

SQL

Soluciones de Data Definition Language (DDL)

CertiDevs

Índice de contenidos

1. Requisitos	1
2. Tabla reseña de película	1
3. Tabla películas favoritas	1
4. Columna last_update	1
5. Modificar columna	2
6. Eliminar columna	2
7. Tabla descuentos	2
8. Añadir columna	2
9. Tabla género película	2
10. Relacionar tablas	3
11. Modificar columna	3
12. Tabla para moneda	3

1. Requisitos

Base de datos sakila.

2. Tabla reseña de película

Crear una tabla para almacenar reseñas de películas:

```
CREATE TABLE movie_reviews (  
  review_id INT AUTO_INCREMENT PRIMARY KEY,  
  film_id INT NOT NULL,  
  customer_id INT NOT NULL,  
  rating DECIMAL(3,2) NOT NULL,  
  review TEXT NOT NULL,  
  review_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (film_id) REFERENCES film (film_id),  
  FOREIGN KEY (customer_id) REFERENCES customer (customer_id)  
);
```

3. Tabla películas favoritas

Crear una tabla para almacenar información de películas favoritas de los clientes.

La tabla debe llamarse customer_favorites.

```
CREATE TABLE customer_favorites (  
  customer_id INT NOT NULL,  
  film_id INT NOT NULL,  
  added_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (customer_id, film_id),  
  FOREIGN KEY (customer_id) REFERENCES customer (customer_id),  
  FOREIGN KEY (film_id) REFERENCES film (film_id)  
);
```

4. Columna last_update

Añadir una columna 'last_update' a la tabla 'actor' para almacenar la fecha y hora de la última actualización del registro:

```
ALTER TABLE actor  
ADD COLUMN last_update TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP;
```

5. Modificar columna

Cambiar el tipo de dato de la columna 'email' en la tabla 'customer' a 'VARCHAR(255)'.

```
ALTER TABLE customer  
MODIFY COLUMN email VARCHAR(255);
```

6. Eliminar columna

Eliminar la columna 'notes' de la tabla 'film_actor':

```
ALTER TABLE film_actor  
DROP COLUMN notes;
```

7. Tabla descuentos

Crear una tabla para almacenar descuentos aplicados a los alquileres de películas.

```
CREATE TABLE rental_discounts (  
    discount_id INT AUTO_INCREMENT PRIMARY KEY,  
    rental_id INT NOT NULL,  
    discount_amount DECIMAL(4,2) NOT NULL,  
    discount_reason VARCHAR(255) NOT NULL,  
    FOREIGN KEY (rental_id) REFERENCES rental (rental_id)  
);
```

8. Añadir columna

Añadir una columna 'phone_number' a la tabla 'staff' para almacenar números de teléfono del personal.

```
ALTER TABLE staff  
ADD COLUMN phone_number VARCHAR(15);
```

9. Tabla género película

Crear una tabla para almacenar información sobre los géneros de las películas

```
CREATE TABLE genre (  
    genre_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
name VARCHAR(50) NOT NULL UNIQUE  
);
```

10. Relacionar tablas

Crear una tabla de relación entre películas y géneros con clave primaria compuesta.

Se trata de una asociación Many To Many.

```
CREATE TABLE film_genre (  
    film_id INT NOT NULL,  
    genre_id INT NOT NULL,  
    PRIMARY KEY (film_id, genre_id),  
    FOREIGN KEY (film_id) REFERENCES film (film_id),  
    FOREIGN KEY (genre_id) REFERENCES genre (genre_id)  
);
```

11. Modificar columna

Cambiar la columna 'rental_duration' en la tabla 'film' para permitir valores NULL.

```
ALTER TABLE film  
MODIFY COLUMN rental_duration TINYINT UNSIGNED NULL;
```

12. Tabla para moneda

Crear una tabla para almacenar la información de los pagos en monedas diferentes:

```
CREATE TABLE payment_currency (  
    payment_id INT NOT NULL,  
    currency_code CHAR(3) NOT NULL,  
    amount DECIMAL(5,2) NOT NULL,  
    PRIMARY KEY (payment_id),  
    FOREIGN KEY (payment_id) REFERENCES payment (payment_id)  
);
```