

Lenguaje SQL

S.Q.L. significa lenguaje estructurado de consulta (Structured Query Language). Es un lenguaje estándar de cuarta generación que se utiliza para definir, gestionar y manipular la información contenida en una Base de Datos Relacional.

Se trata de un lenguaje definido por el estándar ISO/ANSI SQL que utilizan los principales fabricantes de Sistemas de Gestión de Bases de Datos Relacionales.

Es un lenguaje sencillo y potente que se emplea para la gestión de la base de datos a distintos niveles de utilización: usuarios, programadores y administradores de la base de datos.

.Establecer entorno de trabajo

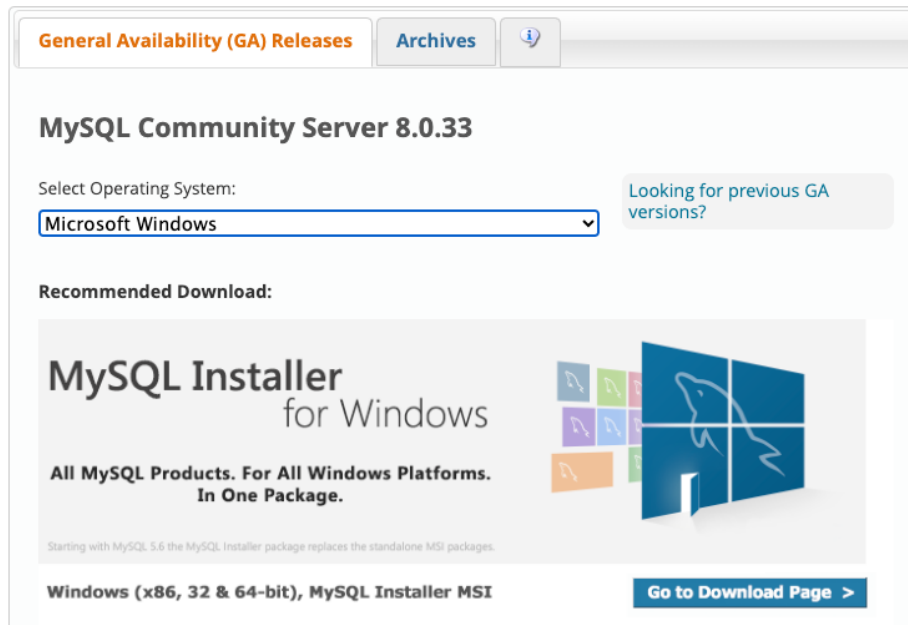
Es importante saber cómo obtener, instalar, configurar y probar las herramientas necesarias para realizar las operaciones de diseño; existen muchas tecnologías para diseñar e implementar bases de datos, se usará MySQL como sistema gestor de bases de datos y MySQL Workbench como herramienta de diseño.

Sistema gestor de bases de datos MySQL

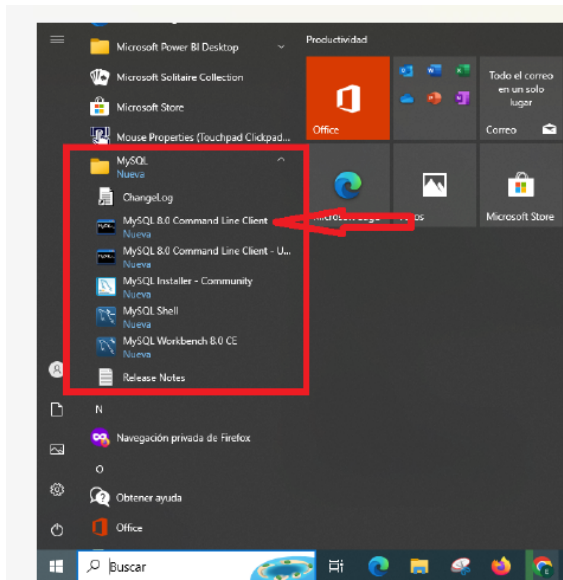
Tiene licencia dual significa que se puede usar la licencia de uso público sin incurrir en gastos; sin embargo, se puede adquirir una licencia de pago que incluye paquetes que robustecen el producto y servicios de soporte técnico especializado.

