

Taller de MySQL



Miguel Angel Alvarez
Carlos Luis Cuenca
Heisler Palma
Rubén Alvarez
Y otros...



desarrolloweb.com/manuales/taller-mysql.html

Introducción: Taller de MySQL

Este es el Manual de MySQL de Desarrolloweb.com, que te ofrece gran cantidad de información para que puedas comenzar a trabajar con este sistema gestor de bases de datos y resolver los problemas y necesidades más frecuentes con los que se enfrentan los desarrolladores.

Este manual está compuesto por diversos artículos prácticos sobre el trabajo con las bases de datos MySQL, que cubren aspectos desde su instalación al trabajo habitual en el día a día.

Hemos escrito este manual varias personas, a lo largo de muchos años. No tienes que pensar en él como una descripción exhaustiva de MySQL, con todos los detalles que debes de conocer para usarlo de manera profesional. Más bien es un taller, en el que vamos resolviendo diferentes prácticas habituales y necesarias para el desarrollo de sitios web y la administración de servidores de base de datos, el acceso cómodo a la información, la creación de backups y muchas otras cosas.

Esperemos que disfrutes de este manual y puedas seleccionar aquellos temas que te interesen, volviendo una y otra vez sobre este texto para consultar el taller que más te interese dependiendo de tu flujo de trabajo, o tu nivel de aprendizaje sobre las bases de datos en general o MySQL en particular.

Encuentras este manual online en:

<http://desarrolloweb.com/manuales/taller-mysql.html>

Autores del manual

Las siguientes personas han participado como autores escribiendo artículos de este manual.

Miguel Angel Alvarez

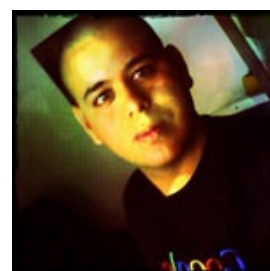
Miguel es fundador de DesarrolloWeb.com y la plataforma de formación online EscuelaIT. Comenzó en el mundo del desarrollo web en el año 1997, transformando su hobby en su trabajo.



Carlos Luis Cuenca

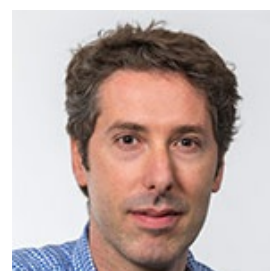
Heisler Palma

Programador autodidacta, músico e investigador de la mente.



Rubén Alvarez

Rubén es doctor en química y programador aficionado con experiencia en PHP.



Russvell Oblitas Valenzuela

Bch. Sistemas e Informática

Manu Gutierrez

Introducción a MySQL

Artículos que te sirven de introducción a MySQL en general, explicando las bases de este sistema gestor de bases de datos.

Instalación de MySQL en Windows

Pasos a seguir para la instalación de la base de datos MySQL y algunas ayudas básicas para comenzar con buen pie.

Uno de los puntos fuertes de las páginas en PHP es la posibilidad de explotar bases de datos mediante funciones de una simplicidad y potencia muy agradecidas. Estas bases de datos pueden servir a nuestro sitio para almacenar contenidos de una forma sistemática que nos permita clasificarlos, buscarlos y editarlos rápida y fácilmente.

Una base de datos es sencillamente un conjunto de tablas en las que almacenamos distintos registros (artículos de una tienda virtual, proveedores o clientes de una empresa, películas en cartelera en el cine...). Estos registros son catalogados en función de distintos parámetros que los caracterizan y que presentan una utilidad a la hora de clasificarlos. Así, por ejemplo, los artículos de una tienda virtual podrían catalogarse a partir de distintos campos como puede ser un número de referencia, nombre del artículo, descripción, precio, proveedor...

La base de datos más difundida con el tandem UNIX-Apache es sin duda MySQL. Como para el caso de Apache, una versión para Windows está disponible y puede ser [descargada](#) gratis.

Su puesta a punto no entraña mucha dificultad. Una vez instalado el programa podemos ejecutar nuestras ordenes en modo MS-DOS. Para ello abrimos una ventana MS-DOS y nos colocamos en el directorio *bin* de *mysql*. En este directorio se encuentran los archivos ejecutables. Aquí habrá que encontrar un archivo llamado *mysqld*. En el caso de la versión más actual durante la redacción de este artículo este archivo es llamado *mysqld-shareware*. Una vez ejecutado este archivo podemos ejecutar el siguiente: *mysql*.

Llegados a este punto veremos cómo un mensaje de bienvenida aparece en nuestra pantalla. En estos momentos nos encontramos dentro de la base de datos. A partir de ahí podemos realizar todo tipo de operaciones por sentencias SQL.

No vamos a entrar en una explicación pormenorizada del funcionamiento de esta base de datos ya que esto nos daría para un manual entero. Daremos como referencia nuestro [tutorial de SQL](#) a partir del cual se puede tener una idea muy práctica de las sentencias necesarias para la [creación](#) y edición de las tablas. También existe una documentación extensa en inglés en el directorio *Docs* de MySQL. A modo de resumen, aquí os proponemos además las operaciones más básicas que, combinadas nuestro [tutorial de SQL](#) pueden dar solución a gran parte de los casos que se os presenten:

Instrucción	Descripción
Show databases;	Muestra el conjunto de bases de datos presentes en el servidor
Use nombre_de_la_base	Determina la base de datos sobre la que vamos a trabajar
Create Database nombre_de_la_base;	Crea una nueva bd con el nombre especificado
Drop Database nombre_de_la_base;	Elimina la base de datos del nombre especificado
Show tables;	Muestra las tablas presentes en la base de datos actual
Describe nombre_de_la_tabla;	Describe los campos que componen la tabla
Drop Table nombre_de_la_tabla;	Borra la tabla de la base de datos
Load Data Local Infile "archivo.txt" Into Table nombre_de_la_tabla;	Crea los registros de la tabla a partir de un fichero de texto en el que separamos por tabulaciones todos los campos de un mismo registro.
Quit	Salir de MySQL

Para evitarnos el tener que editar nuestras tablas directamente sobre archivos de texto, puede resultar muy práctico usar cualquier otra base de datos con un editor y exportar a continuación la tabla en un archivo de texto configurado para dejar tabulaciones entre cada campo. Esto es posible en Access por ejemplo pinchando con el botón derecho sobre la tabla que queremos convertir y eligiendo la opción exportar. Una ventana de dialogo aparecerá en la que elegiremos guardar el archivo en tipo texto. El paso siguiente será elegir un formato delimitado por tabulaciones sin cualificador de texto.

Otra posibilidad que puede resultar muy práctica y que nos evita trabajar continuamente tecleando órdenes al estilo de antaño es servirse de programas en PHP o Perl ya existentes y descargables en la red. El más popular sin duda es [phpMyAdmin](#). Este tipo de scripts son ejecutados desde un navegador y pueden ser por tanto albergados en nuestro servidor o empleados en local para, a partir de ellos, administrar MySQL de una forma menos sufrida.

Asimismo, dentro del directorio bin de MySQL, podemos encontrar una pequeña aplicación llamada MySqlManager. Se trata de una interface windows, más agradable a la vista y al uso que la que obtenemos ejecutando el archivo *mysql*. En este caso, las sentencias SQL deben

realizarse sin el punto y coma final.

Este artículo es obra de *Rubén Alvarez*
Fue publicado por primera vez en 01/01/2001
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Tipos de Datos de Mysql

Listado y descripción de los distintos tipos de datos de MySQL.

Después de la fase de diseño de una base de datos, y una vez se ha realizado el paso a tablas del mismo, es necesario crear las tablas correspondientes dentro de la base de datos. Para cada campo de cada una de las tablas, es necesario determinar el tipo de datos que contiene, para de esa forma ajustar el diseño de la base de datos, y conseguir un almacenamiento óptimo con la menor utilización de espacio. El presente artículo describe cada uno de los tipos de datos que puede tener un campo en Mysql, para la versión 4.xx.xx.

Los tipos de datos que puede haber en un campo, se pueden agrupar en tres grandes grupos:

1. [Tipos numéricos](#)
2. [Tipos de Fecha](#)
3. [Tipos de Cadena](#)

1 Tipos numéricos:

Existen tipos de datos numéricos, que se pueden dividir en dos grandes grupos, los que están en coma flotante (con decimales) y los que no.

TinyInt:

Es un número entero con o sin signo. Con signo el rango de valores válidos va desde -128 a 127. Sin signo, el rango de valores es de 0 a 255

Bit ó Bool:

Un número entero que puede ser 0 ó 1

SmallInt:

Número entero con o sin signo. Con signo el rango de valores va desde -32768 a 32767. Sin signo, el rango de valores es de 0 a 65535.

MediumInt:

Número entero con o sin signo. Con signo el rango de valores va desde -8.388.608 a 8.388.607. Sin signo el rango va desde 0 a 16777215.

Integer, Int:

Número entero con o sin signo. Con signo el rango de valores va desde -2147483648 a 2147483647. Sin signo el rango va desde 0 a 429.4967.295

BigInt:

Número entero con o sin signo. Con signo el rango de valores va desde -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807. Sin signo el rango va desde 0 a 18.446.744.073.709.551.615.

Float:

Número pequeño en coma flotante de precisión simple. Los valores válidos van desde -3.402823466E+38 a -1.175494351E-38, o y desde 1.175494351E-38 a 3.402823466E+38.

Real, Double:

Número en coma flotante de precisión doble. Los valores permitidos van desde -1.7976931348623157E+308 a -2.2250738585072014E-308, o y desde 2.2250738585072014E-308 a 1.7976931348623157E+308

Decimal, Dec, Numeric:

Número en coma flotante desempquetado. El número se almacena como una cadena

|

Tipo de Campo

|

Tamaño de Almacenamiento

||

TINYINT

|

1 byte

||

SMALLINT

|

2 bytes

||

MEDIUMINT

|

3 bytes

||

INT

|

4 bytes

||

INTEGER

|

4 bytes

||

BIGINT

|

8 bytes

||

FLOAT(X)

|

4 ú 8 bytes

||

FLOAT

|

4 bytes

||

DOUBLE

|

8 bytes

||

DOUBLE PRECISION

|

8 bytes

||

REAL

|

8 bytes

||

DECIMAL(M,D)

|

M+2 bytes sí $D > 0$, M+1 bytes sí $D = 0$

||

NUMERIC(M,D)

|

M+2 bytes if $D > 0$, M+1 bytes if $D = 0$

|

2 Tipos fecha:

A la hora de almacenar fechas, hay que tener en cuenta que Mysql no comprueba de una manera estricta si una fecha es válida o no. Simplemente comprueba que el mes esta comprendido entre 0 y 12 y que el día esta comprendido entre 0 y 31.

Date:

Tipo fecha, almacena una fecha. El rango de valores va desde el 1 de enero del 1001 al 31 de diciembre de 9999. El formato de almacenamiento es de año-mes-día

DateTime:

Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-día horas:minutos:segundos

TimeStamp:

Combinación de fecha y hora. El rango va desde el 1 de enero de 1970 al año 2037. El formato de almacenamiento depende del tamaño del campo:

|

Tamaño

|

Formato

||

14

|

AñoMesDiaHoraMinutoSegundo aaaammdhmmss

||

12

|

AñoMesDiaHoraMinutoSegundo aammdhmmss

||

8

|

AñoMesDia aaaamdd

||

6

|

AñoMesDia aammdd

||

4

|

AñoMes aamm

||

2

|

Año aa

|

Time:

Almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. El formato de almacenamiento es de 'HH:MM:SS'

Year:

Almacena un año. El rango de valores permitidos va desde el año 1901 al año 2155. El campo puede tener tamaño dos o tamaño 4 dependiendo de si queremos almacenar el año con dos o cuatro dígitos.

|

Tipo de Campo

|

Tamaño de Almacenamiento

||

DATE

|

3 bytes

||

DATETIME

|

8 bytes

||

TIMESTAMP

|

4 bytes

||

TIME

|

3 bytes

||

YEAR

|

1 byte

|

3 Tipos de cadena:

Char(n):

Almacena una cadena de longitud fija. La cadena podrá contener desde 0 a 255 caracteres.

VarChar(n):

Almacena una cadena de longitud variable. La cadena podrá contener desde 0 a 255 caracteres.

Dentro de los tipos de cadena se pueden distinguir otros dos subtipos, los tipo Text y los tipo BLOB (Binary large Object)

La diferencia entre un tipo y otro es el tratamiento que reciben a la hora de realizar ordenamientos y comparaciones. Mientras que el tipo text se ordena sin tener en cuenta las Mayúsculas y las minúsculas, el tipo BLOB se ordena teniéndolas en cuenta.

Los tipos BLOB se utilizan para almacenar datos binarios como pueden ser ficheros.

TinyText y TinyBlob:

Columna con una longitud máxima de 255 caracteres.

Blob y Text:

Un texto con un máximo de 65535 caracteres.

MediumBlob y MediumText:

Un texto con un máximo de 16.777.215 caracteres.

LongBlob y LongText:

Un texto con un máximo de caracteres 4.294.967.295. Hay que tener en cuenta que debido a los protocolos de comunicación los paquetes pueden tener un máximo de 16 Mb.

Enum:

Campo que puede tener un único valor de una lista que se especifica. El tipo Enum acepta hasta 65535 valores distintos

Set:

Un campo que puede contener ninguno, uno ó varios valores de una lista. La lista puede tener un máximo de 64 valores.

|

Tipo de campo

|

Tamaño de Almacenamiento

||

CHAR(n)

|

n bytes

||

VARCHAR(n)

|

n +1 bytes

||

TINYBLOB, TINYTEXT

|

Longitud+1 bytes

||

BLOB, TEXT

|

Longitud +2 bytes

||

MEDIUMBLOB, MEDIUMTEXT

|

Longitud +3 bytes

||

LOBLOB, LONGTEXT

|

Longitud +4 bytes

||

ENUM('value1','value2',...)

|

1 ó dos bytes dependiendo del número de valores

||

SET('value1','value2',...)

|

1, 2, 3, 4 ó 8 bytes, dependiendo del número de valores

|

Diferencia de almacenamiento entre los tipos Char y VarChar

|

Valor

|

CHAR(4)

|

Almacenamiento

|

VARCHAR(4)

|

Almacenamiento

||

"

|

"

|

4 bytes

|

"

|

1 byte

||

'ab'

|

'ab '

|

4 bytes

|

'ab'

|

3 bytes

||

'abcd'

|

'abcd'

|

4 bytes

|

'abcd'

|||

'abcdefgh'

|

'abcd'

|

4 bytes

|

'abcd'

| 5 bytes |

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en *05/02/2003*

Disponible online en <http://desarrolloweb.com/articulos/34.php>

Trabajo con MySQL desde terminal

El terminal, o línea de comandos, es el modo de acceso a MySQL de más bajo nivel (más cercano a la máquina y menos centrado en el usuario). Te interesa dominar el acceso por línea de comandos antes que cualquier otra cosa, pues es el modo que siempre vas a tener disponible para trabajar con MySQL. Conocer bien el terminal y los comandos para conectar y operar con MySQL será básico en tu formación y te salvará de algún que otro problema. En estos artículos nos centramos en Linux, puesto que los servidores donde vas a tener instalado MySQL habitualmente son basados en Linux, aunque los comandos de acceso a MySQL son los mismos en cualquier sistema operativo.

MySQL por línea de comandos

Acceso a un servidor MySQL y administración de la base de datos por línea de comandos.

Es muy normal que utilicemos MySQL a través de páginas PHP y para administrar la base de datos utilicemos un programa como PhpMyAdmin, pero a veces no nos queda otro remedio que acceder a la base de datos a través de la línea de comandos.

MySQL tiene un programa, que se llama con el mismo nombre de la base de datos (mysql) que sirve para gestionar la base de datos por línea de comandos. Ese programa, en una instalación de Windows se encuentra en un directorio como

C:\Archivos de programa\MySQL\MySQL Server 4.1\bin

El directorio puede variar, por ejemplo, puede estar localizado en la raíz del disco C:, o en cualquier otro lugar donde podamos haber instalado MySQL. Para acceder a la consola de MySQL en Windows tendremos que estar situados dentro de ese directorio.

En Linux, por supuesto, también se puede acceder a MySQL por línea de comandos. Posiblemente desde cualquier directorio podamos acceder a la consola de MySQL, sin necesidad de situarse en el directorio donde esté instalado.

Conectar con el servidor MySQL

Lo primero que tendremos que hacer es conectar con el sistema gestor de MySQL. Para ello, desde la línea de comandos invocamos a MySQL. Para ello, simplemente tenemos que escribir el comando "mysql" e indicarle unas opciones de conexión.

```
% mysql
```

Con el "%" expresamos el principio de la línea de comandos. Ese principio es el prompt que tengamos en nuestra consola de Linux o MsDOS, que puede ser algo como c:\mysql\bin>. El carácter "%", por tanto, no tenemos que escribirlo.

Con esa sentencia se conecta uno con la base de datos con los parámetros por defecto. Es decir, al servidor local, con usuario y password igual a cadenas vacías.

Lo más normal es que tengamos que indicar algún otro dato para conectar con la base de datos, como el usuario, la clave o la dirección del servidor con el que queremos conectar. La sintaxis sería la siguiente:

```
% mysql -h nombre_servidor -u nombre_usuario -p
```

Si deseamos conectarnos a la base de datos en local y con nombre de usuario root tendríamos que escribir:

```
% mysql -h localhost -u root -p
```

Lo primero que nos preguntará será el password para el usuario root. Una vez introducida la clave, ya estaremos dentro de la línea de comandos de MySQL. Con ello el prompt cambiará a algo como esto:

```
mysql>
```

Podríamos haber indicado la contraseña directamente en la línea de comandos para iniciar la sesión con MySQL, pero esto se desaconseja por razones de seguridad. De todos modos, la sintaxis hubiera sido:

```
% mysql -h localhost -u root -pmi_clave
```

Nos fijamos que entre -h y el nombre del host hay un espacio en blanco, igual que entre -u y el nombre de usuario. Sin embargo, entre -p y la clave no debemos poner ningún espacio.

Dentro de la consola de MySQL

Una vez dentro, tendremos a nuestra disposición todas las sentencias de MySQL para el trabajo con la base de datos y el lenguaje SQL.

Lo más normal es que primero te tengas que conectar con una base de datos en concreto, de entre todas las que puedes tener creadas en tu servidor MySQL. Eso se hace con el comando use, seguido del nombre de la base de datos que desees conectar.

```
mysql> use mibasedatos;
```

Esto nos conectaría con la base de datos llamada "mibasedatos".

Nota: Hay que fijarse que todas las sentencias dentro de la línea de comandos de MySQL acaban en ";". Si no colocamos el punto y coma, lo más seguro es que no se ejecute el comando y nos vuelva a salir el prompt para que sigamos introduciendo el comando. Si lo que queríamos era ejecutar la sentencia que habíamos escrito antes, con simplemente entrar el ";" será suficiente. Es decir, no debemos escribir de nuevo la sentencia entera, sólo el ";" y volver a apretar "enter".

Si queremos ver una lista de las bases de datos alojadas en nuestro servidor podemos escribir el comando `show databases`. Así:

```
mysql>show databases;
```

Con esto nos mostraría una lista de las bases de datos de nuestro servidor. Algo como esto:

```
mysql> show databases
-> ;
```

```
+-----+
| Database |
+-----+
| colaboracion |
| local |
| m3000 |
| mysql |
| test |
+-----+
```

```
5 rows in set (0.02 sec)
```

Si queremos crear una base de datos, podremos hacerlo con el comando "create database" seguido del nombre de la nueva base de datos.

```
mysql> create database miprueba;
```

Eso nos creará una base de datos que se llama "miprueba". Como habíamos comentado, si queremos luego usar esa base de datos escribiríamos:

```
mysql> use miprueba;
```

Lógicamente, esta base de datos recién creada estará vacía, pero si estuviéramos usando una base de datos ya creada y queremos ver las tablas que tiene escribiríamos el comando "show tables".

```
mysql> show tables;
```

Si no hay tablas, nos dirá algo como "Empty set", pero si tenemos varias tablas dadas de alta en la base de datos que estamos usando, nos saldrá una lista de ellas:

```
+-----+
| Tables_in_colaboracion |
+-----+
| administrador          |
| prueba                 |
+-----+
```

```
2 rows in set (0.00 sec)
```

Ahora, si deseamos obtener información sobre una tabla, para saber qué campos tiene y de qué tipo, podremos utilizar el comando describe seguido del nombre de la tabla.

```
mysql> describe administrador;
```

Field	Type	Null	Key	Default	Extra
id_administrador	int(11)		PRI	NULL	auto_increment
email_administrador	varchar(100)		MUL		
clave_administrador	varchar(100)				
nombre_administrador	varchar(100)				

```
4 rows in set (0.11 sec)
```

Otras Sentencias SQL

Desde la consola de MySQL podemos indicar por línea de comandos todo tipo de sentencias en lenguaje SQL, como selecciones, inserciones, actualizaciones, creación de tablas, etc. El mecanismo es el que se puede deducir. Simplemente colocamos la sentencia a ejecutar seguida del punto y coma. Veamos una serie de sentencias seguidas y el resultado de ejecutarlas:


```
mysql> create table prueba (id_prueba int);
Query OK, 0 rows affected (0.08 sec)

mysql> insert into prueba (id_prueba) values (1);
Query OK, 1 row affected (0.00 sec)

mysql> insert into prueba (id_prueba) values (2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into prueba (id_prueba) values (3);
Query OK, 1 row affected (0.00 sec)

mysql> select * from prueba;
```

```
+-----+
| id_prueba |
+-----+
|          1 |
|          2 |
|          3 |
+-----+
```

```
3 rows in set (0.00 sec)
```

En definitiva, podemos ejecutar todas las sentencias que se han aprendido en el [manual del lenguaje SQL](#)

Para salir de la línea de comandos de MySQL

Una vez hemos terminado de trabajar con MySQL, si queremos cerrar la conexión con el servidor, simplemente escribimos "quit" desde el prompt de MySQL:

```
mysql> quit
```

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 15/03/2006
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Acceder a MySQL desde terminal en Linux

Acceso y operaciones más básicas para administrar MySQL en un sistema Linux cuando accedemos por el terminal en línea de comandos.

En DesarrolloWeb.com publicamos hace algún tiempo unas notas muy interesantes para todo aquel que trabaja con MySQL y desea [acceder por línea de comandos al sistema gestor de base de datos](#). Ese artículo estaba orientado hacia las personas que trabajan con Windows, aunque es verdad que la mayoría de sentencias de administración funcionan igual en Linux, porque en definitiva lo que se está haciendo es comunicar con MySQL a través del lenguaje SQL, que no difiere en nada de un sistema a otro.

Sin embargo, para las personas que trabajan con MySQL por GNU/Linux, ya sea porque tengan instalado Linux como ordenador de escritorio o porque accedan por SSH a servidores dedicados, algunos asuntos relacionados con el acceso a la interfaz de MySQL cambian un poco. En este artículo vamos a comentar algunos aspectos que podrán interesar a esos usuarios noveles que están deseando unas orientaciones básicas.

Comenzaremos dando algunos detalles sobre cómo acceder a la línea de comandos del sistema Linux, ya sea en tu sistema local, ya sea en un sistema Linux remoto que puedas tener en un servidor dedicado.

Acceso a un servidor por SSH

Como primer paso vamos a comentar cómo hacer una conexión por SSH a un servidor dedicado, o un servidor virtual o hosting, que permita la conexión por SSH. Esto no tiene que ver específicamente con la administración de MySQL, pero puede servir de guía para las personas que están deseando conectarse con un servidor remoto para luego acceder a dicho sistema gestor de base de datos.

En este caso puede que tengas varios sistemas operativos distintos y que desees acceder al servidor Linux por cualquiera de ellos.

Windows:

Para acceder por SSH a un servidor remoto desde tu ordenador de escritorio con Windows yo suelo utilizar el programa Putty, que es superligero y gratuito. Puedes ver información sobre este software en el artículo [Putty: Programa para Telnet y SSH](#)

Linux:

El programa Putty también está disponible para Linux, pero es totalmente innecesario, ya que a través del terminal del propio sistema Linux tienes acceso al comando SSH para conectarte con cualquier servidor.

```
ssh usuario@1.2.3.4
```

Por supuesto, tendrás que editar tu usuario y la IP del servidor al que te quieres conectar. Por ejemplo, para conectarse por root en el servidor con IP 280.230.2.67 escribirías:

```
ssh root@280.230.2.67
```

Mac:

Los ordenadores con sistema Mac OS X tienen el terminal de línea de comandos también integrada entre las aplicaciones disponibles. Para acceder por SSH a otro servidor se utiliza el mismo comando que en Linux.

Acceso a una base de datos MySQL en local en sistemas Linux

Si en tu caso tienes Linux instalado en tu ordenador de escritorio y deseas acceder a MySQL por línea de comandos, con que abras un Terminal es suficiente. Es decir, que aquí no necesitarías conectar por SSH con ningún servidor.

Nota: En mi sistema Ubuntu abro un terminal con la combinación de teclas CTRL + ALT + T. Supongo que en otras distribuciones de Linux este atajo puede cambiar, pero no debe ser muy difícil abrir un terminal. Busca el programa entre los instalados.

Acceso a la interfaz de MySQL de línea de comandos

Una vez en el servidor donde tengas el MySQL al que quieres acceder, simplemente te conectas con el comando:

```
mysql -h localhost -u root -p
```

Tendrás que indicar el usuario MySQL con el que quieras autenticarte. Probablemente sea root en tu instalación de MySQL en local, pero si estás en un servidor en producción quizás sea más inteligente conectar con el usuario y contraseña de acceso a la base de datos MySQL que quieres administrar, que son los mismos que utilizas en, por ejemplo, la función `mysql_connect()` de PHP.

Nota: El servidor MySQL al que nos conectamos es localhost, pues se supone que estás accediendo al sistema gestor que tienes en el servidor con el que te has conectado. Pero también podrías acceder a otros servidores remotos cambiando localhost por el nombre o la IP de ese servidor al que quieras acceder.

En Linux da igual en el directorio donde estés cuando ejecutes ese comando para conectar con MySQL, pues el comando está instalado en el servidor y activo desde cualquier localización.

Una vez ejecutado ese comando, se nos solicitará la clave del usuario que hemos utilizado para conectar. Si la ponemos bien, estaremos dentro de la interfaz de MySQL por línea de comandos.

Si alguna vez queremos salir de nuevo al terminal de Linux, utilizamos el comando

```
exit
```

Comandos típicos de MySQL

Una vez dentro de MySQL puedes realizar comandos para trabajar con el sistema gestor de base de datos en el [lenguaje SQL](#). Lo más típico es que realices acciones como estas:

Ver las bases de datos que tenemos creadas en el sistema:

```
show databases;
```

Nota: observa que los comandos que lanzas a MySQL deben llevar un punto y coma ";" al final, si no lo colocas el comando no se ejecutará hasta que escribas el ";" y pulses la tecla Enter de nuevo.

Cambiar a una base de datos:

```
use base_de_datos;
```

Siendo base_de_datos el nombre de la base de datos a usar.

A partir de aquí puedes realizar cualquier tipo de consultas de selección, actualización, inserción, etc. sobre la base de datos. Para ello utilizas el lenguaje de consulta estándar del cual tenemos un interesante [Tutorial de SQL](#).

Para complementar esta información puedes leer el anterior artículo sobre [Línea de comandos MySQL](#), donde explicamos otra serie de acciones disponibles cuando atacamos a MySQL desde el terminal.

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 09/05/2012
Disponible online en <http://desarrolloweb.com/articulos/mysql-terminal-linux.html>

Crear base de datos MySQL y usuario por línea de comandos

Cómo crear una base de datos MySQL nueva, usando tan solo la línea de comandos, y un usuario con permisos para acceder a ella.

Existen muchas interfaces gráficas para acceder a MySQL que nos permiten realizar de una manera sencilla las tareas de administración como las que vamos a tratar en este artículo, sin embargo, no siempre están disponibles.

En un ambiente de servidor remoto, al que accedemos generalmente por SSH muchas veces nos vemos obligados a usar la línea de comandos, por lo que debemos administrar MySQL conectándonos desde el shell. Esto no supone ningún problema, pero obviamente necesitas saber los comandos que debes usar para realizar las acciones, algo que suele ser un poco menos práctico que hacer las cosas a golpe de clic, pero muchas veces necesario.

En este pequeño taller de MySQL crearemos no solo una base de datos nueva sino además crearemos un usuario al que le asignaremos permisos para poder acceder a esa base de datos. El listado de comandos es cortesía de nuestros amigos de [ApacheCTL](#), que lo han compartido con nosotros.



Acceso a MySQL por consola

Primer paso será acceder al servidor de MySQL por línea de comandos, algo que se realiza con el comando "mysql" indicando el host, usuario y password.

Para esta tarea necesitarás el usuario "root" con el que se configuró MySQL durante la instalación.

Nota: La instalación ya depende de tu servidor. La tendrás que hacer con la línea de comandos, con las instrucciones para tu propia distribución. Usarás los repositorios oficiales y los gestores adecuados, por ejemplo Debian o Ubuntu sería "apt-get" y CentOS "yum".

El comando será algo parecido a esto:

```
mysql -h localhost -u root -p
```

Puedes encontrar otras alternativas y comandos básicos de conexión en el artículo [MySQL desde terminal Linux](#) y también en [MySQL por línea de comandos](#).

