PROGRAMACIÓN ORIENTADA A OBJETOS APLICADA A BASES DE DATOS

Por

LAURA NOUSSAN LETTRY

BrowserSQL – MySQL Workbench en Linux

(Abril 2015, Mendoza)

Aviso Legal

Para Utilizar el BrowserSQL basta tener el ejecutable jar configurado como archivo ejecutable (ver apunte anterior).
Lo que muestran las siguientes imágenes es cómo establecer una conexión. Y cómo utilizar el lenguaje SQL-DDL y SQL-DML para crear objetos y manipularlos
La explicación de cómo se establece una conexión y cómo se utiliza el BrowserSQL está en el tutorial, que tendrían que bajarlo.
Los script de la Base de Datos Escuela están en el sitio web

Cosas a tener en cuenta en la secuencia de ejecución del BrowserSQL:

1. Cuando recién se instala MySQL o cualquier otro DBMS en un sistema se puede decir que al conectarnos nos conectamos a la Instancia. Esta instancia tiene o nos permite ver el Diccinario de datos
2. Cuando creamos la base de datos con el BrowserSQL estamos conectados a la instancia de MySQL.
3. Para poder crear las tablas tenemos que desconectarnos de la instancia MySQL y conectarnos directamente a la base de datos puesto que las tablas pertencerán a la Base de Datos y no a la Instancia. Esto significa que el DBMS puede administrar varias bases de datos.

La importancia de la Normalización para la Integridad de Datos
Como ya vimos en clase la normalización es fundamental para lograr la integridad y seguridad de los datos. Ahora se verá por qué esto es tan importante.
La normalización establece de por sí, cuando está correctamente diseñada a un nivel lógico/físico, una correcta implementación en el DBMS elegido.
Por ejemplo, para crear las Tablas hay que seguir un orden lógico: primero se crean las T***ablas Primarias*** (aquellas que no tienen claves foráneas) y luego las ***Tablas Derivadas*** (derivadas porque tienen claves foráneas y por ende dependen de otras tablas).
Esta regla también es válida cuando se insertan datos ¿por qué? Ejemplo: en la Base de Datos Escuela tenemos estas tablas: Alumnos, Materias, Localidades y Notas.
Las tablas primarias son: Localidades y Materias
Las tablas derivadas son: Alumnos y Notas (en este orden puesto que Notas depende de más de una tabla y Alumnos de una sola tabla pero además Notas depende de alumnos). O sea, todo esto tiene una lógica implícita.
Entonces cuando inserto tengo que tener datos necesariamente en las tablas Primarias o en tablas derivadas que para otra tabla actúan como primarias. Ejemplo: no puedo insertar un Alumno con una localidad que no existe o no puedo insertar una nota de un alumno que previamente no está registrado en la tabla Alumnos. En estos casos si el usuario intenta agregar al alumno una localidad que no existe, el DBMS, si está bien diseñado lógicamente, tendría que impedir el alta o modificación del alumno y enviar un mensaje de error o advertencia al usuario.
Esta regla para crear las tablas e insertar datos en las mismas es válida también para actualizar los datos si se trata de modificarlos. En cambio si la actualización involucra borrar datos se sigue el camino inverso. ¿Por qué? Pues si tengo notas de un Alumno

que necesito borrar no puedo borrar primero el Alumno y después las notas porque se estaría rompiendo la integridad referencial, por lo tanto, para ese alumno primero necesito borrar los registros de dicho alumno en la tabla Notas y después recién borrar el alumno.

Es decir en al caso de borrar datos: primero se borran los registros de las tablas derivadas y luego el de las tablas primarias.

Es importante entender además de SQL utiliza varios lenguajes. Nosotros vemos el SQL-DML y SQL-DDL.

Las sentencias que tienen que ver con la definición de datos están dentro de SQL-DDL: esto involucra la creación de las tablas y otros objetos así como el borrado de tablas completas, su cambio de nombre, etc.

Toda la teoría sobre estas sentencias está en el apunte ***SecuenciaDidactica***

Involucra estas sentencias:

- CREATE
- DROP
- RENAME
- ALTER (ojo aquí no hay uniformidad entre los diferentes DBMS, siempre hay que buscar en la ayuda de cada DBMS). En general esta sentencia permite modificar por ejemplo la estructura de una tabla, como ser agregar un nuevo atributo o columna o bien modificarlo (cambiarle el nombre a la columna, cambiar el tipo de datos de una columna, etc.) o bien eliminar una o más columnas.

En cambio las sentencias de manipulación de datos tienen que ver con objetos previamente creados; es decir con inserciones, modificaciones, eliminaciones de registros en una o más tablas, de consultas en una o más tablas.

