', 'Ni', 'N')	latin1_swedish_ci	LIKE	
fecha_publicación	date		=	

Continuar

Si hacemos clic en **Continuar** obtendremos los resultados que se ven abajo. Los cuatro campos que se muestran son los que se han elegido en el diálogo **Elegir campos**:

←T→	isbn	título	autor_id	idioma
☐  	1-234567-89-0	Cien años de cine (volumen 1)	1	en
 Marcar todos/as / Desmarcar todos Para los elementos que están marcados:   				

Esta una página de resultados estándar. Si los resultados estuviesen en varias páginas, podríamos navegar por ellas y editar, borra datos durante el proceso para el *fragmento que elijamos*. Otra característica de phpMyAdmin es que los campos usados como el criterio se resaltan al cambiar el color del borde de las columnas para que se refleje mejor en la página de resultados. No es necesario especificar que la columna **isbn** se tiene que mostrar. Podemos elegir el título de la columna para la *presentación* y seleccionar la columna **isbn** como *criterio*.

Vista imprimir

Vemos los enlaces **Vista imprimir** y **Vista imprimir (con textos completos)** en la página de resultados. Estos enlaces producen un informe más formal de los resultados (sin la interfaz de navegación) que van directamente a la impresora. En nuestro caso, si usamos la **Vista imprimir** vemos lo siguiente:

Resultado SQL				
Servidor: localhost:3306 Base de datos: dbbook Tiempo de generación: 05-06-2006 a las 22:47:28 Generado por: phpMyAdmin 2.8.0.4 / MySQL 4.1.19-community-nt consulta SQL: SELECT `isbn`, `título`, `autor_id`, `idioma` FROM `libros` WHERE `isbn` = CONVERT(_utf8 '1-234567-89-0' USING latin1) COLLATE latin1_swedish_ci LIMIT 0, 30 ; Filas: 1				
isbn	título	autor_id	idioma	
1-234567-89-0	Cien años de cine (volumen 1)	1	en	

Este informe contiene información sobre el servidor, la base de datos, el tiempo de generación, la versión de phpMyAdmin, la versión MySQL y la consulta SQL que se ha usado. El otro enlace, **Vista imprimir (con texto completo)** imprimiría los contenidos completos de los campos TEXT.

Búsqueda con comodines

Asumamos que estamos buscando algo menos preciso: Todos los libros con cine en el título. Primero, volvemos a la página de búsqueda. Para este tipo de búsqueda usaremos el operador `LIKE` de SQL. Este operador acepta caracteres comodín: El

carácter % (que se ajusta a varios caracteres) y el carácter subrayado (_) (que se ajusta a un sólo carácter). De este modo podemos usar %cine% para que phpMyAdmin encuentre cualquier sub-cadena que se ajuste a 'cine'. Si hubiésemos dejado los comodines, encontraríamos ajustes exactos con esa palabra.

Desde la versión 2.6.0 de phpMyAdmin, este ajuste de sub-cadenas se ha hecho más fácil de acceder al añadir un menú desplegable de **Operadores**. Sólo tenemos que escribir la palabra **cine** y usar el operador LIKE %...% para que se haga ese ajuste. Deberíamos usar esta forma del operador LIKE en tablas grandes (miles de filas), ya que MySQL no usa un índice de recuperación de datos en este caso, lo cual nos llevaría a un tiempo de media hora (o más). Por eso este operador no aparece por defecto en la lista desplegable, a pesar que este método de búsqueda se usa comúnmente en las tablas más pequeñas.

En las versiones anteriores a phpMyAdmin 2.6.0, teníamos que insertar manualmente los caracteres % para obtener '%cine%' y usar el operador LIKE de la lista desplegable.

También especificamos que los resultados se ordenan por título (en orden ascendente). Aquí tenemos una muestra de pantalla que nos muestra como pedir una búsqueda de **cine** con el operador LIKE %...%.

☐ Hacer una consulta (comodín: "%")

Campo	Tipo	Cotejamiento	Operador	Valor
isbn	varchar(25)	latin1_swedish_ci	LIKE	
titulo	varchar(100)	latin1_swedish_ci	LIKE %...%	cine
número_pag	int(11)		=	
autor_id	int(11)		=	
idioma	char(2)	latin1_swedish_ci	LIKE	
descripción	text	latin1_swedish_ci	LIKE	
foto_cubierta	blob		LIKE	
género	set('F', 'Ni', 'N')	latin1_swedish_ci	LIKE	
fecha_publicación	date		=	



El operador LIKE se puede usar para otros tipos de búsquedas comodín, por ejemplo, **Historia%**; lo que buscaría esta palabra al principio del título. Esta forma de la consulta LIKE también tiene el beneficio de usar un índice, si MySQL encuentra uno acelera la recuperación de datos.

Si usamos cualquiera de estos métodos de hacer una consulta, obtenemos estos resultados:

←T→	isbn	título	autor_id	idioma
☐  	1-234567-89-0	Cien años de cine (volumen 1)	1	en
☐  	1-234567-90-0	Cien años de cine (volumen 2)	1	en


[Marcar todos/as](#) / [Desmarcar todos](#) Para los elementos que están marcados:   

Los caracteres comodín disponibles son el carácter % (que se ajusta a cualquier número de caracteres) y el carácter subrayado (_) (que se ajusta a un carácter).

Combinar criterios

Podemos usar criterios múltiples para la misma consulta (por ejemplo, encontrar todos los libros ingleses de más de 300 páginas). Aquí vemos que hay más elecciones de comparación ya que el campo número_pag es numérico.

Seleccionar campos (al menos uno):

isbn
 título
 número_pag
 autor_id
 idioma
 descripción
 foto_cubierta
 género
 fecha_publicación

☐ DISTINCT

registros por página:

Mostrar en este orden:

☒ Ascendente
 ☐ Descendente

Insertar las condiciones de búsqueda (cuerpo de la cláusula "where"): 

Continuar

O Hacer una consulta (comodín: "%")

Campo	Tipo	Cotejamiento	Operador	Valor
isbn	varchar(25)	latin1_swedish_ci	LIKE	
título	varchar(100)	latin1_swedish_ci	LIKE	
número_pag	int(11)		>	600
autor_id	int(11)		=	
idioma	char(2)	latin1_swedish_ci	>=	en
descripción	text	latin1_swedish_ci	<	
foto_cubierta	blob		<=	
género	set('F', 'NI', 'N')	latin1_swedish_ci	!=	
fecha_publicación	date		LIKE	
			NOT LIKE	
			LIKE	
			=	

Continuar

Aplicar una cláusula WHERE

A veces queremos escribir una condición de búsqueda que no se ofrece en la lista de **Función** de la sección **Consulta por ejemplo**; la lista no puede contener todas las variaciones disponibles de un idioma. Digamos que queremos encontrar todos los libros en inglés o en francés. Para ello, podemos usar la sección **Añadir condiciones de búsqueda**:

Insertar las condiciones de búsqueda (cuerpo de la cláusula "where"): ⓘ
 idioma ('en','es')

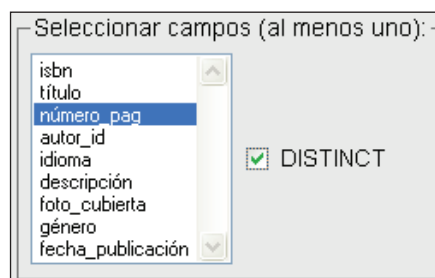
La expresión de búsqueda completa se genera al combinar las condiciones de búsqueda, un AND lógico y el resto de criterios que se han escrito en las líneas **Búsqueda por ejemplo**.

Podemos tener una lista de condiciones de búsqueda más compleja que se escribiría en el mismo recuadro de texto, normalmente entre paréntesis y con operadores AND u OR.

Un enlace **Documentación** nos lleva al manual MySQL, en el que podemos ver la inmensa elección de funciones disponibles (cada función se aplica a un tipo de campo específico).

Obtener resultados DISTINCT

Algunas veces queremos evitar obtener los mismos resultados más de una vez. Por ejemplo, para saber en que ciudades tenemos clientes, con que se vea cada ciudad *una vez* es suficiente. Aquí queremos saber cuantas páginas tienen nuestros libros. En el diálogo **Elegir Campos**, elegimos el campo número_pag y marcamos DISTINCT:



Si hacemos clic en **Continuar** veremos lo siguiente:

←T→			número_pag
☐			200
☐			600

 Si usamos DISTINCT solo veremos dos números de páginas '200' y '600' una vez. Sin esta opción, la fila en la que está '200' hubiese aparecido dos veces.

Búsqueda de una base de datos completa

En los ejemplos anteriores, la búsqueda estaba limitada a una tabla. Con esto asumimos que sabemos en que tabla (y columnas) está almacenada la información necesaria.

Cuando los datos están ocultos en algún lado de la base datos o cuando los *mismos* datos pueden estar en varias columnas (por ejemplo, la columna título o la columna descripción), es más fácil usar el método de búsqueda base de datos.

Podemos entrar en la página **Buscar** en la vista base de datos de la base de datos dbbook:

Servidor: localhost:3306 ▶ Base de datos: dbbook

[Estructura](#)
[SQL](#)
[Buscar](#)
[Generar una consulta](#)
[Exportar](#)
[Importar](#)
[Operaciones](#)

[Privilegios](#)
[Eliminar](#)

Buscar en la base de datos

Palabra(s) o valor(es) a buscar (comodín: "%"):

Encontrado:

- ☒ al menos una de estas palabras¹
- ☐ Todas las palabras¹
- ☐ La frase exacta
- ☐ como expresión regular ⓘ

¹Palabras separadas por un espacio (" ").

En la(s) tabla(s):

[Seleccionar todo](#) / [Deseleccionar todo](#)

[Continuar](#)

En la sección **Palabra(s) o valor(es)**, escribimos lo que queremos encontrar. El comodín % puede ser muy útil aquí. Escribimos **recuerdos**.

En la sección **Encontrar**, especificamos como se trataran los valores que hemos escrito: Puede que necesitemos encontrar cualquier de las palabras que hemos encontrado (en ningún orden particular) o la frase exacta (las palabras en el mismo orden en algún lugar de la columna). Otra elección es usar una expresión regular, que es una manera más compleja de hacer comparación de formas. Mantendremos el valor por defecto **al menos una de las palabras**.

Podemos elegir las tablas para restringir la búsqueda o seleccionar todas las tablas. Ya que sólo tenemos dos tablas (pequeñas), elegimos ambas.



Ya que la búsqueda se hará en cada fila de la tabla elegida, puede que alcancemos los límites de tiempo si el número de filas o tablas es muy grande.

Si hacemos clic en **Continuar** nos encuentra lo siguiente:

Resultados de la búsqueda por "recuerdos" (al menos una de estas palabras):		
0 resultado(s) en la tabla <i>autor</i>		
1 resultado(s) en la tabla <i>libros</i>	Examinar	Borrar
Total: 1 resultado(s)		

Esta es una vista del número de parecidos y las tablas relevantes. Puede que encontremos parecidos en tablas en las que no estamos interesados; sin embargo, los parecidos que parecen prometedores, podemos **Examinarlos** en la página de resultados, o incluso **Suprimir** las filas encontradas.

Resumen

En este capítulo hemos cubierto:

- La búsqueda de una tabla, con criterios de consulta por ejemplo y especificación de criterios adicionales.
- Seleccionar los valores que se muestran y ordenar los resultados.
- Búsqueda con comodines.
- Búsqueda de una base de datos completa

10

Operaciones de las tablas

En los anteriores capítulos, hemos abordado los campos de las tablas. En este capítulo, aprenderemos a desempeñar algunas operaciones que influyen las tablas como un todo. Cubriremos los atributos de las tablas y como modificarlos, también discutiremos las operaciones de tablas múltiples.

En la vista tabla, los diferentes enlaces que activan las operaciones de la tabla se han juntado en una sub-página: **Operaciones**. Aquí tiene una muestra de esta página:

Operaciones

Modificar el ORDER BY de la tabla: isbn (solamente) Ascendente Continuar

Mover tabla a (Base de datos.tabla): dbbook libros Continuar

Opciones de la tabla:

Cambiar el nombre de la tabla a: libros

Comentarios de la tabla:

Motor de almacenamiento: InnoDB

Cotejamiento: latin1_swedish_ci Continuar

Copiar la tabla a (base de datos.tabla):

dbbook .

☐ Unicamente la estructura

☒ Estructura y datos

☐ Solamente datos

☐ Añadir DROP TABLE

☐ Añadir el valor AUTO_INCREMENT

☐ Cambiar (switch) a la tabla copiada Continuar

Mantenimiento de la tabla

- Revisar la tabla
- Defragmentar la tabla
- Vaciar el caché de la tabla ("FLUSH")

Mantenimiento de la tabla

Una tabla, durante su vida, se modifica repetidamente, crece y decrece. Puede que el servidor se interrumpa, dejando algunas tablas dañadas.

Si usamos la sub-página Operaciones, podemos desempeñar varias operaciones, pero no todas las operaciones están disponibles para todos los tipos de tabla:

- **Verificar tabla:** Escanea todas las filas para verificar que los enlaces suprimidos son correctos. Además, se calcula una suma de control para verificar la integridad de las claves; si todo está bien deberíamos ver un mensaje 'OK'.
- **Analizar la tabla:** Analiza y almacena la distribución de las claves; esto se usará en las operaciones subsecuentes JOIN para determinar el orden con el que las tablas deberían unirse.
- **Reparar la tabla:** Repara datos posiblemente corruptos; ¡tenga en cuenta que la tabla puede estar tan corrupta que ni siquiera podamos entrar en la vista tabla para hacerlo! En este caso, vaya a la sección *Operaciones de múltiples tablas* para ver como debe repararla.
- **Optimizar la tabla:** Esto es útil cuando la tabla tiene overhead. Después de supresiones masivas de filas o cambios de longitud en los campos VARCHAR, quedan bits perdidos en la tabla. phpMyAdmin nos advierte en varios lugares (por ejemplo en la vista estructura), si ve que es necesario optimizar la tabla. Esta operación es una especie de defragmentación de la tabla; está disponible si el tipo de la tabla es MYISAM o Berkeley DB.
- **Limpiar la tabla:** Esto debe hacerse cuando han habido numerosos errores de conexión y el servidor MySQL bloquea las siguientes conexiones. Limpiar borrará algunos caches internos y permitirá que recomiencen las operaciones normales.
- **Defragmentar la tabla:** Inserciones y supresiones aleatorias en una tabla InnoDB fragmentan su índice. Esta tabla debería defragmentarse de forma periódica para recuperar los datos con más rapidez.



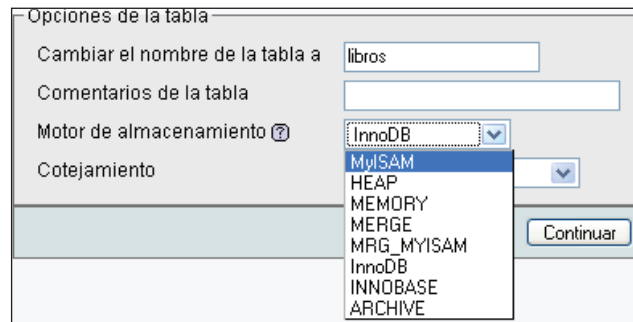
Las operaciones están basadas en las consultas MySQL disponibles: phpMyAdmin sólo llama a estas consultas.

Cambiar los atributos de las tablas

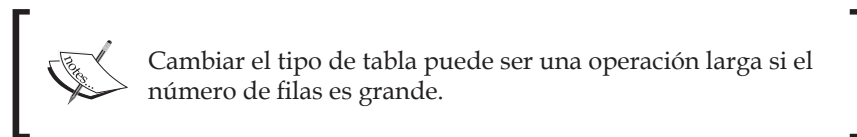
Los atributos son las distintas propiedades de una tabla. En esta sección se discuten las configuraciones de los diferentes atributos de las tablas.

Tipo de tabla

El primer atributo que podemos cambiar es el **tipo de tabla**:



Esto controla el comportamiento de la tabla: Su posición (en el disco en la memoria), la estructura del índice y si soporta transacciones y claves foráneas. La lista desplegable varía dependiendo del tipo de tabla soportada por nuestro servidor MySQL.



Comentarios de las tablas

Este diálogo nos permite escribir comentarios en la tabla. Estos comentarios se mostrarán en los lugares apropiados (por ejemplo, bajo el menú de la vista tabla y el fichero de exportación).



El valor por defecto de `$cfg['ShowTooltipAliasDB']` y `$cfg['ShowTooltipAliasTB']` (`FALSE`) produce el comportamiento que hemos visto antes. Los nombres verdaderos de la base de datos y la tabla se ven en el panel de la izquierda y en la vista base de datos para la sub-página **Estructura**. Los comentarios aparecen cuando se pone el puntero del ratón sobre el nombre de una tabla.

Si uno de estos parámetros se configura `TRUE`, los objetos correspondientes (los nombres de las bases de datos para `DB` y los nombres de tablas para `TB`) se mostrarán sólo como sus comentarios en lugar de mostrar sus nombres. Esta vez, el puntero del ratón muestra el nombre verdadero del objeto. Esto es conveniente cuando los nombres reales de las tablas no tienen significado.

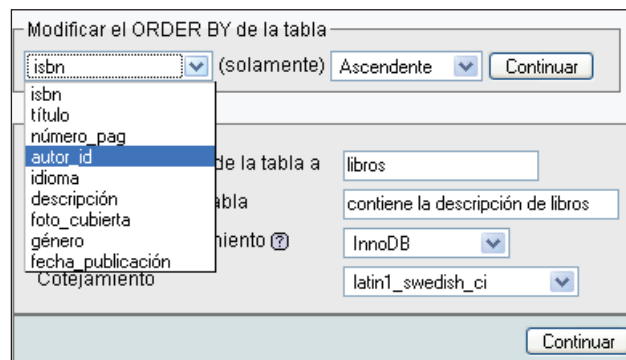
Hay otra posibilidad para `$cfg['ShowTooltipAliasTB']`: El valor `'nested'`. Esto es lo que pasa si usamos esta característica:

- El nombre verdadero de la tabla se muestra en el panel de la izquierda.
- El comentario de la tabla (por ejemplo `proyecto__`) se interpreta con el nombre del proyecto y se muestra como tal; por favor vaya a la sección *Presentación jerárquica de tablas en una base de datos* del capítulo 3.

Orden de las tablas

MySQL aplica un orden por defecto cuando **Examinamos** una tabla sin especificar un orden de clasificación. Este orden por defecto se puede cambiar con el diálogo **Alterar orden de las tablas**. Podemos elegir cualquier campo, y la tabla se reordenará *una vez* en este campo. Hemos elegido `autor_id` en el ejemplo y después de hacer clic en **Continuar**, la tabla se clasifica en este campo.

Este orden por defecto durará mientras no haya cambios en la tabla (no inserciones, supresiones o actualizaciones). Es por eso que phpMyAdmin muestra la alarma (*una vez*).

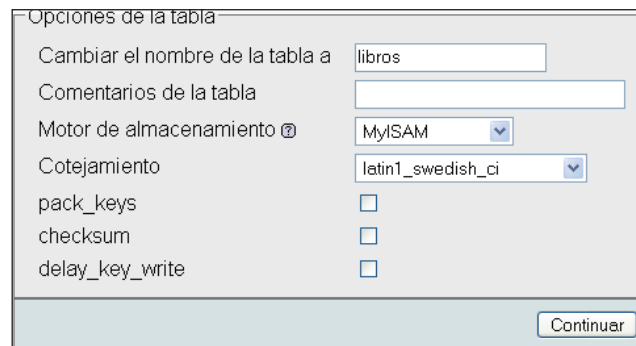


Cuando se ha hecho la clasificación en `autor_id`, los libros del autor 1 se mostrarán primero, seguidos por los libros del autor 2 y así seguirá (estamos hablando examinar por defecto la tabla sin clasificación explícita). También podemos especificar el orden de clasificación: **Ascendente** o **descendente**.

Si insertamos otra fila, que describa un libro nuevo del autor 1 y luego hacemos clic en **Examinar**, el libro no se mostrará junto con el resto de libro de este autor porque la clasificación se ha hecho antes de la inserción.

Opciones de las tablas

Otros atributos que influyen en el comportamiento de la tabla se pueden especificar con el diálogo **Opciones de tabla**:



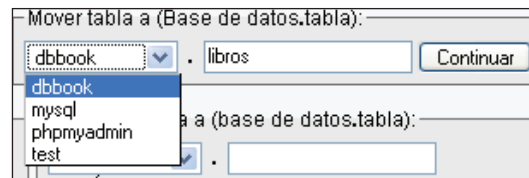
Las opciones disponibles son:

- **Organizar_claves:** Si configuramos este atributo creamos un índice pequeño; se puede leer con más rapidez pero cuesta más tiempo actualizarlo. Está disponible para los tipos de tabla `MyISAM` o `ISAM`.
- **Suma de control:** Esto hace que MySQL calcule la suma de control para cada fila. Esto hace que actualizar sea más lento pero es más fácil encontrar las tablas corruptas. Sólo está disponible para `MyISAM`.
- **retardar_escritura_clave:** Esto ordena a MySQL que no escriba las actualizaciones del índice de inmediato sino que los ponga en cola para más tarde, lo que mejora en funcionamiento. Sólo está disponible para `MyISAM`.
- **Auto_incrementar:** Esto cambia el valor auto-incrementar.

Renombrar, mover y copiar una tabla

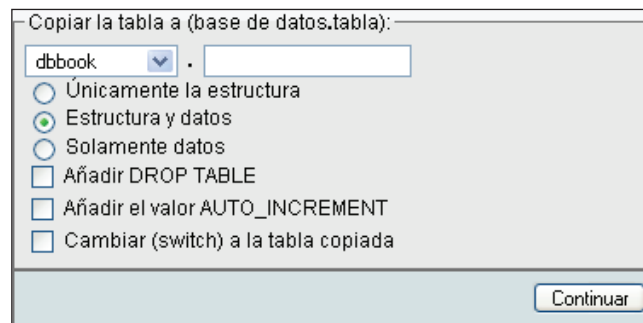
La operación **Renombrar** es la más sencilla de entender: La tabla solo cambia el nombre y se queda en la misma base de datos.

La operación **Mover** (que se ve en la siguiente pantalla) puede manipular la tabla de dos maneras: Cambiar su nombre y *también* la base de datos en la que se almacena:



Mover una tabla no está soportado directamente por MySLQ, de modo que phpMyAdmin tiene que crear la tabla en la base de datos destino, copiar los datos y para terminar eliminar la tabla fuente.

La operación **Copiar** deja la tabla original intacta y copia su estructura o datos (o ambos) en otra tabla, posiblemente en otra base de datos. Aquí, la tabla **copia_libros** será una copia exacta de la tabla fuente **libros**. Cuando hagamos la copia, nos quedaremos en la vista tabla de la tabla **libros** a no ser que hayamos seleccionado **Cambiar a la tabla copiada**.



La **Única copia de la estructura** se hace para crear una tabla de test que tiene la misma estructura.

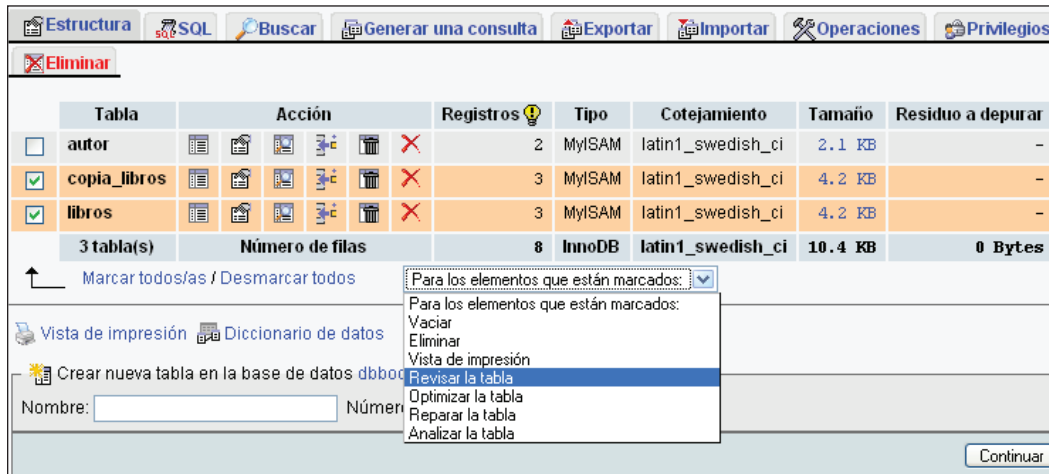
Adjuntar datos a una tabla

El diálogo **Copiar** también se puede usar para adjuntar (añadir) datos de una tabla a otra. Ambas tablas deben tener la misma estructura. Esta operación se consigue al escribir datos en la que queremos copiar los datos de la tabla abierta y eligiendo **Sólo datos**.

Como ejemplo de cuando podemos adjuntar datos es cuando los datos de los libros provienen de diferentes fuentes (varios editores), están almacenados en más de una tabla y queremos agregar los datos en un lugar.

Operaciones en múltiples tablas

En la vista base de datos, hay un recuadro cerca de cada tabla y un menú desplegable debajo de la lista de tabla. Esto nos permite elegir algunas tablas con rapidez y desempeñar una operación en todas estas tablas al mismo tiempo. Aquí seleccionamos las tablas **copia_libros** y **libros** y elegimos la operación **Verificar** para estas tablas.



También podemos seleccionar o deshacer la selección de todos los recuadros con **Marcar todos/Desmarcar todos**.

Reparar una tabla "en uso"

El método tablas múltiples es el único método (a no ser que conozcamos el tipo exacto de consulta SQL) para reparar una tabla corrupta. Esas tablas se pueden mostrar en la **bandeja en uso** de la lista de base de datos. Los usuarios que buscan ayuda en los foros de soporte de phpMyAdmin a menudo reciben este consejo de otros usuarios phpMyAdmin experimentados.

Resumen

En este capítulo hemos cubierto las operaciones que podemos desempeñar en las tablas y hemos incluido:

- Operaciones de mantenimiento de una tabla para reparar y mejorar una tabla.
- Cambiar diferentes atributos de las tablas.
- Mover las tablas, incluyendo renombrar y mover a otra base de datos.
- Operaciones en múltiples tablas.

11

Sistema relacional

Bienvenidos a la parte del libro en la que empezamos a cubrir las características avanzadas. El sistema relacional les permite a los usuarios hacer más cosas con phpMyAdmin como verá en los siguientes capítulos. Este capítulo específico explica como instalar la infraestructura de tablas enlazadas, que es un prerequisite para tener las características avanzadas y explica como definir las relaciones interiores de las tablas.

¿MySQL relacional?

Cuando los programadores de aplicaciones usan PHP y MySQL para construir interfaces Web u otras aplicaciones de manipulación de datos, normalmente establecen relaciones entre las tablas , usando las consultas SQL existentes. Por ejemplo, 'haz una factura con todos sus objetos' y 'saca todos los libros de un autor'.

En las primeras versiones de phpMyAdmin, MySQL almacenaba información sobre *que* tabla pertenecía a *que* base de datos, pero la estructura de los datos relacionales (como las tablas están relacionadas entre ellas) no se almacenaba en MySQL. Las relaciones las hacían temporalmente las aplicaciones que generan resultados con significados: *Las relaciones estaban en nuestra cabeza*.

Los programadores y los usuarios de phpMyAdmin lo consideraron un defecto de MySQL, de modo que el equipo empezó a construir una infraestructura que soportase las relaciones. La infraestructura evolucionó para soportar un conjunto creciente de características especiales. Podemos describir la infraestructura como **metadatos** (datos sobre los datos).

phpMyAdmin 2.2.0 ya tenía la característica **marcador** (que podía recordar consultas que se usaban a menudo, se describe en el capítulo 14), pero la versión 2.3.0 ya tenía un sistema de metadatos generalizado. Las versiones siguientes continuaron ampliando esta característica, el último añadido fue en la familia 2.5x con sus transformaciones basadas en MIME (descritas en el capítulo 16).

InnoDB

Apareció un nuevo tipo de tabla (**InnoDB**) durante el desarrollo de phpMyAdmin. El sub-sistema InnoDB tiene su propia página Web en <http://www.innodb.com>.

Ya que el sub-sistema InnoDB debe activarlo el administrador del sistema puede que no esté disponible en todos los servidores MySQL. Siempre que sea posible debemos usar el tipo de tabla InnoDB porque:

- Soporta la integridad referencial basada en claves foráneas, que son claves en un tabla (o referencia) foránea. En contraste, si solo usa las relaciones internas (se discutirá más tarde) phpMyAdmin no crea una verificación automática de la integridad referencial.
- Las definiciones exportadas de las tablas InnoDB contienen las relaciones definidas y por ello se pueden importar con mayor facilidad y para una mejor interoperatividad entre servidores cruzados.

La característica de claves foráneas de las tablas InnoDB; sólo sirve para las tablas InnoDB; puede reemplazar de manera efectiva la parte de la infraestructura de phpMyAdmin que abarca las relaciones. Ahora veremos como phpMyAdmin crea la interfaz para el sistema de claves foráneas de InnoDB.



Las otras partes de la infraestructura de phpMyAdmin (por ejemplo, los marcadores) no tienen equivalente en InnoDB o MySQL, por ello se tiene que hacer una instalación completa de las características de phpMyAdmin.

Infraestructura de las tablas enlazadas

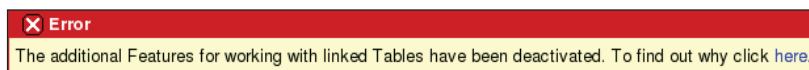
La infraestructura del sistema relacional se almacena en las tablas que siguen una estructura predeterminada. Los datos de estas tablas los genera y los mantiene phpMyAdmin, bajo la base de nuestras acciones desde la interfaz.

Posición de la infraestructura

Hay dos lugares posibles para almacenar estas tablas :

- En la base de datos del usuario porque cualquier programador Web que tenga una base de datos debería poder beneficiarse de estar características.
- En una base de datos dedicada, que llamaremos **pmadb** (base de datos phpMyAdmin). En la instalación multi-usuario (que se discute más tarde), esta base de datos deber ser accesible a varios usuarios mientras que se mantendrán los metadatos privados.