Crear la base de datos

Ya dentro del sistema gestor de MySQL crearás la base de datos con el siguiente comando "create database".

```
create database nombre_de_base_de_datos character set utf8;
```

Como ves, estamos indicando también que se use el juego de caracteres "utf8", lo que es una recomendación importante.

Crear el usuario para acceder a esta base de datos

Podrías usar el propio usuario de root para el acceso a nuestra nueva base de datos, pero por seguridad no es algo que generalmente vas a querer hacer. Es más conveniente crear un usuario nuevo al que asignar privilegios para el acceso únicamente a la base de datos que debe acceder.

Esto incluye la realización de varios comandos. En el primero de todos crearemos el usuario nuevo.

```
create user nuevo_usuario@'localhost';
```

En el segundo vamos a asignar una clave para el acceso con este usuario.

```
set password for 'nuevo_usuario'@'localhost' = PASSWORD('UnAc14v3muyDiFiCiL');
```

Ahora debemos asignar los privilegios para el acceso a la base de datos que nos interesa que acceda este usuario nuevo.

```
grant all on nombre_de_base_de_datos.* to 'nuevo_usuario'@'localhost' ;
```

Una última acción, para asegurarnos que se entregan ya mismo esos privilegios, se reinicia la caché y todo lo que haga falta para que todo esté funcionando como se espera.

```
flush privileges;
```

Con eso es todo! tenemos nuestra base de datos creada y un usuario con privilegios suficientes, listo para ser utilizado desde nuestra programación backend en cualquier lenguaje.

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 07/02/2017
Disponible online en <http://desarrolloweb.com/articulos/crear-bbdd-mysql-linea->

[comandos.html](#)

Acortar la conexión a MySQL por línea de comandos

Truco para realizar más sencilla la instrucción de conexión a la base de datos MySQL por línea de comandos.

Como hemos podido leer en el artículo de [MySQL por línea de comandos](#), para conectar con un servidor MySQL es necesario escribir una serie de parámetros en la instrucción de conexión. Entre esos parámetros se encuentra el host al que deseamos conectar, el nombre de usuario y la contraseña.

Todos los datos de conexión con el servidor MySQL hacen que la instrucción para conectar se haga un poco larga y pesada de escribir una y otra vez, en cada conexión con MySQL. En este artículo veremos una manera sencilla de recortar el proceso de acceso al servidor MySQL, evitando la necesidad de escribir una y otra vez los datos de acceso para conectar.

La idea se trata simplemente de escribir un fichero de texto donde figurarán los datos de conexión con el servidor MySQL. Cuando conectemos, en lugar de escribir esos datos en la línea de comandos, nosotros escribiremos simplemente la llamada a mysql y el propio cliente se encargará de leer ese archivo de texto para obtener los datos del host, usuario y contraseña.

El archivo que tenemos que crear se llama my.cnf. Como decimos, es un archivo de sólo texto, por lo que se podrá crear y editar con cualquier editor de texto plano, como el bloc de notas. La estructura del fichero será la siguiente:

```
[client]
host=servidor_con_el_que_conectar
user=nombre_de_usuario
password=clave
```

Este archivo de texto lo tenemos que colocar en el directorio raíz del sistema, en Windows, por ejemplo, deberíamos colocarlo en C:\my.cnf. En Linux también deberíamos colocar el fichero en el directorio raíz del sistema "/my.cnf".

Obviamente, tenemos que sustituir los datos de conexión por los que necesitemos en nuestro sistema. Un posible ejemplo de archivo de conexión podría ser el siguiente:

```
[client]
host=localhost
user=miusuario
password=loquesea
```

Los distintos parámetros son opcionales. Por ejemplo, si no tenemos password, podríamos dejar sin poner la línea donde se especifica la clave.

Eso es todo, una vez creado el fichero y colocado en el lugar adecuado, podremos acceder a MySQL simplemente con la instrucción "mysql", sin necesidad de especificar cada vez el host, usuario, clave, etc.

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 22/09/2006
Disponible online en <http://desarrolloweb.com/articulos/acortar-conexion-mysql-linea-comandos.html>

MySQL con MAMP y trabajo con el terminal

Notas sobre el trabajo de MySQL con MAMP / MAMP PRO, cómo acceder a la consola de MySQL por línea de comandos en el terminal.

La manera más popular de contar con Apache + PHP + MySQL en Mac OS X es a través del conocido MAMP, una aplicación que te ofrece todo eso y algo más en un solo paquete. Dispones de MAMP o MAMP PRO, ambos te sirven para lo mismo y ambos comparten la misma instalación de MySQL, las mismas bases de datos, etc.

En este artículo quiero dejar unas notas rápidas sobre cómo trabajar con MySQL por línea de comandos o terminal, para no olvidarme algunas investigaciones y pasos que he tenido que realizar para que el sistema me reconozca el comando mysql.



Por qué línea de comandos MySQL

Para administrar una base de datos MySQL contamos con numerosas herramientas, con interfaz web como PhpMyAdmin o con interfaz de aplicación de escritorio como MySQL Workbench o Sequel Pro. La forma menos visual es justamente la línea de comandos, por lo que quizás muchos se preguntarán sobre la necesidad de disponer de un acceso a través del terminal.

Pues bien, el terminal es un aliado que siempre está ahí. El terminal es lo único que tienes en muchos servidores para poder conectarte con una base de datos MySQL y te permite no solo realizar consultas, sino realizar backups, restaurarlos, etc. Cuando tienes un VPS o un dedicado lo que vas a tener siempre es la línea de comandos. Quizás eres de los que prefieren no publicar en tu servidor herramientas como PhpMyAdmin, por la puerta de entrada que

estás abriendo a posibles ataques. Por el mismo motivo, quizás tampoco te agrada la idea de abrir puertos destinados al trabajo con MySQL desde otras redes. Entonces lo que te queda es la línea de comandos.

Trabajar con la línea de comandos siempre te hará más libre, porque evitas depender de terceros programas. Además, si quieres reproducir en tu ordenador local las condiciones de trabajo que vas a tener con los servidores remotos y acostumbrarte al día a día con la línea de comandos, o para hacer tareas de administración de bases de datos no disponibles de otro modo, entonces te interesa usar el cliente de terminal de MySQL.

Dónde está el cliente de MySQL en MAMP

Para acceder al cliente de MySQL por terminal, para gestionar las bases de datos, yo estaba acostumbrado a usar simplemente el comando "mysql". Al menos en Windows e incluso en Windows, pero mi sorpresa es que después de instalar MAMP no está disponible ese comando. El error que te encontrarás, si te pasa como a mí, es que al lanzar el comando recibes el mensaje.

```
mysql: command not found
```

Lo primero que piensas es que no está en el PATH. Entonces te vas al directorio de tu instalación de MAMP y encontrarás el cliente de MySQL por línea de comandos. El archivo se llama mysql y está en la ruta:

```
/Applications/MAMP/Library/bin/
```

Lo más normal es que desde esa ruta invocases el comando de MySQL "mysql" y que te funcionase, pero no es así. Tampoco indicando la ruta completa al cliente /Applications/MAMP/Library/bin/mysql funciona. Hay que hacer algunos pasos

Nota: Atribuyo la autoría de esta guía a donde la he encontrado yo (en inglés):
<http://www.webbykat.com/2012/06/solving-sh-mysql-command-not-found-mamp-pro-2>

1. Primero asegúrate que tu comando "mysql" está en la ruta que te he indicado. Generalmente lo encontrarás en /Applications/MAMP/Library/bin/mysql
2. Crea un archivo llamado ".bash_profile" en tu ruta ~/ (o edita el que puedas tener anteriormente). Ten en cuenta que los archivos que comienzan por un punto "." en Mac OS X (así como en Linux) son archivos ocultos, por lo que no lo encontrarás con finder ese archivo aunque esté creado.
3. Edita ese archivo y agrega la línea.

```
export PATH=$PATH:/Applications/MAMP/Library/bin
```

7. Por favor, ten en cuenta que la ruta de mysql corresponda con la que estás agregando en el ".bash_profile"
8. Guarda el archivo
9. Reinicia el terminal, pues ese nuevo perfil no estará disponible hasta que no salgas del todo del programa terminal. Menú superior "Terminal / Salir de Terminal"

Con esto ya podrías acceder a tu comando mysql para conectar con el sistema gestor de base de datos. El comando será algo como:

```
mysql -h localhost -u root -p
```

A continuación te pedirá tu clave de "root".

Nota: No lo hemos dicho, pero tendrás que asegurarte que el servicio de MySQL está iniciado, claro está.

Para más opciones del cliente mysql de terminal, por favor, lee el artículo de [MySQL por línea de comandos](#), que es válido para cualquier tipo de sistema operativo.

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 17/12/2014
Disponible online en <http://desarrolloweb.com/articulos/mysql-mamp-trabajo-terminal.html>

Copias de seguridad de MySQL

En los siguientes artículos vamos a abordar el backup de MySQL, cómo crear copias de seguridad de bases de datos MySQL y, por supuesto, cómo restaurar esos backups cuando sea necesario. Para ello usaremos las herramientas estándar de línea de comandos, el cliente de MySQL Dump, con el que se realizan los respaldos, y el propio cliente MySQL, con el que se pueden restaurar las copias de seguridad. Sin embargo, es importante mencionar que varios programas de interfaz gráfica permiten hacer estas tareas también, solo que a golpe de clic, lo que puede ser mucho más rápido y simple. En este Manual de MySQL encontrarás una sección dedicada a programas de interfaz gráfica más adelante.

Copias de seguridad en MySQL

Conceptos a tener en cuenta a la hora de hacer una copia de seguridad y métodos disponibles para ello.

El gestor de Bases de datos MySQL incluye varias herramientas para la realización de copias de seguridad de la base de datos. Mediante ellas podremos poner a salvo nuestros datos, para que, en el eventual caso de que se pierdan, poderlos recuperar.

A la hora de hacer una copia de seguridad, lo primero que se hay que tener en cuenta es la integridad de los datos que se estén guardando. En todos los casos es necesario que haya integridad en los datos de una tabla, con esto quiero decir que todos los datos de la tabla deberán estar escritos en la misma, esto puede sonar un poco raro, pero tal y como pasa con todos los gestores de bases de datos, Mysql dispone de diversas "caches" en las que se almacenan datos temporalmente con el objetivo de mejorar en rendimiento, de forma que por ejemplo, una vez hecha una modificación en una tabla, puede ser que los datos no se guarden inmediatamente en disco, hasta que termine, por ejemplo, una consulta que se estaba ejecutando. Por esto, es necesario "forzar" a Mysql a escribir todos los datos en el disco, mediante la sentencia "Flush Tables".

Además es necesario que no se escriba en las tablas mientras se esta haciendo la copia de seguridad de la base de datos, que se consigue con el comando "lock tables", seguido del nombre de la tabla. Puede haber bases de datos en las que sea necesario bloquear todas las tablas al mismo tiempo antes de hacer la copia de seguridad.

Existen varias opciones para realizar la copia de seguridad de una base de datos de Mysql,

- En primer lugar, se podría utilizar alguna herramienta comercial que gestione todo el proceso, esto tiene como ventaja la simplicidad del método, y como inconveniente, que no suelen ser gratis, sino que hay que pagar licencia por la utilización de los mismos. Estas herramientas no se cubrirán en este reportaje.
- En segundo lugar, y a partir de la versión 3.23.25 y posteriores, existe la posibilidad de

- realizar una copia de seguridad a través de la sentencia sql "backup table".
- Como tercera opción, es posible realizar copias de seguridad a través de las herramientas que nos proporciona el propio gestor de base de datos, como pueden ser mysqldump ó mysqlhotcopy.

Nota: Llegado a este punto necesitarás acceder a los siguientes artículos, en los que se detalla el proceso de creación de las copias de seguridad MySQL por dos métodos:

- [Backup MySQL mediante Sentencias Sql](#)
- [Backup MySQL con mysqldump](#)

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en 06/06/2003
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Backup MySQL mediante Sentencias Sql

Cómo realizar un backup de las tablas de una base de datos y su recuperación a partir de sentencias SQL.

Veremos cómo realizar una copia de seguridad de la tabla y su recuperación a partir de sentencias SQL.

Backup de los datos

Este comando nos permite hacer una copia de los ficheros de las tablas de las cuales queremos hacer un backup, actualmente solo funciona con tablas de tipo MyIsam, y copia tanto los ficheros .frm que contienen la definición de la tabla, como los ficheros .myd, que contienen los datos.

Antes de ejecutarse, guarda todos los cambios que pudiera haber en memoria de la tabla, de forma que quede de una manera consistente. Asimismo, durante la ejecución del comando, bloquea la tabla sobre la que se está haciendo la copia de seguridad para que los datos sean consistentes en la tabla. Hay que tener en cuenta que este comando va bloqueando una a una las tablas, según va haciendo la copia de seguridad. Esto podría provocar inconsistencia de datos a nivel de base de datos, si es necesario hacer un backup de todas las tablas en un instante, en cuyo caso habría que utilizar el comando "lock tables" para bloquear todas las tablas antes de comenzar la copia de seguridad.

Al finalizar el comando devuelve una tabla, que contiene los campos:

| **Columna** | **Valores** | | Table | Nombre de la tabla | | Op | Siempre pone "backup" | |
Msg_type | Puede contener status, error, info o warning. | | Msg_text | Mensaje descriptivo
del resultado de la operación |

Y en la que hay un registro por cada tabla que sobre la que se ha hecho backup.

Para realizar una copia de una tabla llamada pedidos, a la carpeta Backups:

```
BACKUP TABLE pedidos TO `./backups/`
```

Recuperación de datos

Para complementar la sentencia "Backup table", Mysql dispone de la sentencia "Restore table", que permite restaurar una tabla a partir de una copia de seguridad que se ha realizado con el comando "Backup Table".

Esta opción solo se puede utilizar si la tabla que se pretende restaurar no se encuentra en la base de datos, ya que en caso afirmativo mostrará un mensaje de error. Al igual que el comando de Backup, esta opción sólo esta disponible para las tablas de tipo MyIsam, y a partir de la versión 3.23.25.

Como el comando de backup, no copia los ficheros de índices, el comando para recuperar los datos, vuelve a reindexar todos los campos que contienen índices, creando los ficheros correspondientes. Al igual que el comando para hacer copias de seguridad, invocar al comando "Restore table" devuelve una tabla, con un registro por cada tabla sobre la que se ha hecho la base de datos, y que contiene la siguiente información:

| **Columna** | **Valores** | | Table | Nombre de la tabla | | Op | Siempre pone "restore" | |
Msg_type | Puede contener status, error, info o warning. | | Msg_text | Mensaje descriptivo
del resultado de la operación |

Para restaurar una tabla llamada pedidos de la carpeta Backups a la base de datos:

```
RESTORE TABLE pedidos FROM `./backups/`
```

Nota: Para continuar aprendiendo métodos para realizar un backup de una base de datos MySQL tenemos el artículo [Backup MySQL con mysqldump](#), que enseña a manejar el comando mysqldump para hacer una copia de seguridad de la base de datos entera.

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en 06/06/2003
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Backup MySQL con mysqldump

El comando mysqldump del sistema gestor de base de datos MySQL sirve para

hacer copias de seguridad. Lo explicamos con detenimiento y vemos sus opciones más importantes.

En este artículo vamos a conocer una herramienta esencial de MySQL, para realizar copias de bases de datos, o backups, en el lenguaje SQL. Se trata de mysqldump, un comando que funciona en el terminal, compatible con cualquier sistema operativo donde corra MySQL. Comenzaremos analizando las opciones disponibles y al final daremos una serie de ejemplos de uso que resultarán útiles para situaciones variadas.

Este comando permite hacer la copia de seguridad de una o múltiples bases de datos. Además permite que estas copias de seguridad se puedan restaurar en distintos tipos de gestores de bases de datos, sin la necesidad de que se trate de un gestor de MySQL. Esto lo consigue creando unos ficheros, que contienen todas las sentencias SQL necesarias para poder restaurar la tabla, que incluyen desde la sentencia de creación de la tabla, hasta una sentencia insert por cada uno de los registros que forman parte de la misma.

Para poder restaurar la copia de seguridad, bastará con ejecutar todas las sentencias SQL que se encuentran dentro del fichero, bien desde la línea de comandos de mysql, o desde la pantalla de creación de sentencias sql de cualquier entorno gráfico como puede ser el Mysql Control Center.

Las limitaciones de la restauración dependerán de las opciones que se han especificado a la hora de hacer la copia de seguridad, por ejemplo, si se incluye la opción `--add-drop-table` al hacer la copia de seguridad, se podrán restaurar tablas que existen actualmente en el servidor (borrándolas primero). Por lo que es necesario estudiar primero los procedimientos que se utilizarán tanto en la copia como en la restauración, para que todo salga correcto!

Opciones disponibles en el comando

El comando dispone de una amplia variedad de opciones que nos permitirá realizar la copia de la forma más conveniente para el propósito de la misma. Algunas de las opciones más importantes son:

`--add-locks`

Añade LOCK TABLES antes, y UNLOCK TABLE después de la copia de cada tabla.

`--add-drop-table`

Añade un drop table antes de cada sentencia create

`-A, --all-databases`

Copia todas las bases de datos. Es lo mismo que utilizar `--databases` seleccionando todas.

`-a, --all`

Incluye todas las opciones de creación específicas de Mysql.

--allow-keywords

Permite la creación de nombres de columnas que son palabras clave, esto se realiza poniendo de prefijo a cada nombre de columna, el nombre de la tabla

-c, --complete-insert

Utiliza inserts incluyendo los nombres de columna en cada sentencia (incrementa bastante el tamaño del fichero)

-C, --compress

Comprime la información entre el cliente y el servidor, si ambos soportan compresión.

-B, --databases

Para copiar varias bases de datos. En este caso, no se especifican tablas. El nombre de los argumentos se refiere a los nombres de las bases de datos. Se incluirá USE db_name en la salida antes de cada base de datos.

--delayed

Inserta las filas con el comando INSERT DELAYED.

-e, --extended-insert

Utiliza la sintaxis de INSERT multilínea. (Proporciona sentencias de insert más compactas y rápidas.)

-#, --debug[=option_string]

Utilización de la traza del programa (para depuración).

--help

Muestra mensaje de ayuda y termina.

--fields-terminated-by=...

--fields-enclosed-by=... --fields-optionally-enclosed-by=... --fields-escaped-by=... --lines-terminated-by=...

Estas opciones se utilizan con la opción -T y tienen el mismo significado que la

correspondiente cláusula LOAD DATA INFILE.

-F, --flush-logs

Escribe en disco todos los logs antes de comenzar con la copia.

-f, --force,

Continúa aunque se produzca un error de SQL durante la copia.

-h, --host=..

Copia los datos del servidor de Mysql especificado. El servidor por defecto es localhost.

-l, --lock-tables.

Bloquea todas las tablas antes de comenzar con la copia. Las tablas se bloquean con READ LOCAL para permitir inserts concurrentes en caso de las tablas MyISAM.

Cuando se realiza la copia de múltiples bases de datos, --lock-tables bloquea la copia de cada base de datos por separado. De forma que esta opción no garantiza que las tablas serán consistentes lógicamente entre distintas bases de datos. Las tablas en diferentes bases de datos se copiarán en estados completamente distintos.

-K, --disable-keys

Se incluirá en la salida *#!/40000 ALTER TABLE tb_name DISABLE KEYS /;* y *#!/40000 ALTER TABLE tb_name ENABLE KEYS /;* Esto hará que la carga de datos en un servidor MySQL 4.0 se realice más rápido debido a que los índices se crearán después de que todos los datos hayan sido restaurados.

-n, --no-create-db

No se incluirá en la salida *CREATE DATABASE /*!32312 IF NOT EXISTS/ db_name;* Esta línea se incluye si la opción --databases o --all-databases fue seleccionada.

-t, --no-create-info

No incluirá la información de creación de la tabla (sentencia CREATE TABLE).

-d, --no-data

No incluirá ninguna información sobre los registros de la tabla. Esta opción sirve para crear una copia de sólo la estructura de la base de datos.

--opt

Lo mismo que --quick --add-drop-table --add-locks --extended-insert --lock-tables. Esta opción le debería permitir realizar la copia de seguridad de la base de datos de la forma más rápida y efectiva.

-pyour_pass, --password[=your_pass]

Contraseña utilizada cuando se conecta con el servidor. Si no se especifica, `=your_pass', mysqldump preguntará la contraseña.

-P, --port=...

Puerto utilizado para las conexiones TCP/IP

--protocol=(TCP | SOCKET | PIPE | MEMORY)

Especifica el protocolo de conexión que se utilizará.

-q, --quick

No almacena en el buffer la sentencia, la copia directamente a la salida. Utiliza mysql_use_result() para realizarlo.

-Q, --quote-names

Entrecomilla las tablas y nombres de columna con los caracteres '.

-r, --result-file=...

Redirecciona la salida al fichero especificado. Esta opción se debería utilizar en MSDOS, porque previene la conversión de nueva línea \n' en nueva línea y retorno de carro \n\r'.

--single-transaction

Utiliza el comando BEGIN antes de realizar la copia desde el servidor. Es muy útil con las tables InnoDB y el nivel de transacción READ_COMMITTED, porque en este modo realizará la copia de seguridad en un estado consistente sin necesidad de bloquear las aplicaciones. Consultar el manual para más detalles.

-S /path/to/socket, --socket=/path/to/socket

El fichero de sockets que se especifica al conectar al localhost (que es el host predeterminado).

--tables

sobreescribe la opción `--databases (-B)`.

`-T, --tab=path-to-some-directory`

Crea un fichero `table_name.sql`, que contiene la sentencia de creación de SQL, y un fichero `table_name.txt`, que contiene los datos de cada tabla. El formato del fichero `.txt` se realiza de acuerdo con las opciones `--fields-xxx` y `--lines--xxx` options. Nota: Esta opción sólo funciona si el comando `mysqldump` se ejecuta en la misma máquina que el demonio `mysqld`, el usuario deberá tener permisos para crear y escribir el fichero en la ubicación especificada

`-u nombre_usuario, --user=nombre_usuario`

El nombre de usuario que se utilizará cuando se conecte con el servidor, el valor predeterminado es el del usuario actual.

`-v, --verbose`

Va mostrando información sobre las acciones que se van realizando (más lento)

`-w, --where='cláusula where'`

Sirve para realizar la copia de determinados registros

`-X, --xml`

Realiza la copia de seguridad en un documento xml

`-x, --first-slave`

Bloquea todas las tablas de todas las bases de datos

Ejemplos de comandos `mysqldump`:

Para realizar la copia de seguridad de la base de datos `mibase` al fichero `copia_seguridad.sql`

```
mysqldump --opt mibase > copia_seguridad.sql
```

Otro ejemplo más complejo de comando `mysqldump` para hacer el backup de una base de datos es el siguiente:

```
mysqldump --opt --password=miclave --user=miuser mibasededatos > archivo.sql
```

En este último caso estamos indicando un nombre de usuario y una clave para acceder a la base de datos sobre la que se está haciendo el backup: `mibasededatos`. Las sentencias SQL para

reconstruir esa base de datos se volcarán en el fichero archivo.sql.

```
mysqldump --opt --password=clave --user=usuario Base_de_datos tabla1 tabla2 > backupdostablas.sql
```

Este comando es prácticamente igual que el anterior, pero en este caso nos realiza la copia de seguridad de únicamente dos tablas "tabla1" y "tabla2", que están en la base de datos "Base_de_datos".

Restaurar la base de datos

Si deseamos recuperar la información de un fichero para restaurar una copia de seguridad de la base de datos lo haremos con el comando mysql. Utilizaremos una sintaxis como esta:

```
mysql mibase < archivo.sql
```

En este ejemplo se restauraría la base de de datos mibase con el backup almacenado en el fichero archivo.sql.

Otro ejemplo más complejo de comando para restaurar una base de datos es el siguiente:

```
mysql --password=miclave --user=miuser mibase < archivo.sql
```

Es el mismo ejemplo que el anterior, pero indicando un nombre de usuario y una clave con las que acceder a la base de datos mibase.

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en 20/06/2003
Disponible online en <http://desarrolloweb.com/articulos/backup-mysql-mysqldump.html>

Subir una base de datos MySQL al servidor

En estos artículos vamos a mostrar, a modo de tutorial, algunos procedimientos posibles para subir una base de datos a un servidor. Este es un paso que tendrás que hacer al desplegar tu aplicación, puesto que generalmente trabajas en local en la etapa de desarrollo y necesitarás hacer una migración de la base de datos (estructura y/o datos) al servidor cuando quieras publicar tu proyecto.

Migrar una base de datos a MySQL

Indicaciones útiles para migrar una base de datos a MySQL, es decir, cuando tenemos que subir una base de datos local en cualquier gestor a una base de datos remota en MySQL.

El último caso en el que nos podemos encontrar a la hora de subir una base de datos a nuestro proveedor de alojamiento es que la base de datos la tengamos creada en local, pero en un sistema gestor distinto del que vamos a utilizar en remoto. En remoto suponemos siempre que vamos a utilizar la base de datos [MySQL](#). En local podríamos disponer de una base de datos [Access](#), [SQL Server](#) o de otro sistema de base de datos.

El proceso de la migración puede ser bastante complejo y, como hay tantas bases de datos distintas, difícil de dar una receta que funcione en todos los casos. Además, aparte de la dificultad de transferir la información entre los dos sistemas gestores de base de datos, también nos influirá mucho en la complejidad del problema el tipo de los datos de las tablas que estamos utilizando. Por ejemplo, las fechas, los campos numéricos con decimales o los booleanos pueden dar problemas al pasar de un sistema a otro porque pueden almacenarse de maneras distintas o, en el caso de los números, con una precisión distinta.

Recomendaciones para migrar de Access a MySQL

Si nuestra base de datos anterior estaba construida en Access lo tenemos bastante fácil, gracias a que [MySQL dispone de un driver ODBC para sistemas Windows](#), que nos permite [conectar Access con el propio MySQL](#) y pasar información fácilmente.

Este tema está relatado en el artículo [Exportar datos de MySQL a Access](#), aunque hay que indicar que si deseamos hacer una exportación desde Access en local a MySQL en remoto puede haber problemas porque no todos los alojadores permiten las conexiones en remoto con la base de datos. Si no tenemos disponible una conexión en remoto con nuestro servidor de bases de datos vamos a tener que cambiar la estrategia un poco.

La idea en este último caso es instalar MySQL en local y realizar la migración desde Access en local a MySQL en local y luego podríamos [hacer un backup de la base de datos local y subirla a remoto](#), tal y como se ha relatado antes.

Recomendaciones para migrar desde SQL Server a MySQL

La verdad es que no he tenido este caso nunca, pero hay que decir que Access también nos puede ayudar en este caso. Access permite seleccionar una base de datos SQL Server y trabajar desde la propia interfaz de Access. La idea es que Access también permite trabajar con MySQL y posiblemente haciendo un puente entre estos dos sistemas gestores podemos exportar datos de SQL Server a MySQL.

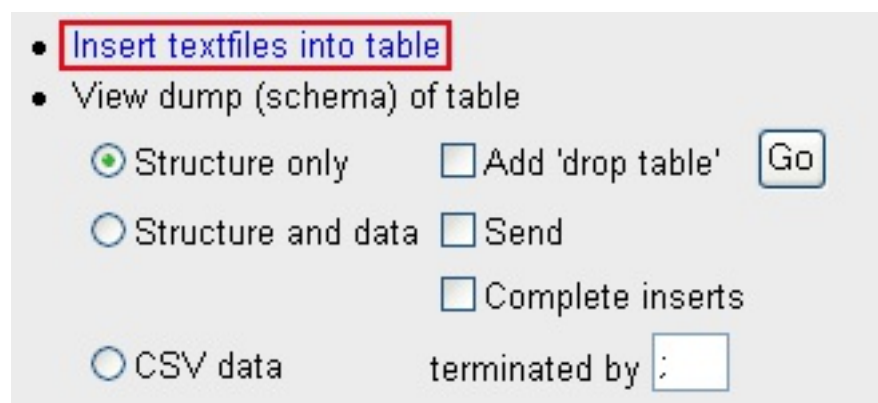
Lo que es seguro que utilizando el propio Access de puente podríamos realizar el trabajo. Primero exportando de SQL Server a Access y luego desde Access a MySQL.

Otras bases de datos u otras técnicas

Si la base de datos origen dispone de un driver ODBC no habrá (en teoría) problema para conectarla con Access, de manera similar a como se conecta con MySQL. Entonces podríamos utilizar Access para exportar los datos, porque desde allí se podrían acceder a los dos sistemas gestores de bases de datos.

Si no tenemos Access, o la base de datos original no tiene driver ODBC, o bien no nos funciona correctamente el proceso y no sabemos cómo arreglarlo, otra posibilidad es exportar los datos a ficheros de texto, separados por comas o algo parecido. Muchas bases de datos tienen herramientas para exportar los datos de las tablas a ficheros de texto, los cuales se pueden luego introducir en nuestro sistema gestor destino (MySQL) con la ayuda de alguna herramienta como PhpMyAdmin.

Para ello, en la página de propiedades de la tabla encontraremos una opción para hacer el backup de la tabla y para introducir ficheros de texto dentro de una tabla (Insert textfiles into table en inglés).



