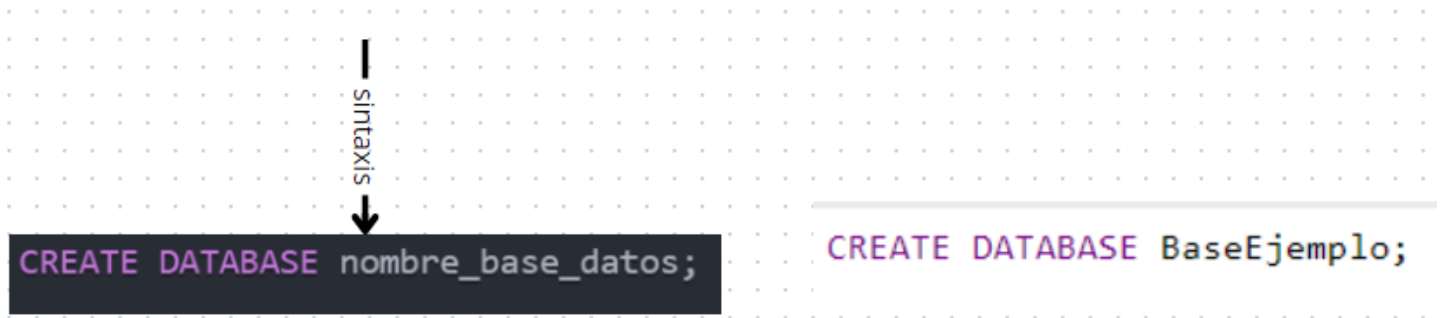


Manejo del DDL (Lenguaje de Definición de Datos) básico

Paso 1: Crear una base de datos

1. Abrir el cliente de MySQL (MySQL Workbench, línea de comandos, etc.).
2. Utilizar el comando CREATE DATABASE para crear una nueva base de datos:

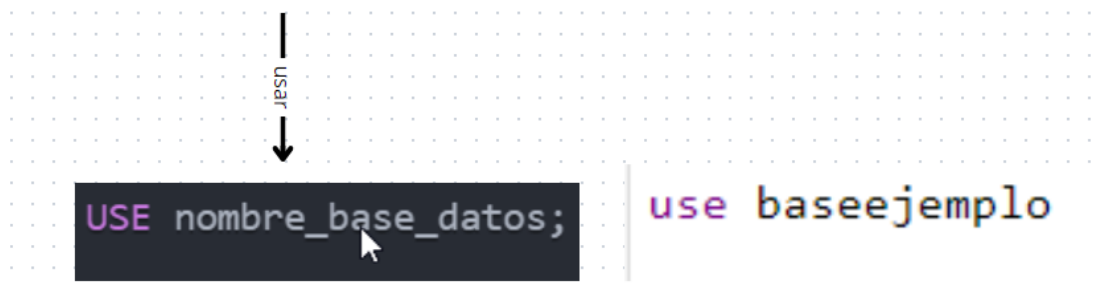


3. Verificar la creación de la base de datos utilizando el comando SHOW DATABASES;

```
1 SHOW DATABASES
```

Paso 2: Seleccionar la base de datos

1. Utilizar el comando USE para seleccionar la base de datos recién creada:



Paso 3: Crear una tabla

1. Entienda qué es una tabla y sus componentes (campos, registros, tipos de datos, etc.).

The diagram shows a table with four columns: **id**, **username**, **password**, and **role**. The first row contains the values 1, arleth64, 12345, and user. The second row contains the values 2, admin, admin27, and admin. Labels with arrows point to the components: "campo" points to the header row, "registro" points to the first data row, and "Tabla" points to the entire table structure.

id	username	password	role
1	arleth64	12345	user
2	admin	admin27	admin

2. Utilizar el comando CREATE TABLE para crear una nueva tabla:

```
CREATE TABLE nombre_tabla (  
    campo1 tipo_dato restricciones,  
    campo2 tipo_dato restricciones,  
    ...  
);
```