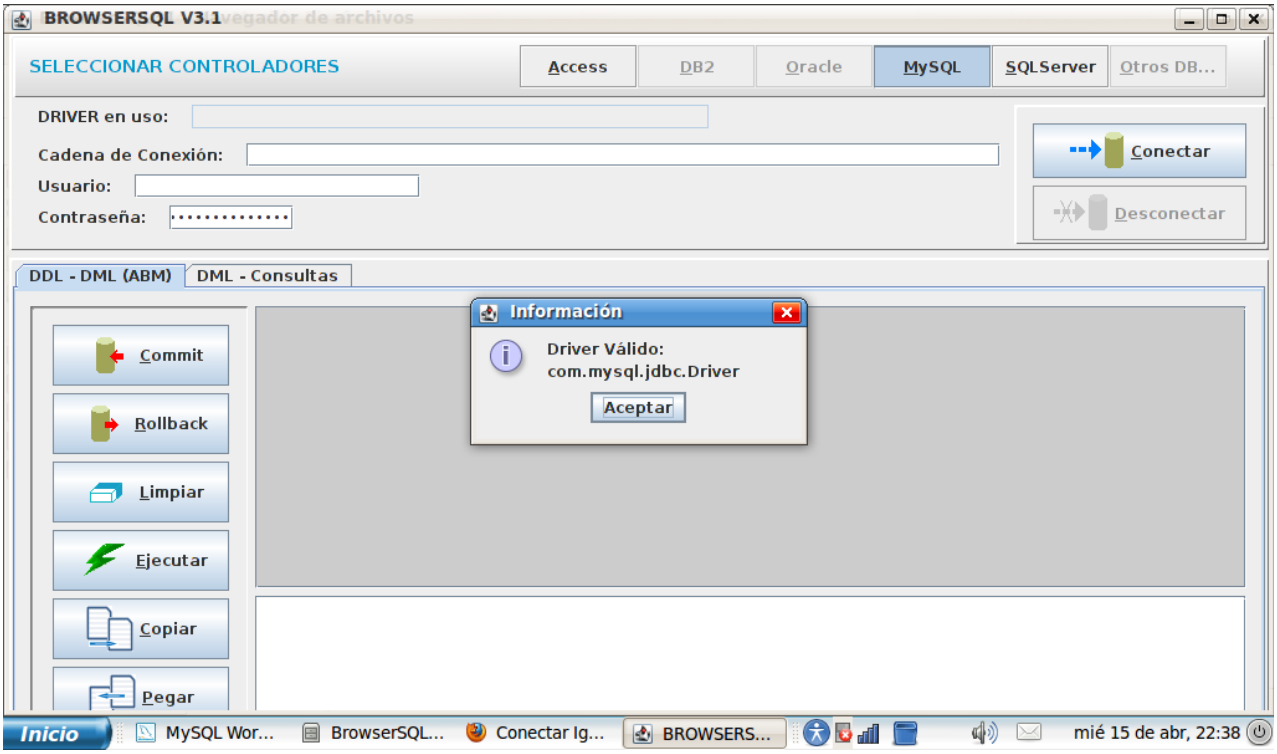
Estas sentencias por lo tanto pertenecen a lo que se conoce como SQL-MDL, o sea Lenguaje SQL para manipulación de datos e involucra a estas sentencias:

- INSERT
- UPDATE
- DELETE
- SELECT

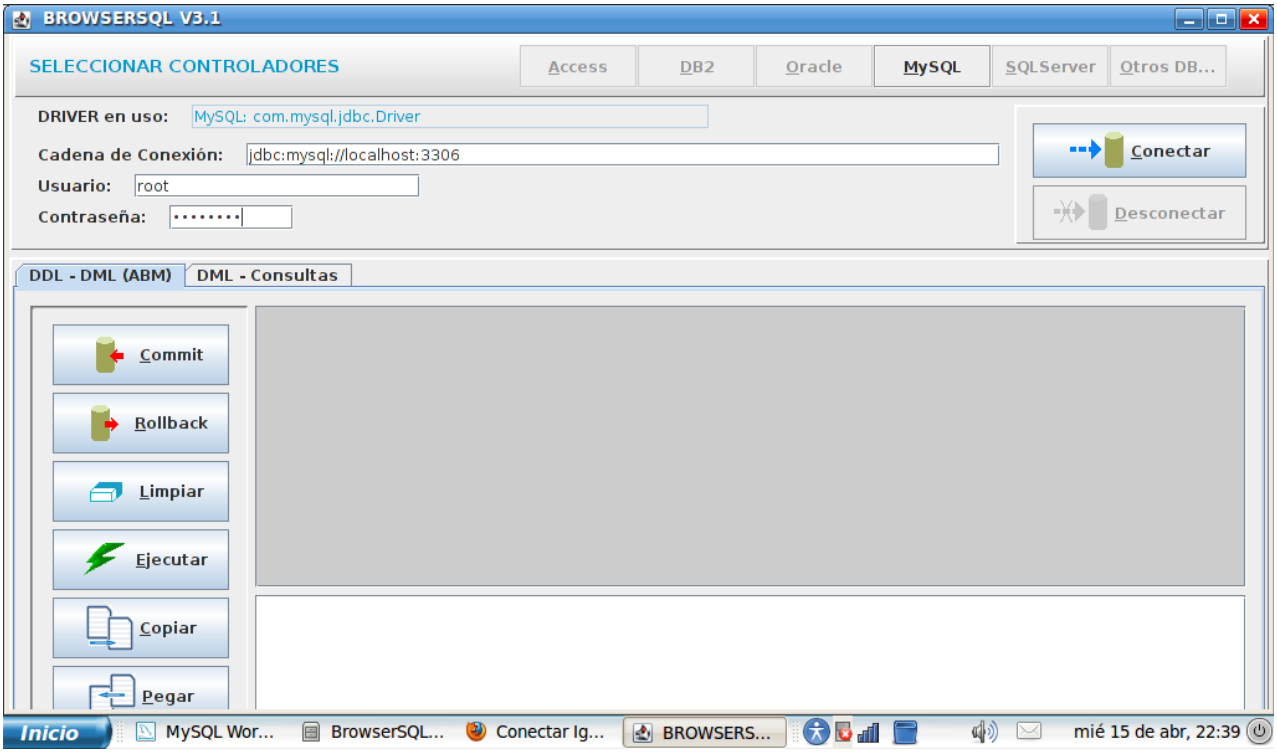
En resumen, todo esto se basa en la normalización y en cuanto a la teoría además de lo que se refuerza en clase se basa en el material que está disponible en los apuntes de la materia en el sitio web, es decir los PDF sobre POOABD-DBMS parte 2, Secuencia Didáctica y el tutorial del BrowserSQL.

A esto se puede agregar los videos y otros tutoriales de este año y años anteriores puesto que el manejo de la base de datos mediante SQL es independiente del sistema operativo.