Ya que esta infraestructura no existe por defecto; y ya que los programadores de phpMyAdmin quieren promocionarla; la interfaz muestra el siguiente mensaje de error por cada base de datos cuando estamos en la vista base de datos:



Instalar la infraestructura de las tablas enlazadas

El mensaje de error previo se muestra incluso si sólo falta parte de la infraestructura. En una instalación nueva todas las partes faltan, por supuesto: Nuestra base de datos no ha oído hablar de phpMyAdmin y necesita equiparse con su infraestructura. Si seguimos el enlace **aquí** de este mensaje nos muestra la siguiente explicación:

```
Server: localhost > Database: dbbook
PMA Database ... not OK [ Documentation ]
General relation features Disabled
```

 El mensaje es el mismo, da igual la base de datos de la que se trate (aquí, **dbbook**) porque la infraestructura se comparte entre todas nuestras bases de datos y tablas (o entre todos los usuarios en una instalación multiusuario).

Como sugiere la anterior pantalla, la **base de datos PMA** no es **Correcta (OK)**. Es importante darse cuenta que el sistema relacional sólo funciona si se cumplen estas dos condiciones:

- Las definiciones correctas están presentes en `config.inc.php`.
- Las tablas correspondientes (y quizá la base de datos) se crean.

Para crear la estructura necesaria que esté acorde con nuestra versión actual de phpMyAdmin, hay un archivo comando llamado `create_tables.sql` está disponible en el subdirectorio `scripts` del directorio de instalación phpMyAdmin. Sin embargo, no deberíamos ejecutarlo a ciegas antes de entender las posibles elecciones: La instalación multi-usuario o la instalación de un usuario.

Instalación multi-usuario

En esta configuración, tenemos una base de datos distintiva (**pmadb**) para almacenar los metadatos y nuestro control de usuario tiene permisos específicos para esta base de datos. Cada usuario escribirá su nombre de acceso y su contraseña, que se usará para acceder a su base de datos. Sin embargo, siempre que se tenga que acceder a **pmadb**, phpMyAdmin usará los privilegios del control de usuario.

Primero tenemos que asegurarnos que el control de usuario pma se ha creado, tal como se explicaba en el capítulo 2 y que su definición en `config.inc.php` es apropiada:

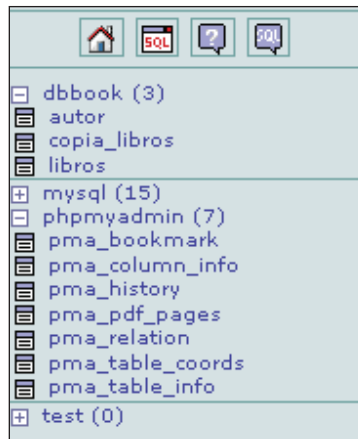
```
$cfg['Servers'][$i]['controluser'] = 'pma';  
$cfg['Servers'][$i]['controlpass'] = 'bingo';
```

Luego usaremos el archivo `scripts/create_tables.sql` para crear la base de datos phpmyadmin, le daremos los permisos necesarios al usuario pma y poblaremos la base de datos con las tablas necesarias. Antes de usar este programa, por favor eche un vistazo al directorio `scripts`. Puede que haya otros programas disponibles para diferentes versiones de MySQL; por ejemplo, phpMyAdmin 2.6.0 tiene `scripts/create_tables_mysql_4_1_2+.sql`, que deberían usarse en lugar de `create_tables.sql` para la versión 4.1.2 de MySQL y superiores.



Tenga cuidado ya que este programa borrará la base de datos phpmyadmin si existe y destruirá todos los metadatos sobre las relaciones.

Un método posible para ejecutar este programa es usar la técnica descrita en el capítulo 8 (*Importar la estructura y los datos*), usar la sub-página importa y el selector de archivos. Para que esto funcione, tenemos que tener el programa `create_tables.sql` en algún lugar de nuestra área de trabajo. Cuando lo hayamos creado, el panel de la derecha tendrá este aspecto:



Ahora tenemos que ajustar todos los parámetros de las características relacionales en `config.inc.php`. Aquí usaremos los valores por defecto mencionados en los comentarios que están dentro del archivo; estos nombres de base de datos y nombres de tablas son los mismos que acabamos de crear:

```
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
$cfg['Servers'][$i]['bookmarktable'] = 'pma_bookmark';
$cfg['Servers'][$i]['relation'] = 'pma_relation';
$cfg['Servers'][$i]['table_info'] = 'pma_table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma_table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma_pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma_column_info';
$cfg['Servers'][$i]['history'] = 'pma_history';
```



Como los nombres de las tablas son sensibles a las mayúsculas y las minúsculas, debemos usar los mismos nombres que ha creado el programa de instalación. Somos libres para cambiar los nombres de las tablas (vea la parte derecha de las directivas de configuración que están en la lista) siempre que los cambiemos en la base de datos del mismo modo.

Cada tabla tiene una función específica:

- **Pmadb:** Define en que base de datos están todas las tablas.
- **Bookmarktable:** Es donde se guardarán los marcadores (vaya al capítulo 14).
- **Relation:** Define las relaciones entre tablas, se usa en muchas de las características de phpMyAdmin.

- `Table_info`: Contiene el campo de presentación (se explica más tarde en este capítulo).
- `Table_coords` y `pdf_pages`: Contiene los metadatos necesarios para dibujar el esquema de relaciones en formato PDF (vaya al capítulo 15).
- `Column_info`: Se usar para escribir los comentarios de las columnas y las transformaciones basadas en MIME (vaya al capítulo 16).
- `History`: Contiene información del historial de las consultas SQL (se explica en el capítulo 12).

Entre cada versión de phpMyAdmin, la infraestructura se puede realzar (los cambios se explican en `Documentation.html`). Es por ello que phpMyAdmin tiene varias verificaciones para asegurarse de la estructura de las tablas. Si sabemos que estamos usando la estructura más nueva, `$cfg['Servers'][$i]['verbose_check']` se puede configurar `FALSE` para evitar las verificaciones, de modo que la velocidad de phpMyAdmin aumente un poco.

Ahora la instalación está completa; comprobaremos las características en las siguientes secciones y capítulos. Podemos hacer una verificación rápida si volvemos a la página de **Inicio**: El mensaje de alarma debería haber desaparecido.

Instalación de un usuario

Incluso si el administrador de sistemas sólo nos ha dado permiso para una base de datos, podemos usar todas las características relacionales de phpMyAdmin.

En esta configuración, usaremos nuestra base de datos normal (asumamos que es **dbbook**) para almacenar las tablas de metadatos y definiremos nuestro nombre de conexión (`marc`) y el control de usuario en `config.inc.php`:

```
$cfg['Servers'][$i]['controluser']      = 'marc';  
$cfg['Servers'][$i]['controlpass']     = 'bingo';
```

El siguiente paso es modificar la copia local del archivo `scripts/create_tables.sql` para poblar nuestra base de datos con las tablas necesarias. Tendrán el prefijo `pma-` para que las podamos reconocer con facilidad. Por favor lea también la nota en la sección *Instalación multi-usuario* sobre el resto de programas disponibles en el directorio `scripts`.



Tenga cuidado ya que este programa borrará las tablas especiales si existen y destruirá todos los metadatos sobre las relaciones.

La primera modificación que hay que hacer es suprimir las líneas de comentarios (que empiezan con dos guiones (-)) para asegurarse que no interfieren (en algunas versiones de phpMyAdmin, la rutina de importación no los entiende). Entonces borramos las siguientes líneas:

```
DROP DATABASE 'phpmyadmin';
CREATE DATABASE 'phpmyadmin';

USE phpmyadmin;

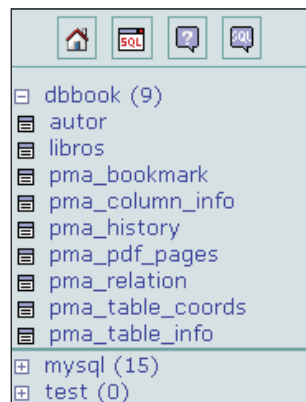
GRANT SELECT, INSERT, DELETE, UPDATE ON 'phpmyadmin'.* TO
    'pma'@localhost;
```

Esto se hace porque no usaremos la base de datos phpmyadmin o el usuario de control pma.

Ahora estamos preparados para ejecutar el programa. Hay dos maneras de hacerlo:

- Como ya tenemos el programa en nuestro editor, podemos copiar las líneas y pegarlas en la caja de consulta de la sub-página SQL.
- Otra manera es usar la técnica que se muestra en el capítulo 8 (*Importar la estructura y los datos*), con la sub-página importar y el selector de archivos. Seleccionamos el programa `create_tables.sql` que acabamos de modificar.

Cuando hayamos terminado la creación, el panel de la izquierda mostrará las tablas especiales pma_ junto con nuestras tablas normales.



El último paso es ajustar todos los parámetros `config.inc.php` que relacionan las características relacionales. Usamos los valores por defecto mencionados dentro del archivo, a excepción del parámetro `pmadb`.

```
$cfg['Servers'][$i]['pmadb'] = 'dbbook';
```


Vista relación

Después de la instalación de la infraestructura de las tablas enlazadas, hay más opciones disponibles en la vista base de datos y la vista tabla. Ahora examinaremos el nuevo enlace de la vista tabla: **Vista relación**. Esta vista se usa para:

- Definir las relaciones de la tabla abierta con otras tablas.
- Elegir el **campo de presentación**.
- Definir los comentarios específicos de columna.

Ya que nuestro objetivo es crear una relación entre la tabla **libros** (que tiene el ID del autor) y la tabla **autores** (que describe cada autor por su ID), empezamos en la vista tabla de la tabla libros y hacemos clic en la **vista relación**.

Relaciones internas de phpMyAdmin

Ya que la tabla libros está en formato MyISAM, vemos la siguiente pantalla (de otro modo, la presentación sería diferente, como se explicará en la sección *Relaciones InnoDB* más tarde):

Enlaces a	
Relaciones internas	
isbn	--
titulo	--
número_pag	--
autor_id	--
idioma	--
descripción	--
foto_cubierta	--
género	--
fecha_publicación	--

Elegir el campo a mostrar: ...

Grabar

Esta pantalla nos permite crear **relaciones internas** (que se almacenan en la tabla **pma_relation**), porque MySQL no tiene una noción relacional para las tablas MyISAM. El carácter guión doble (--) indica que *no* hay relaciones (enlaces) a una tabla foránea.

Definir la relación

Podemos relacionar cada campo de la tabla **libros** a un campo de otra tabla (o en la misma tabla), porque a veces las relaciones de auto referencia son necesarias. La interfaz en encuentra las claves únicas y no únicas en todas las tablas de la misma base de datos y las presenta en un menú desplegable. La elección apropiada aquí es seleccionar el campo **autor_id** correspondiente con el campo **autor_id** de la tabla **autor**. Esto también se llama *definir la clave foránea*.

Enlaces a

	Relaciones internas
isbn	--
titulo	--
número_pag	--
autor_id	--
idioma	--
descripción	autor->autor_id
foto_cubierta	libros->isbn
género	libros->titulo
fecha_publicación	libros->autor_id
	libros->idioma
	libros->descripción
	pma_bookmark->id
	pma_column_info->id
	pma_history->id
	pma_pdf_pages->page_nr
	pma_table_coords->db_name
	pma_table_coords->table_name
	pma_table_coords->pdf_page_number
	pma_table_info->db_name
	pma_table_info->table_name

Elegir el campo a mostrar

Grabar

Luego hacemos clic en continuar y la definición se guarda en la infraestructura de phpMyAdmin. Para suprimir la relación, solo tenemos que volver a la pantalla, elegir la selección de guión doble y hacer clic en **Continuar**.

Definir el campo presentación

La clave primaria de nuestra tabla **autor** es **autor_id**, que es un número único que nos hemos inventado para la clave. Otro campo en nuestra tabla representa a los autores: El nombre. Sería interesante ver el nombre del autor como una descripción informativa de cada fila de la tabla **libros**. Este es el propósito del **campo presentación**. Normalmente deberíamos definir el campo presentación para cada tabla que participa en la relación como una tabla foránea.

Ahora veremos como esta información se muestra en la sección *Beneficios de las relaciones definidas*. Ahora vamos a la **vista relación** de la tabla **autor** (que en este caso es la tabla foránea) y especificamos el campo de presentación. Elegimos **nombre_autor** como el campo de presentación:

Servidor: localhost:3306 ▶ Base de datos: dbbook ▶ Tabla: autor

Examinar Estructura SQL Buscar Insertar Exportar

Enlaces a

Relaciones internas	
autor_id	..
nombre_autor	..
teléfono	..

Elegir el campo a mostrar: ...

- ...
- autor_id
- nombre_autor
- teléfono

Grabar



phpMyAdmin ofrece la definición de un campo de presentación por tabla, este campo se usa en todas las relaciones en que esta tabla se usa como tabla foránea.

La definición de esta relación ya está hecha. Por favor tenga en cuenta que aunque no hemos unido ninguno de los campos de la tabla **autor** con otra tabla, esto se puede hacer. Por ejemplo, podríamos tener un código del país en esta tabla y podríamos crear una relación con el código de país de una tabla país.

Discutiremos los beneficios de haber definido esta relación más adelante, pero primero, tenemos que ver que pasa si nuestras tablas son tipo InnoDB.

Relaciones InnoDB

El tipo de tabla InnoDB nos ofrece un sistema de clave foránea. Para probarlo, primero cambiamos nuestras tablas **libros** y **autor** al tipo InnoDB. Podemos hacerlo en la vista tabla desde la sub-página **Operaciones**. Empezamos a hacerlo con la tabla **autor**:

Motor de almacenamiento ⓘ

Cotejamiento

pack_keys

checksum

delay_key_write

MyISAM

MyISAM

HEAP

MEMORY

MERGE

MRG_MYISAM

InnoDB

INNODB

ARCHIVE

Podemos tener un problema cuando cambiemos el tipo de tabla de la tabla **libros** a InnoDB; tenemos un índice FULLTEXT en esta tabla y algunas versiones de MySQL no lo soportan para el tipo de tabla InnoDB. Primero tenemos que quitar el índice FULLTEXT si recibimos el siguiente mensaje de error:

Servidor: localhost:3306 ▶ Base de datos: dbbook ▶ Tabla: libros

Error

consulta SQL: 

 ALTER TABLE `libros` ENGINE = INNODB

MySQL ha dicho: 

#1214 - The used table type doesn't support FULLTEXT indexes

Asumamos que lo hemos hecho y que hemos cambiado la tabla **libros** a InnoDB.

El sistema de clave foránea de InnoDB mantiene la integridad entre las tablas relacionadas y por eso no podemos añadir un ID no existente a la tabla **libros**. Además, las acciones se pueden programar cuando se hacen operaciones DELETE o UPDATE en la tabla maestra (en nuestro caso, **libros**).

Abrir la tabla **libros** y entrar en la **vista relación** nos muestra una página diferente:

Servidor: localhost:3306 ▶ Base de datos: dbbook ▶ Tabla: libros

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Enlaces a

	Relaciones Internas 	InnoDB
isbn	--	-- ON DELETE -- ON UPDATE --
título	--	-- ON DELETE -- ON UPDATE --
número_pag	--	-- ON DELETE -- ON UPDATE --
autor_id	autor->autor_id	-- ON DELETE -- ON UPDATE --
Idioma	--	-- ON DELETE -- ON UPDATE --
descripción	--	-- ON DELETE -- ON UPDATE --
foto_cubierta	--	-- ON DELETE -- ON UPDATE --
género	--	-- ON DELETE -- ON UPDATE --
fecha_publicación	--	-- ON DELETE -- ON UPDATE --

Elegir el campo a mostrar: autor_id

Grabar

Esta página nos dice que:

- Tenemos relaciones internas definidas en **autor_id** con la tabla **autor**.
- Aún no tenemos ninguna relación InnoDB definida.
- Podemos suprimir la relación interna cuando hayamos definido la misma relación en InnoDB. De hecho, phpMyAdmin nos recuerda que la relación interna no es necesaria cuando ya existe en InnoDB y que por eso sería mejor suprimirla.
- Las opciones ON DELETE y ON UPDATE están disponibles para las relaciones de InnoDB.

Quizás la página también nos indique que nuestra versión de MySQL no esté actualizada (tiene que ser 4.0.13 o superior). Uno de las implicaciones de esto es: No podremos suprimir la relación definida en InnoDB. Por eso phpMyAdmin nos hace esta amable (¡y crucial!) indicación.

En las elecciones posibles de la clave relacionada, vemos las claves definidas en otras tablas InnoDB.

Intentemos suprimir la relación interna del campo **autor_id** y añadir (a pesar que no funciona) una relación tipo InnoDB para el campo **autor_id**:

autor_id	..	autor->autor_id	ON DELETE ..	ON UPDATE ..
-----------------	----	-----------------	--------------	--------------

Hacer clic en **Continuar** genera un mensaje de error: ¡No hay un índice definido! (**autor_id**). Esto pasa porque las definiciones de claves foráneas de InnoDB solo se pueden hacer si ambos campos se han definido como índices (hay otras reglas que están explicadas en el manual de MySQL).

Así pues, volvemos a la página **Estructura** de la tabla **libros** y añadimos un índice ordinario (no único) a **autor_id**:

Índices: ?				
Nombre de la clave	Tipo	Cardinalidad	Acción	Campo
PRIMARY	PRIMARY	3	 	isbn
por_título	INDEX	3	 	título 30
autor_id	INDEX	3	 	autor_id
Crear un índice en 1 columna(s) Continuar				

En la **vista relación**, podemos volver a insertar añadir la relación que queríamos; ¡esta vez funciona!

También podemos configurar algunas acciones con las opciones ON DELETE y ON UPDATE. Por ejemplo, ON DELETE CASCADE hará que MySQL borre automáticamente todas las filas en las tablas (foráneas) relacionadas cuando la fila correspondiente se borre en la tabla maestra. Esto puede ser útil, por ejemplo, cuando la tabla maestra es 'facturas' y la tabla foránea 'objetos-factura'.



Si no lo hemos hecho aún, deberíamos definir el 'campo de presentación' para la tabla **autor** como se explica en la sección *Relaciones internas de phpMyAdmin*.

En la versión actual de phpMyAdmin (2.8.0.4) no podemos ver las tablas desde una base de datos diferente para definir una relación en ellas.

Tablas InnoDB sin la infraestructura tablas enlazadas

Desde la versión 2.6.0 de phpMyAdmin vemos el enlace **vista relación** en la página **Estructura** de una tabla InnoDB aunque la infraestructura no esté instalada. Esto nos lleva a una pantalla en la que podemos definir las claves foráneas: Aquí lo hacemos para la tabla `libros`.

Por favor tenga en cuenta que si elegimos esta manera, el 'campo presentación' para la tabla enlazada (aquí **autor**) no se puede definir; pertenece a la infraestructura de phpMyAdmin; de forma que perderíamos unos de los beneficios (ver la descripción asociada con la clave foránea).

Servidor: localhost:3306 ▶ Base de datos: dbbook ▶ Tabla: libros

Examinar Estructura SQL Buscar Insertar Exportar Importar

Operaciones Vaciar Eliminar

Enlaces a

InnoDB	
isbn	ON DELETE ON UPDATE
titulo	ON DELETE ON UPDATE
número_pag	¡No se ha definido el índice!
autor_id	ON DELETE ON UPDATE
idioma	el índice!
descripción	el índice!
foto_cubierta	el índice!
género	¡No se ha definido el índice!
fecha_publicación	¡No se ha definido el índice!

Grabar

Beneficios de las relaciones definidas

En esta sección veremos los beneficios que podemos verificar en estos momentos; los otros beneficios se mostrarán en el capítulo 13 (*Generador de consultas multi-tabla*) y el capítulo 15 (*Documentación del sistema*). Otros beneficios de la infraestructura de tablas enlazadas aparecerán en el capítulo 14 (*Marcadores*) y el capítulo 16 (*Transformaciones basadas en MIME*).

Estos beneficios están disponibles para las relaciones internas e InnoDB.

Información de la clave foránea

Examinemos la tabla **libros**. Vemos que la clave relacionada (**autor_id**) ahora es un enlace.

←T→	isbn	título	número_pag	autor_id
☐	1-234567-22-0	Recuerdos de futuro	200	2
☐	1-234567-89-0	Cien años de cine (volumen 1)	600	1
☐	1-234567-90-0	Cien años de cine (volumen 2)	600	1

Si movemos el puntero del ratón sobre **autor_id** nos muestra el nombre del autor (tal y como se había definido en el campo de presentación de la tabla **autor**):

←T→	isbn	título	número_pag	autor_id	idioma
☐	1-234567-22-0	Recuerdos de futuro	200	2	en
☐	1-234567-89-0	Cien años de cine (volumen 1)	600	1	en
☐	1-234567-90-0	Cien años de cine (volumen 2)	600	1	en

Si hacemos clic en **autor_id** aparecerá la tabla relevante, autores, de este autor específico.

←T→	autor_id	nombre_autor	teléfono
☐	1	John Smith	111-1111

Lista desplegable de las claves foráneas

En el modo **Insertar** (o en el modo **Editar**) ahora vemos una lista desplegable de las posibles claves por cada campo que tiene una relación definida. La lista contiene las claves y la descripción (campo de presentación) en ambos órdenes: La clave para mostrar el campo y el campo de presentación de la clave. Esto nos permite usar el teclado y escribir la primera letra de cualquier clave o campo de presentación:

autor_id	int(11)		
idioma	char(2)		
descripción	text		

John Smith - 1
 Maria Sunshine - 2
 1 - John Smith
 2 - Maria Sunshine

[ Sólo la clave (en este caso **1**) se almacenará en la tabla **libros**. El campo de presentación está sólo para asistirnos.]

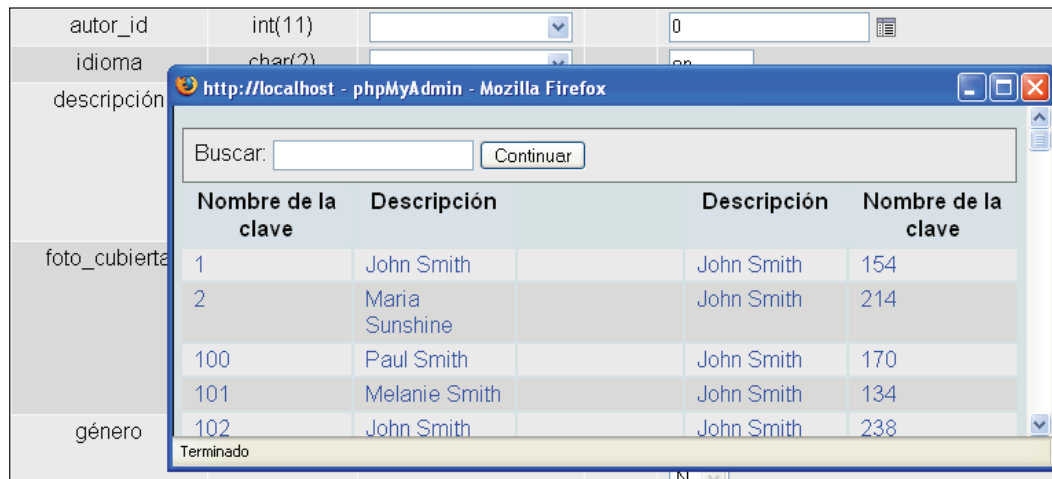
Esta lista desplegable aparecerá si hay un máximo de 200 filas en la tabla foránea. Para tablas más grandes aparece una ventana emergente: La ventana de la tabla foránea que se puede examinar.

Ventana de la tabla foránea que se puede examinar

No podemos usar nuestras tablas **libros** y **autor** actuales para ilustrar este mecanismo; tienen muy pocas entradas. Usaremos una tabla llamada **orders** con una columna **software_id** relacionada con el ID de la columna de la tabla **software**. En el modo insertar de la tabla **order**, vemos un pequeño icono en forma de tabla para **software_id**, como se ve en la muestra de pantalla que sigue:

autor_id	int(11)			
idioma	char(2)		en	
descripción	text			Mostrar los valores extranjeros

Este icono abre otra ventana que presenta los valores de la tabla **software** y un selector de página. En la izquierda, los valores se clasifican con el valor de la clave (aquí la columna **id**) y en la derecha, se clasifican por su descripción:



Si elegimos uno de los valores; haciendo clic en un valor de clave o en la descripción; se cierra esta ventana y el valor aparece en la columna **software_id**.

Verificación de la integridad referencial

Hemos discutido la sub-página **Operaciones** y su sección de **Mantenimiento de tabla** en el capítulo 10. Ahora que hemos definido una relación para la tabla **autor** (una tabla no InnoDB) aparece una nueva elección en la tabla **libros**: **Verificar la integridad referencial**:

Mantenimiento de la tabla

- Revisar la tabla
- Analizar la tabla
- Reparar la tabla
- Optimizar la tabla
- Vaciar el caché de la tabla ("FLUSH")

Comprobar la integridad referencial:

- [autor_id -> autor.autor_id](#)

Un enlace aparece en cada relación definida (aquí **autor_id -> autor.autor_id**), **hacer clic empieza una verificación**. La presencia de la clave correspondiente a la tabla foránea se verifica por cada fila y se hace un informe de errores. Si la página de resultados tiene un informe de resultados con cero filas; ¡buenas noticias!

Esta operación existe para las tablas no InnoDB porque ni MySQL ni phpMyAdmin no refuerzan la integridad referencial. Es posible, por ejemplo, importar datos en la tabla **libros** con valores inválidos para **autor_id**.

Actualización automática de los metadatos

phpMyAdmin sincroniza los metadatos de las relaciones internas con cada cambio que se produce en las tablas. Por ejemplo, renombrar una columna que forma parte de una relación haría que phpMyAdmin la renombrase en los metadatos de las relaciones. Lo mismo ocurre cuando se vacía una columna o una tabla.

Comentar columnas¹

La estructura de MySQL soportar añadir comentarios a una tabla pero no a una columna. En la **vista relación**, esta característica es posible gracias a los metadatos de phpMyAdmin.

Comentarios

número de libro

aproximado

ver tabla autor

Estos comentarios aparecen en varios lugares; por ejemplo en el archivo de exportación (ver capítulo 7), en el esquema relacional PDF (ver capítulo 15) y en el modo **Examinar**:

→T←	ISBN número de libro	Título	número_pág aproximado	autor_id ver tabla autor
 Editar  Borrar	1-234567-22-0	Recuerdos futuros	200	2

1. Las versiones superiores de MySQL soportan los comentarios de las columnas, por eso en las versiones superiores de phpMyAdmin, los comentarios de las columnas están en el enlace Editar de la vista base de datos o cuando se crea una tabla nueva.

Resumen

En este capítulo:

- Hemos cubierto la instalación de la infraestructura necesaria para mantener metadatos especiales (datos sobre las tablas).
- Hemos aprendido a definir relaciones entre tablas, tanto para las tablas InnoDB como para las tablas no InnoDB.
- Hemos examinado el comportamiento modificado de phpMyAdmin en el que hay relaciones presentes: Elegir claves foráneas, obtener información de la tabla foránea y comentar las columnas.

12

Escribir comandos SQL

Este capítulo explica como podemos escribir nuestros propios comandos (consultas) SQL en phpMyAdmin y como podemos mantener un historial de esas consultas.

Recuadro de consulta SQL

phpMyAdmin nos permite hacer numerosas operaciones de bases de datos a través de su interfaz gráfica, pero a veces tenemos que basarnos en las entradas de las consultas SQL para conseguir operaciones complejas. Aquí hay ejemplos de consultas complejas:

```
select department, avg(salary) from employees group by department
having years_experience >10;

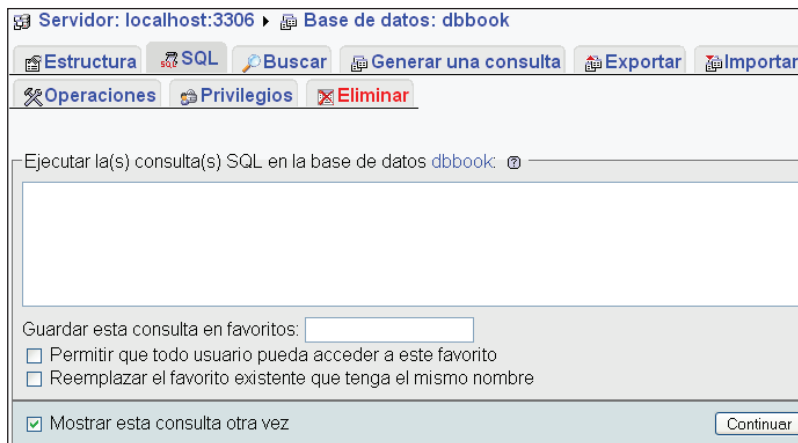
select from_days(to_days(curdate()) +30);
```

El recuadro de consulta está disponible en diferentes lugares en phpMyAdmin.

Vista base de datos

Encontramos el primer recuadro de consulta cuando vamos al menú SQL que está disponible en la vista base de datos. El recuadro es sencillo: Escribimos una sentencia válida MySQL (con suerte) y hacemos clic en **Continuar**. En el capítulo 8 ya hemos mostrado como importar archivos de consulta desde este panel.

Para que aparezca una consulta por defecto en este recuadro, podemos configurar la directiva `$cfg['DefaultQueryDatabase']` que está vacía por defecto. Podemos escribir una consulta como `SHOW TABLES FROM %d` en esta directiva. El parámetro `%d` de esta consulta se reemplazará por el nombre de la base de datos abierta, lo que resulta en `SHOW TABLES FROM 'dbbook'` en el recuadro de consulta.



Servidor: localhost:3306 ▶ Base de datos: dbbook

Estructura SQL Buscar Generar una consulta Exportar Importar

Operaciones Privilegios Eliminar

Ejecutar la(s) consulta(s) SQL en la base de datos dbbook ?

Guardar esta consulta en favoritos:

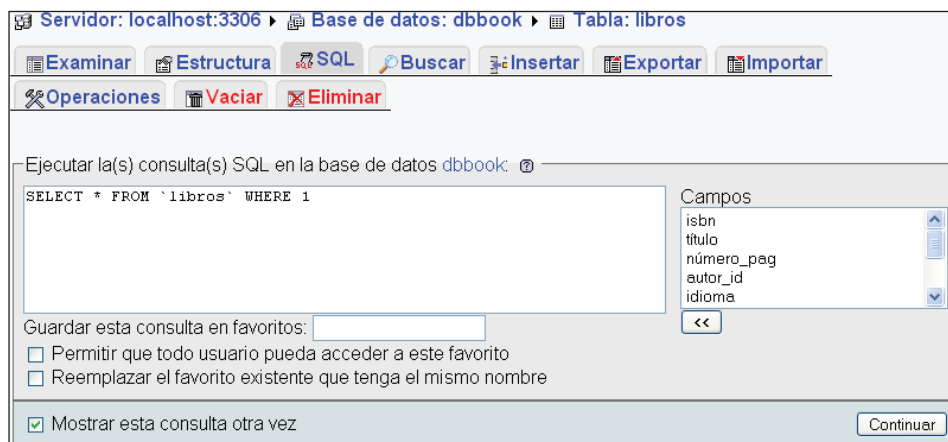
☐ Permitir que todo usuario pueda acceder a este favorito

☐ Reemplazar el favorito existente que tenga el mismo nombre

☒ Mostrar esta consulta otra vez Continuar

Vista tabla

Hay un recuadro disponible en la vista tabla algo diferente del anterior.



