DigitalHouse >
Coding School

Módulo: MySQL

CREATE, Tipo de datos, DROP

1.

Breve repaso

CREATE TABLE

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

SQL

```sql
CREATE TABLE nombre_de_la_tabla (
    nombre_de_columna_1 TIPO_DE_DATO CONSTRAINT,
    nombre_de_columna_2 TIPO_DE_DATO CONSTRAINT,
);
```

SQL

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

CREATE TABLE EJEMPLO

SQL

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

);
```

DigitalHouse >
Coding School

2.

Ejercicio

Group SOLVING

Se deberán armar grupos de tres y en conjunto deben crear la tabla **generos**.

La misma les recordamos tiene:

1) El campo id, numerico y auto incremental
2) El campo nombre que corresponde al nombre del género

IMPORTANTE:

Si terminaste de resolver las consignas **da una mano al que todavía le falta terminar**

PUESTA EN COMÚN

DigitalHouse >
Coding School

3.

Modificar una tabla

DROP TABLE

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

SQL

```sql
DROP TABLE IF EXIST movies;
```

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

```
ALTER TABLE nombre_de_tabla
ADD columna3 TIPO_DE_DATO [FIRST|AFTER columna2]
MODIFY NUEVO_TIPO_DE_DATO
DROP columna4;
```

ALTER TABLE

SQL

```sql
ALTER TABLE movies
ADD rating DECIMAL(3,1) UNSIGNED NOT NULL;
```

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

SQL

```sql
ALTER TABLE movies
MODIFY rating DECIMAL(4,1) UNSIGNED 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

```sql
ALTER TABLE movies
DROP rating;
```

***Borra** la columna **rating**.*

4.

Ejercicio

Group SOLVING

Se deberán armar grupos de tres y en conjunto deben:

1) Crear la tabla **actores**.
2) Luego de crear la tabla deben modificarla y agregarle el campo fecha_nacimiento.

IMPORTANTE:

Si terminaste de resolver las consignas **da una mano al que todavía le falta terminar.**

PUESTA EN COMÚN

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

```
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)

);
```

5.

Ejercicio

Group SOLVING

Se deberán armar grupos de tres y en conjunto deben:

1) Crear la tabla **peliculas** y que la película tenga asociada un genero.

IMPORTANTE:

Si terminaste de resolver las consignas **da una mano al que todavía le falta terminar.**

PUESTA EN COMÚN

6.

Ejercicio

DigitalHouse >
Coding School

Group SOLVING

Se deberán armar grupos de tres y en conjunto deben crear la tabla pivot **peliculas_actores**.

IMPORTANTE:

Si terminaste de resolver las consignas **da una mano al que todavía le falta terminar.**

Group SOLVING

Se deberán armar grupos de tres y en conjunto deben crear la tabla **peliculas_actores**.

IMPORTANTE:

Si terminaste de resolver las 3 consignas **da una mano al que todavía le falta terminar.**

PUESTA EN COMÚN

The end of blue¿?