Conectando con el DBMS y creando la Base de Datos

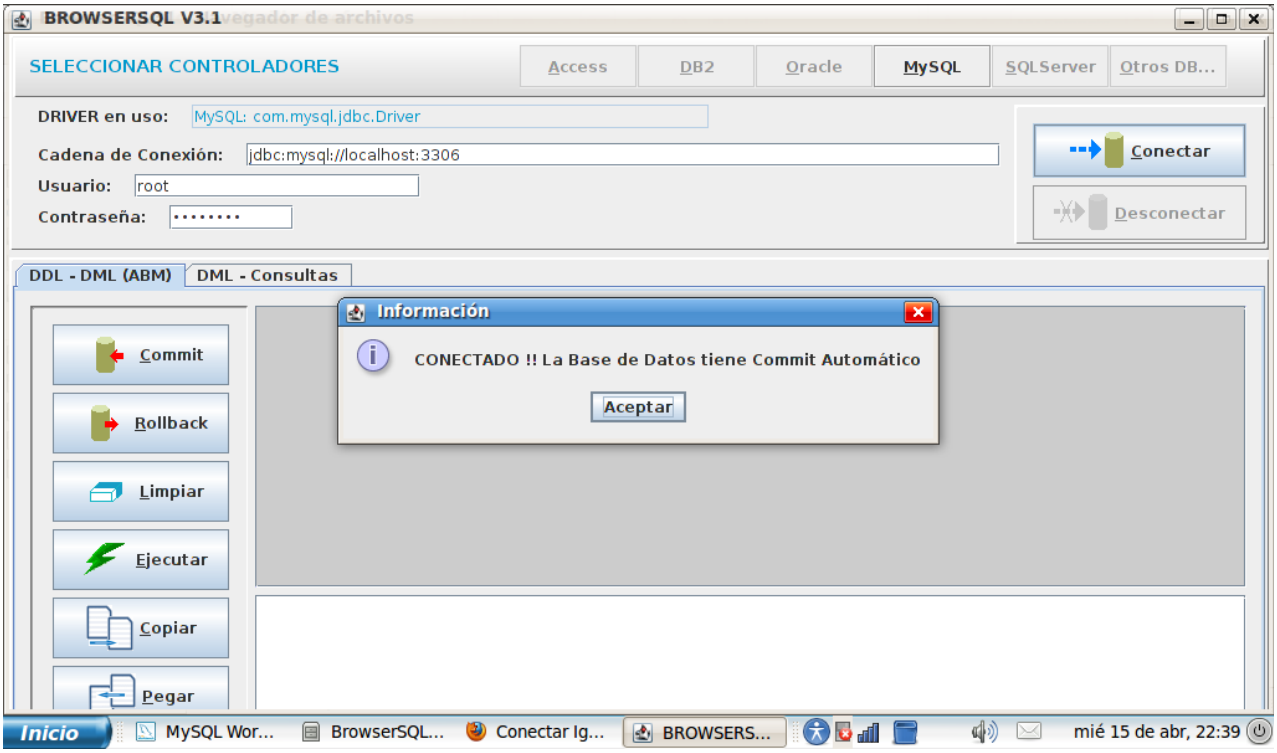


Aquí estamos conectándonos con el DBMS (con la instancia)
Se clicka el botón MySQL (este llama al JDBC o Driver para MySQL que está dentro del directorio Lib en la instalación del Browser)
Si está bien muestra el mensaje Driver válido



En la parte superior escrito en azul figura el nombre del driver que se está utilizando
El usuario tiene que ingresar los datos necesarios para establecer la conexión con el

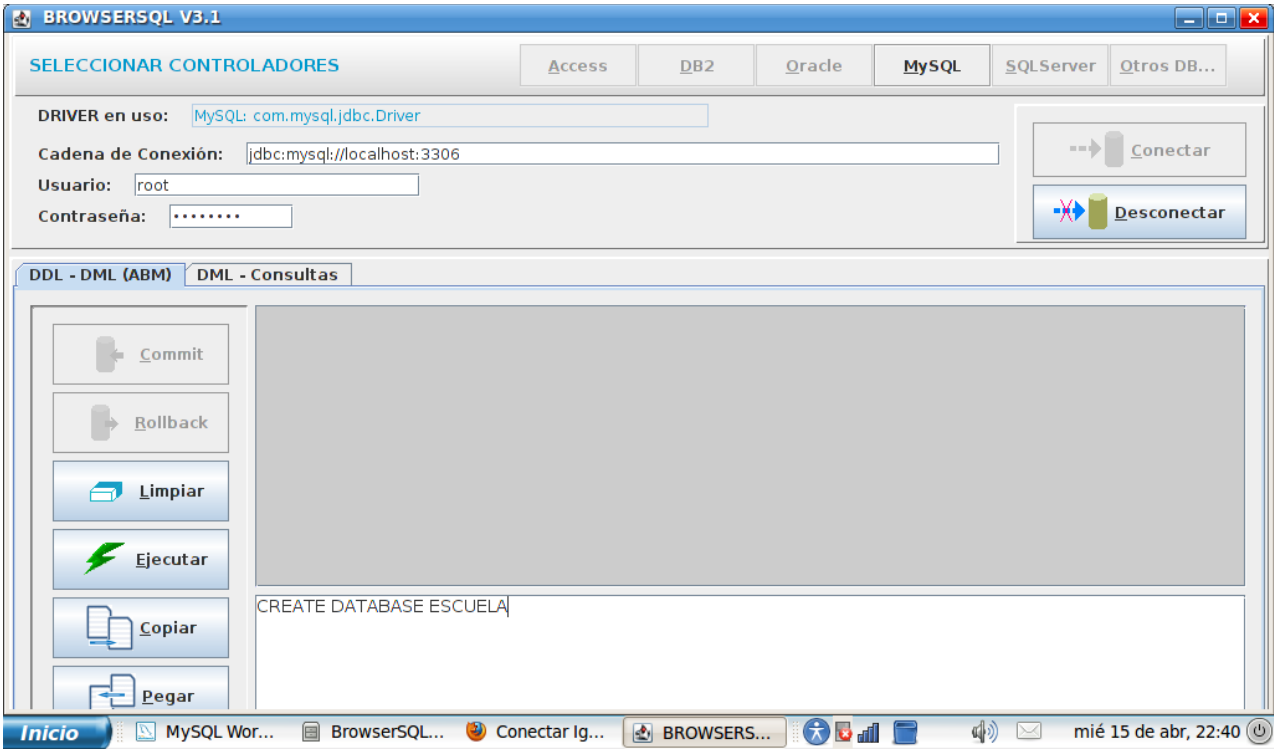
DBMS, esto es:
la cadena de conexión: jdbc.mysql://localhost:3306
la primera parte corresponde al nombre de la clase (esto depende del paquete JDBC del driver) después va el servidor y el puerto: en este caso el localhost y el puerto por omisión que utiliza MySQL es el 3306.
Si quisiéramos conectarnos a otra máquina en lugar de localhost habría que colocar la IP, por ej.
el usuario: es el usuario MySQL con el que vamos a realizar la conexión
la contraseña: es la contraseña del usuario.
Estoy usando el usuario principal y su contraseña (la que debimos colocar al instalar MySQL en la netbook)
Finalmente hay que clickar en el botón ***Conectar***



