Ejercicio 6: Haga un análisis del modelo e incluya las restricciones CHECK que considere necesarias.

Tabla actor

Podríamos añadir un CHECK que compruebe que el nombre y el apellido no están vacíos.

ALTER TABLE actor
ADD CONSTRAINT check_actor_name_not_empty CHECK (first_name <> '' AND last_name <> '');

```
alquilerdvd=# ALTER TABLE actor
ADD CONSTRAINT check_actor_name_not_empty CHECK (first_name <> '' AND last_name <> '');
ALTER TABLE
```

Tabla address

Podríamos añadir uno o varios CHECK que comprueben que no haya datos vacíos. Por ejemplo, consideremos que en la base de datos, no es obligatorio almacenar todos los campos de la tabla dirección, pero supongamos que sí algunos atributos que se consideran esenciales para cualquier entrada en la tabla address. Por ejemplo, que la dirección no esté vacía. También se podría realizar con los atributos de teléfono o código postal

ALTER TABLE address
ADD CONSTRAINT check_phone_not_empty CHECK (address <> '');

```
alquilerdvd=# ALTER TABLE address
ADD CONSTRAINT check_phone_not_empty CHECK (address <> '');
ALTER TABLE
alquilerdvd=#
```

Tabla category

Podríamos asegurarnos de que el nombre no esté vacío, de forma similares como hacemos en la tabla actor

ALTER TABLE category
ADD CONSTRAINT check_category_name_not_empty CHECK (name <> '');

```
alquilerdvd=# ALTER TABLE category
ADD CONSTRAINT check_category_name_not_empty CHECK (name <> '');
ALTER TABLE
alquilerdvd=#
```

Tabla city

En la tabla city, podríamos asegurarnos de que el campo city no está vacío

ALTER TABLE city
ADD CONSTRAINT check_city_name_not_empty CHECK (city <> '');

```
ALTER TABLE
alquilerdvd=# ALTER TABLE city
ADD CONSTRAINT check_city_name_not_empty CHECK (city <> '');
ALTER TABLE
alquilerdvd=#
```

Tabla country

Es importante asegurarse de que el nombre del país no esté vacío.

ALTER TABLE country
ADD CONSTRAINT check_country_name_not_empty CHECK (country <> '');

```
alquilerdvd=# ALTER TABLE country
ADD CONSTRAINT check_country_name_not_empty CHECK (country <> '');
ALTER TABLE
alquilerdvd=#
```

Tabla customer

Podríamos añadir un check que compruebe que el campo activebool esté entre los recogidos por un booleano, es decir: 0 o 1 (0 = FALSE y 1 = TRUE)
Como en otras tablas, podríamos.

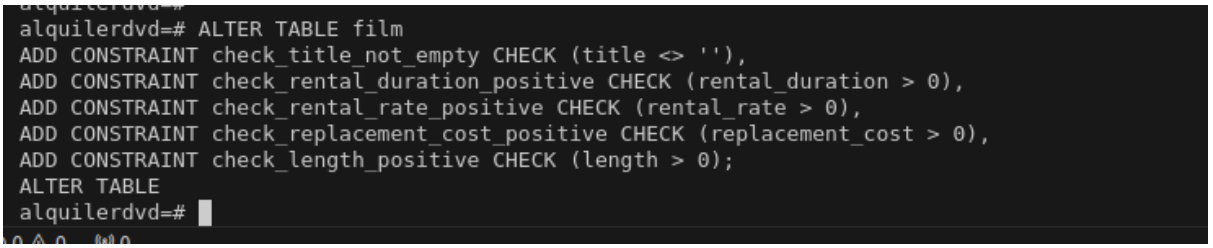
ALTER TABLE customer
ADD CONSTRAINT check_customer_name_not_empty CHECK (first_name <> '' AND last_name <> ''),
ADD CONSTRAINT check_customer_active_boolean CHECK (activebool IN (TRUE, FALSE));

```
alquilerdvd=# ALTER TABLE customer
ADD CONSTRAINT check_customer_name_not_empty CHECK (first_name <> '' AND last_name <> ''),
ADD CONSTRAINT check_customer_active_boolean CHECK (activebool IN (TRUE, FALSE));
ALTER TABLE
alquilerdvd=#
```

Tabla film

En la tabla film podríamos asegurarnos de que tanto la duración de la película, la duración del alquiler de una película, la tarifa de renta y el costo de reemplazo sean positivos, es decir, que sean mayores que cero. También podríamos asegurarnos de que el título de una película no esté vacío.

ALTER TABLE film
ADD CONSTRAINT check_title_not_empty CHECK (title <> ''),
ADD CONSTRAINT check_rental_duration_positive CHECK (rental_duration > 0),
ADD CONSTRAINT check_rental_rate_positive CHECK (rental_rate > 0),
ADD CONSTRAINT check_replacement_cost_positive CHECK (replacement_cost > 0),
ADD CONSTRAINT check_length_positive CHECK (length > 0);



Tablas film_actor, film_category e inventory

No hace falta añadir ninguna restricción CHECK. Sus claves foráneas aseguran su integridad. Y las claves primarias no necesitan restricciones CHECK.