Servidor: localhost:3306 ▶ Base de datos: dbbook ▶ Tabla: libros

Examinar Estructura SQL Buscar Insertar Exportar Importar

Operaciones Vaciar Eliminar

Ejecutar la(s) consulta(s) SQL en la base de datos dbbook ?

`SELECT * FROM `libros` WHERE 1`

Campos

- isbn
- título
- número_pag
- autor_id
- idioma

Guarda esta consulta en favoritos:

☐ Permitir que todo usuario pueda acceder a este favorito

☐ Reemplazar el favorito existente que tenga el mismo nombre

☒ Mostrar esta consulta otra vez Continuar

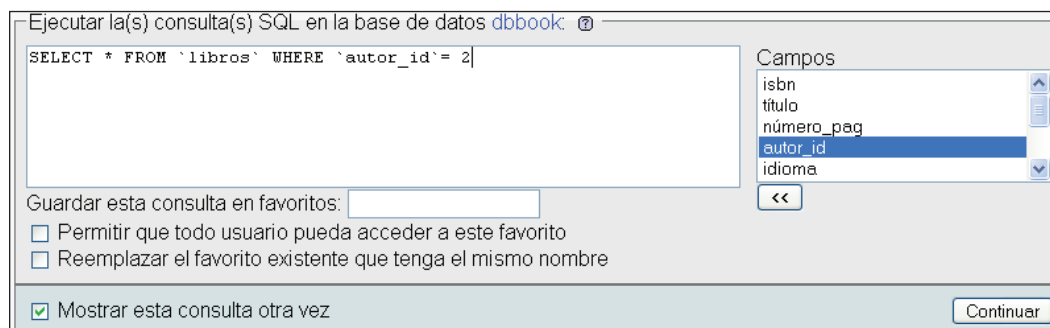
Se puede acceder a él desde el menú **SQL** o desde el menú **Estructura**. El recuadro de consulta se ofrece en dos lugares basado en la información que el equipo de programación de phpMyAdmin ha recibido: La muestra de pantalla es el resultado de acceder al recuadro de consulta desde el menú **SQL**. Algunos usuarios prefieren tenerlo en la misma página que la estructura completa; otros prefieren una página que no muestre la estructura porque sus tablas tienen muchas columnas y su presentación es muy lenta, en especial con conexiones a Internet lentas. La parte inferior tiene elecciones relacionadas con el marcador (se explica en el capítulo 14). También hay un selector de **Campos** y un botón **Insertar** a la derecha. El recuadro ya tiene una consulta por defecto.

Esta consulta (`SELECT *FROM 'libros' WHERE 1`) se genera en la directiva de configuración `$cfg['DefaultQueryTable']`, que contiene `SELECT *FROM %t WHERE 1`. Aquí `%t` se reemplaza por el nombre de tabla abierta. Otro marcador de posición en `$cfg['DefaultQueryTable']` es `%f`, que sería reemplazado por la lista completa de campos de esta tabla, de modo que se produciría la siguiente consulta `SELECT 'isbn','título','número_pag','autor_id','idioma','descripción','foto_cubierta','género' FROM 'libros' WHERE 1`.

WHERE 1 es una condición que siempre es cierta de modo que se puede ejecutar la consulta tal cual. Podemos cambiar 1 por la condición exacta que queremos escribir o escribir una consulta totalmente diferente.

Selector de campos

El selector de Campos es una manera de agilizar la escritura de una consulta. Si elegimos un campo haciendo clic en las flechas <<, el nombre de este campo se copia en la posición actual del cursor del recuadro de consulta. Aquí hemos seleccionado el campo **autor_id**, suprimimos el dígito 1 y hacemos clic en <<. Entonces añadimos la condición `= 2`.



La opción **Mostrar esta consulta otra vez** (que está marcada por defecto) quiere decir que la consulta se quedará en el recuadro después de su ejecución si todavía estamos en la misma página. Esto es mejor para una consulta como `UPDATE` o `DELETE`, que afecta a la tabla pero que no produce una página separada de resultados.

Hacer clic en el recuadro de consulta

El valor por defecto de la directiva de configuración `$cfg['TextareaAutoSelect']` es `TRUE`. Es por eso que el primer clic en este recuadro selecciona todos sus contenidos (es una manera de copiar con rapidez los contenidos en otro sitio o suprimirlos del recuadro).

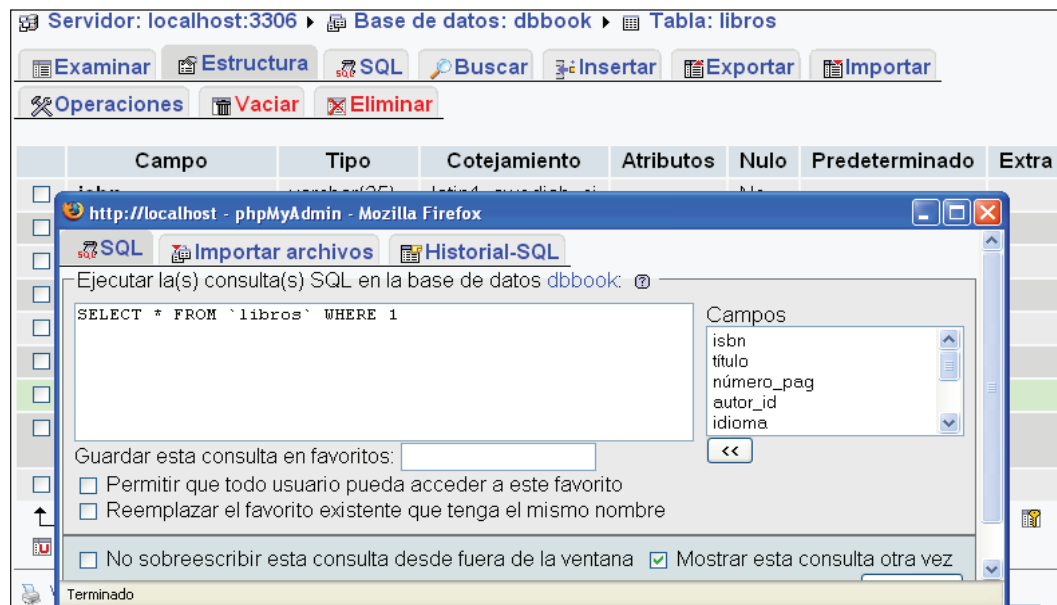
El siguiente clic pone el cursor en la posición de clic. Si la directiva está configurada FALSE, el primer clic no selecciona todos los contenidos de esta zona de texto.

Ventana de consulta

En el capítulo 3 hemos discutido el propósito de esta ventana y el procedimiento para cambiar algunos parámetros (como las dimensiones). Esta ventana se puede abrir con facilidad desde el panel de la izquierda usando el icono **SQL** o el enlace **Ventana de consulta**, es muy conveniente para escribir una consulta y verificarla:



La siguiente pantalla muestra la ventana de consulta que aparece sobre el panel de la derecha:



Contiene el mismo selector de **Campo** y el botón << que se usa en el contexto de la vista tabla.

Esta ventana de consulta emergente sólo aparece si `$cfg['QueryFrameJS']` está configurado `TRUE`; tenemos que usar un navegador con JavaScript activado. Si está configurado `FALSE`, el enlace **Ventana de consulta** sólo entrará en la página normal SQL con el recuadro de consulta.

Opciones de la ventana de consulta

La pestaña **SQL** es la pestaña activa por defecto en esta ventana. Esto viene de la directiva de configuración `$cfg['QueryWindowDefTab']` que tiene **SQL** por defecto. Si queremos que otra pestaña sea la pestaña activa por defecto podemos cambiar `sql` por `files` o `history`. Otro valor, `full`, muestra los contenidos de las tres pestañas a la vez.

En la ventana de consulta, vemos un recuadro para la elección **No sobrescribir esta consulta desde una ventana externa**. Normalmente, no está marcado y los cambios que hacemos cuando navegamos y hacemos consultas se reflejan en la ventana de consulta (esto se llama **sincronización**). Por ejemplo, si elegimos una base de datos diferente o una tabla del panel de la derecha o de la izquierda, la ventana de consulta se actualiza conforme a esas elecciones. Pero si empezamos a escribir una consulta directamente en esta consulta, el recuadro se marcará para proteger los contenidos y quitar la sincronización. De este modo, la consulta escrita se cerrará y se protegerá.

Historial SQL basado en JavaScript

Esta característica recoge todas las consultas SQL *con éxito* que ejecutamos y modifica la ventana de consulta para hacerlas disponibles. Si cerramos la ventana se perderán. Este tipo de historial por defecto es temporal: `$cfg['QueryHistoryDB']` — está configurado `FALSE` por defecto.

El historial basado en JavaScript funciona con navegadores Opera, Mozilla e Internet Explorer.

Historial SQL basado en la base de datos (permanente)

Ya que hemos instalado la infraestructura de las tablas enlazadas (ver capítulo 11), hay disponible un mecanismo historial más poderoso que se pone en funcionamiento cuando configuramos `$cfg['QueryHistoryDB']` en `TRUE`.

Después de haber escrito algunas consultas en el recuadro consulta, se construye un historial:

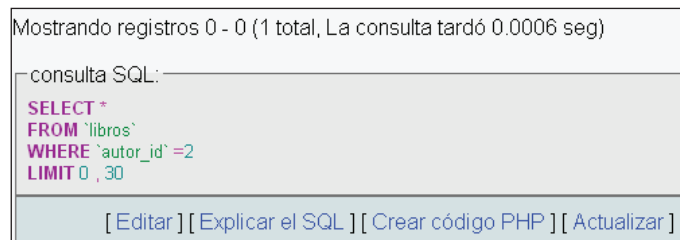


Vemos las últimas consultas con éxito (en orden invertido) y la base de datos en la que se han hecho. Sólo los tipos del recuadro de consulta se mantienen en el historial, no las consultas generadas por phpMyAdmin (por ejemplo al hacer clic en **Examinar**). Se puede hacer clic en ellas para ejecutarlas de inmediato y hay un botón **Editar** disponible para mostrar una consulta registrada en el recuadro de consulta para editarla.

El número de consultas que se guardarán lo controla `$cfg['QueryHistoryMax']`, que está configurado a 25 por defecto. Este límite no se mantiene por razones de funcionamiento sino como un límite práctico para obtener una vista más limpia. Las consultas extra se eliminan en el momento de conexión gracias a un proceso llamado tradicionalmente **recogida de basura**. Las consultas se almacenan en la tabla que se ha configurado en `$cfg['Servers'][$i]['history']`.

Editar la consulta y la ventana de consulta

En la página de resultados de una consulta con éxito, aparece una cabecera con la consulta que se ha ejecutado:



Si hacemos clic en **Editar** se abre la pestaña **SQL** de la ventana de consulta, con la consulta preparada para ser modificada.

Consultas con múltiples sentencias

En la programación PHP/MySQL, sólo podemos mandar una consulta al mismo tiempo usando la función de llamada `mysql_query()`. phpMyAdmin permite mandar numerosas consultas en una transmisión haciendo uso del punto y coma como separador. Imaginemos que escribimos esto en el recuadro de consulta:

```
Insert into autor values (100, 'Paul Smith', '111-2222');
Insert into autor values (101, 'Melanie Smith', '222-3333');
Update autor set teléfono='444-5555' where nombre_autor like '%Smith%';
```

Entonces recibiremos la siguiente pantalla de resultados:

Su consulta se ejecutó con éxito

consulta SQL: `insert into autor values (100,'Paul Smith','111-2222');# Filas afectadas: 1 insert into autor values (101,'Melanie Smith','222-3333');# Filas afectadas: 1 update autor set teléfono='444-5555' where nombre_autor like '%Smith%';# Filas afectadas: 3`

Vemos el número de filas afectadas a través de los comentarios porque `$cfg['VerboseMultiSubmit']` está configurado `TRUE`.

Mandemos la misma lista de consultas otra vez y veamos los resultados:

Error

consulta SQL:

```
INSERT INTO autor
VALUES (100, 'Paul Smith', '111-2222');
```

MySQL ha dicho: ⓘ

#1062 - Duplicate entry '100' for key 1

Es normal recibir un error **Entrada duplicada**: El valor 100 ya existe. ¿Pero que pasa con la siguiente sentencia `INSERT`? La ejecución para en el primer error porque `$cfg['IgnoreMultiSubmitErrors']` está configurado `FALSE`, lo que le dice a phpMyAdmin que *no* ignore los errores en las sentencias múltiples. Si está configurado `TRUE`, el programa intenta todas las sentencias con éxito y conseguimos:

Su consulta se ejecutó con éxito

consulta SQL:
insert into autor values (100,'Paul Smith','111-2222');# Error insert into autor values (101,'Melanie Smith','222-3333');# Error update autos set teléfono='444-5555' where nombre_autor like '%Smith%';# Error

[\[Editar \]](#) [\[Crear código PHP \]](#)

Esta característica no funcionará como esperamos si hemos elegido más de una sentencia `SELECT`. Sólo veríamos los resultados de la última sentencia `SELECT`.

Impreso bonito (destacado de la sintaxis)

Por defecto, phpMyAdmin analiza y destaca los diferentes elementos de cualquier sentencia MySQL que procese. Esto está controlado por `$cfg['SQL']['fmtType']` que está configurado `'html'` por defecto. Este modo usa un color específico para cada elemento diferente (una palabra reservada, una variable, un comentario, etc.), tal como se describe en el conjunto `$cfg['SQL']['fmtColor']`; está en la carpeta `tema de layout.inc.php`. Los valores por defecto son:

```
$cfg['SQL']['fmtColor'] = array(
    'comment'           => '#808000',
    'comment_mysql'     => '',
    'comment_ansi'      => '',
    'comment_c'         => '',
    'digit'             => '',
    'digit_hex'         => 'teal',
    'digit_integer'     => 'teal',
    'digit_float'       => 'aqua',
    'punct'             => 'fuchsia',
    'alpha'             => '',
    'alpha_columnType'  => '#FF9900',
    'alpha_columnAttrib'=> '#0000FF',
    'alpha_reserverWord'=> '#990099',
    'alpha_functionName'=> '#FF0000',
    'alpha_identifier'  => 'black',
    'alpha_variable'    => '#800000',
    'quote'             => '#008000',
    'quote_double'      => '',
    'quote_single'      => '',
    'quote_bactick'     => ''
);
```

En los anteriores ejemplos, `fmtType` estaba configurado `'text'`, porque este modo es más legible en un libro. Este modo inserta rupturas de línea en puntos lógicos dentro de la sentencia MySQL pero aquí no cambia el color. Con `fmtType` configurado en `'html'` el informe de las sentencias SQL de phpMyAdmin sería:

```
Mostrando registros 0 - 29 (180 total, La consulta tardó 0.0376 seg)

consulta SQL:
SELECT *
FROM `autor`
WHERE 1
LIMIT 0 , 30

[ Editar ] [ Explicar el SQL ] [ Crear código PHP ] [ Actualizar ] [ Validar el SQL ]
```

`fmtType` configurado `'none'` quita cualquier tipo de formateo dejando la sintaxis intacta:

```
consulta SQL:
insert into autor values (100,'Paul Smith','111-2222');# Error insert into autor values (101,'Melanie
Smith','222-3333');# Error update autors set teléfono='444-5555' where nombre_autor like '%Smith%';#
Error
```



Los conjuntos multidimensionales que usan para mantener algunos parámetros en el archivo de configuración reflejan el estilo de programación adoptado por el equipo de desarrollo de phpMyAdmin. Esto evita que tengamos nombres de parámetros muy largos.

Validador SQL

Cada vez que phpMyAdmin transmite una consulta, el servidor MySQL la interpreta y manda comentarios. La sintaxis de la consulta debe seguir las reglas de MySQL que no tienen el mismo estándar que SQL. Si seguimos el estándar de SQL nos aseguramos que nuestras consultas se podrán usar con cualquier implementación de SQL.

Se ofrece un servicio externo gratuito: El validador Mimer de SQL. Está directamente disponible en phpMyAdmin, valida nuestra consulta teniendo en cuenta Core SQL-99 rules y envía un informe. La página inicial del Validador está en <http://developer.mimer.com/validator/index.htm>.



Este servicio almacena las consultas que recibe de forma anónima para las estadísticas. Cuando almacena la consulta, cambia el nombre de la base de datos, la tabla y las columnas con nombres genéricos. Las cadenas y los números que forman parte de la consulta se cambian por valores genéricos para proteger la información original.

Requisitos del sistema

Este validador está disponible como un servicio SOAP. Nuestro servidor debe tener XML, PCRE en PHP y soporte PEAR. También necesitamos algunos módulos PEAR. El siguiente comando (que el administrador del sistema ejecuta en el servidor) instala los módulos que necesitamos:

```
pear install Net_Socket Net_URL HTTP_Request Mail_Mime Net_DIME SOAP
```

Si tenemos problemas con este comando debido a que algunos módulos están en estado beta, podemos ejecutar el siguiente comando, que instala SOAP y otros módulos dependientes:

```
pear -d preferred_state=beta install -a SOAP
```

Hacer que el validador esté disponible

Algunos parámetros deben configurarse en `config.inc.php`. Si configuramos `$cfg['SQLQuery']['Validate']` en TRUE habilitaremos el enlace Validar SQL.

También tenemos que habilitar el validador (ya que puede haber otros validadores disponibles en versiones futuras de phpMyAdmin). Esto se hace al configurar `$cfg['SQLValidator']['use']` en TRUE.

Se puede acceder al validador con una cuenta anónima validador, por defecto, como se configura con lo siguiente.

```
$cfg['SQLValidator']['username'] = '';  
$cfg['SQLValidator']['password'] = '';
```

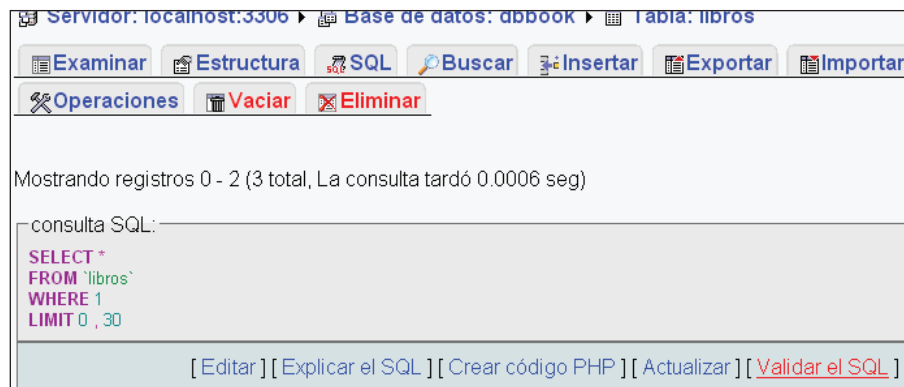
Si la compañía nos ha dado una cuenta, en su lugar podemos poner la información de la cuenta aquí.

Resultados del validador

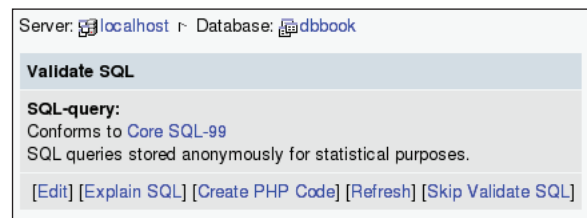
Hay dos tipos de informes que envía el Validador: Uno si la consulta está acorde con el estándar y otro si no.

Consultas acordes con el estándar

Lo intentaremos con una consulta sencilla. `Select* from libros`. Escribimos esta consulta en el recuadro de consulta y lo mandamos como siempre. En la página de resultados ahora vemos un enlace adicional: **Validar SQL**:



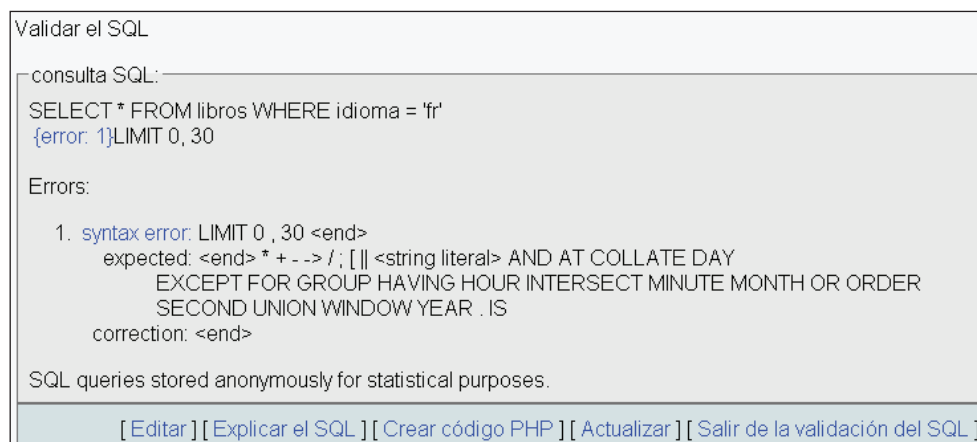
Si hacemos clic en **Validar SQL** vemos el siguiente informe:



Tenemos la opción de **Saltar Validar SQL** para ver nuestra consulta original.

Consultas no acordes con el estándar

Intentemos esta consulta que funciona correctamente en MySQL: `Select * from libros where idioma = 'en'`. Si lo mandamos al validador nos da el siguiente informe:



Cada vez que el validador encuentra un problema añade un mensaje como **{error: 1}** en el punto de error y una nota de pie de página en el informe. Esta vez, el nombre de la columna **idioma** no es estándar, de modo que el validador nos dice que en este punto esperaba un identificador.

Otro caso es el de las comillas. Si hemos hecho clic en **Examinar** de la tabla libros, phpMyAdmin genera `select * from 'libros'`, cerrando el nombre de la tabla con comillas. Esta es la manera de MySQL de proteger los identificadores, que pueden contener caracteres especiales, como por ejemplo espacios, caracteres internacionales o palabras reservadas. Sin embargo, si enviamos esta consulta al Validador, éste nos muestra que las comillas no están acordes con el estándar SQL. Incluso nos da dos errores: Uno por cada comilla.

Validar el SQL

consulta SQL:

SELECT * FROM {error: 1}'libros'{error: 2}` WHERE 1
{error: 3}LIMIT 0,30

Errors:

1. syntax error: ` libros
expected: ({ <ascii identifier> <delimited identifier> <identifier>
LATERAL MODULE ONLY TABLE UNNEST
correction: libros

2. syntax error: ` WHERE
expected: <end> , , { <ascii identifier> <delimited identifier>
<identifier> AS CROSS EXCEPT FOR FULL GROUP HAVING INNER
INTERSECT JOIN LEFT NATURAL ORDER RIGHT UNION WHERE WINDOW
correction: WHERE

3. syntax error: LIMIT 0 , 30 <end>
expected: <end> * + - > / ; < <= < > >= [|| AND AT BETWEEN COLLATE DAY
EXCEPT FOR GROUP HAVING HOUR IN INTERSECT IS LIKE MATCH MINUTE
MONTH NOT OR ORDER OVERLAPS SECOND SIMILAR UNION WINDOW YEAR .
correction: <end>

SQL queries stored anonymously for statistical purposes.

[\[Editar \]](#) [\[Explicar el SQL \]](#) [\[Crear código PHP \]](#) [\[Actualizar \]](#) [\[Salir de la validación del SQL \]](#)

Resumen

En este capítulo hemos cubierto:

- El propósito de los recuadros de consulta.
- Donde podemos encontrarlos.
- Como usar el selector de campos.
- Algunas opciones de la ventana de consulta.
- Como tener un historial de los comandos escritos.
- Consultas con sentencias múltiples.
- Usar el validador SQL.

13

Generador de consultas multi-tabla

Las páginas **Búsqueda** en la vista base de datos o tabla son para búsquedas de una tabla. En este capítulo cubriremos la característica **Consulta por ejemplo** (CPE) de múltiples tablas que está disponible en la vista base de datos.

Muchos de los usuarios de phpMyAdmin trabajan en la vista tabla, tabla por tabla, de modo que tienden a olvidarse del generador de consultas multi-tabla, que es una característica maravillosa para afinar consultas. Para abrir la página de esta característica vista base de datos de la base de datos específica (el generador de consultas sólo tiene soporte para funcionar con una base de datos a la vez) y hacemos clic en **Consulta**.

El generador de consultas no sólo es útil para situaciones multi-tabla, sino también para una tabla. Nos permite especificar criterios múltiples para una columna, una característica que la página **Buscar** de la vista tabla no posee.



Los ejemplos de este capítulo asumen que se ha hecho la instalación de un usuario de la infraestructura de tablas enlazadas; vaya al capítulo 11; de modo que hay más tablas en la base de datos **dbbook**.

La muestra de pantalla que sigue muestra la página CPE inicial. Contiene los siguientes elementos:

- Columnas criterio.
- Una interfaz para añadir filas criterio.
- Una interfaz para añadir columnas criterio.

- Un selector de tabla.
- La zona de consulta.
- Botones para actualizar o ejecutar la consulta.

Servidor: localhost - Base de datos: dbbook

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones

Campo:		
Ordenar:		
Mostrar:	☐	☐
Criterio:		
Insertar: ☐ y luego: ☐		
Borrar: ☐ O: ☒		
Modificar:	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐

Añadir/borrar fila de criterio: Añadir/borrar columna de criterio:

Usar tablas:

- autor
- libros
- pma_bookmark
- pma_column_info
- pma_history
- pma_pdf_pages
- pma_relation

Consulta a la base de datos dbbook:

Elegir tablas

La selección inicial incluye todas las tablas. En consecuencia, el **selector de campos** tiene un gran número de campos. Para nuestro ejemplo, trabajaremos solo con las tablas **autor** y **libros**:

Usar tablas:

- autor
- libros
- pma_bookmark
- pma_column_info
- pma_history
- pma_pdf_pages
- pma_relation

Luego haremos clic en **Actualizar consulta**. Con esto la pantalla se refresca y se reducen el número de campos disponibles en el selector de **Campo**. Siempre podemos cambiar la elección de tabla más tarde, con el mecanismo de nuestro navegador para elecciones múltiples y menús desplegables (normalmente con clic y control!).

Columna de criterios

Se ofrecen tres columnas de criterios por defecto. En esta sección se discuten las opciones que tenemos para editar sus criterios, que incluye opciones para seleccionar campos, clasificar columnas individuales, escribir condiciones para columnas individuales, etc.

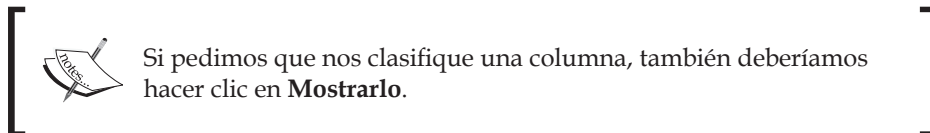
Selector de campos: Una columna o todas las columnas

El selector de **campo** contiene todas las columnas individuales de las tablas elegidas, además tiene una elección especial que termina con un asterisco (*) para cada tabla, lo cual significa que 'todos los campos' se eligen.

Para mostrar todos los campos de la tabla autor, elegimos '**autor.***' y verificamos el recuadro **Mostrar**, sin escribir un **Orden** y un **Criterio**. En nuestro caso, elegimos '**autor.nombre_autor**' ya que podemos escribir algunos criterios para el nombre de autor.

Clasificar

Por cada columna individual elegida podemos especificar una **clasificación** (en orden **Ascendente** o **Descendente**) o dejar que está línea se quede intacta (lo que quiere decir que no hay clasificación). Si elegimos más de una columna clasificada, la clasificación se hará con prioridad de la izquierda sobre la derecha.



Mostrar

Elegimos **Mostrar** para ver la columna en los resultados. Algunas veces, puede que queramos aplicar un criterio a una columna pero no incluirlo en la página de resultados. Aquí añadimos el campo teléfono, pedimos que lo clasifique y elegimos mostrar tanto el nombre como el número de teléfono. También le pedimos que clasifique el nombre en orden ascendente. La clasificación se hará primero por nombre y luego por número de teléfono si los nombres son idénticos. Esto sucede porque hay un criterio de columna a la izquierda de la columna teléfono y por ello tiene mayor prioridad:

Field:	'authors'.author_name	'authors'.phone	
Sort:	Ascending	Ascending	
Show:	☒	☒	☐
Criteria:			
Ins: ☐ And: ☐			
Del: ☐ Or: ☒			
Modify:	Or: ☐ And: ☒ Ins ☐ Del ☐	Or: ☐ And: ☒ Ins ☐ Del ☐	Or: ☐ And: ☒ Ins ☐ Del ☐

Modificar la consulta

En cualquier punto, podemos hacer clic en el botón **Modificar consulta** para ver el progreso la consulta que hemos generado. Seguramente, tendremos que hacer clic al menos una vez antes de ejecutar la consulta. Por ahora, hagamos clic para ver la consulta que hemos generado en la zona de consulta. En los ejemplos que siguen, haremos clic en **Modificar consulta** después de cada modificación:

Campo:	'autor'. 'nombre_autor'	'autor'. 'teléfono'
Ordenar:	Ascendente	Ascendente
Mostrar:	☒	☒
Criterio:		
Insertar: ☐ y luego: ☐ Borrar: ☐ O: ☒		
Modificar:	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐

Hemos elegido dos tablas, pero todavía no hemos elegido ninguna columna de la tabla **libros**. De forma, que esta tabla no se menciona en la consulta generada.

