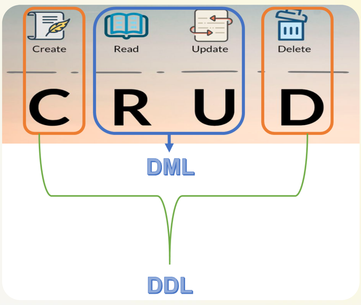
Bases de datos

ELIANA YINETH
LOZANO TRIANA
eylozano@sena.edu.co

*** 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.



1. CREATE - CREAR

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

Sintaxis:

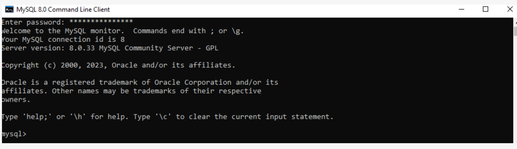
CREATE DATABASE databasename;

Ejemplo:

CREATE DATABASE testDB;

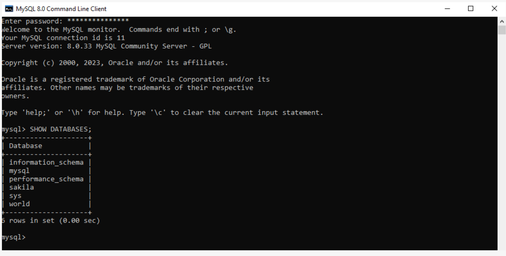
*** PASO A PASO EJERCICIO***

1. Iniciamos y nos autenticamos en Command line client.



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).

SHOW DATABASES;



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;

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
6 rows in set (0.00 sec)

mysql> CREATE DATABASE claseSQL;
Query OK, 1 row affected (0.00 sec)

mysql>
```

No obtenemos ningún error, por lo que, se asume se creó la base correctamente, sin embargo, revisemos las bases de datos.

Como lo hacemos???????

Si pensaste en **SHOW DATABASES;** estás en lo correcto !!

```
mysql> CREATE DATABASE claseSQL;
Query OK, 1 row affected (0.00 sec)

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

mysql>
```

usar base de datos creada

Bases de datos

ELIANA YINETH
LOZANO TRIANA
eylozano@sena.edu.co

Para empezar a usar la base de datos que acabamos de crear, usaremos la palabra reservada:

USE databasename;

```
MySQL 8.0 Command Line Client
| Database           |
+--------------------+
| clasesql           |
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
7 rows in set (0.00 sec)

mysql> USE clasesql;
Database changed
mysql>
```

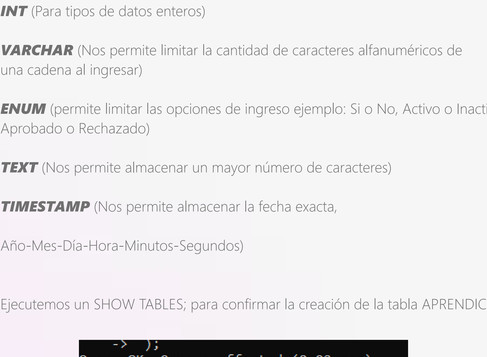
Si ejecutamos un **SHOW TABLES;** que funciona de la misma manera que el **SHOW DATABASES;** sólo que para visualizar las tablas que contiene la base de datos creada y que estamos usando, encontraremos que no tenemos tablas creadas:

```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql>
mysql>
mysql> SHOW TABLES;
Empty set (0.01 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***



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***

Crearemos la tabla **-----> APRENDICES <------**

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

```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql>  CREATE TABLE aprendices(
    ->     id INT,
    ->     nombre_usuario VARCHAR (50),
    ->     correo VARCHAR(50),
    ->     edad INT,
    ->     estado ENUM('Activo', 'Inactivo'),
    ->     intereses TEXT,
    ->     creado TIMESTAMP
    ->  );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

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)

Ejecutemos un SHOW TABLES; para confirmar la creación de la tabla APRENDICES.

