BASE DE DATOS

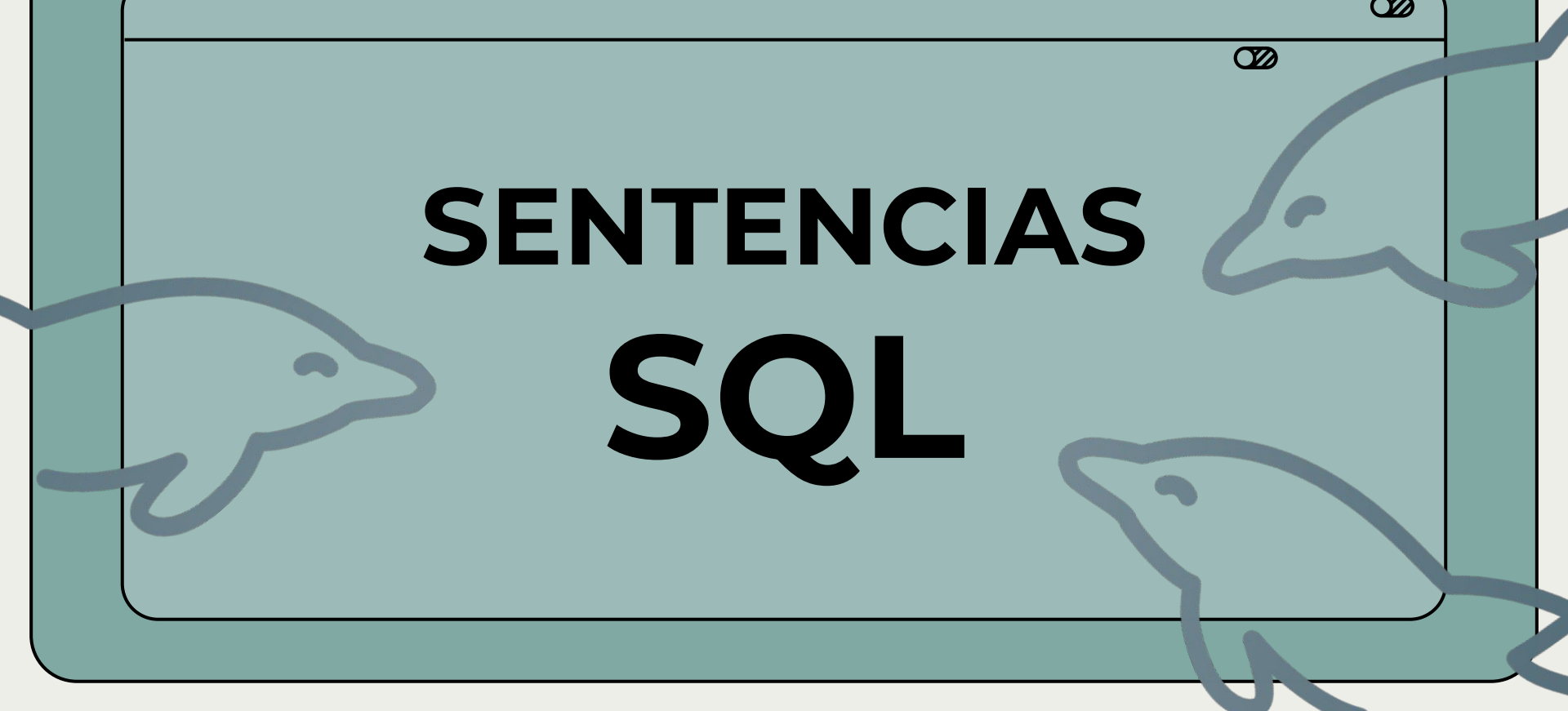
Continuamos con nuestra aventura digital !!

Casa del futuro

En este taller conocerán uno de los motores más populares y gratuitos de base de datos y aprenderán a crear, diseñar y modelar Bases de Datos usando los diagramas de Entidad Relación (DER) con MySQL Workbench.

DATE: 23/06/2023

SENTENCIAS SQL



INDICE

01

CREATE
DROP
ALTER

Comando CREATE DATABASE

Con **CREATE DATABASE** podemos crear una base de datos desde cero.

```SQL
CREATE DATABASE miprimerabasededatos;
USE miprimerabasededatos;
```

Comando CREATE TABLE

Con **CREATE TABLE** podemos crear una tabla desde cero, junto con sus columnas, tipos y constraints.