Criterios

En la línea de **Criterios**, escribimos una condición (respetando la sintaxis de la cláusula WHERE SQL) para cada una de las columnas correspondientes. Por defecto tenemos dos filas de criterios. Para encontrar todos los autores con **Smith** en su nombre, usamos un criterio LIKE y hacemos clic en **Actualizar consulta:**

Campo:	'autor'. 'nombre_autor'	'autor'. 'teléfono'	
Ordenar:	Ascendente	Ascendente	
Mostrar:	☒	☒	☐
Criterio:			
Insertar: ☐ y luego: ☐ Borrar: ☐ O: ☒			
Modificar:	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐
Añadir/borrar fila de criterio: 0 Añadir/borrar columna de criterio: 0 Modificar la consulta			
Usar tablas: autor libros pma_bookmark pma_column_info pma_history pma_pdf_pages pma_relation	Consulta a la base de datos dbbook: <pre> SELECT 'autor'. 'nombre autor', 'autor'. 'teléfono' FROM 'autor' ORDER BY 'autor'. 'nombre autor' ASC, 'autor'. 'teléfono' ASC </pre> Ejecutar la consulta		

Tenemos otra línea disponible para escribir criterios adicionales. Digamos que también encontrar a la autora 'Maria Sunshine'. Esta vez usamos una condición =. Las dos filas de condiciones se unirán por **OR**, seleccionado por defecto en la parte izquierda de la interfaz.

Campo:	'autor'.nombre_autor	'autor'.teléfono	
Ordenar:	Ascendente	Ascendente	
Mostrar:	☒	☒	☐
Criterio:	like %Smith%		
Insertar: ☐ y luego: ☐ Borrar: ☐ O: ☒			
Modificar:	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐
Añadir/borrar fila de criterio: 0 Añadir/borrar columna de criterio: 0 Modificar la consulta			
Usar tablas: autor libros pma_bookmark pma_column_info pma_history pma_pdf_pages pma_relation			
Consulta a la base de datos dbbook: <pre> SELECT 'autor'. nombre_autor', 'autor'. 'teléfono' FROM 'autor' WHERE ('autor'. nombre_autor' like %Smith%) ORDER BY 'autor'. nombre_autor' ASC </pre> Modificar la consulta Ejecutar la consulta			

Para demostrar mejor que la condición OR enlaza ambas filas de criterios, ahora añadamos una condición al número de teléfono:

Campo:	'autor'.nombre_autor	'autor'.teléfono	
Ordenar:	Ascendente	Ascendente	
Mostrar:	☒	☒	☐
Criterio:	like %Smith%		
Insertar: ☐ y luego: ☐ Borrar: ☐ O: ☒	= 'Maria Sunshine'		
Modificar:	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐
Añadir/borrar fila de criterio: 0 Añadir/borrar columna de criterio: 0 Modificar la consulta			
Usar tablas: autor libros pma_bookmark pma_column_info pma_history pma_pdf_pages pma_relation			
Consulta a la base de datos dbbook: <pre> FROM autor WHERE ('autor'. nombre_autor' like %Smith%) OR ('autor'. nombre_autor' = 'Maria Sunshine') ORDER BY 'autor'. nombre autor' ASC, </pre> Modificar la consulta Ejecutar la consulta			

Si examinamos el posicionado de los operadores **AND** y **OR**, podemos ver que las primeras condiciones están enlazadas con **AND** (porque **AND** se elige en la columna **nombre_autor**) y la segunda fila de condiciones está enlazada con el resto con **OR**. La condición que acabamos añadir (**like '%8%'**) no significa que busquemos a todo el mundo, ya que en un ejercicio del capítulo 12 cambiamos el número de teléfono de todos los autores con nombre 'Smith' a '444-5555'.

Si queremos escribir otro criterio en la misma columna, sólo tenemos que añadir una fila de criterios.

Ajustar el número de filas de criterios

El número de filas de criterios se puede cambiar de dos maneras. Primero, podemos seleccionar el recuadro **Insertar** que está debajo de **Criterios** para añadir una fila de criterios (se hace después de **Modificar consulta**):

También podemos usar el diálogo **Añadir/Suprimir una fila de criterios**; aquí elegimos añadir dos filas:

Esto produce la siguiente pantalla:

Campo:	'autor'.nombre_autor	'autor'.telefono	
Ordenar:	Ascendente	Ascendente	
Mostrar:	☒	☒	☐
Criterio:	like %Smith%	like %8%	
Insertar: ☐ y luego: ☐ Borrar: ☐ O: ☒	= 'Maria Sunshihe'		
Insertar: ☐ y luego: ☐ Borrar: ☐ O: ☒			
Insertar: ☐ y luego: ☐ Borrar: ☐ O: ☒			
Insertar: ☐ y luego: ☐ Borrar: ☐ O: ☒			
Modificar:	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐	O: ☐ y luego: ☒ Insertar ☐ Borrar ☐

También podemos suprimir filas de criterios. Esto se puede hacer si elegimos números negativos en el diálogo **Añadir/Suprimir Criterios** o si marcamos el recuadro **Suprimir** que está al lado de la(s) fila(s) que queremos borrar. Quitemos las dos filas que acabamos de añadir ya que no las necesitamos ahora:

Insertar: ☐ y luego: ☐ Borrar: ☒ O: ☒			
Insertar: ☐ y luego: ☐ Borrar: ☒ O: ☒			

El botón **Modificar consulta** refresca la página para mostrar el ajuste especificado.

Ajustar el número de columnas de criterios

Podemos añadir o suprimir columnas con un mecanismo similar: Con los recuadros **Insertar** o **Suprimir** que están debajo de cada columna, o con el diálogo **Añadir/Suprimir Columnas campo**. Ya teníamos una columna sin usar. Aquí hemos añadido una columna, usando el recuadro **Insertar** que está debajo de la columna sin usar (esta vez la necesitaremos):

Campo:	'autor'.nombre_autor	'autor'.telefono		
Ordenar:	Ascendente	Ascendente		
Mostrar:	☒	☒	☐	☐
Criterio:	like %Smith%	like %8%		
Insertar: ☐ y luego: ☐	= 'Maria Sunshihe'			
Borrar: ☐ O: ☒				
Insertar: ☐ y luego: ☐				
Borrar: ☐ O: ☒				
Modificar:	O: ☐ y luego: ☒	O: ☐ y luego: ☒	O: ☐ y luego: ☒	O: ☐ y luego: ☒
	Insertar ☐ Borrar ☐	Insertar ☐ Borrar ☐	Insertar ☐ Borrar ☐	Insertar ☐ Borrar ☐

Joins automáticas

Ahora añadamos algunos campos de nuestra tabla **libros** y veamos que pasa:

Campo:	'autor'.nombre_autor	'autor'.telefono	'libros'.Titulo
Ordenar:	Ascendente	Ascendente	
Mostrar:	☒	☒	☒
Criterio:	like %Smith%	like %8%	
Insertar: ☐ y luego: ☐			
Borrar: ☐ O: ☒			
Modificar:	O: ☐ y luego: ☒	O: ☐ y luego: ☒	O: ☐ y luego: ☒
	Insertar ☐ Borrar ☐	Insertar ☐ Borrar ☐	Insertar ☐ Borrar ☐

Añadir/borrar fila de criterio: 0 Añadir/borrar columna de criterio: 0 [Modificar la consulta](#)

Usar tablas:
autor
libros
países
pma_bookmark
pma_column_info
pma_history
pma_pdf_pages

Consulta a la base de datos dbbook:

```

libros". "genero"
FROM "autor"
LEFT JOIN "libros" ON
"autor"."autor_id" =
"libros"."autor_id"
WHERE (('autor"."nombre_autor"
like %Smith%') AND

```

[Modificar la consulta](#) [Ejecutar la consulta](#)

phpMyAdmin usa su conocimiento de las relaciones definidas entre las tablas para generar una **LEFT JOIN** en el campo clave común **autor_id**.



Puede que haya más de dos tablas implicadas en una **JOIN**.

Ejecutar la consulta

Hacer clic en el botón de **Ejecutar consulta** hace que se ejecute la consulta. En la versión actual de phpMyAdmin (2.8.0.4) no hay una manera fácil (excepto si hacemos clic en el botón **Volver** del navegador) Volver a la página de generación de la consulta cuando hemos ejecutado nuestra consulta. En el siguiente capítulo (*Marcadores*) discutiremos como guardar la consulta generada para ejecutarla más tarde.

consulta SQL:

```

SELECT `autor`,`nombre_autor`,`autor`,`teléfono`,`libros`,`Título`,`libros`,`género`
FROM `autor`
LEFT JOIN `libros` ON `autor`.`autor_id` = `libros`.`autor_id`
WHERE (
(
`autor`.`nombre_autor` LIKE '%Smith%'
)
AND (
`autor`.`teléfono` LIKE '%8%'
)
)
OR (
`autor`.`nombre_autor` = 'Maria Sunshine'
)
ORDER BY `autor`.`nombre_autor` ASC, `autor`.`teléfono` ASC
LIMIT 0, 30

```

[Editar] [Explicar el SQL] [Crear código PHP] [Actualizar] [Validar el SQL]

Mostrar: 30
filas empezando de 0

en modo horizontal y repetir los encabezados cada 100 celdas

→ T ←

nombre_autor	teléfono	Título	género
Maria Sunshine	222-2222	Recuerdos futuros	F

Resumen

Hemos cubierto los siguientes conceptos:

- Abrir el generador de consultas.
- Elegir las tablas.
- Escribir en la columna de criterios.
- Clasificar y mostrar columnas.
- Alterar el número de filas o columnas de criterios.
- Relaciones AND/OR entre filas y columnas.
- Usar joins automáticas entre tablas

14

Marcadores

En este capítulo cubrimos una de las características de las tablas enlazadas: Los marcadores de las consultas. Poder etiquetar las consultas y recuperar por su etiqueta más tarde puede ahorrarnos mucho tiempo. En el capítulo 12 hemos aprendido sobre la característica del historial SQL, que almacena las consultas automáticamente (temporal o permanentemente).

Los marcadores son consultas que :

- Se han almacenado de forma permanente.
- Son visibles.
- Son borrables.
- Están relacionadas a una base de datos.
- Están registradas tan solo como una consecuencia de los deseos del usuario.
- Están etiquetadas.
- Son privadas por defecto (solo están disponibles para el usuario que las ha creado), pero tienen la posibilidad de ser públicas.

Un marcador también puede ser una parte variable, como se explica en la sección *Pasar un valor de parámetro a un marcador* más tarde en el capítulo.

No hay una sub-página marcador para administrar los marcadores. En su lugar, las diferentes acciones sobre los marcadores están disponibles en páginas específicas como por ejemplo las páginas de resultados y las páginas recuadro de consulta.

Crear un marcador después de ejecutar con éxito una consulta

La creación del marcador inicial es posible con el botón **Guardar esta consulta SQL en mis favoritos**. Este botón aparece solo después de ejecutar una consulta que genera resultados (cuando se encuentra al menos una fila), y por eso, este método de crear marcadores sólo almacena las sentencias `SELECT`. Por ejemplo, una consulta compleja producida por un generador de consultas multi-tabla (como se ha visto en el capítulo 13) se puede almacenar como marcador de esta manera, siempre que encuentre resultados.

Veamos un ejemplo. En la página **Buscar** de la tabla **libros**, podemos escribir los valores de búsqueda tal y como se muestran en la siguiente pantalla:

Seleccionar campos (al menos uno):

- ISBN
- Título
- número_pág
- autor_id
- idioma
- descripción
- foto-cubierta
- género
- fecha_publicación

☐ DISTINCT

registros por página: 30

Mostrar en este orden:

☒ Ascendente ☐ Descendente

Insertar las condiciones de búsqueda (cuerpo de la cláusula "where"): ?

Continuar

O Hacer una consulta (comodín: "%")

Campo	Tipo	Cotejamiento	Operador	Valor
ISBN	varchar(25)	latin1_swedish_ci	LIKE %...%	
Título	varchar(100)	latin1_swedish_ci	LIKE %...%	
número_pág	int(11)		=	
autor_id	int(11)		=	1 - John Smith
idioma	enum('en', 'es')	latin1_swedish_ci	=	en es
descripción	text	latin1_swedish_ci	LIKE %...%	
foto-cubierta	blob		LIKE %...%	
género	set('F', 'Ni', 'N')	latin1_swedish_ci	LIKE %...%	
fecha_publicación	datetime		=	

La página de resultados tiene un diálogo marcador. Sólo tenemos que escribir la etiqueta para este marcador y hacer clic en **Guardar esta consulta SQL en mis favoritos** para guardar esta consulta como un marcador. Los marcadores se guardan en la tabla definida por `$cfg['Servers'][$i]['bookmarktable']`.

→T←		Título	número_pág aproximado	idioma	descripción
☐	Editar Borrar	Cien años de cine (volumen 1)	600	en	
☐	Editar Borrar	Cien años de cine (volumen 2)	600	en	

Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Mostrar : filas empezando de

en modo y repetir los encabezados cada celdas

Insertar nueva fila Vista de impresión Exportar

Guardar esta consulta en favoritos

Etiqueta: ☐ Permitir que todo usuario pueda acceder a este favorito

Este diálogo marcador se puede ver en cualquier página que tenga resultados. De hecho, solo tenemos que hacer clic en **Examinar** en cualquier tabla para obtener resultados y luego almacenar esta consulta como un marcador. Sin embargo, no tiene mucho sentido guardar una consulta (en un marcador) que se puede hacer con facilidad con un clic.

Almacenar un marcador antes de mandar una consulta

Algunas veces queremos almacenar un marcador incluso si la consulta no encuentra resultados. Este puede ser el caso si los datos que concuerdan aún no están presentes o si la consulta no es `SELECT`. Para conseguirlo, tenemos disponible el diálogo **Continuar y Guardar esta consulta SQL en mis favoritos** tal como se ve aquí:

- La vista tabla: En cada página en la que se muestra un recuadro de consulta.
- La **ventana consulta**. La pestaña **SQL**.

Ahora vamos a la sub-página **SQL** de la tabla libros, escribimos una consulta y ponemos el marcador libros en francés directamente en el campo **Opciones de etiquetado de marcadores**. Luego, en lugar de hacer clic en **Continuar** para ejecutar la consulta, hacemos clic en **Continuar y marcar esta consulta SQL**, que ejecuta y almacena la consulta. No importa si la consulta no encuentra nada:

Servidor: localhost - Base de datos: dbbook - Tabla: libros *InnoDB free: 4096 kB; ("autor_id") REFER `dbbook/autor` ("autor_id")*

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Ejecutar la(s) consulta(s) SQL en la base de datos dbbook: @

SELECT * FROM `libros` WHERE `idioma` = 'fr'

Campos
ISBN
Título
numero_pág
autor_id
idioma

Guardar esta consulta en favoritos: libros en francés ☐ Permitir que todo usuario pueda acceder a este favorito

☐ Reemplazar el favorito existente que tenga el mismo nombre

☒ Mostrar esta consulta otra vez Continuar

Así es como podemos generar marcadores para las consultas no SELECT como UPDATE, DELETE, CREATE TABLE, etc.

Marcadores multi-consulta

Se puede almacenar más de una consulta en un marcador (separadas por un punto y coma). Esto es muy útil para las consultas no SELECT. Colocar muchas consultas SELECT no nos dará el resultado que queremos ya que sólo vemos los datos del último SELECT.

Recuperar de la lista de marcadores

Ahora podemos encontrar estos marcadores con facilidad en las siguientes páginas:

- La vista tabla: La sub-página **Estructura** o **SQL** de cualquier tabla de **dbbook**.
- La **ventana consulta**. La pestaña **Historial SQL**.
- Mientras navegamos por la tabla **pma_bookmark** (ver la sección *Ejecutar marcadores desde pma_bookmark* más tarde).
- La vista base de datos: La sub-página **SQL** de la base de datos **dbbook**.

Servidor: localhost - Base de datos: dbbook

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Eliminar

Ejecutar la(s) consulta(s) SQL en la base de datos dbbook: @

Guardar esta consulta en favoritos: ☐ Permitir que todo usuario pueda acceder a este favorito

☐ Reemplazar el favorito existente que tenga el mismo nombre

☒ Mostrar esta consulta otra vez Continuar

Consulta guardada en favoritos

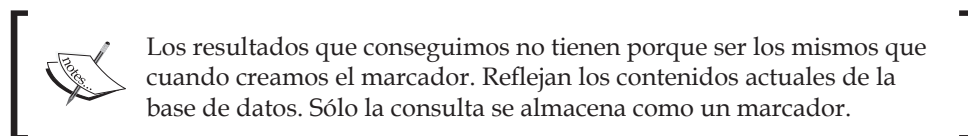
Variable ☒ Enviar ☐ Solamente ver ☐ Borrar

libros autor 1
libros en francés Continuar

El sistema pone números a los marcadores. Hay tres elecciones disponibles para recuperar un marcador: **Ejecutar**, **Sólo ver** y **Suprimir** (**Ejecutar** es la acción por defecto).

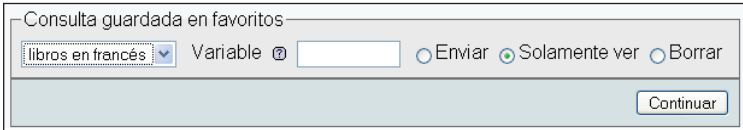
Ejecución de un marcador

Si elegimos el primer marcador y hacemos clic en **Continuar** ejecutaremos la consulta almacenada y veremos los resultados. La página que resulta de la ejecución del marcado no tiene otro diálogo para crear un marcado ya que esto es superfluo.



Manipulación de un marcador

Algunas veces solo queremos asegurarnos que los contenidos de un marcador son correctos. Aquí elegimos el segundo marcador y elegimos **Sólo ver**:



Sólo se mostrará la consulta. Entonces podemos hacer clic en **Editar** y reescribir sus contenidos. Si lo hacemos, estaremos editando una copia de la consulta marcada original. Para guardar esta consulta recién editada, podemos guardarla como un marcador pero esto creará otro marcador *incluso si elegimos la misma etiqueta de marcador*. Los marcadores tienen un número identificador además de una etiqueta elegida por el usuario.

Se puede borrar un marcador con la opción **Suprimir**. No hay un diálogo de confirmación para ver que efectivamente hemos suprimido el marcador. La supresión está seguida de un mensaje que enuncia **El marcador se ha suprimido**.

Marcadores públicos

Todos los marcadores que creamos son privados por defecto. Cuando se crea un marcador, el usuario con el que estamos conectados se almacena junto con el marcador. Supongamos que elegimos **Dejar que todos los usuarios accedan a este marcador** como se muestra en la siguiente pantalla:

☐ Guardar esta consulta en favoritos

Etiqueta:
☒ Permitir que todo usuario pueda acceder a este favorito

Esto tendrá el siguiente efecto:

- Todos los usuarios que tienen acceso a la misma base de datos (la actual) tendrán acceso al marcador.
- La posibilidad que los usuarios obtengan resultados significativos del marcador depende de los privilegios que tengan en las tablas referenciadas por el marcador.
- Los usuarios podrán suprimir el marcador.

Consulta inicial por defecto para una tabla

En los ejemplos anteriores, hemos elegido etiquetas de marcador conforme nuestras preferencias, pero por convención, si un marcador tiene el mismo nombre que una tabla, se ejecutará cuando hagamos clic en **Examinar** esta tabla.

Supongamos que estamos interesados en ver (por defecto, en el modo Examinar) los libros con un número de páginas inferior a 300. Primero generamos la consulta apropiada y luego en los resultados usamos **libros** como etiqueta:

consulta SQL:


```

SELECT 'ISBN' , 'Título' , 'número_pág' , 'autor_id' , 'idioma'
FROM 'libros'
WHERE 'número_pág' <300
LIMIT 0 , 30
    
```

[\[Editar \]](#)
[\[Explicar el SQL \]](#)
[\[Crear código PHP \]](#)
[\[Actualizar \]](#)

Mostrar: 30

filas empezando de 0

en modo horizontal y repetir los encabezados cada 100 celdas

	ISBN	número de libro	Título	número_pág aproximado	autor_id ver tabla autor	idioma
☐	 Editar	 Borrar	1-234567-2...	Recuerdos ...	200	2 en

↶ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Mostrar: 30

filas empezando de 0

en modo horizontal y repetir los encabezados cada 100 celdas

[Insertar nueva fila](#)
[Vista de impresión](#)
[Previsualización para imprimir \(documento completo\)](#)
[Exportar](#)

☐ Guardar esta consulta en favoritos

Etiqueta:
☐ Permitir que todo usuario pueda acceder a este favorito

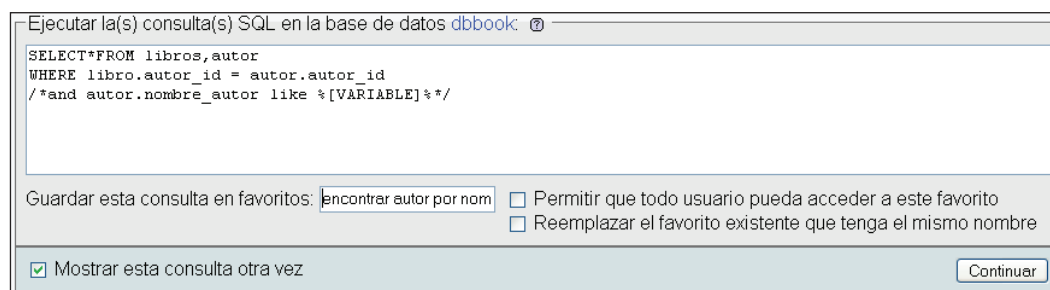
Marcadores parámetros

Si volvemos a mirar el primer marcador que hemos creado (encontrar todos los **libros** del autor 1), nos damos cuenta que, aunque es útil, estaba limitado a encontrar siempre el mismo autor.

Una sintaxis especial de consulta nos permite pasar parámetros a marcadores. Esta sintaxis se aprovecha del hecho que MySQL ignora los comentarios SQL cerrados entre `/*` y `*/`. Si la construcción `/* [VARIABLE] */` existe en algún lugar de la consulta, se expandirá a la hora de ejecutar con el valor apropiado cuando recuperemos el marcador.

Crear un marcador parametrizado

Digamos que queremos encontrar todos los libros de un autor cuando sepamos el nombre del autor. Primeros escribimos la consulta tal y como se muestra en la pantalla. Tenga en cuenta que los caracteres comentario (`/* */`) se expandirán más tarde y que los identificadores se borrarán:



Ejecutar la(s) consulta(s) SQL en la base de datos [dbbook](#): ⓘ

```
SELECT*FROM libros,autor
WHERE libro.autor_id = autor.autor_id
/*and autor.nombre_autor like %[VARIABLE]**/
```

Guardar esta consulta en favoritos: ☐ Permitir que todo usuario pueda acceder a este favorito
☐ Reemplazar el favorito existente que tenga el mismo nombre

☒ Mostrar esta consulta otra vez Continuar

Lo etiquetamos y hacemos clic en **Continuar** y **Guardar esta consulta SQL en mis favoritos**. La primera ejecución no hace nada especial, sólo almacena el marcador.

En este ejemplo, tenemos dos condiciones con la cláusula `WHERE`, una que contiene una sintaxis especial. Si nuestro único criterio en la cláusula `WHERE` requiriese un parámetro, podríamos usar una sintaxis como `WHERE 1 /* AND autor_id = [VARIABLE] */`.

Pasar un valor de parámetro a marcador

Para verificar el marcador, lo recuperamos como de normal y escribimos un valor en el campo **Variable**:

Consulta guardada en favoritos

encontrar autor por nombre Variable @ ☒ Enviar ☐ Solamente ver ☐ Borrar

Luego hacemos clic en **Continuar** y vemos la consulta expandida y los libros del autor Smith:

Mostrando registros 0 - 1 (2 total, La consulta tardó 0.0006 seg)

consulta SQL:

```
SELECT *
FROM libros_autor
WHERE libros_autor_id = autor_autor_id
AND autor_nombre_autor LIKE '%Smith%'
LIMIT 0, 30
```

[Editar] [Crear código PHP]

en modo filas empezando de y repetir los encabezados cada celdas

ISBN	número de libro	Título	número_pág aproximado	autor_id ver tabla autor	idioma	descripción	foto-cubierta	género	fecha_publicación	autor_id
1-234567-89-0		Cien años de cine (volumen 1)	600	1	en		[BLOB - 0 Bytes]	F	0000-00-00 00:00:00	1
1-234567-90-0		Cien años de cine (volumen 2)	600	1	en		[BLOB - 0 Bytes]	F	0000-00-00 00:00:00	1

Ejecutar marcadores desde pma_bookmark

Esta característica solo está disponible para los usuarios que tienen acceso a la tabla **pma_bookmark**. Este es el nombre por defecto que se le da cuando se instala la infraestructura de las tablas enlazadas. En una instalación multi-usuario, esta tabla normalmente está en una base de datos invisible para los usuarios sin privilegios. Si examinamos esta tabla vemos un Nuevo botón **Ejecutar la consulta marcada**, que desencadena la ejecución de la consulta:

→ T ←			id	dbase	user	label	query
☐	 Editar Ejecute la consulta marcada previamente		1	dbbook	root	libros autor 1	SELECT 'Titulo', 'numero_pág', 'idioma', 'descripción' FROM 'libros' WHERE 'autor_id' = 1
☐	 Editar Ejecute la consulta marcada previamente		2	dbbook	root	libros en francés	SELECT * FROM 'libros' WHERE 'idioma' = 'fr'
☐	 Editar Ejecute la consulta marcada previamente		4	dbbook	root	encontrar autor por nombre	SELECT * FROM libros, autor WHERE libros.autor_id = autor.autor_id /*and autor.nombre_autor like %[VARIABLE]%%*/
☐	 Editar Ejecute la consulta marcada previamente		5	dbbook	root	libros	SELECT * FROM 'libros'

Resumen

En este capítulo:

- Hemos visto como registrar marcadores (antes y después de mandar una consulta) y luego como manipularlos. Hemos visto como hacer públicos algunos marcadores.
- Hemos cubierto las nociones: Consulta inicial por defecto para el modo **Examinar**, pasar parámetros a marcadores y ejecutarlos directamente desde la tabla **pma_bookmark**.

15

Documentación del sistema

Producir y mantener una buena documentación sobre la estructura de los datos es crucial para el éxito del proyecto, en especial cuando se trata de un proyecto de equipo. Afortunadamente, phpMyAdmin tiene características que se encargan de esto. Cuando phpMyAdmin genera resultados, siempre hay un enlace **Vista impresión** que se puede usar para generar un informe imprimible de los datos. La característica **Vista impresión** también se puede usar para producir documentación básica y se hace con dos pasos. Con el primer clic en la **Vista impresión** aparece un informe en la pantalla, con el botón **Imprimir** que está abajo en la página. Este botón **Imprimir** genera un informe formateado para la impresora.

Vista impresión de la base de datos

En la sub-página **Estructura** de la base de datos, la **Vista impresión** genera la lista de tablas y se puede ver el tamaño de las tablas, el número de registros, los comentarios de las tablas y la fecha.

Server: localhost Database: dbbook

Table	Records	Type	Size	Comments
authors	4	MyISAM	2.1 KB	Creation: Jul 07, 2004 at 01:28 PM Last update: Jul 07, 2004 at 02:56 PM
books	3	MyISAM	8.5 KB	Creation: Jul 07, 2004 at 01:27 PM Last update: Jul 07, 2004 at 01:27 PM
pma_bookmark	0	MyISAM	1.0 KB	Bookmarks Creation: Jul 08, 2004 at 08:58 AM Last update: Jul 08, 2004 at 08:58 AM
pma_column_info	0	MyISAM	1.0 KB	Column information for phpMyAdmin Creation: Jul 08, 2004 at 08:58 AM Last update: Jul 08, 2004 at 08:58 AM
pma_history	0	MyISAM	1.0 KB	SQL history for phpMyAdmin Creation: Jul 08, 2004 at 08:58 AM Last update: Jul 08, 2004 at 08:58 AM
pma_pdf_pages	0	MyISAM	1.0 KB	PDF relation pages for phpMyAdmin Creation: Jul 08, 2004 at 08:58 AM Last update: Jul 08, 2004 at 08:58 AM
pma_relation	0	MyISAM	1.0 KB	Relation table Creation: Jul 08, 2004 at 08:58 AM Last update: Jul 08, 2004 at 08:58 AM
pma_table_coords	0	MyISAM	1.0 KB	Table coordinates for phpMyAdmin PDF output Creation: Jul 08, 2004 at 08:58 AM Last update: Jul 08, 2004 at 08:58 AM
pma_table_info	0	MyISAM	1.0 KB	Table information for phpMyAdmin Creation: Jul 08, 2004 at 08:58 AM Last update: Jul 08, 2004 at 08:58 AM
9 table(s)	7	--	17.6 KB	

Print

Vista impresión selectiva de la base de datos

Algunas veces preferimos tener un informe de ciertas tablas. Esto se puede hacer desde la sub-página **Estructura** de la base de datos, si elegimos las tablas que queremos y usar el menú desplegable para ir a la **Vista impresión**:

Tabla		Acción					Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar	
☒	autor							2	InnoDB	latin1_swedish_ci	32.0 KB	-
☒	libros							3	InnoDB	latin1_swedish_ci	48.0 KB	-
☐	pma_bookmark							4	MyISAM	latin1_swedish_ci	2.6 KB	160 Bytes
☐	pma_column_info							0	MyISAM	latin1_swedish_ci	1.0 KB	-
☐	pma_history							0	MyISAM	latin1_swedish_ci	1.0 KB	-
☐	pma_pdf_pages							0	MyISAM	latin1_swedish_ci	1.0 KB	-
☐	pma_relation							0	MyISAM	latin1_swedish_ci	1.0 KB	-
☐	pma_table_coords							0	MyISAM	latin1_swedish_ci	1.0 KB	-
☐	pma_table_info							2	MyISAM	latin1_swedish_ci	2.1 KB	-
9 tabla(s)		Número de filas					11	InnoDB	latin1_swedish_ci	89.6 KB	160 Bytes	
↑ Marcar todos/as / Desmarcar todos / Marcar las tablas con residuo a depurar												
Vista de impresión Diccionario de datos Crear nueva tabla en la base de datos dbbook Nombre: Número de campos: Para los elementos que están marcados: Vaciar Eliminar Vista de impresión Revisar la tabla Optimizar la tabla Reparar la tabla Analizar la tabla Continuar												

Vista impresión de la tabla

Incluso hay un enlace **Vista impresión** en la sub-página Estructura de todas las tablas, con esto vemos información sobre las columnas, los índices, el uso de espacio y las estadísticas de las filas, como se ve aquí:

libros						
Comentarios de la tabla: InnoDB free: 4096 kB; (`autor_id`) REFER `dbbook/autor`(`autor_id`)						
Campo	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
ISBN	varchar(25)	No			número de libro	
Título	varchar(100)	No				
número_pág	int(11)	No	0		aproximado	
autor_id	int(11)	No	0	autor -> autor_id	ver tabla autor	
idioma	enum('en', 'es')	No	en			
descripción	text	No				
foto-cubierta	blob	No				
género	set('F', 'Ni', 'N')	No	F			
fecha_publicación	datetime	No	0000-00-00 00:00:00			

Índices:			
Nombre de la clave	Tipo	Cardinalidad	Campo
PRIMARY	PRIMARY	3	ISBN
por_título	INDEX	3	Título 30
autor_idioma	INDEX	3	autor_id idioma

Espacio utilizado:			Estadísticas de la fila:	
Tipo	Uso		Enunciado	Valor
Datos	16,384	Bytes	Formato	dinámico/a
Índice	32,768	Bytes	Filas	3
Total	49,152	Bytes	Longitud de la fila σ	5,461
			Tamaño de la fila σ	16,384 Bytes
			Creación	22-05-2006 a las 21:11:25

Imprimir

Diccionario de datos

Hay disponible un informe más completo sobre las tablas y las columnas de una base de datos en la vista base de datos, desde la sub-página **Estructura**. Sólo tenemos que hacer clic en el **Diccionario de datos** para obtener un informe sobre todas las tablas (del que se muestra una parte aquí):

autor						
Comentarios de la tabla: InnoDB free: 4096 kB						
Campo	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
autor_id	int(11)	No	0			
nombre_autor	varchar(30)	No				
teléfono	varchar(30)	Sí	NULL			
libros						
Comentarios de la tabla: InnoDB free: 4096 kB; (`autor_id`) REFER `dbbook/autor`(`autor_id`)						
Campo	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
ISBN	varchar(25)	No			número de libro	
Título	varchar(100)	No				
número_pág	int(11)	No	0		aproximado	
autor_id	int(11)	No	0	autor-> autor_id	ver tabla autor	
idioma	enum('en', 'es')	No	en			
descripción	text	No				
foto-cubierta	blob	No				
género	set('F', 'NI', 'N')	No	F			
fecha_publicación	datetime	No	0000-00-00 00:00:00			

La columna MIME está vacía hasta que añadamos información relacionada con MIME a algunas columnas; esto se explica en el capítulo 16.

