

## Creación de un contenedor de Docker para nuestra base de datos en MySQL

La siguiente línea de comandos muestra los pasos a seguir para generar nuestra base de datos para nuestra aplicación web, primero obtenemos la imagen de MySQL, luego generamos un volumen en el que se almacenaran nuestros datos y finalmente creamos el servidor de MySQL

Indicaremos nuestro usuario y contraseña de acceso

```
$> sudo docker pull mysql/mysql-server

$> sudo docker volume create mysql-8-data

$> sudo docker run \

  -p 3307:3306 \

  -v mysql-8-data:/var/lib/mysql \

  -e MYSQL_ROOT_PASSWORD=XXXX -e MYSQL_USER=XXXX -e MYSQL_DATABASE=Appweb \

  --name mysql-8 \

  --restart always \

  -d mysql/mysql-server:latest \
```

Una vez que hayamos iniciado nuestro contenedor servidor podremos iniciar nuestra base de datos

```
mysql -h localhost -u root -pmi_clav
```

Una vez dentro de la terminal de mysql generaremos nuestra base de datos “Appweb” con el siguiente comando **CREATE DATABASE Appweb;**

Con el comando **SHOW DATABASES;** podemos comprobar como se ha generado nuestra base de datos, finalmente accedemos a ella **USE Appweb;**

```
mysql> CREATE DATABASE Appweb;
Query OK, 1 row affected (0.04 sec)

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

mysql> USE Appweb;
Database changed
```

una vez que nos encontremos en nuestra base de datos crearemos las tablas que necesitaremos para nuestro proyecto web

primero crearemos la tabla de user, así como sus columnas correspondientes en este caso serán: iduser(PK), name, subname1, subname2, bdate, mail, tlf, status.

Para ello introduciremos el siguiente comando en nuestro servidor

```
CREATE TABLE user (iduser int(4) unsigned PRIMARY KEY auto_increment,  
                    name varchar(30) not null,  
                    sname1 varchar(30) not null,  
                    sname2 varchar(30) not null,  
                    bdate date not null,  
                    mail varchar(50),  
                    tlf int(15) not null,  
                    status varchar(30) not null );
```

```
mysql> CREATE TABLE user (iduser int(4) unsigned PRIMARY KEY auto_increment,  
-> name varchar(30) not null,  
-> sname1 varchar(30) not null,  
-> sname2 varchar(30) not null,  
-> bdate date not null,  
-> mail varchar(50),  
-> tlf int(15) not null,  
-> status varchar(30) not null);  
Query OK, 0 rows affected, 2 warnings (0.12 sec)
```

Luego generaremos la tabla de usertype con sus correspondientes columnas en este caso serán: iduser(PK, FK), type

```
CREATE TABLE usertype (idusuario int(4) unsigned PRIMARY KEY,  
                        FOREIGN KEY (iduser) REFERENCES user (iduser),  
                        type varchar(30) not null);
```

```
mysql> CREATE TABLE usertype (iduser int(4) unsigned PRIMARY KEY auto_increment,  
-> FOREIGN KEY (iduser) REFERENCES user (iduser),  
-> type varchar(30) not null);  
Query OK, 0 rows affected, 1 warning (0.17 sec)
```

Finalmente generaremos la tabla de userarc con sus correspondientes columnas en este caso serán: iduser(PK, FK), pass

```
CREATE TABLE usertype (idusuario int(4) unsigned PRIMARY KEY,  
                        FOREIGN KEY (iduser) REFERENCES user (iduser),  
                        pass varchar(30) not null);
```

```
mysql> CREATE TABLE useracs (iduser int(4) unsigned PRIMARY KEY auto_increment,  
-> FOREIGN KEY (iduser) REFERENCES user (iduser),  
-> pass varchar(30) not null);  
Query OK, 0 rows affected, 1 warning (0.10 sec)
```

Finalmente con la ayuda del comando **SHOW COLUMNS FROM <nombre\_tabla>** comprobaremos la creación de las anteriores tablas.

```
mysql> SHOW COLUMNS FROM user;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| iduser | int unsigned | NO   | PRI | NULL    | auto_increment |  
| name   | varchar(30)  | NO   |     | NULL    |                 |  
| sname1 | varchar(30)  | NO   |     | NULL    |                 |  
| sname2 | varchar(30)  | NO   |     | NULL    |                 |  
| bdate  | date         | NO   |     | NULL    |                 |  
| mail   | varchar(50)  | YES  |     | NULL    |                 |  
| tlf    | int          | NO   |     | NULL    |                 |  
| satatus | varchar(30)  | NO   |     | NULL    |                 |  
+-----+-----+-----+-----+-----+-----+  
8 rows in set (0.00 sec)  
  
mysql> SHOW COLUMNS FROM usertype;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| iduser | int unsigned | NO   | PRI | NULL    | auto_increment |  
| type   | varchar(30)  | NO   |     | NULL    |                 |  
+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.01 sec)  
  
mysql> SHOW COLUMNS FROM useracs;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| iduser | int unsigned | NO   | PRI | NULL    | auto_increment |  
| pass   | varchar(30)  | NO   |     | NULL    |                 |  
+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.01 sec)
```

En caso de querer añadir o modificar algún dato de nuestra base de datos podemos hacer uso de los siguientes comandos

ELIMINAR BASE DE DATOS

```
DROP DATABASE <NOMBRE_baseDATOS>;
```

INTRODUCIR NUEVAS FILAS EN LA TABLA

```
INSERT INTO...
```

BORRAR FILAS EN LA TABLA (IMPORTANTE NO OLVIDARSE NUNCA DEL WHERE)

```
DELETE FROM...
```

MODIFICAR FILAS EN LA TABLA

```
UPDATE...
```