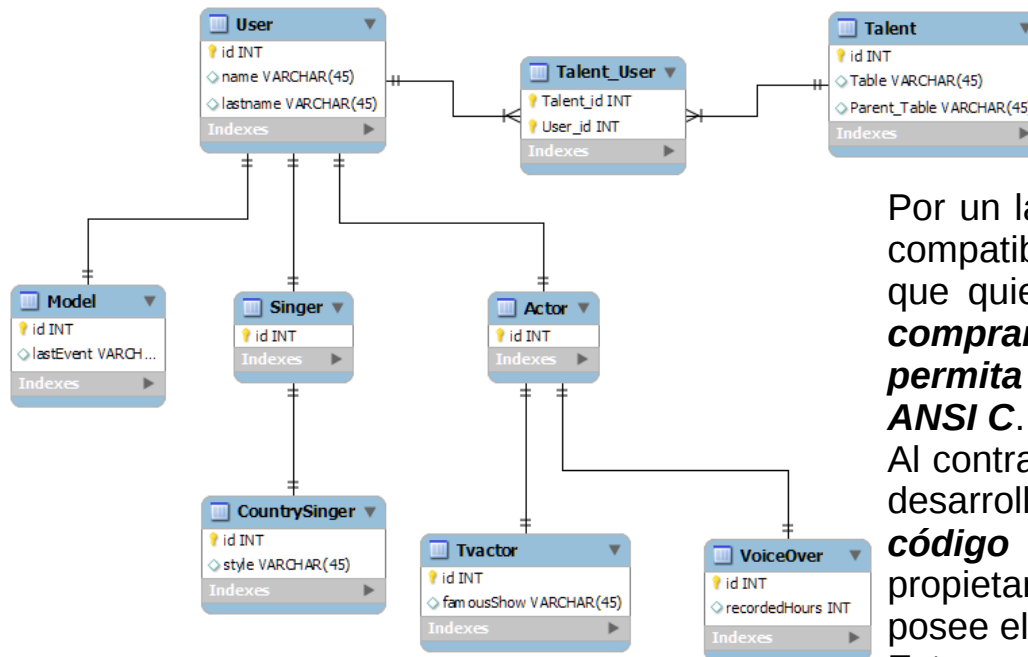




MySQL DB Server

Introducción

MySQL Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de **Sun Microsystems** y ésta a su vez de **Oracle Corporation** desde abril de 2009— desarrolla MySQL como **software libre en un esquema de licenciamiento dual**.



Por un lado se **ofrece bajo la GNU GPL** para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran **incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso**. Está desarrollado en su mayor parte en **ANSI C**.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y **el copyright del código está en poder del autor individual**, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.



Lenguajes de programación

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos **MySQL**, incluyendo **C**, **C++**, **C#**, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, **Java** (con una implementación nativa del driver de Java), Lisp, **Perl**, **PHP**, **Python**, **Ruby**, Gambas, REALbasic (Mac y Linux), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe una interfaz ODBC, llamado **MyODBC** que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP.



MySQL es muy utilizado en **CMSs** web, como **Wordpress**, **Joomla**, **Drupal** o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como **Bugzilla**.

Su popularidad como aplicación web está muy ligada a **PHP**, que a menudo aparece en combinación con MySQL.

MySQL es una base de datos **muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM**, pero puede provocar **problemas de integridad en entornos de alta concurrencia en la modificación**. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante adelantar monitoreos sobre el desempeño para detectar y corregir errores tanto de SQL como de programación.

Instalación de MySQL

La manera recomendada de instalar MySQL en GNU/Linux es utilizando paquetes binarios. Para realizar esto utilizaremos repositorios.

La distribución de GNU/Linux a utilizar será CentOS 6.x (Basada en Red Hat Enterprise Server), la cual utiliza el gestor de paquetes YUM.

1. Instalación a través de yum:

```
yum install mysql mysql-server mysql-devel
```

2. Iniciar, detener, reiniciar el servicio Mysql:

```
service mysqld start
```

```
service mysqld stop
```

```
service mysqld restart
```

3. Una vez realizados los pasos anteriores, podemos usar el script **mysql_secure_installation** para “**segurizar**” nuestra instalación de **MySQL**.

