

ACTIVIDAD 5

Rodrigo Martin Alvarez

[FUENTE DE INFORMACION](#)

DATOS NUERICOS:

Tipo de Dato	Función	Rango/Capacidad
TINYINT	Se utiliza para almacenar números enteros muy pequeños y que no necesitan mucho espacio.	-128 a 127 (con signo) o 0 a 255 (sin signo)
SMALLINT	Ideal para almacenar números enteros pequeños, adecuado para valores como edad o cantidad pequeña.	-32,768 a 32,767 (con signo) o 0 a 65,535 (sin signo)
MEDIUMINT	Usado para almacenar números enteros de tamaño intermedio, como conteos más grandes que SMALLINT.	-8,388,608 a 8,388,607 (con signo) o 0 a 16,777,215 (sin signo)
INT	ETipo de dato entero más comúnmente utilizado para números enteros grandes.	-2,147,483,648 a 2,147,483,647 (con signo) o 0 a 4,294,967,295 (sin signo)

BIGINT	Adecuado para almacenar números enteros muy grandes, como identificadores únicos en grandes bases de datos.	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 (con signo) o 0 a 18,446,744,073,709,551,615 (sin signo)
FLOAT	Utilizado para almacenar números con decimales donde la precisión no es crítica.	-3.402823466E+38 a 3.402823466E+38
DOUBLE	Utilizado para almacenar números con decimales con mayor precisión que FLOAT.	-1.7976931348623157E+308 a 1.7976931348623157E+308
DECIMAL	Utilizado para almacenar números decimales con una precisión exacta, ideal para datos financieros.	Dependiente de M y D

DATOS DE FECHA Y HORA:

Tipo de Dato	Función	Rango/Capacidad
DATE	Utilizado para almacenar fechas sin tiempo.	1000-01-01 a 9999-12-31

DATETIME	Utilizado para almacenar fechas y tiempos combinados.	1000-01-01 00:00:00 a 9999-12-31 23:59:59
TIMESTAMP	Utilizado para almacenar fecha y hora, generalmente para rastrear cambios en registros.	1970-01-01 00:00:01 UTC a 2038-01-19 03:14:07 UTC
TIME	Utilizado para almacenar solo el tiempo del día.	-838:59:59 a 838:59:59
YEAR	Utilizado para almacenar solo el año.	1901 a 2155

DATOS DE CADENA:

Tipo de Dato	Función	Rango/Capacidad
CHAR	Utilizado para almacenar cadenas de texto con longitud fija, ideal para códigos de longitud constante.	1 a 255 caracteres
VARCHAR	Utilizado para almacenar cadenas de texto con longitud variable, adecuado	1 a 65,535 caracteres

	para la mayoría de los textos.	
TINYTEXT	Utilizado para almacenar pequeñas cantidades de texto.	Hasta 255 caracteres
TEXT	Utilizado para almacenar textos moderadamente largos.	Hasta 65,535 caracteres
MEDIUMTEXT	Utilizado para almacenar textos largos.	Hasta 16,777,215 caracteres
LONGTEXT	Utilizado para almacenar textos extremadamente largos.	Hasta 4,294,967,295 caracteres
TINYBLOB	Utilizado para almacenar pequeñas cantidades de datos binarios.	Hasta 255 bytes
BLOB	Utilizado para almacenar datos binarios moderadamente grandes.	Hasta 65,535 bytes
MEDIUMBLOB	Utilizado para almacenar datos binarios grandes.	Hasta 16,777,215 bytes

LONGBLOB	Utilizado para almacenar datos binarios extremadamente grandes.	Hasta 4,294,967,295 bytes
ENUM	Utilizado para almacenar un solo valor de una lista predefinida, ideal para opciones limitadas.	Un valor de la lista predefinida
SET	Utilizado para almacenar cero o más valores de una lista predefinida, adecuado para etiquetas o permisos.	Cero o más valores de la lista predefinida

DATOS ESPECIALES:

Tipo de Dato	Función
GEOMETRY	Utilizado para almacenar cualquier tipo de datos geométricos.
POINT	Utilizado para almacenar un único punto en un espacio bidimensional.
LINESTRING	Utilizado para almacenar una secuencia de puntos que forman una línea.