Esquema relacional en PDF

En el capítulo 11, hemos definido las relaciones entre las tablas **libros** y **autor**. Estas relaciones se usaron para varias funciones de clave foráneas (por ejemplo, obtener una lista de los valores posibles en el modo **Insertar**). Ahora examinaremos una característica que nos permite generar un esquema personalizado relacional de nuestras tablas en el popular formato PDF.

Añadir una tercera tabla a nuestro modelo

Para obtener un esquema más completo, ahora añadiremos otra tabla a nuestra base de datos, la tabla **países**. Aquí tenemos el fichero de exportación:

```
CREATE TABLE 'países' (
  'código_país' char(2) NOT NULL default '',
  'descripción' varchar(50) NOT NULL default '',
  PRIMARY KEY ('código_país')
) TYPE=MyISAM;

INSERT INTO 'país' VALUES ('ca', 'Canadá');
INSERT INTO 'país' VALUES ('uk', 'Reino Unido');
```

Ahora enlazaremos esta tabla con la tabla **autores**. Primero, en la **Vista relación** de la tabla **países**, especificamos el **Campo de presentación**:

Enlaces a	
Relaciones internas	
código país	--
descripción	--
Elegir el campo a mostrar: descripción	

Luego, en la tabla **autor**, añadimos el campo **código_país**, y en la **Vista relación**, lo enlazamos a la tabla **países** recién creada. Debemos recordar hacer clic en **Continuar** para que la relación se guarde. Para este ejemplo, no es necesario escribir ningún dato de autor ya que sólo estamos interesados en el esquema relacional.

Enlaces a	
Relaciones internas	
autor_id	--
nombre_autor	--
teléfono	--
código país	países->código país

Editar páginas PDF¹

En la vista base de datos de la base de datos **dbbook**, vamos a la sub-página **Estructura** y hacemos clic en el enlace **Editar páginas PDF**. Cada esquema relacional es una página y aquí podemos crearlos o editarlos.

Planear la página

En la versión actual de phpMyAdmin, un esquema relacional no se puede expandir a múltiples bases de datos. Pero incluso trabajando con una base de datos, el número de las tablas puede ser grande. Representar las diferentes relaciones entre ellas de manera clara será un reto. Es por ello que usamos varias páginas, cada una muestra algunas tablas y sus relaciones.

1. En las versiones superiores de phpMyAdmin, este enlace no está en la sub-página Estructura sino en la sub-página Operaciones.

También tenemos que tener en cuenta las dimensiones del resultado final. Si imprimimos un folio tenemos menos espacio para mostrar todas nuestras tablas pero tenemos un esquema legible.

Crear una página nueva

Ya que no tenemos ninguna página tenemos que crear una. Ya que la tabla más importante es **libros**, elegimos llamar esta página **libros**.

En el siguiente paso, elegiremos que tablas queremos ver en el esquema relacional. Podemos elegir todas las tablas una por una, pero para un buen principio, se recomienda seleccionar el recuadro **Presentación automática**: Esta opción coge todas las tablas enlazadas de nuestra base de datos y las pone en la lista de las tablas que se tienen que incluir en el esquema. Luego genera las coordenadas apropiadas para que las tablas aparezcan en una presentación en espiral, que empezará desde el centro del esquema. Estas coordenadas están expresadas en milímetros, con (0,0) en la esquina izquierda superior. Luego hacemos clic en **Continuar**.

Crear una nueva página ☒ (Diseño automático)

Editar una página

Ahora obtenemos una página con tres secciones diferentes. La primera es el menú principal, donde hemos elegido la página que queremos trabajar (del menú desplegable); también podemos **Suprimir** la página elegida. También podemos crear un segundo esquema (página).

Elegir la página a editar 1: libros ☒ Editar ☐ Borrar Crear una nueva página ☐ (Diseño automático)

La siguiente sección es la tabla emplazamiento. Ahora vemos los beneficios de la característica **Presentación automática**: Ya tenemos las tres tablas seleccionadas con las coordenadas (X,Y) escritas. Tenemos la opción de añadir una tabla (en la última línea), suprimir una tabla (con el recuadro) y cambiar las coordenadas (representa la posición de la esquina izquierda superior de cada tabla del esquema):

Seleccionar tablas			
cambiar el estado del scratchboard		Reiniciar	
Tabla	Borrar	X	Y
autor	☐ Borrar	410	300
libros	☐ Borrar	300	300
países	☐ Borrar	410	421
--	☐ Borrar	410	421

☒ Nombre de las columnas

Para ayudarnos a configurar las coordenadas exactas, hay disponible un editor visual para los navegadores con JavaScript activado. Este editor aparece si hacemos clic en el botón **Cambiar el estado del scratchboard** una vez, desaparecerá si hacemos clic en este botón otra vez. En el borrador, podemos mover tablas como deseemos si las arrastramos y las dejamos, vemos como las coordenadas cambian de acuerdo con los movimientos. La representación de las tablas que se ven en el borrador es tan sólo una estimación aproximada, comparado con el resultado final PDF.

<u>libros</u>	<u>autor</u>
ISBN	autor_id
Título	nombre_autor
número_pág	teléfono
autor_id	código país
idioma	
descripción	
foto-cubierta	
género	
fecha_publicación	<u>países</u>
	código país
	descripción



Cuando estemos satisfechos con la presentación, haremos clic en **Continuar** para guardarla.

Presentar una página

La última sección de la pantalla es un diálogo para generar un informe PDF. Esta sección también está disponible en la vista base de datos en la sub-página **Estructura**, ahora que hemos creado una definición de página:

Mostrar esquema PDF:

☐ Mostrar la cuadrícula

☒ Mostrar color

☐ Mostrar la dimensión de las tablas

☐ ¿Mostrar todas las tablas que tienen el mismo ancho?

☒ Diccionario de datos

 Formato del Diccionario de Datos Orientación horizontal ▾

 Tamaño del papel A4 ▾

Continuar

Las opciones disponibles son:

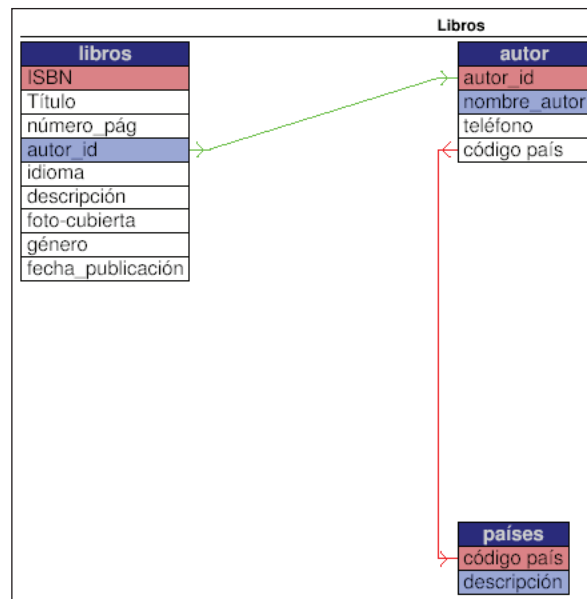
- **Mostrar la parrilla:** El esquema tendrá una presentación de parrilla, con las coordenadas.
- **Mostrar color:** Los enlaces entre las tablas, los nombres de las tablas y las columnas especiales (claves primarias y campos de presentación) estarán en color.
- **Mostrar las dimensiones de la tabla:** Esto muestra las dimensiones de cada tabla en el título de la tabla, por ejemplo 32x30.
- **¿Mostrar todas las tablas con la misma amplitud?** Normalmente, la amplitud se ajusta conforme el nombre de la tabla y de las columnas; la **Misma amplitud** usa la amplitud de la tabla más grande para el resto de las tablas. Con esto podemos mejorar el resultado visual.
- **Diccionario de datos:** Este es el diccionario de datos del que ya hemos hablado en este capítulo; esta configuración lo incluye al principio del informe.
- **Formato del diccionario de datos:** Aquí elegimos la orientación de la impresión del diccionario.
- **Tamaño del folio:** Cambiarlo influenciará el esquema y las dimensiones del borrador.

En `config.inc.php`, los siguientes parámetros definen los tamaños de papel disponibles y la elección por defecto:

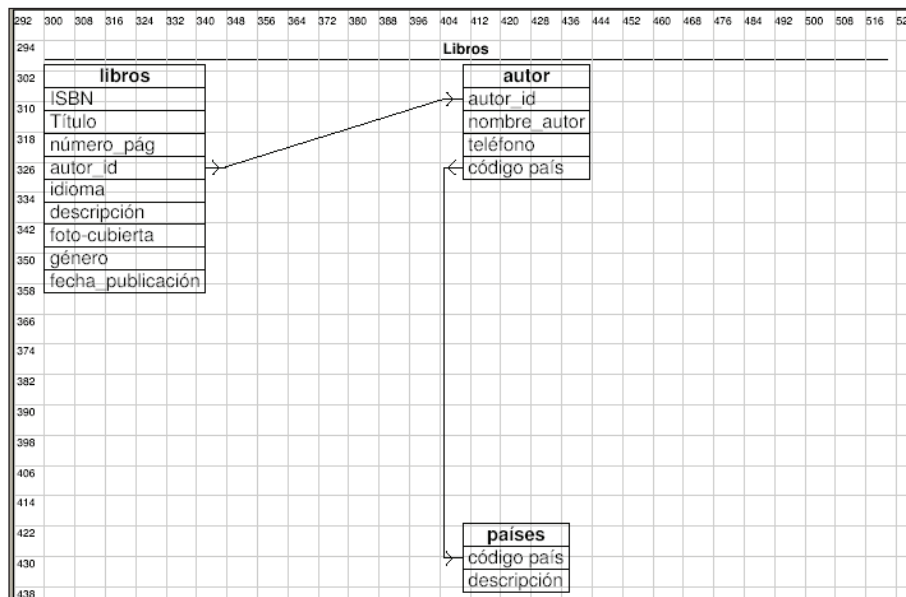
```
$cfg['PDFPageSizes'] = array('A3', 'A4', 'A5', 'letter', 'legal');  
$cfg['PDFDefaultPageSize'] = 'A4';
```

La siguiente pantalla muestra la última página del informe que se ha generado (la página del esquema) en formato PDF. Las primeras cuatro páginas tienen el diccionario de datos con una característica especial: En cada página se puede hacer clic en el nombre de la tabla para llegar al esquema, y en el esquema, se puede hacer clic en cada tabla para llegar a la página correspondiente del diccionario de datos.

Las flechas muestran la dirección de la tabla foránea correspondiente y la línea de relación usa un color específico para caracterizar cada relación. Si se ha seleccionado la opción **Mostrar color**, las claves primarias se muestran en rojo y los campos de presentación en azul:



Aquí hay otro ejemplo de la misma definición de página PDF **libros**, con la parrilla y sin colores:



Una nota sobre las fuentes que se usan

En el esquema PDF, todo el texto que vemos se dibuja con una fuente específica. phpMyAdmin intenta usar la primera fuente definida en el archivo mensaje de idioma actual (por ejemplo, `lang/english-iso-8859-1.inc.php`), conforme a la directiva de configuración `$right_font_family`, que contiene `'arial, helvetica, geneva, sans-serif'`. Sin embargo no hay soporte para todas las familias de fuentes.

Para generar el PDF, phpMyAdmin se basa en la librería `fpdf` (<http://www.fpdf.org>). Esta librería tiene dos maneras de usar las fuentes: **Integradas** y **no integradas**. Las fuentes **integradas** producen un archivo PDF más grande porque toda la fuente tiene que estar incluida en el PDF. Es por eso que la elección por defecto de phpMyAdmin es **no integrada**.

Para que esto funcione, la librería usa las fuentes TrueType presentes en el sistema operativo cliente y necesita una interfaz para estas fuentes. La librería está en los archivos de mapeado de fuentes, en el directorio `libraries/fpdf/font`.

La librería puede cambiar algunas fuentes pero en general deberíamos asegurarnos que el archivo necesario de mapeo esta presente. En ejemplo se sustitución: Para inglés, la primera fuente definida en `$right_font_family` es arial pero phpMyAdmin usa helvetica en su lugar. Los archivos estándar de mapeo de phpMyAdmin son `helvetica`, `courier`, `times` y `tahoma`.

Para añadir una fuente al archivo de mapeo, primero tenemos que añadirla a la librería (como se explica en el tutorial disponible en <http://www.fpdf.org>) y luego modificar el código fuente `pdf_schema.php` de phpMyAdmin.

Resumen

En este capítulo hemos cubierto las características de documentación que ofrece phpMyAdmin:

- Vista impresión para una base de datos o una tabla para una lista de tabla o columnas.
- Diccionario de datos para la lista completa de columnas.
- Esquema relacional PDF, incluyendo como crear y modificar una página de esquema PDF y usar el editor visual (borrador).

16

Transformaciones basadas en MIME

Aquí vamos a cubrir una característica muy poderosa de phpMyAdmin: Su habilidad para transformar los contenidos de las columnas respetando unas reglas específicas: Las transformaciones. Este capítulo describe como podemos transformar los contenidos que vemos en el modo **Examinar** de una tabla. Normalmente, los contenidos exactos de cada fila se muestran, excepto estos:

- TEXT y los caracteres de los campos pueden estar truncados, teniendo en cuenta `$cfg['LimitChars']` y si hemos hecho clic en el icono **Texto completo**.
- Los campos BLOB pueden ser reemplazados por un mensaje como por ejemplo **[BLOB - 1.5 KB]**.

Usaremos el término **celda** para indicar una columna específica de una fila específica. La celda que contiene la foto de la cubierta del libro 'Recuerdos futuros' (una columna BLOB) tiene unos datos crípticos como estos `%PNG\r\n\Z\n\0\0\0\rIHDR\` o un mensaje que muestra el tamaño del BLOB. Sería interesante ver un icono de la imagen directamente en phpMyAdmin, a ser posible la imagen en sí.

Definimos las **transformaciones** como un mecanismo mediante el cual todas las celdas de una columna se cambian cuando la examinamos usando los metadatos definidos en esta columna. Sólo las celdas visibles en la página de resultados se pueden transformar. La lógica de la transformación está codificada en los programas PHP, almacenados en `libraries/transformations`, y recuperados con el uso de arquitectura de enchufe (plug-in).

Para activar esta característica, debemos configurar `$cfg['BrowseMIME']` en `TRUE` en `config.inc.php`. El sistema relacional debe estar instalado (ver el capítulo 11) porque los metadatos necesarios para las transformaciones no están disponibles en la estructura de tablas oficial de MySQL; es un añadido que se ha hecho especialmente para `phpMyAdmin`.



En la sección de documentación del sitio Web de `phpMyAdmin`, hay un enlace que ofrece información adicional a los programadores que les gustaría saber más cosas sobre la estructura interna de los enchufes para poder codificar sus propias transformaciones.

Configuración de las columnas MIME

Si vamos a la vista tabla, a la sub-página **Estructura** de la tabla `libros` y hacemos clic en el enlace **Cambiar** de `foto_cubierta`, veremos tres atributos adicionales; como por ejemplo el tipo **Extensión Multifuncional de correo en Internet (MIME)**; del campo:

- Tipo MIME.
- Navegador transformación.
- Opciones de la transformación.

Comentarios	MIME-type	Transformación del navegador	Opciones de transformación ³

Es posible indicar a `phpMyAdmin` un **sólo** tipo de transformación para un campo específico. Aquí, como el campo `BLOB` puede almacenar cualquier tipo de datos, tenemos que informar al sistema de transformación del formato de los datos y los resultados que queremos para que `phpMyAdmin` interprete y actúe de forma correcta en los datos. Por consiguiente, tenemos que asegurarnos de cargar datos que siempre tengan el mismo formato de archivo.

Primero aprenderemos el propósito de estos atributos y luego intentaremos algunas posibilidades en la sección *Ejemplos de transformación*.

Tipo MIME

Las especificaciones MIME se han elegido como atributos metadatos para categorizar el tipo de datos que almacena la columna. Los valores actuales posibles son:

- Image/jpeg
- Image/png
- Text/plain

La opción **auto detectar** del menú aún se tiene que implementar así que no la usaremos.

Se puede elegir el tipo `text/plain` para una columna que contenga cualquier tipo de texto; por ejemplo texto XHTML o XML.

Transformaciones navegador

Aquí es donde configuramos las transformaciones exactas que se producirán. Puede ser que el tipo MIME no soporte más de una transformación; por ejemplo, para el tipo **image/jpeg** tenemos dos transformaciones disponibles: **Image/jpeg: Imagen en línea** para un icono de la imagen al que se puede hacer clic e **imagen/jpeg: Enlace** para mostrar un enlace.

Como podemos ver en la siguiente imagen, mover el ratón sobre cada elección del menú desplegable ofrece una pequeña explicación de la transformación correspondiente. Hay disponible una Explicación más completa de las transformaciones y las opciones posibles si hacemos clic en el enlace **descripciones de las transformaciones**.

Extra	Comentarios	MIME-type	Transformación del navegador	Opciones de transformación ³
			- application/octetstream: download - application/octetstream: hex image/jpeg: inline - image/jpeg: link	
			Muestra un thumbnail que puede abrirse mediante un clic; opciones: width, height en...	
			- text/plain: dateFormat - text/plain: external - text/plain: formatted - text/plain: imagelink - text/plain: link - text/plain: sql - text/plain: substr	

Opciones de las transformaciones

Veremos en la sección *Ejemplos de transformaciones* que algunas transformaciones aceptan opciones. Por ejemplo, una transformación que genera una imagen necesitará la altura y la anchura en píxeles. Se usa una coma para separar los valores en la lista de opciones, algunas opciones tienen que estar cerradas con comillas.

Algunas opciones tienen un valor por defecto y tenemos que tener cuidado en respetar el orden documentado de las opciones. Por ejemplo, si hay dos opciones y sólo queremos un valor específico de la segunda opción, podemos usar comillas vacías como marcador de posición para la primera opción y dejar que el sistema use su valor por defecto.

Requisitos para generar una imagen

La generación normal de iconos requiere que existan algunos componentes en el servidor Web y que el parámetro de `config.inc.php` esté configurado de manera correcta.

GD2

phpMyAdmin usa algunas funciones internas para generar iconos. Esas funciones necesitan que la librería GD2 esté presente en nuestro servidor PHP.

phpMyAdmin puede detectar la presencia de la librería GD2 correcta pero esta detección requiere tiempo y no se hace una vez por sesión sino *cada vez que se hace algo* en phpMyAdmin.

El valor por defecto del parámetro `$cfg['GD2Available']`, disponible en `config.inc.php`, 'auto' indica que se necesita una detección de la presencia de la librería y su versión.

Si sabemos que la librería GD2 esta disponible, podremos ejecutarla con más rapidez si configuramos `$cfg['GD2Available']` en yes. Si no está disponible, se recomienda poner un no aquí.

Librerías JPEG y PNG

Nuestro servidor PHP necesita tener soporte para las imágenes JPEG y PNG si nuestros usuarios quieren generar iconos para ese tipo de imágenes.

Memoria

En algunos servidores PHP, el valor por defecto de `memory_limit` en `php.ini` es 8 MB. Este valor es demasiado bajo para manipular correctamente imágenes. Por ejemplo, en un test, se necesitó un valor de 11 MB en `memory_limit` para generar el icono de un JPEG de 300 KB.

Ejemplos de transformaciones

Ahora discutiremos unos pocos ejemplos de transformación. Empezaremos por cambiar el tipo de campo del campo **foto_cubierta**.

Icono al que se puede hacer clic (.jpeg o .png)

Cambiamos nuestro campo `foto_cubierta` de `BLOB` a `MEDIUMBLOB` para asegurarnos que podemos cargar fotos con más tamaño que 65 KB. Luego escribimos los siguientes atributos:

MIME-type	Transformación del navegador	Opciones de transformación ³
image/jpeg	image/jpeg: inline	100,80

Aquí las opciones se presentan como **anchura**, **altura**. Si omitimos estas opciones, los valores por defecto son **100, 100**. El código de generación del icono preserva el coeficiente de aspecto original de la imagen, por eso los valores escritos son la anchura y la altura máximas de la imagen generada. Luego cargamos un archivo .jpeg en una celda (con las instrucciones del capítulo 6). Como resultado obtenemos lo siguiente en el modo **Examinar** de esta tabla:

			ISBN número de libro	Título	número_pág aproximado	autor_id ver tabla autor	idioma	descripción	foto-cubierta
☐	 Editar	 Borrar	1-234567-2...	Recuerdos ...	200	2	en		
☐	 Editar	 Borrar	1-234567-8...	Cien años...	600	1	en		[BLOB - 0 Bytes]
☐	 Editar	 Borrar	1-234567-9...	Cien años...	600	1	en		[BLOB - 0 Bytes]
 Marcar todos/as / Desmarcar todos Para los elementos que están marcados:   									

Se puede hacer clic en este icono para ver la foto a tamaño real.



Este icono no se almacena en ningún lugar de MySQL, se genera cada vez que entramos en el modo **Examinar** de este conjunto de filas. En un servidor Pentium-III 500 MHz, normalmente necesitamos un tiempo de generación de 0,5 a 1 segundo por imagen.

Para una imagen .png, tenemos que usar **image/png** como tipo MIME y **image/png: Inline** como transformación.

Otro cosa a tener en cuenta: El parámetro `$cfg['ShowBlob']` no influye en la presentación del icono; puede estar configurado en TRUE o en FALSE.

Enlace a una imagen

Para tener un enlace sin los iconos usamos la transformación **image/jpeg: Link**. No hay opciones de transformación. Este enlace se puede usar para ver o descargar la foto (si hacemos clic con el botón derecho al enlace):

→ ↑ ←			ISBN número de libro	Título	número_pág aproximado	autor_id ver tabla autor	idioma	descripción	foto-cubierta
☐	 Editar	 Borrar	1-234567-22-0	Recuerdos futuros	200	2	en		[BLOB]
☐	 Editar	 Borrar	1-234567-89_0	Cien años de cine (volumen 1)	600	1	en		[BLOB]
☐	 Editar	 Borrar	1-234567-90-0	Cien años de cine (volumen 2)	600	1	en		[BLOB]
↑ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:   									

Formatear datos

Tenemos un campo llamado `fecha_publicación` en nuestra tabla libros; asegurémonos que su tipo es `DATETIME`. Luego configuramos este tipo MIME a **text/plain** y la transformación del navegador **text/plain: Dateformat**. El siguiente paso es editar la fila 'Recuerdos futuros' y escribir 01/01/2003 14:56:00 en el campo **fecha_publicación**. Luego examinamos la tabla y vemos el campo formateado. Si movemos el ratón sobre el campo veremos los contenidos originales sin formato:

fecha_publicación
01-01-2003 a las 14:56:00
2003-01-01 14:56:00

Esta transformación acepta dos opciones. La primera es el número de horas que se añadirán al valor original (por defecto, es cero). Añadir un número de horas puede ser útil si almacenamos todas las horas basado en el **Hora universal coordinada (UCT)** pero queremos verlos para una zona específica (UCT+5). La segunda opción

es el formato en el que queremos ver el tiempo, se hace desde cualquier parámetro `strftime` de PHP. De modo que si ponemos esto en las opciones de transformación '0', 'Year: %Y', obtendremos:



Enlaces de texto

Supongamos que en nuestra tabla **libros**, tenemos que poner dentro del campo de descripción una URL completa: `http://domain.com/abc.pdf`. El enlace del texto se mostraría cuando examinásemos la tabla pero no podríamos hacer clic en él. Ahora veremos el uso del tipo MIME **text/plain** en esta situación.

text/plain: link

En el escenario que acabamos de mencionar, si usamos un tipo MIME **text/plain** y una transformación navegador **text/plain: link**, todavía podemos ver el texto del enlace pero podremos hacer clic en él:

→T←			ISBN número de libro	Título	número_pág aproximado	autor_id ver tabla autor	idioma	descripción
☐	Editar	Borrar	1-234567-22-0	Recuerdos futuros	200	2	en	http://domain.abc/pdf

Si todos los documentos que queremos marcar están en un prefijo URL común, podemos poner este prefijo (por ejemplo, `http://domain.com/`) en la primera opción de transformación con comillas. Luego, en cada celda, solo pondremos la última parte de la URL (`abc.pdf`).

Hay disponible una segunda opción de transformación para configurar el título. Se mostraría en el modo **Examinar** en lugar de los contenidos de la URL, pero con un clic llegaríamos de todos modos a la URL.



Si solo usamos la segunda opción de transformación, tenemos que poner comillas donde se escribe la primera opción, tal y como se ve aquí: ' ', 'esto es el título'.

text/plain: enlace de imagen

Esta transformación es parecida a la anterior, excepto que en la celda ponemos una URL que lleva a una imagen. Esta imagen se extraerá y se mostrará en la celda junto con el texto enlace. A pesar que aquí la imagen está almacenada en el servidor local, podría estar en cualquier lado de la Web.



La primera opción disponible es el prefijo común URL (como en **text/plain: link**), la segunda opción es la anchura de la imagen en píxeles (por defecto: 100) y la tercera es la altura (por defecto: 50).

Como el enlace puede referirse a cualquier tipo de imagen soportada por el navegador, que no tiene porque estar cubierto por el mecanismo de generación de iconos de phpMyAdmin, se ajusta el tamaño de la imagen según las opciones. Para ver la imagen original, podemos hacer clic en el enlace.

Preservar el formato original

Normalmente, cuando muestra el texto, phpMyAdmin escapa algunos caracteres especiales. Por ejemplo, si escribimos **Este libro es bueno ** en el campo de descripción de un libro, normalmente veremos **Este libro es bueno ** cuando examinemos la tabla. Si usamos la transformación **text/plain: formatted** para este campo, veremos lo siguiente cuando examinemos:

→ T ←			ISBN número de libro	Título	número_pág aproximado	autor_id ver tabla autor	idioma	descripción
☐	 Editar	 Borrar	1-234567-22-0	Recuerdos futuros	200	2	en	Este libro es bueno

En este ejemplo, los resultados son correctos. Sin embargo, otros datos HTML escritos en el campo de datos pueden producir resultados sorprendentes (incluyendo páginas HTML inválidas). Por ejemplo, ya que phpMyAdmin presenta los resultados con tablas HTML, una tabla `</table>` sin escapar en el campo de datos arruinaría los resultados.

Presentar partes de un texto

La transformación **text/plain: substr** está disponible para mostrar sólo una parte del texto. Aquí están las opciones:

- Primera: Donde empieza en el texto (por defecto: 0)
- Segunda: El número de caracteres: (por defecto: Todo el texto que queda).
- Tercera: Que se mostrará como sufijo para mostrar que se ha truncado el texto (por defecto: ...)

Recuerde que `$cfg['LimitChars']` trunca los caracteres de todos los campos no numéricos de modo que **text/plain: substr** es un mecanismo para afinarlo campo por campo.

Aplicaciones externas

Las transformaciones que hemos descrito anteriormente se implementan directamente en phpMyAdmin. Sin embargo, algunas transformaciones se hacen mejor con aplicaciones externas existentes.

La transformación **text/plain: external** nos permite mandar datos de las celdas a otra aplicación que se iniciará en nuestro servidor Web, capturar los resultados de esta aplicación y mostrarlo en la posición de la celda.

Esta característica sólo esta soportada por un servidor Linux o UNIX (bajo Microsoft Windows, el proceso PHP no puede capturar con facilidad el resultado y la redirección). Además, PHP no debería funcionar en modo seguro, de lo contrario esta característica puede que no esté disponible en algunos servidores proveídos.

Por razones de seguridad, la ruta exacta de la aplicación y su nombre no se pueden configurar en phpMyAdmin como una opción de transformación. El nombre de la aplicación se configura directamente en uno de los programas de phpMyAdmin.

Primero, en el directorio de instalación de phpMyAdmin, editamos el archivo `text_plain_external.inc.php` en `libraries/tranformations/` y encontramos la siguiente sección.

```
$allowed_programs = array ();
$allowed_programs[0] = '/usr/local/bin/tidy';
$allowed_programs[1] = '/usr/local/bin/validate';
```



Los nombres de los programas de transformación se construyen con el siguiente formato: El tipo MIME, subrayado doble y luego una parte que indique *que* transformación ocurre.

Cada programa permitido se tiene que describir aquí, con un número de índice que empieza de 0 y su ruta completa. Luego guardamos las modificaciones de este programa y las ponemos de nuevo en el servidor si es necesario. La configuración que queda se completa en el mismo panel donde hemos elegido las opciones de transformación del resto de transformaciones navegador.

