

Documentación sobre como crear una base de datos, tablas y atributos.

Realizado por: Yadir Adrian Muñoz Riascos

Materia: DESARROLLO DE BASE DE DATOS

Docente: Brayan Arcos

Paso 1:

Tener el entorno o IDE correctamente configurado para el desarrollo, ya sea MySQL Workbench, MongoDB, HeidiSQL entre otros.

Para este ejemplo documentario usaremos Workbench.



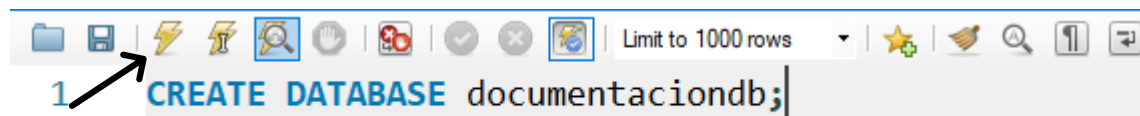
Paso 2:

Crear una base de datos de prueba, en este caso le llamare **documentaciondb** pero en un entorno real se debe nombrar una base de datos dependiendo de los datos que se van a guardar.

Para crear una base de datos ejecutamos el siguiente comando:

```
CREATE DATABASE nombre_base_datos;
```

Luego de escribir los parámetros mencionados, se debe ejecutar dando clic en el icono del rayito amarillo. Como se muestra en la siguiente figura:



IMPORTANTE: no debes tener una base de dato con el mismo nombre para que se pueda crear de forma exitosa. Caso contrario puedes ejecutar el siguiente comando:

```
DROP DATABASE IF EXISTS nombre_base_datos;
```

Y luego ya se podrá ejecutar:

```
CREATE DATABASE nombre_base_datos;
```

Paso 3:

Luego de crear la base de datos podemos crear tablas, para este ejemplo solo se creará una tabla con el nombre **empleados**.

Para crear una tabla se tiene en cuenta los siguientes parámetros:

CREATE TABLE nombre_tabla(

id (este es un identificador, en buenas prácticas siempre se debe registrar este atributo, como también se debe definir como: INT (valor entero) AUTO_INCREMENT (auto incrementable) PRIMARY KEY(llave primaria)).

Atributo1 tipo_dato,

Atributo2 tipo_dato,

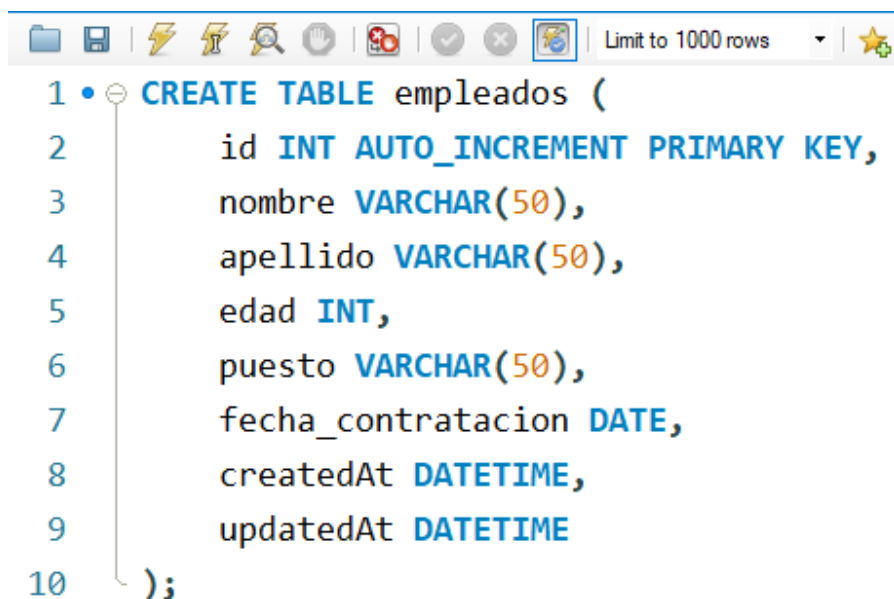
Mas_atributos

);

Luego de haber definido los atributos con su respectivo tipo de dato se procede a dar clic en el rayito amarillo para ejecutar.

