

Bases de datos y software libre. MySQL básico

08

En esta Unidad aprenderás a:

- Instalar, configurar y utilizar el sistema gestor de bases de datos MySQL.
- Realizar cualquier operación de creación, mantenimiento o eliminación de bases de datos y tablas en MySQL.
- Gestionar aspectos de seguridad en MySQL, como el control de los usuarios, o la realización de copias de seguridad.
- Instalar, configurar y utilizar un paquete gráfico para realizar visualmente la administración y mantenimiento de una base de datos MySQL.



8.1 MySQL: un SGBD de código abierto

MySQL es un sistema gestor de bases de datos que se puede encuadrar dentro de la categoría de los programas *open-source*.



Un **programa *open-source*** es aquel cuyo código fuente está disponible para los usuarios y abierto a modificaciones.

El término *open-source* surgió del anteriormente usado *free-software*, que hacía referencia a los programas cuyo código estaba disponible para el usuario, fueran gratuitos o no. Como *free* en inglés significa libre pero también gratis, se prestaba bastante a confusión, por lo que pasó a utilizarse *open-source* («código abierto»).

Aunque *open-source* no siempre implica que su uso sea gratuito, MySQL sí que lo es.



En el caso de que el programa *open-source* se pueda adquirir gratis, el usuario debe distribuir también gratis los cambios que pueda realizar, o los productos que pueda desarrollar basándose en el original. En caso contrario, si el usuario desea hacer negocio con el producto, entonces debe adquirir la licencia comercial de pago.

Mención aparte merecen las licencias de tipo GPL, que pueden consultarse en la siguiente página web: www.gnu.org/copyleft/gpl.html.

En esta unidad trataremos el funcionamiento de MySQL por considerarlo altamente representativo de las bases de datos de libre uso.

A Orígenes e historia de MySQL

MySQL es un caso particular, pues se trata de un programa de licencia *open-source* y gratuito pero que, sin embargo, está mantenido por una empresa, MySQL AB, con sede en Suecia.

El código fuente de MySQL está sólo relativamente abierto y disponible para modificaciones, puesto que es

la empresa MySQL AB la que contrata y coordina los trabajos de mantenimiento del producto. No obstante, los trabajadores contratados, procedentes de todo el mundo, son usuarios del producto que realizan sus encargos a través de Internet.



Orígenes

El origen de MySQL se remonta a la década de los ochenta. Michael Widenius, también conocido como *Monty*, un joven programador que realizaba complejas aplicaciones en lenguaje BASIC, al no encontrar un sistema de almacenamiento de archivos que le resultara satisfactorio, pensó en construir el suyo propio.

Años después, en 1995, y en colaboración con David Axmark, Widenius desarrolló un producto que básicamente era el resultado de sus investigaciones, más dos aportaciones nuevas: el uso del lenguaje SQL y la accesibilidad a través de Internet. Así nació MySQL y también la empresa MySQL AB.



Fig. 8.1. Michael Widenius.



Evolución

La evolución de MySQL se produce con las sugerencias de los usuarios, canalizadas por la empresa MySQL AB, que contrata a programadores de todo el mundo para que, a través de Internet, vayan ampliando y mejorando el producto.



8. Bases de datos y software libre. MySQL básico

8.1 MySQL: un SGBD de código abierto

Las versiones, como es costumbre en este tipo de software libre, se van colgando en Internet para que los usuarios puedan disponer de ellas.

Sin embargo, también como es habitual, hay que distinguir entre *versiones de prueba o beta* y *versiones estables o de producción*.

Las **versiones de prueba o beta** están a disposición de los usuarios para ser testadas en busca de posibles mejoras. Por su parte, las **versiones estables o de producción** son las que ya están preparadas para ser utilizadas a pleno rendimiento.

En la actualidad, 4.1.14 es la última versión estable. Esta versión dispone de características como gestión de transacciones, integridad referencial (bajo ciertas condiciones), uso de índices, incorporación de librerías, etcétera.

La versión de prueba actualmente disponible es la 5.0, que empieza a incorporar elementos hasta ahora no incluidos en MySQL, como el uso de procedimientos, cursores, *triggers* o vistas.

Existe una versión incluso más reciente, la 5.1, pero que aún es un proyecto. Esta versión promete mejoras en la seguridad y en la gestión de claves ajenas, aunque todavía no se conoce cómo se plasmarán dichas características en posteriores versiones estables.



Características de MySQL

Aparte de las características que definen MySQL como programa *open-source*, existen aspectos que lo diferencian de otros productos como, por citar uno conocido, Access. Los atributos a los que hacemos referencia son:

- Posibilidad de crear y configurar usuarios, asignando a cada uno de ellos permisos diferentes.
- Facilidad de exportación e importación de datos, incluso de la base de datos completa.
- Posibilidad de ejecutar conjuntos de instrucciones guardadas en ficheros externos a la base de datos.

Todas estas características, junto con las operaciones más frecuentes que se realizan sobre bases de datos, se estudiarán en los siguientes Apartados. De todas formas, veamos en principio la manera de obtener el producto y, posteriormente, cómo realizar su instalación.

B

Instalación y configuración

Como se ha comentado en repetidas ocasiones ya, MySQL es un producto de libre uso, siempre y cuando no se utilice posteriormente con fines comerciales. Por este motivo, MySQL se encuentra disponible en Internet para que cualquier usuario se lo pueda descargar.

La dirección web de MySQL es www.mysql.com. Este sitio web está en inglés y tiene el aspecto que se aprecia en la Figura 8.2.



Fig. 8.2. Página principal de la Web oficial de MySQL.

Existen multitud de páginas dedicadas a MySQL, bastantes de ellas en castellano. Sin embargo, nosotros descargaremos el producto directamente desde la página principal de MySQL, pues al ser la oficial ofrece mayores garantías al respecto.



Puedes encontrar información en castellano sobre MySQL en multitud de páginas de Internet. Algunos ejemplos son:
www.desarrolloweb.com (taller de MySQL);
www.programacion.com; www.mysql-hispano.org;
www.lawebdelprogramador.com.



Descarga del producto

En la parte derecha de la página web www.mysql.com se puede ver un apartado denominado *Downloads* cuya primera opción es *MySQL Database*. Así pues, en primer lugar, haremos clic en este enlace.

8. Bases de datos y software libre. MySQL básico

8.1 MySQL: un SGBD de código abierto



A continuación, se abrirá una página donde aparece una exposición del producto, de las licencias de uso, de las novedades que ofrece la última distribución disponible y, finalmente, una exhaustiva relación de versiones a descargar, agrupadas por la plataforma a la que van dirigidas (Linux, Solaris, Windows, etc.). Por nuestra parte, trabajaremos con la versión de Windows; por tanto, iremos bajando la página hasta encontrar la sección *Windows Downloads* (véase la Figura 8.3).

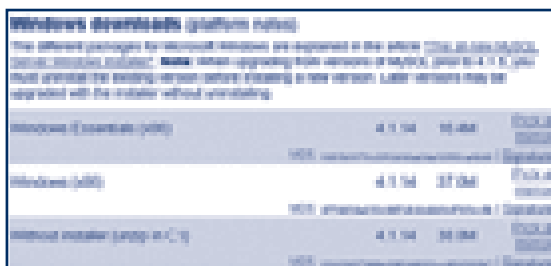


Fig. 8.3. Opciones de descarga de MySQL para Windows.

Como se puede apreciar, existen varias opciones. Por nuestra parte, elegiremos *Windows Essentials*, que es la versión más adecuada para el presente texto. Para empezar la descarga haremos clic en *Pick a mirror*, el enlace que puede verse a la derecha de la figura.



En el CD del alumno puedes encontrar el fichero de instalación de MySQL en su versión 4.1.14. Su nombre es `mysql-essential-4.1.14-win32.msi`.

Una vez seleccionado el producto que deseamos descargar, surgirá una pantalla de registro, que se puede pasar por alto siguiendo hacia abajo o haciendo clic en *Just take me to the downloads*, y aparecerá una lista con los servidores disponibles desde los que se puede bajar el fichero de instalación. Esta lista será similar a la que se reproduce en la Figura 8.4.



Fig. 8.4. Servidores de descarga de MySQL.

La Web ha detectado, a través de la IP de mi ordenador, que el servidor más próximo y seguramente el que más me interesa está en España: se trata de *RedIris*. Así pues, elegiremos dicho servidor y, a continuación, escogemos la opción de bajada por http pulsando en el enlace correspondiente.

Una vez elegida la opción *Guardar* del cuadro de diálogo, empezará la descarga del archivo.



Instalación

En primer lugar, es importante tener en cuenta que el fichero descargado sólo sirve para actualizar versiones de MySQL a partir de la 4.1.5. Si se dispone de una versión más antigua, entonces es necesario desinstalarla previamente.

Para instalar la nueva versión, buscaremos la carpeta en la que hemos guardado el fichero de instalación. Después, ejecutaremos dicho fichero haciendo doble clic sobre el mismo y, seguidamente, aparecerá una pantalla como la que se reproduce en la Figura 8.5.

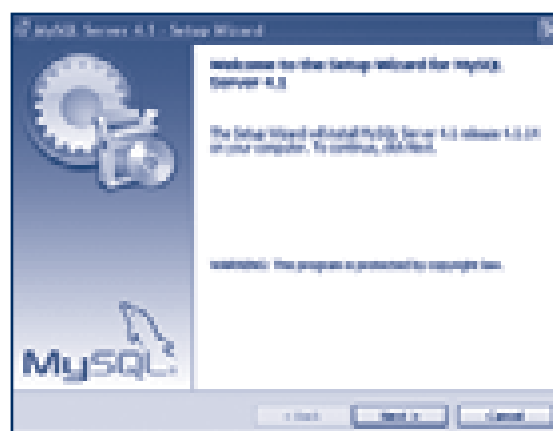


Fig. 8.5. Inicio de la instalación de MySQL.

De entre las diferentes modalidades de instalación que se nos ofrecen a continuación, escogeremos la opción *Typical*. Esta modalidad es la más básica y la que se recomienda para un uso general y personal de la aplicación.

Las otras posibilidades de instalación son *Complete*, mediante la cual se instalan todas las características del producto, y *Custom*, que deja al usuario la elección sobre qué partes instalar y cuáles no.



8. Bases de datos y software libre. MySQL básico

8.1 MySQL: un SGBD de código abierto

En la Figura 8.6 podemos apreciar la pantalla con los tres modos de instalación.

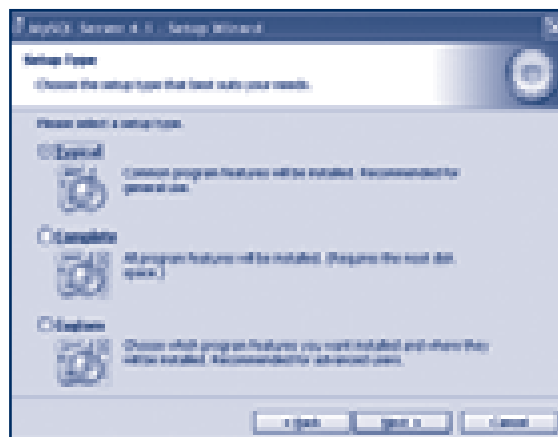


Fig. 8.6. Modos de instalación de MySQL.

Una vez pulsado el botón *Siguiente (Next)*, aparecerá una pantalla en la que se nos muestra un resumen de las opciones escogidas. En esta pantalla todavía disponemos de la posibilidad de detener la instalación, o bien dar un paso atrás. Si decidimos continuar, pulsaremos el botón *Instalar (Install)* para finalmente empezar la instalación propiamente dicha.

Tras las correspondientes pantallas de espera, una vez terminada la copia de ficheros a nuestro disco, se nos indicará la posibilidad de realizar un registro del producto: esta operación la dejaremos aplazada de momento. Así pues, procederemos a seleccionar *Saltar registro (Skip sign-up)*, tal como se puede observar en la Figura 8.7.

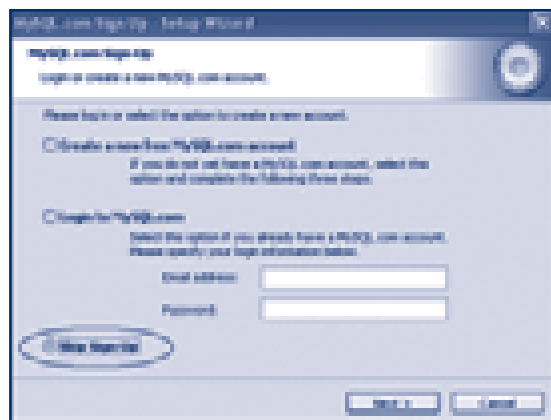


Fig. 8.7. Registro on-line de MySQL.

El registro del producto sirve, entre otras cosas, para tener acceso a contenidos especiales de la página web de MySQL. A continuación, aparecerá una pantalla donde se nos presentará la opción de configurar el servidor de MySQL. Tal como se aprecia en la Figura 8.8, dejaremos marcada la casilla correspondiente y pulsaremos *Terminar (Finish)* para que MySQL se configure y se finalice la instalación.

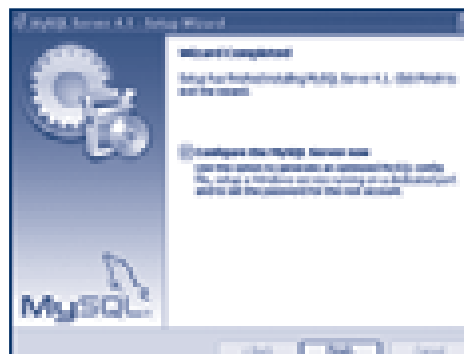


Fig. 8.8. Fin de la instalación, con opción a iniciar el proceso de configuración.

Configuración

Si hemos dejado marcada la casilla *Configurar ahora el servidor MYSQL (Configure the MySQL Server now)* de la última pantalla de instalación (véase la Figura 8.8), entonces en este momento empezará el proceso de configuración.

Conviene tener en cuenta que si tenemos versiones anteriores de MySQL instaladas en nuestra máquina, se nos ofrecerá la posibilidad de volver a configurar el producto, o bien desinstalarlo. De esta manera, se eliminarán los servicios de Windows correspondientes.

Tanto si se trata de la primera instalación, como si estamos volviendo a configurar una instalación anterior, a continuación aparecerá una pantalla que nos invita a decidir si queremos llevar a cabo una instalación *detallada (detailed)* o *estándar (standard)*.

- La **instalación detallada** propone una configuración automática y óptima para el servidor.
- La **instalación estándar** ofrece una configuración para un uso general del producto. Además, la instalación estándar permite seguir el proceso paso a paso y, si procede, hacer cambios en las opciones por defecto que se van ofreciendo.



En la Figura 8.9 se pueden ver ambas posibilidades de configuración. Como se observa, por nuestra parte hemos optado por la estándar.

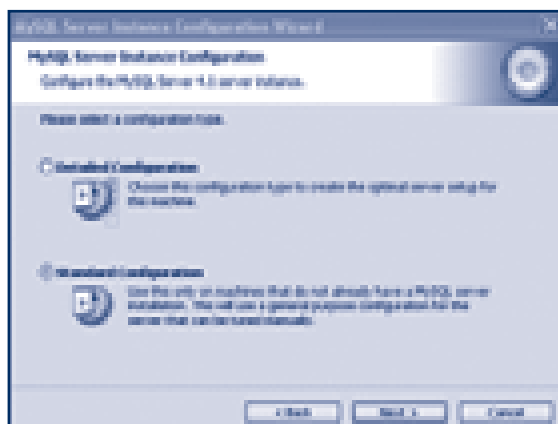


Fig. 8.9. Opciones de configuración de MySQL.

Al elegir el modo estándar, en primer lugar aparecerá una nueva pantalla con dos características a configurar. Por una parte, se nos pregunta si deseamos instalar MySQL como un servicio de Windows. Como hemos visto al estudiar otros productos, es conveniente efectuar la instalación de esta forma, puesto que así tenemos la opción de hacer que se inicie automáticamente cada vez que arranque Windows.

Por tanto, si se desea habilitar dicho inicio automático, entonces dejaremos marcada la casilla *Arrancar el servidor MySQL automáticamente (Launch the MySQL Server automatically)*, tal como aparece en la Figura 8.10. No obstante, si preferimos iniciar el servicio manualmente, entonces desmarcaremos dicha opción.

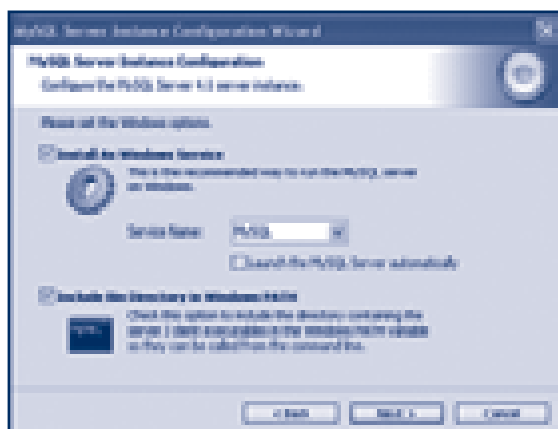


Fig. 8.10. Opciones de configuración del servidor MySQL.

Como se observa en la Figura 8.9, otra opción de configuración de la que disponemos es incluir el directorio *Bin* en el *PATH* de Windows. En este directorio se encuentran las instrucciones necesarias para ejecutar MySQL desde la línea de comandos. Esta casilla también la dejaremos marcada.