DESCARGA:

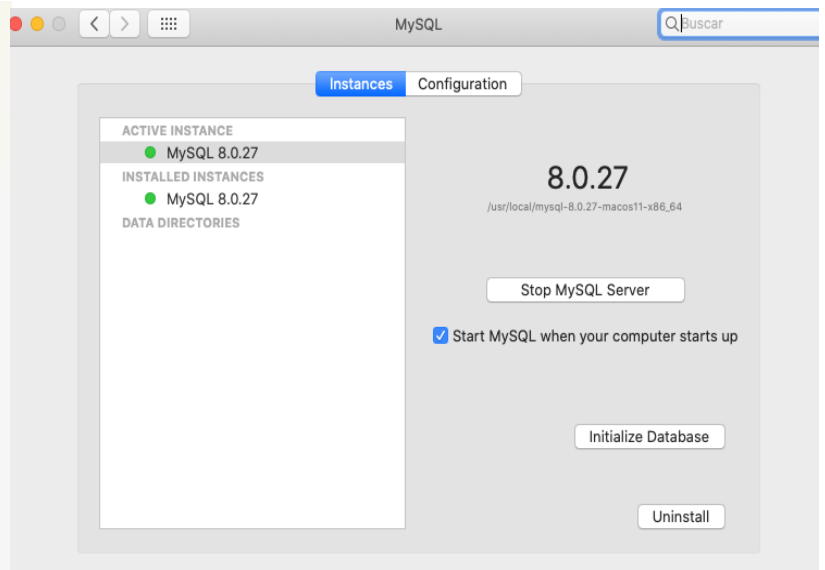
MySQL Server. Seleccionamos la versión de nuestro Sistema Operativo
<https://dev.mysql.com/downloads/mysql/>



WINDOWS



macOS



Sistema de diseño y gestión de bases de datos



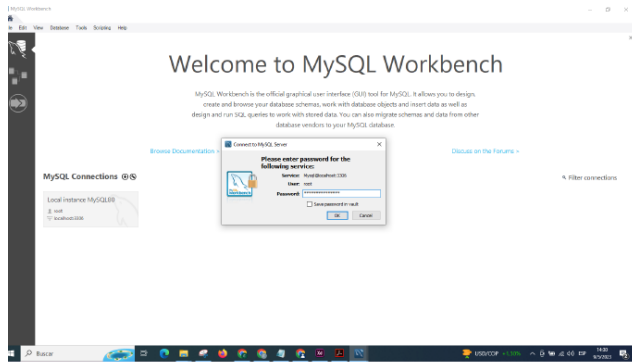
Es una herramienta visual unificada para arquitectos, desarrolladores y administradores de bases de datos. MySQL Workbench proporciona modelado de datos, desarrollo de SQL y herramientas de administración integrales para la configuración del servidor, administración de usuarios, respaldo y muchas otras características y, lo más importante es que es multiplataforma, lo que indica que puede descargarse para sistemas operativos Windows, Linux y Mac.

DESCARGA:

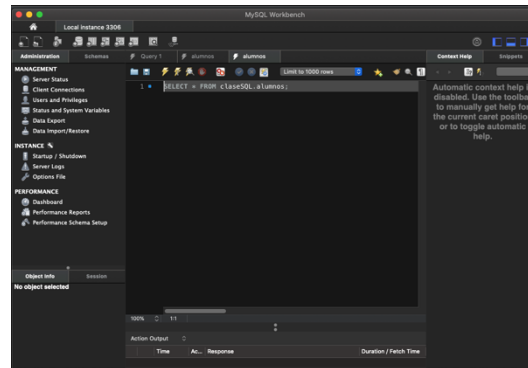
MySQL Workbench

<https://dev.mysql.com/downloads/workbench/>

WINDOWS



macOS



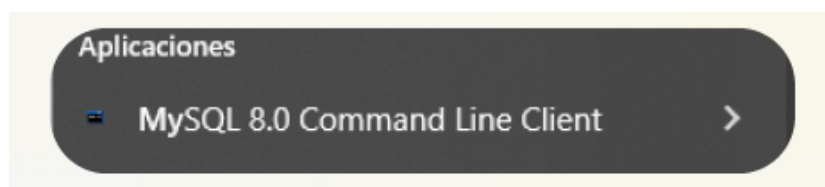
Interface de Línea de Comandos de MySQL

Al iniciar a trabajar con el SGBD, existen varias maneras de enlazar una comunicación con el servidor de MySQL. Generalmente en el software, se usará un API para hacer las consultas y otras operaciones con el servidor. Por ejemplo, en PHP, esta API está integrada con el lenguaje. En este punto se usará MySQL de forma directa, por medio de un cliente ejecutándose desde una consola (una ventana de comandos en Windows, y en MacOS). En otras clases se explicarán los distintos APIs.

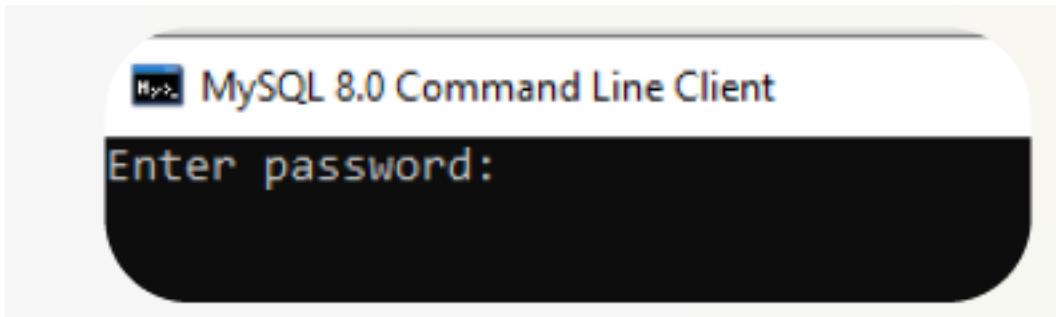
Sistema gestor de bases de datos MySQL

Se debe seguir las siguientes instrucciones para conectarse a la base de datos a través del cliente de línea de comandos:

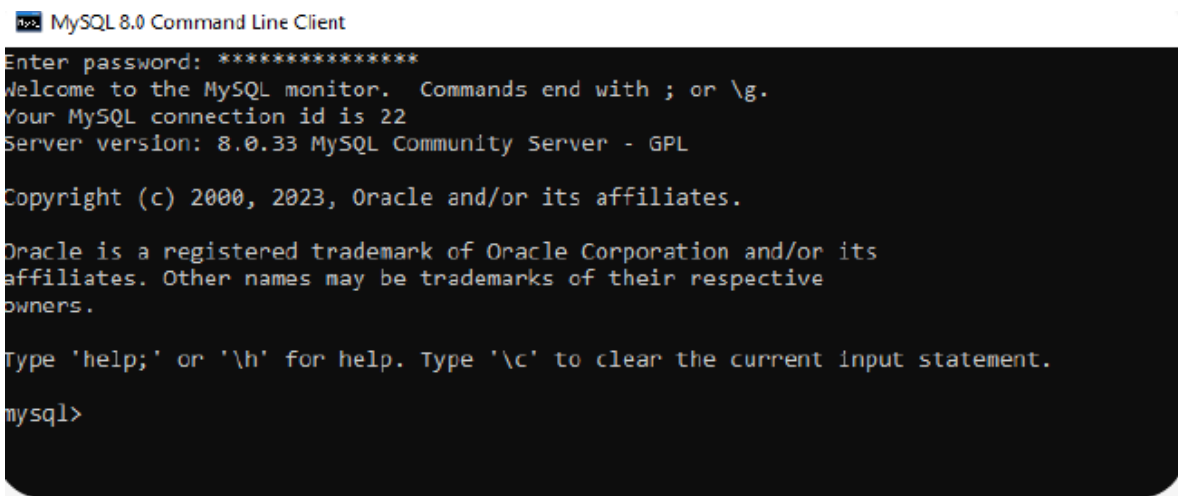
1. Debe abrir inicio y buscar MySQL o teclear "MySQL" y se hace Click en MySQL Command Line Client para-Windows.



