



Bootcamp de Desarrollo Web

Clase 15

Introducción a MySQL

1



Sistema de gestión de bases de datos

Un sistema de gestión de bases de datos (SGBD o DBMS, por sus siglas en inglés) es un software que permite a los usuarios crear, gestionar y manipular bases de datos. Facilita el almacenamiento, la organización y la recuperación eficiente de datos.

Aquí tienes algunos ejemplos populares de sistemas de gestión de bases de datos:



ORACLE

SYBASE

MySQL



Microsoft SQL Server



Introducción a la programación

2



MySQL

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto muy popular. Es un software de servidor de bases de datos que utiliza el **lenguaje SQL** para gestionar y manipular los datos almacenados en la base de datos. MySQL es conocido por ser rápido, confiable y fácil de usar, y es ampliamente utilizado en aplicaciones web y otros entornos.



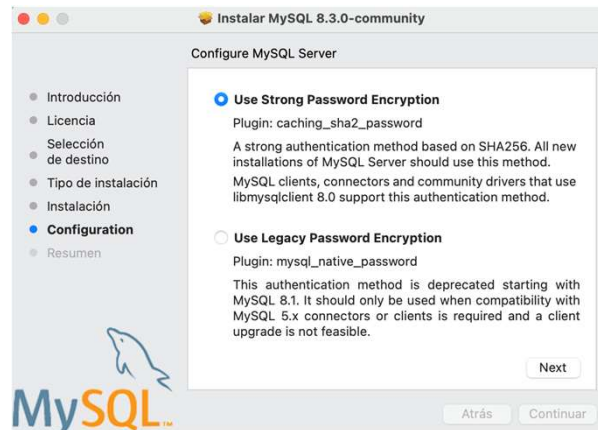
Diferencia entre MySQL y SQL

MySQL es un sistema específico de **gestión de bases de datos** que utiliza SQL como su lenguaje de consulta y manipulación de datos. **SQL, por otro lado, es el lenguaje** estándar utilizado para interactuar con bases de datos relacionales en general. MySQL implementa SQL para permitir a los usuarios trabajar con los datos almacenados en su base de datos MySQL.





