

T04. Introducción a MySQL.

| | |
|---|----|
| Características MySQL..... | 1 |
| Comprobar el servicio..... | 1 |
| Instalación..... | 1 |
| Comprobar el servicio..... | 1 |
| Entrar/salir a MySQL..... | 3 |
| Comprobar usuarios MySQL..... | 3 |
| Comandos útiles..... | 3 |
| Usuarios..... | 4 |
| Scripts..... | 5 |
| Lenguaje DDL..... | 6 |
| Creación de bases de datos..... | 6 |
| Modificar / Eliminar base de datos..... | 7 |
| Creación de tablas..... | 7 |
| Eliminar tabla..... | 8 |
| Insertar, consultar, modificar y borrar registros de una tabla..... | 8 |
| Insertar..... | 8 |
| Consultar..... | 8 |
| Modificar..... | 9 |
| Borrar..... | 9 |
| Modificar estructura de una tabla..... | 9 |
| Ejercicios..... | 11 |

Características MySQL.

- Es un Gestor de Bases de Datos.
- Gestiona base de datos relacionales. Para usar y gestionar las bases de datos utiliza el lenguaje estándar SQL.
- Es open source: el código está disponible y tiene licencia GNU para uso no comercial.
- Rápido, seguro y fácil de usar. Razón por la que es una de las más usadas en Internet.

MySQL dispone de un interfaz gráfico basado en página web lo que permite acceder desde cualquier equipo: PhpMyAdmin. Las páginas web se sirven con el servidor Apache. Además, también está disponible el entorno gráfico de workbench.

El servicio MySQL se ofrece desde el puerto 3306.

Comprobar el servicio.

Instalación.

```
#apt update
#apt install mysql-server
```

Comprobar el servicio.

```
#systemctl start mysql
#systemctl stop mysql
```

```
#systemctl restart mysql
#systemctl status mysql
```

O como alternativa:

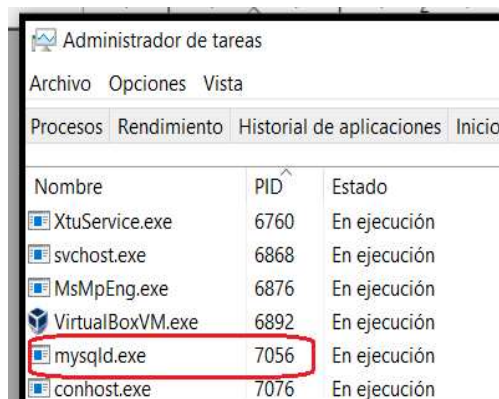
```
#/etc/init.d/mysql stop
#/etc/init.d/mysql start
#/etc/init.d/mysql restart
#/etc/init.d/mysql status
#/etc/init.d/mysql reload
#/etc/init.d/mysql force-reload
```

Como hemos comentado el servidor de MySQL escucha (espera recibir las peticiones de los clientes) por el puerto 3306, por lo que si analizamos los puertos podemos ver si realmente está en funcionamiento. El comando netstat permite hacer un escaneo de puertos y según las opciones que se indiquen nos responderá.

En Windows:

```
c:\> netstat -aon | findstr 3306
```

El listado mostrado nos indica el PID de cada proceso afectado. Para saber cual es el proceso: Ctrl + Alt + Spr / Administrador de tareas / Pestaña detalles. Ordenar por PID y buscar el número indicado.



En Linux:

```
#netstat -h      Mostrar ayuda de opciones
#netstat -ltun   Muestra los números de procesos
#netstat -ltu    Muestra los procesos asociados
```

El significado de las opciones es:

```
-l Listado
-t Puertos TCP
-u Puertos UDP
-n Número de puerto
```

Si netstat no estuviese disponible se instala con:

```
#apt install net-tools
```

Entrar/salir a MySQL.