2. Ingresa la contraseña de root usada en el proceso de instalación de MySQL.

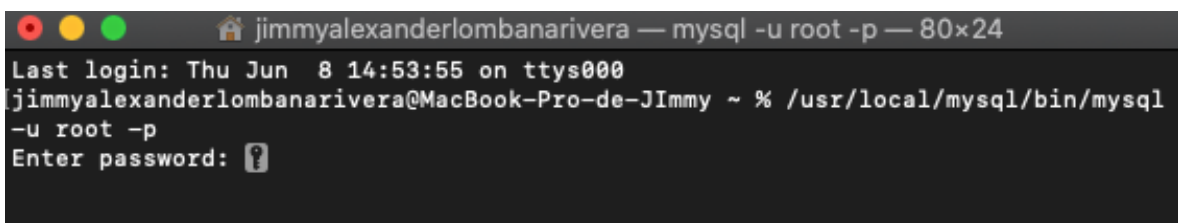


Una vez autenticado en la interface comando de línea vemos este resultado



2. Para MacOS : Ingresamos la terminal y escribimos este comando:

```
/usr/local/mysql/bin/mysql -u root -p
```



Ingresa la contraseña de root usada en el proceso de instalación de MySQL.

Una vez autenticado en la interface comando de línea vemos este resultado

```
jimmyalexanderlombanarivera — mysql -u root -p — 80x24
Last login: Thu Jun  8 14:53:55 on ttys000
[jimmyalexanderlombanarivera@MacBook-Pro-de-Jimmy ~ % /usr/local/mysql/bin/mysql ]
-u root -p
[Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

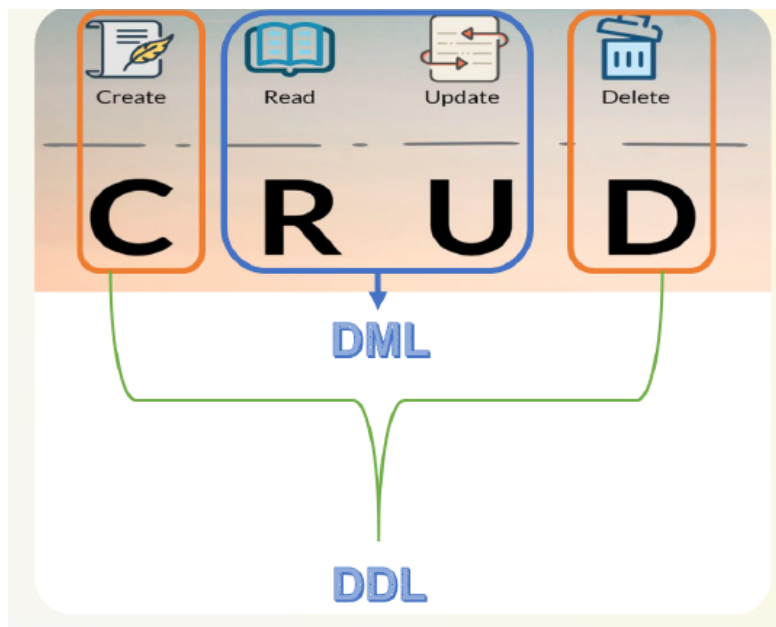
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> █
```

*** CRUD ***

CRUD es el acrónimo de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar). Este concepto se utiliza para describir las cuatro operaciones básicas que pueden realizarse en la mayoría de las bases de datos y sistemas de gestión de información.



BASES DE DATOS
@ING_ JIMMY LOMBANA RIVERA
COMANDOS UTILIZADOS EN SQL

.1. CREATE – CREAM

La declaración CREATE DATABASE se utiliza para crear una nueva base de datos SQL.