POLYGON	Utilizado para almacenar una forma geométrica cerrada formada por una secuencia de puntos.
---------	--

2-INVESTIGACIÓN PARA LA CREACIÓN DE UNA BASE DE DATOS

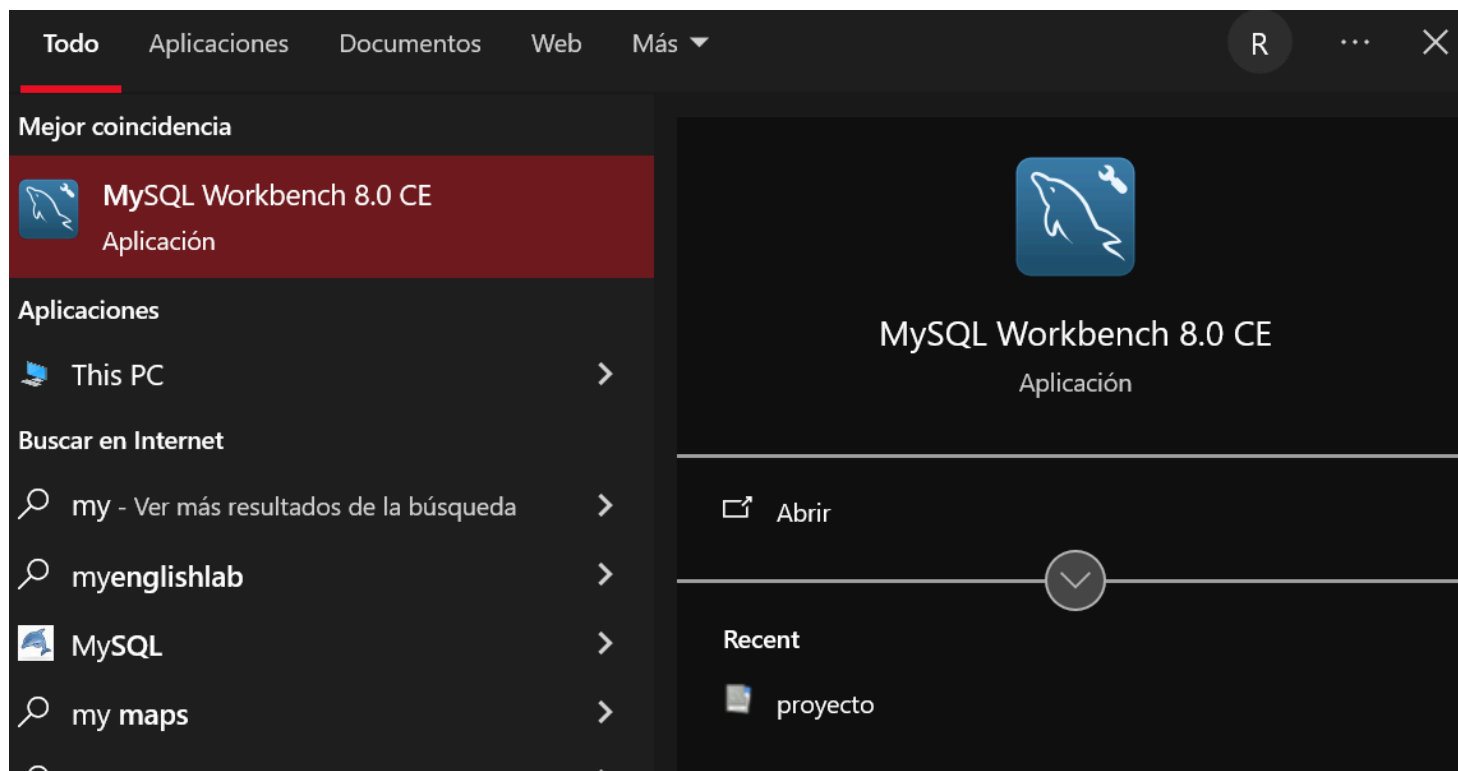
Para crear una base de datos en SQL, usamos `CREATE DATABASE nombre_base_de_datos;`, luego la seleccionamos con `USE nombre_base_de_datos;`.