Instalar MySQL

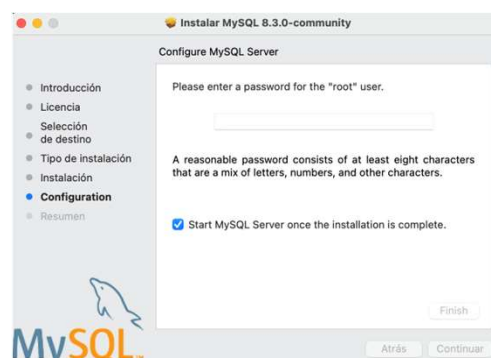


Introducción a la programación

5



Instalar MySQL



Nota: Si utilizas Mac instala también MySQL Workbench

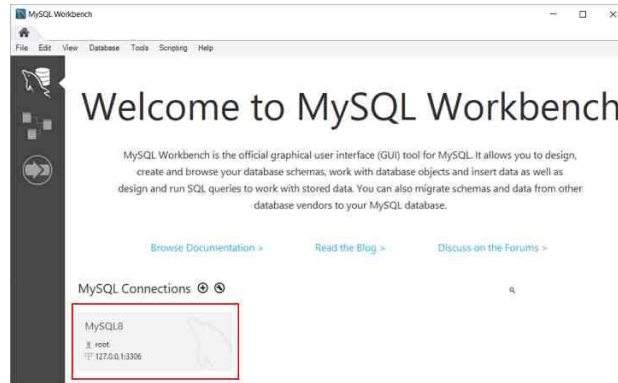
Introducción a la programación

6



Mostrar Bases de Datos

Conectarse al servidor local de mySQL



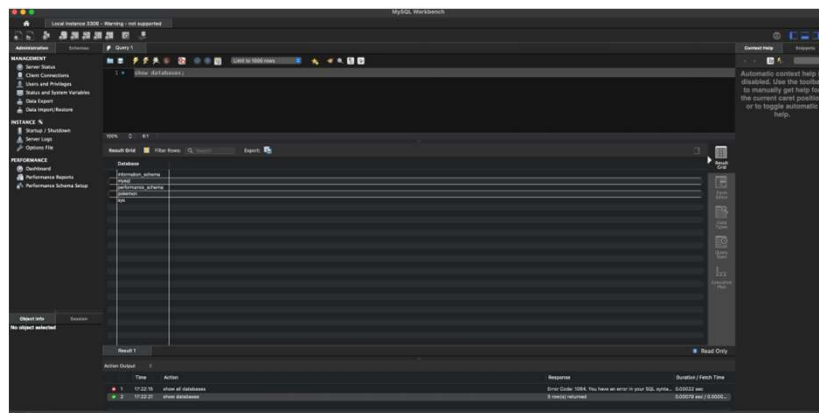
Introducción a la programación

7



Mostrar Bases de Datos

Escribir el comando : **show data bases;**



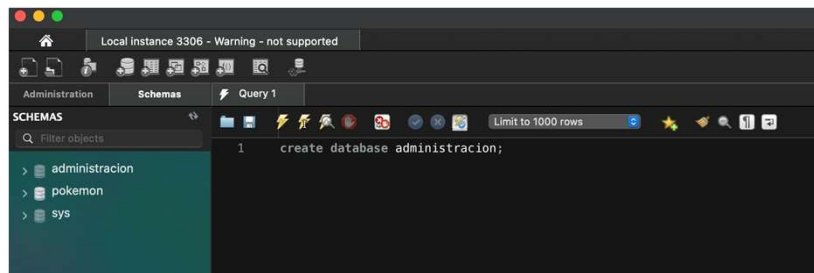
Introducción a la programación

8



Crear Bases de Datos

Se muestran 6 bases de datos creadas cuando instalamos el MySQL pero no la base de datos "administración". Debemos nosotros crear la base de datos "administración" mediante el comando **create database administracion;** y presionar el ícono de ejecución:



Introducción a la programación

9



Base de Datos

Una base de datos almacena sus datos en tablas.

Una tabla es una estructura de datos que organiza los datos en columnas y filas; cada columna es un campo (o atributo) y cada fila, un registro. La intersección de una columna con una fila, contiene un dato específico, un solo valor.

- Cada registro contiene un dato por cada columna de la tabla.
- Cada campo (columna) debe tener un nombre. El nombre del campo hace referencia a la información que almacenará.
- Cada campo (columna) también debe definir el tipo de dato que almacenará.

Introducción a la programación

10



Tablas

Gráficamente acá tenemos la tabla usuarios, que contiene dos campos llamados: nombre y clave. Luego tenemos tres registros almacenados en esta tabla, el primero almacena en el campo nombre el valor "MarioPerez" y en el campo clave "Marito", y así sucesivamente con los otros dos registros.

- Las tablas forman parte de una base de datos.
- Nosotros trabajaremos con la base de datos llamada **administración**
- Para ver las tablas existentes en una base de datos tipeamos:
show tables;



Crear Tabla

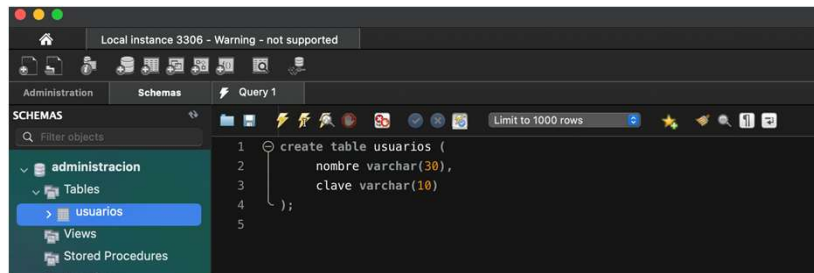
Al crear una tabla debemos resolver qué campos (columnas) tendrá y que tipo de datos almacenarán cada uno de ellos, es decir, su estructura.