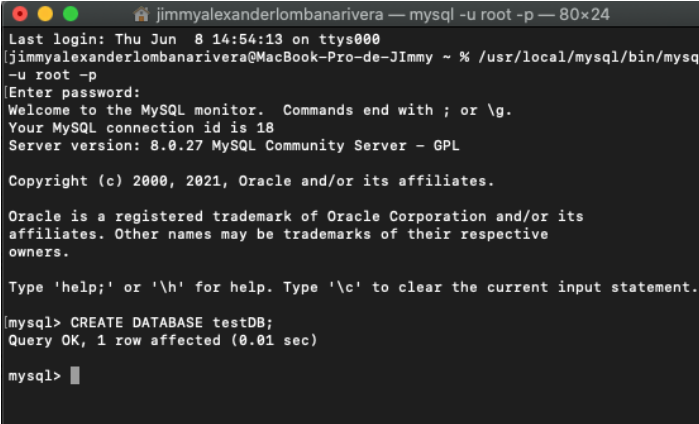
Sintaxis:

CREATE DATABASE databasename;

CREATE DATABASE testDB;

PASO A PASO EJERCICIO***

1. Iniciamos y nos autenticamos en Command line client o terminal para macOS



```
jimmyalexanderlombanarivera — mysql -u root -p — 80x24
Last login: Thu Jun  8 14:54:13 on ttys000
jimmyalexanderlombanarivera@MacBook-Pro-de-Jimmy ~ % /usr/local/mysql/bin/mysql
-u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, 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> CREATE DATABASE testDB;
Query OK, 1 row affected (0.01 sec)

mysql>
```

2. Conocer bases de datos que se encuentran en el sistema (Sólo se verán esas, dado que no se a creado ninguna por parte de nosotros).

Sintaxis:

SHOW DATABASES;

BASES DE DATOS
@ING_JIMMY LOMBANA RIVERA

```
jimmyalexanderlombanarivera — mysql -u root -p — 80x24
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> CREATE DATABASE testDB;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| claseSQL |
| information_schema |
| mysql |
| new_schema |
| performance_schema |
| sys |
| testDB |
+-----+
7 rows in set (0.00 sec)

mysql>
```

Cómo buena práctica tenemos, poner en mayúsculas todas las palabras reservadas.

<https://www.ibm.com/docs/es/psfa/7.1.0?topic=keywords-sql-common-reserved-words>

3. Vamos a crear la siguiente base de datos:
CREATE DATABASE claseSQL;

Una vez creada la base de datos vamos a empezar a usar la base de datos que acabamos de crear, usaremos la palabra reservada:

USE Nombre de la base de datos;

```
jimmyalexanderlombanarivera — mysql -u root -p — 80x24
Last login: Thu Jun 8 15:10:37 on ttys000
jimmyalexanderlombanarivera@MacBook-Pro-de-Jimmy ~ % /usr/local/mysql/bin/mysql
-u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, 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> USE clasesql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

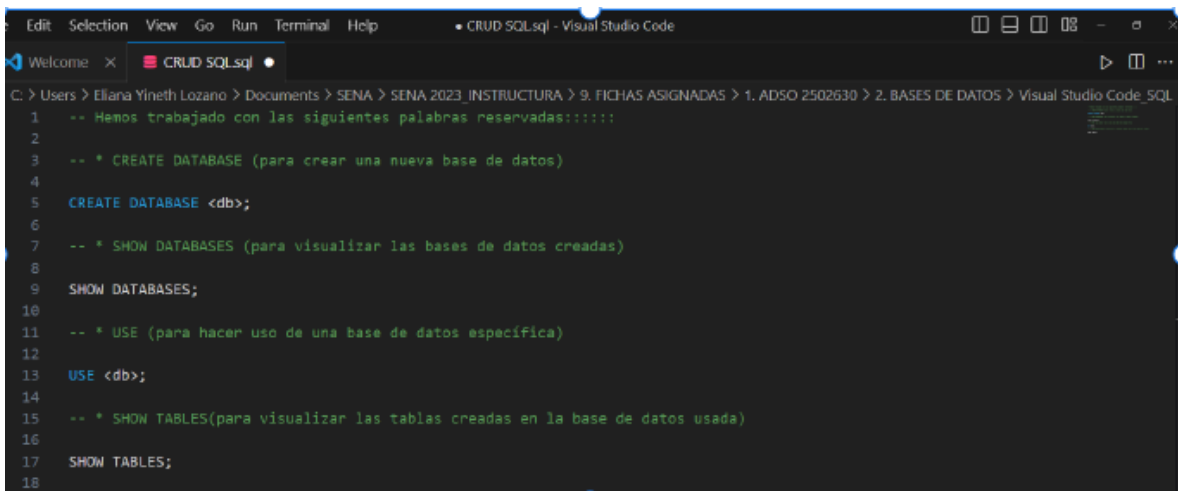

Si ejecutamos un SHOW TABLES; que funciona de la misma manera que el SHOW DATABASES; sólo que para visualizar las tablas que contiene la base de datos creada y que estamos usando, encontraremos que no tenemos tablas creadas:

```
mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql>
```

Ahora crearemos nuestra primera tabla, sin embargo, para trabajar mejor y entender el código haremos uso de Visual Studio Code e iremos guardando y comentando sobre la funcionalidad de las palabras reservadas y el código que vamos a ir trabajando.

*** Por favor crear un repositorio en su cuenta de GIT llamado: "Ejercicios SQL" y agregarme como colaboradora. Guardar en el los archivos que vamos creando y trabajando para las retroalimentaciones correspondientes***



```
1  -- Hemos trabajado con las siguientes palabras reservadas:::::
2
3  -- * CREATE DATABASE (para crear una nueva base de datos)
4
5  CREATE DATABASE <db>;
6
7  -- * SHOW DATABASES (para visualizar las bases de datos creadas)
8
9  SHOW DATABASES;
10
11 -- * USE (para hacer uso de una base de datos específica)
12
13 USE <db>;
14
15 -- * SHOW TABLES(para visualizar las tablas creadas en la base de datos usada)
16
17 SHOW TABLES;
18
```

TIPOS DE DATOS UTILIZADOS PARA INGRESAR DATOS EN SQL

INT (Para tipos de datos enteros)

VARCHAR (Nos permite limitar la cantidad de caracteres alfanuméricos de una cadena al ingresar)

ENUM (permite limitar las opciones de ingreso ejemplo: Si o No, Activo o Inactivo, Aprobado o Rechazado)

TEXT (Nos permite almacenar un mayor número de caracteres)

TIMESTAMP (Nos permite almacenar la fecha exacta, Año-Mes-Día-Hora-Minutos-Segundos)

Una vez entendido los tipos de datos que se utilizan al momento de ingresar datos, procedemos a crear nuestra primera tabla en nuestra base de datos.

SINTASIS

CREATE TABLE aprendices (

```
CREATE TABLE aprendices(  
  id INT,  
  nombre_usuario VARCHAR (50),  
  correo VARCHAR(50),  
  edad INT,  
  estado ENUM('Activo', 'Inactivo'),  
  intereses TEXT,  
  creado TIMESTAMP  
);
```

```
jimmyalexanderlombanarivera — mysql -u root -p — 80x24  
corresponds to your MySQL server version for the right syntax to use near 'alumn  
os' at line 1  
[mysql> drop table alumnos;  
Query OK, 0 rows affected (0.05 sec)  
  
[mysql> SHOW TABLES;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that  
corresponds to your MySQL server version for the right syntax to use near 'TABLE  
ES' at line 1  
[mysql> SHOW TABLES;  
Empty set (0.00 sec)  
  
[mysql> CREATE TABLE aprendices(  
  -> id INT,  
  -> nombre_usuario VARCHAR(50),  
  -> Correo VARCHAR(50),  
  -> edad INT,  
  -> estado ENUM ("Activo", "Inactivo"),  
  -> intereses TEXT,  
  -> creado TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
  -> );  
Query OK, 0 rows affected (0.04 sec)  
  
mysql>
```

Revisaremos la tabla si se creó correctamente
Síntesis;

SHOW TABLES;