The image shows a screenshot of a web-based interface for managing a database table. A red box highlights the option 'Insert textfiles into table'. Below this, there are several options and checkboxes:

- ☒ Structure only
- ☐ Structure and data
- ☐ CSV data
- ☐ Add 'drop table'
- ☐ Send
- ☐ Complete inserts
- terminated by
-

Accediendo a ese enlace podremos ver un formulario donde introducir las características del fichero de texto, como el carácter utilizado como separador de campos, o el terminador de líneas, etc, junto con el propio archivo con los datos, y PhpMyAdmin se encargará de todo el trabajo de incluir esos datos en la tabla.

Location of the textfile	<input type="text"/>	<input type="button" value="Examinar..."/>
Replace table data with file	<input type="checkbox"/> Replace	The contents of the file replaces the contents of the selected table for rows with identical primary or unique key.
Fields terminated by	<input type="text" value=";"/>	The terminator of the fields.
Fields enclosed by	<input type="text" value=""/> <input type="checkbox"/> OPTIONALLY	Often quotation marks. OPTIONALLY means that only char and varchar fields are enclosed by the "enclosed by"-character.
Fields escaped by	<input type="text" value="\"/>	Optional. Controls how to write or read special characters.
Lines terminated by	<input type="text" value="\n"/>	Carriage return: \r Linefeed: \n
Column names	<input type="text"/>	If you wish to load only some of a table's columns, specify a comma separated field list.
[Documentation]		
<input type="button" value="Submit"/> <input type="button" value="Reset"/>		

Como se habrá supuesto, es necesario tener creada la tabla en remoto para que podamos introducirle los datos del fichero de texto.

Cambios de un formato de datos a otro

Toda la migración tiene que tener en cuenta muy especialmente, como ya se señaló, las maneras que tenga cada base de datos de guardar la información, es decir, del formato de sus tipos de datos. Tenemos que contar siempre con la posible necesidad de transformar algunos datos como pueden ser los campos booleanos, fechas, campos memo (texto con longitud indeterminada), etc, que pueden almacenarse de maneras distintas en cada uno de los sistemas gestores, origen y destino.

En algunos casos posiblemente tengamos que realizar algún script que realice los cambios necesarios en los datos. Por ejemplo puede ser para localizar los valores booleanos guardados como true / false a valores enteros 0 / 1, que es como se guarda en MySQL. También las fechas pueden sufrir cambios de formato, mientras que en Access aparecen en castellano (dd/mm/aaaa) en MySQL aparecen en el formato aaaa-mm-dd. PHP puede ayudarnos en la tarea de hacer este script, también Visual Basic Script para Access puede hacer estas tareas complejas y el propio lenguaje SQL, a base de sentencias dirigidas contra la base de datos, puede servir para algunas acciones sencillas.

Este artículo es obra de *Miguel Angel Alvarez*

Fue publicado por primera vez en 23/07/2003
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Subir una base de datos al servidor de Internet

El segundo paso para subir una aplicación PHP al servidor consiste en colocar la base de datos en el

Aparte de los archivos de la página, debemos subir la base de datos con la que tenemos que trabajar. Las bases de datos con las que trabaja PHP son muy variadas y en distintos casos podemos utilizar una u otra, por lo que los modos de subir la base de datos también pueden variar.

Nota: Este artículo y los sucesivos, que tratan sobre subir una base de datos MySQL al servidor, se engloban tanto dentro del [Manual de PHP](#) como del [Taller de MySQL](#). Por ello, será importante disponer de conocimientos de ambas tecnologías para entender y aprovechar estas explicaciones.

Es muy corriente que nuestro proveedor de hosting ofrezca junto con PHP la base de datos MySQL, así que las notas para subir esa base de datos al servidor de este artículo van encaminadas a ofrecer soluciones para esa base de datos.

La base de datos MySQL no se puede subir por FTP, como que se hacía con los archivos del código PHP. Para subirla tendremos que utilizar otros mecanismos. Voy a distinguir entre tres casos distintos en los que nos podríamos encontrar en este momento:

1. **La base de datos que pretendemos subir está vacía.** Tan sólo hemos creado las tablas, pero no hemos introducido datos en ellas o, a lo sumo, tienen algún dato que hemos introducido de pruebas.
2. **La base de datos que queremos subir está completa y es una base de datos MySQL.** En este caso tenemos creada la base de datos en local y con toda la información dentro y, por supuesto, queremos que esa información quede también en la base de datos remota.
3. La base de datos está **completa** (como el caso anterior), pero **no es una base de datos MySQL**. En este caso estaríamos haciendo una migración de la base de datos de un sistema gestor a otro.

Veremos los tres casos por separado en adelante, aunque, antes de ello, vamos a mostrar unas herramientas que nos servirán de mucha ayuda para la administración de cualquier base de datos remota.

Las herramientas en concreto se relatan en el manual [Taller de MySQL](#), son las siguientes:

- [PhpMyAdmin](#). Una aplicación creada en PHP que podemos instalar en nuestro espacio de alojamiento para administrar la base de datos.

- **MySql Control Center** (en adelante MyCC). Una aplicación Windows que permite conectarse a múltiples bases de datos MySQL, que se encuentren en local o en remoto.
- **Access**. También permite administrar una base de datos MySQL conectada en local o en remoto. En este caso se utiliza una interfaz que muchos ya conocen, como es Access, para administrar una base de datos que nada tiene que ver con dicho programa.

En los tres casos lo que nos permite realizar el software de administración son tareas sobre la base de datos de todo tipo, como pueden ser crear tablas, modificarlas, insertar datos, borrarlos, editarlos. Modificar o borrar tablas o campos de las mismas, etc.

La elección de una herramienta o de otra pasa por los recursos que nos permitan utilizar en nuestro proveedor. Básicamente, lo que nos puede decantar a una opción u otra, es si permiten o no conectar de manera remota la base de datos MySQL. Conozco alojamientos donde se permite esa conexión remota y donde no.

Si no permiten conectarnos remotamente nos decantaremos por PhpMyAdmin, pues es una aplicación PHP que se conecta en local y a la que se accede desde una página web y eso lo permiten todos los proveedores, incluso hay muchos que tienen instalado ya este software para administrar las bases de datos.

En caso de que sí nos permitan conectarnos remotamente con la base de datos, eligiremos MyCC o Access, que son aplicaciones Windows mucho más potentes y rápidas que las que utilizan interfaz web, como PhpMyAdmin. Es preferible utilizar MyCC porque está especialmente desarrollado para conectar y operar con bases de datos MySQL.

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 26/06/2003
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Subir base de datos MySQL vacía al servidor

La base de datos que pretendemos subir está vacía. Tan sólo hemos creado las tablas, pero no hemos introducido datos en ellas o, a lo sumo, tienen algún dato que hemos introducido de pruebas.

Es muy normal que hayamos diseñado una base de datos para nuestro proyecto desde o, definiendo las distintas entidades de nuestro modelo de datos, junto con sus campos y sus tipos.

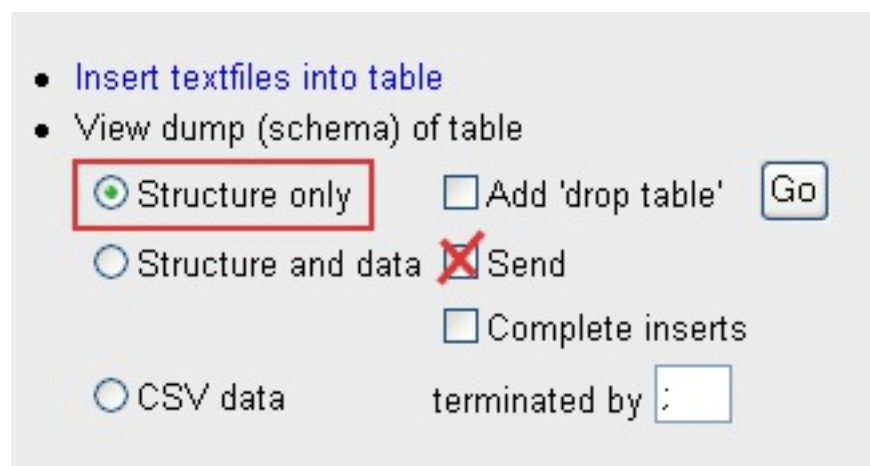
En estos casos lo más probable es que la base de datos esté vacía, o bien contenga datos que hayamos introducido a modo de prueba y que no queramos conservar cuando subamos la aplicación a Internet.

La opción más interesante entonces podría ser crear otra vez las tablas que tenemos en local en la base de datos remota. Para ello tenemos dos posibilidades:

Si tenemos pocas tablas y bastante sencillas: Las podemos crear en remoto con alguna herramienta como [PhpMyAdmin](#) o [MyCC](#).

Si tiene muchas tablas y/o muy complicadas: La recomendación sería hacer un backup de la estructura en local y restaurarla en remoto. Esto nos evitará tener que volver a crear todas las tablas y definir todos sus campos y sus tipos. Puede ser un poco más complicado pero sin duda nos ahorrará tiempo.

Para hacer el backup de la estructura en local podemos utilizar alguna herramienta como [PhpMyAdmin](#), o bien utilizar el comando [mysqldump](#) desde línea de comandos de MS-DOS.



- [Insert textfiles into table](#)
- View dump (schema) of table

☒ Structure only ☐ Add 'drop table'

☐ Structure and data ☒ Send

☐ CSV data ☐ Complete inserts

terminated by

Herramienta de backup de PhpMyAdmin. Está marcada la opción de extraer solamente la estructura de las tablas. Si marcamos además la casilla "Send", nuestro navegador se descargará el backup en un fichero de texto. Si no lo pulsamos simplemente se visualizará.

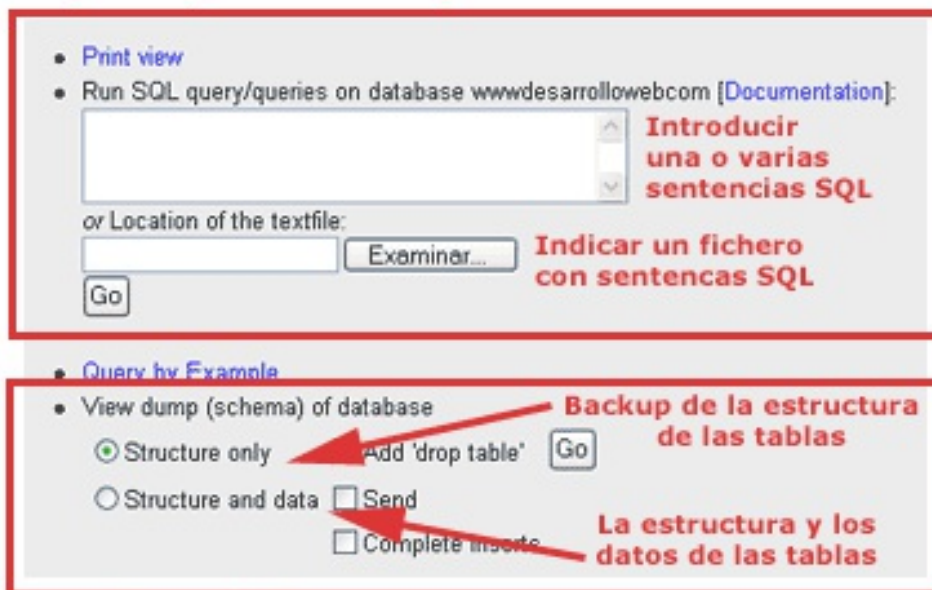
Lo que tenemos que hacer en este caso es un backup de la estructura de la base de datos, es decir, los "create tables" o sentencias SQL para crear las tablas. Sería un montón de sentencias con esta forma:

```
# -----  
#  
# Table structure for table 'comentario'  
#  
  
CREATE TABLE comentario (  
  id_comentario int(5) unsigned NOT NULL auto_increment,  
  id_articulo int(4) DEFAULT '0' NOT NULL,  
  comentario text NOT NULL,  
  fecha int(14) unsigned DEFAULT '0' NOT NULL,  
  revisado tinyint(1) DEFAULT '0' NOT NULL,  
  nombre_comentario varchar(100) DEFAULT 'Nombre no especificado' NOT NULL,  
  email_comentario varchar(100) DEFAULT 'Email sin especificar' NOT NULL,  
  tipo tinyint(1) unsigned DEFAULT '1' NOT NULL,  
  PRIMARY KEY (id_comentario)  
);
```

Para restaurar estas sentencias tenemos opciones tanto dentro de PhpMyAdmin como de MyCC. En ambos casos lo que tenemos que hacer es ejecutar estas sentencias en el servidor MySQL remoto. En PhpMyAdmin tenemos un campo para introducir sentencias SQL y también otro campo para seleccionar un archivo de texto con todas las sentencias SQL, para ejecutarlas una detrás de otra. En MyCC tenemos un botón que nos permite abrir una consola donde introducir una o varias sentencias SQL y ejecutarlas.

Herramienta de backup y restauración de PhpMyAdmin

Para restaurar la tabla desde el backup compuesto por sentencias SQL



• Print view

• Run SQL query/queries on database wwwdesarrolloweb.com [Documentation]:

Introducir una o varias sentencias SQL

or Location of the textfile:

Examinar...

Indicar un fichero con sentencias SQL

Go

• Query by Example

• View dump (schema) of database

Structure only

Add 'drop table'

Go

Structure and data

Send

Complete inserts

Backup de la estructura de las tablas

La estructura y los datos de las tablas

Para obtener el backup de la base de datos

Botón para introducir sentencias SQL en MyCC



Repetimos, esto sólo nos servirá para subir la estructura de la base de datos y no los datos que contenga. Si deseamos subir también la información de la base de datos entonces debemos utilizar otras estrategias, relatadas próximamente.

Este artículo es obra de *Miguel Angel Alvarez*
 Fue publicado por primera vez en 26/06/2003
 Disponible online en <http://desarrolloweb.com/articulos/34.php>

Subir una base de datos MySQL con la estructura y los datos

Como transferir una base de datos MySQL que tenemos en local a nuestro servidor remoto, incluyendo tanto la estructura de las tablas como sus datos.

Si la base de datos que deseamos subir está llena de información y deseamos que se conserve una vez subida la base de datos a remoto, tenemos que realizar un backup de la base de datos y restaurarlo en remoto.

Nota: Estas recomendaciones están pensadas para subir una base de datos MySQL que podamos tener en local a una base de datos MySQL que hayamos contratado en remoto. Si la base origen no es MySQL estaríamos hablando de una migración de bases de datos, pero esto lo veremos en un [artículo más adelante](#).

En este caso el procedimiento sería muy parecido al de [subir una base de datos vacía](#), relatado anteriormente, con la salvedad de que ahora debemos extraer no solo la estructura de la base de datos, sino también los registros que contiene.

Para ello podemos utilizar mysqldump, según se relata en [este artículo](#), o bien [PhpMyAdmin](#), seleccionando la opción que indica que el backup contenga la estructura y los datos (Structure and data en versiones en inglés).

La estructura y los datos vendrán en un fichero de texto con una serie de sentencias SQL para crear las tablas y los insert necesarios para introducir cada uno de los datos.

Para restaurar la base de datos lo haremos tal como se ha relatado para el caso de que la base de datos estuviera vacía, con la ayuda de una instalación de PhpMyAdmin en remoto o un MyCC que se conecte a la base de datos contratada en el servidor de Internet.

Si tenemos problemas para subir el fichero de backup de la base de datos es posible que en nuestro proveedor de alojamiento nos pueda ayudar a subir el fichero y restaurarlo. Como el proveedor dispone de los servidores en sus propias instalaciones, tiene muchas más posibilidades que nosotros para trabajar con las bases de datos, sin temor a que las lentas comunicaciones por Internet arrojen errores en la restauración de los datos.

Si nuestro proveedor no puede ayudarnos, seguramente disponga y nos indique algún mecanismo para realizar la tarea sin lugar a errores. Puede ocurrirnos con algún proveedor que nos diga que se encarga de todo pero nos exija el pago de las horas de trabajo del informático que va a restaurar el backup de la base de datos. Si no pone facilidades ni siquiera en esto posiblemente sea mejor ir pidiéndoles que nos devuelvan el dinero invertido porque su servicio no sería muy bueno.

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 23/07/2003
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Configurar el acceso remoto a MYSQL de manera segura

Tutorial para explicar cómo configurar tu servidor y el cliente, para poder

acceder a MYSQL vía remota de manera segura. te explicaremos cómo configurar el servido sin tener el puerto abierto y sin que nadie más pueda entrar en nuestra base de datos.

En el momento de escribir el artículo [Gestor de MySQL HeidiSQL](#) me encontraba con varios años como *webmaster* en *hostings* compartidos (como muchos lectores). Sin embargo, al crecer en visitas y necesidades y estar tan limitado con las características propias de los alojamientos compartidos, migré a VPS y luego a servidores dedicados.

Al final, la administración de servidores dedicados traslada las preocupaciones propias de los proveedores de hosting a tu cabeza, y con el tema más delicado “La seguridad” (ejemplo de ello: en los días en que se escribía este artículo, se descubrió la vulnerabilidad **heartbleed**, que hizo sudar a todos los administradores, que tuvieron que actualizar rápidamente sistema y generar nuevos certificados). Claro, existen servicios manejados que aumentan mucho más el costo del hosting en los cuales estás pagando por la administración y mantenimiento con un costo promedio de 100\$ adicionales al servicio, lo que no siempre es una opción.

Este artículo te ayudará si te encuentras en un caso similar, tienes que administrar tu propio servidor y necesitas dormir tranquilo, sabiendo que la configuración mantiene una elevada seguridad y que tu base de datos está bien protegida de accesos no autorizados. Si es tu caso, puedes seguir leyendo. Si no es así, te recomendamos avanzar con otros asuntos interesantes sobre este sistema en el [Manual de MySQL](#).



Bien: Queremos acceder a MySQL vía remota, pero no queremos tener el puerto abierto y queremos que solo nosotros podamos acceder.

Este artículo se basa en que tienes el siguiente escenario:

1. En tu servidor está corriendo una versión de Linux
2. Usas como cortafuegos “Configure Config Server Firewall” el famoso CSF
3. Tienes MySQL corriendo y el puerto 3306 se encuentra cerrado (como debe)

En el artículo [Gestor de MySQL HeidiSQL](#) tienes el escenario de que, si bien usas el panel para autorizar los IPs que se conectarán a MySQL, tienes la dificultad del propio “meollo móvil” donde te conectarás en diferentes sitios a redes WiFi. ...O tu proveedor cambia a cada rato la dirección IP de tu conexión. Así que no resulta práctico estar chequeando a cada rato la dirección IP.

Lo primero es visitar www.no-ip.org y abrirnos una cuenta gratuita (No-IP permite identificar

tu PC con un nombre de dominio fácil de recordar).

Create Your No-IP Account

Username: Password: Confirm Password:

Hostnames:

☐ Create my hostname later

That address is also available with these [Enhanced DNS](#) domains for only \$19.95 a year:

- [elnombrequeeligas.ddns.me](#) [elnombrequeeligas.noip.us](#)
- [elnombrequeeligas.ddns.net](#) [elnombrequeeligas.hopto.me](#)
- [elnombrequeeligas.no-ip.ca](#) [elnombrequeeligas.dnsfor.me](#)

Upgrading to Enhanced DNS now will save you time and money later.

En el formulario de inscripción de una vez podemos escoger el nombre de dominio que queramos gratis.

También podemos ingresar luego en la opción Hosts/Redirects >> Manage Hosts y crear el dominio que queramos.

Add a host

Fill out the following fields to configure your host. After you are done click 'Create Host' to add your host.

Own a domain name?
Use your own domain name with our DNS system. [Add](#) or [Register](#) your domain name now or read more for pricing and features.

Hostname Information

Hostname:

Host Type: ☒ DNS Host (A) ☐ DNS Host (Round Robin) ☐ DNS Alias (CNAME)
☐ Port 80 Redirect ☐ Web Redirect ☐ AAAA (IPv6)

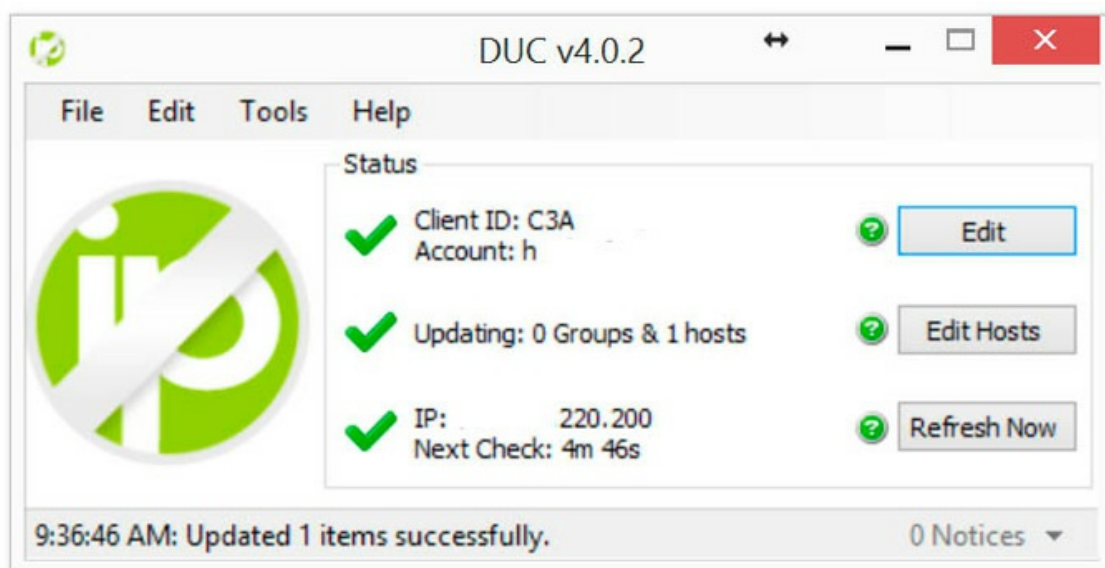
IP Address:

Assign to Group: [Configure Groups](#)

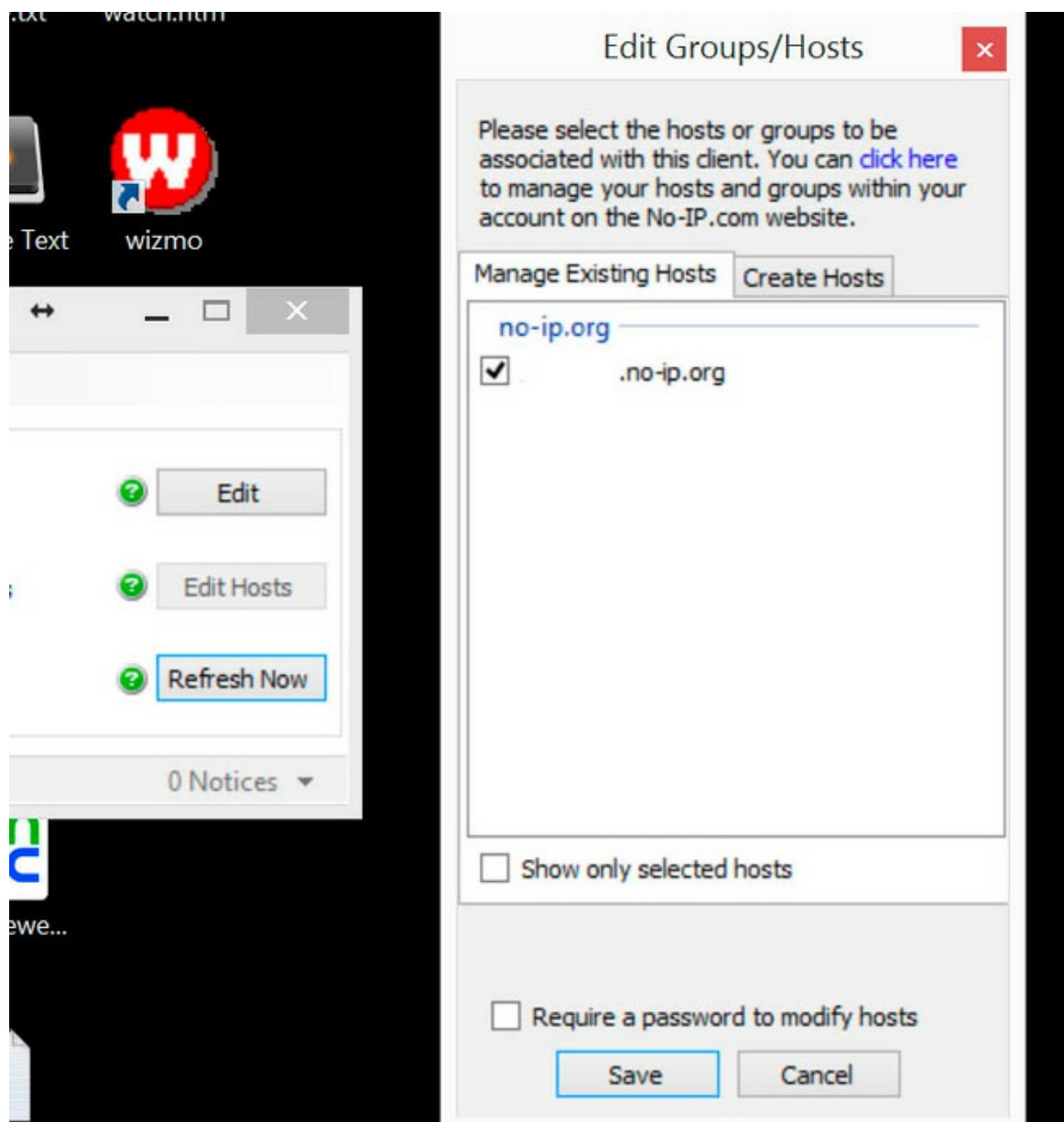
Enable Wildcard: ☐ Wildcards are a Plus / Enhanced feature. [Upgrade Now!](#)

Todo esto es para una sola cosa: asociar nuestra dirección IP dinámica al dominio creado.

Podemos hacerlo descargando el software DUC de la misma página.



Después de ingresar nuestros *login* y contraseña, debemos activar nuestro *host* en “EDIT HOST”



Si usamos un *router*, opcionalmente podemos usarlo también ingresando directamente nuestras credenciales y dominio en la opción Dynamic DNS.

En este ejemplo un TP-LINK en la opción Dynamic DNS.

TP-LINK

DDNS

Service Provider: No-IP (www.no-ip.com) [Go to register...](#)

User Name: [Redacted]

Password: [Redacted]

Domain Name: cina.sytes.net

☒ Enable DDNS

Connection Status: Succeeded!