IMPORTANTE: Como buenas prácticas es recomendable añadir siempre los campos: id, createdAt, updatedAt. Ya que estos nos informan cuando se crea un registro y cuando se actualiza.

En la siguiente figura se muestra como es la estructura y la forma correcta para crear una tabla:



The image shows a screenshot of a SQL editor interface. At the top, there is a toolbar with various icons for file operations, execution, and search. Below the toolbar, the SQL code for creating a table named 'empleados' is displayed. The code is as follows:

```
1 • ○ CREATE TABLE empleados (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     nombre VARCHAR(50),  
4     apellido VARCHAR(50),  
5     edad INT,  
6     puesto VARCHAR(50),  
7     fecha_contratacion DATE,  
8     createdAt DATETIME,  
9     updatedAt DATETIME  
10 );
```

Descripción de los atributos y su tipo de datos de la imagen anterior:

id, tiene un tipo de dato **INT**, esto quiere decir que el dato a insertar debe ser un número; **AUTO_INCREMENT**, Cada vez que se inserta un nuevo registro en la tabla, la columna definida con **AUTO_INCREMENT** incrementa su valor automáticamente en 1 (o en otro valor si se especifica) respecto al valor anterior; **PRIMARY KEY**, actúa como un identificador único para las filas en la tabla, permitiendo que cada registro sea distinguible de los demás.

nombre VARCHAR(50) se utiliza para almacenar cadenas de texto variables hasta un máximo de 50 caracteres. Se eligió este tamaño para permitir nombres de hasta 50 caracteres, lo cual es suficiente para la mayoría de los nombres personales. VARCHAR permite ahorrar espacio en la base de datos, ya que sólo ocupa el espacio necesario para cada cadena.

apellido VARCHAR(50) Al igual que el campo nombre, se utiliza VARCHAR(50) para almacenar apellidos de hasta 50 caracteres. Esto cubre una amplia variedad de apellidos.

edad INT es suficiente para almacenar la edad de una persona, que generalmente será un valor entre 0 y 120.

puesto VARCHAR(50) Se usa VARCHAR(50) para almacenar el nombre del puesto de trabajo del empleado. El tamaño de 50 caracteres es adecuado para la mayoría de los títulos de puestos de trabajo, que pueden variar en longitud, especialmente si incluyen descripciones detalladas.

fecha_contratacion DATE se usa para almacenar fechas sin incluir la hora. Es ideal para representar la fecha de contratación, ya que no se requiere información de tiempo. El formato es YYYY-MM-DD, lo que facilita la manipulación y comparación de fechas.

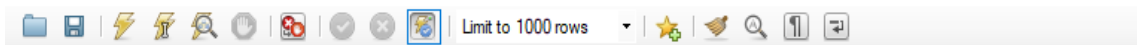
createdAt DATETIME Se utiliza DATETIME para registrar la fecha y la hora exacta en la que se creó el registro. Esto es importante para auditoría y seguimiento de registros, permitiendo saber cuándo fue insertado cada registro en la tabla.

updatedAt DATETIME Similar a createdAt, updatedAt almacena la fecha y hora de la última actualización del registro.

Ahora mostrare como se puede gestionar la base de datos con diferentes comandos

Insertar Datos en la Tabla empleados


Para insertar datos en la tabla **empleados**, se utiliza el comando **INSERT INTO**. Tal como se ve en el siguiente ejemplo:



```
1 • USE documentaciondb;
2
3 • INSERT INTO empleados (nombre, apellido, edad, puesto, fecha_contratacion, createdAt, updatedAt)
4   VALUES ('Tatiana', 'Fonseca', 19, 'Atención al cliente', '2024-01-15', '2024-01-15', '2024-01-15');
```

Actualizar Datos en la Tabla empleados


Para actualizar datos en la tabla **empleados** se utiliza el comando **UPDATE** Tal como se ve en el siguiente ejemplo:



```
1 • USE documentaciondb;  
2  
3 • UPDATE empleados  
4   SET puesto = 'Gerente'  
5   WHERE id = 1;
```