```
[mysql> SHOW TABLES;  
+-----+  
| Tables_in_clasesql |  
+-----+  
| aprendices          |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>
```

Ejecutemos DESC aprendices; para analizar cómo está creada la tabla, evidenciando cantidad de columnas, tipos de datos entre otros,

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
nombre_usuario	varchar(50)	YES		NULL	
Correo	varchar(50)	YES		NULL	
edad	int	YES		NULL	
estado	enum('Activo','Inactivo')	YES		NULL	
intereses	text	YES		NULL	
creado	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

rows in set (0.00 sec)

EJERCICIO;

La declaración DROP TABLE se usa para eliminar una tabla existente en una base de datos.

*** Por favor creen una tabla llamada: estudiantes, con la misma estructura de la creada para aprendices ***

Ahora, hagamos uso de DROP TABLE estudiantes;

```
mysql> DROP TABLE estudiantes;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Haremos uso de los CONSTRAINTS, REGLAS O RESTRICCIONES, en las columnas de la tabla aprendices que hemos conservado. es decir que nos permitirá darle integridad o coherencia a los datos que vamos a ingresar.

Eliminaremos la tabla aprendices y la realizaremos de nuevo para ingresar datos coherentes y integridad.

```
mysql> CREATE TABLE aprendices(
  -> id INT AUTO_INCREMENT PRIMARY KEY,
  -> Nombre_Usuario VARCHAR(50) UNIQUE NOT NULL,
  -> Edad INT UNSIGNED NOT NULL,
  -> Estado ENUM("Activo","Inactivo") DEFAULT "Inactivo",
  -> Intereses TEXT,
  -> Creado TIMESTAMP DEFAULT CURRENT_TIMESTAMP
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql> █
```

Analizaremos las nuevas sintaxis

LA PRIMERA LINEA : id IN AUTO_INCREMENT PRIMARY KEY: Se va incrementar el Id y va hacer la key primaria

LA SEGUNDA LINEA : nombre usuario VARCHAR(50) UNIQUE NO NULL, : Se utiliza para no inserten valores duplicados en una columna:

LA TERCERA LINEA: Edad INT UNSIGNED NOT NULL: Para que no ingresen números negativo.

LA CUARTA LINEA: Estado ENUM ("Activo", "Inactivo") DEFAULT "Inactivo", Para que cualquiera de las dos quede por defecto:

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
Nombre_Usuario	varchar(50)	NO	UNI	NULL	
Edad	int unsigned	NO		NULL	
Estado	enum('Activo','Inactivo')	YES		Inactivo	
Intereses	text	YES		NULL	
Creado	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

6 rows in set (0.01 sec)

Una vez creado la tabla procederemos a ingresar los datos:

COMANDOS UTILIZADOS:

INSERT INTO "nombre tabla" ("columna1", "columna2", ...) VALUES ("valor1", "valor2",...)

Explicación: Nombre tabla: Es el nombre de la tabla en la que vamos a insertar registros.

columna1, columna2,...: Son las columnas de la tabla en la que vamos a insertar datos.

```
Database changed
mysql> INSERT INTO aprendices (Nombre_Usuario,Edad, Estado, Intereses) VALUE("Jimmy Lombana", 31, "Activo" , "Aprendiendo con ustdes");
Query OK, 1 row affected (0.01 sec)

mysql> █
```

Ahora realicemos una consulta a nuestra base de datos para recuperar los datos ingresados en la tabla aprendices
SELECT * FROM Nombre de la tabla

```
mysql> SELECT * FROM aprendices;
+----+-----+-----+-----+-----+-----+
| id | Nombre_Usuario | Edad | Estado | Intereses | Creado |
+----+-----+-----+-----+-----+-----+
| 1 | Jimmy Lombana | 31 | Activo | Aprendiendo con ustdes | 2023-06-08 17:42:42 |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> █
```

Insertemos un par 12 Registros más.