Login Logout

Save

En los dos casos un PING a nuestro dominio debe devolver una respuesta correcta (consola de sistema o CMD si usas Windows).

```
C:\Windows\system32>ping cina.sytes.net

Haciendo ping a cina.sytes.net [ 220.200 ] con 32 bytes de datos :
Respuesta desde bytes=32 tiempo=1ms TTL=64
Respuesta desde 220.200: bytes=32 tiempo=1ms TTL=64
Respuesta desde 220.200: bytes=32 tiempo=1ms TTL=64
Respuesta desde 220.200: bytes=32 tiempo=4ms TTL=64

Estadísticas de ping para 220.200:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 4ms, Media = 1ms

C:\Windows\system32>
```

Cuando todo está funcionando es hora de hacer que nuestro servidor nos permita ingresar a MySQL desde nuestro computador con MySQL HeidiSQL u otro gestor de conexiones que usemos; incluso podemos usar nuestro MySQL local para conectarnos en el de nuestro servidor.

```
C:\AppServ\MySQL\bin>mysql -h DOMINIO DE NO-IP -u USUARIO DE MYSQL -p
Enter password: xxxxxxxxxxxx
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 62715
Server version: 5.1.73 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

La opción `-h` seguida de nuestro dominio no permite conectarnos a servidores externos (muy útil para hacer *backup* de nuestras bases de datos o subir grandes cantidades de datos).

¡Atención!, antes de intentar ingresar, vamos a configurar nuestro servidor para ello:

La configuración sería la siguiente:

Ingresamos a nuestra consola de servidor, editamos el archivo de configuración `"/etc/csf/csf.conf"` en la línea `DYNDNS` y cambiamos o escribimos `DYNDNS = "600" ---//` actualizará nuestra dirección IP cada 10 minutos. En mi caso particular lo tengo en `300 ---` (cinco minutos).

Editamos el archivo `"/etc/csf/csf.dyndns"` y agregamos nuestro dominio de esta manera: `tcp:in:d=3306:s=dominiocreadoen-no-ip` Como se puede ver en la parte `tcp:in:d=3306: 3306` es el número de puerto de conexión MySQL.

Cuando estemos listos, reiniciamos el *firewall* con el comando `CSF -r` Esperamos los minutos indicados para que CSF actualice tu dirección y listo.

Nota: Si administras tus vps o servidor via SSH Puedes usar esto mismo cambiando o agregando `"tcp:in:d=22:s= dominiocreadoen-no-ip "`, siendo el puerto 22 el puerto de conexión SSH o el que hayas seleccionado para conectarte.

Es muy importante que abras una ventana/conexión adicional de SSH, pruebes esto y si todo sale mal en la otra ventana conectada puedas recuperar la configuración, (esto en los casos donde tu única administración sea por SSH).

Con esto solo tu IP tendrá acceso a tu servidor vía SSH -----

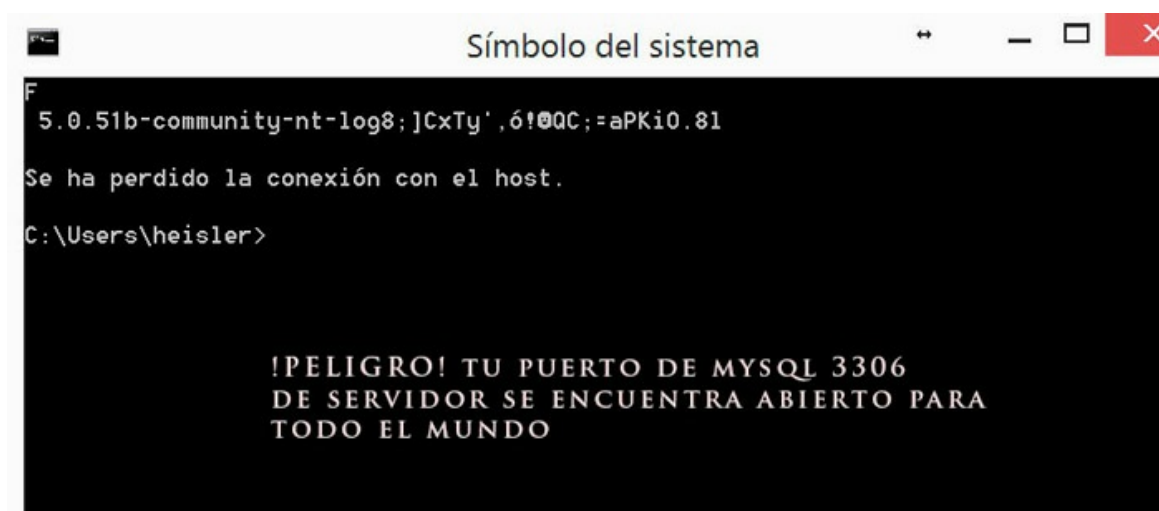
Tu servidor solo aceptará conexiones desde tu computador hacia MySQL.

Tendrás el puerto 3306 cerrado para evitar una conexión no autorizada y podrás conectar hacia MySQL remotamente.

- Inicialmente esto se hacía con dyndns.com pero ya no ofrecen una versión gratuita y en el caso de www.noip.com Vitalwerks Internet Solutions, ofrece este servicio gratis desde 1999 y esperamos que continúe así por muchos años más.
- Probar si nuestro puerto 3306 se encuentra abierto desde Windows hacia nuestro

servidor (abierto para todo el mundo significa que tu servidor es vulnerable).

Desde la consola de Sistema CMD debes ejecutar el comando "telnet nuestraIPdeservidor 3306" o bien "telnet nuestrodominio 3306"



```
Símbolo del sistema

F
5.0.51b-community-nt-log8;]CxTy',6!@QC;=aPKi0.81

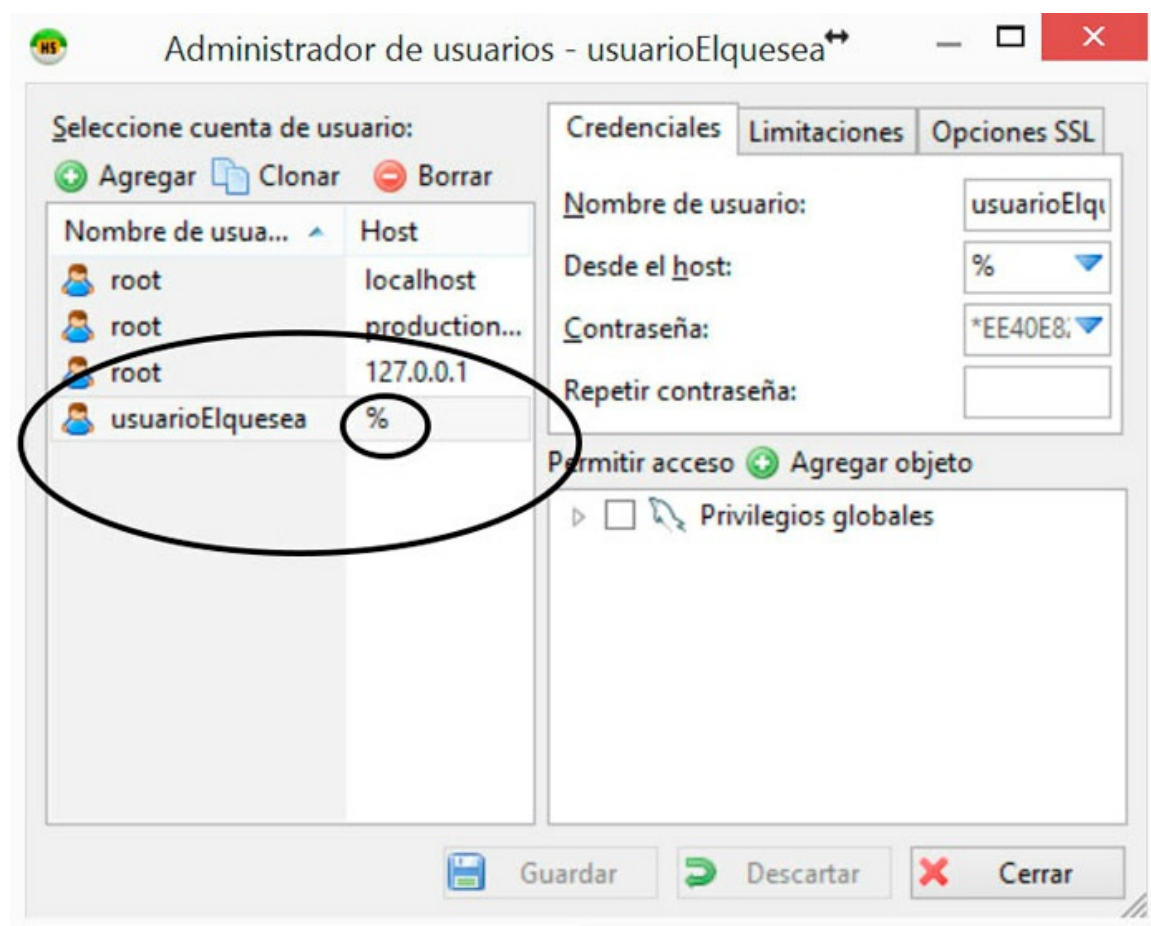
Se ha perdido la conexión con el host.

C:\Users\heisler>

!PELIGRO! TU PUERTO DE MYSQL 3306
DE SERVIDOR SE ENCUENTRA ABIERTO PARA
TODO EL MUNDO
```

Si recibes este tipo de respuesta tienes el puerto 3306 y debes configurar apropiadamente tu CSF para impedir el acceso por dicho puerto.

- El usuario que uses para acceder remotamente a MySQL debe tener los permisos para ello.
- Recomendamos no usar ROOT para acceder remotamente.
- Crear nuevo usuario con los permisos correspondientes y la opción % en vez de localhost o 127.0.0.1



Este artículo fue probado y escrito escuchando el último concierto de Soda Stereo 2007.

Este artículo es obra de *Heisler Palma*
 Fue publicado por primera vez en 30/05/2014
 Disponible online en <http://desarrolloweb.com/articulos/acceso-remoto-mysql.html>

Sistema de usuarios de MySQL

En esta parte del tutorial de MySQL te vamos a explicar cómo administrar los usuarios de MySQL y sus permisos. El sistema de usuarios de MySQL permite asignar los privilegios de acceso a los datos y su modificación por parte de los usuarios. Muchas veces cuando te entregan una base de datos te indican el usuario y contraseña con la que te tienes que conectar, pero hay ocasiones en las que tú mismo tienes que crear estos usuarios y asignarles los permisos adecuados para realizar las operaciones pertinentes.

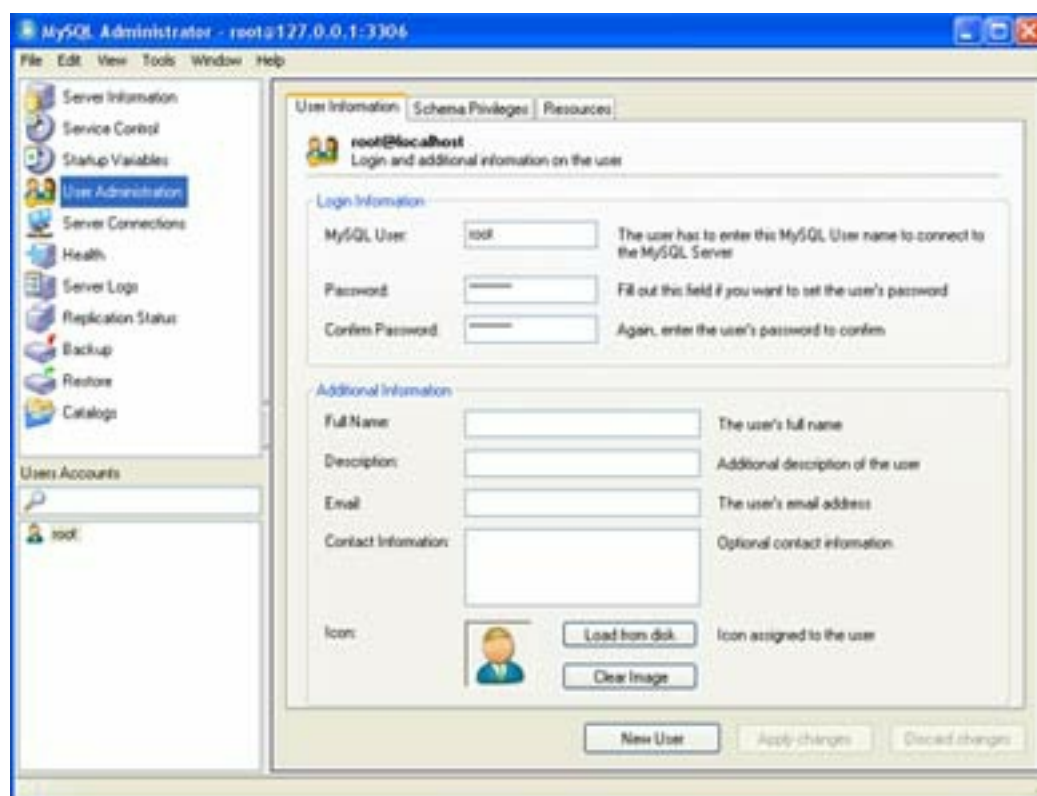
Gestionar usuarios MySQL

Trabajamos con la herramienta MySQL Administrator para dar de alta y gestionar usuarios de la base de datos, asignar privilegios, posibles hosts de acceso, etc.

Los usuarios de MySQL se gestionan a través de la base de datos llamada "mysql", concretamente con la tabla "user". Sin embargo, el manejo de esa tabla es relativamente complicado para un usuario que no sea del todo experto, por lo que será mucho más cómodo usar la aplicación MySQL Administrator para realizar la gestión de usuarios.

MySQL Administrator es una herramienta que provee directamente MySQL, que se puede descargar desde la propia página de MySQL y que ya introdujimos anteriormente en nuestro [Taller de MySQL: Introducción a MySQL Administrator](#).

Entre las distintas opciones de MySQL Administrator tenemos una sección para la gestión de usuarios, sus permisos, host de acceso, etc. La podemos ver y seleccionar en la barra de opciones de la izquierda.



Inicialmente sólo hay un usuario creado, en muchas de las configuraciones típicas, que es el usuario root. La lista de usuarios creados aparece en la parte de abajo de la barra de la izquierda. Podemos seleccionar cualquier usuario para editar sus propiedades.

Las propiedades de usuarios se dividen en tres pestañas:

User information:

Con los datos de login (nombre de usuario y contraseña de acceso) y otros datos personales del usuario, que es opcional completar.

Schema Privileges:

Con los permisos asociados a ese usuario. Luego los comentaremos.

Resources:

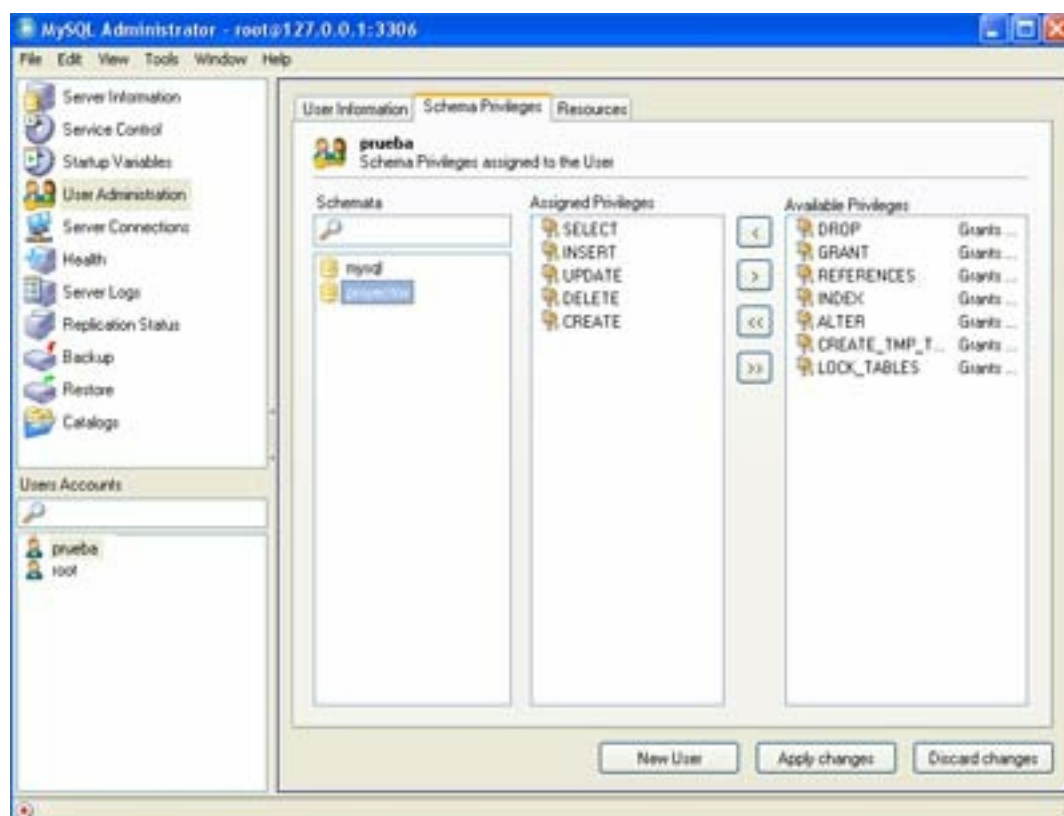
Con los recursos disponibles para ese usuario.

En la parte de debajo de la ventana encontraremos tres botones, para crear un nuevo usuario, para aplicar los cambios realizados en los formularios de edición de un usuario y para descartarlos.

Si queremos crear un usuario podemos pulsar el botón de "New User" y aparecerá la ventana de propiedades del usuario, en la pestaña de User Information, para que introduzcamos el nombre para este usuario y la contraseña que vamos a asignar.

Podemos entrar en otras pestañas para terminar de introducir los datos del usuario. La pestaña de Schema Privileges es especialmente interesante, ya que nos permite seleccionar los permisos de este usuario para cada base de datos de nuestro sistema.

La ventana de Schema Privileges nos permite seleccionar una base de datos y entonces accedemos a una lista con todos los privilegios posibles para permitir o denegar. Inicialmente, para un usuario nuevo, todos los permisos están denegados, así que tendremos que seleccionar los que deseamos otorgar. Para ello simplemente seleccionamos entre los "available privileges" (permisos disponibles) y los pasamos a "assigned privileges" (permisos asignados).



Nota: Este artículo continúa en [Definir los host de conexión de los usuarios a MySQL](#).

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 15/06/2006
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Definir los host de conexión de los usuarios a MySQL

Con MySQL Administrator se pueden definir distintos ordenadores o host desde los que un usuario se puede conectar a una base de datos MySQL.

Vamos a continuar la explicación de la gestión de usuarios en bases de datos MySQL que ya comenzamos en el artículo [Gestionar usuarios MySQL](#).

Anteriormente los usuarios y los distintos host se definían en MySQL con un formato usuario@host. Por ejemplo pepe@localhost quería decir que el usuario pepe se puede conectar solamente desde el propio ordenador donde está el servidor de la base de datos. Por ejemplo, pepe@% significaba que el usuario pepe se puede conectar desde cualquier host. Los permisos asignados a pepe@localhost y pepe@% pueden ser distintos.

Ahora el manejo de host de cada usuario se hace de manera distinta.

Inicialmente cualquier usuario creado se puede conectar desde cualquier host o computador posible. Si esta situación no nos conviene, podemos definir nuevos host pulsando con el botón derecho del ratón sobre el usuario que queremos añadir ordenadores de acceso. Entonces nos sale un formulario para indicar el nombre del host que queremos añadir a ese usuario, o su dirección IP.

En cualquier momento, si hacemos doble clic sobre un usuario y nos sale debajo una lista de hosts, es que ese usuario se puede conectar desde cualquier de esos host y además, podremos asignar permisos distintos al usuario cuando se conecte desde cada ordenador posible. De modo que un usuario si se conecta, por ejemplo desde localhost, puede tener más permisos que si se conecta desde otros ordenadores de la red local o de Internet.



Por ejemplo, en la imagen anterior se entiende que el usuario prueba se puede conectar desde localhost (el ordenador local) y desde otro equipo llamado "otrohost". Pero tener cuidado con esta lista de host, porque a lo mejor todavía estamos permitiendo que el usuario prueba se pueda conectar desde cualquier host y es posible que no deseemos que eso se permita, sino que sólo se deje acceder si se accede desde esos host especificados.

Para darnos cuenta de este detalle podemos ver en la ventana de propiedades del usuario el nombre del usuario que sale, para saber los host que están permitidos.

Lo vemos con una imagen:



En esta imagen nos hemos situado en el host localhost del usuario prueba. Vemos que en la parte de arriba, en la parte de propiedades del usuario aparece prueba@localhost. Eso quiere decir que está definido el usuario prueba siempre que se conecte desde localhost y que estamos en la ventana de propiedades de ese usuario.

Ahora veamos la siguiente imagen:



Si queremos evitar que el usuario prueba se pueda acceder desde cualquier equipo, simplemente pulsamos con el botón derecho en el nombre de usuario y seleccionamos la opción que pone "remove host from which the user can connect". Entonces, si hemos pulsado sobre el nombre del usuario nos quitará el acceso del usuario desde cualquier host. Si hacemos la misma acción sobre cualquier nombre de host, como localhost, nos quitará la posibilidad de que el usuario se pueda conectar desde ese ordenador.

Conclusión

Si queremos, siempre podemos echar un vistazo a la tabla user de la base de datos mysql, que es donde se guardan los usuarios de la base de datos y así podemos ver de qué manera se han implementado internamente los cambios que hemos hecho con MySQL Administrator.

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 22/06/2006
Disponible online en <http://desarrolloweb.com/articulos/definir-host-conexion-usuarios-mysql.html>

Programas de interfaz gráfica para MySQL

Esta serie de programas te servirán para administrar MySQL de una manera cómoda, con programas que se gestionan mediante una interfaz de usuario sencilla de manejar. Los programas de interfaz gráfica te sirven para ahorrarte el acceso a MySQL por línea de comandos, aunque nunca debes pensar en ellos como una sustitución del terminal, sino como una ayuda puntual para facilitar tu día a día e incrementar tu productividad en el trabajo con MySQL.

Gestor de MySQL HeidiSQL

Cómo utilizar HeidiSQL para administrar MySQL en servidores remotos y cómo configurar la IP para que se pueda conectar con el servidor mediante el panel de control cPanel.

