

Curso de MySQL y Java

Introducción: instalación, arranque y primeros pasos

1. Material

En este curso vamos a trabajar con el servidor mysql versión 4.0.18. También utilizaremos una herramienta llamada 'control center' que permite realizar gran parte de las tareas de gestión de las BDs creadas en mysql.

- Servidor y cliente de mysql
- Herramienta MySQL control-center

Este software puede descargarse gratuitamente de la siguiente página web:
<http://www.mysql.com/>

Un manual muy completo online sobre mysql se encuentra en la siguiente página web:
<http://dev.mysql.com/doc/mysql/en/index.html>

2. Instalación

(En windows la instalación con los ficheros binarios es muy simple. Basta ejecutar el fichero setup.exe y seguir las instrucciones del instalador)

2. Arranque, configuración y parada del servidor

Todos los ficheros ejecutables se encuentran en directorio \$HOME/bin (ejemplo: d:\mysql\bin).

Existen varios servidores dependiendo del sistema operativo:

- mysqld-nt (para XP-NT y 2000)
- mysqld-max-nt
- mysqld (este es el general)

Para arrancar el servidor se debe ejecutar (por ejemplo en 2000):

```
shell>> mysqld-nt
```

La configuración de la BD se guarda en un fichero **my.ini**. Existe una herramienta que permite modificar dicha configuración **winmysqladmin.exe**. Ejemplo:

```
#This File was made using the WinMySQLAdmin 1.4 Tool
#04/06/2004 12:47:28
#Uncomment or Add only the keys that you know how works.
#Read the MySQL Manual for instructions

[mysqld] #Indica que las opciones siguientes son para el servidor:
basedir=C:/mysql
#bind-address=147.156.27.115
datadir=C:/mysql/data
#language=C:/mysql/share/your language directory
#tmpdir#=
log_bin
```

Para crear un servicio asociado al servidor de mysql se debe ejecutar lo siguiente:

```
shell>> mysqld-nt --install
```

Así se puede conseguir que mysql arranque en el arranque del sistema operativo (inicio automático) .

Si se quiere parar la BD:

- Se para el servicio (desde el panel de control de Windows) o,
- Se utiliza la herramienta (winmysqladmin.exe) o,
- mediante la herramienta mysqladmin

```
shell>> mysqladmin -u root -p shutdown
```

3. Conexión con el servidor

Una vez instalado el servidor de mysql estamos en condiciones de conectarnos a él y empezar a trabajar. Para conectarnos al servidor de mysql se utiliza el cliente mysql (se encuentra en el directorio bin del directorio donde se ha instalado mysql).

```
Shell> mysql -u root -p
```

Por defecto la BD tiene un usuario para conectarse (root) que tiene todos los privilegios para poder hacer cualquier cosa sobre la BD. Este usuario de administrador puede conectarse a la BD sin necesidad de un password después de la instalación. Es MUY IMPORTANTE tan pronto se instale la BD añadir un password a este usuario.

```
Shell> set password for 'user'@'ordenador'=PASSWORD('nuevo');
```

Una vez conectados con el servidor mysql, podemos realizar consultas en SQL. Para hacer una prueba escribe lo siguiente que devuelve la versión de mysql y la fecha actual:

```
mysql> select version(),current_date;
```

Esta consulta ilustra varias cosas sobre mysql:

- Un comando normalmente consiste en una sentencia sql terminada con un ';'.
- Cuando se escribe un comando, mysql lo manda al servidor para su ejecución y muestra los resultados. Se muestra otro prompt para saber que está listo para otra consulta
- Mysql muestra los resultados en forma de tabla, en la que la primera fila contiene las etiquetas para cada columna.
- Mysql devuelve el número de filas recuperadas así como el tiempo que necesito para su ejecución.

Las palabras claves pueden introducirse en minúsculas o mayúsculas (no es sensible a tipo de letras).

En el siguiente ejemplo se muestra como mysql se puede utilizar como si se tratara de una calculadora:

```
mysql> select sin(PI()/4), (4+1)*5;
```

Un comando puede ocupar una o varias líneas. El final se delimita con un ‘;’

```
mysql> select  
->user(),  
->current_date;
```

Si quieres cancelar un commando que está en proceso de introducción se puede cancelar tecleando ‘\c’.

Creación de BDs, tablas y usuarios en MySQL

1. Cómo crear y utilizar una BD en MySQL

En esta sección vamos a ver como se crean BDs, tablas, como se cargan datos en una tabla y como se recuperan los datos.