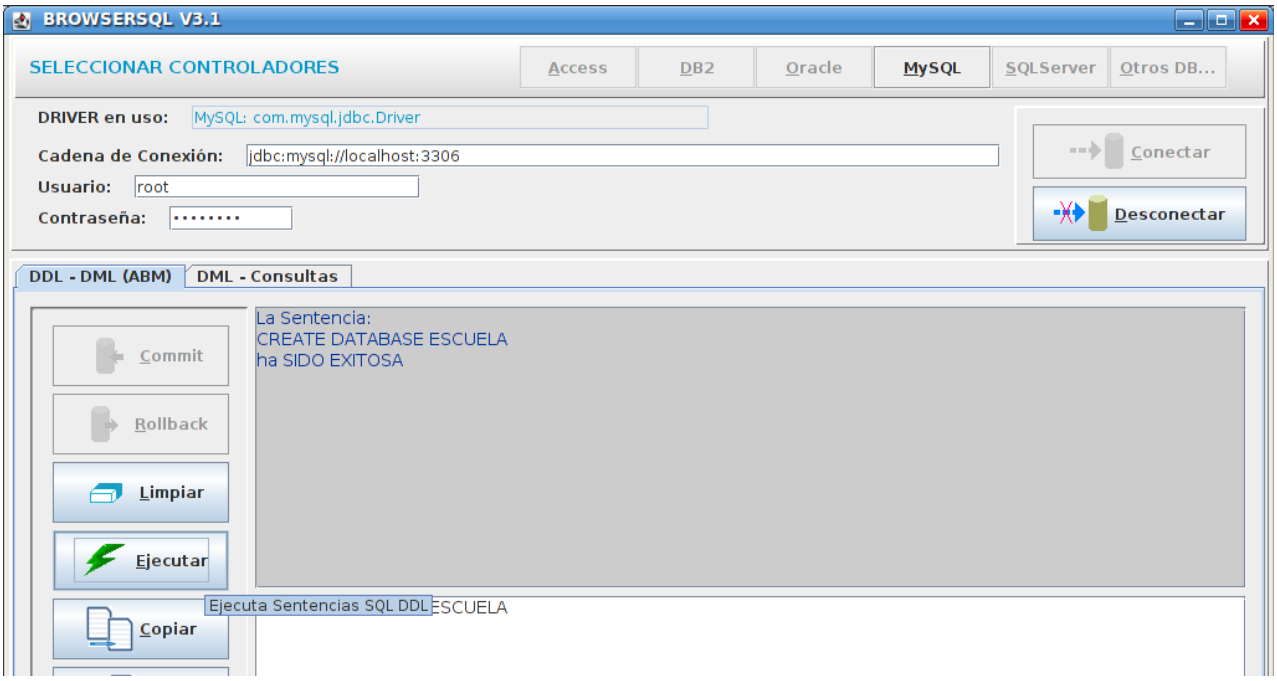
Si la conexión fue exitosa aparecerá el mensaje de la imagen, sino aparecerá un mensaje de error que indicará sucintamente el motivo (estos mensajes son enviados directamente desde el driver)

Es importante destacar que la cadena de conexión o la dirección ingresada es la de la instancia MySQL--> jdbc.mysql://***localhost:3306***

Estando conectados a la Instancia de MySQL creamos la primer base de datos: ESCUELA