Estoy seguro de que no seré el único del mundo que trabaja con MySQL y desea poder hacer las cosas más rápidas y sencillas. Si tu servidor está administrado por un panel de control como cPanel, quizás te habrás dormido literalmente, como yo, pulsando *links* y esperando una y otra vez hasta llegar a tu administrador PhpMyAdmin.

¡¡Todo esto puede ser mucho más corto!! y para ello te vamos a ayudar ofreciéndote una herramienta llamada HeidiSQL, que está muy bien para poder llevar la administración de una base de datos MySQL que está en un servidor remoto.

HeidiSQL es un ligero programa para Windows que nos ofrece una interfaz amigable para administrar MySQL, pero también sistemas gestores SQL Server de Microsoft. Permite navegar por las bases de datos y las tablas, editando cualquier información, creando registros, modificando tablas, vistas, procedimientos, *triggers* y en general todo aquello que necesitaremos en el día a día de la administración de bases de datos.

En este artículo te contaremos algo más sobre HeidiSQL, pero también te ayudaremos a configurar el acceso remoto a tu servidor, validando la IP desde la que te conectas para que tu MySQL te permita el acceso desde otras redes, sin por ello perder en seguridad.

Pero antes de ponernos a ello quiero explicarte por qué es una buena idea utilizar algún sistema como HeidiSQL para administrar rápidamente una base de datos MySQL.

El tortuoso camino hacia PhpMyAdmin

Yo tengo el servidor administrado con el panel de control cPanel, pero me consta que este mismo problema que voy a relatar es muy similar con otros paneles de control como Plesk.

Estoy saturado de hacer siempre las mismas acciones para poder administrar la base de datos MySQL y por eso quiero compartir contigo unos atajos para que puedas acceder más fácilmente a la administración de tu base de datos y además de manera segura!

La historia se repite, una y otra vez, al acceder a MySQL desde el cPanel:

Tecleamos nuestro dominio.com/cpanel

Esperamos...

Nos solicita el usuario y la contraseña, una vez introducida esperamos...



Una vez dentro del cPanel tenemos decenas de opciones y se pierde de vista dónde se encuentra PhpMyAdmin...

La primera vez, movemos la caja "Base de datos" con *drag* a la primera posición, para que no se pierda entre las múltiples opciones.



Hacemos clic en PhpMy Admin y esperamos...

En servidores debidamente configurados, el PhpMyAdmin se accede desde cPanel usando https y con un número de puerto diferente al 80, por cuestiones de seguridad. De ahí la razón de entrar al phpMyAdmin desde el cPanel...

Una vez cargada la pagina



Hacemos clic en la base de datos a usar y esperamos...

Realmente es tortuosa e incómoda la edición de tablas y registros porque tardamos otros 2 o 3 clics más para alcanzar un objeto de nuestra base de datos.

Contemos los "Esperamos" de este artículo y multipliquemos por el tiempo de carga y salvo

accesos muy rápidos o locales, hemos gastado mucho tiempo si este patrón de trabajo lo repetimos varias veces al día.

Conocer HeidiSQL

A comienzos de la popularización en la web de MySQL, había comenzado una fiebre de construcción de clientes para dicho gestor de base de datos, que desafortunadamente eran en muchos casos incompatibles con cada nueva versión de MySQL.

Entre la multitud se fue abriendo paso MySQLFront por ser rápida, sin errores y gratuita. Cuál sería la sorpresa (hace tiempo ya) al intentar descarga en su momento de la página oficial y leer que el proyecto había sido vendido...

Afortunadamente, el desarrollador había continuado el proyecto con otra herramienta que sigue siendo libre de uso y gratuita: www.heidisql.com

Y este maravilloso cliente nos puede eliminar los "esperando" y dejarnos todo el asunto de la edición con dos clics. No necesitamos decir lo agradecidos que podremos estar con el uso de esta herramienta, que además de ser gratuita, nos permite hacer las mismas cosas que ya venimos haciendo con otros administradores como PhpMyAdmin.

Entre otras posibilidades, HeidiSQL nos permite:

- Conectar con varios servidores a la vez en una única ventana.
- Conectar con servidores MySQL por línea de comandos.
- Conectar con SSH o realizar conexiones SSL.
- Editar tablas, vistas, procedimientos almacenados, triggers, eventos agendados...
- Crear reportes SQL.
- Exportar o importar datos desde o hacia otras fuentes o bases de datos. Por ejemplo, importar datos que haya en ficheros de texto o exportar los datos de las tablas a ficheros de texto con diversos formatos como CSV, HTML, XML, SQL, arrays de PHP, etc.
- Administrar los privilegios de los usuarios.
- Escribir consultas con resaltado de código SQL y completado de código y preformato de código SQL para una mejor lectura.
- Monitorizar procesos del cliente y matarlos si lo necesitamos.
- Búsquedas de un texto, no solo en una tabla, sino en múltiples, por si no sabemos dónde se encontraba.
- Optimización y reparación de tablas, etc.

Configurar el acceso remoto a tu servidor de base de datos con cPanel

Para usarlo, la única condición es que tu servidor acepte conexiones remotas a MySQL, cosa que no está disponible en todos los servidores Linux.

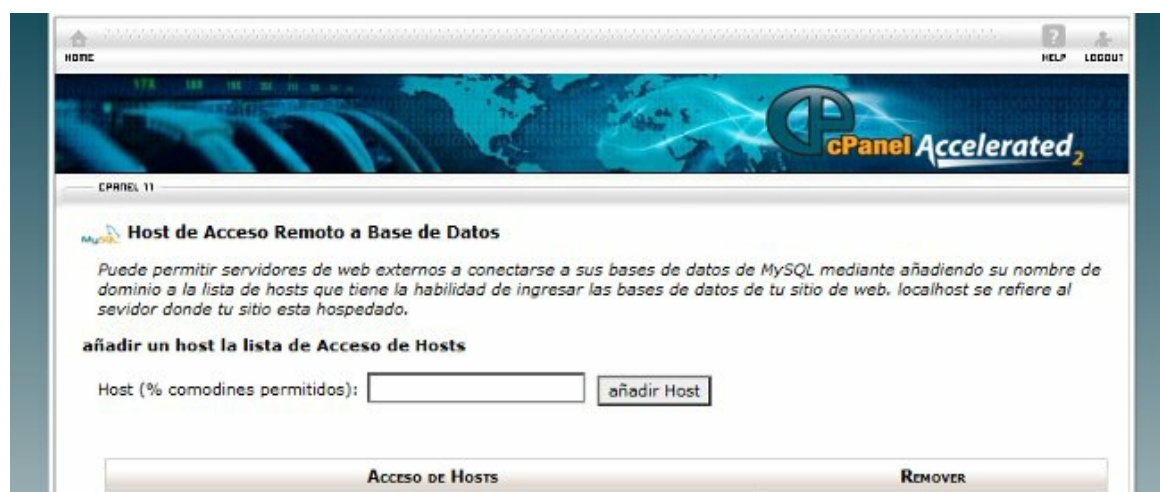
Si tu servidor permite dichas conexiones, debes averiguar cuál es tu IP pública en paginas como www.cualesmiip.com y anotar cuál es el numero de IP que tienes en el acceso a Internet.

Nota: debo ser enfático en el asunto porque muchas veces la dirección IP de nuestro computador no es la dirección pública que asigna nuestro proveedor de Internet, sino una IP local cuando estamos conectados a un *router*, etc. Por otra parte, cabe decir también que muchas veces nuestra IP va cambiando con el tiempo, lo que se conoce como IP dinámica. En estos casos podemos intentar hablar con nuestro proveedor para que nos proporcionen una IP estática, que siempre sea la misma, lo que nos ahorrará tener que configurar repetidas veces la IP desde la que se permite acceder a nuestro servidor.

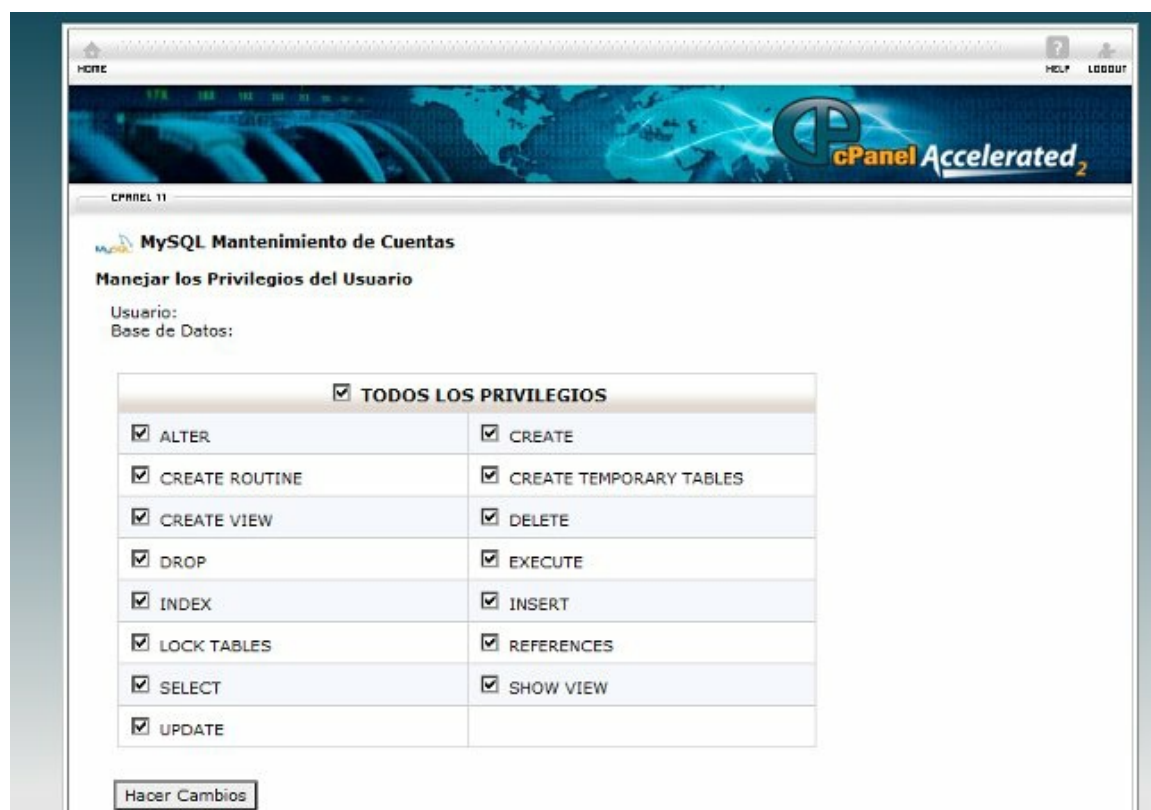
Nos dirigimos a nuestro cpanel: En nuestra caja de base de datos hacemos clic en Mysql Remota.



En la siguiente página debemos introducir nuestra dirección IP pública:



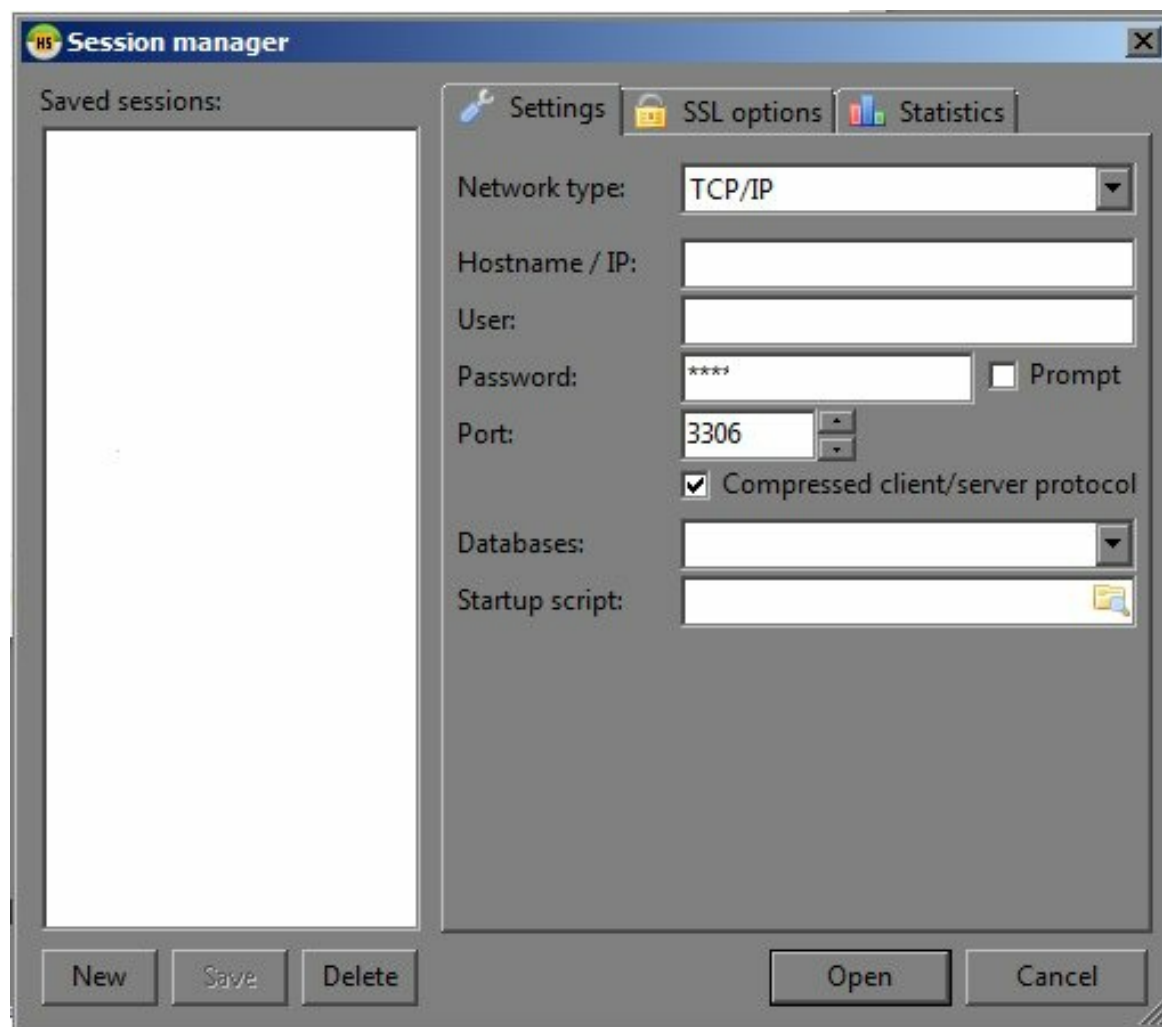
Vamos a utilizar el usuario habitual para nuestra conexión a MySQL. En algunos casos es necesario volver a asignar los permisos luego de introducir una IP en la opción de acceso Remoto.



Y una vez realizados todos estos pasos, procederemos a instalar HeidiSQL.

El programa, al iniciar, nos mostrará un formulario de conexiones y debemos configurar

nuestra conexión.



Es tan sencillo como escribir el Hostname, User, Password (en algunos casos debemos escribir el nombre de la base de datos).

Nota: Debo señalar que el uso repetido de "en algunos casos" es porque después de experimentar con muchas compañías de hosting con servidores compartidos y dedicados he notado que existen diferencias de configuración según la versión de Linux, de cpanel y demás que nos impulsan a probar antes de desistir si no logramos conectarnos a MySQL por vía remota.

Por ejemplo, algunos *Hosting* te asignan otro nombre para acceder a MySQL.

Algunos debemos agregar la combinación de nombre de usuario de acceso mas usuario de base de datos.

Ejemplo "nombredeusuario_nombreusermysql"

Al crear el usuario cPanel nos indica el nombre exacto de todos modos.

Con una interfaz intuitiva HeidiSQL nos permite modificar rápidamente las propiedades de una base de datos tabla o campo o contenido de algún registro.

Como estamos editando la base de datos directamente, en algunos casos (si navegamos por el contenido de un registro) cambiando el foco por ejemplo, se actualiza el contenido automáticamente.

Así que debemos tener cuidado porque no existe un "Undo" o Deshacer.

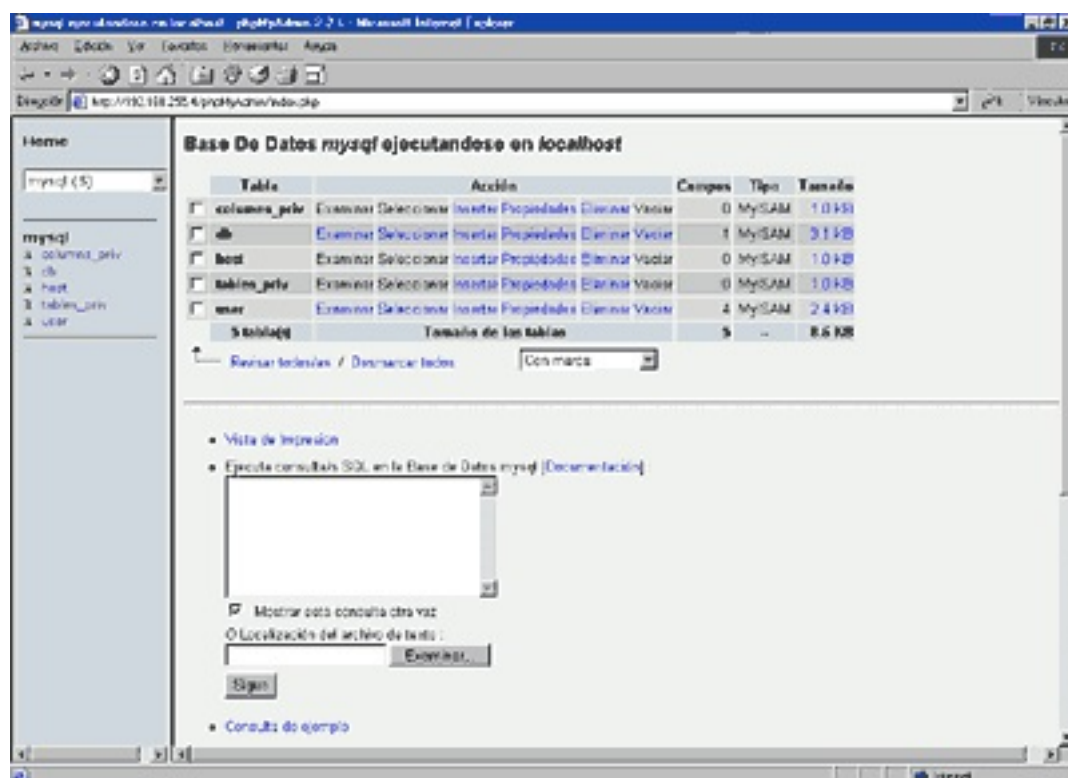
Espero que puedas aprovechar esta herramienta y reducir el tiempo edición a tu base de datos en MySQL.

Este artículo es obra de *Heisler Palma*
Fue publicado por primera vez en 10/07/2012
Disponible online en <http://desarrolloweb.com/articulos/gestor-mysql-heidisql.html>

phpMyAdmin

Un proyecto de código abierto en PHP para administrar la base de datos MySQL a través de una interfaz web. Descripción funcionalidades y ayudas para la instalación.

phpMyAdmin es un programa de libre distribución en PHP, creado por una comunidad sin ánimo de lucro, que sólo trabaja en el proyecto por amor al arte. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva.



La aplicación en si no es más que un conjunto de archivos escritos en PHP que podemos copiar

en un directorio de nuestro servidor web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas. La herramienta nos permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas y un largo etcétera, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

Página de phpMyAdmin

La página de inicio del proyecto es <http://www.phpmyadmin.net/>. Desde allí podemos descargar los ficheros de la última versión de la aplicación, que posteriormente debemos colocar en nuestro servidor web. También podemos encontrar a phpMyAdmin dentro de la red Sourceforge.net, que es un sitio que recoge multitud de proyectos "Open Source" (código abierto).

Hay varias versiones disponibles, pero es recomendable escoger la que nos aconsejen como la última versión estable (The last stable versión). En el momento de escribir este artículo era la 2.2.6. De modo que, si nuestro sistema es Windows, descargaremos el archivo phpMyAdmin-2.2.6-php.zip

Los archivos que hemos descargado son de la versión 4 de PHP, aunque también ofrecen la posibilidad de bajarse los archivos que guardan compatibilidad con la versión 3 de PHP, para que aquellos que no dispongan del motor de PHP más actual.

La pagina de inicio del programa también nos ofrece la posibilidad de ver un demo online, aunque nos avisan de que el servidor donde se aloja puede estar caído.

<http://www.phpmyadmin.net/phpMyAdmin/>

Home

desarrolloweb (9)

desarrolloweb

- academia_seccion
- seccion_curso
- tacademia
- tcertificados
- tcursos
- tmetodologia
- tprovincias
- tseccion
- ttipos

Base De Datos *desarrolloweb* - tabla *tacademia* ejec

Mostrando campos 0 - 2 (3 total)

SQL-query : [Editar]
SELECT * FROM `tacademia` LIMIT 0, 30

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

		id_academia	nombre	contrasena	persona	em
Editar	Borrar	1	academia12	password12	persona12	em
Editar	Borrar	2	academia2	password2	persona2	em
Editar	Borrar	9	333	333	33	33

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

Instalando phpMyAdmin

Una vez descargada la última versión la tenemos que descomprimir, con lo que obtendremos los ficheros PHP que conforman la herramienta y colocarlos dentro del directorio de publicación de nuestro servidor web.

Nota: Recordamos que phpMyAdmin es un proyecto escrito en PHP, por lo que necesitaremos colocar los archivos en un servidor web que permita programación de páginas PHP.

Además, deberemos acceder a la herramienta a través de la dirección del servidor web, seguida del directorio en el que tenemos los archivos que hemos descomprimido. Por ejemplo, si nuestro servidor es el [PWS](#) y hemos colocado los archivos dentro del directorio de publicación (Generalmente C:\Inetpub\wwwroot), en el subdirectorio phpMyAdmin, debemos escribir algo como <http://localhost/phpMyAdmin>

Si tuviéramos instalado un servidor Apache los colocaríamos en la carpeta que hayamos indicado como "documentRoot", que suele ser htdocs.

Lo primero que podemos leer es el archivo de la documentación, que encontramos junto con los archivos de phpMyAdmin. Explica datos generales del programa, como sus requerimientos, instrucciones de instalación, configuración, preguntas frecuentes, etc.

Posteriormente, tal como explica la documentación, hay que editar el archivo config.inc.php para cambiar los valores de host de la base de datos (ordenador que tiene instalado el MySQL) y el usuario y password con el que nos conectamos. Se pueden configurar muchos aspectos en

la herramienta, aunque ahora solo comentaré los que he encontrado esenciales para hacerla funcionar, en la documentación tenemos un apartado dedicado por completo a especificar el sentido de cada variable.