Para crear una tabla, utilizamos `CREATE TABLE nombre_tabla (...);`, por ejemplo: `CREATE TABLE clientes (id INT PRIMARY KEY AUTO_INCREMENT, nombre VARCHAR(100) NOT NULL, email VARCHAR(100) UNIQUE NOT NULL, telefono VARCHAR(15), fecha_registro DATE DEFAULT CURRENT_DATE);`, donde definimos los campos con sus respectivos tipos de datos.

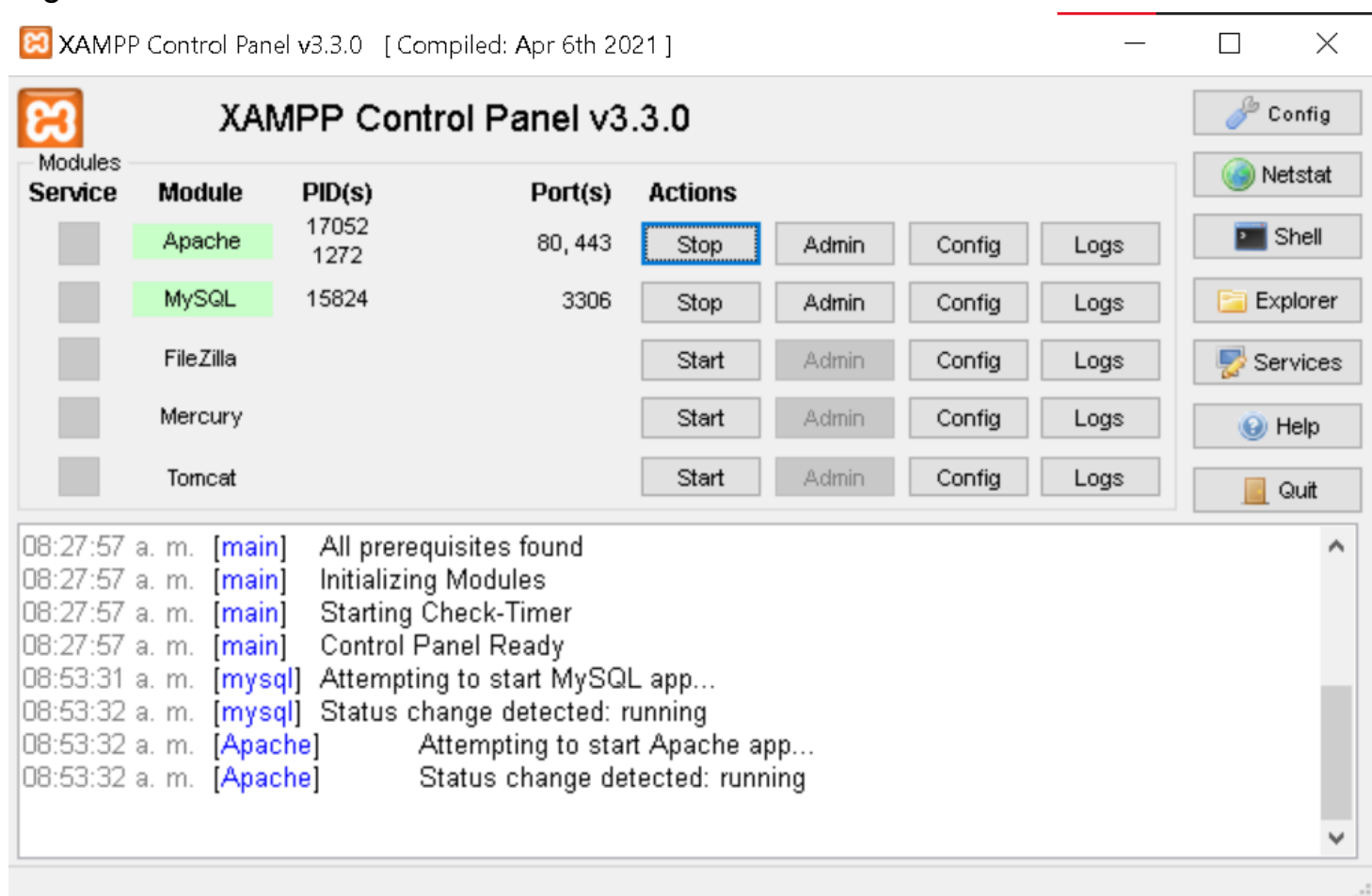
Para insertar registros usamos `INSERT INTO clientes (nombre, email, telefono) VALUES ('Juan Pérez', 'juan@example.com', '555-1234');`, o bien insertar múltiples valores con `INSERT INTO clientes (nombre, email, telefono) VALUES ('Ana López', 'ana@example.com', '555-5678'), ('Carlos Ruiz', 'carlos@example.com', '555-9876');`.