Por supuesto elegimos **text/plain: external** en el menú transformaciones.

Como primera opción, escribimos el número de aplicación (por ejemplo, 0 sería para la aplicación tidy). La segunda opción tiene los parámetros que necesitamos para que la aplicación funcione. Si queremos que phpMyAdmin aplique la función `htmlspecialchars()` a los resultados, ponemos un **1** como el tercer parámetro (se hace por defecto). Podríamos poner **0** para evitar proteger el resultado con `htmlspecialchars()`.

Si queremos evitar reformatar las líneas de la celda, ponemos **1** como cuarto parámetro. Con esto se usará el modificador `NOWRAP`, se hace por defecto.

Ejemplo de aplicaciones externas: Clasificación en la celda

Este ejemplo muestra como podemos clasificar los contenidos textuales de una celda. Empezamos modificando el programa `text_plain__external.inc.php` como se ha mencionado en la sección anterior, y añadimos el programa `sort`:

```
$allowed_programs[2] = '/bin/sort';
```

Tenga en cuenta que nuestro programa tiene el número de índice 2.

Primero añadimos un campo **TEXT** a nuestra tabla **libros**: El campo **palabras clave** y escribimos la información relacionada con MIME:

Transformación del navegador	Opciones de transformación ³
text/plain: external	'2','f'

El '2' se refiere al número de índice de `sort` y el parámetro `'-f'` es para `sort`, lo que hace que el programa clasifique los caracteres mayúsculas y minúsculas *juntos* en lugar de clasificar las mayúsculas *antes* que las minúsculas.

Luego hacemos clic en **Editar** la fila del libro *Cien años de cine (volumen 1)*, escribimos algunas palabras clave en ningún orden particular y hacemos clic en **Continuar y grabar**:

palabras_clave	text		historia cine Smith
----------------	------	----------------------	---------------------------

Para verificar los efectos del programa externo, examinamos nuestra tabla y vemos las palabras clave clasificadas en la celda:

número_pág aproximado	autor_id ver tabla autor	idioma	descripción	foto-cubierta	género	fecha_publicación	palabras_clave
200	2	en	 http://localhost/phpMyAdmin-2.8.0.4/themes/original/img/logo_left.png	[BLOB]	F	Year: 2003	cine historia Smith

Por supuesto, las palabras clave se ven clasificadas en esta celda, y la opción -f ha puesto la mayúscula S de **Smith** en el orden correcto.

Resumen

En este capítulo hemos visto como podemos mejorar la navegación si transformamos datos con diferentes métodos. Podemos:

- Ver un icono e imágenes a tamaño real de campos BLOB . jpeg y . png.
- Generar enlaces y formatos de fechas.
- Ver sólo partes de textos.
- Ejecutar programas externos para reformatear los contenidos de cada celda.

17

Conjunto de caracteres y colaciones

Este capítulo explica como phpMyAdmin almacenar y recoge nuestros datos, como trabaja con las características conjunto de caracteres y colación que están disponibles en MySQL. El comportamiento del programa depende de la versión de MySQL que se usa.

Un **conjunto de caracteres** describe como se codifican los símbolos de una lengua o dialecto específicos. Una colación contiene las reglas para comparar los caracteres del conjunto de caracteres; tendrá más información sobre esto en la sección *MySQL 4.1.x y superiores* de este capítulo.

El conjunto de caracteres que se usa para *almacenar* nuestros datos puede no ser el mismo que se usa para *mostrarlo*, por lo que habrá discrepancias en los datos. De modo, necesitamos transformar los datos.

Archivos de idioma y UTF-8

UTF-8 es un esquema de codificar para representar caracteres de Unicode (vea <http://www.unicode.org>). Unicode en estos momentos soporta más de 600 idiomas, lo cual es el principal beneficio para usarlo en lugar de otros conjuntos de caracteres de ISO o de Windows, especialmente en un producto multi idioma como phpMyAdmin.

Tenga en cuenta que el navegador tiene que soportar UTF_8 (la mayoría de los navegadores actuales lo hacen). El kit de distribución de phpMyAdmin incluye una versión UTF_8 de casi todos los archivos de idioma del subdirectorio `lang`, algunas sólo están disponibles en UTF-8.

Algunos archivos también están codificados con conjuntos de caracteres ISO o Windows, con el objetivo de soportar navegadores antiguos o porque no tenemos una versión convertida a UTF_8 de ese idioma en ese momento.

La posibilidad de elegir un archivo idioma UTF-8 en el **selector de idioma** depende de la versión phpMyAdmin que se usa, la versión MySQL y; antes de phpMyAdmin 2.6.0; de algunas configuraciones de `config.inc.php`.

Versiones de MySQL anteriores a 4.1.x

phpMyAdmin no puede basarse en MySQL, en las versiones anteriores a MySQL 4.1.x, para transformar los datos conforme a un conjunto de caracteres porque esta característica no existe en esas versiones de MySQL. De modo, que la codificación la hace phpMyAdmin directamente, antes de mandar los datos al servidor MySQL y cuando los recibe.

Codificación de datos

El servidor PHP tiene que soportar los módulos `iconv` o `recode` para codificar datos. Si este no es el caso, se generara el siguiente mensaje de error:

No se puede cargar la extensión `iconv` o `recode` necesaria para la conversión charset, configure php para permitir el uso de estas extensiones o desactive la conversión charset en phpMyAdmin.

Si se muestra este mensaje, consulte la documentación de su sistema (PHP o el sistema operativo) para ver como se instalan las extensiones.

Antes de la versión 2.6.0 de phpMyAdmin, el archivo `config.inc.php` por defecto no usaba la codificación UTF-8: El parámetro `$cfg['AllowAnywhereRecoding']` estaba configurado en `FALSE`. Así pues, no se ofrecían idiomas UTF-8 en el **selector de idioma**. Para activarlo, solo teníamos que cambiar el parámetro a `TRUE`.

En phpMyAdmin 2.6.0, el parámetro todavía está configurado `FALSE` por defecto, pero las elecciones UTF-8 se presentan en el **selector de idiomas**. Esta situación puede dar problemas de codificación; vea la sección *El impacto de cambiar* de este capítulo.

Otro parámetro `$cfg['RecodingEngine']`, especifica el motor de codificación que se usa (las elecciones son `auto`, `iconv` o `recode`), pero podemos dejarlo en `auto` a no ser que prefiramos especificar otro. Con `auto`, phpMyAdmin primero prueba con el módulo `iconv` y luego con el módulo `recode`.

Conjunto de caracteres

Cuando consideremos las opciones disponibles, debemos entender que, cuando está conectado a un servidor anterior a MySQL 4.1.x, phpMyAdmin tiene un soporte limitado para convertir conjuntos de caracteres. Ahora podemos especificar que conjunto de caracteres se aplica a una consulta y sus resultados. El conjunto de caracteres que se usa por defecto está definido en el siguiente parámetro.

```
$cfg['DefaultCharset'] = 'iso-8859-1';
```

Esta es la única elección por defecto; los usuarios pueden elegir otro conjunto de caracteres de la lista de elecciones de este parámetro (el parámetro actual de `config.inc.php` tiene más):

```
$cfg['AvailableCharsets'] = array(
    'iso-8859-1'
    'iso-8859-2'      'iso-8859-3'
    'iso-8859-4');;
```

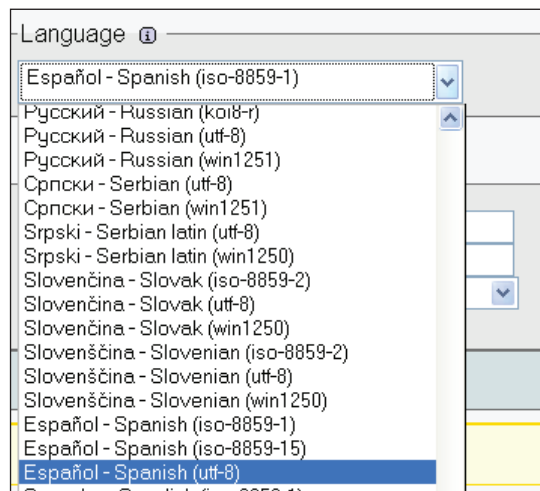
Estas elecciones se muestran a los usuarios en el mismo orden con el que las hemos definido en el parámetro `$cfg['AvailableCharsets']`. De modo que podemos poner las elecciones más populares arriba. Se puede usar cualquier carácter soportado por los motores de recodificación `iconv` o `recode`.

Digamos que estamos usando phpMyAdmin 2.6.0 pero que `$cfg['AllowAnywhereRencoding']` se ha dejado en su valor por defecto `FALSE`. Eso significa que en la página de inicio vemos:

No hay un selector **MySQL Charset**. El conjunto de caracteres definido en `$cfg['DefaultCharset']` se usará para comunicarnos con MySQL.

Elegir el conjunto de caracteres efectivo

Ahora configuramos `$cfg['AllowAnywhereRecoding']` TRUE. Luego elegimos español (es.utf-8) en el **selector de idiomas**. La página de **inicio** ha cambiado:



La opción **Juego de caracteres MySQL** aparece sólo si el **idioma** elegido usa una codificación UTF-8. A partir de ahora, en cualquier comunicación que haya entre el servidor Web y el servidor MySQL se usará este conjunto de caracteres MySQL.



Esta elección de conjunto de caracteres se recuerda, durante treinta días, con un mecanismo de cookies. Así pues, tendremos que elegir otra vez el conjunto de caracteres, dependiendo del lugar donde están almacenadas las cookies, el ordenador local o el servidor de Internet, si nos conectamos a phpMyAdmin desde otro ordenador.

El impacto de cambiar

Cuando elegimos un conjunto de caracteres, todos los datos almacenados en MySQL se recodificarán con este conjunto de caracteres, de modo que obtendremos resultados incorrectos cuando recuperemos los datos si cambiamos el conjunto de caracteres que usa phpMyAdmin. No hay una manera fácil de encontrar *qué* conjunto de caracteres se usó para almacenar una fila particular de datos.

Aquí hay un ejemplo con nuestra tabla **autor**. Primero creamos un autor nuevo con el carácter é en su nombre:

Campo	Tipo	Función	Nulo	Valor
autor_id	int(11)		☐	3
nombre_autor	varchar(30)		☐	André Smith
teléfono	varchar(30)		☐	888-9999
código país	char(2)		☐	ca - Canadá

Aquí no hay problemas para insertar, navegar o buscar este autor nuevo ya que el conjunto de caracteres elegido `iso-8859-1` puede tratar el carácter `é`.

Supongamos que más tarde cambiamos (en la página de **inicio**) el **Juego de caracteres MySQL** a `utf-8`, tendremos un problema al navegar por la tabla autor:

→T←			autor_id	nombre_autor	teléfono	código país
☐	 Editar	 Borrar	1	John Smith	111-1111	
☐	 Editar	 Borrar	2	Maria Sunshine	222-2222	
☐	 Editar	 Borrar	3	André Smith	888-9999	ca

El mismo problema ocurre, en `phpMyAdmin 2.6.0`, si cambiamos un conjunto de caracteres a otro en la página **inicio**, cuando nuestro `$cfg['AllowAnywhereRecoding']` está configurado en `FALSE`. De modo que se recomienda no cambiar de conjunto de caracteres si nuestro sistema no tiene configurada la conversión necesaria.

Importar y exportar con conjuntos de caracteres

Si `$cfg['AllowAnywhereRecoding']` está configurado `TRUE`, entonces en las páginas en las que hay un recuadro de consulta, el diálogo de **Posición del archivo texto** se modifica para añadir la elección del conjunto de caracteres que se va a importar:

En diálogo **Exportar** también podemos elegir el conjunto de caracteres del archivo que hemos creado:

MySQL 4.1.x y superior

Desde la versión MySQL 4.1.x, el servidor MySQL recodifica los caracteres por nosotros. Además se han tomado medidas en MySQL para que éste pueda indicar el conjunto de caracteres y la colación de cada base de datos, tabla e incluso campo. Un conjunto de caracteres por defecto para una base de datos se aplica a cada una de sus tablas, a no ser que en el nivel de la tabla se ignore; el mismo principio se aplica a todos los campos.

En la versión actual de phpMyAdmin (2.8.0.4), el soporte para las características conjuntos de caracteres y colación de MySQL 4.1.x ya no es experimental como en las versiones previas.

El parámetro `$cfg['AllowAnywhereRecoding']` no tiene impacto en la versión MySQL 4.1.x o superiores.

Colaciones

Cuando se tienen que hacer comparaciones entre caracteres el sistema (MySQL en este caso) tiene que seguir unas reglas precisas. Por ejemplo, ¿'A' es equivalente a 'a'? ¿Es 'André' equivalente a 'Andre'? Estas reglas se llaman colaciones.

Elegir una buena colación es importante para obtener los resultados que queremos cuando buscamos datos, por ejemplo desde la página **Buscar** de phpMyAdmin.

Para una introducción a las colaciones vaya a <http://dev.mysql.com/doc/mysql/en/Charset-general.htm>, y para una explicación más mecánica de los algoritmos que se usan vaya a <http://www.unicode.org/reports/tr10>.

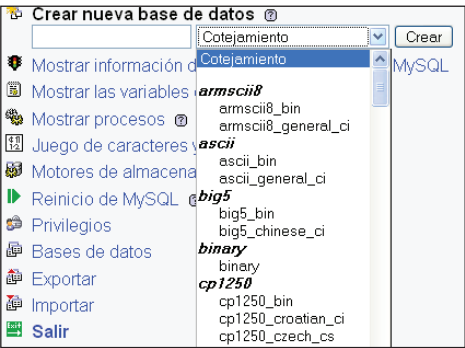
Página de inicio

Así es como aparece la página de **inicio** cuando nos conectamos a un servidor MySQL 4.1.x (las secciones que siguen muestran los cambios en detalle):



Crear una base de datos

Cuando creamos una base de datos, elegimos su conjunto de caracteres y su colación por defecto con el diálogo **Colación**. Esta configuración se puede cambiar más tarde (vaya a la sección *Vista base de datos*).



Conjuntos de caracteres y colaciones disponibles

Desde la página de inicio, el enlace conjunto de **caracteres y colaciones** abre la vista servidor en la sub-página **Charsets**, en la que se ve una lista de que conjuntos de caracteres y colaciones soporta el servidor MySQL. La colación por defecto de cada conjunto de caracteres se muestra en color de fondo diferente (usar el marcado de filas en color definido en `$cfg[ 'BrowseMarkerColor' ]`):

Servidor: localhost	
Bases de datos	SQL
Estado actual	Variables
Juegos de caracteres	
Juego de caracteres y sus cotejamientos	
Cotejamiento	Descripción
armSCII8 (ARMSCII-8 Armenian)	
armSCII8_bin	Armenio, Binario
armSCII8_general_ci	Armenio, independiente de mayúsculas y minúsculas
ascii (US ASCII)	
ascii_bin	Europea occidental (multilingüe), Binario
ascii_general_ci	Europea occidental (multilingüe), independiente de mayúsculas y minúsculas
big5 (Big5 Traditional Chinese)	
big5_bin	Chino tradicional, Binario
big5_chinese_ci	Chino tradicional, independiente de mayúsculas y minúsculas
binary (Binary pseudo charset)	
binary	Binario
cp1250 (Windows Central European)	
cp1250_bin	Europeo central (multilingüe), Binario
cp1250_croatian_ci	Croata, independiente de mayúsculas y minúsculas
cp1250_czech_cs	Checo, dependiente de mayúsculas y minúsculas
cp1250_general_ci	Europeo central (multilingüe), independiente de mayúsculas y minúsculas
cp1251 (Windows Cyrillic)	

Conjunto de caracteres y colación efectivos

phpMyAdmin elige el conjunto de caracteres 'efectivo'; que es el que queremos ver en nuestro navegador; el que mejor se ajusta al idioma que hemos elegido. De modo que por ejemplo, en la página de **inicio** vemos:

MySQL - 4.1.19-community-nt	phpMyAdmin - 2.8.0.4
Servidor: localhost	▶ Versión del cliente: 4.1.19 ▶ Extensiones PHP utilizadas: mysqli
▶ Versión del protocolo: 10 Servidor: localhost via TCP/IP ▶ Usuario: root@localhost	Language : Español - Spanish
Juegos de caracteres de MySQL: UTF-8 Unicode (utf8)	Tema / Estilo: Original
Cotejamiento de las conexiones MySQL: utf8_unicode_ci	Documentación de phpMyAdmin Página oficial de phpMyAdmin ▶ [ChangeLog] [CVS] [Lists]

La información del conjunto de caracteres (como lo ve el charset de MySQL) pasa al servidor MySQL, le pide a MySQL que transforme los caracteres que manda nuestro navegador a este conjunto de caracteres y que interprete lo que recibe desde el navegador. Recuerde que todas las tablas y campos tienen información conjunto de caracteres que describe como se codifican los datos.

También podemos elegir que conjunto de caracteres y colación se usaran en nuestra conexión al servidor MySQL con el diálogo **colación de conexión**. Normalmente, el valor por defecto debería funcionar, pero si escribimos algunos caracteres usando un conjunto de caracteres diferente, podemos elegir el conjunto de caracteres apropiado en este diálogo.



Si un carácter no existe en el idioma elegido, MySQL Manda un interrogante para sustituirlo; por ejemplo, el nombre del autor **André Smith** y el idioma elegido es **búlgaro (bg-win1251)**, que no tiene un carácter é.



Vista base de datos

En la vista base de datos, también podemos cambiar el conjunto de caracteres por defecto de una base de datos:

 Comentario de la base de datos:

 Cambiar el nombre de la base de datos a:

 Copiar la base de datos a:

- ☐ Únicamente la estructura
- ☒ Estructura y datos
- ☐ Solamente datos
- ☒ CREAR BASE DE DATOS antes de copiar
- ☐ Añadir DROP TABLE
- ☐ Añadir el valor AUTO_INCREMENT
- ☐ Añadir restricciones
- ☐ Seleccionar la base de datos copiada

 Cotejamiento:

En la página **Estructura** de la base de datos, podemos ver la colación usada para cada tabla:

Servidor: localhost - Base de datos: dbbook

[Estructura](#)
[SQL](#)
[Buscar](#)
[Generar una consulta](#)
[Exportar](#)
[Importar](#)
[Operaciones](#)
[Privilegios](#)
[Eliminar](#)

Tabla	Acción						Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
☐ autor							3	MyISAM	latin1_swedish_ci	3.1 KB	-
☐ libros							3	MyISAM	latin1_swedish_ci	384.7 KB	-
☐ paises							2	MyISAM	latin1_swedish_ci	2.0 KB	-
☐ pma_bookmark							4	MyISAM	latin1_swedish_ci	2.6 KB	128 Bytes
☐ pma_column_info							3	MyISAM	latin1_swedish_ci	3.3 KB	-
☐ pma_history							0	MyISAM	latin1_swedish_ci	1.0 KB	-
☐ pma_pdf_pages							2	MyISAM	latin1_swedish_ci	3.0 KB	-
☐ pma_relation							2	MyISAM	latin1_swedish_ci	3.1 KB	-
☐ pma_table_coords							6	MyISAM	latin1_swedish_ci	2.2 KB	-
☐ pma_table_info							3	MyISAM	latin1_swedish_ci	2.1 KB	-
10 tabla(s)	Número de filas						28	InnoDB	latin1_swedish_ci	407.1 KB	128 Bytes

[Marcar todos/as](#) / [Desmarcar todos](#) / [Marcar las tablas con residuo a depurar](#)

Vista tabla

En la vista tabla, en la sub-página **Operación**, podemos cambiar la información sobre conjunto de caracteres y la colación por defecto (vea la muestra de pantalla que sigue):

Opciones de la tabla

Cambiar el nombre de la tabla a:

Comentarios de la tabla:

Motor de almacenamiento:

Cotejamiento:

pack_keys: ☐

checksum: ☐

delay_key_write: ☐

En sub-página **Estructura** de la vista tabla también podemos configurar el conjunto de caracteres de cada columna:

Campo	Tipo	Longitud/Valores ¹	Cotejamiento
campo1	VARCHAR	30	latin1_swedish_ci

¹ Si el tipo de campo es "enum" o "set", por favor ingrese los valores entre comillas (" "). Si alguna vez necesita poner una barra invertida ("\"), use una barra invertida doble ("\\").

² Para valores predeterminados, por favor ingrese los valores entre comillas (" ").

³ Por favor ingrese los valores para las opciones de transformación de caracteres entre comillas (" "). Si alguna vez requiere insertar un Backslash ("\"), use una barra invertida doble ("\\").

Para una lista de opciones de transformación disponibles, consulte el manual de MySQL.

armSCII8

armSCII8_bin

armSCII8_general_ci

ascii

ascii_bin

ascii_general_ci

big5

big5_bin

big5_chinese_ci

binary

binary

cp1250

cp1250_bin

cp1250_croatian_ci

cp1250_czech_cs

cp1250_general_ci

cp1251

cp1251_bin

cp1251_bulgarian_ci

Importar y exportar con un conjunto de caracteres

En los resultados de exportación podemos ver la información DEFAULT CHARSET y COLLATE para la tabla y sus columnas.

```
CREATE TABLE 'autor' (
  'autor_id' int(11) NOT NULL default '0',
  'nombre_autor' varchar(30) collate latin1_general_ci NOT NULL
  default '',
```

```
'teléfono' varchar(30) collate latin1_general_ci default NULL,
'código_país' char(2) collate latin1_general_ci NOT NULL default
'',
PRIMARY KEY ('autor_id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
```

Vista servidor

En la vista Servidor, podemos obtener estadísticas sobre las bases de datos (vaya a la sección *Información de las bases de datos*, del capítulo 18). Cuando nuestro servidor soporta conjuntos de caracteres y colaciones, vemos una columna de información adicional, **Colación**, como en esta página:

Servidor: localhost

[Bases de datos](#)
[SQL](#)
[Estado actual](#)
[Variables](#)
[Juegos de caracteres](#)
[Motores](#)
[Privilegios](#)
[Procesos](#)

[Exportar](#)
[Importar](#)

Estadísticas de la base

	Base de datos	Cotejamiento	Tablas	Filas	Datos	Índices	Total	Residuo a depurar	
☐	db1	latin1_swedish_ci	3	0	48.0 KB	0 Bytes	48.0 KB	0 Bytes	
☐	dbbook	latin1_swedish_ci	10	28	382.1 KB	25.0 KB	407.1 KB	128 Bytes	
☐	mysql	latin1_swedish_ci	16	1,457	209.9 KB	59.0 KB	268.9 KB	866 Bytes	
	Total: 3	latin1_swedish_ci	29	1,485	640.0 KB	84.0 KB	724.0 KB	994 Bytes	

Soporte Kanji

Si phpMyAdmin detecta el uso de japonés, chequea si PHP soporta la función secuencias multi-byte `mb_convert_encoding()`. Si están disponibles, se presentan botones de radio adicionales para elegir entre `EUC-JP` y codificaciones `SJIS` japonesas en algunas páginas:

- Exportar
- Insertar texto de un archivo
- Recuadro consulta

Resumen

En este capítulo hemos cubierto:

- El uso de los archivos de idiomas de phpMyAdmin, incluyendo UTF-8.
- La manera en que phpMyAdmin tiene que recodificar nuestros datos en MySQL antes de su versión 4.1.x.
- El impacto de cambiar un conjunto de caracteres por otro.
- Las características conjunto de caracteres y colación de MySQL desde la versión 4.1.x

18

Administración del servidor MySQL con phpMyAdmin

En este capítulo discutiremos como un administrador del sistema puede usar las características de administración del servidor phpMyAdmin para mantener las cuentas de usuario día a día, para verificar el servidor y para protegerlo.

Entrar en la vista servidor

Se puede acceder a la vista Servidor desde la página de **inicio** si elegimos uno de los siguientes enlaces:

- **Mostrar información de tiempo de ejecución de MySQL**
- **Mostrar las variables del sistema MySQL**
- **Mostrar procesos**
- **Privilegios**
- **Bases de datos**
- **Exportar**

Los enlaces superiores sólo son visibles si estamos conectados como usuario privilegiado. Cuando estamos en la vista servidor, vemos un menú del resto de sub-páginas relacionadas con el servidor.

Administración de los usuarios y los privilegios

La sub-página **Privilegios** de la vista servidor contiene los diálogos para administrar las cuentas de usuario MySQL, sus privilegios globales, a nivel de base de datos y de tabla. Esta sub-página está centrada en el usuario y es jerárquica: Por ejemplo, cuando editamos los privilegios del usuario, podemos ver sus privilegios globales además de sus privilegios específicos para las bases de datos. Podemos ir más abajo (en la base de datos) para ver los privilegios específicos de las tablas para *esta* combinación usuario-base de datos.

Información general del usuario

La primera página que aparece cuando entramos en la sub-página Privilegios es la **información general del usuario**. Esto muestra todas las cuentas de los usuarios y un resumen de sus privilegios globales:

Servidor: localhost

Bases de datos SQL Estado actual Variables Juegos de caracteres Motores Privilegios Procesos Exportar

Importar

Vista global de usuarios

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [Mostrar todo]

	Usuario	Servidor	Contraseña	Privilegios globales	Conceder
☐	cualquiera	localhost	No	USAGE	Sí
☐	marc	%	Sí	USAGE	Sí
☐	pepito	localhost	Sí	USAGE	No
☐	pma	localhost	No	USAGE	No
☐	root	localhost	Sí	ALL PRIVILEGES	Sí

↑ Marcar todos/as / Desmarcar todos

Agregar un nuevo usuario

Eliminar a los usuarios seleccionados

(Revocar todos los privilegios activos de los usuarios y borrarlos después.)

☐ Eliminar las bases de datos que tienen los mismos nombres que los usuarios.

Continuar

Nota: phpMyAdmin obtiene los privilegios de los usuarios 'directamente de las tablas de privilegios MySQL'. El contenido de estas tablas puede diferir de los privilegios que usa el servidor si es que se han realizado cambios manuales en él. En este caso, nuevamente deberá cargar la página de privilegios antes de continuar.

Desde esta página podemos:

- Editar los privilegios de un usuario con el enlace **Editar** en la línea de un usuario.
- Usar los recuadros para suprimir usuarios con el diálogo **Suprimir los usuarios elegidos**.
- Acceder a la página donde está el diálogo **Añadir usuario**.

- Recargar los privilegios. Los privilegios efectivos (en los que el servidor basa sus decisiones de acceso) están en la memoria del servidor. Las modificaciones a los privilegios que se han creado en la interfaz de phpMyAdmin se hacen en la memoria y el disco, en la base de datos **mysql**. Las modificaciones que se hacen directamente en la base de datos **mysql** no tiene efecto inmediato. La operación **Recargar** lee los privilegios de los usuarios desde la base de datos y los hace efectivos en la memoria.

La lista de usuarios que se muestra tiene las siguientes características:

- **Usuario:** Los usuarios en orden alfabético.
- **Proveedor:** Aparece una definición del proveedor de ese usuario, el valor % indica todos los proveedores. Es el nombre de la máquina o la dirección IP desde la que el usuario se conecta al servidor MySQL.
- **Contraseña:** Tiene **Yes** si hay una contraseña definida y **No** si no la hay. El valor de la contraseña no se puede ver en la interfaz de phpMyAdmin o directamente mirando a la tabla `mysql.user`, ya que la contraseña está ofuscada por un algoritmo de cálculo de una dirección.
- **Privilegios globales:** Se ve una lista de ellos para cada usuario.
- **Otorgar:** Contiene **Yes** si el usuario puede otorgar sus privilegios a otros.

Añadir un usuario

El enlace **Añadir un usuario nuevo** muestra un diálogo para crear una cuenta de usuario:

 **Agregar un nuevo usuario**

Información de la cuenta

Nombre de usuario:

Servidor:

Contraseña:

Debe volver a escribir:

Generar la contraseña:

Privilegios globales ([Marcar todos/as](#) / [Desmarcar todos](#))

Nota: Los nombres de los privilegios de MySQL están expresados en inglés

Datos

- ☐ SELECT
- ☐ INSERT
- ☐ UPDATE
- ☐ DELETE
- ☐ FILE

Estructura

- ☐ CREATE
- ☐ ALTER
- ☐ INDEX
- ☐ DROP
- ☐ CREATE TEMPORARY TABLES
- ☐ EXECUTE

Administración

- ☐ GRANT
- ☐ SUPER
- ☐ PROCESS
- ☐ RELOAD
- ☐ SHUTDOWN
- ☐ SHOW DATABASES
- ☐ LOCK TABLES
- ☐ REFERENCES
- ☐ REPLICATION CLIENT
- ☐ REPLICATION SLAVE

Limites de recursos

Nota: si cambia los parámetros de estas opciones a 0 (cero), remueve el límite.

MAX QUERIES PER HOUR

MAX UPDATES PER HOUR

MAX CONNECTIONS PER HOUR

Nombre del usuario

El menú **nombre de usuario** ofrece dos elecciones. **Usar el campo texto** nos permite escribir el nombre del nuevo usuario en un recuadro a su derecha y **Cualquier usuario** nos permite crear un usuario anónimo (el usuario en blanco). Elijamos el **Usar el campo texto** y escribamos **bill**.

Proveedor

Por defecto, este menú está configurado **Cualquier proveedor**, con lo que usará el valor % de proveedor. La elección **Local** quiere decir localhost. La elección **Usar tabla de proveedores** (que crea un valor en blanco en el campo del proveedor) quiere decir mirar en la tabla `mysql.hosts` para los privilegios específicos de la base de datos. **Usar campo texto** nos permite escribir el valor exacto para el proveedor que queremos. Elijamos **Local**.

Contraseña

A pesar que es posible crear un usuario *sin* contraseña (con la elección **No contraseña**), es mejor *tener* una contraseña. Tenemos que escribirla dos veces (ya que no podemos ver lo que hemos escrito) para confirmar la contraseña. Escribamos **bingo**.

Privilegios globales

Los privilegios globales gobiernan el acceso a todas las bases de datos de este usuario, que también se llaman privilegios súper usuario. Un usuario normal no debería tenerlos a no ser que haya buenas razones.

Por supuesto, si estamos creando un súper usuario seleccionamos todos los privilegios globales que este usuario necesita. Estos privilegios están divididos en los grupos **Datos**, **Estructura** y **Administración**.

En nuestro ejemplo, **bill** no tendrá ningún privilegio global.

Límites de los recursos

Podemos limitar los recursos que un usuario utiliza en este servidor (por ejemplo, el número máximo de consultas por hora) Zero significa que no hay límite. No pondremos límites de recursos a **bill**.