SQL

```sql
CREATE TABLE nombre_de_la_tabla (
    nombre_de_la_columna_1 TIPO_DE_DATO CONSTRAINT,
    nombre_de_la_columna_2 TIPO_DE_DATO CONSTRAINT
)
```

SQL

```sql
CREATE TABLE post (
    id INT PRIMARY KEY AUTO_INCREMENT,
    titulo VARCHAR(200)
)
```

Ejemplo CREATE TABLE

SQL

```sql
CREATE TABLE peliculas (
    id  INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(500) NOT NULL,
    rating DECIMAL(3,1) NOT NULL,
    awards INT DEFAULT 0,
    release_date DATE NOT NULL,
    length INT NOT NULL
);
```

FOREIGN KEY

Cuando creemos una columna que contenga una id foránea, será necesario usar la sentencia **FOREIGN KEY** para aclarar a qué tabla y a qué columna hace referencia aquel dato.

Es importante remarcar que la tabla “**clientes**” deberá existir antes de correr esta sentencia para crear la tabla “ordenes”.

SQL

```sql
CREATE TABLE ordenes (
    orden_id INT NOT NULL,
    orden_numero INT NOT NULL,
    cliente_id INT,
    PRIMARY KEY (orden_id),
    FOREIGN KEY (cliente_id) REFERENCES clientes(id)
);
```

Comando DROP TABLE

DROP TABLE borrará la tabla que le especifiquemos en la sentencia.

```SQL
DROP TABLE IF EXIST peliculas;
```

Comando ALTER TABLE

ALTER TABLE permite alterar una tabla ya existente y va a operar con tres comandos:

- **ADD**: para agregar una columna.
- **MODIFY**: para modificar una columna.
- **DROP**: para borrar una columna.

Ejemplos ALTER TABLE

```SQL
ALTER TABLE peliculas
ADD rating DECIMAL(3,1) NOT NULL;
```

Agrega la columna **rating**, aclarando tipo de dato y constraint.

```SQL
ALTER TABLE peliculas
MODIFY rating DECIMAL(4,1) NOT NULL;
```

Modifica el decimal de la columna **rating**. Aunque el resto de las configuraciones de la tabla no se modifiquen, es necesario escribirlas en la sentencia.

```SQL
ALTER TABLE peliculas
DROP rating;
```

Borra la columna **rating**.

02 INSERT UPDATE DELETE

INSERT

Existen dos formas de agregar datos en una tabla:

- Insertando datos en **todas** las **columnas**.
- Insertando datos en las **columnas** que **especifiquemos**.

Todas las columnas

Si estamos insertando datos en todas las columnas, no hace falta aclarar los nombres de cada columna. Sin embargo, el orden en el que insertemos los valores, deberá ser el mismo orden que tengan asignadas las columnas en la tabla.

SQL

```sql
INSERT INTO table_name (columna_1, columna_2, columna_3, ...)
VALUES (valor_1, valor_2, valor_3, ...);
```

SQL

```sql
INSERT INTO artistas (id, nombre, rating)
VALUES (DEFAULT, 'Shakira', 1.0);
```

Columnas específicas

Para insertar datos en una columna en específico, aclaramos la tabla y luego escribimos el nombre de la o las columnas entre los paréntesis.

```SQL
INSERT INTO artistas (nombre)
VALUES ('Calle 13');
```

```SQL
INSERT INTO artistas (nombre, rating)
VALUES (‘Maluma’, 1.0);
```

UPDATE

UPDATE modificará los registros existentes de una tabla. Al igual que con **DELETE**, es importante no olvidar el **WHERE** cuando escribimos la sentencia, aclarando la condición.

SQL

```sql
UPDATE nombre_tabla
SET columna_1 = valor_1, columna_2 = valor_2, ...
WHERE condición;
```

SQL

```sql
UPDATE artistas
SET nombre = 'Charly Garcia', rating = 1.0
WHERE id = 1;
```

DELETE

Con **DELETE** podemos borrar información de una tabla. Es importante recordar utilizar siempre el **WHERE** en la sentencia para agregar la condición de cuáles son las filas que queremos eliminar. Si no escribimos el **WHERE**, estaríamos borrando **toda** la **tabla** y no un registro en particular.

```SQL
DELETE FROM nombre_tabla WHERE condición;
```

```SQL
DELETE FROM artistas WHERE id = 4;
```