El **PATH** (termino inglés cuya traducción podría ser *camino*) es una variable que suelen tener todos los sistemas operativos. Contiene una lista de directorios o carpetas donde se buscará la información que no se encuentra en la carpeta actual, es decir, en la que nos hallamos en ese momento. Es muy útil sobre todo para, desde la línea de comandos, poder ejecutar instrucciones guardadas en ficheros que no se encuentran en la misma carpeta de trabajo.

A continuación, se nos brindará la posibilidad de asignar una contraseña al usuario que va a crearse. Dicho usuario se denominará *root* y será el administrador del sistema. Este nombre de usuario es muy común, sobre todo en sistemas multiusuario basados en UNIX.

En cuanto a la contraseña, se puede asignar cualquiera. Por nuestra parte, la contraseña que elegiremos será *obdoc*, la misma que hemos empleado en unidades anteriores (véase la Figura 8.11).

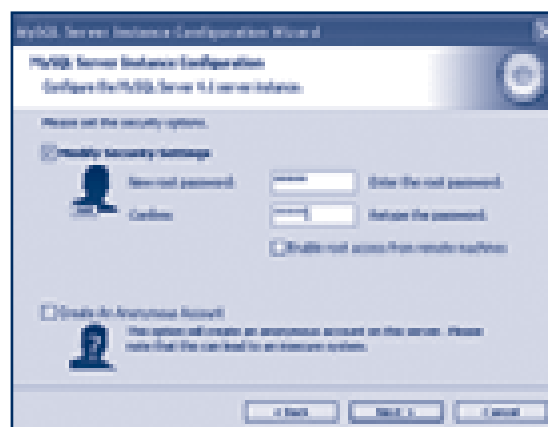


Fig. 8.11. Configuración del usuario root.

En esta misma pantalla se encuentran otras dos posibilidades más que el usuario puede configurar:

- Crear un usuario anónimo (sin nombre ni contraseña), opción que lógicamente rehusaremos dejando sin marcar la casilla correspondiente.



8. Bases de datos y software libre. MySQL básico

8.1 MySQL: un SGBD de código abierto

- Entrar a nuestro servidor desde máquinas remotas, a través de la red o incluso desde Internet (*Enable root access from remote machines*). Por obvios motivos, esta posibilidad tampoco la contemplaremos. Esta segunda opción deberá activarse sólo si se instala MySQL en un servidor de red al cual se accederá desde los diferentes terminales que se le conectan.

Una vez pulsado el botón *Siguiente (Next)*, aparecerá una pantalla similar a la que se reproduce en la Figura 8.12, donde se nos ofrece la posibilidad de volver atrás o cancelar la instalación. Por nuestra parte continuaremos y, para ello, pulsaremos el botón *Ejecutar (Execute)*.

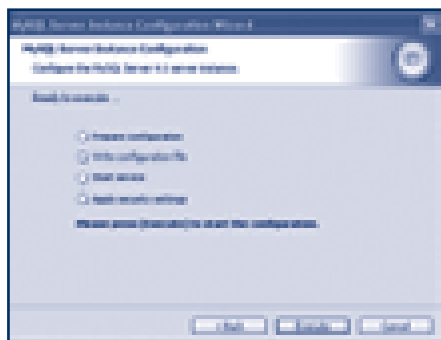


Fig. 8.12. Ejecución del proceso de configuración.

Tras pulsar el botón *Execute*, irán marcándose las casillas que se ven en la Figura 8.12 según se vayan realizando las distintas tareas.

Si surge algún problema, entonces se detendrá el proceso y aparecerá un aviso de error con una breve explicación de lo sucedido. Muchos de los errores que tienen lugar durante esta fase de la configuración provienen de instalaciones anteriores, con lo que la solución será desinstalar totalmente cualquier otra versión de MySQL que haya en el ordenador.

En otras ocasiones el problema puede deberse a la falta de permisos para crear el servicio, puesto que la instalación la debería realizar un usuario con permisos de administrador en el ordenador. El error también puede obedecer a la presencia de un servicio de MySQL anterior o incompatible con el que se intenta instalar. En este caso habría que eliminar los servicios que ocasionan el problema.

Los servicios de Windows se pueden eliminar de varias formas, pero no desde *Herramientas Administrativas/Ser-*

vicios. Una opción es ir al registro de Windows y buscar el siguiente apartado:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
Services
```

Desde este apartado localizaremos y eliminaremos el servicio correspondiente.

La otra opción es ir a *Inicio/Ejecutar* y teclear `cmd.exe` para salir del entorno MS-DOS. Seguidamente escribiremos `sc delete nombre-servicio ____` y, automáticamente, desaparecerá el servicio correspondiente. Así, por ejemplo, podríamos borrar el servicio de MySQL con `sc delete mysql ____`.

Finalmente, si todo el proceso se ha llevado a cabo de forma adecuada, debería aparecer una pantalla similar a la que se reproduce en la Figura 8.13.

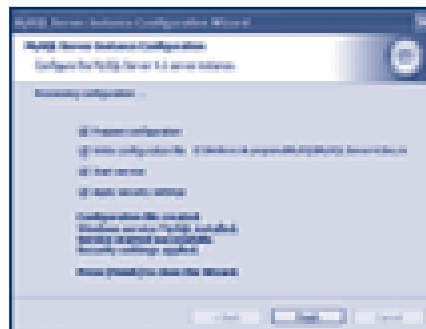


Fig. 8.13. MySQL se ha instalado y configurado correctamente.

Así pues, ya tenemos instalado y configurado nuestro servidor. En la Figura 8.14 puede observarse, dentro de la lista de servicios de Windows, el correspondiente a MySQL.

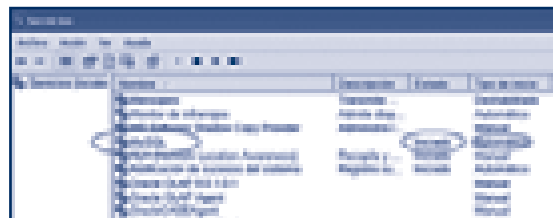


Fig. 8.14. El servicio Windows de MySQL.



Si no has elegido la opción de inicio automático, debes arrancar manualmente el servicio cada vez que quieras trabajar con MySQL.



Caso práctico



- 1 Si has ido realizando los pasos explicados en el Apartado anterior y todo ha funcionado correctamente, entonces tendrás MySQL instalado en tu ordenador. Sin embargo, vas a efectuar una sencilla comprobación.

Se pide

Comprueba si la variable de sistema *PATH* se ha actualizado, incluyendo ahora la referencia a las carpetas correspondientes a MySQL.

Solución

Hay dos formas de consultar el contenido de la variable de sistema *PATH*. Una de ellas es, tal como se hacía en MS-DOS, saliendo a la línea de comandos (*Inicio/Ejecutar/cmd.exe*) y, a continuación, tecleando *SET* que nos muestra el contenido de todas las variables del sistema. En la Figura 8.15 puede verse el resultado.



Fig. 8.15. Consulta de *PATH* desde la línea de comandos.

Dado que nos encontramos en Windows, también podemos consultar el valor de cualquier variable del sistema. Para ello, nos dirigiremos al panel de control, entraremos en *Sistema* y seleccionaremos seguidamente la pestaña *Opciones avanzadas*.

Como puede observarse en la Figura 8.16, en la parte inferior del recuadro aparece un botón con el título *Variables de entorno*. Si hacemos clic en dicho botón, veremos el contenido de las variables de sistema, tanto su valor en general como el valor actual para el usuario activo.

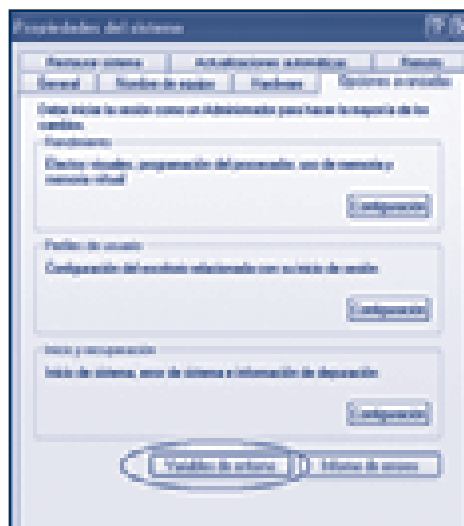


Fig. 8.16. Consulta de las variables de sistema desde Windows.

Un SGBD cliente/servidor

MySQL, como SGBD cliente/servidor que es, está formado por dos componentes:

- **Servidor.** Se encarga de recoger, interpretar y ejecutar todas las peticiones de los usuarios de la base de datos. Todas las tareas de configuración que hemos realizado se refieren a él.
- **Cliente.** Es la parte con la que trabajarán directamente los usuarios.

Ambos componentes pueden instalarse por separado en diferentes ordenadores, generalmente un servidor y varios clientes, si durante la instalación se elige la variante personalizada. Una vez presentado el modo de configurar el servidor, ha llegado el momento de abordar el componente *cliente*, que es el más importante de cara al usuario final.

En primer lugar, vamos a estudiar el programa cliente básico, que ofrece acceso a MySQL en modo texto desde una línea de comandos.



8.2 El cliente de MySQL en modo texto

El **cliente en modo texto** de MySQL es un programa que, a través de la línea de comandos, permite trabajar con las bases de datos en un entorno interactivo donde el usuario debe escribir las sentencias que desea ver ejecutadas.

A Entorno de trabajo

Veamos, para empezar, la manera de acceder al cliente de MySQL en modo texto.

Acceso al programa

El acceso al programa cliente en modo texto puede realizarse de dos formas distintas:

- A través del punto del menú que se ha creado durante la instalación y que puede apreciarse en la Figura 8.17.



Fig. 8.17. Acceso al cliente MySQL desde el menú.

Si optamos por este tipo de acceso, entraremos directamente como usuario *root*. Para completar el proceso sólo se nos pedirá la contraseña, como se muestra en la Figura 8.18. Conviene recordar que la contraseña, a no ser que se haya cambiado, es *obdoc*.

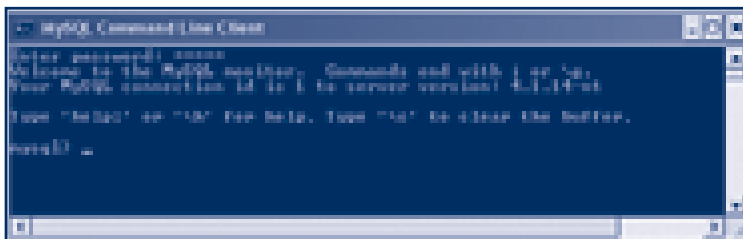


Fig. 8.18. Entrada al cliente MySQL desde el menú.

- Saliendo en primer lugar a la línea de comandos y, después, tecleando la siguiente orden:
`mysql -h servidor -u usuario -p`

En esta sentencia, servidor es el nombre del servidor al que nos conectamos (*localhost* si trabajamos en modo local) y *usuario* es el nombre de usuario con el que queremos entrar a la base de datos. Para entrar como local y con el usuario *root*, por ejemplo, deberíamos incorporar una instrucción como la que se reproduce en la Figura 8.19.



Fig. 8.19. Entrada al cliente MySQL desde la línea de comandos.

Si escogemos la segunda opción, podremos seleccionar tanto el servidor como el usuario. Sin embargo, al elegir la primera posibilidad, accederemos directamente a la máquina local y como usuario *root*, que tiene permisos de administrador. Por tanto, esta opción automática no debería estar accesible para los usuarios finales.

Conviene tener en cuenta que la instrucción `mysql` se encuentra en la carpeta `c:\Archivos de programa\MySQL\MySQL Server 4.1\bin`, pero nosotros hemos podido ejecutarlo desde `C:\` sin ningún problema. Ése era precisamente el objetivo de la modificación que se ha realizado sobre la variable *PATH*, tal como hemos visto anteriormente.

Entorno de trabajo

Si la conexión ha sido correcta, el indicador de sistema habrá cambiado y ahora aparecerá como `mysql>`. Así pues, podremos empezar a trabajar sobre el SGBD.

Desde el indicador ejecutaremos normalmente sentencias SQL como las que se han presentado en unidades anteriores del libro.

Por tanto, podremos consultar, por ejemplo, la versión de MySQL con la que estamos trabajando y, al mismo tiempo, la fecha actual. La versión de MySQL nos la proporciona la función `version()`, mientras que la fecha actual se guarda en la variable del sistema `current_date`.



Como ya sabes por unidades anteriores, en una sentencia SQL pueden utilizarse tanto funciones del lenguaje, como es el caso de `version()`, como variables del sistema, por ejemplo `current_date`.

De esta manera, la sentencia completa que deberíamos introducir sería la que se aprecia en la Figura 8.20.

```
mysql> select version(), current_date;
+-----+-----+
| version() | current_date |
+-----+-----+
| 4.1.14-nt | 2005-09-25   |
+-----+-----+
1 row in set (0.00 sec)
```

Fig. 8.20. Consulta de variables en MySQL.

Las sentencias SQL deben acabar con un punto y coma. En caso contrario, dicha sentencia se considera inacabada y aparece una línea nueva cuyo indicador es una flecha. En esta línea terminaremos de escribir la sentencia que ha quedado incompleta, tal como puede observarse en la Figura 8.21.

```
mysql> select version(), current_date
-> ;
+-----+-----+
| version() | current_date |
+-----+-----+
| 4.1.14-nt | 2005-09-25   |
+-----+-----+
1 row in set (0.00 sec)
```

Fig. 8.21. Finalización de una sentencia incompleta.

Por otra parte, si queremos abandonar la sentencia sin que ésta se ejecute, lo conseguiremos escribiendo `\c`. A modo de ejemplo, en la Figura 8.22 aparece una consulta donde se ha cometido un error y se cancela sin ejecutarse.

```
MySQL Command Line Client
mysql>
mysql>
mysql> select version, current_date
-> \c
mysql>
mysql>
```

Fig. 8.22. Interrupción de una sentencia.

Finalmente, si deseamos salir de MySQL volviendo a la línea de comandos de Windows, podemos usar por igual `exit` o `quit`.

B Uso de bases de datos en modo texto

Ahora que ya conocemos el entorno de trabajo, es hora de que presentemos el modo de acceder a las bases de datos para comenzar a utilizarlas.

En primer lugar, es importante conocer el sitio donde se guardan las bases de datos que iremos creando, así como el fichero de configuración en el cual se guarda la información acerca de dicha ubicación.



Localización de las bases de datos

Las bases de datos de MySQL se guardan, en entornos Windows, en la carpeta `c:\Archivos de Programa\MySQL\MySQL Server 4.1\Data`, su ubicación por defecto, como puede apreciarse en la Figura 8.23. Sin embargo, dicha ubicación puede cambiarse. Para ello deberíamos:

- Copiar todo el contenido de la carpeta *Data* a la nueva localización.
- Modificar el fichero `my.ini` que se encuentra en la misma carpeta que *Data*, tal como también se observa en la Figura 8.23. Para ello, nos dirigiremos a la sección *SERVER SECTION* y cambiaremos el valor de la variable *Datadir* (véase la Figura 8.24).

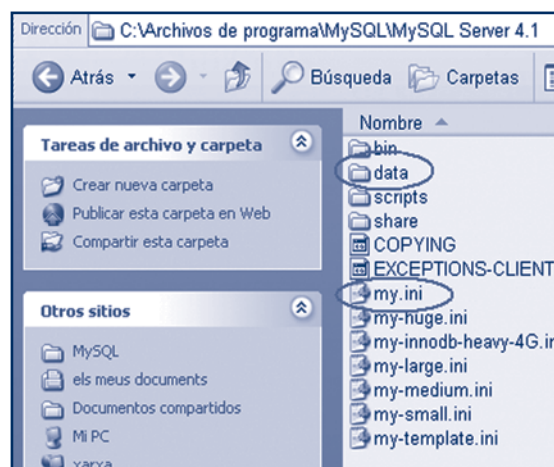


Fig. 8.23. Localización de la carpeta *Data* y el fichero de configuración `my.ini`.

8. Bases de datos y software libre. MySQL básico

8.2 El cliente de MySQL en modo texto

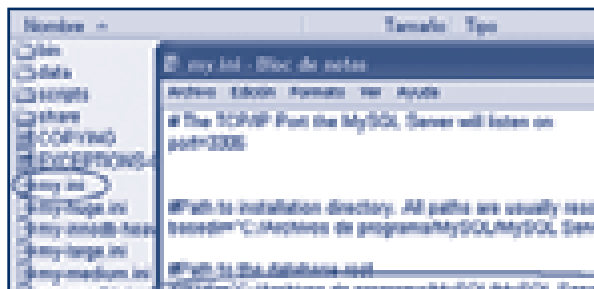


Fig. 8.24. *Contenido del fichero my.ini.*

Dentro de la carpeta *Data* se irán creando diversas carpetas para las diferentes bases de datos que se vayan dando de alta. Por defecto, MySQL incluye una base de datos denominada *mysql* que guarda información sobre el propio sistema. En la Figura 8.25 puede observarse la subcarpeta correspondiente a dicha base de datos.



Fig. 8.25. Subcarpetas para las bases de datos.

Utilización de las bases de datos

Lógicamente, deberemos saber qué bases de datos están disponibles antes de poder empezar a usarlas. Para obtener una lista de las bases de datos accesibles al usuario, usaremos la sentencia `SHOW DATABASES`. El resultado, en nuestro caso, es el que se muestra en la Figura 8.26.

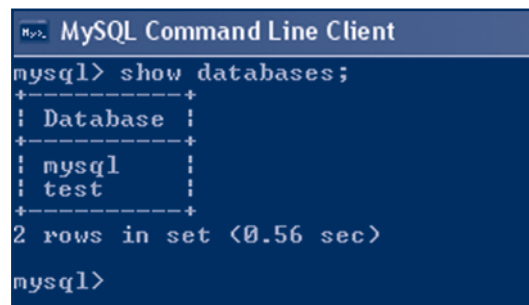


Fig. 8.26. Lista de bases de datos disponibles.

Como se aprecia en esta figura, el número de bases de datos, así como sus nombres, coincide con el de subcarpetas dentro de *Data*, algo que ya se podía prever.

De todas formas, ya sabemos de unidades anteriores que las bases de datos están formadas por tablas y que son éstas las que contienen la información de la base de datos. Así pues, para realizar una consulta deberemos siempre conocer el nombre de las tablas en las que se encuentra la información que queremos recuperar.

Existe una sentencia para ver las tablas que forman una base de datos, pero para ello debemos, en primer lugar, abrir dicha base de datos.



Al contrario de otros sistemas como Oracle, la conexión al SGBD no está asociada a la apertura de una base de datos en concreto, sino que este proceso se realiza con posterioridad a la conexión.

La instrucción que permite abrir una base de datos es `USE`, seguida del nombre de la base de datos que se quiere abrir. Otra posibilidad que veremos más adelante consiste en utilizar el comando `CONNECT`. En la Figura 8.27 se muestra cómo se abre la base de datos *mysql*.

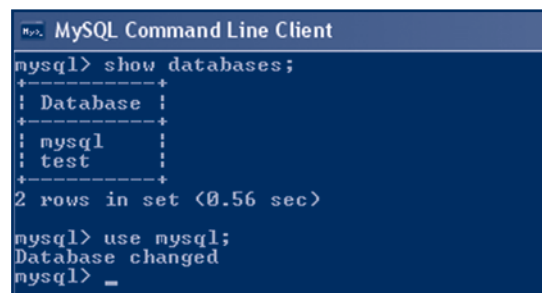


Fig. 8.27. *Apertura de una base de datos.*

El mensaje *Database changed* nos indica que la apertura de la base de datos se ha realizado correctamente.

Una vez abierta la base de datos *mysql*, estamos en condiciones de obtener una lista de las tablas que dicha base de datos contiene. Para ello, usaremos la instrucción `SHOW TABLES` sin ningún parámetro adicional, ya que no se necesita hacer referencia a la base de datos, puesto que sólo podremos listar las tablas que previamente hemos abierto y están contenidas en dicha base de datos. En la Figura 8.28 se muestra la lista de la base de datos en cuestión obtenida tras este proceso.



