

# Sistemas de Información

## Práctica 1

Curso 2023/24

3º Curso

### El entorno MySQL

#### Introducción

La implementación del Sistema Gestor de Bases de Datos (SGBD) MySQL que vamos a utilizar en las prácticas sigue una arquitectura típica cliente-servidor: una sesión MySQL está formada por los siguientes procesos:

- Un proceso servidor de bases de datos (proceso "mysqld")
- El cliente de usuario (que pueden ser múltiples programas)

distinta de aquella en la que se ejecuta el servidor.

En las primeras prácticas a realizar no será necesario programar aplicaciones clientes (que se podrían implementar en diversos lenguajes para los que existen versiones de la librería: C, Java, Perl ...) si no que vamos a utilizar una aplicación cliente proporcionada con el entorno MySQL y

*El entorno MySQL sigue una arquitectura cliente-servidor. Para conectarnos al servidor "mysqld" utilizaremos el cliente "mysql"*

Un proceso servidor gestiona un conjunto de bases de datos en una determinada máquina. Un proceso cliente que desee acceder a una base de datos hace llamadas a una librería. La librería envía las peticiones de usuario a través de la red al proceso servidor que atiende la petición.

Los clientes pueden ejecutarse en la misma máquina o en una

que se denomina "mysql".

Sobre un SGBD se realizan dos tipos de operaciones:

- utilizar las bases de datos que gestiona, y
- administrar el propio SGBD.

En esta práctica comenzaremos a familiarizarnos con ambos tipos de operaciones.

### Objetivos

...

Esta práctica tiene como objetivos fundamentales:

- Configurar el entorno de realización de prácticas.
- Comprender el entorno de trabajo y empezar a conocer su uso y posibilidades.
- Crear y utilizar nuestra primera base de datos.
- Aprender a realizar consultas sencillas de una base de datos.

### Entorno

...

Si el alumno desea realizar las prácticas en su ordenador deberá instalar el entorno MySQL siguiendo las instrucciones indicadas en esta práctica (si utiliza OpenSUSE) o bien las instrucciones de instalación para su sistema operativo.

## Instalando MySQL

En este apartado se describe el procedimiento de instalación del software necesario para realizar las prácticas de la asignatura “Sistemas de Información”. Dicho software consiste en el SGBD y herramientas adicionales MySQL. Este software es libre, en su versión *community*, que es la que utilizaremos, y existen versiones para

todos los sistemas operativos habituales. En este documento se explica la instalación de la versión 5.5.33 de MySQL sobre el sistema operativo OpenSUSE, versión 12.3. No obstante, el procedimiento de instalación de otra versión y/o sobre otro sistema operativo es muy similar al aquí descrito.

### Instalación del software de MySQL (servidor, clientes y herramientas)

Por defecto, las versiones recientes de OpenSuse traen instalado un gestor de bases de datos denominado MariaDB. Por tanto, lo primero que debemos hacer es desinstalar dicho gestor y sus clientes. Para ello debemos ejecutar como superusuario el siguiente comando:

```
zypper remove mariadb mariadb-client mariadb-errormessages
```

Como determinadas aplicaciones del sistema necesitan un gestor de bases de datos, al desinstalar MariaDB con el comando anterior, automáticamente se instalan el servidor y los clientes de MySQL. Por tanto, sólo necesitamos instalar algunas herramientas adicionales. Para ello utilizaremos el comando:

```
zypper install mysql-community-server-tools
```

### Configuración del servidor MySQL

Para que el gestor de bases de datos se comporte como esperamos en esta asignatura, debemos cambiar algunos parámetros de configuración de su funcionamiento.

