**SQL**

**MySQL:** Es un sistema de manejo de bases de datos que utiliza SQL como lenguaje de distinción de datos (LLD) y como lenguaje de manipulación de datos (LMD)

**Para empezar, vamos a D:\MySQL\mysql-9.0.1-winx64\bin (la carpeta bin del programa) y en la barra para explorar archivos escribimos cmd**

**Para iniciar el servidor desde la consola:**

**mysqld**

o

**mysqld --console**

(este es para q nos diga si aparecen mensajes de error en la misma ventana de comandos)

El mensaje que tendría que aparececr al final es “… ready for connections..”

Para detener el servidor desde la consola:

**mysqladmin -u root shutdown**

**Teoria**: El servidor de MySQL sigue el modelo Cliente-Servidor, esto quiere decir que el servidor una vez que esta corriendo va a escuchar por conecciones en un numero de IP y un numero de puerto y los clientes que pueden ser programas se conectan a través de la red con el servidor y hacer consultar (queries) u otras operaciones.

Nosotros lo haremos todo a través del localhost pero también se podría hacer a través de la red.

Para permitir que los programas clientes se conecten al servidor MySQL permite definir distintos usuarios con diferentes privilegios

Mientras tengamos la ventana abierta estará corriendo el servidor.

**Para entrar como cliente** vamos a la carpeta bin escribimos en la barra de exploración de carpetas:

mysql -u root

Esto ejecuta al cliente y se conecta al servidor utilizando el usuario root

Si funciona todo bien veremos el prompt mysql y la versión del servidor, desde aca podemos ejecutar consultas:

**show databases;**

Este comando nos muestra las bases de datos que están creadas en el servidor, la primera vez solo mostrara las bases de datos propias de SQL que almacenan los datos de los usuarios y configuraciones del servidor.

Para setear la contraseña:

Set password for root@localhost=’bd-dcic’;

**Siempre que ejecutamos un comando desde el cliente se finaliza con “;”**

Ahora como tenemos contra tenemos q ingresar asi:

**mysql -u root -p**

Luego la contra:

bd-dcic

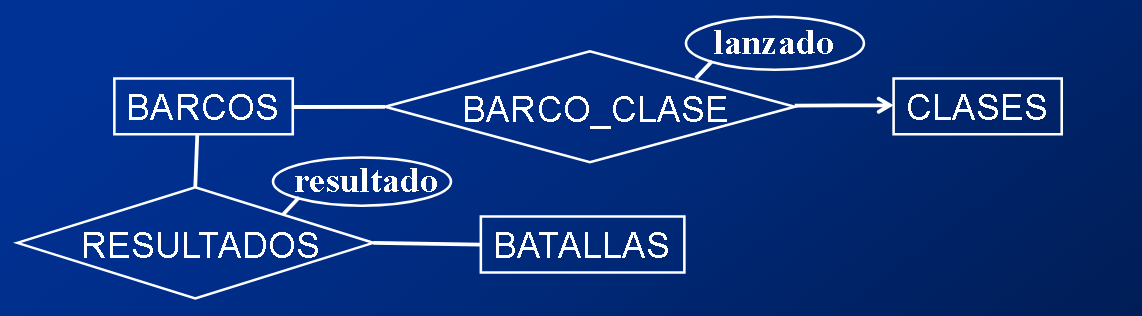
**Crear una base de datos**

Para crear una base de datos utilizando el cliente mysql hay dos opciones:

1. **Versión más practica**: Poner una secuencia de sentencias SQL en un archivo de texto, y ejecutar toda la secuencia de una vez. (Batch Mode - sección 3.5 del manual)
2. Ingresar las sentencias SQL una a una desde la línea de comandos, es decir, crear la base de datos y luego crear tabla por tabla. (ver sección 3.3 del manual).

Ejemplo:

Consideremos una B.D. sobre barcos y batallas:



BARCOS (nombre barco, id, capitan)

CLASES (clase, tipo, pais, nro caniones, calibre, desplazamiento)

BARCO CLASE (nombre barco, clase, lanzado)

BATALLAS (nombre batalla, fecha)

RESULTADOS (nombre barco, nombre batalla, resultado)

Para crear esta base de datos utilizaremos la opción de batch mode.