3. Tipos de datos más comunes:

Tipo de Dato	Descripción
INT	Entero de longitud fija (longitud depende del subtipo utilizado: TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT)
FLOAT, DOUBLE	Números de punto flotante (precisión de 7 y 15 dígitos decimales, respectivamente)
DECIMAL	Números de punto fijo (precisión decimal exacta)
CHAR	Cadena de caracteres de longitud fija
VARCHAR	Cadena de caracteres de longitud variable
TEXT	Cadena de caracteres de longitud variable grande (hasta 65,535 caracteres)
BLOB	Datos binarios de longitud variable grande (hasta 65,535 bytes)
DATE	Fecha en formato 'YYYY-MM-DD'
TIME	Hora en formato 'HH:MM:SS'
DATETIME	Fecha y hora en formato 'YYYY-MM-DD HH:MM:SS'
TIMESTAMP	Marca de tiempo (fecha y hora)
BOOLEAN	Valor lógico (0 o 1)
ENUM	Conjunto de valores permitidos

4. Restricción

Restricción	Descripción
PRIMARY KEY	Identifica de manera única cada fila de la tabla. No puede haber valores nulos ni duplicados.
FOREIGN KEY	Establece una relación entre dos tablas, asegurando la integridad referencial.
UNIQUE	Garantiza que todos los valores en una columna sean distintos, no puede haber duplicados. Permite valores nulos.
NOT NULL	Especifica que la columna no puede contener valores nulos.
DEFAULT	Proporciona un valor predeterminado para una columna cuando no se especifica un valor al insertar un nuevo registro.
AUTO_INCREMENT	Utilizado para columnas de clave primaria, genera automáticamente un valor numérico único e incremental para cada nueva fila.
CHECK	Permite definir una expresión que debe evaluarse como verdadera para todos los valores insertados o actualizados en esa columna.

```
CREATE TABLE producto (
  id INT PRIMARY KEY AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  descripcion TEXT,
  precio DECIMAL(10, 2) NOT NULL,
  categoria_id INT NOT NULL,
  stock INT NOT NULL DEFAULT 0,
  fecha_creacion DATETIME NOT NULL,
  FOREIGN KEY (categoria_id) REFERENCES categoria(id),
  CHECK (precio > 0)
);
```

```
CREATE TABLE categoria (
  id INT PRIMARY KEY AUTO_INCREMENT,
  nombre VARCHAR(50) NOT NULL UNIQUE,
  descripcion TEXT
);
```

Restricciones utilizadas:

- PRIMARY KEY: La columna id es la clave primaria de la tabla producto.
- AUTO_INCREMENT: La columna id se incrementa automáticamente con cada nuevo registro.
- NOT NULL: Las columnas nombre, precio y categoria_id no pueden tener valores nulos.
- FOREIGN KEY: La columna categoria_id es una clave foránea que referencia a la clave primaria id de la tabla categoria.
- DEFAULT: La columna stock tiene un valor predeterminado de 0.
- CHECK: La restricción CHECK (precio > 0) asegura que el precio del producto sea siempre un valor positivo.

5. Verificar la creación de la tabla utilizando el comando SHOW TABLES

Paso 4: Modificar la estructura de una tabla

1. Agregar un nuevo campo a la tabla utilizando ALTER TABLE:

```
ALTER TABLE nombre_tabla
ADD nuevo_campo tipo_dato restricciones;
```

ejemplo ↓

```
ALTER TABLE producto  
ADD descuento DECIMAL(5, 2) NOT NULL DEFAULT 0;
```

2. Modificar la definición de un campo existente:

```
ALTER TABLE nombre_tabla  
MODIFY campo_existente nuevo_tipo_dato nuevas_restricciones;
```

```
ALTER TABLE producto  
MODIFY precio DECIMAL(10, 3) NOT NULL CHECK (precio >= 0);
```

3. Eliminar un campo de la tabla:

```
ALTER TABLE nombre_tabla  
DROP COLUMN campo_a_eliminar;
```

```
ALTER TABLE users  
DROP COLUMN email_alternativo;
```

Paso 5: Eliminar una tabla

1. Utilizar el comando DROP TABLE para eliminar una tabla existente:

```
DROP TABLE nombre_tabla;
```