```
mysql> select * from user;
```

Ahora bien, para realizar una consulta sobre un número menor de campos deberemos, en primer lugar, conocer la estructura de la tabla. En MySQL disponemos de la instrucción `DESCRIBE` que proporciona información sobre los campos que forman una tabla determinada. Por ejemplo, para ver los campos que forman la tabla `user`, escribiríamos:

```
mysql> describe user;
```

En cualquier caso, al visualizar la estructura de la tabla *user* podemos encontrar, al principio, los campos *Host*, *User* y *Password*. En la Figura 8.29 se reproduce el resultado de la consulta sólo con estos tres campos.



Las bases de datos que se han presentado hasta ahora contienen información sobre el sistema, así que es mejor no trabajar con ellas. Vamos a recordar las instrucciones de creación y manipulación, tanto de bases de datos como de tablas, así como su utilización en MySQL.

Las instrucciones para crear bases de datos y tablas no difieren de las que se han estudiado en las unidades correspondientes al lenguaje SQL. Así pues, no nos extenderemos demasiado en este Apartado, donde vamos a presentar algunos ejemplos simplemente como recordatorio, al tiempo que creamos una base de datos nueva que nos servirá para trabajar durante el resto de la Unidad.

La sentencia para crear una base de datos, como ya se sabe, es `CREATE DATABASE`. Así pues, vamos a dar de alta una nueva base de datos, que denominaremos *pruebas*. A continuación, volveremos a ejecutar `SHOW DATABASES` para comprobar si, efectivamente, nuestra nueva base de datos aparece en la lista. Esta última operación se puede apreciar en la Figura 8.30.





8. Bases de datos y software libre. MySQL básico

8.2 El cliente de MySQL en modo texto

Si abrimos la base de datos mediante `use` y ejecutamos `SHOW TABLES`, comprobaremos que la base de datos está vacía, ya que la acabamos de crear. Para dar de alta una tabla usaremos otra sentencia que ya se conoce: `CREATE TABLE`.



Al crear una tabla debes enumerar los campos que la formarán, así como el nombre y el tipo de cada campo.

Es conveniente conocer los tipos de datos que podemos usar en MySQL, a pesar de que no varían sustancialmente respecto a los que se han ido presentando a lo largo de este manual. No obstante, en la Tabla 8.1 presentamos los tipos de datos más importantes.

Tipo	Descripción
<code>varchar (n)</code>	Puede contener cualquier carácter. Longitud variable con un tamaño máximo de <i>n</i> bytes, siendo el límite para <i>n</i> de 65536
<code>char (n)</code>	Puede contener cualquier carácter, pero con longitud fija de <i>n</i> bytes. Los espacios no ocupados se rellenan con blancos. Valor máximo de <i>n</i> es 255
<code>date</code>	Una fecha
<code>datetime</code>	Una fecha y una hora
<code>time</code>	Una hora
<code>year (n)</code>	Un año con <i>n</i> dígitos (2 o 4)
<code>bit (n)</code>	Un número de <i>n</i> bits. Sólo en versiones de MySQL superiores a la 5
<code>bool</code>	Lógico: 0 falso, 1 verdadero
<code>tinyint (n)</code>	Un entero con un máximo de <i>n</i> cifras. Valores entre -128 y 127
<code>smallint (n)</code>	Un entero con un máximo de <i>n</i> cifras. Valores entre -32768 y 32767
<code>mediumint (n)</code>	Un entero con un máximo de <i>n</i> cifras. Valores entre -8388608 y 8388607
<code>int (n)</code>	Un entero con un máximo de <i>n</i> cifras. Valores entre -2147483648 y 2147483647
<code>bigint (n)</code>	Un entero con un máximo de <i>n</i> cifras. Para valores no incluidos en los intervalos anteriores
<code>float (n, d)</code>	Un número real con <i>n</i> cifras de las cuales <i>d</i> son decimales. Valores entre -3.402823466E+38 y 3.402823466E+38
<code>double (n, d)</code>	Un número real con <i>n</i> cifras de las que <i>d</i> son decimales. Para valores con mayor precisión que los anteriores, o no incluidos en el intervalo
<code>enum ('a', 'b', 'c'...)</code>	Puede incluir cualquiera de los elementos enumerados en la lista

Tabla 8.1. Tipos de datos en MySQL.

Al igual que ocurre con otros lenguajes, los tipos pueden combinarse con otros atributos como, por ejemplo, `UNSIGNED`, que indica que sólo se admiten valores positivos, o bien `ZEROFILL`, que hace que los espacios en blanco a la izquierda se rellenen con ceros.

A continuación, presentamos un Caso práctico que nos servirá para consolidar los conocimientos adquiridos.



Caso práctico

2 Vas a repasar la creación de tablas.

Se pide

En la base de datos *pruebas* debes crear dos tablas, *alumnos* y *asignaturas*, con la siguiente descripción:

```
alumno(#idalu, nomalu, fnacimiento)
asignatura(#idasi, nomasi)
```

Solución

Con los conocimientos que hemos adquirido hasta el momento, podemos crear las tablas con las siguientes instrucciones:

```
mysql> create table alumnos
        (idalu int(4) unsigned,
         nomalu varchar(30),
         fnacimiento date);
mysql> create table asignatura
        (idasi smallint(2) unsigned,
         nomasi varchar(20));
```

La solución propuesta es sólo una de las posibles soluciones a este caso. Es probable que se eche en falta la presencia de claves primarias, puesto que la definición de claves primarias, foráneas y otras restricciones similares se presentarán más adelante.



Introducción y manipulación de datos

Las instrucciones para introducir y manipular datos en una tabla han sido ampliamente estudiadas en unidades anteriores. Sin embargo, aprovecharemos este punto para recordar las más importantes en la siguiente tabla.

8. Bases de datos y software libre. MySQL básico

8.2 El cliente de MySQL en modo texto



Instrucción	Función
INSERT INTO...	Introducir nuevas filas en la tabla
DELETE FROM...	Borrar filas en la tabla
UPDATE...	Modificar filas en la tabla

Tabla 8.2. Instrucciones de introducción y modificación de filas.

No vamos a insistir en la sintaxis y el uso de estas instrucciones. Nos parece más importante incidir en aquellos aspectos en los que MySQL dispone de alguna funcionalidad añadida o diferente.

Así pues, para insertar filas en una tabla podemos, además de hacerlo manualmente con `INSERT`, aprovechar la instrucción `LOAD DATA INFILE...INTO TABLE...` que permite la importación de datos contenidos en un fichero de texto. Los requisitos para que esta sentencia funcione son los siguientes:

- El fichero de texto debe estar en la misma carpeta que la tabla a la cual desea importarse.
- En el fichero de texto debe haber un tabulador de separación entre cada columna.

Supongamos que disponemos del fichero `asig.txt` con el formato que puede apreciarse en la Figura 8.31.



1	OBD OC
2	FOL
3	RET
4	IMESI
5	IMAOC

Fig. 8.31. El fichero de texto `asig.txt`.

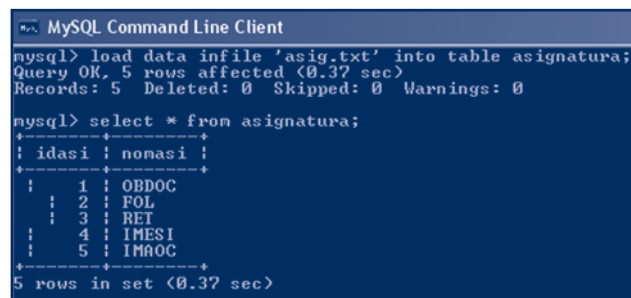
El fichero debe estar guardado en la misma carpeta *pruebas*, dentro de *Data*. El espacio entre las dos columnas ha de insertarse pulsando el tabulador y no la barra espaciadora.

A continuación, nos dirigiremos a MySQL, abriremos la base de datos *pruebas*, y teclearemos:

```
mysql> load data infile 'asig.txt' into table
asignatura;
```

Como se puede observar, el nombre del fichero de texto va entre comillas simples, mientras que el nombre de la tabla se introduce sin ningún delimitador. Si el proceso se realiza correctamente, aparecerá por pantalla el número de filas que se han añadido a la tabla.

En la Figura 8.32 se puede apreciar el proceso completo, así como la posterior comprobación.



```
mysql> load data infile 'asig.txt' into table asignatura;
Query OK, 5 rows affected (0.37 sec)
Records: 5 Deleted: 0 Skipped: 0 Warnings: 0

mysql> select * from asignatura;
+----+-----+
| idasi | nonasi |
+----+-----+
| 1 | OBD OC |
| 2 | FOL |
| 3 | RET |
| 4 | IMESI |
| 5 | IMAOC |
+----+-----+
5 rows in set (0.37 sec)
```

Fig. 8.32. Importación de filas desde un archivo.

Si la importación da problemas, en ocasiones se resuelven añadiendo en el fichero de texto un carácter al final de cada línea que sirva para marcar el salto a la siguiente. Si optamos por esta posibilidad, entonces la instrucción de importación queda de la siguiente manera:

```
LOAD DATA INFILE 'fichero.txt' INTO TABLE tabla
LINES TERMINATED BY 'carácter';
```

Como es lógico, deberá usarse como marcador de final de línea un carácter que no pueda encontrarse en otro punto de la fila, lo cual llevaría a confusión y a una errónea importación de los datos.

Si recordamos el funcionamiento de `INSERT`, la importación que acabamos de hacer sería equivalente a haber tecleado desde la línea de comandos la siguiente instrucción:

```
mysql> insert into asignatura
values(1,'OBD OC'), (2,'FOL'), (3,'RET'), (4,'IMESI')
, (5,'IMAOC');
```

Veamos seguidamente cómo implementar las claves, tanto primarias como ajenas, así como otras restricciones de uso común.



8. Bases de datos y software libre. MySQL básico

8.2 El cliente de MySQL en modo texto

C Restricciones en MySQL

En MySQL, como en otros sistemas gestores de bases de datos, existe la posibilidad de incorporar ciertas restricciones y propiedades especiales en las definiciones de los campos de una tabla. Así pues, a continuación vamos a analizar las restricciones más frecuentes, empezando por aquellas de mayor importancia: las claves primarias y ajenas.

Claves primarias

Para definir la clave primaria de una tabla se usa la expresión **PRIMARY KEY**, que puede introducirse en la misma línea de definición del campo que hará de clave, o bien al final de la declaración de campos. No obstante, cuando la clave primaria esté formada por más de un campo, será obligatorio definirla al final.

Así pues, para definir la clave primaria de la tabla *alumnos* que hemos creado en el Caso práctico 2, deberíamos haber escrito la instrucción de la siguiente manera:

```
mysql> create table alumnos
      (idalu int(4) unsigned primary key,
       nomalu varchar(30),
       fnacimiento date);
```

O también:

```
mysql> create table alumnos
      (idalu int(4) unsigned,
       nomalu varchar(30),
       fnacimiento date,
       primary key (idalu));
```

En nuestro caso, la tabla ya está creada. Para añadirle la clave primaria, sin borrarla y volverla a crear, se puede emplear el comando **ALTER TABLE**, que permite modificar la estructura de una tabla. Además, se usará la cláusula **ADD**.

Por tanto, la instrucción sería la siguiente:

```
mysql> alter table alumnos add primary key
      (idalu);
```

En la Figura 8.33 puede verse el proceso de adición de la clave primaria a la tabla y, posteriormente, la com-



Fig. 8.33. Creación de una clave primaria.

probación mediante la cláusula **DESCRIBE**.

Claves ajenas

Como se vio en las primeras unidades del libro, la fun-



La **clave ajena** hace referencia a un campo que se añade a una tabla para hacer de enlace con otra. Dicho campo hará la función de clave primaria en la tabla referenciada.

ción de las claves ajenas consiste en garantizar la integridad referencial entre tablas.

Supongamos que disponemos de una tabla *provincia* cuya clave primaria es *idprov*. Además, tenemos una tabla *ciudad* que queremos enlazar con la tabla *provincia* para que quede constancia de la provincia a la que pertenece cada ciudad. Así pues, añadiríamos a la tabla *ciudad* un campo, denominado también *idprov*, que haría referencia a la clave primaria de la tabla *provincia* y que en la tabla *ciudad* realizaría la función de clave ajena.

La integridad referencial entre ambas tablas se garantizaría si, además de saber a qué provincia corresponde cada ciudad, se asegura que no podrá asignarse un código de provincia a una ciudad si previamente dicha provincia no existe en la tabla *provincias*.

En la definición del campo, las claves ajenas se especifican en MySQL mediante la cláusula **REFERENCES** y, a continuación, el nombre de la tabla referenciada con su clave primaria entre paréntesis. Siguiendo con nuestro ejemplo, se debería escribir:

```
mysql> create table ciudad
      (idciud int(2) unsigned primary key,
```



```
nomciud varchar(25),
idprov int(2) unsigned references
    provincia(idprov));
```

Lógicamente, para poder hacer referencia a la tabla *provincia*, ésta debe haber sido declarada previamente. Como regla general, es conveniente declarar en primer lugar aquellas tablas que no contienen claves ajenas.

Es realmente importante tener en cuenta que los campos enlazados (la clave primaria y la clave ajena) deben ser exactamente del mismo tipo. Por tanto, si *idprov* se hubiera definido como `int(2) UNSIGNED` en la tabla *provincia*, y como `int(2)` en la tabla *ciudad*, no se aceptaría la restricción de clave ajena y aparecería un error.

De la misma forma que ocurría con las claves primarias, las claves ajenas pueden definirse al final de la tabla. Sin embargo, cuando la clave ajena está formada por varios campos, entonces es obligatorio hacerlo de esta forma. Además, si la definición se incluye al final de la tabla, deberá incluirse la cláusula `FOREIGN KEY`. Así pues, en el caso anterior se debería escribir:

```
mysql> create table ciudad
(idciu int(2) unsigned primary key,
nomciu varchar(25),
idprov int(2) unsigned,
foreign key (idprov) references
provincia(idprov));
```

Así mismo, el campo o campos que forman parte de la clave ajena deben estar indexados, para lo que usaremos la cláusula `INDEX` y el nombre del campo o campos que forman el índice entre paréntesis. De esa forma, la creación de la tabla *ciudad* quedaría, de manera definitiva, como sigue:

```
mysql> create table ciudad
(idciu int(2) unsigned primary key,
nomciu varchar(25),
idprov int(2) unsigned,
index(idprov),
foreign key (idprov) references
provincia(idprov));
```

Aún queda otra cuestión importante por resolver: en MySQL las tablas pueden ser de varios tipos, entre ellos *MyIsam* o *InnoDB*. Aunque más adelante se volverá sobre esta cuestión, es importante saber que las claves ajenas sólo funcionarán si las tablas son de tipo *InnoDB*,

al menos en la versión de MySQL que estamos utilizando. Es necesario realizar la comprobación, puesto que por defecto MySQL crea las tablas como *MyIsam*.

Otras restricciones

En la definición de campo es posible incluir otras restricciones, además de las de clave. Las más importantes son:

- **Valor único.** En ocasiones interesa que un campo que no pertenece a una clave primaria tenga, de todas formas, un valor único para cada fila de la tabla.

Podemos pensar, por ejemplo, en el NIF de un fichero de clientes.

En estos casos se utiliza la cláusula `UNIQUE`. He aquí el resultado:

```
mysql> create table cliente
(idCli int(4) unsigned primary key,
CliNom varchar(25),
CliPob varchar(25),
CliNif varchar(15) unique);
```

- **Autoincrementable.** Muchas veces la clave principal de una tabla es simplemente un código que no tiene mayor significado que la identificación de cada elemento.

Si es así, puede ser interesante hacer que dicho código se vaya incrementando de forma automática cada vez que se da de alta una fila. Así, el primer cliente tendría el código *1*, el segundo tendría el *2*, y así sucesivamente, sin intervención del usuario. Para ello se usa la cláusula `AUTO INCREMENT` en la definición del campo. He aquí el resultado:

```
mysql> create table cliente
(idCli int(4) unsigned primary key

auto_increment,
CliNom varchar(25),
CliPob varchar(25),
CliNif varchar(15) unique);
```

- **Valor nulo.** Cuando un campo es clave primaria, no puede tener en absoluto un valor nulo, es decir, dejarse sin valor (no se debe confundir con valor *0*, o con espacios en blanco). Sin embargo, esa misma restricción puede aplicarse también a campos no



8. Bases de datos y software libre. MySQL básico

8.2 El cliente de MySQL en modo texto

clave, mediante la cláusula `NOT NULL`.

Pensemos, por ejemplo, en que sea obligatorio poner un valor al NIF cuando se da de alta un cliente. He aquí el resultado:

```
mysql> create table cliente
      (idCli int(4) unsigned primary key
      auto_increment,
      CliNom varchar(25),
      CliPob varchar(25),
      CliNif varchar(15) not null
      unique);
```

- **Valor por defecto.** Al dar de alta una fila en una tabla, y antes de introducir algún valor, todos los campos tienen el valor `NULL` (vacío).

A veces, existe un campo que tendrá siempre el mismo valor en todas las filas, salvo contadas excepciones.

Por ejemplo, si en nuestro fichero de clientes sabemos que la gran mayoría son de Valencia, podemos hacer que el campo `CliPob` asuma, por defecto, el valor *Valencia* sin tener que escribirlo cada vez. Para ello, se usa la cláusula `DEFAULT` en la definición del campo. He aquí el resultado:

```
mysql> create table cliente
      (idCli int(4) unsigned primary key
      auto_increment,
      CliNom varchar(25),
      CliPob varchar(25) default
      'Valencia',
      CliNif varchar(15) not null
      unique);
```

Es interesante conocer una particularidad del tipo `enum`: al declarar el valor por defecto, si se pone éste entre comillas simples, hará referencia al elemento de la lista cuyo valor coincida con el indicado; si, por el contrario, el valor por defecto consiste en un número sin comillas simples, entonces se está haciendo referencia al número de orden en la lista de `enum`.

Imaginemos que creamos una tabla de hoteles con un campo *categoría* cuyo valor puede ser 3, 4 o 5 y que queremos que, por defecto, asuma el valor 3. Esto se podría conseguir indistintamente mediante dos formas:

```
...categoría enum('3','4','5') default '3'
...categoría enum('3','4','5') default 1
```

En ambos casos, el valor por defecto será 3.



Caso práctico

- 3 Vas a aplicar los conceptos presentados hasta ahora a las tablas creadas en el Caso práctico 2. Recordemos cuáles eran:

```
mysql> create table alumnos
      (idalu int(4) unsigned,
      nomalu varchar(30),
      fnacimiento date);
mysql> create table asignatura
      (idasi smallint(2) unsigned,
      nomasi varchar(20));
```

Se pide

En la base de datos *pruebas*, debes modificar la tabla *alumnos* de forma que el campo *idalu* sea clave primaria y autoincrementable, mientras que la fecha de nacimiento sea obligatoria. Además, debes modificar la tabla *asignatura* para que el campo *idasi* sea también clave primaria y autoincrementable.

Finalmente, para saber de qué asignaturas se ha matriculado cada alumno, es preciso crear la tabla *matrícula*. Para ello, deberás basarte en la siguiente descripción:

```
matrícula(#idalu, #idasi)
      alumnos      asignatura
```

Solución

En primer lugar, para modificar la estructura de una tabla, se deberá usar `ALTER TABLE` junto con la cláusula `ADD` si estamos añadiendo elementos.

En nuestro caso, se añadirá a la tabla *alumnos* la propiedad de clave primaria para el campo *idalu*. Además, se modificarán los campos *idalu* y *fnacimiento*, el primero para hacerlo autoincrementable y el segundo para que no acepte el valor nulo.

Así pues, para modificar cualquier propiedad de un campo que no sea `PRIMARY KEY`, usaremos `ALTER TABLE` con la cláusula `MODIFY`.



Caso práctico (continuación)



Al modificar un campo con `ALTER TABLE` se deberá repetir toda la especificación de dicho campo, incluso aquellas características que no cambian. Por tanto, la sentencia para realizar los cambios en *alumnos* se correspondería con la que se muestra en la Figura 8.34.

```
MySQL Command Line Client
mysql> alter table alumnos
-> add primary key(idalu),
-> modify idalu int(4) auto_increment,
-> modify fnacimiento date not null;
Query OK, 0 rows affected (0.66 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Fig. 8.34. Modificación de la tabla *alumnos*.

A continuación, aplicaremos el mismo método para modificar la tabla *asignatura*, tal como se aprecia en la Figura 8.35.

```
MySQL Command Line Client
mysql> alter table asignatura
-> add primary key(idasi),
-> modify idasi smallint(2) auto_increment;
Query OK, 5 rows affected (0.67 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

Fig. 8.35. Modificación de la tabla *asignatura*.

Para finalizar, crearemos la tabla *matrícula* con sólo dos campos: el *código de alumno* y el *código de asignatura*. Ambos serán claves ajenas respecto a las tablas *alumnos* y *asignatura*, respectivamente. Además, juntos formarán la clave primaria de la tabla *matrícula*.

Recordemos que un campo debe estar indexado para poder ser clave ajena. Teniendo en cuenta esta premisa y aplicando los conocimientos adquiridos en esta unidad, la sentencia de creación de la tabla *matrícula* se correspondería con la que se reproduce en la Figura 8.36.

```
MySQL Command Line Client
mysql> create table matricula
-> (idalu int(4) not null,
-> idasi smallint(2) not null,
-> primary key(idalu, idasi),
-> foreign key(idalu)
-> reference alumnos(idalu),
-> foreign key(idasi)
-> reference asignatura(idasi));
Query OK, 0 rows affected (0.66 sec)
```

Fig. 8.36. Modificación de la tabla *matrícula*.

La sentencia que se reproduce en la Figura 3.36 cuenta con algunas particularidades destacables.

En primer lugar, se ha definido la clave primaria después de la declaración de campos, puesto que en este caso la clave está formada por más de un campo.

En segundo lugar, se ha creado un índice para cada uno de los campos que vamos a definir como clave ajena.

Finalmente, se han añadido las cláusulas `FOREIGN KEY` al final para definir las claves ajenas. Además, `REFERENCES` no se ha puesto al lado de cada campo porque antes deben crearse los índices: no puede definirse una clave ajena si previamente el campo no está indexado.

Para terminar con este Apartado, conviene tener presente una singularidad más de la instrucción `ALTER TABLE`.

Si se desea cambiar el nombre de un campo, no utilizaremos la cláusula `MODIFY`, sino `CHANGE` seguida del antiguo nombre del campo, el nombre nuevo y el resto de propiedades.

Para cambiar, por ejemplo, el nombre al campo *fnacimiento* y dejarlo como *fnac*, la modificación se debería realizar tal como se muestra en la Figura 8.37.

```
MySQL Command Line Client
mysql> alter table alumnos
-> change fnacimiento fnac date not null;
Query OK, 0 rows affected (0.63 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe alumnos;
+----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default |
+----+-----+-----+-----+-----+
| idalu | int(4) | YES | PRI | NULL |
| nomalu | varchar(30) | YES | | NULL |
| fnac | date | YES | | 0000-00-00 |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Fig. 8.37. Modificación del nombre de un campo.



8.3 Seguridad en MySQL

Como casi siempre cuando se mencionan bases de datos, los principales aspectos de la seguridad están relacionados con el control de los accesos a la información, además de la protección de ésta. En este Apartado abordaremos cómo, de manera básica, se pueden realizar ambas tareas desde el mismo cliente texto de MySQL.

A Gestión de usuarios

La gestión de usuarios en MySQL es muy sencilla y no difiere prácticamente de lo que se ha ido presentando en unidades anteriores.

Este hecho obedece a que las sentencias SQL que ya conocemos se utilizan también en MySQL, así como los permisos típicos.

En cualquier caso, conviene repasar el uso de dichas sentencias, haciendo hincapié en las particularidades del producto que se está tratando.

Creación de usuarios en MySQL

A partir de la versión 5 de MySQL se podrán crear usuarios mediante la sentencia `CREATE USER` típica, con la sintaxis siguiente:

```
CREATE USER user [IDENTIFIED BY [PASSWORD]
'password']
[, user [IDENTIFIED BY [PASSWORD] 'password']]
...
```

Como se puede apreciar, simplemente se indica el nombre de usuario y, de forma opcional, la contraseña. Sin embargo, en las versiones anteriores a la 5 como, por ejemplo, la que nosotros estamos utilizando, no existe todavía esta opción.

Por tanto, debemos buscar otro método para crear usuarios. Mediante la siguiente sentencia este problema queda resuelto:

```
GRANT tipo-permiso [(lista-columnas)]
ON {tabla | * | *.* | base-datos.*}
TO usuario [IDENTIFIED BY 'contraseña']
```

```
[WITH GRANT OPTION];
```

Como se ha repetido en numerosas ocasiones, la sentencia `GRANT` se emplea para asignar permisos a usuarios ya existentes. No obstante, en MySQL el usuario se crea automáticamente en caso de no existir. En esta versión, esta posibilidad es la única de la que disponemos para crear usuarios nuevos, a no ser que nos atrevamos a manipular directamente la tabla `user`.

Antes de mostrar un ejemplo, en la Tabla 8.3 presentamos los permisos que pueden asignarse en MySQL.

Veamos un ejemplo de esta instrucción haciendo uso de un Caso práctico.

Al igual que en otros SGBD, la sentencia que permite retirar permisos a los usuarios es `REVOKE`, cuya sintaxis es la siguiente:

```
REVOKE tipo-permiso [(lista-columnas)]
ON {tabla | * | *.* | base-datos.*}
FROM usuario;
```

Los tipos de permiso son los mismos que se presentan en la Tabla 8.3.

Permiso	Operaciones permitidas
ALL	Todas
USAGE	Ninguna
ALTER	Modificar estructura
CREATE	Crear
DELETE, DROP	Borrar columnas/tablas
INSERT	Insertar
SELECT	Realizar consultas
UPDATE	Modificar datos

Tabla 8.3. Permisos en MySQL.



Caso práctico



- 4 Vas a abrir la base de datos *pruebas* al exterior, dejando la posibilidad de consultar información a cualquier usuario.

Se pide

En la base de datos *pruebas*, crea un usuario con nombre *invitado* y contraseña *hola* que pueda acceder a cualquier tabla de la base de datos, pero sólo para realizar consultas.

Solución

Habilitaremos al usuario invitado para realizar consultas (permiso *SELECT*) mediante la instrucción *GRANT*, que al mismo tiempo lo dará de alta. La ejecución del comando puede apreciarse en la Figura 8.38.

```
MySQL Command Line Client
mysql>
mysql>
mysql> grant select on pruebas.*
-> to invitado identified by 'hola';
Query OK, 0 rows affected (0.54 sec)
```

Fig. 8.38. Creación del usuario invitado.

A continuación, saldremos de MySQL y volveremos a entrar como invitado, nos conectaremos a la base de datos *pruebas* y realizaremos una consulta sobre la tabla *asignatura*. El resultado de este proceso puede observarse en la Figura 8.39.

```
C:\WINDOWS\system32\cmd.exe - mysql -h localhost -u invitado
C:\mysql> -h localhost -u invitado -p
Enter password:
Warning: Using a password on the command line interface can be insecure.
mysql> select * from asignatura;
+-----+-----+
| id    | nombre |
+-----+-----+
| 1    | física |
| 2    | química |
| 3    | matemáticas |
| 4    | historia |
| 5    | filosofía |
+-----+-----+
5 rows in set (0.00 sec)
```

Fig. 8.39. El usuario invitado realiza una consulta.

Seguidamente, trataremos de añadir una línea a la tabla *asignatura*, la misma que acabamos de consultar. El resultado de este intento puede apreciarse en la Figura 8.40.

```
C:\WINDOWS\system32\cmd.exe - mysql -h localhost -u invitado
mysql> insert into asignatura values(6,'MPI');
ERROR 1044 (42000): Access denied for user 'invitado'
mysql>
```

Fig. 8.40. El usuario invitado no puede insertar filas.

Aunque el mensaje de error no aclara del todo la situación, está claro que el sistema no permite que el usuario *invitado* inserte filas nuevas en la tabla. Esto obedece a que el único permiso que se le ha otorgado a dicho usuario es *SELECT*, y este permiso sólo le habilita para realizar consultas similares a la que ha efectuado en la Figura 8.38.



Consultar permisos

Para obtener una lista de los usuarios, debemos conectarnos a la base de datos *mysql* y ejecutar la siguiente consulta: `mysql> select user from user;`

Así, obtendremos una lista con todos los usuarios definidos. Para saber con qué bases de datos puede trabajar un usuario y con qué permisos, usaremos `SHOW GRANTS FOR usuario`.

Supongamos que deseamos localizar al usuario *invitado* y consultar sus permisos en las diferentes bases de datos creadas. En la Figura 8.41 puede apreciarse el resultado del proceso.

```
C:\WINDOWS\system32\cmd.exe - mysql -h localhost -u root
mysql> select host, user from user;
+-----+-----+
| host | user |
+-----+-----+
| %    | invitado |
| localhost | root |
+-----+-----+
2 rows in set (0.00 sec)

mysql> show grants for invitado;
+-----+
| Grants for invitado@% |
+-----+
| GRANT USAGE ON *.* TO 'invitado'@'%' IDENTIFIED BY 'hola' |
| GRANT SELECT ON `pruebas`.* TO 'invitado'@'%' |
+-----+
```

Fig. 8.41. Consulta de los permisos de un usuario.



8. Bases de datos y software libre. MySQL básico

8.3 Seguridad en MySQL

Como se observa en la figura, el usuario *invitado* tiene el permiso `USAGE` para todos los archivos de todas las bases de datos. En otras palabras, este usuario sólo puede conectarse. Sin embargo, dispone además del permiso `SELECT` para la base de datos *pruebas*, que le habilita a la hora de realizar consultas.



Borrado de usuarios

En MySQL es posible borrar usuarios mediante la instrucción `DROP USER` que, al contrario de `CREATE USER`, también existe en versiones anteriores a la 5. No obstante, en estas versiones sólo pueden eliminarse usuarios sin permisos, por lo que antes de borrar una cuenta de usuario deberán eliminarse todos sus permisos mediante la instrucción `REVOKE`.

Imaginemos que deseamos eliminar al usuario *invitado* que hemos creado para la práctica anterior. Los pasos a seguir son los que se reproducen en la Figura 8.42.

```
C:\WINDOWS\system32\cmd.exe - mysql -h
mysql> revoke all on pruebas.*
-> from invitado;
Query OK, 0 rows affected (0.40 sec)

mysql> drop user invitado;
Query OK, 0 rows affected (0.00 sec)

mysql> select user from user;
+-----+
| user |
+-----+
| root |
+-----+
1 row in set (0.00 sec)
```

Fig. 8.42. Eliminación de una cuenta de usuario.

B Copias, exportación e importación

El control de la seguridad de la información supone una parte muy importante en la gestión de bases de datos. Dejando aparte la gestión de usuarios, la principal medida a la hora de asegurar los datos es la realización de copias de seguridad.



Copias de seguridad

En primer lugar, es preciso señalar que no sirve de nada copiar los ficheros de datos de la carpeta correspon-

diente a la base de datos (en nuestro caso, *pruebas*) desde *Data* a otro directorio.

El motivo reside en que en la propia carpeta raíz *Data* se encuentran los ficheros correspondientes al diccionario de datos, que incluyen información necesaria para gestionar el contenido de la base de datos.

En todo caso, si queremos duplicar la carpeta, deberíamos hacerlo con todo el contenido de *Data*, con lo que realmente estaríamos copiando todas las bases de datos. Para copiar sólo una base de datos, o incluso solamente ciertas tablas, la mejor opción es crear un fichero de sentencias en lenguaje SQL que permitan, llegado el caso, reconstruir los ficheros y recuperar su contenido.

Una posibilidad muy sencilla para realizar copias de seguridad consiste en la utilización del comando `MYSQLDUMP`.

Las numerosas opciones que ofrece `MYSQLDUMP` pueden consultarse desde la línea de comandos, sin entrar en MySQL, tecleando `mysqldump-help`.

Para copiar nuestra base de datos *pruebas*, tanto la estructura de las tablas como su contenido, saldremos a la línea de comandos tecleando `cmd.exe` en el apartado *Ejecutar* del menú Windows y escribiremos:

```
mysqldump --u=root --password=obdoc --opt --
complete_insert pruebas > pruebas.sql
```

Las opciones `--u=root` y `--password=obdoc` no necesitan comentarios. La opción `--opt` reúne en sí misma varias características de configuración necesarias para realizar la copia. `--complete_insert` indica que se va a incluir en el fichero una copia de los datos y, a continuación, figura el nombre de la base de datos, en nuestro caso *pruebas*.