Todos los parámetros de configuración de MySQL se encuentran en un fichero denominado *my.cnf*, y que en OpenSUSE, al igual que en otros sistemas operativos, se encuentra en el directorio */etc*. Las modificaciones a realizar son las siguientes:

1. Añadir debajo de la línea que contiene:

```
[mysqld]
```

la siguiente línea:

```
default-storage-engine = innodb
```

2. Descomentar lo que está debajo de la línea:

```
# Uncomment the following if you are using InnoDB tables
```

De tal forma, que para la versión de MySQL a que hace referencia este documento, quedará lo siguiente:

```
# Uncomment the following if you are using InnoDB tables
innodb_data_home_dir = /var/lib/mysql
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /var/lib/mysql
# You can set .._buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
innodb_buffer_pool_size = 16M
innodb_additional_mem_pool_size = 2M
# Set .._log_file_size to 25 % of buffer pool size
innodb_log_file_size = 5M
innodb_log_buffer_size = 8M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50
```

3. Ejecutar el sistema gestor de bases de datos. Esta operación en OpenSUSE se realiza mediante el siguiente comando:

```
sudo /usr/sbin/rcmysql start
```

## Prueba de la instalación

Para probar si hemos realizado adecuadamente los pasos de instalación indicados, podemos realizar las siguientes dos pruebas:

1. Comprobar que el servidor está operativo y aceptando conexiones.

Para hacer esta comprobación, deberemos ejecutar el siguiente comando:

```
mysqlshow -u root
```

Si todo está correcto, deberíamos obtener como respuesta un listado de bases de datos que está gestionando el SGBD.

2. Comprobar que hemos modificado adecuadamente el fichero *my.cnf*.

Para hacer esta comprobación, deberemos ejecutar el siguiente comando:

```
mysql -u root -e "show engines;"
```

Como resultado obtendremos un listado en el que el valor "DEFAULT" debe aparecer asociado a la entrada "InnoDB".

## Operaciones finales

Podemos instalar la puesta en marcha del SGBD en los procedimientos de arranque de nuestro sistema, de tal manera que se lance automáticamente cada vez que se inicie el sistema.

Para hacer esto en OpenSUSE debemos ejecutar el siguiente comando:

```
sudo /sbin/insserv mysql
```

---

HASTA AQUÍ SE DESCRIBE EL PROCEDIMIENTO DE INSTALACIÓN QUE PUEDE SER DIFERENTE EN FUNCIÓN DE LA VERSIÓN DE MYSQL Y EL SISTEMA OPERATIVO UTILIZADO

## Finalizando la instalación

### Hacer seguro nuestro SGBD

El SGBD MySQL se instala sin clave para su usuario administrador (“root”). Debemos, por tanto, asignarle una.

Además de esta operación, existen una serie de operaciones adicionales que debemos realizar para asegurar que nuestro SGBD no tiene problemas de seguridad. Por suerte, el entorno instalado nos proporciona un *script* que nos permite realizar todas estas operaciones en un solo paso. Este *script* se denomina `mysql_secure_installation`.

Este script nos planteará una serie de cuestiones. Para todas ellas debemos seleccionar “Yes”

(en Linux basta con pulsar “Enter”), salvo cuando nos pida que introduzcamos una nueva clave para el DBA<sup>1</sup> (“root”), cuestión a la que debemos responder con la clave que le deseamos asignar.

#### Ejercicio 1:

Hacer seguro el SGBD y asignar una clave al DBA siguiendo el procedimiento descrito en este apartado.

---

<sup>1</sup>El administrador de un SGBD se suele denominar DBA (acrónimo de *DataBase Administrator*). Esta es la denominación que utilizaremos para él a partir de ahora.

## Administrando nuestro SGBD

Los usuarios definidos para el SGBD y los definidos en nuestro sistema operativo son independientes. Cuando instalamos MySQL, éste sólo tiene definido un usuario que será el **administrador del SGBD**. El identificador de este usuario es “root” (que no debemos confundir, por tanto, con el usuario “root” del administrador de nuestro sistema operativo) y al finalizar la instalación no tiene clave asignada.

### Introducción

En este punto ya tenemos nuestro entorno operativo y vamos, en primer lugar, a aprender a realizar algunas operaciones de administración básicas y que nos van a hacer falta para continuar con nuestro trabajo.

Las operaciones de administración del SGBD las debemos realizar como DBA (“root”) y, además son las únicas que deberíamos hacer con dicho usuario.