Para ver que BDs existen actualmente en el servidor:

```
mysql> show databases;
```

Cuando se ejecuta este comando aparece una lista de las BDs existentes. Por defecto existen dos BDs ya creadas: mysql y test.

Es posible que no veas todas las BDs existentes (solo para las que tienes privilegio).

Para acceder a una BD concreta se debe ejecutar el siguiente comando:

```
mysql> use test
```

Este comando hace que la BD accesible sea la BD con nombre 'test'. Destacar que no es necesario el ';' para este comando.

Para crear una BD, se necesita utilizar el siguiente comando:

```
mysql> CREATE DATABASE mibd;
```

Para acceder a esta BD es necesario utilizar el comando 'USE' visto anteriormente explícitamente.

Cada vez que se accede a mysql es necesario indicar que BD se va a utilizar. Para ello se puede utilizar el comando anterior o indicarlo cuando se invoca a mysql de la siguiente forma:

```
shell> mysql -h host -u user -p mibd
```

Date cuenta que si quieres añadir tu password en el comando anterior deberías escribir el password seguido a la opción '-p' (sin espacios):

```
shell> mysql -h host -u user -pmiclave mibd
```

En este momento hemos creado una BD, pero una BD vacía. Ahora debemos crear las tablas que contendrán nuestros datos. Para ver que tablas existen en una BD concreta se puede utilizar el siguiente comando:

```
mysql> show tables
```

Si se quiere crear una tabla con una estructura determinada (nombre de los campos y el tipo de dato para cada uno de ellos) se utiliza la siguiente sentencia:

```
mysql> CREATE TABLE mitabla(name VARCHAR(20), propietario VARCHAR(20), fecha DATE);
```

Una vez creada la tabla se puede visualizar la estructura mediante la siguiente instrucción:

```
mysql> describe mitabla;
```

Después de la creación de la tabla, es cuando podemos insertar datos en ellas. Una manera sencilla de cargar datos en una tabla es a través de un fichero de texto. Este fichero debe contener cada registro de la tabla en una fila diferente en el fichero ASCII. Los datos nulos se especifican con un '\N'. Un ejemplo de esto se muestra a continuación:

Name	propietario	fecha
Elnombre	\N	1998-12-28

Si este fichero tiene el nombre de 'datos.txt' se podría cargar en la tabla 'mitabla' de la siguiente manera:

```
mysql> LOAD DATA LOCAL INFILE "datos.txt" INTO TABLE mitabla;
```

Otra manera de introducir datos en una tabla es con la sentencia de INSERT de MYSQL.

```
mysql> INSERT INTO mitabla VALUES('minombre',NULL,'1998-07-24');
```

2. Descripción del diccionario de datos en MySQL

En la BD mysql existen dos bases de datos ya creadas en la instalación: test y mysql. Esta última es muy importante para el funcionamiento del sistema de gestión de base de datos. Corresponde a lo que se conoce como diccionario de datos de una BD (relaciones que sirven para el correcto funcionamiento del SGBD). Veamos estas tablas:

Nombre tabla
User
Host
Db
tables_priv
columns_priv
Fun.

Estas tablas contienen la información sobre los privilegios de un determinado usuario sobre una BD, cuando se conecta de un determinado ordenador y quiere acceder a ciertas tablas o columnas de una tabla.

En la tabla user se almacena los usuarios autorizados junto con sus *password* (encriptados) para conectarse desde un determinado host. Se puede tener diferentes privilegios para el mismo usuario si se conecta de diferentes máquinas (incluso se puede tener diferente password).

Veamos un ejemplo:

User	host	password	select	delete
Root	localhost		'y'	'y'
Root	%		'y'	'y'
	localhost			
	%			

Este significa que existe un usuario 'root' que se puede conectar desde 'localhost' sin password (MUY PELIGROSO) y con todos los privilegios existentes otorgados. La segunda entrada de esta tabla indica que este mismo usuario se puede conectar desde cualquier ordenador (comodín '%'). Existe un usuario anónimo (en blanco) que se puede conectar desde cualquier host.