Finalmente, la salida de la instrucción se redirige a un fichero con extensión `.sql`, que contendrá todas las instrucciones necesarias para recuperar los ficheros de la base de datos si fuera preciso.

Es posible realizar también una copia de seguridad de una o varias tablas. Para ello, se incluirán los nombres de las tablas correspondientes a continuación del nombre de la base de datos. Si, por ejemplo, deseáramos



copiar la tabla *asignatura*, resultaría un comando como el que se reproduce en la Figura 8.43.

```
C:\Programas\MySQL>mysql -u root -p --password=root
mysql> select * into outfile 'C:\copias\asig.txt'
--opt --fields-enclosed-by='"' --terminated-by=';'
-- from asignatura;
```

Fig. 8.43. Copia de seguridad de la tabla *asignatura*.

Parte del contenido del fichero *asig.sql* que se ha generado durante el proceso de copiado puede apreciarse en la Figura 8.44.

```
--
-- Table structure for table 'asignatura'
--
DROP TABLE IF EXISTS `asignatura`;
CREATE TABLE `asignatura` (
  `idasi` smallint(5) NOT NULL auto_increment,
  `nombre` varchar(100) default NULL,
  `primer` tinyint(1) default '0',
  PRIMARY KEY (`idasi`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping data for table 'asignatura'
--
INSERT INTO `asignatura` (`idasi`, `nombre`, `primer`) VALUES
(1, 'asignatura', '0'),
(2, 'asignatura', '1'),
(3, 'asignatura', '0'),
(4, 'asignatura', '1'),
(5, 'asignatura', '0');
```

Fig. 8.44. Contenido del fichero *asig.sql*.

Así pues, con el sistema de copias que hemos visto, para recuperar los datos simplemente será necesario ejecutar el fichero *.sql* que se ha creado durante la copia.

Exportación de datos

En el Apartado anterior hemos presentado una forma de exportar datos que consistía en crear un fichero de sentencias SQL que posteriormente se pudieran ejecutar en otra base de datos.

Sin embargo, una exportación a otro tipo de programa, por ejemplo a Excel, no puede hacerse de esta manera.

Para exportar datos a otros programas, una buena opción consiste en la generación, para cada tabla, de un fichero de texto cuyo contenido coincida con el de la tabla copiada, generalmente con los campos entre comillas u otro separador.

La sentencia para realizar dicha exportación es la siguiente:

```
select * into outfile '\copias\asig.txt' ->
fields optionally enclosed by '"'
-> terminated by ';'
-> from asignatura;
```

Como se puede observar, la cláusula `FIELDS OPTIONALLY ENCLOSED BY` se usa para definir el carácter separador de campos (columnas) y `terminated by` para hacer lo propio con el separador de registros (filas).

Importación de datos

La importación de datos desde la línea de comandos se realiza a través de una sentencia que ya se ha presentado con anterioridad: `LOAD DATA INFILE`.

A dicha sentencia deberemos, de todas formas, añadirle las opciones `FIELDS OPTIONALLY ENCLOSED` y `TERMINATED BY` como se hayan usado en el proceso de exportación.

Así pues, podríamos recuperar el fichero *asig.txt*, generado en el apartado anterior, según puede observarse en la Figura 8.45.

```
C:\WINDOWS\system32\cmd.exe - mysql -h localhost -u root
mysql> load data infile '\copias\asig.txt'
-> into table asignatura
-> fields optionally enclosed by '"'
-> terminated by ';';
Query OK, 5 rows affected (0.40 sec)
Records: 5 Deleted: 0 Skipped: 0 Warnings: 0
```

Fig. 8.45. Importación de datos desde un fichero de texto.



ODBC (*Open DataBase Connectivity*) es un *driver* desarrollado por Microsoft que permite el acceso a bases de datos desde cualquier aplicación que soporte dicho *driver*.

Otra forma, seguramente más adecuada, de conectar MySQL con otros productos para la exportación e importación de datos es mediante el uso de *ODBC*.

En el próximo Apartado, dedicado al uso de MySQL en modo gráfico, incluiremos una sección dedicada a la conexión con otros programas a través de la mencionada técnica.



8. Bases de datos y software libre. MySQL básico

8.4 MySQL gráfico: EasyPHP

8.4 MySQL gráfico: EasyPHP

Existen diversas posibilidades para trabajar con MySQL en un entorno gráfico. Nosotros hemos optado por presentar un paquete muy completo que incluye todas las herramientas necesarias para llevar a cabo una completa gestión y administración de un SGBD basado en MySQL, incluso a través de la Web.

A El paquete EasyPHP. Componentes

El paquete **EasyPHP** surge ante la exigencia de combinar diversos productos necesarios para desarrollar un servidor web que ofrezca acceso a bases de datos.

EasyPHP contiene los principales productos relacionados con el acceso a bases de datos a través de la Web y proporciona una conectividad total entre ellos, además de gran potencia y flexibilidad.

Una de las grandes ventajas de EasyPHP es que los productos que ofrece son todos de libre uso, lo cual convierte a este paquete en una potente *suite* gratuita para el desarrollo de webs con acceso a bases de datos.

En primer lugar, veamos cuáles son los productos que se incluyen en EasyPHP, así como la función de cada uno de ellos.

Componentes

La versión 1.8 del paquete EasyPHP incluye los siguientes componentes:

- Servidor Apache para Web, versión 1.3.33
- Lenguaje de programación PHP, versión 4.3.10
- Base de datos MySQL versión 4.1.9
- phpMyAdmin, centro de control de MySQL, versión 2.6.1

A continuación, explicaremos paso a paso el proceso de instalación, configuración y puesta en marcha del paquete EasyPHP.

Posteriormente nos centraremos en el funcionamiento de phpMyAdmin, producto que nos interesa, dado que desde allí se realizarán las tareas de mantenimiento y administración de las bases de datos en MySQL.

B Instalación y configuración de EasyPHP

El paquete EasyPHP puede instalarse de dos formas diferentes: bien producto a producto por separado, bien mediante una instalación global y automática. Por nuestra parte, escogeremos esta segunda opción.

Descarga del producto

Como se ha comentado anteriormente, EasyPHP es un producto *open-source* que podemos obtener gratuitamente a través de la Web. La página oficial, desde la que efectuaremos la descarga, es www.easyphp.org y está en francés. En la parte izquierda de la página principal aparece un menú como el que se reproduce en la Figura 8.46.

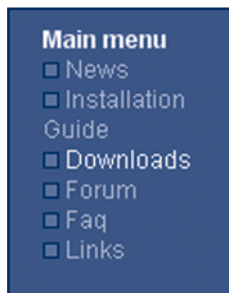


Fig. 8.46. Menú de la página web de EasyPHP.

Al seleccionar la opción *Downloads*, surgirá una lista con los productos disponibles para su descarga. Por nuestra parte, descargaremos e instalaremos EasyPHP 1.8. Al hacer clic sobre el producto elegido, se nos mostrará una lista de servidores desde los que podemos realizar la descarga. Escogemos uno cualquiera y descargaremos el fichero `easyphp1-8_setup.exe`.

Es conveniente también descargar el fichero `languages.txt`, pues contiene la información necesaria para disponer de una amplia lista de idiomas entre los que poder escoger.



Ambos ficheros, tanto `easyphp1-8_setup.exe` como `languages.txt`, los encontrarás en el CD del alumno.



⚙ Instalación del paquete EasyPHP

Una vez descargado el paquete EasyPHP 1.8 en nuestro disco duro, haremos doble clic sobre el fichero `easyphp1-8_setup.exe` y empezará, por tanto, la instalación propiamente dicha. En primer lugar, aparecerá una ventana que nos permite escoger el idioma en el que tendrá lugar el proceso de instalación. Ante la ausencia del castellano, optaremos por el inglés (véase la Figura 8.47).

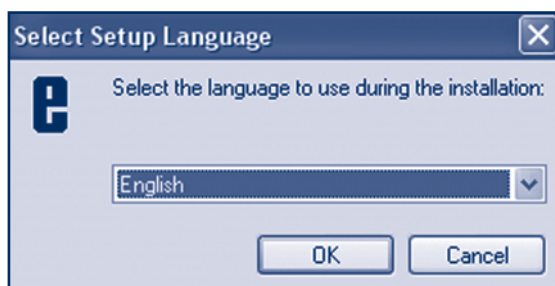


Fig. 8.47. Selección de idioma para la instalación.

Posteriormente, surgirá una pantalla de bienvenida, donde pulsaremos el botón *Siguiente* (Next). En la pantalla que aparecerá a continuación, se nos invita a aceptar la licencia de usuario propuesta. Es preciso tener en cuenta que, aunque el uso es gratuito, existen ciertas condiciones que conviene conocer.

Una vez aceptadas las condiciones, y tras otra pantalla de información en la que volveremos a pulsar *Siguiente* (Next), se nos pedirá un directorio para la instalación del producto. Aceptaremos el que se nos ofrece por defecto, tal como puede apreciarse en la Figura 8.48.

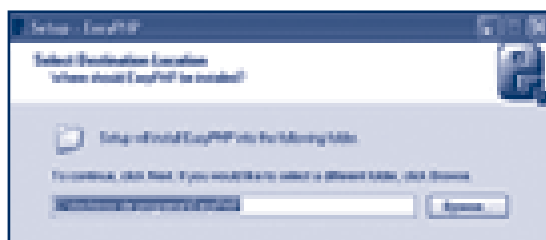


Fig. 8.48. Selección de la carpeta de instalación.

Seguidamente, se nos pedirá la ubicación del acceso directo al programa dentro del menú *Inicio*. Por nuestra parte, aceptaremos también la opción que aparece en la pantalla. Finalmente, dará comienzo el proceso de instalación del producto en nuestro ordenador, proceso que se completará de manera bastante rápida.

Si todo el proceso de instalación se ha realizado de forma correcta, aparecerán algunas pantallas donde tendremos que ir pulsando el botón *Continuar*. Finalmente, surgirá la pantalla de inicio de los servidores necesarios para el funcionamiento del paquete, pantalla que se muestra en la Figura 8.49.

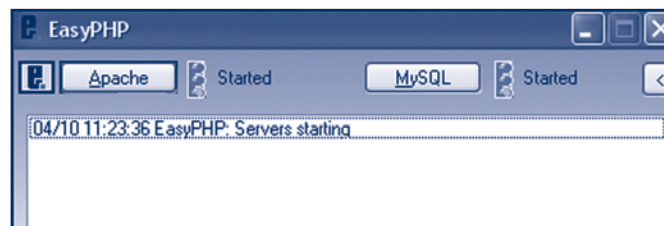


Fig. 8.49. Arranque correcto de los servidores.

Como puede apreciarse en la figura anterior, los servidores *Apache* y *MySQL* han arrancado correctamente, con lo cual ya tenemos el producto instalado y en funcionamiento.

Si el servidor *MySQL* no arranca, puede deberse a varios motivos. Una de las principales causas es tener otro servidor de *MySQL* ejecutándose al mismo tiempo, por ejemplo el que hemos instalado al principio de la unidad. También se pueden producir conflictos con el servicio de administración *IIS*. Por todo ello, en caso de fallo al iniciar los servidores, es conveniente consultar la lista de servicios de Windows y desactivar los que pueden causar conflictos. Este procedimiento es el que hemos seguido con *IIS Server*, tal como se muestra en la Figura 8.50.

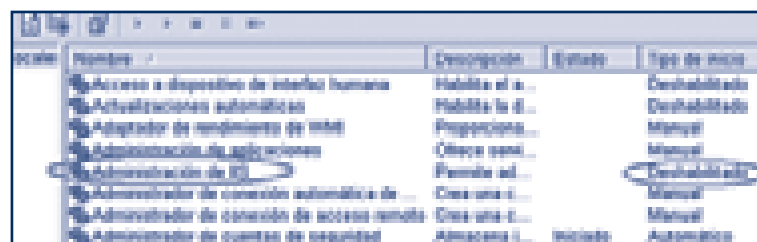


Fig. 8.50. Servicios de Windows incompatibles con EasyPHP.

Como bien hemos advertido, el producto se pone en funcionamiento de manera automática. Si queremos hacerlo manualmente, podemos dirigirnos al menú *Inicio/Programas/EasyPHP* (o el nombre que se le haya dado al apartado) y hacer clic en la opción *EasyPHP*.



8. Bases de datos y software libre. MySQL básico

8.4 MySQL gráfico: EasyPHP

Cuando el programa está en marcha, aparecerá en la barra de herramientas, en la parte inferior derecha de la pantalla, un icono similar al que reproducimos en la Figura 8.51.



Fig. 8.51. Icono correspondiente a EasyPHP en la barra de herramientas.

Para comprobar el estado de los servidores, basta con pulsar sobre dicho icono con el botón izquierdo del ratón. Si hacemos clic con el botón derecho, entonces aparecerá un menú donde se nos ofrece la posibilidad de detener el funcionamiento de los servidores o reiniciarlos, entre otras posibilidades. Más adelante examinaremos el resto de opciones.



Comprobación

A continuación, vamos a verificar el funcionamiento de los diferentes productos que forman EasyPHP. Además, aprovecharemos este Apartado para comentar distintos aspectos de la configuración que merecen destacarse.

En primer lugar, comprobemos el servidor web *Apache*. Si en el menú que aparece al hacer clic con el botón derecho del ratón sobre el icono de EasyPHP seleccionamos la opción *Web local (Local Web)*, surgirá una pantalla como la que se muestra en la Figura 8.52.

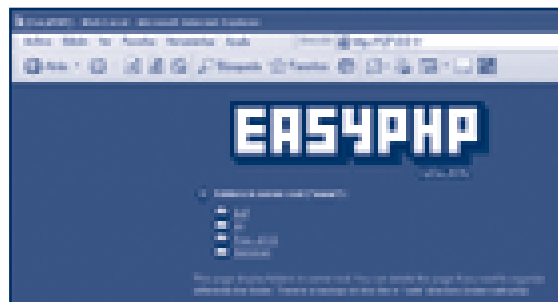


Fig. 8.52. Pantalla principal del servidor web.

La anterior pantalla es la que figura como página principal de nuestro servidor web. La dirección 127.0.0.1 que aparece en la barra de direcciones es la dirección de nuestra máquina local. Si más adelante creamos una página web y la dejamos en la carpeta que corresponde a nuestro servidor, ésta será la página que aparecerá en esta dirección.

Como nuestro objetivo no consiste en explicar el funcionamiento y configuración del servidor web, pasemos a comprobar el producto que realmente nos interesa: el servidor de MySQL.



El entorno gráfico: phpMyAdmin

El acceso al servidor de MySQL tiene lugar a través de un entorno gráfico denominado *phpMyAdmin*, entorno que abordaremos en los siguientes Apartados.



Acceso a phpMyAdmin

El entorno **phpMyAdmin** funciona como una Web, puesto que está preparado para acceder a él remotamente. Si deseamos entrar en modo local, abriremos nuestro navegador web y escribiremos la dirección <http://localhost/mysql/>, que nos permitirá acceder a una página como la que se aprecia en la Figura 8.53.

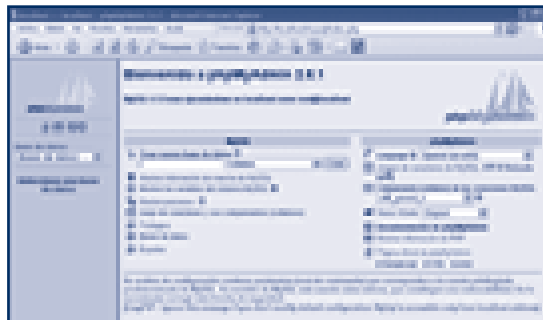


Fig. 8.53. Pantalla principal de phpMyAdmin.

En la parte derecha de la anterior pantalla se puede observar un desplegable que da opción a escoger el idioma de trabajo. Ahora sí que podemos elegir el castellano, entre otros muchos idiomas disponibles. No obstante, para tener esta lista actualizada, debemos copiar el fichero `langues.txt`, que habíamos descargado antes, a la misma carpeta (`c:\Archivos de Programa\EasyPHP`).



Gestión de bases de datos con phpMyAdmin

Vamos a empezar a trabajar con *phpMyAdmin*. En primer lugar, observamos que en la parte izquierda de la pantalla reproducida en la Figura 8.53 hay un selector bajo el nombre *Bases de Datos* que nos permite escoger la base de datos con la que deseamos trabajar (véase la Figura 8.54).



Fig. 8.54. Selector de bases de datos.

Si es la primera vez que trabajamos con este producto, lógicamente no dispondremos de ninguna base de datos creada. En la parte central de la pantalla reproducida en la Figura 8.53 disponemos de la opción *Crear nueva base de datos* que nos permitirá precisamente crear una nueva base de datos (véase la Figura 8.55).



Fig. 8.55. Apartado para crear una base de datos.

Debemos tener en cuenta que las bases de datos se crean en la carpeta *Data* dentro de `c:\Archivos de programa\EasyPHP\mysql`.

Sin embargo, la ubicación puede cambiarse modificando el fichero `my.ini` que se encuentra también en `c:\Archivos de programa\EasyPHP\mysql`. Además, podemos dirigirnos al menú contextual que aparece al hacer clic con el botón derecho sobre el icono de EasyPHP, en el apartado *configuration/mysql*, para obtener el mismo resultado. De cualquiera de las dos formas, surgirá el contenido del fichero `my.ini`, tal como se aprecia en la Figura 8.56.

