

Comandos Básicos de SQL

Primeros pasos

Temas:

1. [Conexión al motor de MySQL](#)
2. [SHOW, CREATE, USE y DROP DATABASE](#)
3. [CREATE, DROP, SHOW, DESCRIBE, TRUNCATE TABLES](#)
4. [Instrucción INSERT](#)
5. [Instrucción UPDATE](#)
6. [Instrucción DELETE](#)
7. [Instrucción ALTER TABLE](#)

Todos los ejercicios están realizados en los contenedores que estamos utilizando en clases.

v1_23.5.26

Conexión al motor de MySQL [\(volver\)](#)

Utilizaremos el usuario del contenedor, root y clave root, pero antes que eso debemos ingresar al contenedor. (ver apunte de comandos docker)

```
bash-4.2# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.39 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

-u especifica el nombre de usuario y -p es para la password al presionar ENTER nos pide la pass, el prompt mysql> nos indica que ya podemos escribir en la consola de MySQL, si fuera un servidor remoto agregamos -h indicando la ip o nombre del servidor:

```
$~ mysql -h host -u user -p
Enter password: ****
```

SHOW, CREATE, USE y DROP DATABASE [\(volver\)](#)

Una vez que nos conectamos podemos ver las bases de datos a las que tenemos acceso

```
mysql> SHOW DATABASES;
+-----+
| Database                |
+-----+
| information_schema      |
| espacios_culturales    |
| mysql                   |
| performance_schema     |
| sys                     |
+-----+
5 rows in set (0.00 sec)

mysql>
```

Ahora creamos una DB de prueba para los ejemplos, luego con USE seleccionamos nuestra DB. El comando USE se utiliza para seleccionar la base de datos que queremos utilizar

```
mysql> CREATE DATABASE ejemplo DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> USE ejemplo;
Database changed
mysql>
```

```
mysql> DROP DATABASE ejemplo;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>
```

DROP DATABASE: Elimina la base de datos

SHOW DATABASES: me muestra todas las bases de datos existentes

CREATE DATABASE: Crea una nueva base de datos (no olvidar especificar CHARACTER y COLLATE)

USE nombre_db: Se utiliza para seleccionar una base de datos escribiendo su nombre

CREATE, DROP, SHOW, DESCRIBE, TRUNCATE TABLE [\(volver\)](#)

Crearemos un pequeño ejemplo de base de datos llamado “colegio” con 3 tablas “aulas”, “alumnos” y “aulas_alumnos”

```
mysql> CREATE DATABASE colegio DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> USE colegio;
Database changed
```

```
mysql> CREATE TABLE `alumnos` (
  -> `id` INT NOT NULL AUTO_INCREMENT ,
  -> `nombre` VARCHAR(50) NOT NULL ,
  -> `apellido` VARCHAR(50) NOT NULL ,
  -> `dni` VARCHAR(10) NOT NULL ,
  -> PRIMARY KEY (`id`),
  -> UNIQUE (`dni`)
  -> ) ENGINE = InnoDB COMMENT = 'Tabla de Alumnos';
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE `aulas` (
  -> `id` INT NOT NULL AUTO_INCREMENT ,
  -> `materia` VARCHAR(50) NOT NULL ,
  -> `catn_mesas` TINYINT UNSIGNED NOT NULL DEFAULT '30' ,
  -> `cant_sillas` TINYINT NOT NULL DEFAULT '60' ,
  -> `profesor_nombre` VARCHAR(50) NULL DEFAULT NULL ,
  -> `ultima_limpieza` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ,
  -> PRIMARY KEY (`id`)
  -> ) ENGINE = InnoDB;
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE `aulas_alumnos` (
```

```
-> `id` INT NOT NULL AUTO_INCREMENT ,
-> `alumno_id` INT NOT NULL ,
-> `aula_id` INT NOT NULL ,
-> PRIMARY KEY (`id`)
-> ) ENGINE = InnoDB COMMENT = 'Relacion entre alumnos y un aula';
Query OK, 0 rows affected (0.03 sec)
```

Luego de crear las tabla podremos verlas de la siguiente manera

```
mysql> SHOW TABLES;
+-----+
| Tables_in_colegio |
+-----+
| alumnos           |
| aulas             |
| aulas_alumnos     |
+-----+
3 rows in set (0.00 sec)
```

También podemos ver cómo están compuestas esas tablas

```
mysql> DESCRIBE alumnos;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| nombre     | varchar(50)   | NO   |     | NULL    |                |
| apellido   | varchar(50)   | NO   |     | NULL    |                |
| dni        | varchar(10)   | NO   | UNI | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

La instrucción **DROP** la utilizamos para eliminar la tabla

La instrucción **TRUNCATE** sirve para vaciar la tabla

Instrucción INSERT [\(volver\)](#)

Usamos **INSERT INTO** para insertar datos a una tabla la sintaxis sería:

```
INSERT INTO `nombre_tabla` (`campo1`, `campo2`, `campo3`) VALUES ("valor1","valor2","valor3");
```

otra forma puede ser un INSERT multiple

```
INSERT INTO `nombre_tabla` (`campo1`, `campo2`, `campo3`)
VALUES ("valor1","valor2","valor3"),("valor1","valor2","valor3"),("valor1","valor2","valor3");
```

```
mysql> INSERT INTO `alumnos`(`nombre`, `apellido`, `dni`) VALUES
```

```
("Andres", "Romano", "11222333");  
Query OK, 1 row affected (0.00 sec)  
  
mysql> INSERT INTO `aulas`(`materia`) VALUES ("bases de datos");  
Query OK, 1 row affected (0.01 sec)
```

Nota: como se ve los nombres de los campos van separados por , y los valores también. Se debe respetar el orden y si hay campos que tengan valor por DEFAULT esos no los especificamos ni en los campos ni en los valores a menos que necesite cambiar ese valor entonces si, los especifico en campos y en valores