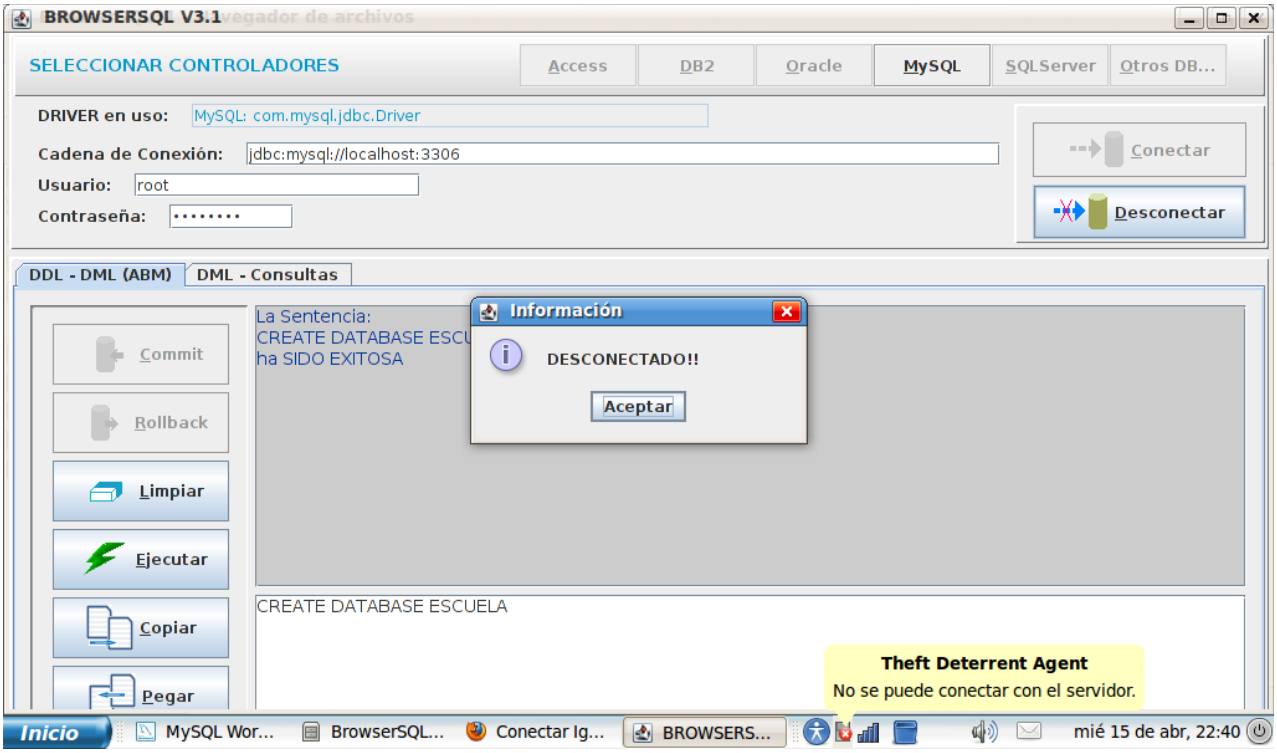


La sentencia utilizada de SQL-MDL es: CREATE [DATABASE |TABLE …] object-name
Específicamente CREATE DATABASE ESCUELA
En Linux es importante recordar que los objetos y variables son sensibles a mayúsculas y minúsculas; es decir no es lo mismo en Linux (como sucede con Windows) decir ESCUELA que decir Escuela o escuela. Para Linux serían tres bases de datos diferentes y si creo la base de datos como ESCUELA y en otra oportunidad quiero consultarla como escuela me dará error.

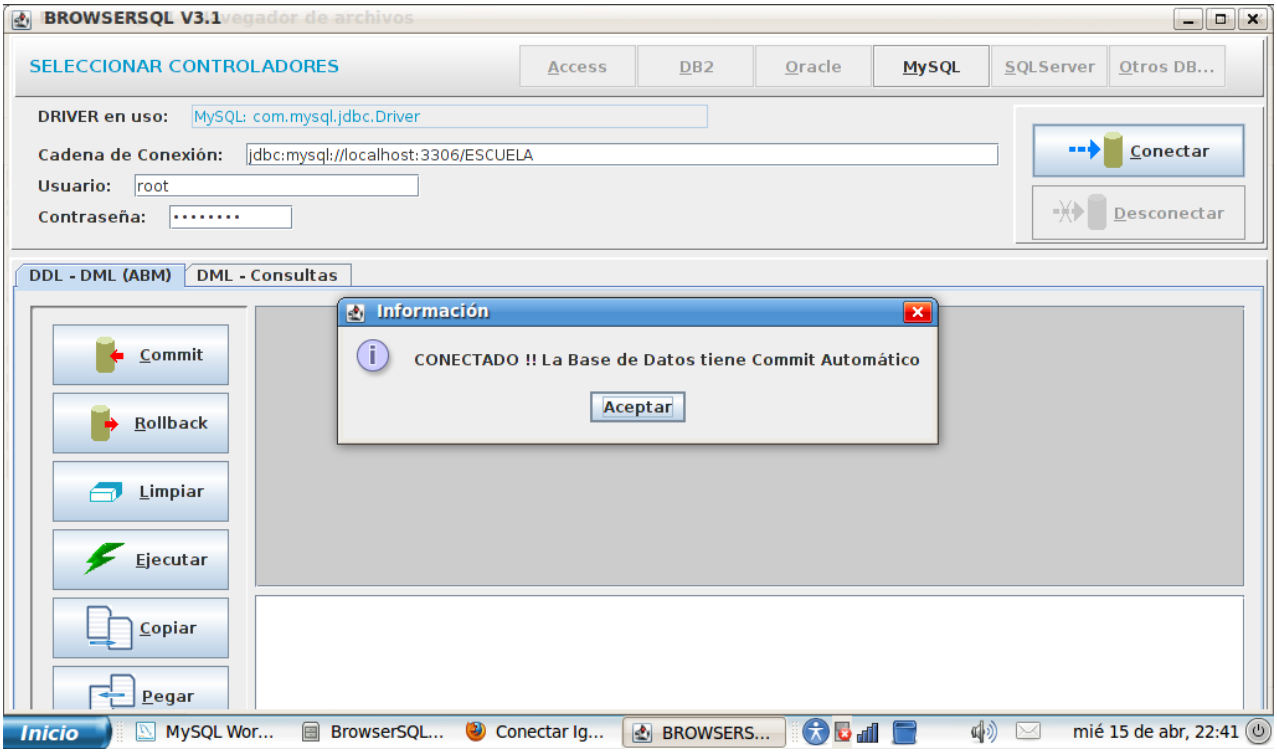


La imagen muestra que hemos ejecutado la sentencia y el resultado se muestra en el panel de mensajes (sobre cómo usar los diferentes botones y funcionalidades de los paneles consultar el Tutorial del BrowserSQL)

Ahora que hemos creado la base de datos ESCUELA, si queremos trabajar con ella tendremos que crear las Tablas. *Pero primero tenemos que desconectarnos de la instancia MySQL!!* y conectarnos al a Base de datos en sí. Recordar que MySQL así como otros DBMS pueden manejar y administrar más de una base de datos.