```
    ->  );
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW TABLES;
+--------------------+
| Tables_in_clasesql |
+--------------------+
| aprendices         |
| estudiantes        |
+--------------------+
2 rows in set (0.00 sec)

mysql>
```

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

```
MySQL 8.0 Command Line Client
mysql> DESC aprendices;
+----------------+--------------------------+------+-----+---------+-------+
| 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  |     | NULL    |       |
+----------------+--------------------------+------+-----+---------+-------+
7 rows in set (0.01 sec)

mysql>
```

Drop Table

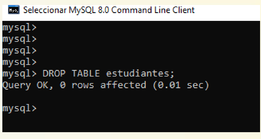
Bases de datos

ELIANA YINETH
LOZANO TRIANA
eylozano@sena.edu.co

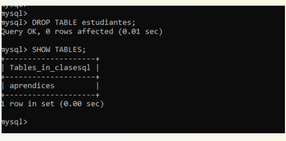
La declaración **DROP TABLE** se usa para eliminar una tabla existente en una base de datos.

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

Ahora, hagamos uso de **DROP TABLE estudiantes;**



Posterior, hagamos un **SHOW TABLES**, para confirmar la eliminación de la tabla estudiantes.



Haremos uso de los **CONSTRAINTS, REGLAS O RESTRICCIONES,** en las columnas de la tabla aprendices que hemos conservado.

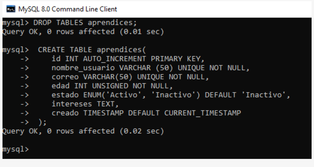
es decir que nos permitirá darle integridad o coherencia a los datos que vamos a ingresar.

```
48
49  CREATE TABLE aprendices(
50     id INT AUTO_INCREMENT PRIMARY KEY,
51     nombre_usuario VARCHAR (50) UNIQUE NOT NULL,
52     correo VARCHAR(50) UNIQUE NOT NULL,
53     edad INT UNSIGNED NOT NULL,
54     estado ENUM('Activo', 'Inactivo') DEFAULT 'Inactivo',
55     intereses TEXT,
56     creado TIMESTAMP DEFAULT CURRENT_TIMESTAMP
57  );
```

¿Que pasa si simplemente copiamos, pegamos y ejecutamos este código en consola ?? ¿que se nos indica? ¿Hay un error?

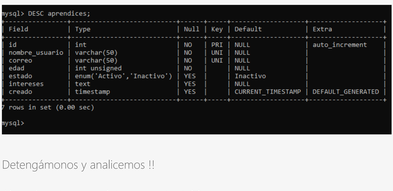
Hagamos uso del **DROP TABLES aprendices**, para hacer uso de esta nueva difinición o actualización.

Ahora si, creemos nuevamente nuestra tabla.



Describamos nuestra nueva tabla, como lo hacemos?

DESC aprendices;



Detengámonos y analicemos !!

Ahora insertaremos datos a nuestra tabla.

INSERT INTO;

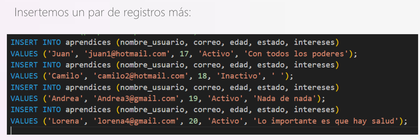
```
INSERT INTO aprendices (nombre_usuario, correo, edad, estado, intereses)
VALUES ('Eliana', 'eylozano@sena.edu.co', 31, 'Activo', 'Con todos los poderes');
```

```
mysql>
mysql> INSERT INTO aprendices (nombre_usuario, correo, edad, estado, intereses)
    -> VALUES ('Eliana', 'eylozano@sena.edu.co', 31, 'Activo', 'Con todos los poderes');
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

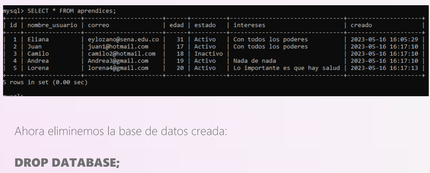


Insertemos un par de registros más:

```
INSERT INTO aprendices (nombre_usuario, correo, edad, estado, intereses)
VALUES ('Juan', 'juan1@hotmail.com', 17, 'Activo', 'Con todos los poderes');
INSERT INTO aprendices (nombre_usuario, correo, edad, estado, intereses)
VALUES ('Camilo', 'camilo2@hotmail.com', 18, 'Inactivo', ' ');
INSERT INTO aprendices (nombre_usuario, correo, edad, estado, intereses)
VALUES ('Andrea', 'Andrea3@gmail.com', 19, 'Activo', 'Nada de nada');
INSERT INTO aprendices (nombre_usuario, correo, edad, estado, intereses)
VALUES ('Lorena', 'lorena4@gmail.com', 20, 'Activo', 'Lo importante es que hay salud');
```

Hagamos una consulta sobre la tabla para visualizar los datos:

SELECT * FROM



Ahora eliminemos la base de datos creada:

DROP DATABASE;