La siguiente muestra de pantalla muestra el estado de la pantalla antes de hacer clic en **Continuar para crear** esta definición de usuario (otros campos se han dejado con su valor por defecto):

Información de la cuenta

Nombre de usuario: Use el campo de texto: bill

Servidor: Local localhost

Contraseña: Use el campo de texto: xxxxxx

Debe volver a escribir: xxxxxx

Generar la contraseña:

Editar un usuario

Esta página aparece después de crear un usuario, o cada vez que hagamos clic en el enlace **Editar** de un usuario en la **Información general del usuario**. Hay cuatro secciones en esta página, cada una con un botón **Continuar** distintivo, de modo que cada sección se opera de manera independiente y tiene un propósito distintivo.

Editar privilegios

Esta sección tiene el mismo aspecto que el diálogo **Añadir un nuevo usuario** y se usa para ver y cambiar los privilegios globales.

Privilegios específicos de las bases de datos

Privilegios específicos para la base de datos

Base de datos Privilegios Conceder Privilegios específicos para la tabla Acción

Ninguna

Añadir privilegios a esta base de datos: Use el campo de texto:

En esta sección definimos las bases de datos a las que el usuario tiene acceso y los privilegios exactos. Ahora vemos **None** porque no hemos definido ninguna. Hay dos maneras de definir los privilegios de las bases de datos. Primero, podemos elegir una de las bases existentes del menú desplegable:

Añadir privilegios a esta base de datos: Use el campo de texto:

Use el campo de texto:

- db1
- dbbook
- mysql

Con esto le daremos privilegios para esta base de datos solamente. También podemos elegir **Usar el campo texto** y escribir un nombre de base de datos. Podemos escribir un nombre de una base de datos que no existe de modo que el usuario pueda crearla más tarde (siempre que le demos permisos `CREATE` en el siguiente panel) o podemos usar caracteres especiales como el subrayado y el símbolo del porcentaje como comodines.

Por ejemplo, al escribir **bill** aquí permitiríamos a **bill** crear una base de datos y escribir **bill%** permitiría la creación de cualquier base de datos que comience con **bill**. Para nuestro ejemplo, escribiremos **bill**.

La siguiente pantalla se usa para configurar los privilegios de **bill** en la base de datos **bill** y crear privilegios específicos de las tablas.

Para aprender más cosas sobre un privilegio específico, podemos mover el ratón sobre el nombre del privilegio (que siempre está en inglés) y aparecerá una explicación sobre el privilegio en el idioma que estemos usando. Le damos privilegios `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `CREATE`, `ALTER`, `INDEX` y `DROP` a **bill** en esta base de datos y hacemos clic en **Continuar**.

Usuario 'bill'@'localhost' - Base de datos bill : Editar los privilegios

Privilegios específicos para la base de datos ([Marcar todos/as](#) / [Desmarcar todos](#))

Nota: Los nombres de los privilegios de MySQL están expresados en inglés

Datos	Estructura	Administración
☒ SELECT	☒ CREATE	☐ GRANT
☒ INSERT	☒ ALTER	☐ LOCK TABLES
☒ UPDATE	☒ INDEX Permite crear nuevas bases de datos y tablas.	☐ REFERENCES
☒ DELETE	☒ DROP	
	☐ CREATE TEMPORARY TABLES	

[Continuar](#)

Privilegios específicos para la tabla

Tabla	Privilegios	Conceder	Privilegios específicos para la columna	Acción
			Ninguna	

Añadir privilegios a esta tabla:

[Continuar](#)

Después de crear los privilegios, la interfaz se queda en la misma página de modo que podemos refinar los privilegios aún más. No podemos asignar los privilegios específicos de las tablas de momento porque la base de datos para la que hemos creado los privilegios no existe.

La manera de volver a la página de privilegios generales de bill es haciendo clic en el título '**bill@localhost**':

 Usuario '**bill**'@'**localhost**' - Base de datos **bill**

Esto nos lleva de vuelta a la página inicial excepto por un cambio en una sección:

Privilegios específicos para la base de datos				
Base de datos	Privilegios	Conceder	Privilegios específicos para la tabla	Acción
dbbook	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER	No	No	 
Añadir privilegios a esta base de datos: Use el campo de texto: 				
				Continuar

Vemos los privilegios existentes en la base de datos **bill** para el usuario **bill** (que podemos Editar o Revocar) y podemos añadir privilegios para **bill** en otra base de datos. También vemos que **bill** no tiene privilegios específicos de tabla en la base de datos **bill**.

Cambiar la contraseña

Este diálogo forma parte de la página **Editar usuario**, podemos cambiar la contraseña de **bill** o suprimirla, permitiendo que **bill** se conecte sin contraseña:

Cambio de contraseña

☐ Sin contraseña
 ☒ Contraseña:

Debe volver a escribir:

Cambiar la información de la cuenta o copiar el usuario

Este diálogo se puede usar para cambiar la información de la cuenta del usuario o para copiar la información de la cuenta a un usuario nuevo:

Cambiar la información de la cuenta / Copiar el usuario

Información de la cuenta

Nombre de usuario: Use el campo de texto: bill

Servidor: Local localhost

Contraseña: No cambiar la contraseña

Debe volver a escribir:

Generar la contraseña: Generar Copiar

Crear un nuevo usuario con los mismos privilegios y...

☒ ...mantener el anterior.

☐ ...borrar el viejo de las tablas de usuario.

☐ ...revocar todos los privilegios activos del viejo y eliminarlo después.

☐ ...borrar el viejo de las tablas de usuario y luego volver a cargar los privilegios.

Continuar

Por ejemplo, supongamos que Bill nos llama y nos dice que prefiere que el nombre de usuario sea **billy** en lugar de **bill**. Sólo tenemos que añadir una **y** al nombre de usuario, elegir **Local** como proveedor y seleccionar **Suprimir el antiguo de las tablas de usuario**:

Cambiar la información de la cuenta / Copiar el usuario

Información de la cuenta

Nombre de usuario: Use el campo de texto: billy

Servidor: Local localhost

Contraseña: No cambiar la contraseña

Debe volver a escribir:

Generar la contraseña: Generar Copiar

Crear un nuevo usuario con los mismos privilegios y...

☐ ...mantener el anterior.

☒ ...borrar el viejo de las tablas de usuario.

☐ ...revocar todos los privilegios activos del viejo y eliminarlo después.

☐ ...borrar el viejo de las tablas de usuario y luego volver a cargar los privilegios.

Continuar

Después de esta operación, **bill** no existirá en la base de datos **mysql** y los privilegios se transferirán a **billy**, incluyendo los privilegios de la base de datos **bill**. Pero la definición del usuario **bill** todavía existe en la memoria y todavía es efectiva. Si hubiésemos elegido la opción **Suprimir el antiguo de las tablas de usuario y recargar los privilegios más tarde** en su lugar, la definición del usuario **bill** hubiese dejado de ser válida de inmediato.

Alternativamente, podemos crear otro usuario basado en el anterior si hacemos clic en la elección **mantener el antiguo**. Podemos elegir transferir la contraseña al nuevo usuario con **No cambiar la contraseña** o cambiar si la escribimos dos veces. La opción **Revocar todos los privilegios activos** terminan de inmediato con los privilegios efectivos de este usuario si está conectado.

Eliminar un usuario

Esto se hace desde la sección **Información general del usuario** de la página **Privilegios**. Seleccionamos el usuario que queremos suprimir y luego usamos una de las siguientes opciones (en la sección **Suprimir usuarios seleccionados**) como sea más apropiado:

- **Sólo borrar los usuarios de la tabla de privilegios:** La supresión solo ocurre en las definiciones que están en el disco, las definiciones efectivas de los usuarios que están en la memoria todavía funcionan.
- **Revocar todos los privilegios activos de los usuarios y suprimirlos más tarde:** Suprime todos los privilegios efectivos en la memoria antes de revocarlos.
- **Suprimir los usuarios y recargar los privilegios más tarde:** Esta opción significa que no tenemos que recordar hacer clic en el enlace **Recargar los privilegios**, lo que hace que las cuentas de los usuarios no funcionen.

También podemos elegir **Vaciar la base de datos que tiene el mismo nombre que el usuario**.

Información de la base de datos

Cuando entramos en la sub-página base de datos vemos la lista de las bases de datos existentes (sin las estadísticas):

Servidor: localhost

[Bases de datos](#)
[SQL](#)
[Estado actual](#)
[Variables](#)
[Juegos de caracteres](#)
[Motores](#)
[Privilegios](#)
[Procesos](#)

[Exportar](#)
[Importar](#)

Bases de datos

Base de datos	
☐ db1	
☐ dbbook	
☐ mysql	
Total: 3	

[↑](#)
[Marcar todos/as](#) / [Desmarcar todos](#)
Para los elementos que están marcados: [✖](#)

[⚙](#)
Activar las estadísticas

⚠ Nota: Activar aquí las estadísticas de la base de datos podría causar tráfico pesado entre el servidor web y el servidor MySQL.

[⚙](#)
Crear nueva base de datos

También vemos el enlace **Activar estadísticas**. Por defecto las estadísticas no están activadas porque el tamaño de los datos y los índices *para todas las tablas de todas las bases de datos* pueden costarle a MySQL recursos de servidor valiosos.

Activar las estadísticas

Si hacemos clic en este enlace aparece una página modificada:

Estadísticas de la base

	Base de datos	Cotejamiento	Tablas	Filas	Datos	Índices	Total	Residuo a depurar
☐	db1	latin1_swedish_ci	3	0	48.0 KB	0 Bytes	48.0 KB	0 Bytes
☐	dbbook	latin1_swedish_ci	10	28	382.1 KB	25.0 KB	407.1 KB	128 Bytes
☐	mysql	latin1_swedish_ci	16	1,455	210.0 KB	59.0 KB	269.0 KB	Revisar los privilegios para la base de datos "dbbook".
	Total: 3	latin1_swedish_ci	29	1,483	640.1 KB	84.0 KB	724.1 KB	1.8 KB

[↑](#)
[Marcar todos/as](#) / [Desmarcar todos](#)
Para los elementos que están marcados: [✖](#)

[⚙](#)
Desactivar las estadísticas

Clasificar estadísticas

Por defecto, la lista está clasificada por orden de nombre de base de datos ascendente. Si necesitamos encontrar la base de datos con más tablas, o la que ocupa más espacio, con un simple clic en la columna **Tablas** o **Total** nos las clasifica de ese modo. Un segundo clic cambia el orden de clasificación.

Verificación de los privilegios de las bases de datos

El enlace **Verificar privilegios** muestra los privilegios de una base de datos específica. Un privilegio global en el usuario puede que se muestre aquí ya que también le da acceso a esta base de datos. También podemos ver los privilegios específicos de esta base de datos. El enlace **Editar** nos manda a otra página para editar los privilegios del usuario mencionado.

🌱 Usuarios con acceso a "dbbook"					
Usuario	Servidor	Tipo	Privilegios	Conceder	Acción
cualquiera	localhost	global	USAGE	Sí	
bill	localhost	global	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP	No	
marc	%	global	USAGE	Sí	
root	localhost	global	ALL PRIVILEGES	Sí	

Vaciar las bases de datos seleccionadas

Aquí vemos una operación que no debería tomarse a la ligera. Para vaciar algunas bases de datos, marcamos los recuadros que están al lado de las bases de datos que queremos vaciar y usamos el enlace **Vaciar bases de datos**. Luego aparece una pantalla de confirmación.

Operaciones del servidor

Los enlaces **Estatus**, **Variables** y **Procesos** están disponibles para obtener información sobre el servidor SQL y actuar sobre procesos específicos.

Verificación del estado del servidor

Estas estadísticas reflejan la actividad total del servidor MySQL incluyendo (pero no limitado a) la actividad generada por las consultas mandadas por phpMyAdmin.

Página general del estado

El enlace **Estado** produce información del tiempo de ejecución de este servidor. La página tiene numerosas secciones. Primero obtenemos información sobre el tiempo de ejecución y el tiempo de arranque, luego los valores total y medios del tráfico y las conexiones (el $\emptyset$ significa medio):

Servidor: localhost

Bases de datos

SQL

Estado actual

Variables

Juegos de caracteres

Motores

Privilegios

Procesos

Exportar

Importar

Información acerca del tiempo de ejecución del proceso principal

[Actualizar]

[Resetear]

[MySQL - Documentación]

Este servidor MySQL ha estado activo durante 0 días, 3 horas, 7 minutos y 34 segundos. Se inició en 29-06-2006 a las 13:41:53.

[Gestor]

[Cache de consultas]

[Procesos]

[Log binario]

[Datos temporales]

[Delayed inserts]

[Cache principal]

[Vínculos (Joins)]

[Replicación]

[Organizando (sorting)]

[Tablas]

Tráfico del servidor: Estas tablas muestran las estadísticas de tráfico en la red de este servidor MySQL desde su inicio.

Tráfico		ø por hora	Conexiones		ø por hora	%
Recibido	7 MB	2 MB	Número máx. de conexiones concurrentes	2	---	---
Enviado	15 MB	5 MB	Intentos fallidos	0	0.00	0.00%
Total	22 MB	7 MB	Abortado	0	0.00	0.00%
			Total	321	102.68	100.00%

Al lado se ven las estadísticas de las consultas (aquí parcialmente). El número medio de consultas por hora, minuto y segundo dan una buena indicación de las cargas del servidor.

Esto está seguido por las estadísticas de cada comando MySQL, con el número absoluto de veces, media por hora y el porcentaje del número total de comandos con que se ejecutó un comando específico.

Estadísticas de consulta: Desde su inicio, 4,761 consultas han sido enviadas al servidor.							
Total	ø por hora	ø por minuto	ø por segundo				
5 k	1.52 k	25.38	0.42				
Tipo de consulta		ø por hora	%	Tipo de consulta		ø por hora	%
admin commands	0	0.00	0.00%	restore table	0	0.00	0.00%
alter db	0	0.00	0.00%	revoke	6	1.92	0.14%
alter table	9	2.88	0.20%	revoke all	0	0.00	0.00%
analyze	0	0.00	0.00%	rollback	0	0.00	0.00%
backup table	0	0.00	0.00%	savepoint	0	0.00	0.00%
begin	0	0.00	0.00%	select	850	271.90	19.14%
change db	1 k	402.10	28.31%	set option	640	204.73	14.41%
change master	0	0.00	0.00%	show binlog events	0	0.00	0.00%
check	0	0.00	0.00%	show binlogs	35	11.20	0.79%
checksum	0	0.00	0.00%	show charsets	160	51.18	3.60%
commit	0	0.00	0.00%	show collations	160	51.18	3.60%
create db	0	0.00	0.00%	show column types	0	0.00	0.00%
create function	0	0.00	0.00%	show create db	0	0.00	0.00%
create index	0	0.00	0.00%	show create table	90	28.79	2.03%
create table	0	0.00	0.00%	show databases	19	6.08	0.43%
delete	0	0.00	0.00%	show errors	0	0.00	0.00%
delete multi	0	0.00	0.00%	show fields	291	93.09	6.55%
do	0	0.00	0.00%	show grants	47	15.03	1.06%

La siguiente sección presenta otra información del servidor (mostrada parcialmente otra vez):

Variable	Valor	Descripción
Flush_commands	1	El número de enunciados FLUSH ejecutados.
Slow_queries	0	El número de consultas que han tomado más segundos que los registrados en long_query_time. [MySQL - Documentación]

Gestor		Empezar ▴
Variable	Valor	Descripción
Handler_commit	0	El número de enunciados COMMIT internos.
Handler_delete	3	El número de veces que una fila fue eliminada de una tabla.
Handler_discover	0	El servidor MySQL puede solicitar al motor de almacenamiento NDB Cluster si conoce acerca de una tabla con un nombre dado. Esto se llama descubrimiento. Handler_discover indica el número ocasiones que las tablas han sido descubiertas.
Handler_read_first	148	El número de veces en que el primer registro ha sido leído desde un índice. Si este valor es alto, sugiere que el servidor está haciendo gran cantidad de escaneos completos del índice; por ejemplo, SELECT col1 FROM foo, asumiendo que col1 está indizado.
Handler_read_key	271	El número de solicitudes hechas para leer una fila basado en una llave. Si este valor es alto, es una buena indicación de que sus consultas y tablas están indizadas apropiadamente.
Handler_read_next	337	El número de solicitudes hechas para leer la siguiente fila en un orden clave. Este se incrementa si usted está consultando una columna índice con un limitante de rango o si usted está haciendo un escaneo del índice.
Handler_read_prev	0	El número de solicitudes hechas para leer la fila previa en un orden clave. Este método de lectura se usa principalmente para optimizar a ORDER BY ... DESC.

Estado InnoDB

En los servidores que soportan InnoDB aparece un enlace al final de la página Estado, que cuando se hace clic en él, da información sobre el subsistema InnoDB, incluyendo el último error InnoDB que ha ocurrido.

```

InnoDB Status

=====
040709 15:27:16 INNODB MONITOR OUTPUT
=====
Per second averages calculated from the last 2 seconds
-----
SEMAPHORES
-----
OS WAIT ARRAY INFO: reservation count 3, signal count 3
Mutex spin waits 0, rounds 0, OS waits 0
RW-shared spins 6, OS waits 3; RW-excl spins 0, OS waits 0
-----
TRANSACTIONS
-----
Trx id counter 0 7424
Purge done for trx's n:o < 0 6989 undo n:o < 0 0
Total number of lock structs in row lock hash table 0
LIST OF TRANSACTIONS FOR EACH SESSION:
---TRANSACTION 0 0, not started, process no 17315, OS thread id 1158499248
MySQL thread id 115, query id 2522 localhost root
SHOW INNODB STATUS
-----
FILE I/O
-----
I/O thread 0 state: waiting for i/o request (insert buffer thread)
I/O thread 1 state: waiting for i/o request (log thread)
I/O thread 2 state: waiting for i/o request (read thread)
I/O thread 3 state: waiting for i/o request (write thread)
Pending normal aio reads: 0, aio writes: 0,
  ibuf aio reads: 0, log i/o's: 0, sync i/o's: 0
Pending flushes (fsync) log: 0; buffer pool: 0
48 OS file reads, 4 OS file writes, 4 OS fsyncs
0.00 reads/s, 0 avg bytes/read, 0.00 writes/s, 0.00 fsyncs/s

```

Variables del servidor

Las sub-página **Variable** muestra diferentes configuraciones del servidor MySQL, que se pueden definir, digamos, en el archivo de configuración `my.cnf` de MySQL. Estos valores no se pueden cambiar en phpMyAdmin:

Servidor: localhost

[Bases de datos](#)
[SQL](#)
[Estado actual](#)
[Variables](#)
[Juegos de caracteres](#)
[Motores](#)
[Privilegios](#)
[Procesos](#)

[Exportar](#)
[Importar](#)

Variables y parámetros del servidor

Variable	Valor de la sesión / Valor global
back log	50
basedir	C:\Archivos de programa\MySQL\MySQL Server 4.1\
binlog cache size	32,768
bulk insert buffer size	8,388,608
character set client	utf8
(Valor global)	latin1
character set connection	utf8
(Valor global)	latin1
character set database	latin1
character set results	utf8
(Valor global)	latin1

Procesos del servidor

La sub-página **Procesos** está disponible para los súper usuarios y los usuarios normales. Un usuario normal sólo verá los procesos que le pertenecen, mientras que un súper usuario verá todos los procesos.

En esta página vemos una lista de todos los procesos del servidor y un enlace **Matar** que nos permite terminar un proceso específico.

 **Lista de procesos**

Su consulta se ejecutó con éxito

consulta SQL:

SHOW PROCESSLIST

[Editar] [Crear código PHP] [Actualizar] [Validar el SQL]

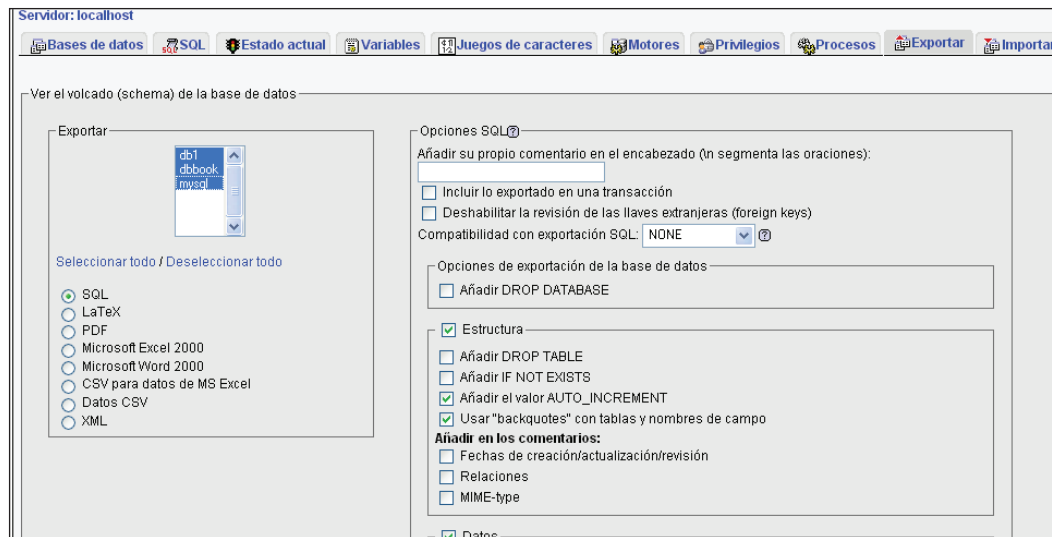
← T →	Identificación	Usuario	Servidor	Base de datos	Comando	Tiempo	Estado actual	consulta SQL
Matar el proceso	341	root	localhost:4620	Ninguna	Sleep	0	---	---
Matar el proceso	342	root	localhost:4621	mysql	Query	0	---	SHOW PROCESSLIST

Este ejemplo sólo tiene un proceso en funcionamiento, creado por el comando `SHOW PROCESSLIST` (se ha hecho para generar esta página), este proceso no se puede matar porque no duró lo suficiente. Normalmente veríamos más procesos en funcionamiento en un servidor con más actividad.

Exportación completa del servidor

En el capítulo 7 explicamos como exportar una base de datos o una tabla. Es posible que un súper usuario o un usuario normal exporten al menos una o más bases de datos en una operación.

El enlace **Exportar** nos manda a la página que se ve abajo, que tiene la misma estructura que el resto de páginas de exportación, excepto por las lista de bases de datos:



[ Exportar bases de datos largas puede que funcione o no: Depende de su tamaño, las opciones que se hayan elegido y los componentes del PHP del servidor Web (especialmente el tamaño de la memoria y el tiempo de ejecución).]

Resumen

En este capítulo, hemos visto varias características disponibles para los administradores del sistema:

- La administración de las cuentas de usuarios.
- Administración de privilegios.
- Verificación de los privilegios de las bases de datos.
- Verificación del estatus del servidor.
- Exportación completa del servidor.

19

Procedimiento para solucionar anomalías y soporte

Este capítulo propone una guía para resolver los errores comunes y da pistas sobre como evitar estos problemas. También explica como interactuar con el equipo de desarrollo para soporte, informe de anomalías y contribuciones.

Requisitos del sistema

Al principio del archivo `Documentation.html`, que está incluido con el kit que nos hemos descargado, hay una sección que discute los requisitos del sistema para la versión particular de phpMyAdmin que estamos usando. Es crucial que estos requisitos se cumplan y que el entorno esté bien configurado para que no haya problemas.

Algunos síntomas que aparecen como anomalías en phpMyAdmin en realidad los causa el entorno del servidor. Algunas veces, el servidor Web no está configurado para interpretar correctamente los archivos `.php`, o el componente PHP del servidor Web no funciona con la extensión `mysql`. Las cuentas MySQL pueden estar mal configuradas. Esto puede suceder tanto en los servidores caseros como en los servidores proveídos.

Cuando sospechamos que algo no va bien, podemos probar un sencillo programa PHP, `test.php`, que contiene la siguiente verificación para ver si el componente PHP responde bien:

```
<?php
echo 'hello';
?>
```

Deberíamos ver el mensaje **hello**. Si funciona, podemos probar otro programa:

```
<?php
Phpinfo ();
?>
```

Este mensaje muestra información sobre el componente PHP, incluyendo las extensiones disponibles. Deberíamos ver al menos una sección sobre MySQL (que nos demuestre que la extensión `mysql` está disponible) que nos da información sobre la **versión Cliente API** de MySQL.

También podemos probar otros programas PHP que conectan con MySQL, para ver si el problema es más general que tan sólo phpMyAdmin sin funcionar. Como consejo general, deberíamos usar las últimas versiones estables de cada componente.

Configuración básica

Siempre deberíamos verificar dos veces como hemos hecho la instalación, incluyendo los permisos adecuados y las titularidades. Puede haber errores tipográficos cuando modificamos `config.inc.php`.

Resolver los errores comunes

Para poder resolver un problema, primero deberíamos localizar el origen del mensaje de error. Aquí hay varios componentes que pueden generar estos mensajes:

- Servidor MySQL: Estos mensajes los retransmite phpMyAdmin que muestra **MySQL said** seguido por el mensaje.
- El componente PHP del servidor Web (por ejemplo, **Parser error**).
- El servidor Web (el error se puede ver desde el navegador o en los archivos de conexión del servidor Web).
- El navegador Web (por ejemplo, errores JavaScript).

Las secciones *Mensajes de error y otros problemas* están basadas en los diferentes mensajes que se han encontrado en el foro de ayuda de phpMyAdmin, también son una explicación al material que se encuentra en la sección FAQ de `Documentation.html`.

Mensajes de error

Esta sección se refiere a los mensajes de error explícitos, tal y como los muestra phpMyAdmin.

No se puede cargar la extensión MySQL

Para conectarse al servidor MySQL, PHP necesita una **extensión MySQL**; un conjunto de funciones MySQL. Esta extensión puede ser parte del servidor PHP (compilada dentro) o puede ser necesario cargarla de forma dinámica, pero phpMyAdmin ha intentado cargarla y ha fallado. El error implica que ningún otro programa de PHP puede conectarse al servidor MySQL.

La extensión que se requiere está en el archivo que se puede llamar `mysql.so` en Linux y UNIX o `mysql.dll` en Windows. Si nuestro servidor PHP proviene de un paquete de programas, podemos encontrar e instalar otro paquete de programas probablemente llamado `php-mysql` (el nombre depende de la distribución). Si no, podemos compilar nuestro servidor PHP con la extensión apropiada, tal y como se explica en la documentación PHP. Al menos una de las distribuciones conocidas de Linux (RedHat 8.0) no ofrece esta extensión como parte del diálogo de instalación del servidor Web/PHP, a pesar que los paquetes están presentes en el disco de instalación 3.

MySQL Said: No se puede conectar al servidor local MySQL

Este mensaje indica que el servidor MySQL no funciona o que no se puede llegar a él desde el servidor Web. También puede ser un problema de configuración de socket (Linux/UNIX) o de transmisión (Windows).

Problema de socket (Linux / UNIX)

El socket que está configurado en `php.ini` (tiene un ejemplo abajo) no corresponde con el socket del servidor MySQL en funcionamiento:

```
mysql.default_socket = /tmp/mysql.sock
```

Como resultado, PHP no puede llegar a MySQL. Podemos cambiarlo a:

```
mysql.default_socket = /var/lib/mysql/mysql.sock
```

Sin embargo, para estar seguros, debemos encontrar la posición exacta de este socket.

Problema de nombre de transmisión correcto (Windows)

Es un problema parecido al de arriba, pero en Windows se puede resolver ajustando `mysql.default_socket`, pero esta vez se usará el nombre de transmisión correcto para conectar de manera local al servidor MySQL. Por ejemplo:

```
mysql.default_socket = MySQL
```

Error # 2003: El servidor no responde

Si el servidor MySQL no está en la misma máquina que el servidor Web y no responde, phpMyAdmin; desde la versión 2.6.0; detecta el error y por consiguiente manda un informe.

MySQL Said: Acceso denegado

Este error se puede resolver cuando entendamos los parámetros de conexión relevantes.

Cuando usamos autenticación http

No podemos usar el mecanismo de seguridad del servidor basado en un archivo `.htaccess` y la autenticación `http` en `config.inc.php` juntos. Una solución alternativa es usar el tipo de autenticación `cookie` en lugar de `http`.

Cuando usamos autenticación http, cookie o config

El parámetro proveedor de `config.inc.php` debe cuadrar con el proveedor definido en los permisos de acceso al usuario. Algunas veces, un administrador de sistema puede crear una cuenta autorizando al usuario `bill` y el proveedor `localhost`. Si intentamos usar el proveedor `127.0.0.1` en `config.inc.php`, MySQL lo rechazará a pesar que lleve a la misma máquina. Puede ocurrir el mismo problema si probamos con el nombre real de la máquina (`mysql.domain.com`) y la definición que se ha hecho para `localhost`.

Acceso denegado ... "usando password: NO"

Si el mensaje termina usando **password: NO**, significa que no estamos transmitiendo una contraseña y que MySQL rechaza este intento de conexión. Puede que no se haya configurado el valor de la contraseña en `config.inc.php`.

Acceso denegado ... "usando password: YES"

Se transmite una contraseña, pero la combinación proveedor/nombre de usuario/contraseña ha sido rechazada por MySQL.

Alarma: no se puede añadir información de cabecera

Este problema lo causan algunos caracteres como líneas en blanco, espacios y otros caracteres que están presentes en `config.inc.php`; antes del marcador `<?php` o después del marcador `?>` al final. Deberíamos suprimirlos con un editor que soporte archivos `.php` como se discute en el capítulo 2.

MySQL Said: Error 127, la tabla se tiene que reparar

En el panel de la izquierda, hacemos clic en el nombre de la base de datos, seleccionamos el nombre de tabla (usando el recuadro necesario) en la que está el error y elegimos **Reparar** en el menú desplegable de abajo. Hay más detalles disponibles en el capítulo 10.

La columna BLOB que se usa en la especificación de clave no tiene tamaño de clave

MySQL requiere que los índices configurados en la columna BLOB tengan un tamaño limitado. Ya que la sencilla técnica para crear índices que está disponible no nos permite especificar el tamaño, la manera correcta es crear una columna sin un índice y luego ir a la página Estructura y usar el diálogo **Crear un índice**. Aquí podemos elegir la columna BLOB y configurar un tamaño para el índice.