Como se puede ver, estos usuarios no tienen todavía password asignado (están en blanco). Para asignar o modificar un password se utiliza la siguiente sentencia:

```
mysql> SET PASSWORD FOR root@localhost=PASSWORD('miclave');
```

Otra forma es mediante la utilización de la herramienta mysqladmin, de la siguiente forma:

```
c:\mysql\bin> mysqladmin -u root -p password 'clave'
```

Por supuesto si se sabe lo que se está haciendo, otra opción para modificar la clave es la siguiente:

```
mysql> UPDATE USER set Password=PASSWORD('miclave') where user='root';  
mysql> FLUSH PRIVILEGES;
```

En el ejemplo anterior la segunda sentencia es necesaria para que el sistema vuelva a leer las tablas del diccionario de datos (y así actualice todos estos cambios realizados).

3. Creación de nuevos usuarios en MySQL

Para añadir nuevos usuarios en mysql se utiliza la sentencia **GRANT** que permite crear un nuevo usuario otorgándole un conjunto de privilegios o permisos para trabajar en las distintas bases de datos y tablas almacenados.

Veamos algunos ejemplos:

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON nombrebd.* to  
usul identified by 'laclave';  
mysql> GRANT ALL PRIVILEGES ON *.* to superusuario@localhost  
identified by 'suclave';  
mysql > GRANT SELECT ON *.* to superusuario@'%' identified by  
'suclave';  
mysql> GRANT USAGE ON *.* to nada@localhost;
```

Para eliminar todos estos permisos se utiliza la sentencia REVOKE:

```
mysql> REVOKE SELECT, INSERT, UPDATE, DELETE ON nombrebd.* from  
usul identified by 'laclave';
```

4. Como crear tablas, índices y claves ajenas

Para crear y modificar las tablas de la BD se utiliza la siguiente sentencias SQL (ver manual de referencia de SQL):

- create table
- alter table

Veamos un ejemplo de utilización de la sentencia de creación de tabla con varios tipos distintos de datos:

```
mysql> create table mitabla (idtabla mediumint auto_increment  
primary key, nombre varchar(30) not null, direccion char(40),  
precio float, tipo enum('si', 'no'), fechaini date);
```

La sentencia alter table se utiliza para modificar la definición anterior. Por ejemplo para añadir una nueva columna de tipo blob:

```
mysql> alter table mitabla add column (comentario blob);
```

O por ejemplo para eliminar una columna:

```
mysql> alter table mitabla drop column comentario;
```

En mysql existen distintos tipos de tablas. Por defecto el tipo de tabla utilizado por mysql es el tipo MyISam. En la mayoría de los casos trabajar con este tipo de tablas puede ser suficiente. Ahora bien, si se desea que se soporten transacciones es necesario que las tablas sean de un tipo especial InnoDB. El tipo de tabla que se creará se especifica en la propia sentencia de creación de la tabla de la siguiente manera:

```
mysql> create table tipo(nombretipo VARCHAR(30) PRIMARY KEY,  
dias int) type=innodb;
```

También es necesario definir las tablas como InnoDB en el caso de que se deseen incorporar restricciones de claves ajenas al diseño.

Para crear índices sobre las columnas de las tablas se utilizan la sentencia

- create index on

```
mysql> create index ind_dias on tipo(dias)
```

La sentencia anterior crea un índice en el campo dias de la tabla tipo con el nombre ind_dias.

Los índices también se pueden incluir en la misma sentencia de creación de tablas de la siguiente manera:

```
mysql> create table tipo(nombretipo VARCHAR(30) PRIMARY KEY,  
dias int, index ind_dias (dias)) type=innodb;
```

Para crear una clave ajena en una tabla de tipo InnoDB es necesario que las columnas implicadas estén indexadas (se haya creado un índice sobre ellas):

```
mysql> create table padre (idpadre int primary key)
type = innodb;
```

```
mysql> create table hijo(idhiujo int primary key,
id_padre int, index (id_padre),
foreign key (id_padre) REFERENCES padre(idpadre)) type = innodb;
```

Si intento insertar los siguientes datos en la tabla hijo:

```
mysql> insert into hijo values(1,1);
```

Obtendría el siguiente resultado:

```
mysql> ERROR 1216: Cannot add or update a child row: a foreign
key constraint fails
```

Para evitar que al borrar o actualizar hayan datos inconsistentes con las restricciones impuestas se puede utilizar lo siguiente:

```
mysql >create table hijo (idhijo int PRIMARY KEY,
id_padre int, index (id_padre),
FOREIGN KEY (id_padre) REFERENCES padre(idpadre) on delete
cascade on update cascade) type=innodb;
```