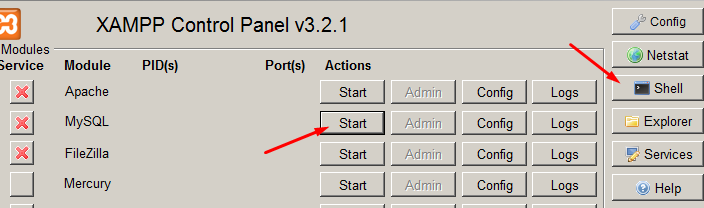
Se accedió al programa xaamp, se activo el Mysql y a su vez se ejecutó “El terminal”



Luego a través de los siguientes comando se importó la base de datos Shakila.sql y a su vez se los datos que esta contiene.

Dentro del terminal o consola, se introdujo la secuencia que indica que queremos conectarnos a MySQL con el usuario root por defecto que y sin contraseña .

**# mysql -u root –p**

**MariaDB [(none)]>**

Del archivo que nos hemos descargado de la base de datos Sakila, vemos que hay dos archivos, “sakila-schema.sql” corresponde a la estructura de la base de datos y “sakila-data.sql” que corresponden a los datos. Estos se compiaron en el disco C:/ del ordenador.

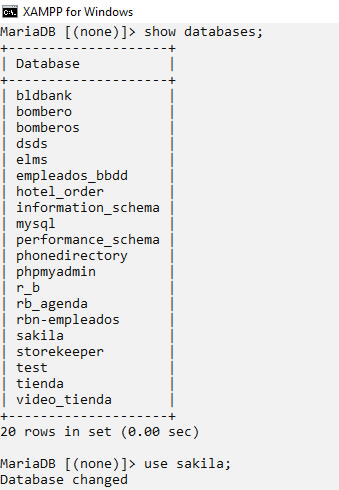
Lo primero será importar la estructura, para lo cual teniendo el archivo “sakila-schema.sql” descargado en C:/

**MariaDB [(none)]> SOURCE C:/sakila-schema.sql;**

Luego, lo mismo para los datos poniendo el nombre de fichero correspondiente en la misma ubicación de C:/

**MariaDB [(none)]> SOURCE D:/sakila-data.sql;**

Una vez importado los dos archivos, se ha creado la base de datos shakila con toda su estructura y datos en su interior

Creacion de tablas y campos:

Se creo una tabla con sus compos y contenidos en dentro de la base de datos Sakila que lleva por nombre “Artistas\_CD” y que lleva como campos para ser rellenados a futuro: nombre, apellidos, agrupación, email, teléfono y país. Todo ello después de la línea de comando.

**MariaDB [sakila]>**

**CREATE TABLE IF NO EXIST `artista\_cd` (**

**`nombre` varchar(16) NOT NULL,**

**`Apellidos` varchar(30) NOT NULL,**

**`agrupacion` varchar(30) NOT NULL,**

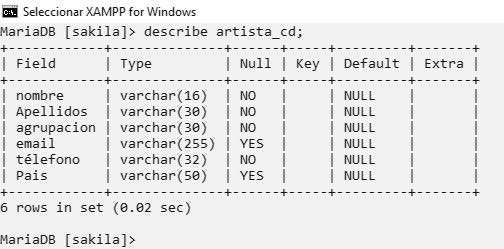
**`email` varchar(255) DEFAULT NULL,**

**`télefono` varchar(32) NOT NULL,**

**`Pais` varchar(50) DEFAULT NULL ) ENGINE=InnoDB DEFAULT CHARSET=UTT-8**

Luego dentro de la línea de comando se escribió la sentencia y la tabla y contenido fue creada con éxito:

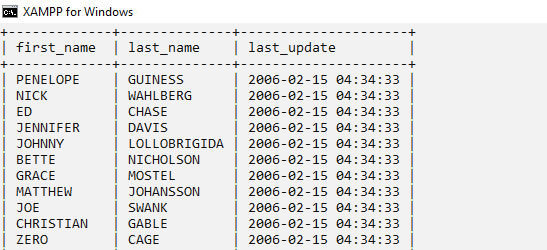
**MariaDB [sakila]> describe artista\_cd;**

****

**CONSULTAS EN LA BASE DE DATOS SAKILA**

Se realizó una consulta básica en la tabla “actor” por “nombre”, “apellidos” y “fecha”

**SELECT first\_name,last\_name, last\_update FROM actor;**

****

**Se realiza una consulta para ver que clientes están en la tabla rental**

**SELECT rental.rental\_id,customer.first\_name,customer.last\_name,rental.rental\_date,rental.return\_date**

**FROM rental**

**INNER JOIN customer**

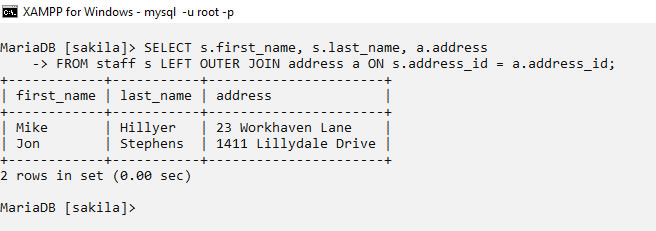
**ON rental.customer\_id=customer.customer\_id**

**JOIN**

Se utilizo el Joint para la consulta de los nombres y apellidos, así como la dirección de cada miembro del personal y nombre del personal de staff.

**SELECT s.first\_name, s.last\_name, a.address**

**FROM staff s LEFT OUTER JOIN address a ON s.address\_id = a.address\_id;**

****

**INNER JOIN**

Muchas tiendas de DVD producen una lista diaria de alquileres vencidos para que los clientes puedan ser contactados y pedirles que devuelvan sus DVD vencidos.

Para una consulta de este tipo, se buscan películas en la tabla de alquiler con una fecha de retorno NULA y donde la fecha de alquiler sea más antigua que la duración del alquiler especificada en la tabla de películas. Si es así, la película está vencida y debemos producir el nombre de la película junto con el nombre del cliente y el número de teléfono.

**SELECT CONCAT(customer.last\_name, ', ', customer.first\_name) AS customer,**

**address.phone, film.title**

**FROM rental INNER JOIN customer ON rental.customer\_id =**

**customer.customer\_id**

**INNER JOIN address ON customer.address\_id = address.address\_id**

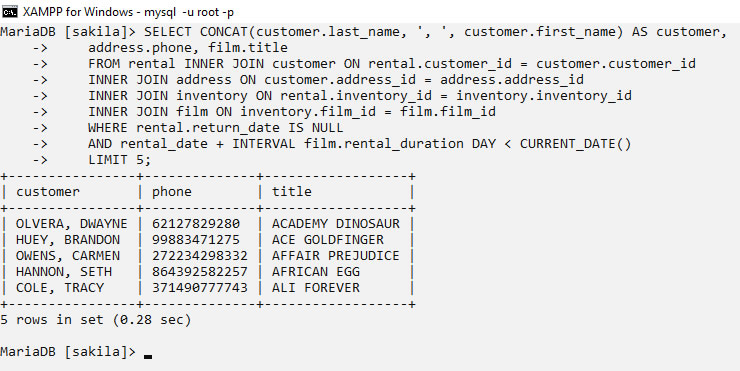
**INNER JOIN inventory ON rental.inventory\_id = inventory.inventory\_id**

**INNER JOIN film ON inventory.film\_id = film.film\_id**

**WHERE rental.return\_date IS NULL**

**AND rental\_date + INTERVAL film.rental\_duration DAY < CURRENT\_DATE()**

**LIMIT 5;**

****