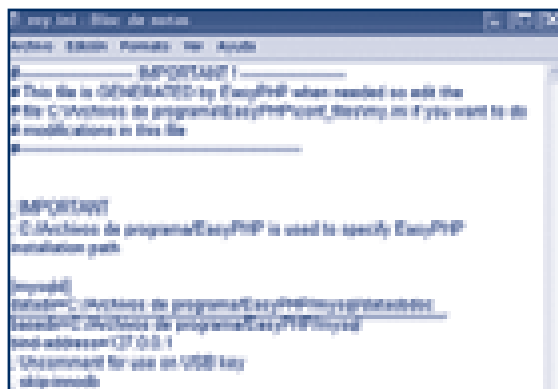


Fig. 8.56. Configuración del servidor MySQL.

Como se observa en la ventana anterior, se ha cambiado la línea marcada en rojo de manera que la carpeta que figura al final sea *dataobdoc*, en lugar de *Data*, como aparecía originalmente. De esta forma dispondremos de una carpeta nueva y vacía en donde poder experimentar libremente todas las opciones que se irán presentando en los siguientes Apartados. Así pues, vamos a crear nuestra base de datos *pruebas*, tal como se observa en la Figura 8.57.



Fig. 8.57. Creación de la base de datos pruebas.

A continuación podemos abrir la lista de bases de datos y comprobaremos, según puede apreciarse en la Figura 8.58, que *pruebas* aparece junto a las bases de datos que se crean automáticamente durante la instalación.



Fig. 8.58. La base de datos pruebas ha sido creada.

Para abrir una base de datos de la lista, simplemente haremos clic sobre su nombre y surgirá una pantalla similar a la que se reproduce en la Figura 8.59. En la parte inferior aparecerán, en su momento, las tablas incluidas en la base de datos.

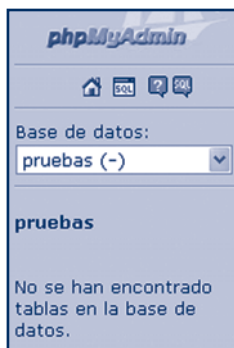


Fig. 8.59. La base de datos pruebas está vacía.



8. Bases de datos y software libre. MySQL básico

8.4 MySQL gráfico: EasyPHP

La base de datos *pruebas*, como puede observarse, está vacía. Ahora es el momento de empezar a crear las tablas que contendrán la información.



Creación y mantenimiento de tablas

Al crear una base de datos aparece en la pantalla una opción para crear tablas nuevas de forma manual, como puede observarse en la Figura 8.60.

No se han encontrado tablas en la base de datos.

Crear nueva tabla en la base de datos pruebas:

Nombre:

Campos:

Fig. 8.60. Opción de creación de tablas.

A continuación procederemos a crear, por tanto, la tabla *alumnos*, con la estructura que se ha ido aplicando en apartados anteriores:

```
idalu          int(4) unsigned
nomalu         varchar(30)
fnacimiento    date
```

En primer lugar, escribiremos el nombre de la tabla en el cuadro de texto que hemos mostrado en la Figura 8.60 y especificaremos que contendrá tres campos.

Tras pulsar el botón *Continúe*, aparecerá la pantalla de definición de campos, que consta de los siguientes apartados:

- **Campo.** En esta casilla introduciremos el nombre del campo, en nuestro caso *idalu*.
- **Tipo.** Elegiremos el tipo del campo entre los que aparecen al desplegar la lista. En el caso de *idalu* escogeremos el tipo *int*, por ejemplo.
- **Longitud/valores.** En esta casilla especificaremos la longitud del campo o, en el caso de tipos enumerados, los posibles valores que puede tomar. En nuestro caso, dejaremos en 4 la longitud de *idalu*.
- **Ordenación (Collation).** Hace referencia al criterio de ordenación alfabética y, a no ser que tengamos problemas de este tipo, es mejor dejarlo en blanco o con el valor por defecto.
- **Atributos.** En este apartado podemos escoger entre algunos modificadores del tipo como, por ejemplo,

unsigned, que es el que seleccionaremos para el campo *idalu*.

- **Nulo.** En esta casilla podemos escoger entre *null* o *not null* para especificar, respectivamente, si el campo admite valores nulos o no. En nuestro caso, lo dejaremos en *not null*.
- **Predeterminado.** Esta opción permite introducir un valor por defecto para que, al dar de alta una nueva fila, el campo se inicialice con dicho valor. En el caso de *idalu* no es necesario, puesto que escogeremos otra opción posterior para su inicialización.
- **Extra.** Precisamente en esta casilla es donde especificaremos el valor por defecto del campo *idalu*, puesto que se nos permite definir el campo como *auto_increment*, es decir, que va tomando valores sucesivos por defecto, empezando desde el 1.

Al final de cada campo se encuentran unos apartados encabezados por un icono identificador. Si pasamos con el ratón sobre dicho icono veremos cuál es su significado. En la Figura 8.61 puede apreciarse un detalle de esta parte de la definición de campo.



Fig. 8.61. Definición de claves y otras características de campo.

El primer icono corresponde a la característica de clave primaria, y se deberá seleccionar en todos aquellos campos que formen parte de dicha clave. En nuestro caso, sólo marcaremos como clave primaria el campo *idalu*, el primero de la lista.

El segundo icono hace referencia a la indexación del campo: habrá un índice asociado si lo dejamos marcado y no lo habrá si lo dejamos desmarcado. En el caso de *idalu* no lo marcaremos, puesto que el hecho de ser clave primaria ya implica la existencia de un índice asociado al campo.

El tercer icono permite definir la unicidad o no del campo. En los campos que sean clave primaria por sí



mismos no será necesario seleccionar esta característica. En los campos que no sean clave primaria por sí mismos, marcaremos la casilla si no deseamos admitir valores duplicados para el campo.

La cuarta columna está marcada en los campos que no poseen ninguna característica especial por lo que respecta a claves, índices o valores únicos. Si la seleccionamos manualmente, estamos anulando todas las definiciones realizadas anteriormente.

La última columna permite la introducción de un texto explicativo de la tabla. Es importante hacer constar que el tipo de la tabla debe definirse como *InnoDB* si nuestra intención es crear claves ajenas, puesto que el resto de tipos de tabla, al menos en las versiones actuales, no admiten dichas claves. Si no se va a trabajar con integridad referencial y, por tanto, no son necesarias las claves ajenas, entonces se puede optar por el tipo *MyISAM*, que es más rápido.

La selección del tipo de tabla se realiza en la lista desplegable *Tipo de tabla* que aparece justo debajo de la definición de campos (véase la Figura 8.62).

Fig. 8.62. Definición del Tipo de tabla.

En las Figuras 8.63 y 8.64 se observa el resultado de la definición completa de los campos de nuestra tabla *alumnos*.

Fig. 8.63. Definición de campos de la tabla *alumnos*.

Fig. 8.64. Definición de campos de la tabla *alumnos*.

Una vez completada la definición de la tabla, pulsaremos el botón *Grabar* para proceder a su creación definitiva.

Una vez guardada la tabla, aparecerá su nombre en la parte izquierda de la pantalla, en la lista correspondiente a la base de datos de la que forma parte, que en nuestro caso se trata de *pruebas*. En la Figura 8.65 puede comprobarse cómo la nueva tabla se encuentra en la lista.

Fig. 8.65. La tabla *alumnos* aparece en la base de datos.

Al seleccionar la tabla *alumnos* en la lista, surgirá una pantalla con varias opciones que permiten realizar diversas operaciones sobre la tabla escogida. Estas opciones son:

- **Estructura.** Esta pantalla se corresponde, como es obvio, con la estructura de la tabla. Su formato es muy similar al visto durante su creación. Desde esta ventana se pueden realizar cambios como la inclusión de nuevas columnas o la modificación de atributos. Aparte de los iconos que ya conocemos, aparecen dos nuevos: *Editar*, que seleccionaremos cuando queramos realizar modificaciones en la fila; *Suprimir*, que permite eliminar el campo o campos seleccionados. Estos iconos aparecen en otras pantallas, siempre con el mismo significado de modificar y borrar, respectivamente.



8. Bases de datos y software libre. MySQL básico

8.4 MySQL gráfico: EasyPHP

En esta misma pantalla también pueden crearse índices y relaciones, algo que veremos con más detalle cuando expliquemos la forma de crear claves ajenas. En la Figura 8.66 se reproduce la pantalla Estructura.

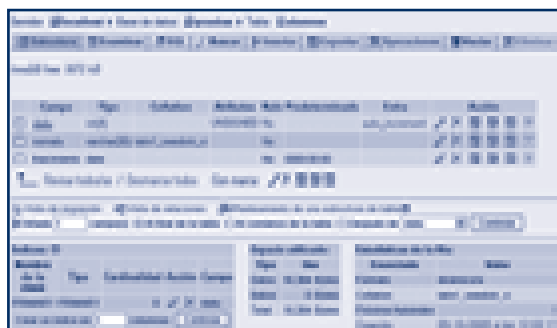


Fig. 8.66. Estructura de una tabla.

- **Insertar.** Esta pestaña se encuentra más a la derecha que *Examinar*, pero la vemos en primer lugar porque es preciso insertar datos antes de poder explorar el contenido de la tabla.

Al seleccionar *Insertar* aparece una pantalla, similar a la que se reproduce en la Figura 8.67, donde introduciremos los valores de uno o varios campos (por defecto dos). A continuación, escogeremos entre la opción de volver o continuar insertando registros. En cualquier caso, pulsaremos el botón *Continúe*.



Fig. 8.67. Introducción de valores en una tabla.

- **Examinar.** Esta pestaña se activa una vez que la tabla ya contiene datos. Al seleccionarla, aparece una pantalla con los datos contenidos en la tabla. El formato con el que aparecen los datos puede variarse al modificar las opciones que se muestran en la parte inferior de la pantalla, tal como se aprecia en la Figura 8.68.

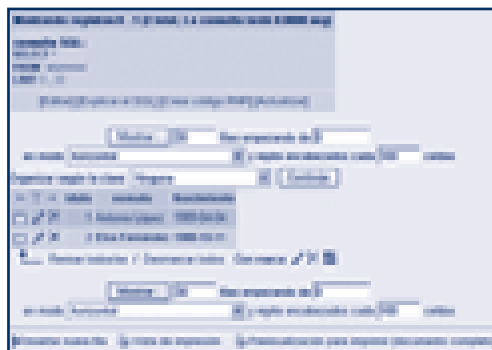


Fig. 8.68. Examinar el contenido de la tabla.

- **SQL.** Esta opción permite realizar cualquier consulta en formato SQL, como puede observarse en la Figura 8.69.

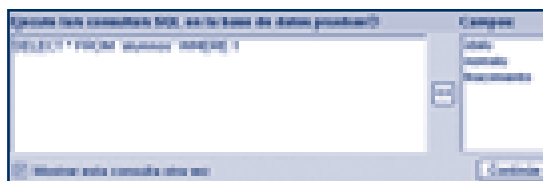


Fig. 8.69. Introducción manual de una consulta SQL.

También se puede obtener dicha consulta de un archivo de texto externo, como se aprecia en la Figura 8.70.



Fig. 8.70. Ejecución de una consulta externa.

- **Buscar.** Desde esta opción se pueden realizar consultas utilizando una especie de asistente (véase la Figura 8.71). Para seleccionar campos alternos, deberemos hacer clic con el ratón sobre cada uno de ellos, mientras mantenemos pulsada la tecla *Mayúsculas*.
- **Exportar.** Desde este apartado se examinará en detalle cuando hagamos referencia a la exportación e importación de datos mediante *phpMyAdmin*.
- **Operaciones.** Al seleccionar esta pestaña aparecen varias operaciones que pueden realizarse sobre la tabla escogida, como por ejemplo:
 - Modificar el orden de la tabla.
 - Cambiar el nombre de la tabla.

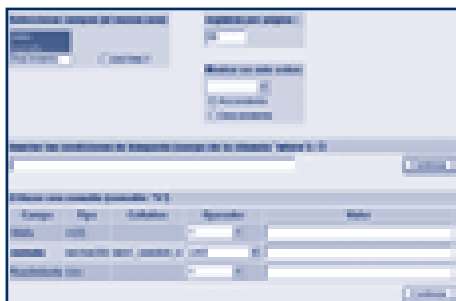


Fig. 8.71. Confección de consultas con el asistente.

- Mover o copiar la tabla a otra tabla, incluso de diferente base de datos.

Cambiar el tipo de la tabla, opción muy útil si se nos ha olvidado definirla como *InnoDB*.

Realizar operaciones de mantenimiento como desfragmentación, vaciado de caché, etc.

La pantalla correspondiente a esta opción puede apreciarse en la Figura 8.72.

- **Vaciar.** Esta opción permite eliminar todos los registros introducidos. Sin embargo, la tabla sigue existiendo, aunque vacía.

- **Eliminar.** Mediante esta opción se borra por completo una tabla, tanto la información contenida como la propia tabla. Una vez que una tabla ha sido eliminada, dejará de aparecer en la lista de tablas que forman la base de datos.

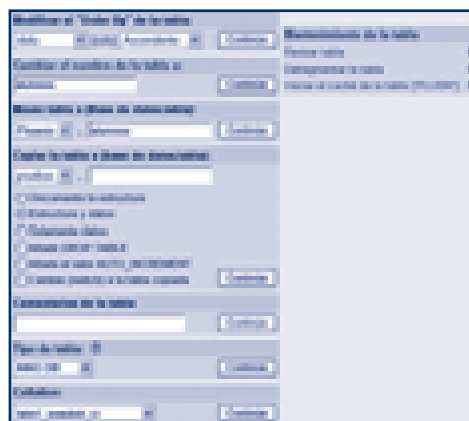


Fig. 8.72. Opciones del apartado Operaciones.

Caso práctico



- 5 En el Apartado anterior has creado la base de datos *pruebas* y la tabla *alumnos*. En este Caso práctico vas a crear alguna tabla más en la misma base de datos.

Se pide

Creas la tabla *asignaturas* con la siguiente estructura:

```
idasi smallint(2) unsigned,
nomasi varchar(20)
```

Solución

Para proceder a la creación de esta nueva tabla, simplemente hemos de seguir los mismos pasos que se han dado para crear la tabla *alumnos*. Por tanto, en primer lugar, abriremos la base de datos *pruebas* haciendo clic en la parte superior de la pantalla, sobre su nombre (véase la Figura 8.73).



Fig. 8.73. Volver a la pantalla principal de pruebas.

Una vez en la pantalla principal de la base de datos, volveremos a ver la opción de crear una nueva tabla. En este momento procederemos a dar los mismos pasos efectuados para crear la tabla *alumnos*. En las Figuras 8.74 y 8.75 puede apreciarse el resultado final, tal como debería haber quedado.



Fig. 8.74. Creación de la tabla asignaturas (1).

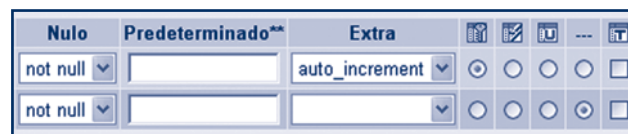


Fig. 8.75. Creación de la tabla asignaturas (2).

Es preciso tener presente que la tabla debe ser de tipo *InnoDB*, puesto que posteriormente nos veremos en la necesidad de definir claves ajenas y opciones de integridad referencial. Si por un descuido la tabla se ha creado como *MyIsam*, podemos enmendar nuestro error yendo a la opción *Operaciones* y cambiando el tipo a *InnoDB*.



8. Bases de datos y software libre. MySQL básico

8.4 MySQL gráfico: EasyPHP

Claves ajenas e integridad referencial

Como hemos mencionado en diferentes ocasiones, la integridad referencial sólo está implementada de momento sobre tablas del tipo *InnoDB*. Por tanto, para trabajar con claves ajenas es preciso comprobar que las tablas son del tipo mencionado.

La creación de claves ajenas puede realizarse de dos formas, bien mediante sentencias SQL directamente, bien a través del asistente. En cualquier caso, hay que tener en cuenta las dos condiciones ya mencionadas en su momento pero que conviene recordar:

- Los campos relacionados han de ser exactamente del mismo tipo.
- Los campos relacionados han de estar indexados por sí mismos, es decir, no basta con que formen parte de un índice junto a otros campos.

Por su parte, para indexar un campo disponemos de dos opciones:

- Hacerlo durante la creación del mismo, mediante el icono *Índices*, tal como se ha apuntado anteriormente.
- Crear los índices posteriormente abriendo la tabla y entrando en *Estructura*, como puede apreciarse en la Figura 8.76.


Índices: ?				
Nombre de la clave	Tipo	Cardinalidad	Acción	Campo
PRIMARY	PRIMARY	0	 	idasi
Crear un índice en 1 columnas <input type="button" value="Continúe"/>				

Fig. 8.76. Pantalla de creación de índices.

Los campos que ya tienen un índice de tipo *PRIMARY*, como es el caso de *idasi* en la figura anterior, no necesitan indexarse de nuevo. Sin embargo, los campos que formen parte de un índice compuesto sí que deberán indexarse por separado si intervienen en alguna relación con otras tablas.

Para crear el índice, simplemente hay que pulsar el botón *Continúe* y seleccionar el campo que se desea indexar. Para borrar o modificar un índice disponemos, en la columna *Acción*, de los botones que ya conocemos.

Una vez indexados en ambas tablas los campos que intervienen en la relación, es el momento de enlazarlos y especificar las acciones encaminadas a asegurar la integridad referencial.

Para ello, seleccionaremos la tabla que contiene las claves ajenas y entraremos en la opción *Vista de relaciones* que aparece en la pestaña *Estructura*, tal como se observa en la Figura 8.77.