Como indicamos en la introducción, en las primeras prácticas vamos a utilizar como cliente de acceso a nuestro SGBD el cliente *mysql* proporcionado como parte del entorno MySQL. Por tanto, lo primero que debemos aprender es a utilizar dicho programa.

*mysql* es, básicamente un intérprete de comandos interactivo: cada vez que introduzcamos un comando SQL, lo enviará al SGBD, obtendrá una respuesta y nos la presentará. Además de esto nos proporciona algunas funcionalidades adicionales. Es decir, viene a ser para el SGBD lo que es la *shell* para el sistema operativo.

Para ejecutar este cliente utilizaremos el siguiente comando<sup>2</sup>:

```
mysql -h maquina -u usuario -p bd
```

donde

- *maquina* es el nombre de la máquina en que se ejecuta el servidor. Si el servidor se ejecuta en la misma máquina que el cliente, como es nuestro caso, esta opción no es necesaria.
- *usuario* es el nombre de usuario con el que queremos conectarnos a la base de datos. En este ejercicio, dado que vamos a actuar como DBA, deberemos utilizar el valor *root*.
- *bd* es el nombre de la base de datos, de las gestionadas por el servidor, a la que nos queremos conectar. Si no vamos a realizar operaciones sobre ninguna base de datos concreta, como es el caso de este ejercicio, este parámetro se puede omitir.

Como ya indicamos, el programa *mysql* es un intérprete que nos permite introducir, editar y ejecutar de manera interactiva comandos SQL sobre el SGBD. Por tanto, una vez que hayamos

accedido nos presentará un *prompt* y se quedará a la espera de que introduzcamos comandos. Podemos introducir comandos de dos tipos:

1. Comandos SQL: podemos introducir comandos en SQL que se ejecutarán sobre la SGBD a que estamos conectados. Estas sentencias pueden ocupar varias líneas y se ejecutarán en el momento que las terminemos con un “;”. Para obtener ayuda sobre la sintaxis de estos comandos podemos teclear: `help contents`
2. Comandos propios del programa *mysql*, y que, por tanto, no supondrán un acceso al SGBD. Estos comandos se escriben en una sola línea y se distinguen de los anteriores en que empiezan por el símbolo “\”. Para obtener ayuda sobre la sintaxis y significado de estos comandos podemos teclear \? o \h (podemos ver que los propios comandos de petición de ayuda son comandos de este tipo)

### Ejercicio 2:

Conectar al SGBD como DBA utilizando el comando *mysql*.

Probar diferentes comandos propios de *mysql* que pueden ser de interés como pueden ser: \T, \s, \c y, obviamente \q para abandonar el programa.

Comprobar también que el intérprete proporciona algunas facilidades de edición de comandos parecidas a las de la *shell*, como pueden ser: repetir las últimas líneas con la flechas arriba/abajo, etc.

*mysql* va a ser nuestra herramienta básica de trabajo. Debemos dedicar el tiempo suficiente a aprender su funcionamiento.

<sup>2</sup>El comando *mysql* admite muchos otros parámetros. Para consultarlos debemos consultar la página de manual de este comando.

## Gestión de usuarios

Tal como indicamos al comienzo de esta sección, el usuario DBA sólo debe utilizarse para realizar labores administrativas, no como usuario normal que programe y/o utilice las bases de datos gestionadas por el SGBD. Por tanto, en nuestro sistema deben existir otros usuarios que serán los que interactúen con las bases de datos.

La gestión de usuarios se considera una labor administrativa y, por tanto, debemos realizarla como DBA. SQL nos proporciona tres comandos para gestionar usuarios<sup>3</sup>:

- **CREATE USER:** para crear usuarios.

---

<sup>3</sup>SQL no distingue entre minúsculas y mayúsculas. En los enunciados utilizaremos mayúsculas para distinguir las palabras reservadas que forman parte de un comando.

- **ALTER USER:** para modificar determinados atributos de un usuario existente.
- **DROP USER:** para eliminar usuarios.

### Ejercicio 3:

Crear, al menos, un nuevo usuario en nuestro SGBD. Dicho usuario debe poder acceder desde cualquier máquina<sup>4</sup> y tener una clave asignada.