A partir de la versión 7 de **CentOS (CentOS 7.x)**, se ha reemplazado **MySQL** por **MariaDB**.

MariaDB motor de base de datos de **código abierto equivalente a MySQL** y se puede instalar con **yum install mariadb mariadb-server mariadb-devel**.

Si se desea instalar **MySQL**, es necesario añadir el **mysql-community repo** (repositorio de MySQL community) al sistema: **rpm -Uvh <http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm>**

...entonces se podrá instalar MySQL siguiendo los pasos anteriormente vistos.

Operaciones básicas con MySQL

Ya instalado y ejecutándose el servidor MySQL podremos logearnos desde el shell de nuestro servidor CentOS con el comando:

```
mysql -u nombreusuario -p
```

Deberemos ingresar la password del del usuario de mysql, en este caso nuestro usuario es “root” que corresponde al usuario administrador de MySQL (***no tiene relación con el usuario root del sistema***, sólo se llaman igual) Una vez logueados dentro del servidor MySQL nos encontraremos con el siguiente prompt:

```
mysql>
```

Ya estamos en condiciones de comenzar a ingresar instrucciones al servidor de Bases de Datos.

Crear una Base de Datos y su usuario.

Una buena práctica para el trabajo con bases de datos y el desarrollo/implementación de aplicaciones es ***NUNCA*** utilizar al ***usuario administrador del sistema de bases de datos*** (para el caso de MySQL es el usuario ***root***). Esto evitará potenciales problemas de seguridad y minimizará el riesgo de dañar o de modificar alguna base de datos de sistema.

En estos casos se debe crear una base de datos específica para la aplicación o sistema que se desarrollará y un usuario también específico con permisos totales a ella

a) Crear una base de datos:

Una base de datos en MySQL se crea mediante la instrucción:

```
mysql> CREATE DATABASE Nombre_de_la_Base_de_Datos;
```

b) Crear un usuario para el sistema de base de datos:

Una vez creada la Base de Datos, debemos crear un usuario en el sistema, el cual será quien utilice la base de datos:

```
mysql> GRANT USAGE on *.* to 'nombre_usuario'@'localhost' identified by  
'nueva_password';
```

Operaciones básicas con MySQL

c) Asignar privilegios totales sobre la base de datos creada al usuario:

Una vez que tenemos la Base de Datos y el Usuario correspondientes, asignamos los permisos:

Para acceso local:

```
mysql> grant all privileges on Nombre_de_la_Base_de_Datos.* to  
'nombre_usuario'@'localhost';
```

Para acceso remoto:

```
mysql> grant all privileges on Nombre_de_la_Base_de_Datos.* to  
'nombre_usuario'@'%';
```

Completados los pasos anteriores, ya podemos hacer uso de la base de datos con su usuario correspondiente:

- Cerramos la sesión de root: **mysql> quit**

- Iniciamos sesión conectándose a la base de datos creada con su usuario:

```
mysql Nombre_de_la_Base_de_Datos -u nombre_usuario -p
```

Lenguaje SQL

SQL (por sus siglas en inglés **Structured Query Language**) es un lenguaje declarativo de acceso a **bases de datos relacionales** que permite especificar diversos tipos de **operaciones** en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas.