Instrucción UPDATE [\(volver\)](#)

Usamos **UPDATE** para actualizar el valor de uno o más campos de una tabla, la sintaxis sería:

UPDATE nombre_tabla SET nombre_tabla.nombre_campo = "nuevo_valor" WHERE nombre_tabla.id = value

```
mysql> UPDATE alumnos SET alumnos.nombre = "Andres Dario" WHERE alumnos.id =  
1;  
Query OK, 1 row affected (0.00 sec)  
Rows matched: 1  Changed: 1  Warnings: 0  
  
# aquí actualizamos el valor NULL que se grabó en nombre_profesor  
  
mysql> UPDATE aulas SET aulas.profesor_nombre = "Fabian Lopez" WHERE aulas.id  
= 1;  
Query OK, 1 row affected (0.00 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

Nota: Acá es muy importante no olvidarse del WHERE (ósea DONDE) se utiliza WHERE para especificar dónde se va a actualizar el o los datos, en el ejemplo vemos que se actualiza el nombre especificando al alumno con id = 1. En el caso de que me olvide la cláusula WHERE la modificación se aplicará a toda la tabla, ¡hay que tener cuidado con esto!

Instrucción DELETE [\(volver\)](#)

Usamos **DELETE** para eliminar un registro la sintaxis sería:

DELETE FROM `nombre_tabla` WHERE `id` = value

```
mysql> DELETE FROM `alumnos` WHERE alumno.id = 1
```

Nota: Acá también es muy importante el WHERE y también se utiliza el id para especificar que registro voy a eliminar. No es una práctica muy habitual, dependiendo de la información que sea que vamos a eliminar a veces es mejor utilizar el campo de "Estado" y marcar ese registro con un estado que indique que no se utiliza

Instrucción ALTER TABLE [\(volver\)](#)

La sentencia **ALTER** nos permite realizar modificaciones, vamos a modificar la tabla “alumnos” agregando más campos, utilizando **ADD**, por ejemplo el campo “email”. También utilizaremos el comando **AFTER** para especificar después de... indicando así la ubicación del nuevo campo. Haremos modificaciones utilizando **CHANGE** y **MODIFY** y también agregaremos índices en la tabla aulas_alumnos con sus correspondientes restricciones **CONSTRAINT**. También podemos combinar **ALTER** con **DROP**, para eliminar alguna columna o de ser requerido eliminar un índice (llave foránea)

```
# agregamos el campo email despues del campo dni
```

```
mysql> ALTER TABLE alumnos ADD email VARCHAR(255) NULL DEFAULT NULL AFTER dni;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
# agreguemos también la fecha en que se creó el registro
```

```
mysql> ALTER TABLE alumnos ADD fecha_de_alta TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP AFTER `email`;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
# ahora agregamos como índice los campos alumno_id y aula_id
# de la tabla aulas_alumnos, para poder usarlos como FOREIGN KEY
```

```
mysql> ALTER TABLE aulas_alumnos ADD INDEX (`alumno_id`);
Query OK, 0 rows affected, 1 warning (0.01 sec)
Records: 1 Duplicates: 0 Warnings: 0
```

```
mysql> ALTER TABLE aulas_alumnos ADD INDEX (`aula_id`);
Query OK, 0 rows affected, 1 warning (0.01 sec)
Records: 1 Duplicates: 0 Warnings: 0
```

```
# renombramos la columna dni a documento y la transformamos en INT
# primero se pone el nombre original de la columna y luego la defino
# con su nuevo nombre y tipo
```

```
mysql> ALTER TABLE alumnos CHANGE dni documento INT NOT NULL;
Query OK, 3 rows affected (0.04 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
# modificamos una columna especificando su nuevo tipo
```

```
mysql> ALTER TABLE alumnos MODIFY nombre VARCHAR(100) NOT NULL;
Query OK, 3 rows affected (0.04 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
# y no nos olvidemos del campo de estado, podríamos necesitarlo más adelante
```

```
mysql> ALTER TABLE alumnos ADD estado BOOLEAN NOT NULL DEFAULT TRUE AFTER
email;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
# ahora agregamos las restricciones correspondientes con las llaves foráneas
```

```
mysql> ALTER TABLE `aulas_alumnos` ADD CONSTRAINT `fk_alumno_id` FOREIGN KEY
```

```
(`alumno_id`) REFERENCES `alumnos` (`id`) ON DELETE RESTRICT ON UPDATE CASCADE;
```

Query OK, 0 rows affected (0.04 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> ALTER TABLE `aulas_alumnos` ADD CONSTRAINT `fk_aula_id` FOREIGN KEY (`aula_id`) REFERENCES `aulas` (`id`) ON DELETE RESTRICT ON UPDATE CASCADE;
```

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

eliminar una columna

```
mysql> ALTER TABLE aulas DROP COLUMN profesor_nombre;
```

Query OK, 0 rows affected (0.06 sec)

Records: 0 Duplicates: 0 Warnings: 0

eliminar una FOREIGN KEY, si no recuerdan el nombre pueden ejecutar:

*# mysql> SELECT * FROM information_schema.table_constraints WHERE*

constraint_schema = 'colegio';

```
mysql> ALTER TABLE aulas_alumnos DROP FOREIGN KEY fk_alumno_id;
```

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates: 0 Warnings: 0

eliminar un INDEX

```
mysql> ALTER TABLE aulas_alumnos DROP INDEX alumno_id;
```

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates: 0 Warnings: 0

renombrar la tabla

```
mysql> ALTER TABLE aulas_alumnos RENAME alumnos_aulas;
```

Query OK, 0 rows affected (0.00 sec)