• Creamos un archivo de texto (batallas.sql) con la secuencia de sentencias necesarias para crear la base de datos.

• Luego desde el cliente ejecutamos toda la ruta del archivo (la parte de “mysql>” es el promt que aparece en el cmd):

mysql> \. batallas.sql

El profe para hacerlo más fácil puso el archivo en la carpeta bin de MyQSL entonces ya esta en la carpeta donde se abrió el cmd.

Este ultimo comando lo que hace es que tome como dato de entrada el archivo y ejecute en secuencia todas las sentencias SQL que hay dentro del archivo

Nota**: si una de las sentencias produce un error**, es posible que alguna de las otras sentencias de la secuencia se ejecute con éxito. Antes de ejecutar el archivo nuevamente**, borrar la base de datos desde el cliente** con:

**mysql> drop database batallas;**

• El comando drop database no elimina los usuarios. Si la secuencia de sentencias crea algún usuario, es necesario eliminarlos mediante:

**mysql> drop user nombre\_usuario@host;**

**Archivos SQL**

#Archivo batch (batalllas.sql) para la creación de la Base de datos del práctico de SQL

#Lo que esta después del "#" es un comentario

# Creo de la Base de Datos

**CREATE** **DATABASE** batallas;

# selecciono la base de datos sobre la cual voy a hacer modificaciones

**USE** batallas;

#-------------------------------------------------------------------------

# Creación Tablas para las entidades

**CREATE** **TABLE** barcos ( #creamos la tabla barcos

nombre\_barco VARCHAR(45) NOT NULL**1**,

capitan VARCHAR**2**(45) NOT NULL,

id INT UNSIGNED NOT NULL AUTO\_INCREMENT**3**,

**CONSTRAINT4** pk\_barcos

**PRIMARY** **KEY5** (nombre\_barco),

**KEY** (id)

) ENGINE=InnoDB**6**;

El tipo VARCHAR es una cadena de caracteres de longitud que varia y que tiene como máximo 45 caracteres.

**1.**La restricción **NOTNULL** indica que no puede tener valores nulos.

**2.INT** **UNSIGNED** quiere decir que será entero y será no signado, o sea solo puede tomar valores positivos

**3.AUTO\_INCREMENT** es para cuando en la tabla barcos insertamos una tupla si no especificamos el valor del id le asigna automáticamente un numero siguiente al número que fue ingresado. De esta forma manejamos automáticamente lo que son los identificadores a los que les asignamos valores de forma creciente.

**4 y 5.** Para la llave primaria **PRIMARY KEY** opcionalmente se puede poner la palabra **CONSTRAIT** y un nombre, esto nos sirve para cuando por ejemplo insertamos un barco y ese nombre de barco ya existía, entonces como no puede haber 2 nombres de barcos iguales, nos va a dar un error en la PRIMARY KEY y nos retorna lo que escribimos luego del CONSTRAIT, en este caso seria “pk\_barcos”. Esto nos sirve de referencia para saber cual fue la PRYMARY KEY que fallo.

**6**. **ENGINE=InnoDB** aca definimos de que tipo va a ser la tabla de MySQL, se puede no poner nada, pero este tipo de tablas son las mas completas y son las que utilizaremos.

Para la definición de esquemas:

CREATE TABLE nombre\_tabla (

atrib1 <tipo> [<restricción>],

atrib2 <tipo> [<restricción>],

....

atribN <tipo> [<restricción>],

<restriccion\_integridad\_1>,

....

<restriccion\_integridad\_k>);

Las restricciones de integridad son las llaves primarias y las llaves foráneas.

Podemos:

Definir el esquema de la tabla (atributos y sus tipos).

Definir, si existen, restricciones sobre los atributos.

Un valor no puede ser nulo (not null)

Restricciónes de valores (Ej. Unsigned, Unique), subconjunto válido, etc.

Definir restricciones a nivel de tabla

Clave primaria: primary key ( A1, ..., An )

Clave foránea: foreign key ( A1, ..., An ) references R( B1, ..., Bn )

Índices: key( A1, ..., An ), index ( A1, ..., An)

MIN 35:33