Entrada:

```
$mysql [-u usuario -p] [-h ipservidormysql]
```

Desde el sistema operativo podemos pedir ayuda para entrar en mysql:

```
$mysql --help
```

Una alternativa para entrar es via telnet:

```
$telnet ipservidormysql 3306
```

Una vez dentro podemos usar para pedir ayuda:

```
mysql>help
```

Para Salir:

```
mysql>quit
```

```
mysql>exit
```

Otra alternativa es Ctrol+D

Comprobar usuarios MySQL.

Dentro de mysql está hay una base de datos “mysql” que contiene información en general. Para utilizar una base de datos primero hay que seleccionarla. Para ello se utiliza el comando “USE”:

```
mysql>USE mysql;
```

Para ver el contenido de una tabla hay que especificar que atributos queremos ver y de que tabla de las que estén definidas en la base de datos seleccionada. Esto se hace con el comando “SELECT”:

```
mysql>SELECT user, host, plugin FROM user;
```

user: el primer user es el nombre de un atributo de la tabla user (segundo user).

Ese comando nos mostraría los usuarios actuales de mysql, desde donde pueden acceder y si la contraseña que utilizan es la del sistema o es propia de mysql.

Comandos útiles.

Podemos distinguir entre comandos y funciones. Un comando es una instrucciones con la que se le indica al SGBD que realice una acción sin esperar ningún dato como resultado de la ejecución de la misma, aunque si pueden mostrar un listado. Ejemplo: crear una base de datos o una tabla, borrar un usuario,.. Sin embargo las funciones son instrucciones que realizan un cálculo y nos devuelve su resultado, como puede ser: ¿quién es el usuario actual?, ¿qué fecha es hoy?,..

Algunos comandos de interés:

SHOW DATABASES;

Muestra las bases de datos disponibles.

SHOW TABLES;

Muestra las tablas de una base de datos.

CREATE DATABASE nombredb;

Crea la base de datos indicada.

DROP DATABASE nombredb;

Borra la base de datos indicada.

| | |
|-----------------------|---|
| USE nombredb; | Selecciona la base de datos indicada. |
| DESCRIBE nombretabla; | Muestra la estructura de una tabla. |
| SHOW CREATE DATABASE; | Muestra la instrucción de crear un base de datos. |
| SHOW CREATE TABLE; | Muestra la instrucción de crear una tabla. |

Algunas funciones de interés:

| | |
|------------------------|---------------------------------------|
| SELECT version(); | Muestra la versión del SO y de MySQL. |
| SELECT current_date(); | Muestra la fecha de hoy. |
| SELECT user(); | Muestra el usuario actual. |
| SELECT now(); | Muestra fecha y hora. |

Usuarios.

Los usuarios de la base de datos han de crearse con una contraseña y definirle los permisos con los que acceden a MySQL.

1. Creación/eliminación de usuarios.

```
mysql>CREATE USER nombre@localhost IDENTIFIED BY 'contraseña';
mysql>DROP USER nombre@localhost;
```

2. Asignación/eliminación de privilegios

```
mysql>GRANT [permiso] ON [basedato].[tabla] TO usuario@localhost;
```

Donde permiso puede ser:

- ALL PRIVILEGES: Permite acceso a todas las bases de datos del sistema.
- CREATE: Permite crear bases de datos y tablas.
- DROP: Permite eliminar bases de datos y tablas.
- DELETE: Permite eliminar registros de tablas.
- INSERT: Permite insertar registros de tablas.
- SELECT: Permite leer registros de tablas.
- UPDATE: Actualiza registros seleccionados de tablas.
- GRANT OPTION: Permite borrar privilegios de usuarios.

Para eliminar los privilegios:

```
mysql>REVOKE [permiso] ON [basedatos].[tabla] FROM usuario@localhost;
```

Para ver los permisos de un usuario determinado:

```
mysql>SHOW GRANTS FOR usuario;
```

3. Después de asignar o eliminar los privilegios es necesario refrescar el sistema. Esto se hace con el comando:

```
mysql>FLUSH PRIVILEGES;
```

Scripts.