---

<sup>4</sup>Para que un usuario pueda acceder desde cualquier máquina al SGBD se debe utilizar este comando para crear dos usuarios en el SGBD. Así, por ejemplo, para el usuario "mramos" se deben crear los usuarios "mramos" y "mramos@localhost".

*Consideramos que la mejor manera de aprender las posibilidades y utilización de cada comando SQL es analizando su sintaxis. Por tanto, salvo en casos concretos, en los enunciados de prácticas no indicaremos el comando completo que se debe teclear para realizar una operación. La sintaxis y descripción de un comando se puede obtener utilizando el comando **help** de **mysql** seguido del nombre del comando que queremos consultar (por ejemplo: **help create user**).*

## Creación de bases de datos

Los usuarios normales pueden realizar operaciones de DDL y DML sobre las bases de datos existentes, pero no crear nuevas bases de datos o eliminarlas del sistema. Éstas también se consideran labores administrativas y, por tanto, debemos realizarlas como DBA.

SQL nos proporciona tres comandos básicos para gestionar bases de datos:

- **CREATE DATABASE:** para crear bases de datos.
- **ALTER DATABASE:** para modificar determinados atributos de una base de datos existente.
- **DROP DATABASE:** para eliminar bases de datos.

### Ejercicio 4:

Crear, al menos, una nueva base de datos en el sistema que se llame "practicas"

## Gestión de permisos

Los SGBD actuales utilizan un sistema complejo de control de operaciones permitidas a cada usuario sobre cada objeto basado en la asignación de diferentes permisos y roles a los usuarios.

Inicialmente sólo el usuario DBA puede gestionar los permisos, aunque luego el control de los mismos sobre determinados objetos se puede ir delegando en otros usuarios.

SQL nos proporciona dos comandos para gestionar bases de datos:

- GRANT: para asignar permisos.
- REVOKE: para revocar permisos.

### Ejercicio 5:

Asignar al usuario creado en el ejercicio 3 permisos para realizar cualquier operación (privilegio "ALL") sobre cualquier objeto de la base de datos "practicas" creada en el ejercicio 4.

## Utilizando nuestras bases de datos

### Introducción

Las operaciones que hemos realizado como DBA en el apartado anterior nos han permitido crear un entorno operativo sobre el que trabajaremos habitualmente a partir de ahora en esta y las siguientes prácticas.

En este punto dejaremos de ver el SGBD como un administrador y nos conectaremos como usuario normal (con el usuario creado en la práctica 3) a la base de datos "practicas" (creada en la práctica 4) para interactuar con ella como

programador de bases de datos (introduciendo comandos DDL de SQL) y usuario avanzado de bases de datos (ejecutando comandos DML de SQL).

En concreto, en lo que resta de esta práctica aprenderemos a realizar las operaciones más básicas que nos permitan crear tablas y relaciones (DDL), así como introducir datos y consultarlos (DML).

### Crear tablas

Como ya sabemos, una base de datos relacional consiste en un conjunto de relaciones. En el punto en que nos encontramos nuestra base de datos está vacía: no contiene ninguna relación. Por tanto, nuestra primera operación debe ser crear, al menos, una relación.

Las bases de datos SQL pueden almacenar no sólo relaciones, sino que también permiten crear tablas, aunque éstas van a ser estructuras que no van a tener demasiado interés para nosotros.

Tanto las tablas como las relaciones se crean en SQL con el comando CREATE TABLE.

### Ejercicio 6:

Crea dos tablas denominadas "Profesores" y "Alumnos" con los atributos que estimes oportuno. Para crear una tabla debes utilizar solamente los campos obligatorios del comando. Es decir, su sintaxis más sencilla.



## Introducir datos

Para introducir datos en una tabla se utiliza el comando INSERT.

### Ejercicio 7:

Introduce varias filas de datos en tus tablas. Introduce varias filas repetidas para comprobar que las tablas lo permiten.