src: <https://es.wikipedia.org/wiki/SQL>

Operaciones básicas

Sentencia	Uso	Ejemplo
Create Table	Crea una nueva tabla con una estructura determinada.	create table alumnos(rut CHAR(9), nombre CHAR(30), apellido CHAR(30), email CHAR(30), edad INT);
Insert	Inserta un registro a la tabla.	insert into alumnos values('123456789', 'Pedro', 'Perez', 'pperez@perez.cl', 26);
Select	Busca en la tabla según un criterio determinado y devuelve el resultado de la búsqueda. SELECT nombre_del_campo from nombre_de_la_tabla where condicion=valor;	select nombre from alumnos; select nombre from alumnos where edad > 21; select * from alumnos where rut='14.123.123-8';
Update	Actualiza el valor de un campo en un registro determinado. UPDATE nombre_tabla SET valor_a_actualizar WHERE condición;	UPDATE alumnos SET edad=41 WHERE rut='14.123.123-8'; <u>IMPORTANTE:</u> Si NO se incluye una condición para el UPDATE. Éste realizará el cambio de valor a TODOS los registros de la tabla.
Delete	Elimina registros de una tabla según una condición. DELETE from nombre_tabla WHERE condición;	DELETE from alumnos where rut='14.123.123-8'; <u>IMPORTANTE:</u> Si NO se incluye una condición para el DELETE. Éste eliminará TODOS los registros de la tabla.

src: <http://www.w3schools.com/sql/>

Backup y Restore de una base de datos.

Backup

La herramienta ***mysqldump*** permite generar copias de seguridad de una o todas las bases de datos del servidor MySQL.

Mysqldump genera un archivo de texto que representa la base de datos. Por defecto, el archivo de texto contiene una lista de sentencias SQL que usaría para recrear la base de datos , pero también se puede exportar la base de datos en otro formato como CSV o XML.

El comando para realizar el respaldo (desde la shell) es:

```
mysqldump -u root -p nombre_base_de_datos > backup_base_datos.sql
```

Si se desea respaldar todas las bases de datos:

```
mysqldump -u root -p --all-databases > backup_de_todas_bd.sql
```

*(Se puede leer la página **man** de ***mysqldump*** para ver todas sus opciones.)*

Restore

Para restaurar el respaldo realizado con ***mysqldump***, primero se debe crear una base de datos vacía para recibir los datos a restaurar. Una vez realizado lo anterior, la restauración se realiza con:

```
mysql -u root -p base_de_datos_vacia < backup_base_datos.sql
```


Configuración MySQL

El archivo de configuración de MySQL es: my.cnf y se encuentra (por defecto) en el directorio /etc.
Algunas de las directivas de configuración más comunes:

[mysqld]	
user = mysql	Usuario bajo el cual se ejecuta el servidor
pid-file = /var/run/mysqld/mysqld.pid	Archivo en el que el servidor guarda el ID del proceso demonio de escucha (daemon).
socket = /var/run/mysqld/mysqld.sock	Archivo en el que el servidor crea el socket
port = 3306	Puerto desde donde se aceptarán conexiones
log = /var/log/mysql.log	Archivo donde se registra el log
basedir = /usr	Directorio donde se encuentra MySQL.
datadir = /var/lib/mysql	Directorio donde están los datos
tmpdir = /tmp	Directorio temporal (para crear pej. tablas temporales)
language = /usr/share/mysql/english	Archivo de mensajes
skip-locking	No usar sistema de bloqueo (para hacer chequeos se tendrá que parar el servidor)
skip-networking	No escucha conexiones TCP/IP
bind-address = 127.0.0.1	Dirección de red donde está escuchando el servicio (si se asigna 127.0.0.1 significa que está escuchando en localhost, no acepta conexiones remotas)
max_allowed_packet = 16M	Tamaño máximo permitido para el paquete
max_connections = 100	Cantidad máxima de conexiones permitidas

Configuración MySQL

[mysqld_safe]

log-error=/var/log/mysql.log	Archivo dónde se registrarán los errores del servidor.
------------------------------	--

[client]

port=3306	Puerto dónde se conectarán los clientes (acceso via TCP/IP)
-----------	---

Socket=/var/run/mysqld/mysqld.sock	Acceso mediante sockets Unix (localhost)
------------------------------------	--

Más información sobre tuning y optimización de MySQL Server

<http://dev.mysql.com/doc/refman/5.0/en/optimizing-the-server.html>

<http://stackoverflow.com/questions/10905226/mysql-my-cnf-performance-tuning-recommendations>

<https://www.linode.com/docs/databases/mysql/tuning-your-mysql-database>

Configuración MySQL