```
$cfgPmaAbsoluteUri
```

Debemos asignarlo a la ruta completa necesaria para acceder a phpMyAdmin. Podría ser algo como <http://localhost/phpMyAdmin> o <http://www.midominio.com/phpMyAdmin>

```
$cfgServers[$i]['host'] string
```

El nombre del host de la base de datos. Por ejemplo localhost, si es que es el mismo ordenador donde estamos instalando phpMyAdmin y la base de datos. También podría ser la dirección IP del ordenador al que nos conectamos.

```
$cfgServers[$i]['user'] string
```

```
$cfgServers[$i]['password'] string
```

El par usuario/contraseña que debe utilizar phpMyAdmin para conectarse con el servidor MySQL.

Con estas sencillas configuraciones ya podemos acceder a phpMyAdmin y trabajar con nuestra base de datos a golpe de ratón, que resulta muy de agradecer teniendo en cuenta que, en caso de no tener esta herramienta u otra parecida, la otra opción consistiría en utilizar el lenguaje SQL, y, en caso de que la base de datos esté alojada remotamente en Internet, no podríamos hacerlo sino es con acceso por TELNET al servidor de la base de datos.

Nota: En DesarrolloWeb puedes conocer más cosas de PHP y MySQL.

En la [sección de PHP](#), podras aprender mucho de PHP, algo sobre MySQL e incluso sobre el lenguaje SQL. En el [directorio dedicado a MySQL](#) hay algunas referencias a artículos y enlaces externos.

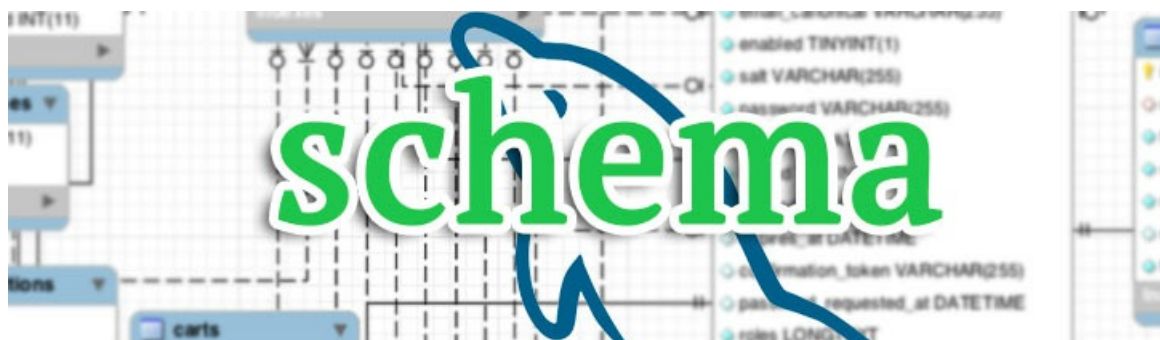
Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 19/07/2002
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Esquema gráfico de la base MySQL con MySQL Workbench

Cómo realizar el esquema gráfico, relaciones entre tablas y sus distintos campos, nombres, etc. usando el programa gratuito MySQL Workbench.

En este artículo te vamos a explicar cómo conseguir realizar rápidamente un gráfico del esquema de tu base de datos MySQL, usando un software gratuito que seguramente conozcas, llamado MySQL Workbench.

MySQL Workbench es un programa creado por Oracle, el actual dueño del sistema gestor de base de datos MySQL. Sus funciones son diversas, pudiendo realizar la administración completa de una base de datos, crear, editar o eliminar tablas, definir relaciones, ver sus datos, entre sus opciones más básicas. En resumen, es un completo programa, que podría sustituir al popular PhpMyAdmin, pero de una manera mucho más profesional.



La parte que vamos a tratar en este artículo es la de crear el esquema de relaciones entre tablas, que es algo que nos vendrá bien tener a mano cuando estamos desarrollando un proyecto.

Conectar con una base de datos

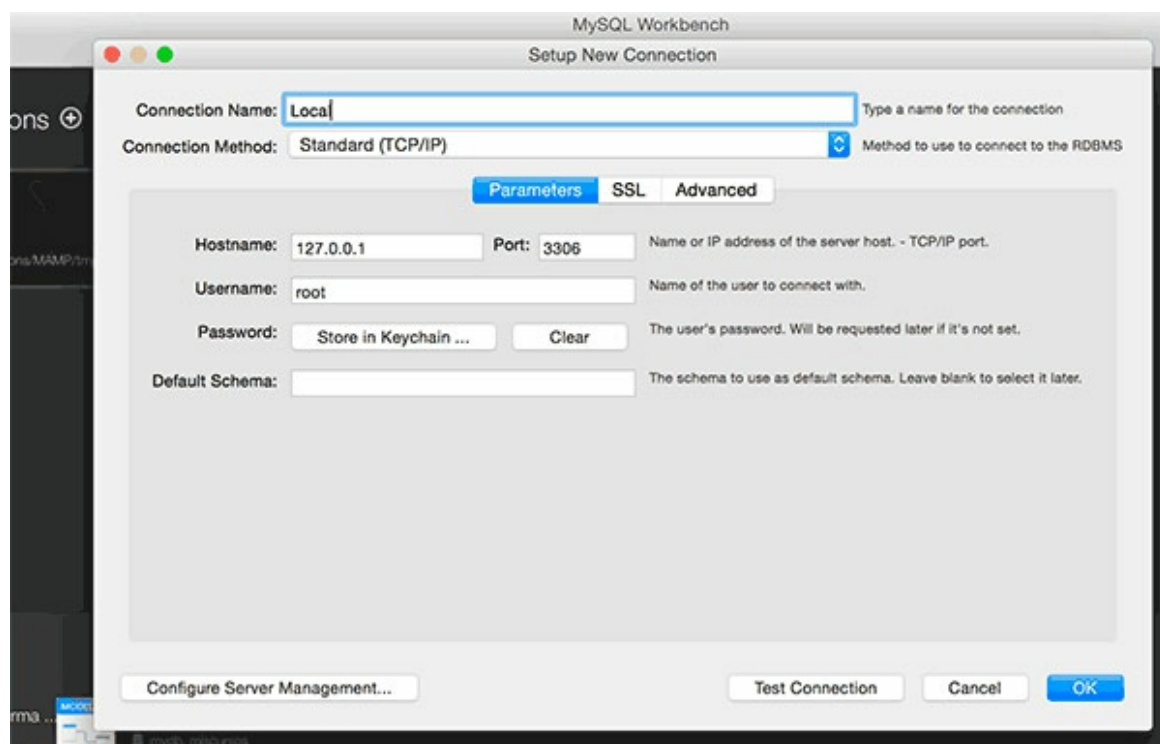
Lógicamente, uno de los pasos para obtener el esquema será conectar con la base de datos que tengamos que obtener la representación de su schema. Este paso se realiza desde la pantalla "Home" de MySQL Workbench, desde el icono de la casita en las pestañas del programa.

Aquí podremos administrar las conexiones con base de datos, lo que nos facilitará el acceso a diversos servidores de MySQL con los que trabajemos diariamente. Para crear nuevas conexiones tienes que apretar el botón "+".



Esta parte de la conexión depende mucho del servidor MySQL con el que quieras enlazarte, porque existen diversos tipos de conexiones posibles, desde locales a remotas, así como diversos protocolos, como el TCP/IP o el SSH. Los datos de conexión los debes de conocer tú, pero si estás conectando a un servidor local generalmente usarás TCP/IP y hostname 127.0.0.1. Usuario root normalmente y la clave que tengas.

Una vez configurada la conexión, aprietas el botón "Ok", aunque también puedes antes probar que funciona con el botón "Test Connection".

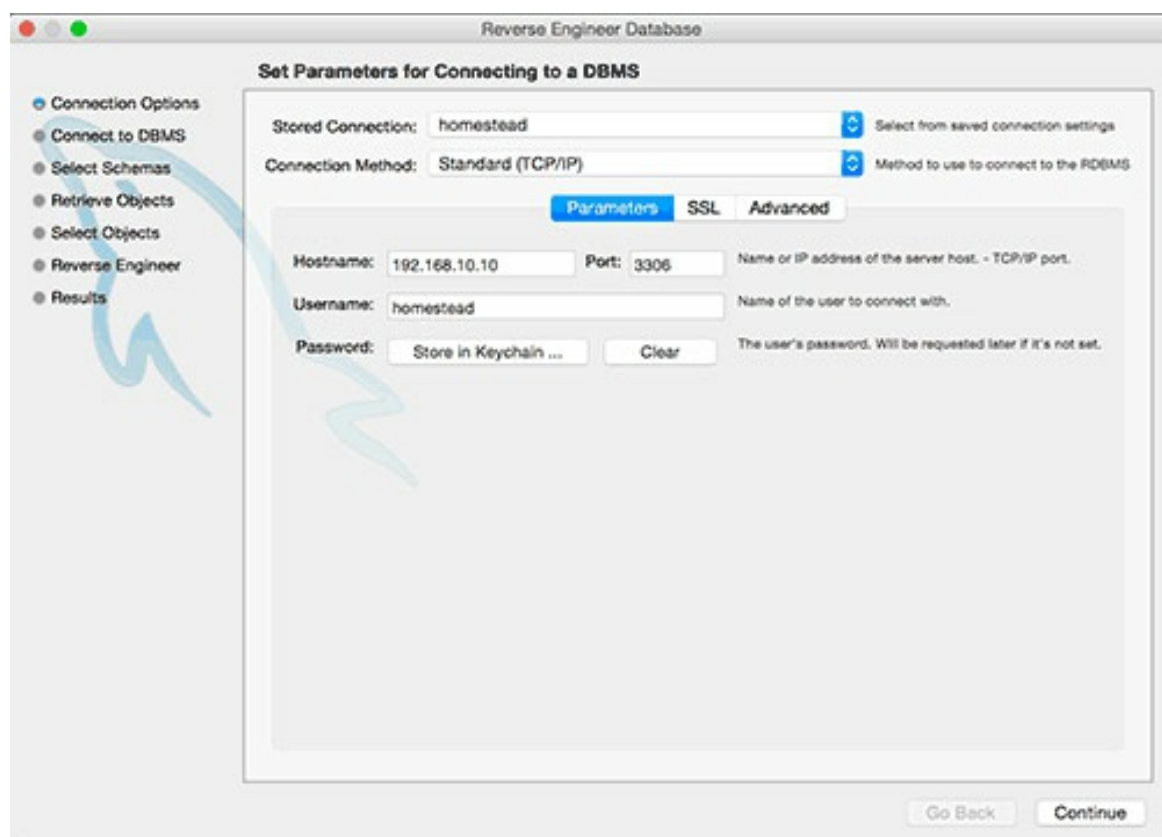


Nota: Para trabajar con una base de datos determinada, tendrás que elegirla en la sección "SCHEMAS", en la parte de la izquierda.

Database / Reverse Engineer

A partir del menú "Database, Reverse Engineer" podrás comenzar el proceso para crear el esquema de la base de datos que necesites. Podrás encontrar un asistente para realizar la serie de pasos necesarios.

Comienzas indicando la conexión que quieras usar, dentro de todas las conexiones que puedes tener con distintos servidores MySQL, método de conexión, etc. Eso te rellenará los campos de conexión del formulario de abajo. Aprietas "Continue".



Entonces MySQL Workbench se conecta para tomar la lista de los schemas disponibles en el servidor conectado. Apretas "Continue" una vez el proceso termine.

Aparecerá entonces la lista de los esquemas disponibles. Seleccionas pues el que te interese y luego el botón para continuar.

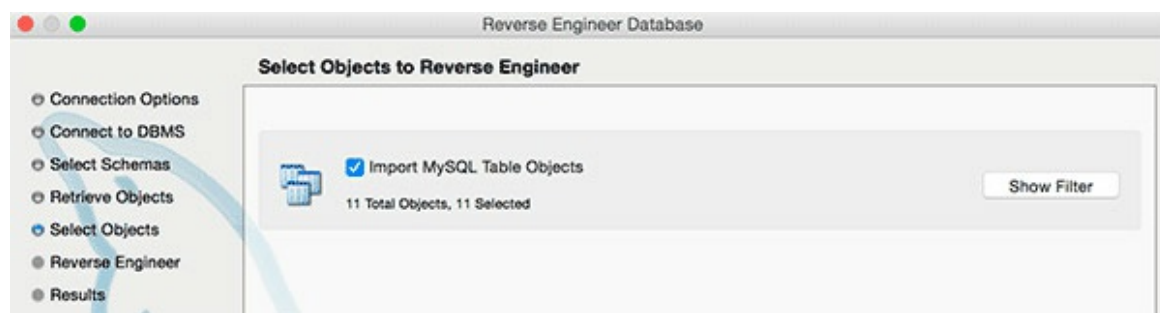


Nota: Lo que conocemos popularmente como una "base de datos", osea un conjunto de tablas relacionadas entre si, en MySQL Workbench se conoce con la palabra "schema". El concepto es equivalente.

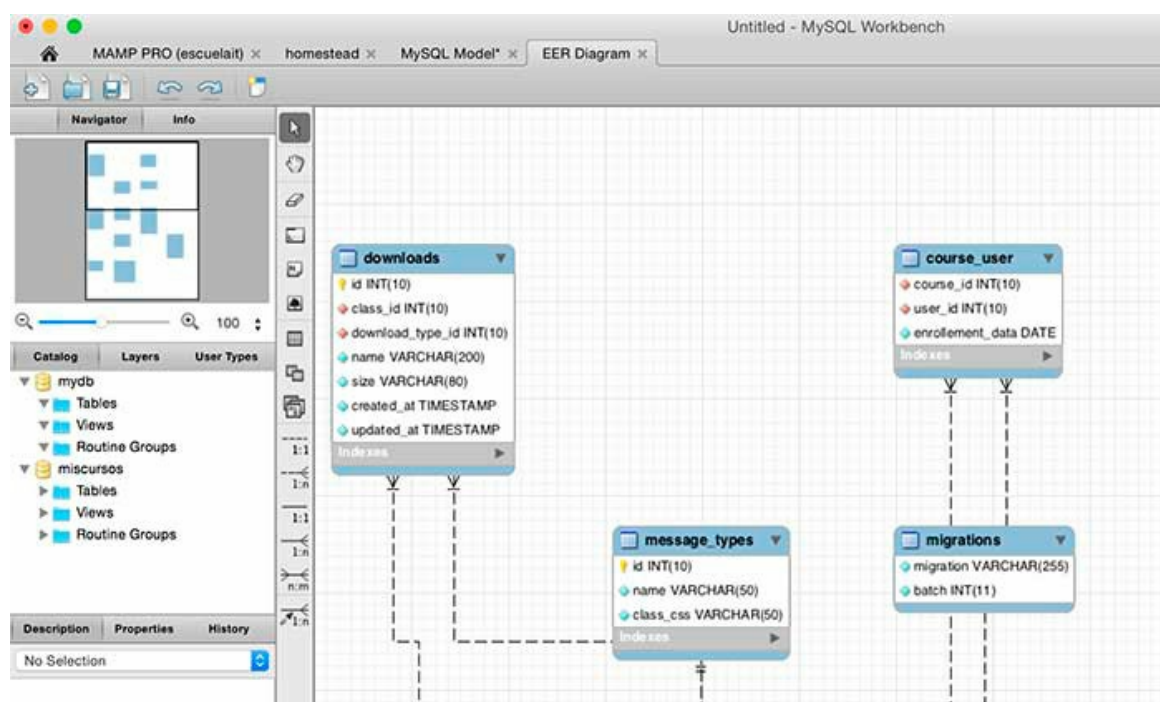
Esa operación realizará la lectura de todas las tablas y sus descripciones, relaciones, etc. Apretas de nuevo "Continue"

Por último nos aparecen los objetos de los que se va a realizar la ingeniería inversa para

obtener el esquema de relaciones entre tablas. Seleccionas y entonces pulsas el botón "Execute".



Terminado el proceso aparece ya el esquema de tu base de datos. Tendrás que pulsar "Continue" una vez más para cerrar el asistente.

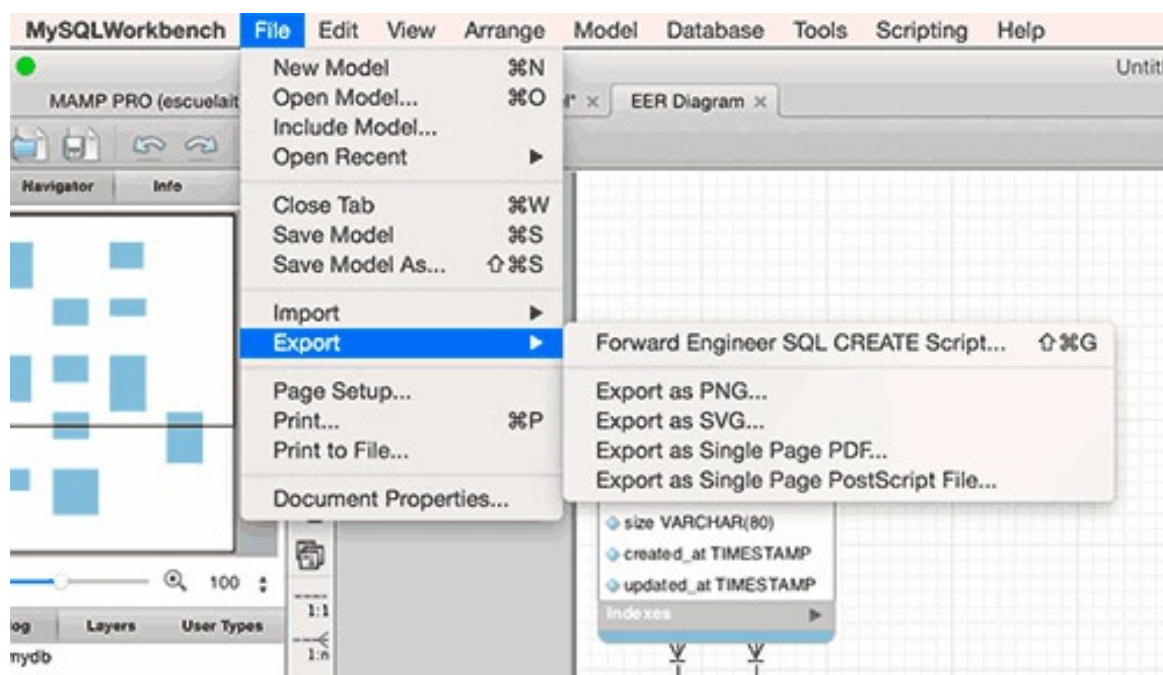


El esquema te aparece en una pestaña nueva dentro de MySQL Workbench. Es el momento de acomodar tus tablas para colocarlas en el lugar más lógico, según te interese que se organice el diagrama. Simplemente pinchas y arrastras las tablas donde quieras.

Exportar el esquema de la base de datos MySQL a PDF

El esquema lo puedes salvar en el formato interno de MySQL Workbench, con el botón correspondiente, eso te facilitará abrirlo más adelante para recolocar las tablas por ejemplo. Pero generalmente lo que querrás hacer es exportar el diagrama a un PDF para poder abrirlo con otros programas, enviarlo a tus compañeros o imprimirlo para tenerlo a mano durante el desarrollo.

Esto se hace fácilmente desde el menú "File / Export / Export as Single Page PDF...". Abrirá un diálogo para seleccionar el archivo PDF a generar. Esta parte ya no tiene más misterio.



Esto es todo! El diagrama ha quedado listo para que el esquema de la base de datos esté siempre a mano.

Este artículo es obra de *Miguel Angel Alvarez*
 Fue publicado por primera vez en 28/09/2015
 Disponible online en <http://desarrolloweb.com/articulos/esquema-grafico-base-mysql-workbench.html>

Sequel Pro: interfaz gráfica para MySQL

Conoce Sequel Pro, un software de interfaz gráfica para administración de bases de datos MySQL muy potente e intuitivo. Disponible para Mac OS X.

A lo largo del [Manual de MySQL](#) hemos podido conocer diversas herramientas de interfaz gráfica para administrar bases de datos MySQL de una manera cómoda. Hoy le toca el turno a una de las aplicaciones que más vengo usando durante los últimos años, que es Sequel Pro.

Realmente recomiendo usar Sequel Pro, incluso más que otros programas más populares como lo puedan ser MySQL Workbench. El motivo es que me parece muy cómodo y más fácil de usar que otras alternativas, aunque tiene un detalle que resultará muy inconveniente para la mayoría de los lectores: sólo funciona en Mac OS X. Así pues, si tú trabajas con Windows o Linux no te queda otra que buscar otro programa de interfaz gráfica. Puedes consultar más alternativas en el mencionado [Manual de MySQL](#).



Pros y contras de la Interfaz gráfica para MySQL en general

Antes de comenzar a hablaros de Sequel Pro debemos comentar aunque sea brevemente sobre los pros y los contras de trabajar con programas de interfaz gráfica para la administración de MySQL. Básicamente la interfaz gráfica nos ofrece más comodidad, pero no siempre la tendremos disponible allá donde vayamos!

Como ventajas podemos destacar:

- Te evita tener que escribir comandos constantemente para operar con las bases de datos, con lo que puedes acelerar un poco tu trabajo y ahorrar algo de complejidad en tu día a día.
- Permite hacer operaciones como backups o restauraciones de bases de datos muy rápidamente.
- En el caso de Sequel Pro, sirve para administrar tanto bases de datos locales como remotas.

Como desventajas se deben enumerar los siguientes puntos:

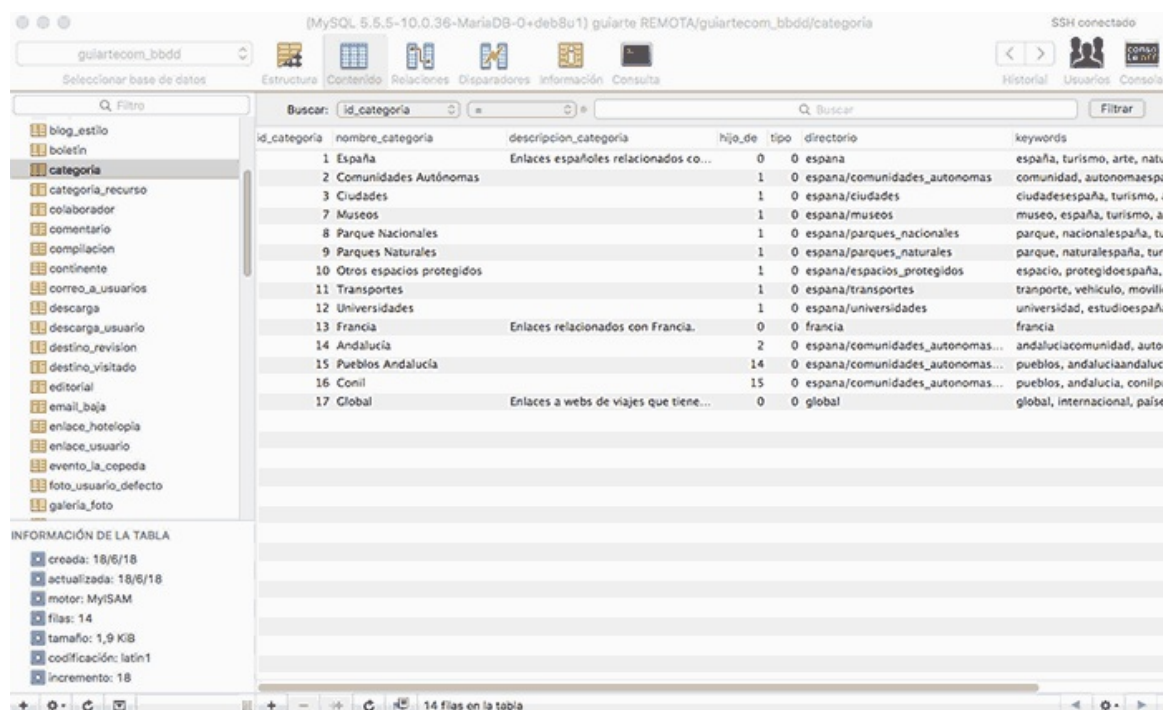
- No siempre lo tendrás disponible. Habrá momentos en tu vida profesional que tendrás que conectar a un servidor, en un lugar donde no tengas tu programa de interfaz gráfica o en casos en los que no se permita este tipo de accesos. En ese momento, si lo único de que dispones es la línea de comandos, te será esencial dominarla.
- Puede provocar que no aprendas tanto SQL y las herramientas de administración de MySQL de bajo nivel, como [MySQL Dump](#). Puede ser un problema a tu crecimiento profesional pero sobre todo un problema cuando lo único de que dispongas sea la línea de comandos.

Por tanto, nuestra recomendación es: Aprender siempre a [trabajar con MySQL desde la línea de comandos](#). Cuando ya tengas un conocimiento y experiencia suficiente, comenzar a usar los programas de interfaz gráfica.

Funcionalidades y características de Sequel Pro

Sequel Pro es un software gratuito, que se puede descargar y usar libremente para cualquier tipo de proyecto y organización. Básicamente sirve como cliente de MySQL, permitiendo conectar a servidores locales y remotos de una manera sencilla y realizar prácticamente todas las tareas de administración de las bases de datos.

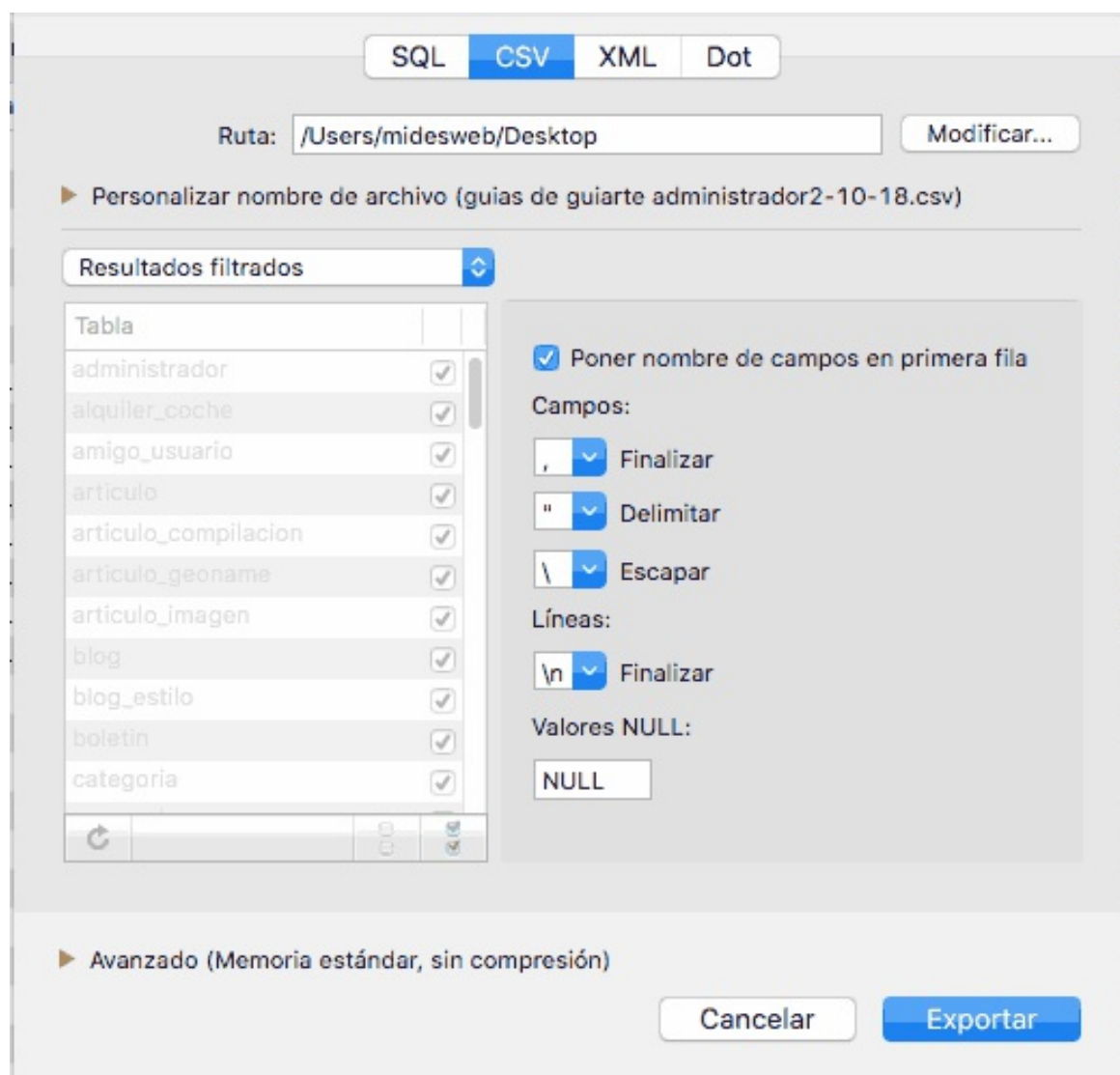
Esta es una pantalla general de Sequel Pro en la que se muestra el detalle de registros de una tabla:



Algunas de las funcionalidades más importantes de Sequel Pro son:

- Exploración de tablas para consultar su estructura, datos, relaciones, etc.
- Creación y modificación de la estructura de tablas
- Panel para lanzar consultas de una manera rápida
- Creación y restauración de backups, tanto en sentencias SQL como en otros formatos, como CSV o XML.
- Permite filtrar información de las tablas cómodamente, para encontrar el dato que se necesite.
- Ofrece diversas maneras de conexión a bases de datos remotas, como por ejemplo por SSH, lo que hace muy sencillo que puedas administrar la base de datos alojada en un servidor de Internet, de manera gráfica y casi como si estuvieras en frente de la máquina remota.
- Realiza de manera automática la paginación de los datos de tablas grandes, de modo que los puedas consultar de manera sencilla y con suficiente velocidad.

En la siguiente imagen puedes ver el diálogo para exportar una base de datos a CSV.



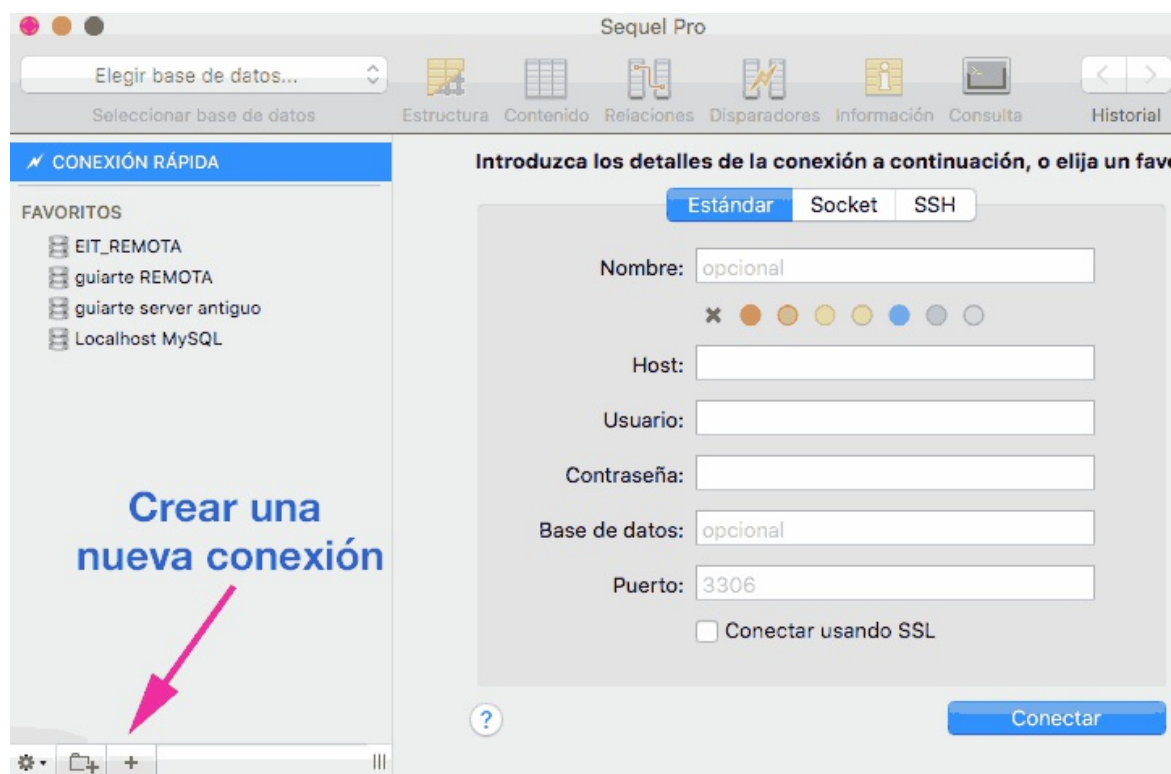
Conexiones locales y remotas con Sequel Pro

Es posible administrar bases de datos que tengas instaladas en tu ordenador local, el que usas para desarrollar, pero también aquellas que tengas en servidores en la misma red o en servidores de internet (dedicados, VPS...).

Para ello existe un panel para la gestión de conexiones, que a decir verdad es el escollo más importante para los usuarios que comienzan con Sequel Pro. Puede que tengas dudas al conectar a bases de datos al principio, pero una vez conectados al sistema gestor las tareas de administración se hacen de manera bastante sencilla.

El primer paso recomendado para hacer una conexión con una base de datos es crear dicha conexión, lo que resultará mejor que la opción de conexión rápida, dado que así podrás reutilizar esa conexión en futuras ocasiones sin tener que configurar nada. Para crear la conexión encontrarás un botón con el signo "+" en la ventana de conexión.


En la siguiente imagen está señalado el botón de nueva conexión. Si no lo marcas la opción que estará activa será la de conexión rápida, que no guardará tus datos de acceso. Además, en la columna de la izquierda podrás ver las conexiones almacenadas en el programa.



Para bases de datos locales

Se realiza mediante la conexión estándar. El nombre de host que se usa es 127.0.0.1 y solamente tienes que indicar tu usuario y clave configurada en tu servidor local. Opcionalmente puedes indicar el nombre de la base de datos o modificar el puerto, o dejar 3306 que es el puerto predeterminado de MySQL.

En la siguiente imagen puedes ver el cuadro de diálogo de conexión de Sequel Pro, configurado para el acceso al servidor local. Si no le indicas la base de datos podrás conectar luego a cualquier base de datos del servidor, desde la pantalla principal del programa.



Bases de datos remotas

Lo más normal en estos casos es conectar por SSH, que nos permite acceder a cualquier servidor donde tengas credenciales (ya sea mediante usuario/clave o por una llave SSH).

Esta configuración es un poco más compleja, pero seguro que después de dos o tres intentos lo conseguirás. Los detalles importantes que debes conocer son:

- El host de MySQL se refiere al servidor donde está la base de datos, una vez te has conectado por SSH. Por ejemplo, si te conectas por SSH a un servidor y en ese mismo servidor está alojado el demonio de MySQL, entonces el host MySQL será 127.0.0.1. Es un error habitual poner aquí la IP del servidor.
- El Host SSH es la IP de nuestro servidor remoto, sobre el que tenemos un modo de acceso por SSH. Entonces tendrás que colocar el usuario y también la llave SSH o clave que uses para acceder. Todos estos datos son los que tienes que usar para conectar por línea de comandos con tu servidor remoto.

En la siguiente imagen puedes ver el diálogo de conexión con un servidor de base de datos remoto con los datos rellenos, para que te sirva de guía para realizar tu propia configuración en Sequel Pro.

Introduzca los detalles de la conexión a continuación, o elija un favorito

Estándar Socket SSH

Nombre: guiarte REMOTA

Host MySQL: 127.0.0.1

Usuario: guiarte REMOTA

Contraseña:

Base de datos: guiarte REMOTA

Puerto: 3306

Host SSH: 51.6...

Usuario SSH: root

Clave SSH: ~/.ssh/guiarte REMOTA

Puerto SSH: opcional

☐ Conectar usando SSL

Conectar

Conclusión

Si has trabajado con MySQL, usando cualquier herramienta de interfaz gráfica, Sequel Pro no te resultará en problema alguno, ya que es bastante elemental y sencillo de usar. Si no es el caso, te resultará muy cómodo y agradecido realizar muchas de las operaciones del día a día que vienes realizando mediante comandos.

Sequel Pro es una excelente opción para los desarrolladores, por ser fácil, intuitivo y muy rápido. El único problema es que a día de hoy no está disponible en otros sistemas aparte de OS X. Descargas y más información disponible en la página del producto:

<https://sequelpro.com/>

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 02/10/2018
Disponible online en <http://desarrolloweb.com/articulos/sequel-pro-interfaz-grafica-mysql.html>

MySQL Administrator (Conocido hoy como MySQL Workbench)

Un programa muy útil para administrar, visualmente y de manera sencilla, servidores de bases de datos MySQL.

Actualización: Este artículo trata sobre un software que ha cambiado de nombre, pues MySQL Administrator fue la base del programa actual conocido actualmente como MySQL Workbench. Hoy ya no encontrarás MySQL Administrator para descarga en Internet y si lo encuentras tampoco te lo recomendaríamos instalar, debido a que ha desaparecido. Si te interesa tener un programa de interfaz gráfica para MySQL te recomendamos el propio MySQL Workbench o encontrar otras posibilidades en el [Manual de MySQL](#). De hecho, si buscas por este software en la página de MySQL podrás ver que se te redirige a el programa MySQL Workbench.