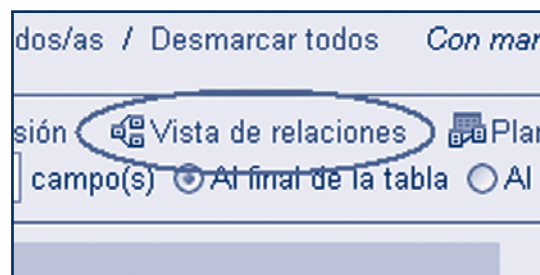


Fig. 8.77. Apartado para crear o modificar relaciones.

Dentro de *Vista de relaciones* aparecen todos los campos de la tabla, aunque solamente en aquellos que pueden ser claves ajenas se nos dará la posibilidad de elegir con qué clave primaria de otra tabla se relacionará el campo. Además, para cada relación podemos establecer qué operación se llevará a cabo cuando se intente modificar o borrar el valor de la clave primaria asociada. Las opciones de las que disponemos son las siguientes:

- **CASCADE.** Al cambiar el valor de la clave primaria se actualizarán automáticamente los valores de las claves ajenas. Si se borra una clave primaria, entonces se eliminarán todos los registros con claves ajenas que hagan referencia a la clave primaria borrada.
- **SET NULL.** Si se cambia o se borra una clave primaria, entonces las claves ajenas relacionadas cambiarán su valor a *NULL*.
- **NO ACTION.** Las claves ajenas no sufrirán ninguna variación ante un cambio o borrado de las claves primarias relacionadas. Lógicamente, esto supone una violación de la integridad referencial y, por tanto, no es aconsejable.
- **RESTRICT.** No se permite borrar ni cambiar el valor de claves primarias mientras tengan claves ajenas relacionadas. En caso necesario, deberán eliminarse los registros relacionados de otras tablas previamente al cambio o borrado de la clave primaria.

Veamos un ejemplo mediante un Caso práctico.



Caso práctico

- 6 En la base de datos *pruebas* dispones de las tablas *alumnos* y *asignaturas*. Se desea relacionar ambas tablas en una relación M:N.

Se pide

Crea la tabla *matrícula* con la siguiente estructura:

```
idasi smallint(2) unsigned
idalu int(4) unsigned
```

A continuación crea las relaciones entre la nueva tabla y el resto, de forma que se asegure la integridad referencial, que no se pueda borrar un alumno ni una asignatura con matrículas relacionadas y que al cambiar la clave primaria de un alumno o una asignatura se cambien automáticamente las claves ajenas correspondientes en la tabla *matrícula*.

Solución

En primer lugar, crearemos la tabla de la manera que ya hemos puesto en práctica anteriormente. El resultado sería similar al que se reproduce en la Figura 8.78.

Nombre	Tipo	Caracteres	Restricciones	Auto Incrementado	Tabla	Acciones
idasi	smallint(2)		unsigned			
idalu	int(4)		unsigned			

Fig. 8.78. Estructura de la tabla *matrícula*.

Como se puede observar, *idalu* e *idasi* se han creado con las mismas propiedades que tienen en las tablas *alumnos* y *asignaturas*, respectivamente. Sin embargo, hay una excepción: en la tabla *matrícula* no tienen la propiedad *auto_increment*, ya que su valor no se genera automáticamente, sino que será introducido por el usuario.

Queda un último detalle antes de definir las relaciones. Como hemos comentado con anterioridad, los campos que actúan como claves ajenas deben tener un índice propio. Esto no ocurre en ese instante, puesto que los campos *idalu* e *idasi* forman parte de un mismo índice en la tabla *matrícula*. Por tanto, el siguiente paso consistirá en crear un índice para cada uno de los campos por separado. Para ello, nos dirigiremos al apartado *Índices* que se encuentra en la parte central de la pantalla y crearemos los índices tal como se puede observar en la Figura 8.79.

Nombre de la clave	Tipo	Cardinalidad	Acción	Campo
PRIMARY	PRIMARY	0		idalu idasi
idasi_1	INDEX	0		idasi
idalu_1	INDEX	0		idalu

Fig. 8.79. Creación de los índices de la tabla *matrícula*.

Una vez creada la nueva tabla con sus campos e índices correspondientes, para definir su relación con el resto iremos a la opción *Vista de relaciones*. Al hacer clic sobre la misma, aparecerá una pantalla como la que se reproduce en la Figura 8.80.

Fig. 8.80. Pantalla de creación de relaciones.

En este momento introduciremos las relaciones de la tabla *matrícula* con las tablas *alumnos* y *asignaturas*. Es evidente que el campo *idalu* enlaza la tabla *matrícula* con la tabla *alumnos*. Por su parte, *idasi* enlaza la tabla *matrícula* con la tabla *asignaturas* (véase la Figura 8.81).

Fig. 8.81. Creación de relaciones en la tabla *matrícula*.

Como se observa en la Figura 8.81, para la propiedad *ON DELETE* se ha escogido el valor *RESTRICT*, de manera que no se admiten borrados de alumnos que tengan fichas asociadas en la tabla *matrícula*. Lo mismo se ha hecho con *idasi*.

Para la propiedad *ON UPDATE* se ha elegido el valor *CASCADE*, de forma que si cambia el identificador de un alumno o una asignatura en sus respectivas tablas, lo hará también en *matrícula*.



8. Bases de datos y software libre. MySQL básico

8.4 MySQL gráfico: EasyPHP

D Seguridad con phpMyAdmin

La seguridad con phpMyAdmin se implementa tal como hemos visto en el Apartado correspondiente de MySQL. Así pues, nos limitaremos a comentar ciertos aspectos de la gestión de usuarios.

Gestión de usuarios con phpMyAdmin

Desde la pantalla principal de phpMyAdmin, antes de abrir ninguna base de datos podemos crear usuarios mediante la opción *Privilegios*, cuya pantalla se reproduce en la Figura 8.82.

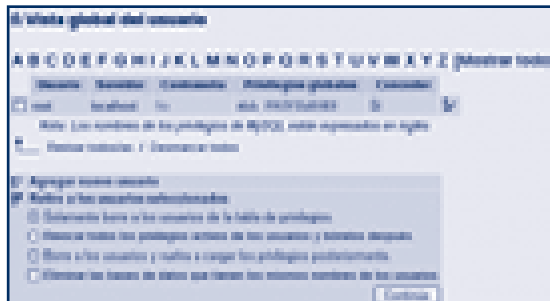


Fig. 8.82. Vista global de usuarios.



Un mensaje en la parte inferior de la pantalla nos indica la necesidad de recargar los privilegios cada vez que hagamos algún cambio desde esta opción.

Como se puede apreciar en la Figura 8.82, de momento sólo existe el usuario *root*, creado durante la instalación. Si seleccionamos dicho usuario y vamos a *Modificar* encontraremos los siguientes apartados:

- **Privilegios globales.** Se aplican a todas las bases de datos.
- **Privilegios específicos.** Se aplican sólo a las bases de datos y tablas seleccionadas.
- **Cambio de contraseña.** Permite cambiar la contraseña previamente introducida, o crearla si no se ha hecho anteriormente.
- **Cambio de login.** Permite cambiar el nombre de usuario, el servidor al que se conecta y la contraseña.

Estas opciones son muy sencillas y no necesitan mayor comentario. Sin embargo, es interesante resaltar un aspecto de la seguridad: el usuario *root* puede acceder

a la base de datos sin contraseña. Para asignar una contraseña al usuario *root* accedemos a *Privilegios*, seleccionamos el usuario, entramos al icono *Modificar*, y, desde *Cambio de contraseña*, efectuamos la operación. Inmediatamente dejaremos de tener acceso a phpMyAdmin y aparece un mensaje similar al de la Figura 8.83.

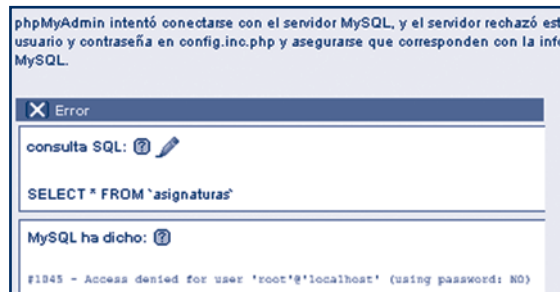


Fig. 8.83. Error de conexión con phpMyAdmin.

Para solucionar este problema, accederemos al fichero *config.inc.php*, que se encuentra en *c:\Archivos de programa\Easyphp\phpMyAdmin*, lo abrimos mediante un editor de php o con *WordPad* (no con el bloc de notas), y buscaremos la parte que aparece en la Figura 8.84.

```
$cfg['Servers'][$i]['auth_type'] = 'config';  
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = '';  
  
$cfg['Servers'][$i]['only_db'] = '';
```

Fig. 8.84. Detalle del fichero *config.inc.php*.

En la línea correspondiente a *password* pondremos la contraseña. En cuanto al tipo de conexión que aparece en la primera línea, por defecto *config*, contamos con las siguientes opciones:

- **config.** No pide identificación.
- **cookies.** Leerá el nombre de usuario y la contraseña de un fichero del disco duro.
- **http.** Se pedirá cada vez el nombre de usuario y la contraseña. Es la mejor opción si el *phpMyAdmin* es accesible desde red y es, por tanto, la conexión que nosotros escogemos. Así, el fichero quedará tal como se puede apreciar en la Figura 8.85.

```
$cfg['Servers'][$i]['auth_type'] = 'http' ;  
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = 'obdoc';
```

Fig. 8.85. Cambios en el fichero *config.inc.php*.



Al reiniciar EasyPHP, por ejemplo con el botón derecho sobre el icono de la barra de herramientas inferior derecha y pulsando *Reiniciar (Restart)*, al entrar a *phpMyAdmin* aparecerá la pantalla que se ve en la Figura 8.86.

Fig. 8.86. Identificación de entrada a *phpMyAdmin*.

E Exportación e importación de datos con *phpMyAdmin*

La exportación e importación de datos puede realizarse directamente mediante sentencias SQL, o a través de ODBC.

Importación de datos desde SQL

Para importar datos desde ficheros SQL entraremos en la pestaña *SQL* y, desde la opción *Localización del archivo de texto*, buscaremos y ejecutaremos el fichero correspondiente (véase la Figura 8.87).

Fig. 8.87. Ejecución de instrucciones SQL desde un fichero externo.

Exportación de datos

Desde la pestaña *Exportar* se pueden exportar tanto tablas como bases de datos enteras. Al hacer clic en dicha pestaña, aparecen diversas opciones como la elección del tipo de fichero generado, la exportación sólo de la estructura o de la estructura y datos o la generación de un fichero SQL. Estas opciones se reproducen en la Figura 8.88.

Conexión a través de ODBC

En la página web de MySQL, en el apartado de descargas, se encuentra la última versión del driver ODBC para Windows.

Fig. 8.88. Opciones de exportación.



La última versión de ODBC es un fichero que ocupa unos 8 Mb. Podrás encontrarlo en el CD que se adjunta con el libro (*MyODBC-3.51.11-1-win.exe*).

Una vez descargado, se ejecuta y simplemente hay que ir respondiendo a las preguntas que van apareciendo en pantalla para completar su instalación. A continuación, al entrar en *Panel de Control/Herramientas Administrativas/Orígenes de datos/ODBC*, deberíamos ver una pantalla similar a la que se muestra en la Figura 8.89.

Fig. 8.89. Driver ODBC para MySQL.

Para configurar el *driver* lo seleccionaremos en la lista e introduciremos el nombre de usuario (*root*) y la contraseña (*obdoc*), así como el servidor al cual nos conectaremos, que será, si trabajamos como local, *localhost*. También podemos seleccionar la base de datos a la que entraremos por defecto cuando nos conectemos vía ODBC. Pulsando *test* podemos comprobar si el funcionamiento es correcto. Si es así, desde cualquier programa que incluya ODBC podremos traspasar información a nuestra base de datos.

Para concluir, es interesante que el lector sepa que existen otras interfaces gráficas para MySQL aparte de *phpMyAdmin*. Algunas de ellas son *Gnome MySQL Client*, *MySQLNavigator*, *MySQLGUI* (desarrollada por los propios creadores de MySQL) o *WebMin*.



Actividades



- 1 ¿Qué ventajas tiene instalar MySQL como un servicio de Windows? Razona tu respuesta y enumera dichas ventajas.
- 2 Incluir las carpetas de MySQL dentro de la variable del sistema `PATH` tiene una ventaja, ¿cuál es?
- 3 ¿Qué diferencia hay entre entrar a MySQL desde la opción que se crea en el menú *Inicio*, o hacerlo desde la línea de comandos entrando con *Ejecutar/cmd.exe*?
- 4 ¿Cuál es la función de la variable `Datadir` dentro del fichero de configuración `my.ini`?
- 5 ¿Cuál es la función de la sentencia `DESCRIBE`?
- 6 ¿Cómo afecta a una variable de tipo numérico la cláusula `UNSIGNED`? ¿Y la cláusula `ZEROFILL`?
- 7 ¿En qué casos puede declararse un campo como clave primaria en la misma línea de definición del campo? Justifica tu respuesta.
- 8 ¿En qué consiste la restricción de valor único?
- 9 ¿Cuál es la función de la cláusula `AUTO_INCREMENT`?
- 10 ¿Cuáles son las características del tipo de datos `enum`?
- 11 ¿Para qué modificaciones de una tabla no puede usarse la cláusula `MODIFY` de la instrucción `ALTER TABLE`?
- 12 ¿Qué instrucción usaremos en MySQL para crear un usuario?
- 13 ¿Qué instrucción podemos usar en MySQL para conocer los privilegios de un usuario?
- 14 ¿Cuáles son las ventajas e inconvenientes de las tablas *InnoDB* respecto a las de tipo *MyIsam*? Razona tu respuesta y enumera dichas ventajas e inconvenientes.
- 15 ¿Cómo podemos convertir una tabla *InnoDB* al tipo *MyIsam*? Explica brevemente los pasos que deberías dar.
- 16 Al definir una relación, ¿qué significa el valor `CASCADE` en la propiedad `ON DELETE`? ¿Y el mismo valor en la propiedad `ON UPDATE`?
- 17 ¿Cómo podemos evitar que se borren registros que tienen asociados a su vez registros de otras tablas a través de claves ajenas?
- 18 Explica una de las formas de copiar una tabla con otro nombre en la misma base de datos. Si lo consideras oportuno, pon un ejemplo.
- 19 ¿Cuál es la función del *driver ODBC* para MySQL?
- 20 Aparte de *ODBC*, ¿qué otras opciones existen para importar y exportar datos desde MySQL?
- 21 ¿Qué opciones de conexión podemos configurar desde el fichero `config.inc.php`? Confecciona una lista con dichas opciones.
- 22 ¿Qué diferencia existe entre los privilegios globales y los privilegios específicos? Pon un par de ejemplos de cada uno de ellos.
- 23 ¿Por qué no es aconsejable el valor `NO ACTION` en la propiedad `ON DELETE` de una relación? Justifica tu respuesta.



En `bd_con_easyphp.htm` hay un vídeo donde se explica cómo se gestionan las BD mysql en entorno gráfico EasyPHP.



Prácticas



- 1 Crea, usando MySQL en modo texto, una base de datos llamada *ejercicios*.
- 2 En la base de datos anterior crea los siguientes ficheros:
`proveedor` (#idprv,nomprv,tfno)
`producto` (#idprd,nomprd,pvp)
- 3 Inserta diversos valores en las tablas *proveedor* y *producto*.
- 4 Crea un fichero de texto con extensión SQL que contenga las instrucciones necesarias para reconstruir la base de datos *ejercicios* en caso de fallo, o si se quiere exportar la información a otra base de datos.
- 5 Accede a través de tu navegador a phpMyAdmin y comprueba que no tienes ningún problema en hacerlo. En caso contrario, realiza los cambios de configuración que sean necesarios.
- 6 Crea una base de datos llamada *actividades* desde phpMyAdmin. Importa las tablas que tenías en la base de datos *ejercicios* usando el fichero SQL creado en la actividad 4.
- 7 Comprueba si las tablas que has importado son del tipo *InnoDB*. En caso contrario, modifícalas para que lo sean.
- 8 Crea desde phpMyAdmin una nueva tabla, que denominarás *tarifas*, que sirva para relacionar *proveedor* y *producto*. Además, dicha tabla deberá contener un campo llamado *pcoste* que hará referencia al precio de coste que cada proveedor ofrece para cada producto. Las actualizaciones deberán realizarse en cascada y los borrados de las claves primarias que tengan claves ajenas asociadas no deberán permitirse.
- 9 Inserta varias filas en la tabla *tarifas* y comprueba que se cumple la integridad referencial, así como todas las restricciones incluidas en la definición de la tabla.
- 10 Crea un usuario llamado *invitado* que sólo pueda entrar a consultar la información contenida en la base de datos, pero no pueda modificar ni borrar nada.
- 11 Realiza los cambios necesarios en la configuración de phpMyAdmin para permitir el acceso al usuario creado en la actividad 10. Entra con dicha clave de usuario y comprueba si las restricciones de acceso que has definido se cumplen.
- 12 Instala el *driver ODBC* de MySQL que puedes encontrar en su página web, realiza los cambios de configuración que sean necesarios y efectúa una importación de datos desde una base de datos en Access de la información contenida en la base de datos *actividades*.
- 13 Realiza una exportación desde phpMyAdmin de la información de la tabla *asignaturas* a un fichero de Excel.
- 14 Utiliza las opciones de exportar para realizar una copia de seguridad completa de la base de datos *actividades* a un fichero de texto. Abre posteriormente dicho fichero con un editor de textos y comprueba el resultado.
- 15 Configura ODBC de manera que puedas conectarte por esta vía tanto a la base de datos *ejercicios* como a *actividades*. Necesitarás, por tanto, dos entradas diferentes para el mismo *driver ODBC*.