Tabla inventory

Añadiremos una restricción CHECK para evitar que haya nombres de idiomas vacíos.

ALTER TABLE language
ADD CONSTRAINT check_language_name_not_empty CHECK (name <> '');

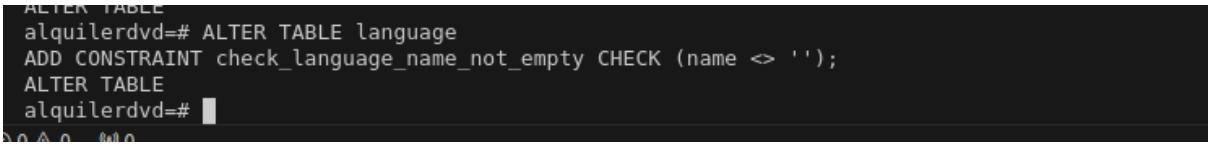


Tabla Payment

En la tabla payment, hay que asegurarse que la cantidad del pago sea mayor o igual que cero (en caso de que queramos que haya la posibilidad de que algunas películas se puedan alquilar durante cierto tiempo durante un período de tiempo, como por ejemplo, una promoción de la tienda. En caso contrario, se puede usar > 0 en vez de >= 0). También hay que asegurarse de que la fecha del pago no esté vacía.

ALTER TABLE payment
ADD CONSTRAINT check_amount_positive CHECK (amount >= 0),
ADD CONSTRAINT check_payment_date_not_null CHECK (payment_date IS NOT NULL);

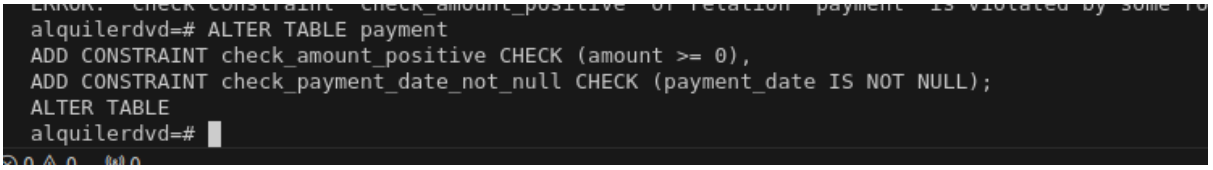


Tabla rental

Asegurarse de que en la tabla rental, los valores de la fecha de alquiler no esté vacío, y que, en caso de que se haya devuelto la película alquilada, la fecha donde se devolvió sea posterior a la fecha de cuando se alquiló la película.

ALTER TABLE rental
ADD CONSTRAINT check_rental_date_not_null CHECK (rental_date IS NOT NULL),
ADD CONSTRAINT check_return_date CHECK (return_date IS NULL OR return_date > rental_date);

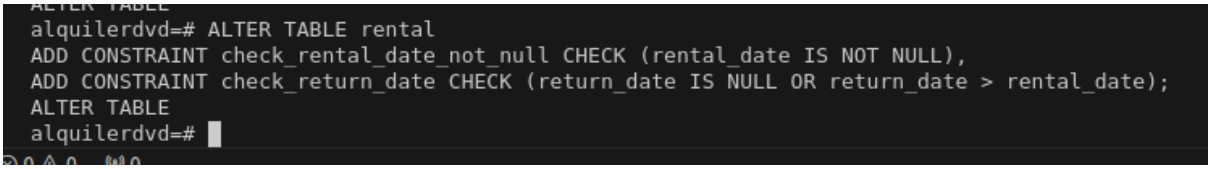


Tabla staff

Hay que cerciorarse de que el nombre y el apellido no estén vacíos, y que el booleano que indica si el staff está activo (FALSE = inactivo y TRUE = activo), esté entre los valores propios de un booleano, es decir, falso o verdadero.

ALTER TABLE staff
ADD CONSTRAINT check_staff_name_not_empty CHECK (first_name <> '' AND last_name <> ''),
ADD CONSTRAINT check_staff_active_boolean CHECK (active IN (FALSE, TRUE));

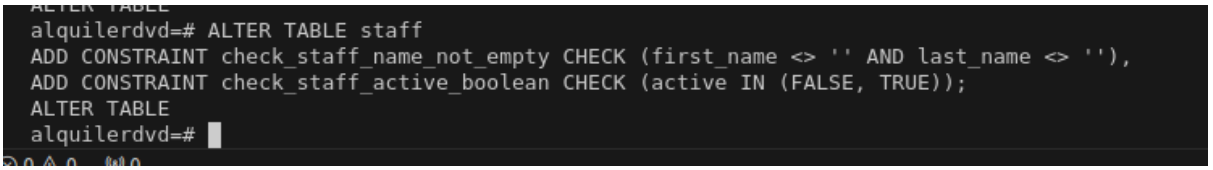


Tabla store

La clave primaria no necesita restricción CHECK, y su clave foránea garantiza la integridad.