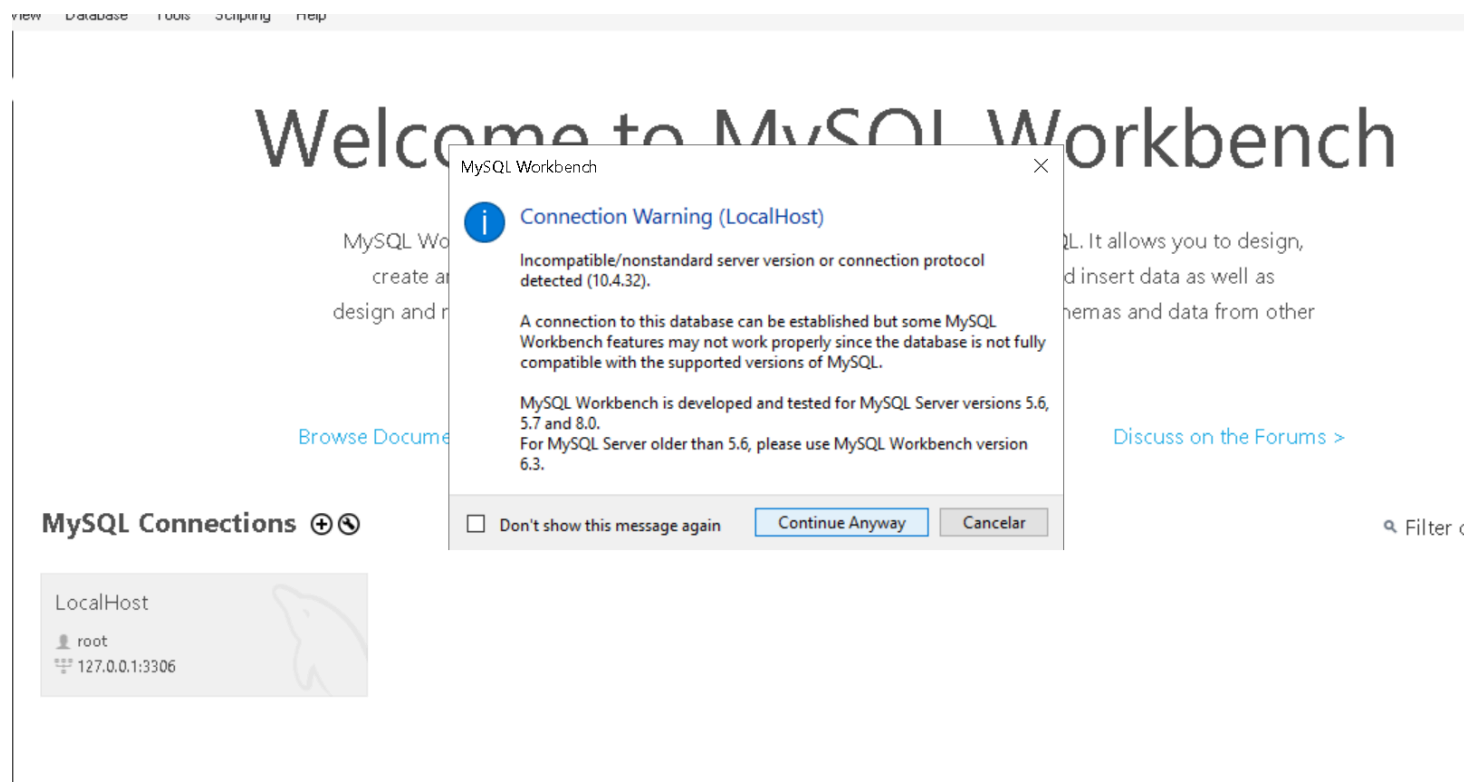
Para visualizar los datos almacenados en la tabla usamos `SELECT * FROM clientes;`. Esto cubre la estructura básica para crear una base de datos, definir tablas e insertar registros en SQL.