## Consultar datos

En SQL todas las consultas a una base de se realizan con una sola sentencia: SELECT. Esta es, con diferencia, la sentencia más compleja de SQL. Por ahora la utilizaremos en su forma más sencilla que nos permite obtener el contenido completo de una tabla:

```
SELECT * FROM nombre_tabla;
```

### Ejercicio 8:

Consulta los datos almacenados en tus tablas y comprueba que, efectivamente son tablas.

## Eliminar tablas

Como ya sabemos, a la hora de trabajar con bases de datos relacionales, las tablas son estructuras que carecen de interés para nosotros. Por tanto, vamos a eliminar las tablas que hemos creado.

Para eliminar, tanto tablas como relaciones, en SQL se utiliza el comando DROP TABLE.

### Ejercicio 9:

Elimina las tablas creadas en el ejercicio 6.

## Crear relaciones

En este ejercicio vamos a crear nuestra primera (muy sencilla) base de datos. Para ello crearemos un conjunto de relaciones.

Como ya sabemos, una relación es una tabla que tiene **una clave**, es decir, que no puede tener filas repetidas (ya que no puede haber dos filas con la misma clave)

### Ejercicio 10:

Repite los ejercicios 6, 7 y 8, pero en este caso creando relaciones en vez de tablas. Para ello, crea cada una de ellas con un atributo adicional que contendrá el “codigo” del profesor (un número entero) y que será el campo clave<sup>5</sup>.

Comprueba que, efectivamente se comportan como relaciones.

---

<sup>5</sup>Para introducir una clave utilizaremos la restricción a nivel de columna PRIMARY KEY en la sentencia CREATE TABLE.



## Claves compuestas

### Ejercicio 11:

Prueba ahora a crear una relación en la que la clave esté formada por un conjunto de varios atributos en vez de uno sólo. Para ello debes utilizar la restricción PRIMARY KEY a nivel de tabla en vez de a nivel de columna.

Comprueba que se pueden introducir filas en las que se repitan valores de alguno de los campos que forman la clave, pero no para el conjunto de todos ellos.

## Carga de comandos

Para terminar esta práctica aprenderemos a utilizar el comando de *mysql* “\.”.

Normalmente, la introducción directa de comandos a través del intérprete no es la forma más adecuada de trabajar. Lo normal es crear nuestro programa SQL con todos sus comandos en un fichero de texto, de tal manera que podamos conservarlo y modificarlo adecuadamente. Una vez creado un fichero con diversos comandos SQL podemos cargarlo y ejecutar todos sus comandos desde el intérprete me-

diante el comando “\.” seguido del nombre del fichero.

### Ejercicio 12:

Prueba este modo de trabajo que será el que utilizarás habitualmente.

Para ello crea un programa SQL en un archivo con todos los comandos necesarios para crear las relaciones del ejercicio anterior y posteriormente ejecútalo cargándolo en el intérprete *mysql* desde el sistema de ficheros.

*El programa `mysqldump` permite crear copias de seguridad de nuestras bases de datos en un fichero, almacenando los comandos necesarios para crearlas de nuevo (consulta su uso en la página de manual correspondiente). Los ficheros generados por esta utilidad se pueden recargar en el mismo o en otro SGBD MySQL mediante el comando “\.” de *mysql*.*

## Resumen

Los principales resultados esperados de esta práctica son:

- Disponer del entorno necesario para trabajar con bases de datos MySQL.
- Aprender a interactuar con el SGBD, conociendo y utilizando adecuadamente las posibilidades del cliente *mysql*.
- Ser capaces de crear y utilizar relaciones sencillas.

Como trabajo adicional del alumno, se proponen las siguientes líneas:

- Intentar utilizar todos los comandos propios del programa *mysql* y entender sus usos y posibilidades.
- Experimentar con los comandos SHOW DATABASES, SHOW TABLES y DESCRIBE.
- Aprender a utilizar los parámetros de línea de comandos del programa *mysql*, ya que algunos son muy útiles para realizar determinadas operaciones habituales.
- Analizar y experimentar con los diferentes campos del comando SQL para crear relaciones (CREATE TABLE)

Recuerda que el dominio de las herramientas que hemos utilizado en esta práctica es básico para el desarrollo del resto de prácticas y proyectos de la asignatura