IIS: No hay un fichero de entrada especificado

Es un problema de permisos. El **servidor de información por Internet (IIS)** debe poder leer nuestros programas. Como el servidor funciona bajo el usuario IUSR_machiname, tenemos que hacer lo siguiente:

- Hacer clic con el botón de la derecha en la carpeta donde hemos instalado phpMyAdmin.
- Elegir **Propiedades**.
- Bajo la pestaña **Seguridad**, hacer clic en **Añadir** y seleccionar el usuario IUSR_machiname de la lista.
- Asegurarnos que este usuario tiene permisos de lectura al directorio.

Un error "404:página no encontrada" cuando se modifica una fila

Esto pasa cuando nuestro parámetro `$cfg['PmaAbsoluteUri']` de `config.inc.php` no está configurado bien. El capítulo 2 explica como configurar este parámetro.

Otros problemas

Aquí cubrimos soluciones a problemas que no aparecen en la pantalla como un mensaje específico de error.

Página en blanco o caracteres raros

Por defecto, phpMyAdmin usa técnicas de almacenamiento de resultados en la memoria temporal y de compresión para acelerar la transmisión de los resultados al navegador. Estas técnicas pueden interferir con otros componentes del servidor Web causando problemas de presentación. Podemos configurar `$cfg['OBGzip']` en `FALSE` en `config.inc.php`. Esto debería resolver el problema.

No se puede crear una base de datos

En la página de inicio, aparece **No hay permisos** al lado del diálogo **Crear base de datos** si phpMyAdmin detecta que la cuenta con la que nos hemos conectado no tiene permiso para crear una base de datos. Esta situación ocurre a menudo en servidores proveídos, en los que el administrador del sistema prefiere crear una base de datos por cada cliente.

Si no estamos en un servidor proveído, este mensaje tan sólo refleja el hecho que no tenemos el permiso global `CREATE`.

Problemas al importar archivos largos o cargando BLOBs largos

Normalmente, estos problemas indican que hemos alcanzado algún límite durante la transferencia. El capítulo 8 explica estos límites y las acciones recomendadas. Una solución sería dividir nuestros archivos texto (vaya a Internet para buscar **divisores de archivos**).

MySQL pérdida de la contraseña Root

El manual MySQL explica la solución general en http://www.mysql.com/doc/en/Resetting_permissions.html.

La solución implica parar el servidor MySQL y reiniciarlo con la opción especial **skip-grant-tables** (que básicamente inicia el servidor sin seguridad). La manera de parar y reiniciar el servidor depende de la plataforma de servidor que se use. Luego podemos conectarnos al servidor desde phpMyAdmin como súper usuario (por ejemplo root) con cualquier contraseña. El siguiente paso es cambiar la contraseña de root (vaya al capítulo 17). Luego podemos parar el servidor MySQL y reiniciarlo con el procedimiento normal (la seguridad se activará de nuevo).

Duplicar nombres de campo cuando creamos una tabla

Aquí hay un síntoma curioso: Cuando intentamos crear una tabla que tenga un campo llamado FIELD1, por ejemplo, del tipo `VARCHAR (15)`, parece que phpMyAdmin manda un comando para crear dos campos idénticos llamados FIELD1. El problema no lo causa phpMyAdmin sino el entorno. En este caso, el servidor Apache parece estar bien configurado para funcionar con los programas de PHP cuando de hecho no lo está. Sin embargo, el error solo aparece en algunos programas.

El problema ocurre cuando dos conjuntos de directivas diferentes (conflictivas/opuestas) se usan en el archivo de configuración de Apache:

```
SetOutputFilter PHP
SetInputFilter PHP
```

Y

```
AddType application/x-httpd-php .php
```

Estos conjuntos de directivas pueden estar en dos archivos de configuración Apache diferente de modo que es difícil encontrarlas. La manera recomendada es usar `AddType`, de modo que tenemos que poner comentarios en el resto de líneas, como se muestra en el siguiente recorte y reiniciar Apache:

```
#SetOutputFilter PHP
#SetInputFilter PHP
```

La ventana de autenticación aparece más de una vez

Este problema ocurre cuando intentamos iniciar phpMyAdmin con una URL diferente a la que hemos configurado en `$cfg['PmaAbsoluteUri']`. Por ejemplo, un servidor puede tener más de un nombre o puede que intentemos usar la dirección IP en lugar del nombre.

phpMyAdmin cambia el tamaño de la columna

A veces MySQL decide cambiar el tipo y el tamaño de la columna para una definición de columna más eficiente. Esto suele pasar con `CHAR` y `VARCHAR`.

Ver muchas bases de datos que nos son nuestras

Este problema ocurre la mayor parte de las veces cuando actualizamos MySQL 4. La actualización automática del servidor da privilegios globales a todos los usuarios a `CREATE TEMPORARY TABLES`, `SHOW DATABASES` y `LOCK TABLES`. Estos privilegios también permiten a los usuarios ver los nombres de todas las bases de datos (pero no sus tablas) hasta que la actualización otorgue las tablas tal y como se describe en el manual MySQL. Si los usuarios no necesitan estos privilegios, podemos revocarlos y sólo verán las tablas para las que tengan derechos.

No poder almacenar un valor mayor que 127

Esto normal si hemos definido el tipo de columna `TINYINT` ya el valor máximo de esta columna es 127. Pueden aparecer problemas similares con otros tipos de columnas numéricas. Si cambiamos el tipo de columna a `INT` expandiremos el rango disponible de valores.

Buscar soporte

El punto de inicio para el soporte es la página de inicio <http://www.phpmyadmin.net> que tiene secciones sobre documentación y soporte (comentarios). Aquí encontrará enlaces para los foros de discusión y algunos rastreadores como:

- Rastreador de errores.
- Rastreador de RFE (requested features o características pedidas).
- Rastreador de las traducciones.
- Rastreador de los parches.
- Rastreador del soporte.

FAQ

El archivo `Documentation.html`, que es parte del producto, contiene una larga sección FAQ, con preguntas y respuestas numeradas. Se recomienda mirar la sección FAQ como primer recurso de ayuda.

Foros de ayuda

El equipo de desarrollo le recomienda que primero use los foros del producto para buscar el problema que tenga y luego empiece un foro nuevo de discusión antes de escribir un informe de errores.

Crear una cuenta SourceForge

Se recomienda crear una cuenta (gratuita) SourceForge y usarla para escribir en los foros. Esto nos permite rastrear mejor las preguntas y las respuestas.

Elegir el título del tema

Es importante elegir con cuidado el título del resumen para empezar un nuevo tema en el foro. Títulos como "¡Ayuda!", "¡Ayudad al novato!", "Problema o "error phpMyAdmin" son difíciles de tratar porque las respuestas están relacionadas con este título y la referencia futuras se vuelve problemática.

Leer las respuestas

Como la gente leerá y casi siempre contestará nuestra pregunta, dar opiniones en el foro sobre las respuestas puede ayudar a la persona que ha respondido y al resto de gente que tenga el mismo problema.

Rastreador de soporte

Es otro lugar para pedir ayuda. Además, si hemos escrito un informe de errores, que en realidad es una petición de soporte, el informe se moverá al rastreador de soporte. Con su cuenta de usuario SourceForge, le notificarán del cualquier cambio en el rastreador.

Rastreador de errores

En este rastreador vemos los errores que aún no se han corregido, más los errores que se han corregido para la siguiente versión (es para evitar tener informes de errores duplicados).

Descripción del entorno

Ya que los programadores intentan reproducir el problema mencionado, describirles nuestro entorno sirve de ayuda. Esta descripción puede ser corta pero debe contener los siguientes datos:

- La versión phpMyAdmin (sino el equipo espera que sea la versión actual estable).
- El nombre del servidor Web y la versión.
- La versión PHP.
- La versión MySQL.
- El nombre del servidor y la versión.

Normalmente, especificar el sistema operativo en el que está el servidor Web o el cliente no es necesario a no ser que nos demos cuenta que el error pertenece solo a un SO. Por ejemplo, la FAQ 5.1 describe un problema en el que usuario no podía crear una tabla con más de 14 campos. Esto sólo pasa bajo Windows 98.

Descripción del error

Deberíamos dar una descripción precisa de lo que pasa (incluyendo el mensaje de error, los resultados esperados y los resultados efectivos que obtenemos). Los informes se administran con facilidad si sólo tienen un problema por informe de errores (a no ser que los problemas estén claramente relacionados).

Algunas veces, puede ser útil adjuntar un pequeño fichero de explicación al informe de errores para ayudar a los programadores a reproducir el problema. Las muestras de pantalla son bienvenidas.

Contribuir al proyecto

Desde 1998, cientos de personas han contribuido a traducir, crear código para nuevas características, sugerencias o arreglar errores.

El código base

El equipo de desarrollo mantiene un código en desarrollo desde el cual sacan versiones nuevas de forma periódica. En la sección de descargas de la página de inicio, un párrafo describe como usar CVS para obtener la última versión en desarrollo (también se puede hacer si descargamos el programa instantáneo CVS). Se considerará con más prioridad (actualización de traducción, parche, nueva característica) si se refiere al código base más nuevo y no a una versión antigua de phpMyAdmin.

Actualización de las traducciones

Si echamos un vistazo a la lista de 47 idiomas del proyecto, nos damos cuenta que no todos están mantenidos por igual. Podemos intentar unirnos al traductor oficial de un idioma en particular para proponer correcciones o traducciones de los últimos mensajes añadidos. Si esta persona no contesta, podemos mandar nuestras modificaciones al rastreador de traducciones en un archivo comprimido (.zip).

Parches

El equipo de desarrollo puede administrar parches con más facilidad si los mandamos en forma de `context diff` contra el código actual, con una explicación del problema resuelto y la nueva característica que se ha conseguido. Se acredita oficialmente a los contribuidores en `Documentation.html` por su trabajo.

Versiones futuras de phpMyAdmin

Aquí están las características que el equipo de desarrollo está considerando para una posible implementación:

- Mejorar el soporte de las nuevas de MySQL 4.1/5.0.
- Configuración basada en bases de datos con preferencias de usuario.
- Soporte para las sesiones PHP.
- Expandir el producto para soportar otros sistemas de bases de datos.

Resumen

En este capítulo:

- Hemos visto como evitar problemas con un servidor bien configurado.
- Hemos explicado los errores comunes con sugerencias para solucionarlos.
- Hemos visto donde pedir ayuda.
- La sección *Contribuir al proyecto* explicaba como ayudar para mejorar phpMyAdmin.

Índice

Símbolos

-with-bz2 compilador 110
-with-zlib compilador 110

A

Actualización de los criterios, generación de consultas 183
Administración del índice 62, 93
 Índices de campos múltiples 95
 Índices de un campo 94
 Sentencia EXPLAIN 98
Administración del servidor
 Administración del estado general 248
 Administración del estado InnoDB 249
 Variables, servidor MySQL 250
 Verificación del estado 247
Almacenamiento de la contraseña,
 autenticación cookie 28
Añadir datos entre las tablas 146
Añadir la opción 'drop table', exportación de la estructura 106
Añadir la opción valor AUTO INCREMENT, exportación de la estructura 106
Añadir en la opción comentarios, estructura exportación 106
Añadir un usuario Nuevo, diálogo 238
Archivo php.ini 122
Arquitectura de las aplicaciones Web 7
Atributo comentarios de la tabla 143
Atributos de los campos, edición, estructura de las tablas 87
Atributo orden de las tablas 144
Atributo tipo de tabla 143

Autenticación

 Autenticación HTTP 27
Auto módulo, codificación de los datos 40, 224

B

Borrar bases de datos 83
Borrar datos 80
Botón mostrar todo, barra de navegación 70
Búsqueda en la base de datos 138
Búsqueda en una tabla 131
 Búsqueda con comodines 134
 Campos mostrados, selección 132
 Consulta por ejemplo, búsqueda con 133
Búsqueda de datos 131
Búsqueda de múltiples criterios, búsqueda en tablas 136

C

Campo Objeto Binario Extendido 89
Campo TEXT 88
Campos LONGTEXT 88
Características de la administración del servidor 237
Cargar datos 121
Cerrar los nombres de una tabla y de los campos con la opción comillas, exportar la estructura 106
Clave foránea 150
Cláusula WHERE, búsqueda en las tablas 137
Codificación de datos 224
Codificación de los datos 40, 224

- Codificación EUC-JP, soporte Kanji 234, 252
- Codificación SJIS, soporte Kanji 234, 252
- Colaciones 229
- Columnas de criterio, ajustar el número de 187
- Columnas de criterio, generación de consultas 181
- Comando SHOW PROCESSLIST, administración del servidor 251
- Comandos SQL, escribir
 - Consultas con múltiples sentencias 173
 - Edición de las consultas 172
 - Vista base de datos 167
 - Vista tabla 168
- Comparación de formatos CSV y SQL, importar datos 126
- Configuración auto_increment, opciones de tabla atributo 145
- Configuración básica, testeo de la 23
- Configuración checksum, opciones de la tabla atributo 145
- Configuración delay_key_write, opciones de la tabla atributo 145
- Configuración multi-servidor 23
- Configuración pack_keys, opciones de la tabla atributos 145
- Conjuntos de caracteres 225
 - Cambiar, vista base de datos 232
 - Cambiar, vista tabla 233
 - Colaciones 229
 - Conjunto efectivo 231
 - Conjuntos disponibles 230
 - Crear una base de datos 230
 - Elegir 226
 - Importar y exportar 227, 233
 - Vista del servidor 234
- Conjunto de caracteres, exportar bases de datos
 - Elegir un conjunto de caracteres efectivos 226, 252
 - Importar y exportar 227
- Comparación de caracteres 229
- config.inc.php, archivo 18
 - Configurar servidores múltiples 24
- Contribuir al desarrollo de phpMyAdmin 262

- Control de acceso basado en la IP, seguridad 30
- Control user, autenticación 26
- Cookie, autenticación 27
- Copia de seguridad, exportación de los datos 101
- Creación de base de datos
 - Restricciones, permisos 57
- create_table.sql, archivo 151

D

- Datos específicos del servidor, instalación de phpMyAdmin 20
- dbbook, base de datos ejemplo 58
 - Crear la base de datos 60
 - Diseño de la base de datos 60
 - Panel de entrada de los datos, ajuste de 65
- Discriminación entre mayúsculas y minúsculas 153
- Documentación
 - Base de datos selectiva, vista impresión 202
 - Diccionario de datos 203
 - Edición de páginas PDF 205
 - Fuentes 210
 - Generación de imágenes 214
 - Tabla, vista impresión 203

E

- Editar páginas PDF, documentación
 - Crear páginas 206
 - Diseñar páginas 205
 - Editar páginas 206
 - Mostrar páginas 207
- Editor HomeSite 18
- Editor PHP 17
- Editor PHPEdit 18
- Editor UltraEdit 17
- Editor WordPad 17
- Esquema de codificación UTF-8 40, 223
- Extensión multifuncional del correo en Internet (MIME) 212
- Exportación de archivos divididos, exportar la tablas 117
- Exportación de datos 108
- Exportación de la estructura 106
- Exportación de la tabla 116

Exportar 101

Aspectos del rendimiento 108

Exportar bases de datos 251

Conjunto de caracteres 225, 252

F

Filas de criterio, ajustar el número 185

Formato LaTeX, exportar datos 112

Formato de datos CSV, exportar datos 110

Formato SQL, exportar datos 103

Formato SQL, sub-panel Exportar 103

Formato XML, exportar datos 114

Foro de ayuda, soporte phpMyAdmin 260

Fuentes de errores 254

Fuentes de phpMyadmin, documentación 210

Función `mb_convert_encoding`, soporte Kanji 234, 252

Función `mysql_query` 173

Función `strftime`, PHP 109

Funciones, disponibles para manipular datos 77

G

Generación de consultas, múltiples tablas 179

Generación de consultas multi-tabla

Actualizar consultas 182

Ejecución de la consulta 188

Joins automáticas de las tablas 187

Mostrar los resultados 182

Selección de campos 181

Selección de tabla 180

Prestación de clasificación 182

Guardar archivo como sub-panel, exportar datos 109

Guardar archivo como sub-panel

Opciones de compresión 109

Guardar los archivos de exportación en el servidor 119

H

Historial de los comandos SQL

Basado en la base de datos 171

Basado en JavaScript 171

I

Importar datos 121

Archivos CSV 126

Archivos SQL 123

Cargar interfaz 126

Subir el directorio 129

Volcados comprimidos 130

Índice FULLTEXT 96

Índice Fulltext 87

Infraestructura de las tablas enlazadas

Instalación de un usuario 154

Meta-datos 150

Posición 150

Vista relación 156

Insertar datos de un textfile en el diálogo de la pestaña, importar datos 127

Instalación de phpMyAdmin

Instalación multi-usuario 25

Servidores locales, en 18

Instalación de la tabla multi-usuario 152

Instalación de tabla de un usuario 154

Integridad referencial 164

IUSR_machinename, servidor IIS 19

L

Límite del tamaño del fichero, importar datos 122

M

Manipulación de datos 75

Borrar tablas 82

Edición de filas múltiples 79

Filas, duplicar 78

Filas múltiples 81

Icono lápiz 75

Tecla tab 76

Todas las filas, tablas múltiples 82

Todas las tablas, una tabla 81

Una fila 80

Manipulación de la estructura de la tabla 85

Campo ENUM 91

Campo SET 91

Campos, añadir 85

Campos LONGTEXT 88

Optimización de las tablas 97

- Modo vertical 86
- Mantenimiento de la tabla 142**
- Marcadores**
 - Almacenamiento 193
 - Características 191
 - Creación 192
 - Consulta inicial por defecto 196
 - Ejecución 195
 - Manipulación 195
 - Marcadores multi-consulta 194
 - Marcadores parametrizados, crear 197
 - Pasar, a marcadores 198
 - Público 195
 - Recuperar marcadores 194
- Mensajes de error**
 - Acceso denegado a MySQL 256
 - Caracteres raros 258
 - Error 404 – no se encuentra la página 257
 - Errores relacionados con BLOB 257
 - Hay problemas en la transmisión de la contraseña 256
 - IIS – no hay un fichero de entrada especificado 257
 - La ventana de autenticación aparece más de una vez 259
 - MySQL Said- error 127, se tiene que reparar la tabla 257
 - MySQL Said – no se puede conectar al servidor local MySQL 255
 - Nombres de campos duplicados 259
 - No se puede añadir información de cabecera, alarma 256
 - No se puede almacenar un valor mayor que 127 260
 - No se puede cargar la extensión MySQL 255
 - No se puede crear una base de datos 258
 - phpMyAdmin ha cambiado el tamaño de la columna 259
 - Problemas cargando BLOBs grandes 258
 - Se ha perdido la contraseña root de MySQL 258
 - Ver muchas bases de datos que no son nuestras 260
- Menú número de página, barra de navegación 69**
- Meta-datos 149**
 - Transformaciones, usados para las 211
- Modo mínimo, panel de la izquierda 43**
- Modo navegación, mostrar los resultados de la consulta 66**
 - Barra de navegación 68
 - Clasificación de los datos 70
 - Destacar los resultados 71
 - Longitud de la columna, limitar 72
 - Personalizar el aspecto 73
- Modo vertical, barra de navegación 69**
- Módulo iconv, codificar datos 252**
- Módulo recodificar, codificación de los datos 225, 252**
- MySQL**
 - Naturaleza relacional 149
 - Soporte para exportar bases de datos 228, 252
- MySQL Said – acceso denegado, mensaje de error**
 - Mientras se usa la autenticación config 256
 - Mientras se usa la autenticación HTTP 256
- MySQL-webadmin 10**
- mysqldump, exportar datos 101**
- O**
 - Omisión de bases de datos, privilegios de las bases de datos 247**
 - Operación comprobar la tabla, mantenimiento de la tabla 142**
 - Operación limpiar la tabla, mantenimiento de la tabla 142**
 - Operación mover, tablas 145**
 - Operación copiar, tablas 146**
 - Operación de tabla analizar, mantenimiento de la tabla 142**
 - Operación renombrar, tablas 145**
 - Operación reparar la tabla, mantenimiento de la tabla 142**
 - Operación volver a cargar, sub-página privilegios 239**
 - Operaciones multi-tabla 147**
 - Operaciones de la tabla 141**
 - Operadores lógicos, que se usan en la búsqueda 137**
 - Opción campos cerrados por, formato de datos CSV 111**

- Opción campos escapados por, formato de datos CSV 111
- Opción campos terminados por, formato de datos CSV 111
- Opción comentarios, formato LaTeX 113
- Opción complete inserts, exportar datos 108
- Opción datos, formato LaTeX 112
- Opción estructura, formato LaTeX 112
- Opción extended inserts, exportar datos 108
- Opción incluir comentarios de tabla, formato LaTeX 112
- Opción INSERT, formato SQL, exporter datos 105
- Opción línea terminada por, formato CSV 111
- Opción ON DELETE CASCADE, vista relación 161
- Opción ON DELETE, vista relación 160
- Opción ON UPDATE, vista relación 160
- Opción poner nombres de campos en la primera fila, formato de datos CSV 111
- Opción reemplazar Nulo por, formato de datos CSV 111
- Opción relaciones, formato LaTeX 113
- Opción tipo MIME, formato LaTeX 113
- Opción título de tabla, formato LaTeX 112
- Opciones de compresión, exportar datos 110
- Opciones de transformación 213
- Optimizar las operaciones de la tabla, mantenimiento de la tabla 142

P

- Página de inicio, panel de la derecha 49
- Página Personal Inicial (PHP) 8
- Panel de la derecha, interfaz phpMyAdmin 35, 49
 - Página inicial 49
 - Vista base de datos 50
 - Vista tabla 52
- Panel de la izquierda, interfaz phpMyAdmin 35
 - Modo completo 45
 - Presentación jerárquica de las tablas en la base de datos 46
- Parámetro memory_limit, php.ini 122

Parámetro upload_max_filesize, php.ini 122

Parámetros de configuración

- \$cfg['AllowAnywhereRecording'] 40, 224
- \$cfg['AllowArbitraryServer'] 24
- \$cfg['AllowUserDropDatabase'] 83
- \$cfg['AvailableCharsets'] 225, 252
- \$cfg['BgcolorTwo'] 73
- \$cfg['Border'] 73
- \$cfg['BrowseMarkerColor'] 71, 231
- \$cfg['BrowsePointerColor'] 71
- \$cfg['BZipDump'] 110
- \$cfg['CharEditing'] 65
- \$cfg['CharTextareaCols'] 65
- \$cfg['CompressOnFly'] 110
- \$cfg['Confirm'] 80
- \$cfg['DefaultLang'] 38
- \$cfg['DefaultPropDisplay'] 86
- \$cfg['DefaultQueryDatabase'] 167
- \$cfg['DefaultQueryTable'] 169
- \$cfg['DefaultTabDatabase'] 51
- \$cfg['DefaultTabServer'] 53
- \$cfg['DefaultTabTable'] 52
- \$cfg['ExecTimeLimit'] 122
- \$cfg['Export'] 103
- \$cfg['Export'] [format'] 103
- \$cfg['IgnoreMultiSubmitErrors'] 173
- \$cfg['Lang'] 39
- \$cfg['LeftBgColor'] 42
- \$cfg['LeftDisplayLogo'] 41
- \$cfg['LeftDisplayServers'] 41, 48
- \$cfg['LeftFrameLight'] 43
- \$cfg['LeftFrameTableLevel'] 47
- \$cfg['LeftFrameTableSeparator'] 47
- \$cfg['LeftPointerColor'] 42
- \$cfg['LeftWidth'] 42
- \$cfg['LimitChars'] 72, 219
- \$cfg['LoginCookieRecall'] 29
- \$cfg['LongTextDoubleTextarea'] 88
- \$cfg['MaxRowss'] 69
- \$cfg['ModifyDeleteAtLeft'] 75
- \$cfg['MySQLManualBase'] 55
- \$cfg['NavigationBarIconic'] 69
- \$cfg['OBGzip'] 258
- \$cfg['PersistentConnections'] 21
- \$cfg['PmaAbsoluteUri'] 20, 259
- \$cfg['PmaAbsoluteUri_DisableWarning'] 20
- \$cfg['PropertiesIconic'] 75, 87

- \$cfg['ProtectBinary'] 90
- \$cfg['QueryFrame'] 54
- \$cfg['QueryFrameS'] 54, 171
- \$cfg['QueryHistoryDB'] 171
- \$cfg['QueryHistoryMax'] 172
- \$cfg['QueryWindowDefTab'] 171
- \$cfg['QueryWindowHeight'] 55
- \$cfg['QueryWindowWidth'] 55
- \$cfg['RecordingEngine'] 40, 224
- \$cfg['RepeatCells'] 73
- \$cfg['RestrictColumnTypes'] 77
- \$cfg['RestrictFunctions'] 77
- \$cfg['RightBgColor'] 49
- \$cfg['RightBgImage'] 49
- \$cfg['SaveDir'] 120
- \$cfg['Servers'][\$i]['AllowDeny']['order'] 31
- \$cfg['Servers'][\$i]['AllowDeny']['rules'] 31
- \$cfg['Servers'][\$i]['auth_type'] 27
- \$cfg['Servers'][\$i]['blowfish_secret'] 28
- \$cfg['Servers'][\$i]['controluser'] 26
- \$cfg['Servers'][\$i]['history'] 172
- \$cfg['Servers'][\$i]['only_db'] 32
- \$cfg['Servers'][\$i]['port'] 21
- \$cfg['Servers'][\$i]['socket'] 21
- \$cfg['Servers'][\$i]['user'] 23
- \$cfg['Servers'][\$i]['verbose'] 21
- \$cfg['Servers'][\$i]['verbose_check'] 154
- \$cfg['Servers'][\$i]['controlpass'] 154
- \$cfg['Servers'][\$i]['controluser'] 154
- \$cfg['ServersDefault'] 24
- \$cfg['ShowAll'] 69
- \$cfg['ShowBlob'] 89
- \$cfg['ShowChgPassword'] 50
- \$cfg['ShowFunctionFields'] 77, 78
- \$cfg['ShowSQL'] 59
- \$cfg['ShowStats'] 50
- \$cfg['SQLQuery']['Validate'] 176
- \$cfg['SQLValidator']['username'] 176
- \$cfg['SQP']['fmtColor'] 174
- \$cfg['SQP']['fmtType'] 174
- \$cfg['TextareaAutoSelect'] 169
- \$cfg['TextareaCols'] 88
- \$cfg['ThBgColor'] 73
- \$cfg['VerboseMultiSubmit'] 173
- \$cfg[ShowTooltipAliasDB] 143
- \$cfg[ShowTooltipAliasTB] 143

phpMyAdmin

- Administración del servidor 247
- Características 13
- Configuración básica 19
- Detección de errores 254
- Diálogo quitar los usuarios elegidos 238
- Editar privilegios 241
- Información prerrequerida (necesaria) 15
- Integridad referencia, falta de verificación automática 150
- Interfaz 35
- Meta-datos, actualizaciones automáticas de 165
- Requisitos del sistema 253
- Panel de conexión 35
- Parámetro PmaAbsoluteURI 20
- Perspectiva histórica 10
- Privilegios específicos de las bases de datos 241
- Privilegios del usuario 240
- Seguridad 21
- Soporte 260
- Vista relaciones internas 156
- pma_bookmark 198**
- Poblar tablas manualmente, base de datos de ejemplo 63**
- PowerArchiver 16**
- Posición del diálogo archivo de texto, importar datos 123**
- Privilegio CREATE, creación de bases de datos 57**
- Privilegios específicos de la base de datos**
 - Activar estadísticas 246
 - Clasificar estadísticas 246
- Privilegios del usuario**
 - Añadir un usuario 239
 - Borrar un usuario 245
 - Editar los privilegios del usuario 241
- Problemas de enchufe (socket) 255**
- Problema de transmisión correcta, Windows 255**
- Protección a nivel del directorio, seguridad 29**
- Protección de los datos en tránsito, seguridad 32**

R

- Rastreador de errores, soporte de phpMyAdmin 262
- Recuadro de consultas, escribiendo comandos SQL 167
- Recuadros Null de columna, manipulación de datos 76
- Referencias a sitios Web
 - mysql.com 55
- Reglas, acceso de control basado en la IP 30
- Relaciones definidas, tablas enlazadas, ventajas 162
- Relaciones, tablas 149
- Relaciones de las tablas
 - Esquema relacional, documento PDF 204
- Restricción de la base de datos, seguridad 32

S

- Seguridad
 - Instalación de phpMyAdmin 29
- Selector de idioma, panel de acceso 35, 38
- Sentencia CREATE TABLE 124
- Sentencia DROP TABLE IF EXISTS 106
- Sentencia INSERT 124
- Sentencia LOAD DATA INFILE, importar datos 121
- Sentencia LOAD DATA LOCAL INFILE, importar datos 126
- Servidor Windows local, instalación en 18
- Sincronización de consultas 171
- Solución de las anomalías 25
- Soporte Kanji, exportar bases de datos, idioma japonés 234, 252
- Sub-página operaciones, operaciones de la tabla 141, 158
- Sub-página procesos
 - Administración del servidor 251
- Sub-panel exportar 103
- Sub-panel opciones SQL, exportar datos 105
- Sub-página estructura 51

T

- Tablas enlazadas 150
 - Marcadores 191
- Tablas MyISAM, administración del

- índice 96

- Tiempo límite de ejecución, importar datos 122

- Tipo de autenticación config 22

- Tipo de tabla InnoDB 150

- Definir las relaciones 158

- Índice FULLTEXT, soporte para 158

- Transformación

- Aplicaciones externas 219

- Clasificación en las celdas, aplicación externa 220

- Conservar el formato 218

- Enlaces a imágenes, ofrecer 216

- Enlaces del texto, ofrecer 217

- Formato de los datos 216

- MIME 212

- Mostrar partes del texto 219

- Navegador 212

- Pestaña en la que se puede hacer clic 215

- Transformaciones del navegador 213

- Transformaciones basadas en MIME 211

- características de phpMyAdmin 212

V

- Validador SQL 175

- Configuración 176

- Consultas no conformes con el estándar 178

- Consultas conformes con el estándar 176

- Informes 176

- Requisitos del sistema 176

- Ventana de consulta, escribiendo comandos SQL 170

- Ventana de consulta, interfaz phpMyAdmin 36, 54

- Vista base de datos

- Característica consulta por ejemplo (CPE) 179

- Consultas multi-tabla 179

- Vista base de datos, panel de la derecha 36

- Vista impresión, búsqueda en las tablas 134

- Vista relación, tablas enlazadas 156, 159

- Característica comentarios en la columna 165

- Definición del campo mostrar 157

- Definición de las relaciones 157

- Vista servidor, panel de la derecha 53

- Volcado, exportar datos 101