- Abrazo mortal**, 21
- Acceso directo por posición**, 17
- Acceso directo por valor**, 17
- Acceso secuencial por posición**, 17
- Acceso secuencial por valor**, 17
- Access**, 67, 68, 69, 70, 71, 72, 73, 74, 76, 79, 80, 81, 84, 85, 87, 88, 89, 90, 92, 93, 94, 161, 178
- Actualizar**, 77, 78, 85, 88
- AddNew**, 177
- Administrador**, 26, 27
- Adodb**, 130
- Adox**, 130
- And**, 100, 105, 107, 108, 109, 110, 114, 117, 179
- Any**, 119
- Aplicación de bases de datos**, 129
- Archivo**, 14, 15, 22, 24
- Arquitectura cliente/servidor**, 27, 28
- As**, 112, 113, 115
- Asc**, 102
- Asistente de administración**, 188
- Asistente de configuración**, 204
- Atributo**, 12, 13, 14, 15, 17, 23, 35, 36, 38, 39, 40, 41, 44, 45, 46, 50, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62,
- Atributo multivalor**, 15, 58
- Autoincrementable**, 239, 240
- Avg**, 100, 110, 111, 112, 113, 114, 115, 119, 126
- Base de datos**, 7, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30,
- Base de datos distribuida**, 29
- Base de datos replicada**, 29
- Begin declare section**, 155, 156, 157
- Beta**, 226
- Between**, 100, 105, 108, 109, 114, 127
- BOF**, 179
- Borrar**, 79
- Caja de texto**, 164, 165, 169, 175, 177, 178, 179, 182, 183
- Campo**, 14, 15, 17, 20, 21, 25, 72, 74, 75, 82
- Caption**, 163, 164, 172, 173, 175, 180
- Cardinalidad de una relación**, 43, 48
- Cardinalidad de una tabla**, 50
- Cascada**, 77
- Case**, 26
- CDate**, 177
- Checkbox**, 169
- Cláusula**, 100, 102, 103, 104
- Clave**, 13, 14, 17
- Clave ajena**, 50, 55, 56, 57, 59, 60, 63, 123, 124, 135, 136, 138,
- Clave primaria**, 14, 38, 39, 50, 51, 52, 53, 55, 56, 57, 58, 59, 60, 61, 62, 63, 74, 123, 124, 135, 136, 138,
- Clave principal**, 74, 75, 123
- Claves alternativas**, 14
- Claves candidatas**, 14
- Cliente en modo texto**, 231, 232
- Close base de datos**, 131, 132, 134, 137, 139, 143
- Close cursor**, 157
- CommandButton**, 164
- Compartición**, 19, 21, 22
- Config**, 256
- Config.inc.php**, 256
- Conjunto**, 99, 109, 115, 117, 124, 126
- Connect**, 156, 157
- Connect reset**, 156, 157
- Constraint**, 135, 136, 137, 138, 139, 140, 141, 142, 143
- Consulta**, 67, 72, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 92, 174, 180, 181, 182, 183
- Consulta multitabla**, 115, 116
- Control Data**, 174, 175, 176, 177, 178, 180, 181, 182
- Cookies**, 256
- Copiar**, 79
- Count**, 100, 111, 112, 115
- Create**, 100
- Create database**, 132
- Create table**, 133, 134, 135, 136, 137, 138, 139, 140, 142, 143, 147, 148, 149, 150
- Create user**, 129, 146, 147
- Create view**, 145, 147, 149
- Criterio**, 84, 85
- Cruzadas**, 86, 87
- Cuadro de texto**, 163, 164, 165, 174, 177, 178
- Cuarta forma normal (4FN)**, 58, 61, 62, 63
- Cuota**, 214
- Cursor**, 157
- Dao**, 180
- Data bound grid control**, 175
- Database**, 174, 176, 180, 181, 182, 183
- Databasename**, 174, 176
- Datasource**, 174, 175, 176, 177, 180
- Date()**, 122, 123
- Datevalue**, 183
- Dato**, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31,
- Dbase**, 161
- DBGrid**, 175, 176, 180, 181
- DCL**, 99, 129
- DDL**, 99, 100, 129
- Declare cursor**, 157
- Delete**, 86, 100, 125, 126, 129, 145, 148, 179, 182, 183
- Delphi**, 161
- Dependencia funcional**, 59, 60, 61, 63
- Dependencia multievaluada**, 62
- Dependencia transitiva**, 60
- Desc**, 99, 102, 104, 114, 115
- Detalle**, 83
- Dim**, 179, 181, 182, 183
- DML**, 99, 100, 126, 129
- Dominio**, 12, 13
- Drop database**, 132
- Drop table**, 144, 147
- EasyPHP**, 246, 247, 248, 249, 256, 257
- Editor de Visual Basic**, 129
- Eliminación**, 79, 85, 86
- Else**, 166, 168, 179
- End**, 166, 168, 172, 179, 182, 183
- End declare section**, 155, 156, 157
- End if**, 166, 179
- Entidad**, 8, 12, 13, 14, 15, 16, 19, 23, 25, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57
- Entidad débil**, 39, 45, 55, 56, 57
- Entidad fuerte**, 39, 55, 57,
- EOF**, 179
- Espacio de tablas (tablespace)**, 202, 203, 204, 211, 212, 214, 220
- Evento**, 165, 167, 169, 170, 171, 172, 173, 177, 178, 180, 181, 182
- Exec sql**, 155, 156, 157
- Execute**, 183
- F5**, 166, 175, 176, 177, 178
- Fecha/Hora**, 72
- Fetch**, 157
- Fichero**, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 31,
- Fichero indexado**, 17
- Ficheros calculados**, 17
- Ficheros de constantes**, 16
- Ficheros externos**, 16
- Ficheros históricos**, 16
- Ficheros internos**, 16
- Ficheros maestros**, 16
- Ficheros relativos**, 17
- Ficheros secuenciales**, 17
- Fields**, 177, 179
- For**, 168
- Foreign key**, 138, 140, 141, 142
- Forma normal de Boyce-Codd (FNBC)**, 57, 61, 62
- Formateo de salidas**, 198, 199, 200
- Formatos**, 208, 209, 210
- Formulario**, 67, 72, 80, 88, 89, 90, 91,



- 92, 93, 95, 161, 162, 163, 164, 165, 167, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 182, 183
From, 100, 102, 103, 104, 105, 106, 107, 108, 109, 181, 182, 183
Funciones de agregado, 100, 126
Grado, 43, 50
Grant, 129, 147, 148, 149, 154
Grid computing, 187, 188
Group by, 100, 111, 112, 113, 115, 119
Having, 100, 113, 115, 119
Host string/Cadena de conexión, 194, 206
http, 227, 256
Identificador, 13, 14
If, 166, 167, 168, 179
Imprimir, 79, 80, 88, 92
In, 100, 105, 109, 114
Inconsistencia, 21
Índice, 17, 18, 204, 212, 213
Información, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 29, 30, 31
Informe, 67, 72, 80, 92, 93, 94, 95
InnoDB, 239, 251, 253, 254
InputDialog, 166, 167
Insert, 129, 145, 148, 149, 182, 183
Insert into, 120, 121, 122, 123, 126
Insert into select, 123
Instalación avanzada, 191
Instalación básica, 191
Instancia, 193, 194, 195, 202, 207, 219
Integer, 179
Integridad, 19, 20, 22, 24, 25, 52, 53, 60
Integridad referencial, 78, 130, 138, 144
Interbase, 100
Interfaz gráfica, 162, 165
Intervalo de impresión, 80
iSQL *Plus, 188, 200, 201, 202, 206, 211
Join, 118, 119
Label, 163, 174, 175
Langues.txt, 246, 248
LCD, 26
LDD, 26
Like, 100, 102, 105, 110, 114
Listener, 193, 219
LMD, 26
Load, 169, 170, 180
Load data infile, 237, 245
LostFocus, 169
Max, 100, 110, 111, 112, 113, 114, 126
Memo, 72
Memoria, 9, 10, 11, 12, 16,
Memoria externa, 10, 11, 12, 16
Memoria interna, 9, 10, 11, 12, 16
Menú, 172, 173
Min, 100, 111, 112, 113, 114, 119, 121, 126
Modelo de datos, 35
Modelo en red, 36, 49
Modelo entidad/relación, 35, 37
Modelo jerárquico, 36, 49
Modelo orientado a datos, 37
Modelo orientado a objetos, 36
Modelo orientado al concepto/semántico, 36, 37, 50
Modelo relacional, 35, 36, 37, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
Moneda, 72
Movefirst, 179
Movelast, 177
Movenext, 179
Msgbox, 165, 166, 167, 179
My.ini, 233, 234, 249
MyIsam, 239, 251, 253
MySQL, 100
Mysqldump, 244
Name, 172, 173
Natural join, 62
Neptuno, 21
Next, 168
Normalización, 54, 57, 59, 62, 63,
Not, 100, 105, 107, 108, 110, 179
Not null, 135, 136, 137, 138, 139, 142, 143
Número, 72
Objeto, 129
ODBC, 245, 257
Office, 68, 69, 70, 71, 73, 89, 92,
Ole DB, 130
Open base de datos, 134, 137, 139, 142, 143
Open cursor, 157
Open-source, 225, 226
Opendatabase, 182, 183
Openrecordset, 181, 182, 183
Operador de comparación, 102, 110
Operador de relación, 107, 108
Or, 101, 102, 105, 107, 109, 110, 114
Oracle, 68, 99, 100, 161
Oracle Application Server 10g, 188
Oracle Database 10g, 188, 190
Oracle Enterprise Manager 10g, 188
Order by, 100, 102, 104, 111, 113, 114, 115, 119
Parámetro, 81, 84, 85
Particiones, 39, 40, 45, 46, 56,
Path, 229, 231, 232
Perfil, 205, 214, 216, 218
Permisos, 146, 147, 150, 151, 152, 153, 154
phpMyAdmin, 246, 248, 252, 256, 257
Plantilla, 188, 204, 205, 206, 207
Primary key, 136, 137, 138, 139, 140, 143
Primera forma normal (1FN), 57, 58, 59, 61, 62
Privilegios, 146, 147, 148, 149, 150, 152, 154, 214, 215, 216, 217, 218
Privilegios de objeto, 215
Privilegios del sistema, 215, 217
Privilegios específicos, 256
Privilegios globales, 256
Procedimiento, 131
Programa, 129, 155, 156, 157
Proyecto, 162, 165, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 180, 181
Quinta forma normal (5FN), 58, 62
Recordset, 177, 179, 181, 182, 183
Recordsource, 174, 176
Redundancia, 19, 20, 21
Registro, 14, 15, 17, 18, 19, 20, 22, 23, 27
Relación, 19, 22, 35, 36, 37, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 60, 72, 76, 77, 78, 87
Relación 1:1, 41, 43, 49, 55
Relación 1:M, 41, 42, 43, 48, 49, 55, 56, 57
Relación binaria, 22, 40, 43, 48, 49
Relación M:N, 41, 42, 43, 48, 55, 57,
Relación reflexiva, 22, 40, 43
Relación ternaria, 22, 40, 43, 48, 49, 55
Relación universal, 54, 58
Repositorio, 26
Restricción, 41, 53, 58,
Restricciones de columna, 135, 136, 141, 142
Restricciones de tabla, 137, 138, 139, 141, 142
Resumen, 83, 85, 94
Revoke, 129, 148, 149, 154
Rman, 219
Rol, 214, 216, 217, 218, 221
Roles, 146, 149, 150, 151, 152
Root, 229, 230, 232, 244, 256, 257
Segmento, 204
Segunda forma normal (2FN), 57, 59, 60, 61, 62
Seguridad, 130, 144, 146, 150, 151, 152, 153, 154
Selección, 71, 73, 75, 77, 81, 82, 83, 90, 94
Select, 129, 145, 146, 148, 149, 155, 156, 157, 180, 181, 182, 183
Sentencia externa, 119
Sentencia subordinada, 119



- Set**, 181, 182, 183
- SGBD**, 67, 68, 72, 81, 99
- SGBDR**, 99
- Show**, 170
- Sí/No**, 72
- Sid**, 192, 205
- Sistema de información**, 10, 11, 12
- Sistema gestor de ficheros**, 18
- Sistema informático**, 7, 10, 11, 12, 14, 21
- Sistemas gestores de bases de datos**, 24, 25, 27,
- SQL**, 26, 81, 85, 86, 99, 161, 180, 181, 182, 183
- SQL *Plus**, 188, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 206, 209, 211, 212, 214
- Sql close**, 157
- SQL embebido**, 155, 157
- Sql open**, 157
- String**, 181, 183
- Sub**, 131, 132, 134, 136, 139, 142, 143, 145
- Subconsulta**, 119, 122, 123, 125, 126
- Subentidades**, 39, 45, 46, 47, 56,
- Sum**, 100, 110, 111, 114, 115
- Tabla**, 35, 36, 38, 40, 41, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 67, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 88, 89, 90, 91, 92, 93, 94
- Tercera forma normal (3FN)**, 57, 60, 61, 62, 63
- Text**, 164, 171, 174, 177, 178, 179, 182, 183
- Textbox**, 163
- Texto**, 72
- Then**, 166, 179
- Tipo de datos**, 133, 151
- Tupla**, 50
- Unique**, 136, 137, 138
- Unload**, 169, 170
- Update**, 85, 100, 123, 124, 126, 129, 145, 148, 156, 157, 182, 183
- Usuarios**, 144, 146, 147, 148, 149, 150, 152, 153, 154
- Val**, 171, 177, 182, 183
- Valor nulo**, 239, 240
- Valor por defecto**, 240, 250
- Valor único**, 239
- Variable**, 131, 168, 169, 178, 179, 180, 181, 182, 183
- Variables huéspedes**, 155, 156, 157
- VBA**, 161
- Vista**, 144, 147, 148, 149
- Vista de relaciones**, 254, 255
- Vista preliminar**, 80
- Visual Basic**, 129, 161, 162, 164, 165, 166, 168, 169, 171, 172, 173, 174, 176, 177, 180, 182, 183,
- Visual C**, 161
- Visual Fox pro**, 161
- Visual J**, 161
- Visual Net**, 161
- Wend**, 168, 179
- Where**, 100, 102, 103, 106, 107, 108, 109, 110, 113, 116, 182, 183
- While**, 168, 178, 179, 180
- Workspace**, 201



Uso del CD del alumno

Como habrás podido observar, este libro viene acompañado de un CD con información complementaria.

En esta sección queremos darte algunas orientaciones para que puedas sacarle el mayor provecho posible.

A Requisitos

El CD está preparado para funcionar bajo los sistemas operativos Windows, Macintosh y Linux. Los únicos requisitos para su correcto funcionamiento son:

- Un navegador, preferiblemente Internet Explorer 6.0 o superior.
- Un *plug-in* de Flash para poder verlo en el navegador instalado. La aplicación comprueba que esté el *plug-in* necesario y en caso contrario ofrece una dirección de Internet para instalarlo.
- Conexión a Internet para poder consultar el contenido de los diferentes menús.

B Arranque

El CD contiene un fichero de auto-arranque que permitirá que nada más introducir el CD en el lector, éste se inicie solo.

En caso de que esta característica estuviera deshabilitada desde el sistema operativo, bastaría con hacer doble clic sobre el archivo «logo.html» para que se iniciara la aplicación.

C Estructura

El CD está dividido en cuatro apartados: *Documentación*, *Descargas*, *Enlaces* y *Glosario*. Vamos a explicarte brevemente lo que puedes encontrar en cada uno de ellos.

Documentación

Cuando escojas una Unidad determinada, verás una lista desplegable que muestra la documentación que hemos seleccionado para ampliar los contenidos de esa Unidad.

Sólo tienes que pinchar sobre el descriptor que te interese y el archivo o la página web en cuestión se abrirá directamente en el marco principal de la interfaz. Recuerda que tienes un menú por cada Unidad.

Descargas

En este apartado, hemos incluido una serie de descargas de utilidades que te ayudarán a profundizar en los contenidos del módulo. Cuando abras esta sección, encontrarás un listado con el nombre de la utilidad, la página web desde la que descargarla, la plataforma sobre la que funciona y una breve descripción para que sepas lo que puedes hacer con ella. Si quieres realizar la descarga, sólo tienes que pinchar sobre la página web correspondiente.

En algunos casos, el link te llevará directamente al cuadro de diálogo de la descarga; en otros casos, hemos considerado más útil que el link conduzca a la página web que alberga el archivo para que seas tú quien busque la descarga correspondiente. La razón para esto es que, en un futuro, algunas páginas web podrán ofrecer actualizaciones o nuevas versiones de las utilidades que adjuntamos y no queremos que te las pierdas dirigiéndote directamente a un archivo que puede no existir ya.

Enlaces

En este apartado, hemos colocado un listado bastante extenso de todas las páginas web que pueden ofrecerte más datos para el estudio de la materia. Las hemos clasificado por categorías, de manera que esta sección sea un instrumento útil para tu futura carrera profesional. Encontrarás estas páginas agrupadas en las siguientes secciones:

- Arquitectura de red
- Clientes
- Comunicaciones
- Documentación
- Entidades y Asociaciones
- Estándares
- Gestión remota
- Internet
- Mensajería
- Monitorización
- Redes
- Seguridad
- Servidores
- Utilidades

Glosario

En esta última sección, hemos incorporado todos los términos que aparecen en las Unidades del libro. Están ordenados alfabéticamente, de manera que sólo deberás pinchar en la letra correspondiente para encontrar el término que busca.