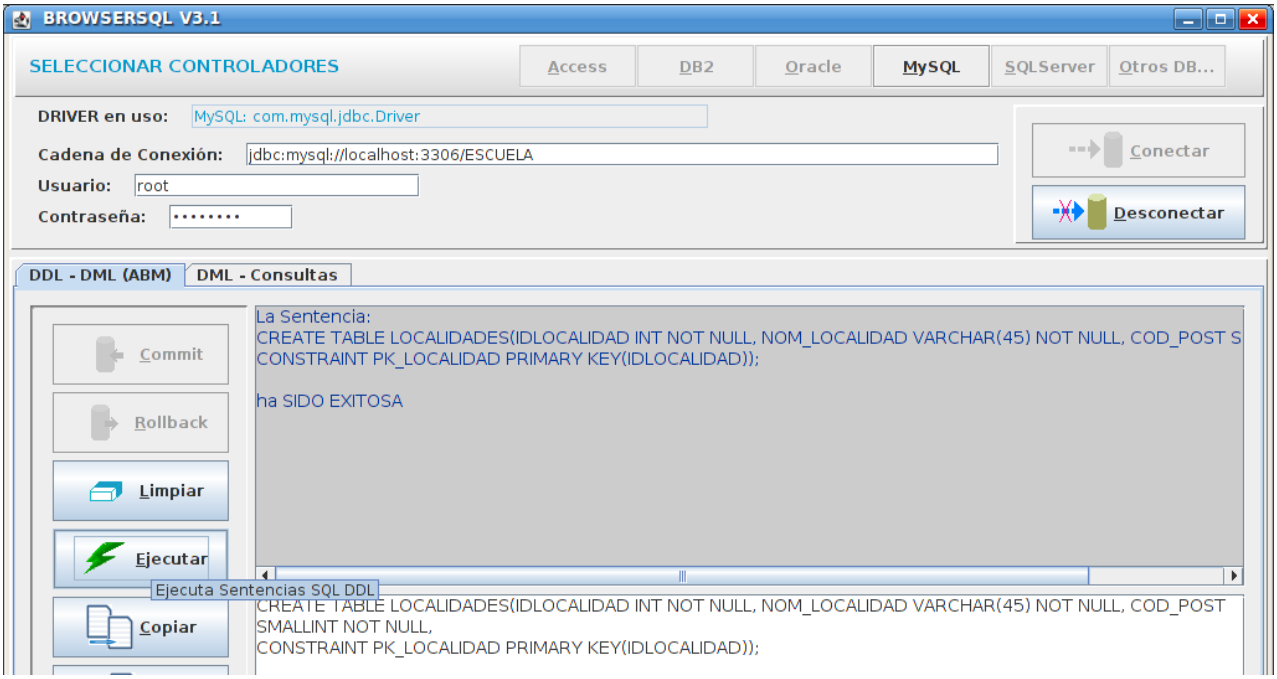


Creando las Tablas de la Base de Datos e insertando Datos

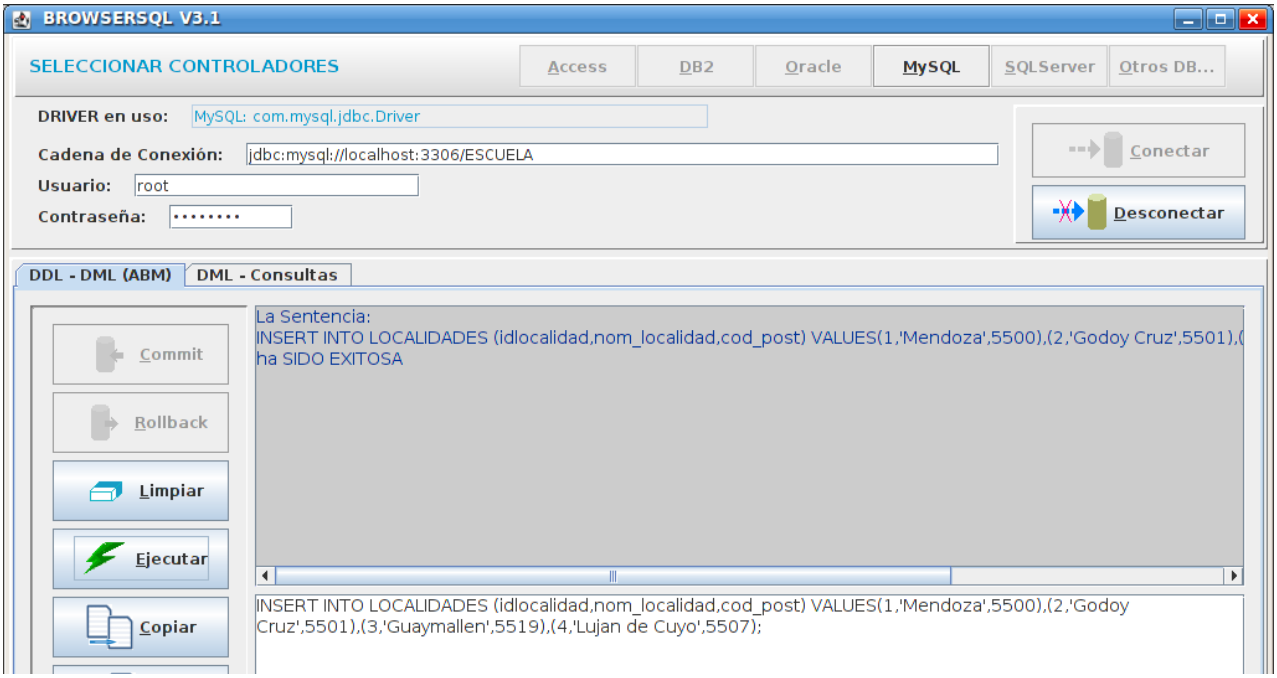


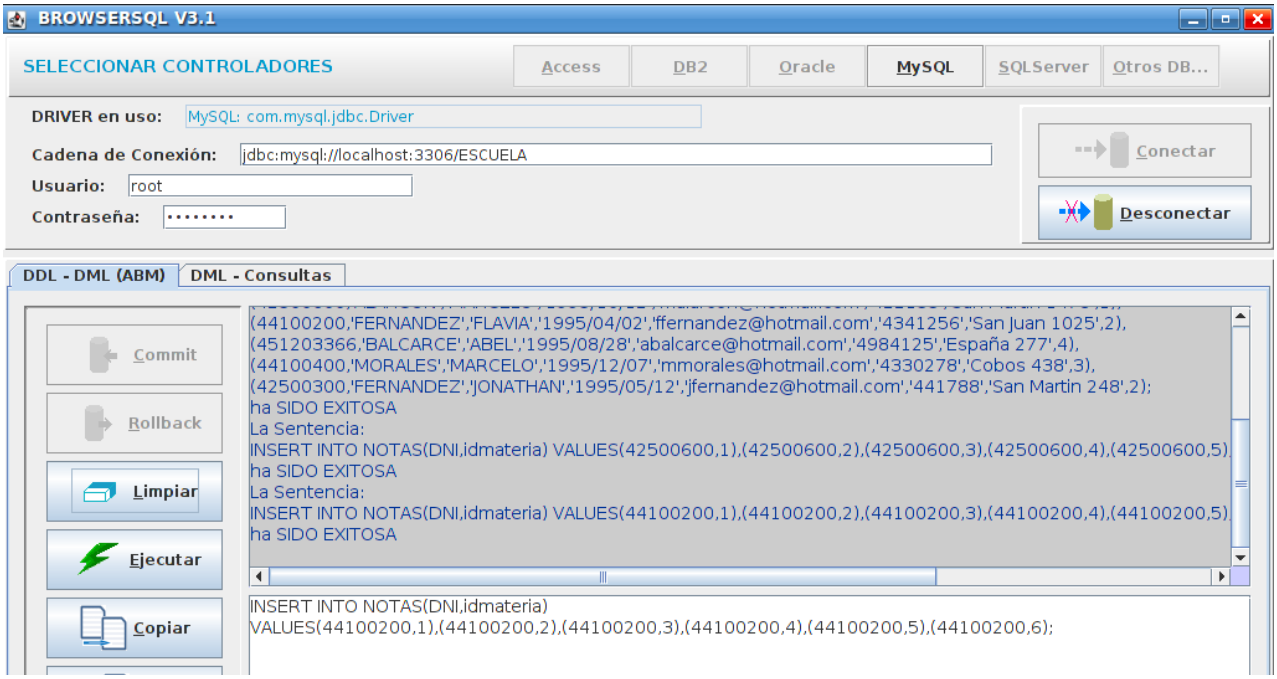
Como podemos apreciar ahora estamos conectados a la base de datos ESCUELA ya que así lo tenemos que indicar en la cadena de coneción: jdbc.mysql://localhost:3306/ESCUELA

En el sitio web de la materia están los scripts para crear las tablas y los inserts de la base de datos ESCUELA. Dichos scripts han sido utilizados en años anteriores así pues también están las sentencias para el DBMS SQL Server, o sea, utilicen las de MySQL.



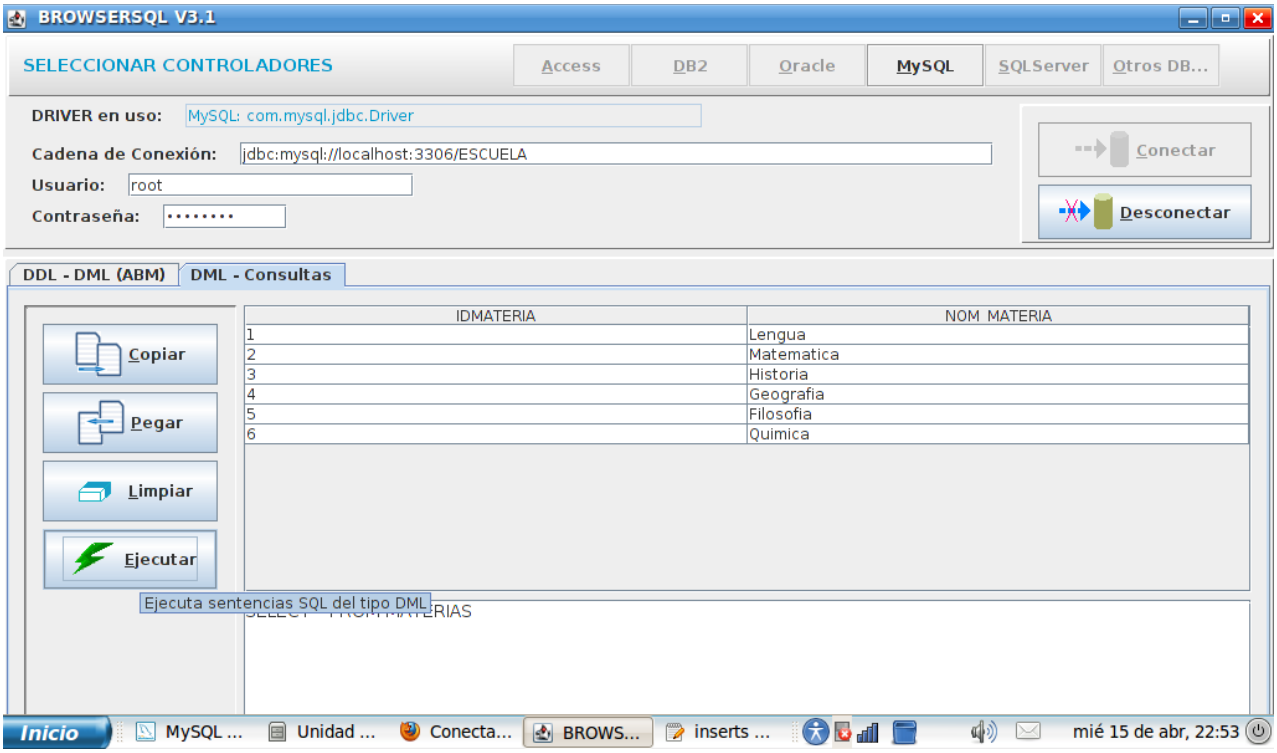
Desde el archivo de script he copidado la sentencia CREATE TABLE LOCALIDADES.... y luego la he ejecutado. Como LOCALIDADES es una tabla primaria, lo mismo que MATERIAS, se puede empezar por cualquiera de ellas. En el panel de mensajes de ejecución puede leerse que la sentencia ha sido exitosa, es decir ya está creada la tabla. Podríamos seguir con Materias y también con Alumnos puesto que Alumnos no depende de Materias y ya existe la tabla Localidades. El asunto es no olvidar la dependencia entre tablas. La última por lógica a crear es Notas.
La siguiente imagen muestra la inserción de datos en la tabla LOCALIDADES, lógicamente también se copia la sentencia desde el ***script que previamente se escribe basándose en el diseño lógico y físico que hemos llevado a cabo con anterioridad.***