3-CREACION DE BASE DE DATOS



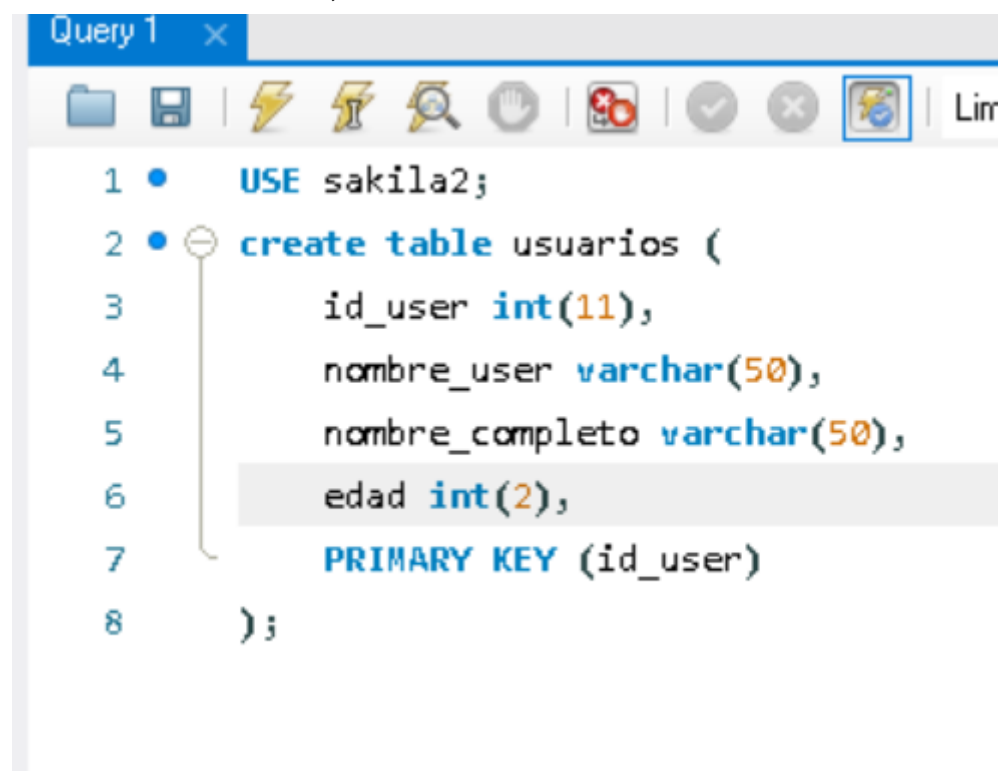
Nos colocamos en el explorador de windows y buscamos MYSQL Workbench, en seguida le damos en abrir.





Abrimos XAMPP y MYSQL, para conectarnos con el LocalHost

primero accedemos a MYSQL y despues creamos una base de daros con CREATE DATABASE sakila2;



En seguida creamos una tabla y le establecemos como primary key a algún atributo.

Database management tool interface showing the schema for the `usuarios` table.

Schema Definition:

```
CREATE TABLE usuarios (
  id_user int(11) PK,
  nombre_user varchar(50),
  nombre_completo varchar(50),
  edad int(2)
);
```

SQL Query:

```
SELECT * FROM sakila2.usuarios;
```

Result Grid:

	id_user	nombre_user	nombre_completo	edad
•	NULL	NULL	NULL	NULL

Verificamos que la tabla se creó de manera correcta.