MySQL Administrador es el nuevo software de administración de servidores de Bases de Datos de MySQL que ha creado MySQL AB. Se trata de un software multiplataforma, que por el momento se encuentra disponible para Linux y Microsoft Windows y que cuenta con un entorno gráfico de usuario muy intuitivo.

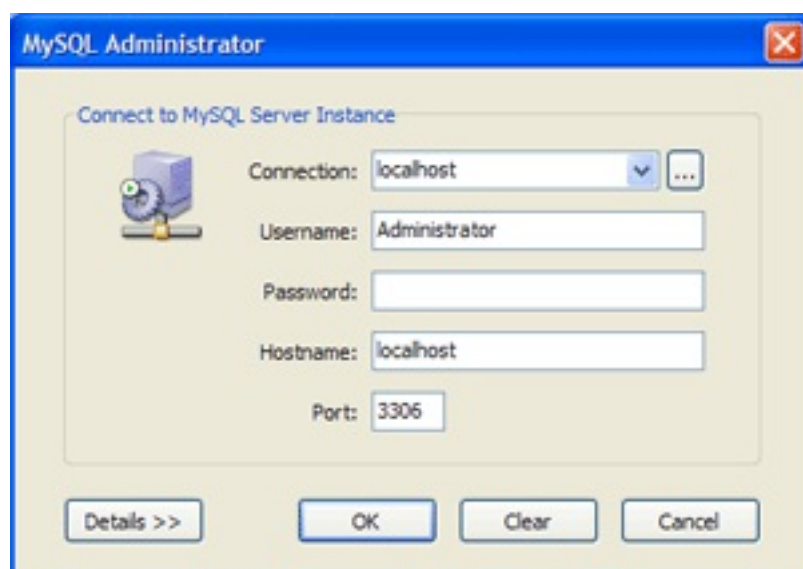
Este nuevo producto suple las carencias que tiene MySQL Control Center en el área de Administración de servidores. MySQL Control Center en estos momentos está deprecado, ha quedado obsoleto y no sigue desarrollándose. Se ha sustituido por el conjunto de programas MySQL Administrator y MySQL Query Browser.

MySQL Administrador es una herramienta que permite realizar tareas administrativas sobre servidores de MySQL incluyendo:

- la configuración de las opciones de inicio de los servidores
- inicio y detención de servidores
- monitorización de conexiones al servidor
- administración de usuarios
- monitorización del estado del servidor, incluyendo estadísticas de uso
- visualización de los logs de servidor
- gestión de copias de seguridad y recuperaciones
- visualización de catálogos de datos

Instalación y configuración:

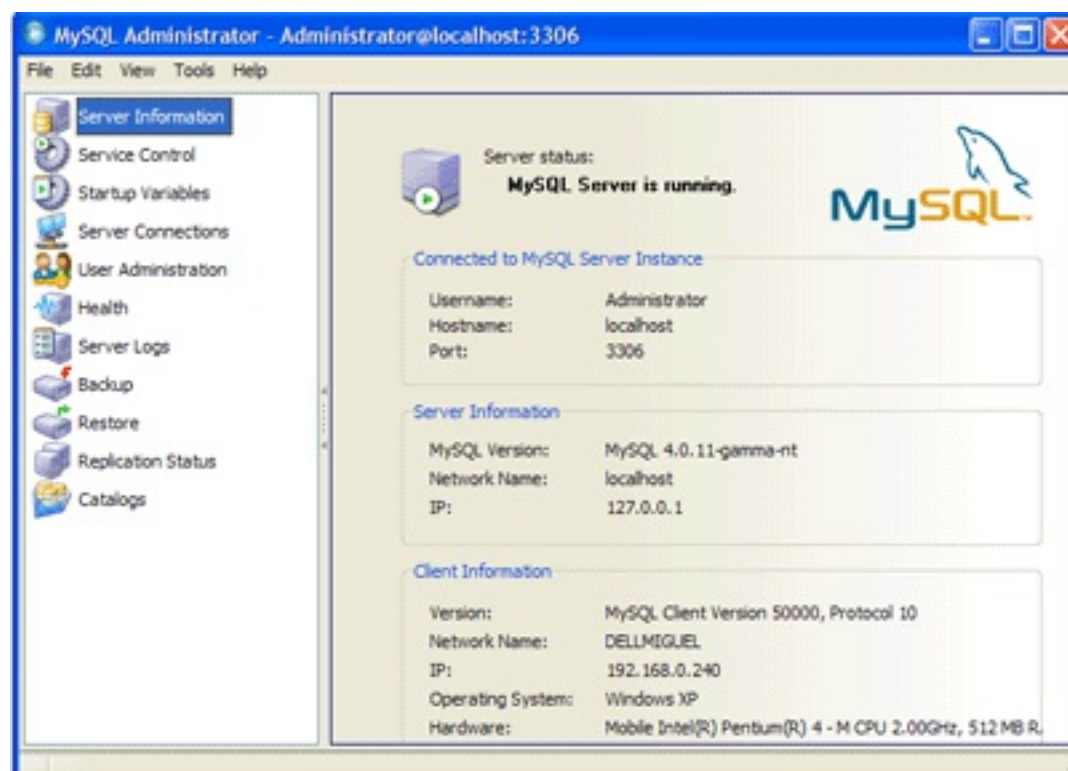
La descarga del programa se hace desde la página web de MySQL, en la página de MySQL. Existen una serie de FAQs que ayudan en la instalación en caso de dudas, aunque la misma es bastante simple, en <http://www.mysql.com/products/administrator/>.



Una vez instalado es necesario configurar una primera conexión contra un servidor de MySQL para poder acceder a la administración del mismo.

En la pantalla de configuración se deberán especificar los datos del servidor, nombre de usuario, contraseña y puerto, de la base de datos a la cual se desea conectar.

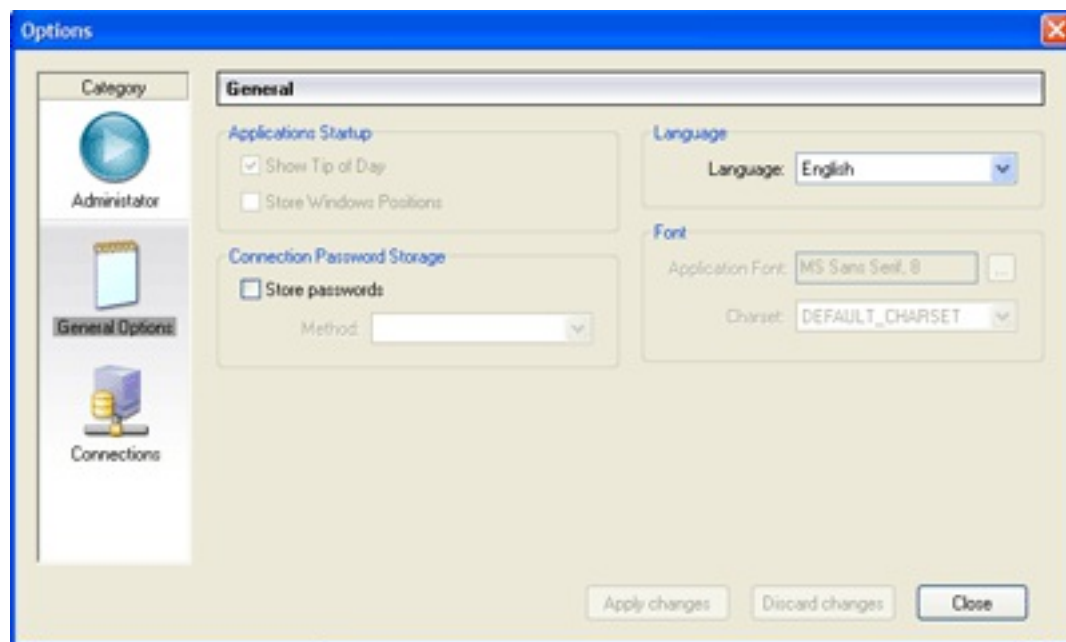
Una vez introducidos de forma correcta, aparece la pantalla principal del programa:



Desde la pantalla principal se puede ver la información relativa al servidor de base datos al que se ha conectado, así como las características de la máquina en la que se está ejecutando el programa.

En la barra de menú a través de Tools, se tiene acceso a una consola de sistema (Windows o

Linux dependiendo del caso), un cliente de conexión a MySQL bajo línea de comandos y a MySQL Query Browser. Desde este mismo menú, se accede a las opciones de configuración del administrador y de las conexiones que hay configuradas:



En la sección de conexiones se encuentran almacenadas todas las conexiones a servidores que hay configuradas.

Algunas utilidades accesibles desde la ventana principal del programa son:

- **Service control:** Inicio y detención de servidores (sólo accesible si se ha conectado con un servidor MySQL en la máquina local).
- **Startup variables:** Configuración del servidor y las variables de inicio (sólo accesible si se ha conectado con un servidor MySQL en la máquina local).
- **User Administration:** Para la gestión de usuarios y permisos.
- **Server connections:** Visualiza y gestiona las conexiones abiertas con el servidor de bases de datos.
- **Health:** Información sobre la carga del servidor.
- **Server Logs:** El historial de logs del servidor.
- **Replication Status:** Con información de los sistemas replicados.
- **Backup:** Para hacer una copia de seguridad de las bases de datos.
- **Restore:** Para restaurar las copias de seguridad.
- **Catalogs:** Para mostrar las bases de datos, visualizar, crear y editar las tablas.

Nota: En el Taller de MySQL tenemos otros artículos para explicar tareas básicas de administración de MySQL. Tenemos un artículo que puede ser interesante para leer ahora que explica cómo realizar la [administración de usuarios con MySQL Administrator](#).

Conclusión

A la base de datos MySQL le faltaba una herramienta como MySQL Administrator. No cabe duda que se trata de un programa extremadamente útil e imprescindible para administrar visualmente servidores MySQL.

Muchas de las opciones de configuración de la base de datos son muy sencillas de entender y de utilizar, por lo menos todas las opciones más básicas. En conjunto con MySQL Query Browse, nos permite gestionar cualquier aspecto de una base de datos MySQL.

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en 26/01/2005
Disponible online en <http://desarrolloweb.com/articulos/34.php>

MySQL en sistemas Windows y conexión Access

En esta sección vamos a ver algunos detalles específicos para los usuarios de Windows. Además veremos cómo conectar MySQL desde Access, realizando tareas de administración de MySQL desde Access o migrando bases de datos que tengas en Access a MySQL.

Instalar el driver ODBC para conectar con una base de datos MySQL

Instrucciones para instalar el driver ODBC necesario para conectar con una base de datos MySQL desde un sistema Microsoft Windows.

Instalar este driver sirve para que desde un sistema Microsoft Windows se pueda acceder a una base de datos MySQL. Las aplicaciones son variadas, por ejemplo podemos utilizarlo para [crear un DSN](#) asociado a una base de datos MySQL, de modo que nuestras páginas ASP podrían acceder a dicha base de datos. Otra aplicación es acceder desde Access a la base de datos MySQL y exportar o importar datos (migrar los datos [desde Access a MySQL](#) y [desde MySQL a Access](#)), incluso para crear un [back-end de nuestra base MySQL en interfaz Access](#).

Primero hay que descargar la última versión de Myodbc de la página de Mysql:
<http://www.mysql.com/products/connector/odbc/>

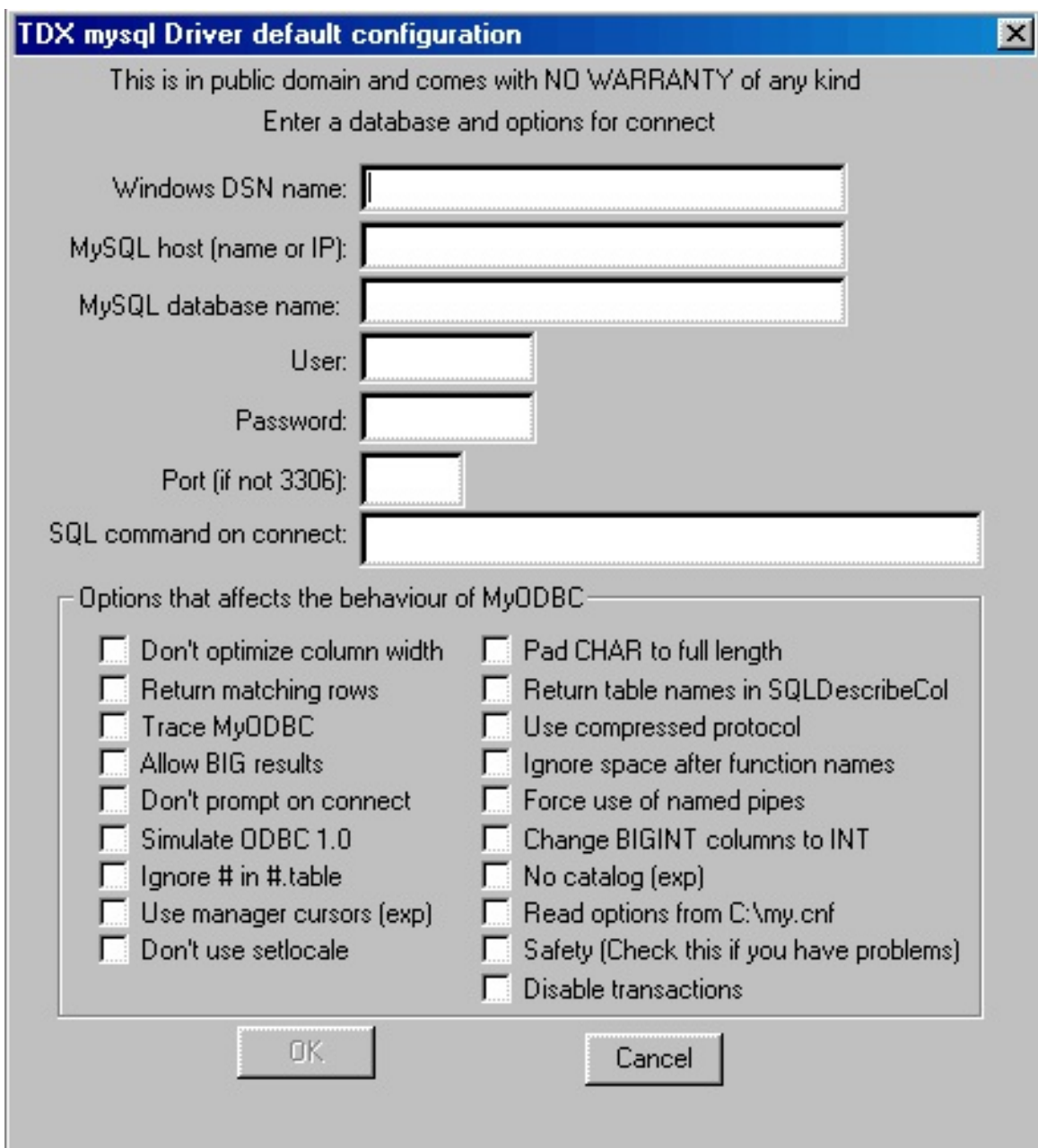
Nota: Puede que nuestro sistema tenga que actualizarse. En el ordenador que tiene el sistema Windows XX y Access 2000 habría que actualizar a la versión 6 de Microsoft Jet:

<http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q239114&>

Cuando ya tenemos todo, instalamos la actualización de Microsoft Jet, y descomprimos e instalamos el driver ODBC de Mysql. Cuando pregunta en la pantalla de “Data Sources” haz clic en “Close” para terminar.

Una vez se ha instalado el driver ODBC, accede al panel de control de ODBC de 32 Bits (Botón Inicio-> Configuración-> Panel de control-> Fuentes de datos ODBC 32 bits).

En este punto, tendrás que elegir si quieres utilizar el driver para un solo usuario (DSN de usuario), o para cualquier usuario del ordenador (DSN de Sistema). Una vez hayas elegido uno, haz clic en el botón de “Agregar” para añadir una nueva fuente de datos y a continuación, selecciona el driver de Mysql. Aparecerá la siguiente pantalla:



TDX mysql Driver default configuration

This is in public domain and comes with NO WARRANTY of any kind
Enter a database and options for connect

Windows DSN name:

MySQL host (name or IP):

MySQL database name:

User:

Password:

Port (if not 3306):

SQL command on connect:

Options that affects the behaviour of MyODBC

<input type="checkbox"/> Don't optimize column width	<input type="checkbox"/> Pad CHAR to full length
<input type="checkbox"/> Return matching rows	<input type="checkbox"/> Return table names in SQLDescribeCol
<input type="checkbox"/> Trace MyODBC	<input type="checkbox"/> Use compressed protocol
<input type="checkbox"/> Allow BIG results	<input type="checkbox"/> Ignore space after function names
<input type="checkbox"/> Don't prompt on connect	<input type="checkbox"/> Force use of named pipes
<input type="checkbox"/> Simulate ODBC 1.0	<input type="checkbox"/> Change BIGINT columns to INT
<input type="checkbox"/> Ignore # in #.table	<input type="checkbox"/> No catalog (exp)
<input type="checkbox"/> Use manager cursors (exp)	<input type="checkbox"/> Read options from C:\my.cnf
<input type="checkbox"/> Don't use setlocale	<input type="checkbox"/> Safety (Check this if you have problems)
	<input type="checkbox"/> Disable transactions

OK Cancel

En ella tendrás que rellenar los siguientes campos:

Windows DSN name:

Nombre de la fuente de datos que estará disponible desde Windows.

Mysql host (name or IP):

#####

Nombre o dirección IP del ordenador donde se encuentra instalado el servidor Mysql.

Mysql Database Name:

Nombre de la base de datos con la que se trabajará desde la fuente de datos

User:

Nombre de usuario con el que se accederá al servidor de bases de datos.

Password:

Contraseña del usuario.

Port:

Sirve para especificar el puerto en el que se encuentra el servidor Mysql, hay que poner un valor en caso de que no se esté utilizando el predeterminado, que es el 3306.

Una vez están estas opciones configuradas, se puede hacer clic en "OK" para cerrar las ventanas.

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en 20/09/2002
Disponible online en <http://desarrolloweb.com/articulos/34.php>

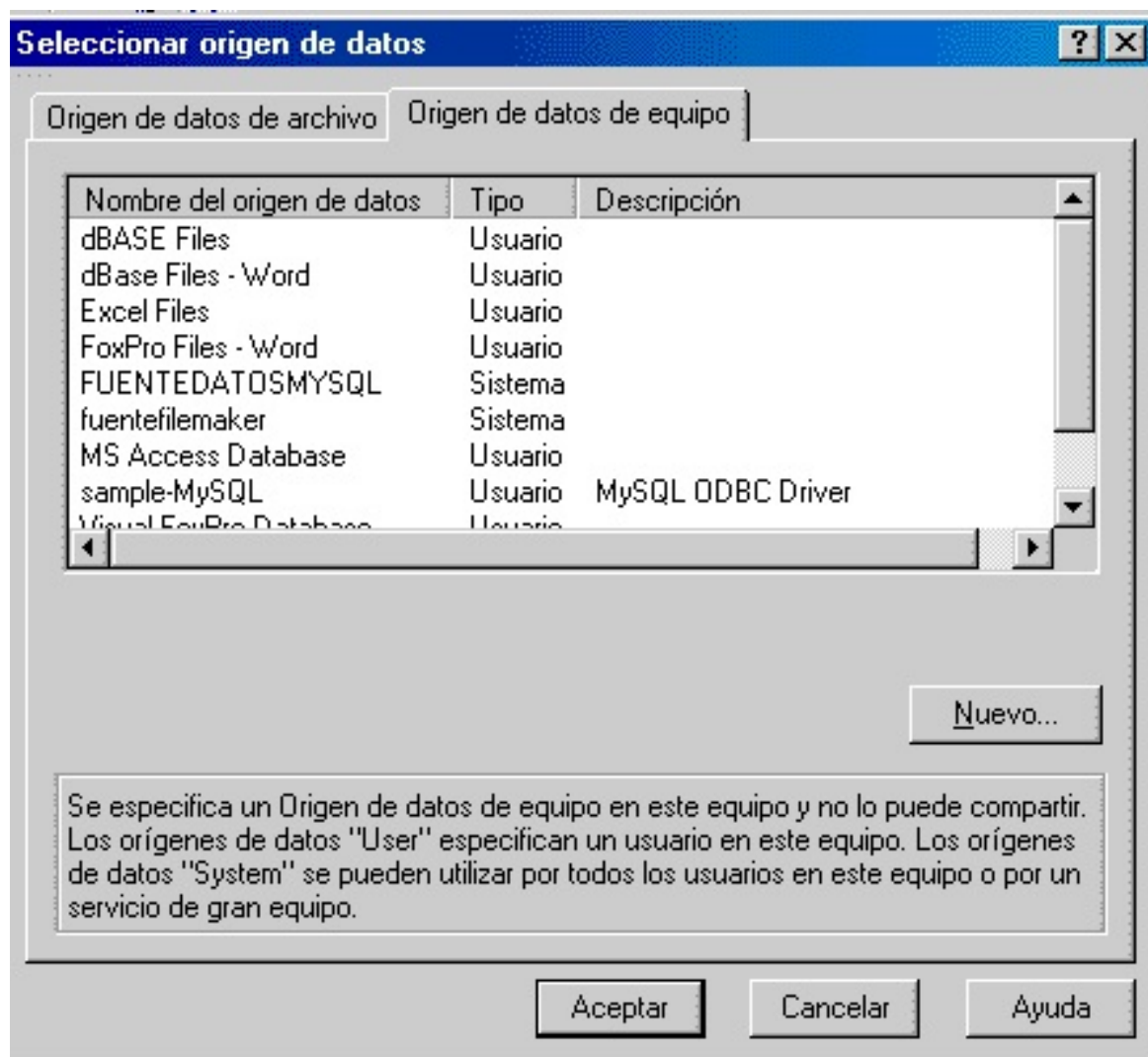
Exportar datos de MySQL a Microsoft Access 2000

Os explicamos cómo recuperar información almacenada en un servidor de datos MySQL hacia una base Access 2000.

Migrar datos de una base de datos a otra es algo a lo que muchos de nosotros hemos tenido que confrontarnos en algún momento. A continuación os explicamos cómo recuperar información almacenada en un servidor de datos Mysql hacia una base Access 2000.

Nota: Para realizar esta tarea es necesario que hayamos descargado el driver ODBC y lo hayamos instalado en nuestro sistema Windows. Esta labor se puede conocer en un artículo de DesarrolloWeb.com: [Instalar el driver ODBC para MySQL](#).

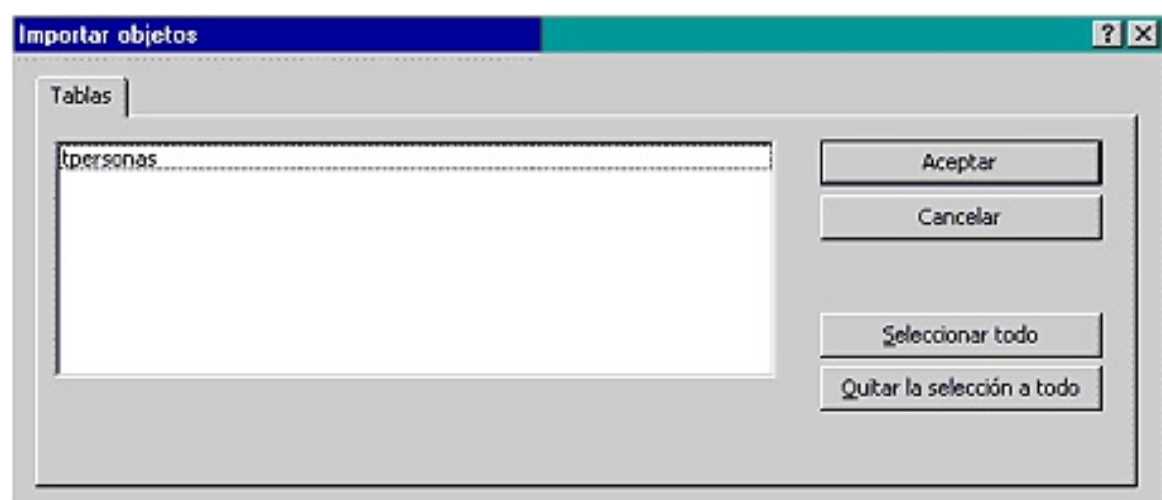
Para importar una tabla de Mysql a Microsoft Access, desde Access, y con la base de datos en la que se quieren importar los datos abierta, seleccionar el menu Archivo->Obtener datos Externos->Importar. En la pantalla de Importar datos, en la opción Tipo de archivo seleccionar ODBC databases().



Seleccionar origen de datos de equipo, y dentro de esta, el nombre de la fuente de datos que hemos creado anteriormente. Una vez la has seleccionado, y has hecho clic sobre "Aceptar", aparecerá la pantalla de configuración del driver por si deseas marcar para esta acción en concreto, algunas de las opciones de configuración que aparecen en el driver ODBC, si no deseas marcar ninguna, clic sobre "OK".

Nota: Pudiera ser en algún caso que los tipos de los datos de la base en los sistemas MySQL y Access no sean totalmente compatibles y se produzca alguna anomalía al exportarlos. Realmente es una posibilidad que pensamos, aunque en las pruebas que hemos realizado no hemos visto ningún tipo de problema, bien es cierto que los campos que hemos trabajado no eran muy raros.

Aparecerá una ventana donde pregunta qué tabla de Mysql se desea exportar a Access:



Selecciona la tabla , y haz clic sobre "Aceptar"

Nota: Si estamos exportando los datos hacia o desde un servidor de bases de datos alojado en algún proveedor de Hosting, tenemos que tener en cuenta que estos no siempre incluyen en su paquete básico el acceso remoto al servidor de base de datos, o requiere de un aviso explícito por parte del cliente para su configuración.

Referencia: Si deseamos realizar una migración de datos en el otro sentido, es decir, desde Access hacia MySQL, será muy indicado leer otro artículo en DesarrolloWeb que explica el proceso detalladamente. [Exportar datos de Access 2000 a MySQL.](#)

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en 24/08/2002
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Exportar datos de Access 2000 a MySQL

Explicamos una manera práctica de migrar datos de una base de datos Access a un servidor de datos MySQL.

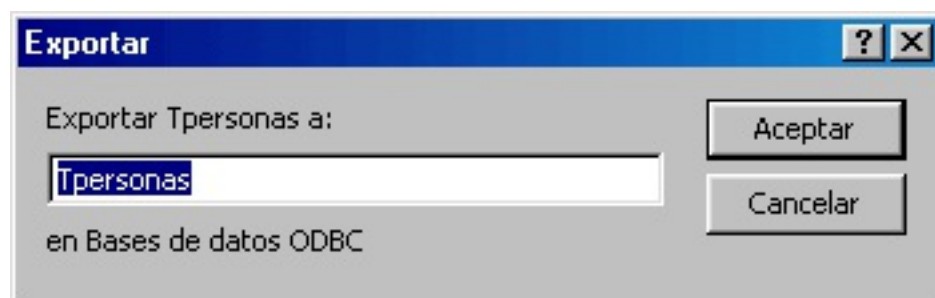
No es de extrañar que hayamos comenzado a hacer nuestros pinitos en la web sirviéndonos de una base de datos sencilla como Access. Tampoco es de extrañar que, llegado el momento, pasemos a cosas más serias y nos pasemos a un servidor de datos como MySQL. Aquí os mostramos una manera bastante práctica de migrar los datos de la una a la otra.

Nota: Para realizar esta tarea es necesario que hayamos descargado el driver ODBC y lo hayamos instalado en nuestro sistema Windows. Esta labor se puede conocer en un artículo

de DesarrolloWeb.com: [Instalar el driver ODBC para MySQL](#).

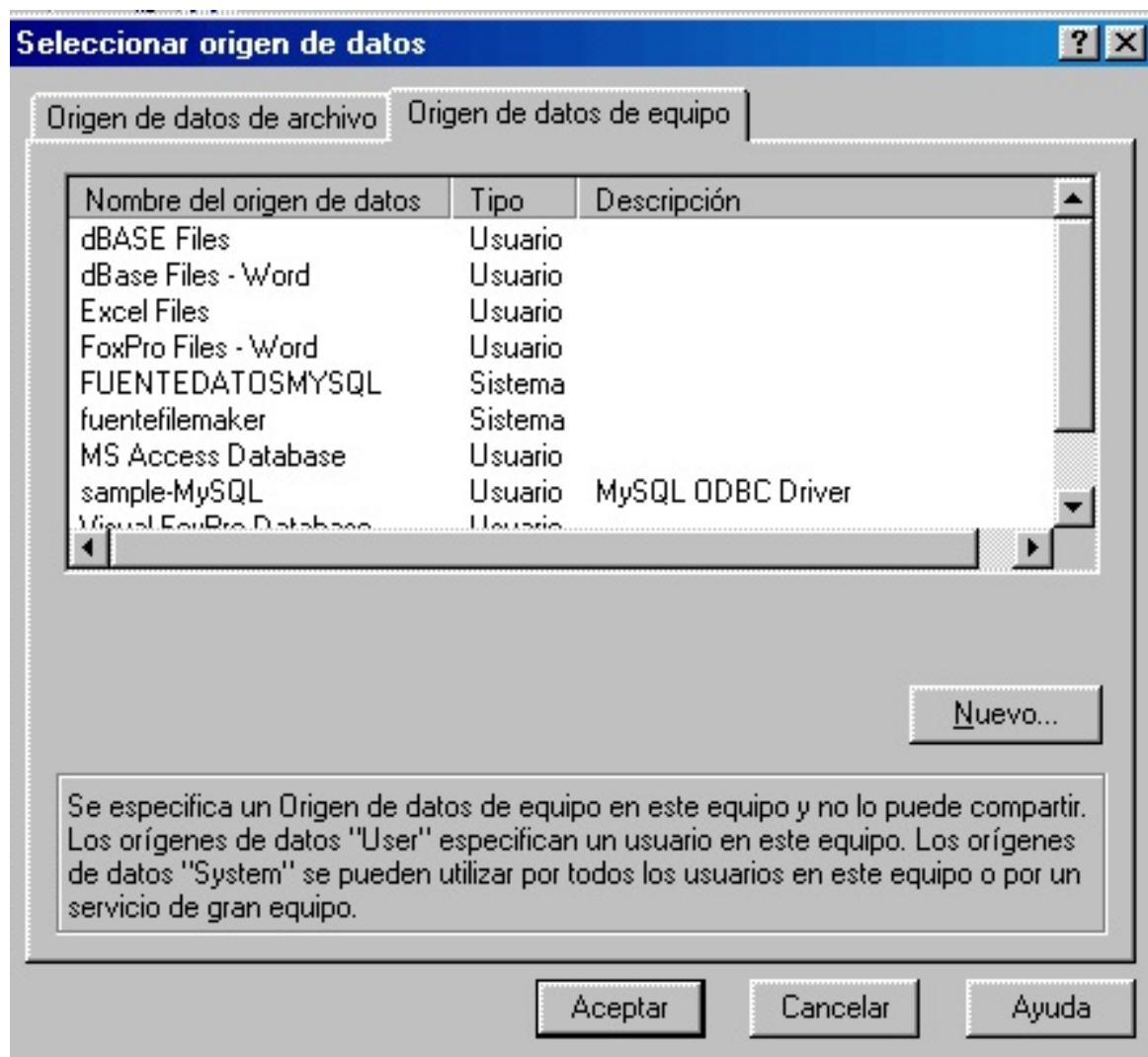
Para exportar una tabla a Mysql, hay que abrir la base de datos y seleccionar la tabla. Después, hacer clic sobre Archivo->Exportar. En la pantalla de exportar, en la opción Guardar como tipo, seleccionar ODBC databases().

Una vez se ha hecho esto, aparece una ventana que nos pregunta el nombre que le queremos dar a la tabla en Mysql, por defecto aparece el mismo.



Haz clic sobre "Aceptar", y aparecerá la pantalla en la que se pide que selecciones el origen de datos ODBC:

Nota: Pudiera ser en algún caso que los tipos de los datos de la base en los sistemas MySQL y Access no sean totalmente compatibles y se produzca alguna anomalía al exportarlos. Realmente es una posibilidad que pensamos, aunque en las pruebas que hemos realizado no hemos visto ningún tipo de problema, bien es cierto que los campos que hemos trabajado no eran muy raros.



Seleccionar origen de datos de equipo, y dentro de esta el nombre de la fuente de datos que hemos creado anteriormente. Una vez la has seleccionado y has hecho clic sobre "Aceptar", aparecerá la pantalla de configuración del driver por si deseas marcar para esta acción en concreto algunas de las opciones de configuración que aparecen en el driver ODBC. Si no deseas marcar ninguna, haz clic sobre "OK" y los datos comenzarán a exportarse.

Nota: Si estamos exportando los datos hacia o desde un servidor de bases de datos alojado en algún proveedor de Hosting, tenemos que tener en cuenta que estos no siempre incluyen en su paquete básico el acceso remoto al servidor de base de datos, o requiere de un aviso explícito por parte del cliente para su configuración.

Referencia: Si deseamos realizar una migración de datos en el otro sentido, es decir, desde MySQL hacia Access, será muy indicado leer otro artículo en DesarrolloWeb que explica el proceso detalladamente. [Exportar datos de MySQL a Microsoft Access 2000.](#)

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en 28/08/2002
Disponible online en <http://desarrolloweb.com/articulos/34.php>

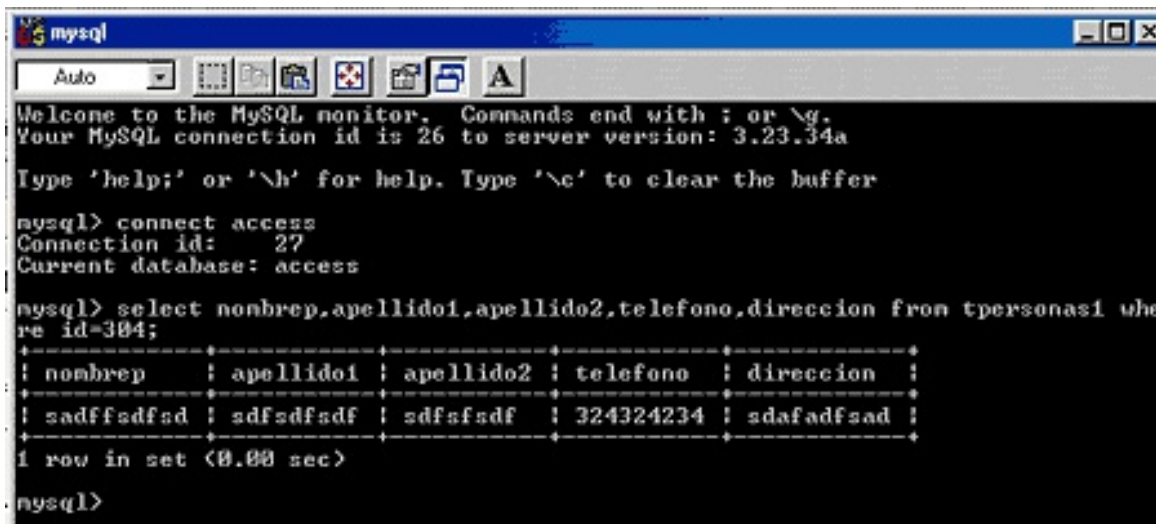
Administrar y actualizar una base de datos MySQL desde Access

Si nuestras páginas utilizan una base de datos debemos contar con una herramienta de administración y actualización de la base de datos, o programarla nosotros. Ilustramos como Access puede ayudarnos a hacerlo fácil.

Uno de los mayores problemas de los que adolece actualmente Mysql es el no poseer un entorno gráfico que satisfaga a la mayor parte de los usuarios. Existen magníficos proyectos a través de página Web, como PHPmysqlAdmin, pero muchas veces te encuentras con gente que le gustaría tener algo parecido a un Access, posibilidad de incluir formularios para la entrada de datos, o de informes sobre los datos almacenados dentro de una tabla.

Dado que no existe hoy en día ninguna herramienta parecida al Access para trabajar con Mysql, el presente artículo expone la forma de trabajar con las bases de datos de Mysql utilizando el entorno gráfico de Access. Al terminarlo, podrás utilizar los formularios, consultas e informes de Access con los datos de los ficheros de Mysql.

De esta forma, el encargado de actualizar los datos de una página, podrá trabajar desde Access, con la comodidad de los formularios, los menús desplegables etc., y los datos serán enviados automáticamente a Mysql.