- La tabla debe ser definida con un nombre que la identifique y con el cual accederemos a ella.
- Creamos una tabla llamada "usuarios", tipeamos:

```
create table usuarios (  
    nombre varchar(30),  
    clave varchar(10)  
);
```



Crear Tabla



Introducción a la programación

13

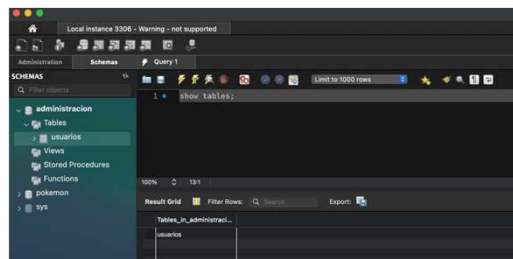


Tipos de Llaves

Para ver las tablas existentes en una base de datos tipeamos nuevamente:

show tables;

- Ahora aparece "usuarios" entre otras que ya pueden estar creadas.



Introducción a la programación

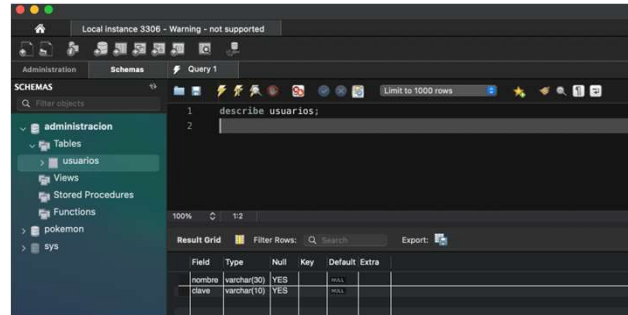
14



Describir Tabla

Para ver la estructura de una tabla usamos el comando **describe** junto al nombre de la tabla:

describe usuarios;



Introducción a la programación

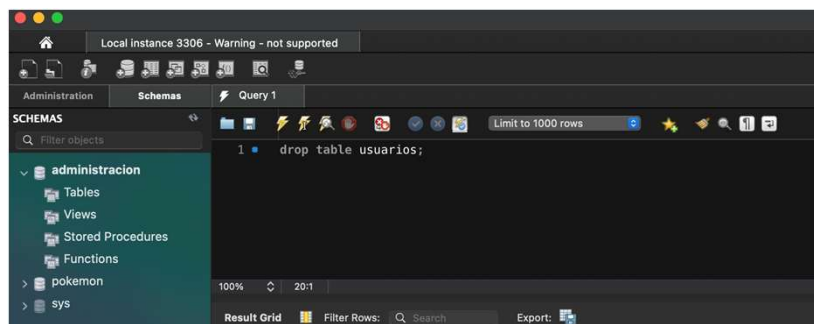
15



Eliminar tabla

Para eliminar una tabla usamos:

drop table usuarios;



Introducción a la programación

16

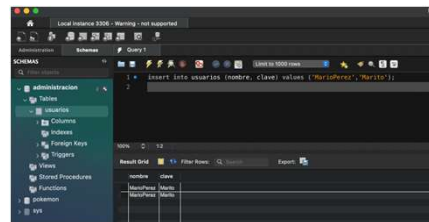


Ingresar datos

Al ingresar los datos de cada registro debe tenerse en cuenta la cantidad y el orden de los campos.

Ahora vamos a agregar un registro a la tabla:

insert into usuarios (nombre, clave) values ('MarioPerez','Marito');



Introducción a la programación

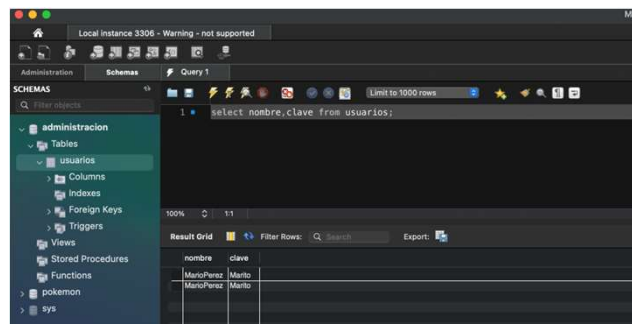
17



Ver contenido tabla

Para ver los registros de una tabla usamos "select":

select nombre,clave from usuarios;