Conocer los comandos y las instrucciones que proporciona un SGBD ofrece una visión extra que posibilita automatizar tareas rutinarias y permite solucionar problemas que no se pueden solucionar desde el entorno gráfico.

Los comandos se pueden poner en un fichero de texto plano y se ejecutarán uno detrás de otro, por lo que el orden en el aparezcan es importante (caso de dependencias de claves foráneas, por ejemplo). Esos scripts pueden ser llamados desde el SO o desde el propio entorno de comandos del servidor MySQL.

- Desde el SO:
`$mysql -u usuario -p < nombre_script > resultado`
Donde “resultado” será un fichero donde se almacenará el resultado de la ejecución del script.
- Desde interprete de comandos de mysql:
`mysql> source nombre_script;`

La extensión del script ha de ser .sql

El script se puede documentar utilizando el carácter “#” para indicar que una línea es un comentario.

Otro aspecto importante de gestionar es el envío de un script a otro ordenador. Para ello utilizaremos el comando **scp**, que realiza la transferencia encriptada mediante ssh. Es necesario tener permiso de escritura en el destino.

Sintaxis:

`> scp [opciones] origen destino`

Donde

Opciones:

- P Port permite especificar una entrada diferente al servidor (el puerto predeterminado para el puerto TCP para el comando es 22).
- q Ejecutará la operación en modo silencioso (quiet), lo que significa que solo se mostrarán los errores críticos.
- r Es para copia recursiva, que incluirá todos los subdirectorios.
- 4 ó -6 Se pueden usar si quieres elegir la versión de protocolo empleada, ya sea IPv4 o Ipv6. Por defecto IPv4.
- p Conservará los tiempos de modificación iniciales y los atributos del archivo.
- u Borrará el archivo fuente después de que se complete la transferencia.
- c Permitirá la compresión de los datos mientras se lleva a cabo la operación de transferencia.

Origen o destino: usuario@ordenador:/directorio/nombre_fichero. Si estamos en el directorio o carpeta donde esta el fichero a transferir no hace falta especificar ni el usuario ni el ordenador.

Ejemplos: `>scp veterinariodb.sql joseant@146.59.232.252:/home/joseant/veterinariodb.sql`
`>scp joseant@146.59.232.252:/home/joseant/datos.sql datos.sql`

En el primer caso transferimos a un ordenador remoto el archivo “veterinariodb.sql” y en el segundo caso transferimos desde el ordenador remoto al local “datos.sql”.

Lenguaje DDL.

El lenguaje DDL (Data Definition Language) o Lenguaje de Definición de Datos está formado por los comandos que nos permiten definir los datos en el SGBD, estos son:

- CREATE: permite crear objetos tales como bases de datos o esquemas, tablas e índices.
- DROP: permite eliminar un objeto existente.
- ALTER: permite modificar la definición de un objeto.
- SHOW: permite consultar los objetos de la base de datos.

Otras sentencias o comandos de utilidad:

- USE: con ella especificamos la base de datos con la que se va a trabajar.
- DESCRIBE: permite mostrar información sobre la estructura de una tabla.

Creación de bases de datos.

Sintaxis:

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] nombre_bd  
[especificación_create[, especificación_create]..];
```

Donde especificación_create:

```
[DEFAULT] CHARACTER SET juego_caracteres  
| [DEFAULT] COLLATE nombre_colación
```

Los términos DATABASE y SCHEMA son sinónimos, solo se mantienen por compatibilidad con otros SGBD.

La base de datos se creará si no existe. Pero, ¿qué juego de caracteres utilizará? y ¿cómo se van a ordenar los símbolos de esos caracteres?, ¿es lo mismo “á” que “a” o “A”? Eso es lo que se define con el juego de caracteres y la colación. Se puede ver un listado completo de todas las disponibles:

```
mysql> SHOW CHARACTER SET;  
mysql> SHOW COLLATION;
```

El juego de caracteres si no se especifica se toma por defecto **latin1**.

La colación puede ser:

- case-sensitive (_cs): las mayúsculas y las minúsculas las trata distintas.
- case-insensitive (_ci): las mayúsculas y las minúsculas las trata igual.
- binary (_bin): dos caracteres son iguales si los valores de su representación numérica son iguales.

Utilizaremos el juego de caracteres: latin1 y la colación: latin1_spanish_ci

Para mostrar las bases de datos activas en el servidor se utiliza el comando:

```
mysql> SHOW DATABASES;
```

Por defecto aparecen las bases de datos: mysql y information_schema, estas contienen metadatos de los usuarios y permisos del SGBD.

Para usar una base de datos primero se pone en uso con el comando:

```
mysql> USE nombredb;
```

Se puede ver que comando se ha utilizado para crear una determinada base de datos:

```
mysql> SHOW CREATE DATABASE nombre_bd;
```

Modificar / Eliminar base de datos.

Solo se permite modificar el juego de caracteres o la colación. La sintaxis es:

```
mysql> ALTER DATABASE nombre_bd COLLATE nombre_colación;
```

```
mysql> ALTER DATABASE nombre_bd CHARACTER SET nombre_caracteres;
```

Para eliminar una base de datos:

```
mysql> DROP {DATABASE | SCHEMA} [IF EXISTS] nombre_bd;
```

Creación de tablas.

Sintaxis.

```
CREATE TABLE [IF NOT EXISTS] nombre_tabla  
  (descripcion_campos)  
  [opciones];
```

Descripcion_campos, cada campo requiere una estructura como la siguiente y van separados por comas:

```
nombre_campo tipo_dato [NOT NULL | NULL] [DEFAULT valor] [AUTO_INCREMENT]  
[UNIQUE [KEY] | [PRIMARY] KEY]
```

Si hay claves compuestas, por estilo se recomienda ponerlo detrás de la definición de todos los campos:

```
[CONSTRAINT [nombre_constraint]] PRIMARY KEY (nombre_campo1,..)
```

Si hay claves foráneas, por estilo se recomienda ponerlo detrás de la definición de los campos y de la clave primaria si fuese compuesta:

```
[CONSTRAINT [nombre_constraint]] FOREIGN KEY (nombre_campo1,..)  
REFERENCES nombre_tabla (nombre_campo1,..)  
[ON DELETE opcion_referencia]  
[ON UPDATE opcion_referencia]
```

Opcion_referencia:

```
RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
```

RESTRICT: usado por defecto por MySQL. Impide que se puedan actualizar o eliminar las filas de los valores referenciados por las claves ajenas.

CASCADE: Permite actualizar o eliminar las filas que tienen valores referenciados por las claves ajenas.

SET NULL: Asigna NULL a las filas que tienen valores referenciados por claves ajenas.

NO ACTION: en MySQL es equivalente a RESTRICT.

SET DEFAULT: no es posible usar esta opción si se utiliza el motor de almacenamiento InnoDB.

Opciones, al definir una tabla se pueden establecer las siguientes opciones:

AUTO_INCREMENT [=] valor

| [DEFAULT] CHARACTER SET [=] character_name

| [DEFAULT] COLLATE [=] collation_name

| ENGINE [=] engine_name

AUTO_INCREMENT: especifica el valor inicial que se usará en un campo

AUTO_INCREMENT.

CHARACTER SET: especifica el juego de caracteres que usará la tabla.

COLLATE: especifica el tipo de cotejamiento que vamos a usar en la tabla.

ENGINE: especifica el tipo de almacenamiento que tendrá la tabla. Los más habituales en MySQL son InnoDB y MyISAM. Por defecto es InnoDB.

Eliminar tabla.

DROP TABLE [IF EXISTS] nombre_tabla;

Se eliminará la tabla especificada con “nombre_tabla”.

Es conveniente proteger la sentencia con la opción: “IF EXISTS”, sobre todo en procesos automáticos y/o en scripts, de otra forma daría error si no existe la tabla.

Insertar, consultar, modificar y borrar registros de una tabla.

Insertar.

El comando para introducir datos en una tabla es INSERT INTO que pertenece al lenguaje de manipulación de datos, DML. Se verá con detalle más adelante. Pero, para poder introducir algunos datos en las tablas que creamos y hacer pruebas, vamos a ver una introducción:

INSERT INTO nombre_tabla [(campo1, campo2,...)] VALUES (valor1, valor2,...);

Donde la lista de campos: campo1, campo2,.. son campos de la tabla y la lista de valores: valor1, valor2,.. la lista de valores que se le asignaran a esos campos por orden de aparición en el nuevo registro.

Si la lista de campos no aparece, entonces en la lista de valores han de aparecer todos los campos de la tabla y en el mismo orden en el que se definieron.

Consultar.

De la misma forma el comando para hacer las consultas es: SELECT, que merece una atención especial por la variedad de opciones que ofrece. Para adelantar de forma simplificada su funcionamiento:


```
SELECT {campo1, campo2,.. | *} FROM nombre_tabla;
```

Si se especifica la lista de campos, mostrará solo esos campos y en el orden indicado. Con el “*” muestra todos los campos de la tabla.

Modificar.

Si necesitamos modificar algún dato de la tabla se utiliza el comando UPDATE. Cuya sintaxis abreviada es:

```
UPDATE nombre_tabla SET nombre_atributo = valor [, nombre_atributo = valor]..  
[WHERE condición]
```

En una instrucción se pueden cambiar más de un atributo a la vez, habría que separar por coma cada pareja de “atributo=valor”.

Si no se especifica WHERE, se cambiarán todos los registros de la tabla.

Ejemplos:

```
UPDATE tmascota SET raza=“podenco” WHERE codigo=12;
```

Cambiará el contenido de “raza” a “podenco” del registro cuyo “codigo” es “12” en la tabla “tmascota”.

Si queremos que el cambio afecte a un conjunto de registros podemos especificar la condición IN (lista): WHERE codigo IN (2,4,5,10); cambiaría “raza” a esos cuatro registros.

Borrar.

La instrucción DELETE nos permite borrar registros de una tabla. Se debe de usar siempre con la cláusula “WHERE condición”, pues si no se hace así se eliminarán todos los registros de la tabla.

La sintaxis es como sigue;

```
DELETE FROM nombre_tabla [WHERE condición];
```

Ejemplos:

```
mysql> DELETE FROM mascota;
```

```
mysql> DELETE FROM mascota WHERE nombre = “Misifú”;
```

En el primer caso borraría todos los registros de la tabla mascota. Y en el segundo solo borraría el registro o registros que tengan en el atributo nombre el valor “Misifú”.

Modificar estructura de una tabla.

En muchas ocasiones hace falta modificar los atributos de una tabla: añadir o eliminar algún campo, cambiar el tipo de dato o su longitud, establecer o eliminar alguna clave foránea,.. Si la tabla no tiene datos se podría eliminar y volver a crearla, pero si tiene datos se utiliza la sentencia ALTER TABLE y no se pierde la información.

Sintaxis:

```
ALTER TABLE nombre_tabla
```

[especificación [, especificación]..]

Especificaciones (usar solo una),:

- ADD [COLUMN] nombre_columna definicion_columna [FIRST | AFTER nombre_columna]
- ADD {INDEX|KEY} [nombre_indice] [tipo_indice] (columna_indice)
- ADD [CONSTRAINT [simbolo]] PRIMARY KEY [tipo_indice] (**columna_indice**)
- ADD [CONSTRAINT [simbolo]] FOREIGN KEY [nombre_indice] (**columna_indice**,..) REFERENCES nombre_tabla (nombre_campo1,..)
[ON DELETE opcion_referencia]
[ON UPDATE opcion_referencia]
- ALTER [COLUMN] nombre_columna {SET DEFAULT literal | DROP DEFAULT}
- CHANGE [COLUMN] nombre_columna_anterior nombre_columna_nuevo definicion_columna [FIRST | AFTER nombre_columna]
- [DEFAULT] CHARACTER SET [=] nombre_caracteres [COLLATE [=] nombre_collate]
- CONVERT TO CHARACTER SET nombre_caracteres [COLLATE nombre_collate]
- DROP [COLUMN] nombre_columna
- DROP {INDEX | KEY} nombre_indice
- DROP PRIMARY KEY
- DROP FOREIGN KEY fk_simbolo
- MODIFY [COLUMN] nombre_columna definicion_columna [FIRST | AFTER nombre_columna]
- ORDER BY nombre_columna [, nombre_columna]..
- RENAME {INDEX | KEY} nombre_indice_anterior nombre_indice_nuevo
- RENAME [TO | AS] nuevo_nombre_tabla

Donde **columna_indice** será:

nombre_columna [(longitud)] [ASC | DESC]

Tipo_indice:

USING {BTREE | HASH}

Para más información utilizar la información oficial de MySQL.

Ejercicios.

1. Sobre un servidor MySQL en el que seas administrador realiza las siguientes operaciones:
 - a) Crea tres usuarios, cuyos nombres sean personajes históricos.
 - b) Crea dos bases de datos, cuyos nombres sean barrios de tu localidad de origen.
 - c) Asigna cada una de las bases de datos creadas a dos de los personajes (una a cada uno) y al tercero asigna las dos bases de datos.
 - d) Comprueba que están creadas las bases de datos y los usuarios.
 - e) Sal de MySQL y entra con cada usuario y comprueba que puede ver solo las bases de datos que les has asignado.
 - f) Elimina un usuario y una base de datos.
 - g) Comprueba que se han realizado las eliminaciones.
2. Crea una base de datos que se llame: Veterinario y en ella una tabla que se llame mascota con los siguientes campos:
 - a) Código: numérico y campo clave.
 - b) Nombre: carácter de hasta 20 letras.
 - c) Raza: carácter de 15 letras
 - d) Fecha nacimiento: fecha.
 - e) Nombre del dueño: carácter de hasta 25 letras.

Realiza los siguientes trabajos:

- Añade al menos 5 registros.
 - Consulta todas las mascotas y todos los campos.
 - Consulta solo nombre de mascota y nombre del dueño.
 - Consulta solo nombre de mascota, fecha de nacimiento y nombre de dueño.
3. Modifica el nombre del dueño y fecha de nacimiento de alguna mascota.
 4. Cambia el nombre de la raza de tres mascotas a la vez.
 5. Cambia el nombre de un dueño a “Juan Antonio Rodríguez Fernández”. Si da error modifica el diseño de la tabla ampliando el nombre del dueño a 45 caracteres y vuelve a intentarlo.
 6. Crea una tabla para los dueños de las mascotas, esta tabla se llamará: “amo” y tendrá las siguientes columnas:
 - a) Código_amo: numérico, clave principal y auto incremental.
 - b) Nombre: carácter de 45 letras.
 - c) Dirección: carácter de 50 letras.
 - d) Tfno: carácter de 10 letras.
 7. Toma nota de los códigos de los dueños y modifica la columna de las mascotas para que albergue los códigos de los dueños en lugar de los nombres y actualiza de forma que sea coherente.
 8. Modifica el campo código de dueño de las mascotas para que sea clave foránea. De forma que si se cambia el código del dueño se actualice automáticamente en las mascotas y que no permita eliminar un dueño si tiene mascota.

9. Añade nuevas mascotas con todos los datos. Prueba códigos de dueño que existan y que no existan y analiza los resultados.
10. Cambia el código de algún dueño y comprueba si se ha actualizado en las mascotas.
11. Intenta borrar algún dueño que tenga mascota. ¿Qué ocurre? ¿por qué?