```
mysql
Auto
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26 to server version: 3.23.34a

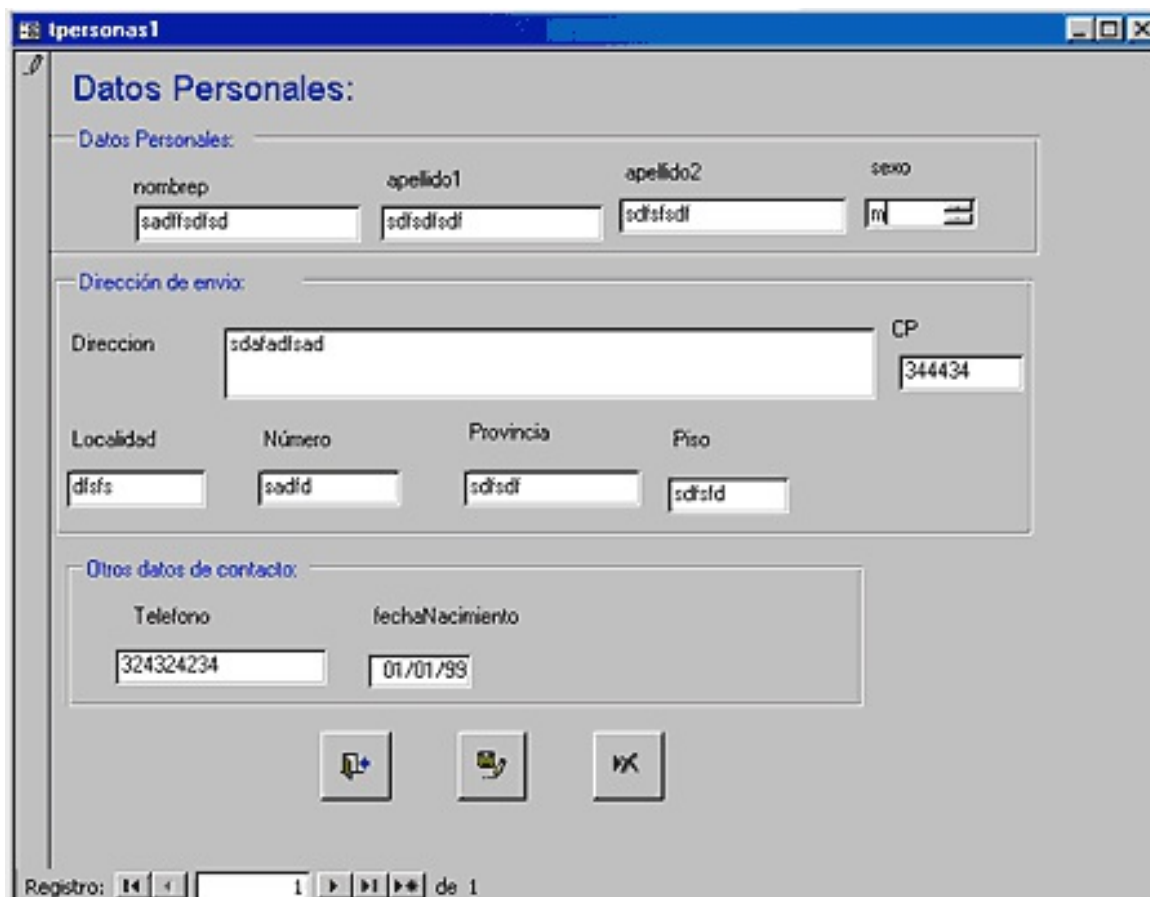
Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> connect access
Connection id: 27
Current database: access

mysql> select nombrep,apellido1,apellido2,telefono,direccion from tpersonas1 where id=384;
+-----+-----+-----+-----+-----+
| nombrep | apellido1 | apellido2 | telefono | direccion |
+-----+-----+-----+-----+-----+
| sadffsdfsd | sdfsd fsdf | sdfsd fsdf | 324324234 | sadafadsad |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Imagen de la consola de MySQL



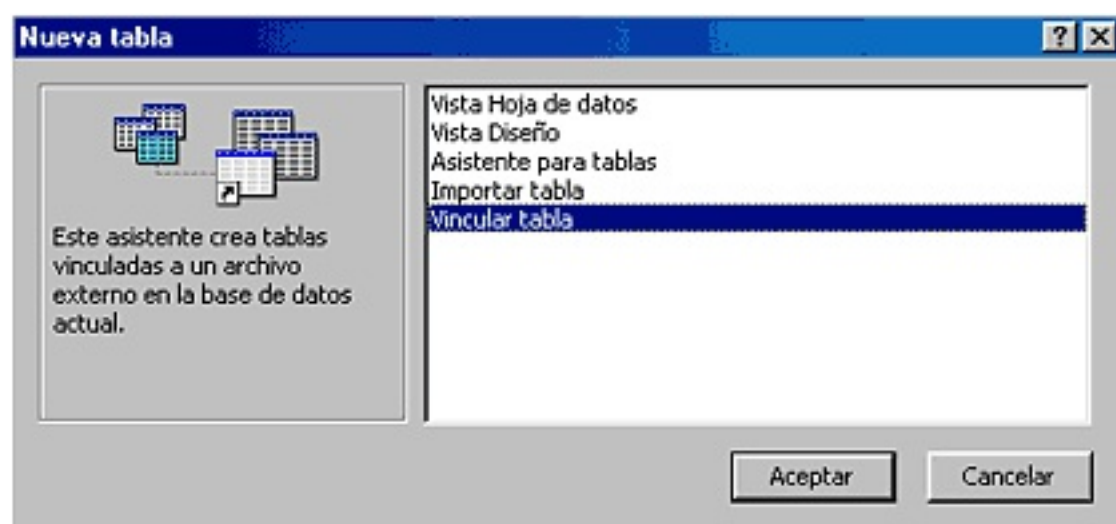
Formulario de Access que accede a una base de datos MySQL

Nota: Dos imágenes que se diferencian un mundo. Con las dos se puede administrar una base de datos MySQL, una corresponde a la consola MySQL (la herramienta más básica) y la otra a un formulario con Access, que sin duda ofrece una interfaz mucho más agradable.

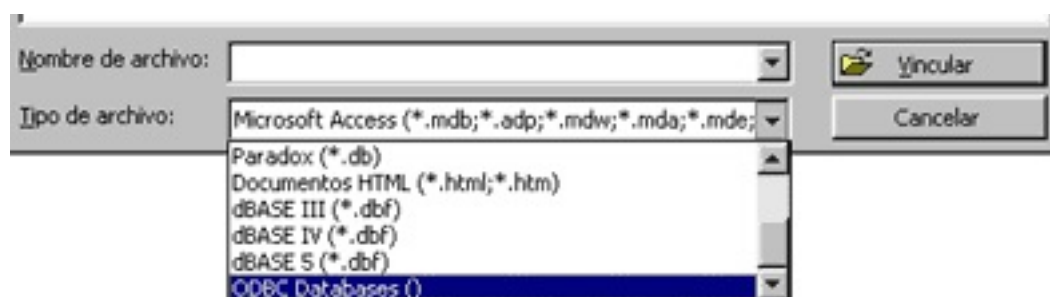
Además, para la generación de formularios e informes se pueden utilizar los asistentes... todo un ahorro de tiempo!!

Referencia: Para realizar esta tarea es necesario que hayamos descargado el driver ODBC y lo hayamos instalado en nuestro sistema Windows. Esta labor se puede conocer en un artículo de DesarrolloWeb.com: [Instalar el driver ODBC para MySQL](#).

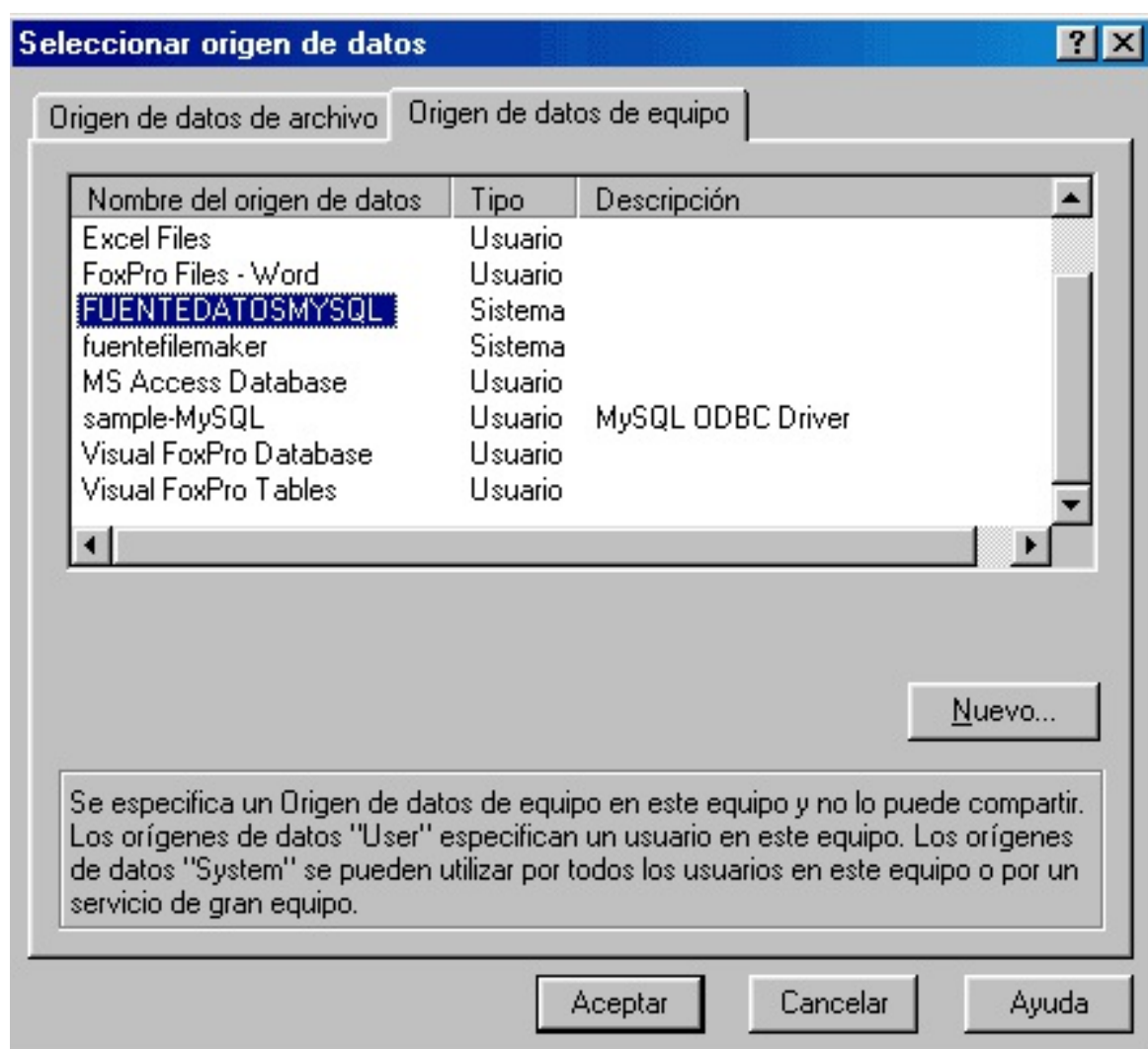
Una vez tenemos el driver MyODBC instalado, lo primero que hay que hacer es crear una base de datos en blanco desde la cual se vincularán las tablas. Una vez creada, se hace clic sobre la opción de crear nueva tabla. Aparecerá la siguiente ventana en la que se seleccionará crear nueva tabla vinculada:



Aparece la ventana de vincular una tabla de una base de datos, en la parte inferior se selecciona en tipo de archivo: **fuentes de datos ODBC()**

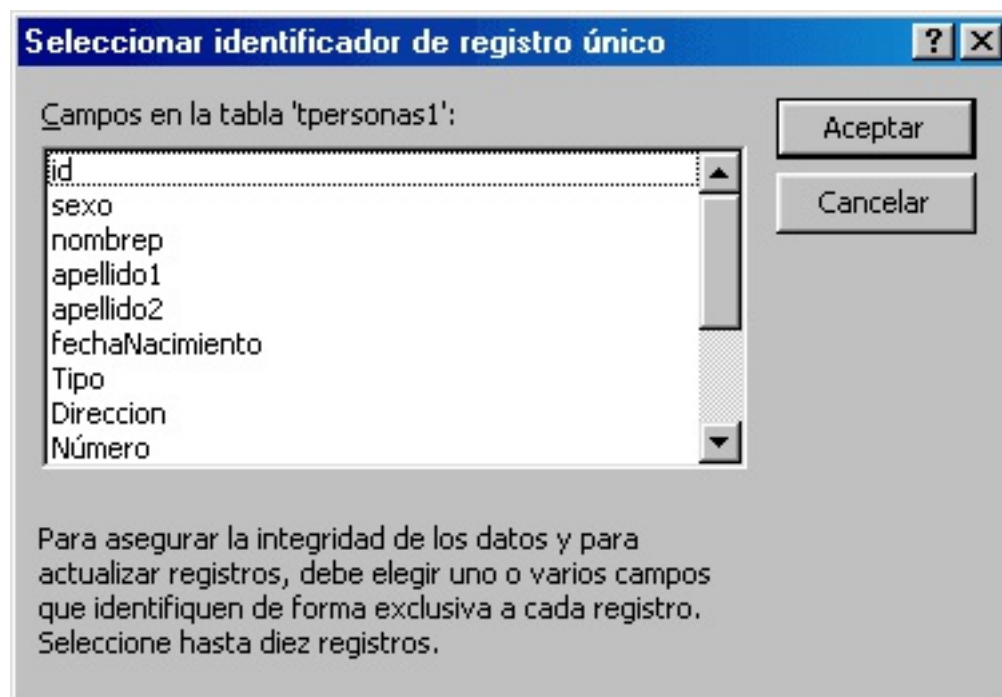


Al hacer clic sobre Vincular, aparece la ventana para seleccionar un Origen de datos, se selecciona dentro de la pestaña de fuentes de datos del Equipo, la fuente de datos que creamos en la primera parte del artículo:



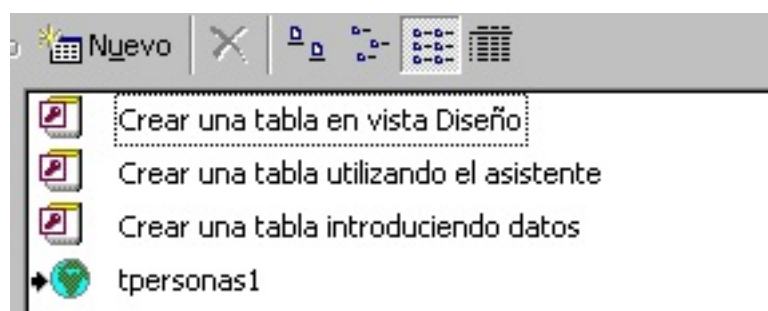
Una vez se selecciona, se hace clic sobre Aceptar, y aparece la ventana de configuración de la fuente de datos ODBC de Mysql. Como ya esta configurada, hacemos clic sobre OK, y aparecerá la ventana en la que se pueden elegir entre las tablas que contiene la base de datos para la cual hemos configurado la fuente de datos ODBC.

Se selecciona una tabla, y a continuación aparecerá una ventana donde deberemos especificar, hasta un máximo de diez, los campos que forman parte de la clave.



Se seleccionan los campos, y se hace clic sobre aceptar.

Aparecerá una nueva tabla dentro de la base de datos de Access. A diferencia del resto de tablas, esta no existe físicamente dentro del fichero de Access, sino que todas las modificaciones que se realizan sobre la misma, se envían a través de ODBC al fichero MYSQL.



A partir de ahora, se podrán crear formularios, consultas e informes sobre esta tabla tal y como se hace normalmente desde Access.

Nota: Algunos proveedores de Hosting no incluyen en su paquete básico el acceso remoto al servidor de base de datos, o requiere de un aviso explícito por parte del cliente para su configuración.

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en 20/10/2002
Disponible online en <http://desarrolloweb.com/articulos/34.php>

Más tutoriales de MySQL

En esta sección del taller de MySQL veremos otra serie de artículos relacionados con este sistema gestor de bases de datos que pueden venir puntualmente bien para algunos desarrolladores.

Habilitando InnoDB en MySQL

Este manual va para los que son usuarios del Mysql que empiezan, en especial para los que por facilidad utilizan el AppServ en cualquier versión.

Empezare por mi problema cuando leía un manual sobre integridad referencial, pues necesitaba utilizar bases de datos del tipo INNOBD, o en todo caso del tipo BDB (Berkeley Data Base), pues al momento de crear la tabla del siguiente modo:

```
CREATE TABLE Prueba
(
  id int not null auto_increment,
  nombre varchar(100),
  primary key(id)
) ENGINE = InnoDB;
```

Pues por defecto las tablas en Mysql antes de la versión 5.0.2, están configuradas para el Tipo MyISAM, entonces para que puedan ser InnoDB debemos declara de la forma ENGINE = InnoDB, como en el ejemplo, pero el problema en especial en los que instalan los AppServ es que la configuración en el **my.ini** que se encuentra en la Carpeta **C:/Windows** se le agrega la siguiente línea:

```
skip-innodb(*)
```

Lo que hace esta línea es que no permite crear tablas en el formato InnoDB, lo que no nos permite trabajar con integridad referencial, a diferencia de las versiones posteriores que agregan en el **my.ini**, que se encuentra en **C:/Archivos de Programas/MySQL/MySQL Server 5.0** en especial en los instaladores independientes las siguientes líneas:

```
# The default storage engine that will be used when create new tables when default-storage-engine=INNODB(**)
```

Estas a diferencia de las anteriores lo que permite es que todas las tablas que se crean sean del Tipo InnoDB por defecto, claro esta que las tablas del sistema las que vienen en el Mysql, deben ser del Tipo MyISAM ya que estas trabajan en este tipo, explicado las instalaciones vayamos a las soluciones.

Bueno la más fácil sería cambiar la primera línea (*) por la segunda ()**, pero que pasaría si deseamos crear tablas del Tipo MyISAM, pues tendríamos que declarar al crear la tabla de la forma siguiente:

```
CREATE TABLE Prueba
(
  id int not null auto_increment,
  nombre varchar(100),
  primary key(id)
) ENGINE = MyISAM;
```

Esto tal vez les genere un problema para los que generalmente crean sus tablas en el tipo MyISAM, entonces en el archivo **my.ini** se debe solamente borrar la línea de Skip-innoDB o comentarla para que nos permita crear tablas del tipo InnoDB, y el problemita ya está solucionado.

Antes de terminar como nos aseguramos que tenemos soporte para el tipo InnoDB, pues simplemente ejecutamos la siguiente sentencia:

```
SHOW VARIABLES LIKE '%innodb%';
```

La que mostrara entre otras cosas lo siguiente:

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_innodb   | YES   |
... .
```

Esta nos permite saber si está o no habilitado esta característica y así poder solucionar el problema si es que lo tenemos.

Ahora ya está solucionado, a trabajar con las tablas InnoDB, así poder hacer Integridad referencial, la que nos permite usar Primary Key, o Foreign Key. Bueno para terminar disculpas por la falta de orden y por no ser tan claro, espero les sirva de algo este manual.

Este artículo es obra de *Russvell Oblitas Valenzuela*
Fue publicado por primera vez en 02/08/2007
Disponible online en <http://desarrolloweb.com/articulos/habilitando-innodb-en-mysql.html>

Contar caracteres con MySQL

Hacemos uso de las funciones para contar caracteres en MySQL. Varios ejemplos

de sentencia SQL que cuenta caracteres de strings.

En este taller de MySQL vamos a hacer uso de las funciones del lenguaje SQL para contar caracteres de un campo guardado en una tabla. Es decir, vamos a seleccionar diversos registros de una tabla y vamos a contar los caracteres de uno de sus campos.

En el juego de funciones de MySQL para string (cadenas de caracteres) tenemos varias funciones para contar caracteres:

```
LENGTH(cadena)
```

Recibe una cadena y cuenta y devuelve el número de caracteres.

```
CHARACTER_LENGTH(cadena)
```

Recibe también una cadena y devuelve el número de caracteres contados.

```
CHAR_LENGTH(cadena)
```

Es un sinónimo de CHARACTER_LENGTH. Aunque no lo tienen todas las versiones de MySQL.

La diferencia entre LENGTH y CHARACTER_LENGTH es que en CHARACTER_LENGTH un carácter "multibyte" cuenta como un solo carácter. En LENGTH cuenta el número de bytes de la cadena. Así que en el caso de tener una cadena con 5 caracteres que ocupan 2 bytes cada uno, LENGTH devolvería 10 y CHARACTER_LENGTH sólo 5.

Por ahora vamos a utilizar CHARACTER_LENGTH, que parece que nos devolverá los resultados que esperamos obtener, el número de caracteres de la cadena, independientemente que la codificación pueda ser multi-byte.

```
SELECT character_length( 'un texto' )
```

Esta sentencia devolvería el valor 8, que es el número de caracteres que tiene el la cadena "un texto".

Pero ahora pongamos que queremos sacar todos los países de una tabla, junto con el número de caracteres que tiene cada una de las cadenas del nombre del país.

```
SELECT CHARACTER_LENGTH(nombre_pais), nombre_pais FROM pais
```

Esto nos devolvería un conjunto de registros con el número de caracteres de cada nombre del país y luego otro campo con el nombre del país.

```
SELECT nombre_pais FROM pais where CHARACTER_LENGTH(nombre_pais)=6
```

Esto nos devuelve los nombres de país que tienen 6 caracteres.

```
SELECT CHARACTER_LENGTH(nombre_pais) as 'numcaracteres', nombre_pais FROM pais ORDER BY CHARACTER_LENGTH(nombre_pais)
```

Esta última sentencia devuelve un conjunto de registros donde el primer campo, que hemos dado el nombre de "numcaracteres" para luego referirnos a él, muestra el número de caracteres y el segundo campo el nombre del país. Luego está ordenado por la longitud de la cadena, contada también por sus caracteres.

Este artículo es obra de *Miguel Angel Alvarez*
Fue publicado por primera vez en 04/01/2008
Disponible online en <http://desarrolloweb.com/articulos/contar-caracteres-con-mysql.html>

¿Porqué MySQL es lento con grandes tablas?

Si has leído y estudiado suficiente acerca de MySQL probablemente habrás escuchado que MYSQL no es la elección acertada para manejar tablas con mas de 1.000.000 de registros.

Por otro lado tambien puede que tengas conocimiento que [MySQL](#) es el motor de compañías como Google, Yahoo, Technorati y estas manejan algunos billones de registros y consiguen un gran rendimiento.

La pregunta que puedes estar haciendote es cuál es la razón...

La razón es que normalmente estas tablas estan diseñadas y entendidas para trabajar con [MySQL](#), si diseñas tus datos considerando que puede hacer y que no puede hacer [MySQL](#) probablemente conseguiras un buen rendimiento, cualquier sistema de administración de bases de datos es diferente con respecto a los otros, lo que funciona bien y es eficaz en [Oracle](#), MS SQL, [PostgreSQL](#) no debe de ser lo que mayor rendimiento ofrece en MYSQL.

Incluso en el sistema de almacenamiento tienen muchas diferencias que los hace diferentes.

Las tres claves que deberias de tener en cuenta con tablas muy grandes son: Buffers, indices y consultas.

Buffers

Como ya sabrás un **buffer** es una ubicación de la memoria reservada para el almacenamiento temporal de información digital.

La primera cosa que deberías tener muy clara es el hecho de que hay una gran diferencia entre cuando los datos están en memoria y cuando no están en memoria.

Si empezaste con un tamaño de memoria y notas un descenso gradual del rendimiento porque la base de datos está creciendo una buena solución sería asegurarte que tienes suficiente memoria para el volumen de datos que estás utilizando esto podrías realizarlo con diferentes técnicas.

Indices

Los índices son usados para encontrar rápidamente los registros que tengan un determinado valor en alguna de sus columnas. Sin un índice, MySQL tiene que iniciar con el primer registro y leer a través de toda la tabla para encontrar los registros relevantes.

Aún en tablas pequeñas, de unos 1000 registros, es por lo menos 100 veces más rápido leer los datos usando un índice, que haciendo una lectura secuencial por lo tanto queda claro que los índices son realmente eficaces para acelerar el acceso a datos.

Antes de proseguir con la explicación he de aclarar algunos términos como escaneo completo, que es leer todos los registros de la tabla de manera secuencial.

Cuando [MySQL](#) encuentre que hay un índice en una columna, lo usará en vez de hacer un escaneo completo de la tabla. Esto reduce de manera importante los tiempos de CPU y las operaciones de entrada/salida en disco.

Dejame explicártelo con datos, considerando una tabla que tiene 100 Bytes por registros, con una unidad [SCSI](#) nosotros podríamos obtener 100MB/segundo de velocidad de lectura que nos daría alrededor de 1.000.000 de registro por segundo, si hablamos de tablas tipo MyISAM.

Aquí te dejo un ejemplo, he creado una tabla con 30 millones de registros, con una columna ('val') que tiene 10000 valores diferentes, vamos a ver las diferencias entre el tiempo de un escaneo total de la tabla y un escaneo utilizando un rango de valores por ejemplo entre 1 y 100, el resultado es el siguiente:

```
1.
mysql> SELECT count(pad) FROM large;
2.
+-----+
3.
| count(pad) |
4.
+-----+
5.
| 31457280 |
6.
+-----+
7.
1 row IN SET (4 min 58.63 sec)
8.
```



```

9.
mysql> SELECT count(pad) FROM large WHERE val BETWEEN 1 AND 100;
10.
+-----+
11.
| count(pad) |
12.
+-----+
13.
| 314008 |
14.
+-----+
15.
1 row IN SET (29 min 53.01 sec)

```

Como puedes observar el resultado es contradictorio ha tardado 5 minutos en contar 30 millones de registros pero lo que sorprende es que ha tardado 30 minutos en contar el 1% de esos registros que eran los valores entre 1 y 100.

Hay muchas maneras de optimizar el trabajo con este tipo de consultas, por ejemplo ordenar primero el los valores y entonces hacer la búsqueda sobre estos registros ya ordenados. Esta solución reduciría el tiempo de ejecución de la consulta pero sin duda sería un resultado relativamente parecido. El uso de índices se antoja esencial en este tipo de ejemplos en los que tenemos grandes cantidades de datos en nuestras tablas, la mejora en la obtención de los datos puede ser muy significativa.

1. Indexe las columnas sobre las que realiza la búsqueda, no sobre las que selecciona

Las mejores columnas para indexar son en las que aparecen la cláusula WHERE o las nombradas en las cláusulas join.

2. Utilice índices únicos

Los índices trabajan mejor sobre columnas con valores único, y peor con aquellas que tiene muchos valores duplicados. Por ejemplo, si una columna contiene fechas y tiene varios valores diferentes, un índice diferenciará las filas fácilmente, sin embargo, no le ayudará tanto si se utiliza en una columna para registrar el valores como verdadero, falso y que contiene sólo los dos valores "V" y "F" (cualquiera que sea que busque, tomará casi la mitad de las filas).

3. Utilice índices cortos

Si va a indexar una columna de cadenas, especifique una longitud prefijada, siempre que sea razonable hacerlo así, por ejemplo si tiene una columna CHAR (200), no indexe la columna entera si la mayor parte de los valores son únicos dentro de los 10 o 20 primeros caracteres. Indexar estos 10 o 20 primeros caracteres le ahorrara mucho espacio en el índice, y probablemente hará más rápidas su consultas.

4. No abuse de los índices

Los índices deben ser actualizados, y posiblemente reorganizados, cuando modifique los contenidos de sus tablas.

Si tiene un índice que raramente, o nunca, se usa estas hastas realentizando el sistema volviendo más lenta de forma innecesaria las modificaciones de la tabla.

5.Considere el tipo de comparaciones a realizar en un registro

Los índices se usan para operaciones <, <="," =,""">=, > y BETWEEN. Tambien se usan par operaciones LIKE, cuando el patrón tiene un prefijo literal. Si sólo usa una columna para otro tipo de operaciones, tales como STRCMP(), no tiene sentido indexarlo.

Conclusión

Como conclusión tu puedes conseguir que MySQL rinda a buen rendimiento con grandes cantidades de datos pero para ello debes tener en cuenta sus limitaciones y saber cuales son las características que ofrecen mejor rendimiento.

[Fuente: en ingles](#)

Este artículo es obra de *Manu Gutierrez*
Fue publicado por primera vez en 08/11/2007
Disponible online en <http://desarrolloweb.com/articulos/porque-mysql-es-lento-con-grandes-tablas.html>

Connector/J

El driver Connector/J permite trabajar desde Java con bases de datos MySQL.

Mysql Connector es un driver creado por Mysql AB que te permitirá trabajar con Mysql desde programas escritos en Java. A diferencia de otros drivers, este es de libre distribución, y tiene un buen rendimiento.

MySQL Connector/J es un driver nativo de Java que convierte las llamadas generadas por JDBC en el protocolo de red que utiliza la base de datos de Mysql. Permite al desarrollador trabajar con el lenguaje de programación Java y de esta forma construir programas que interactuan con Mysql.

El MySQL Connector/J es un driver JDBC tipo IV y contiene todas las características de JDBC para manejar Mysql.

En el desarrollo de las últimas versiones se ha incrementado bastante la velocidad del driver, ganando en rapidez así como en eficiencia. El driver soporta resultados de datos "streaming" lo que permite al usuario recoger un gran número de filas sin la necesidad de utilizar un buffer de memoria. El driver implementa un protocolo de paquetes grande que permite enviar filas y campos BLOBs de hasta 2 GigaBytes. En la versión de desarrollo del mismo se implementan

las nuevas características añadidas en el API 3.0 de JDBC.

El driver ha sido probado para trabajar con las siguientes herramientas de bases de datos y servidor de aplicaciones:

- Apache
- Jboss
- BEA Weblogic
- IBM VisualAge for Java
- IBM WebSphere Application Server 4.0
- Forte for Java
- Oracle SQL/J
- TableGen

Descarga

Para instalar el driver hay que descargarlo de la página de Mysql [Connector/J](#)

Este artículo es obra de *Carlos Luis Cuenca*
Fue publicado por primera vez en 20/09/2002
Disponible online en <http://desarrolloweb.com/articulos/34.php>