Introducción a la programación

18



Tipos básicos de datos

En SQL, los tipos de datos básicos se utilizan para definir el tipo de valores que pueden almacenarse en una columna de una tabla. Aquí hay algunos de los tipos de datos básicos más comunes en SQL:

INTEGER: Este tipo de dato se utiliza para almacenar números enteros. Dependiendo de la implementación específica de SQL, puede haber diferentes variantes de este tipo de dato, como TINYINT, SMALLINT, INT, BIGINT, etc., que varían en el rango de valores que pueden contener.



Tipos básicos de datos

FLOAT o REAL: Estos tipos de datos se utilizan para almacenar números decimales (números de punto flotante). La precisión y el rango de estos números pueden variar según la implementación específica de SQL.

CHAR o VARCHAR: Estos tipos de datos se utilizan para almacenar cadenas de caracteres de longitud fija (CHAR) o variable (VARCHAR). CHAR almacena una cadena de caracteres de longitud fija, mientras que VARCHAR almacena una cadena de caracteres de longitud variable.



Tipos básicos de datos

DATE o TIMESTAMP: Estos tipos de datos se utilizan para almacenar valores de fecha y hora. DATE se utiliza para almacenar solo la fecha, mientras que TIMESTAMP se utiliza para almacenar la fecha y la hora.

BOOLEAN: Este tipo de dato se utiliza para almacenar valores de verdad (verdadero o falso).

BLOB o CLOB: Estos tipos de datos se utilizan para almacenar grandes cantidades de datos binarios (BLOB) o grandes cantidades de datos de caracteres (CLOB), como imágenes, archivos de audio, texto largo, etc.



Seleccionar algunos datos

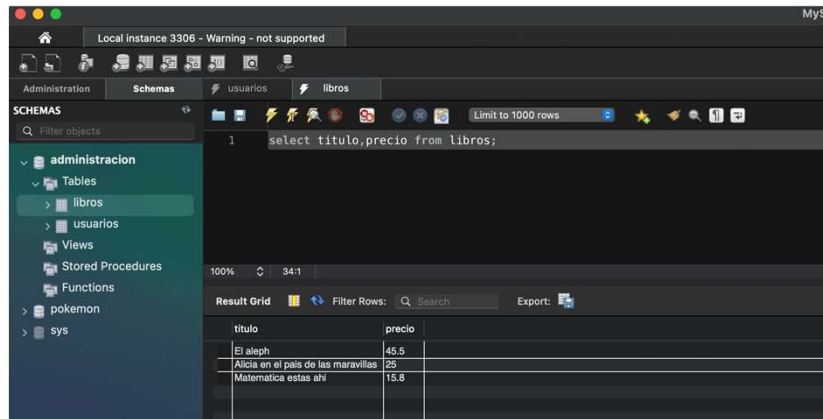
El comando "select" recupera los registros de una tabla. Con el asterisco (*) indicamos que seleccione todos los campos de la tabla que nombramos.

Podemos especificar el nombre de los campos que queremos ver separándolos por comas:

```
select titulo,autor,editorial from libros;
```



Seleccionar algunos dato



Introducción a la programación

23



Seleccionar datos específicos

Existe una cláusula, "**where**" que es opcional, con ella podemos especificar condiciones para la consulta "select". Es decir, podemos recuperar algunos registros, sólo los que cumplan con ciertas condiciones indicadas con la cláusula "where". Por ejemplo, queremos ver el usuario cuyo nombre es "MarioPerez", para ello utilizamos "where" y luego de ella, la condición:

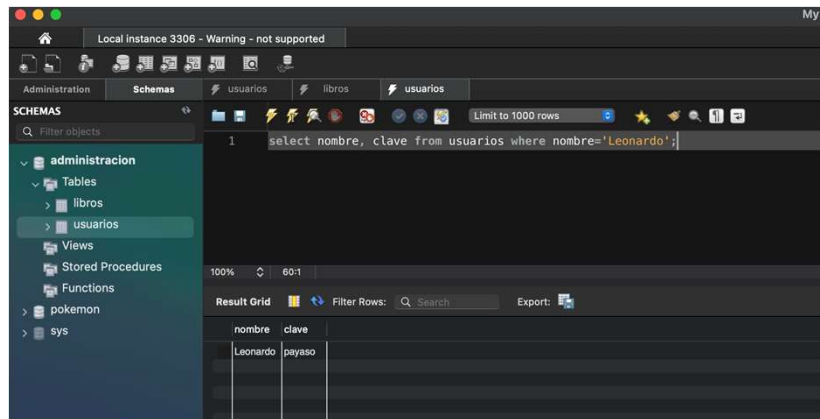
```
select nombre, clave from usuarios where nombre='MarioPerez';
```

Introducción a la programación

24



Seleccionar datos específicos



Introducción a la programación

25



Operadores Relacionales

Los operadores relacionales vinculan un campo con un valor para que MySQL compare cada registro (el campo especificado) con el valor dado.

Los operadores relacionales son los siguientes:

- = igual**
- <> distinto**
- > mayor**
- < menor**
- >= mayor o igual**
- <= menor o igual**

Introducción a la programación

26



Borrado de registros de una tabla

La ejecución del comando indicado en la línea anterior borra TODOS los registros de la tabla.

Si queremos eliminar uno o varios registros debemos indicar cuál o cuáles, para ello utilizamos el comando "delete" junto con la cláusula "where" con la cual establecemos la condición que deben cumplir los registros a borrar. Por ejemplo, queremos eliminar aquel registro cuyo nombre de usuario es 'Leonardo':

```
delete from usuarios where nombre='Leonardo';
```



Borrado de registros de una tabla

Si solicitamos el borrado de un registro que no existe, es decir, ningún registro cumple con la condición especificada, no se borrarán registros, pues no encontró registros con ese dato.

El comando **delete** hay que tener mucho cuidado en su uso, una vez eliminado un registro no hay forma de recuperarlo. Si por ejemplo ejecutamos el comando:

```
delete from usuarios;
```

Si la tabla tiene 1000000 de filas, todas ellas serán eliminadas.



Modificación de registros de una tabla (update)

Utilizamos "update" junto al nombre de la tabla y "set" junto con el campo a modificar y su nuevo valor.

- El cambio afectará a todos los registros.
- Podemos modificar algunos registros, para ello debemos establecer condiciones de selección con "where".
- Por ejemplo, queremos cambiar el valor correspondiente a la clave de nuestro usuario llamado 'MarioPerez', queremos como nueva clave 'Boca', necesitamos una condición "where" que afecte solamente a este registro:

```
update usuarios set clave='Boca'  
where nombre='MarioPerez';
```



Modificación de registros de una tabla (update)

Una clave primaria es un campo (o varios) que identifica 1 solo registro (fila) en una tabla.

Para un valor del campo clave existe solamente 1 registro. Los valores no se repiten ni pueden ser nulos.

Veamos un ejemplo, si tenemos una tabla con datos de personas, el número de documento puede establecerse como clave primaria, es un valor que no se repite; puede haber personas con igual apellido y nombre, incluso el mismo domicilio (padre e hijo por ejemplo), pero su documento será siempre distinto.



Modificación de registros de una tabla (update)

Si tenemos la tabla "usuarios", el nombre de cada usuario puede establecerse como clave primaria, es un valor que no se repite; puede haber usuarios con igual clave, pero su nombre de usuario será siempre distinto.

Establecemos que un campo sea clave primaria al momento de creación de la tabla:

```
create table usuarios (  
  nombre varchar(20),  
  clave varchar(10),  
  primary key(nombre)  
);
```

Introducción a la programación

31



Bootcamp de Desarrollo Web

Clase 15

Introducción a MySQL

32