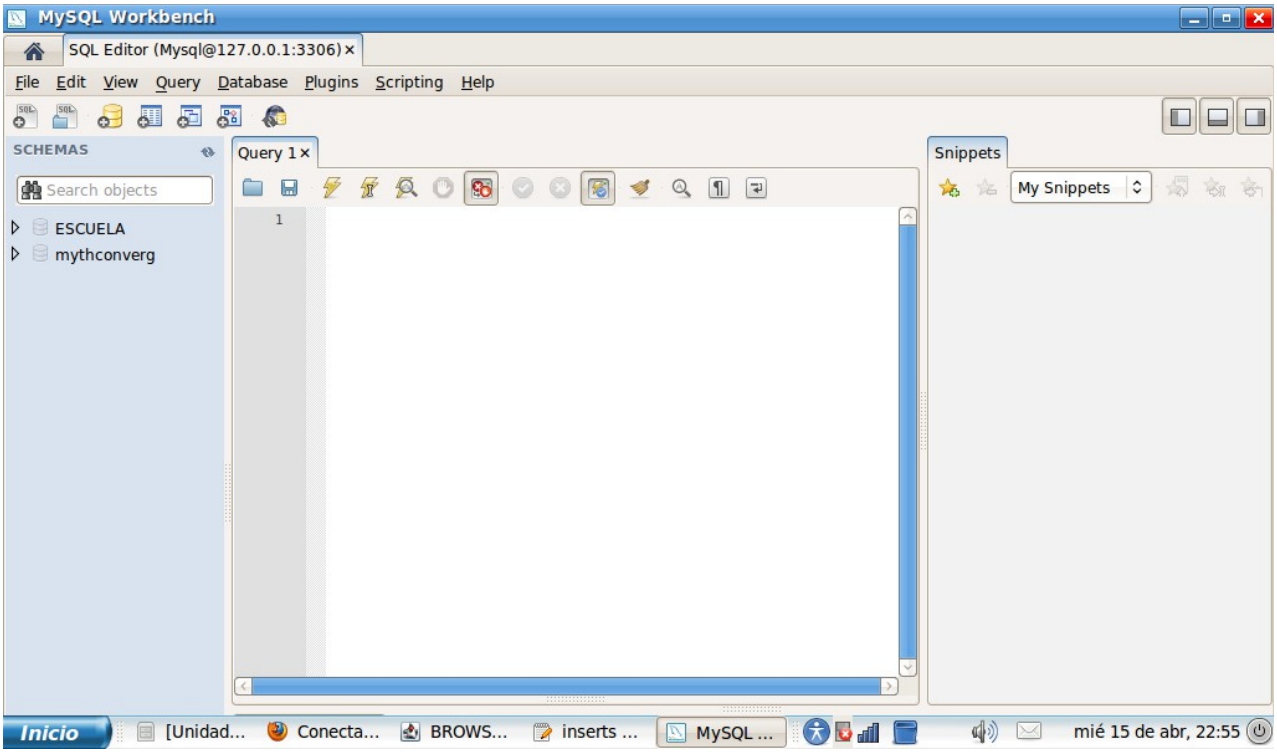
Se puede apreciar la ejecución del script de inserción para otras tablas.

Finalmente una vez insertados todos los datos en las tablas consultamos en la base de datos.
La segunda solapa del browser, DML-Consultas, nos permite ingresar una sentencia SELECT



Así con el BrowserSQL podemos ver los datos que tiene la tabla MATERIAS

La siguiente imagen muestra qué se ve en MySQL en modo gráfico (utilizamos para ello el Workbench)



Podemos apreciar que ya existe la base de datos ESCUELA. Al expandir podremos ver las tablas

Si bien con el Workbench podemos modificar la estructura de la tabla, ese no es el objetivo ahora sino poder ver qué información tiene las tablas y corroborar visualmente lo

que nos muestra el BrowserSQL