Eliminar un Registro en la Tabla empleados


Para eliminar datos en la tabla **empleados** se utiliza el comando **DELETE** Tal como se ve en el siguiente ejemplo:



```
1 • USE documentaciondb;  
2  
3 • DELETE FROM empleados  
4   WHERE id = 1;
```

Añadir una Columna o atributo Nuevo a la Tabla empleados

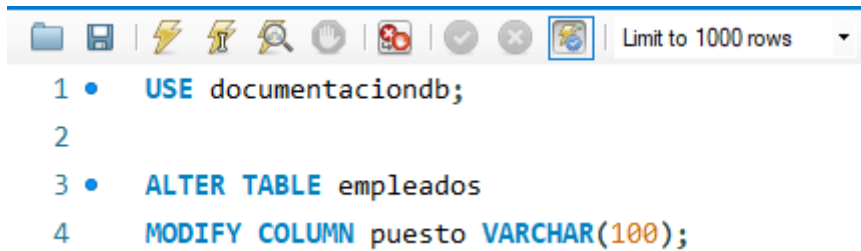
Para añadir una columna nueva a la tabla empleados se utiliza el comando **ALTER TABLE** y **ADD COLUMN**, se usa de la siguiente manera:



```
1 • USE documentaciondb;  
2  
3 • ALTER TABLE empleados  
4   ADD COLUMN direccion VARCHAR(100);
```

Modificar una Columna en la Tabla empleados

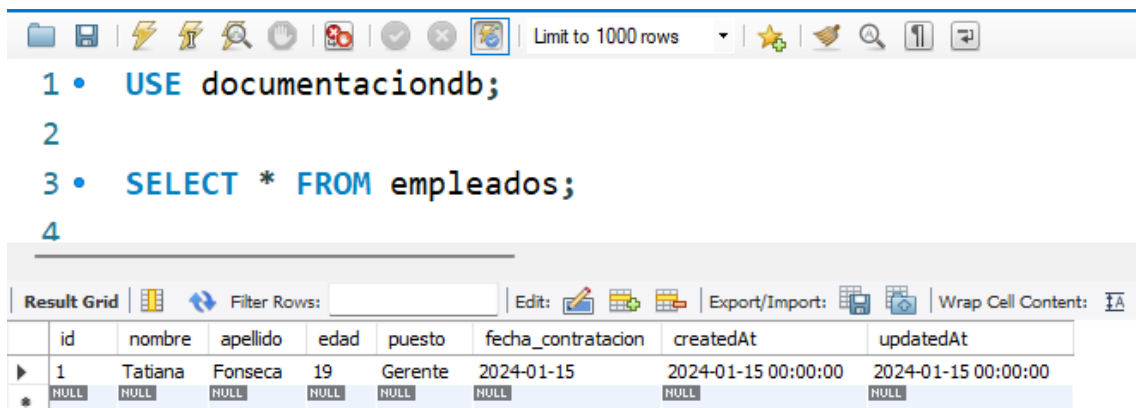
Para modificar una columna nueva en la tabla empleados se utiliza el siguiente comando **ALTER TABLE** y **MODIFY COLUMN** se usa de la siguiente manera:



```
1 • USE documentaciondb;
2
3 • ALTER TABLE empleados
4   MODIFY COLUMN puesto VARCHAR(100);
```

Obtener valores

Para obtener información de los registros que hay registrado en la tabla se hace ejecutando el siguiente comando: **SELECT * FROM nombre_tabla**, cabe mencionar que hay muchas formas de mostrar los datos dependiendo la necesidad. A continuación, se muestra en la imagen como podemos obtener la información registrada en nuestra tabla:



```
1 • USE documentaciondb;
2
3 • SELECT * FROM empleados;
4
```

id	nombre	apellido	edad	puesto	fecha_contratacion	createdAt	updatedAt
1	Tatiana	Fonseca	19	Gerente	2024-01-15	2024-01-15 00:00:00	2024-01-15 00:00:00
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

En conclusión:

Esta documentación proporciona una guía básica sobre cómo crear una base de datos, crear una tabla como también: insertar, actualizar, eliminar, y modificar datos en una tabla de **empleados** utilizando SQL en MySQL Workbench. Estos comandos son esenciales para la gestión de bases de datos, permitiendo una administración eficiente de la información.