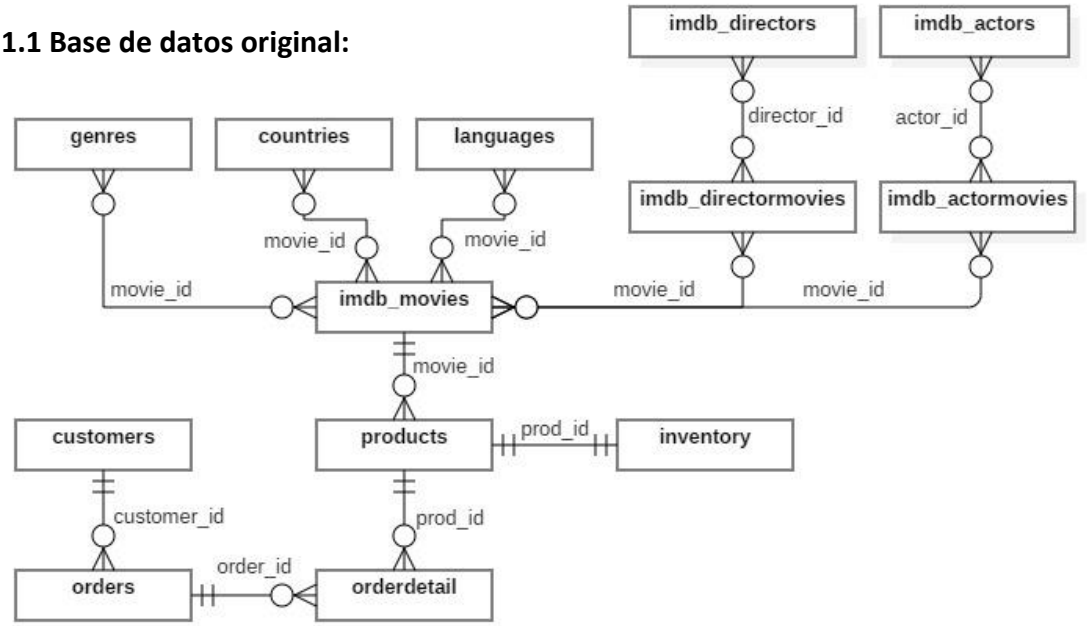
Memoria Práctica 3 Sistemas Informáticos

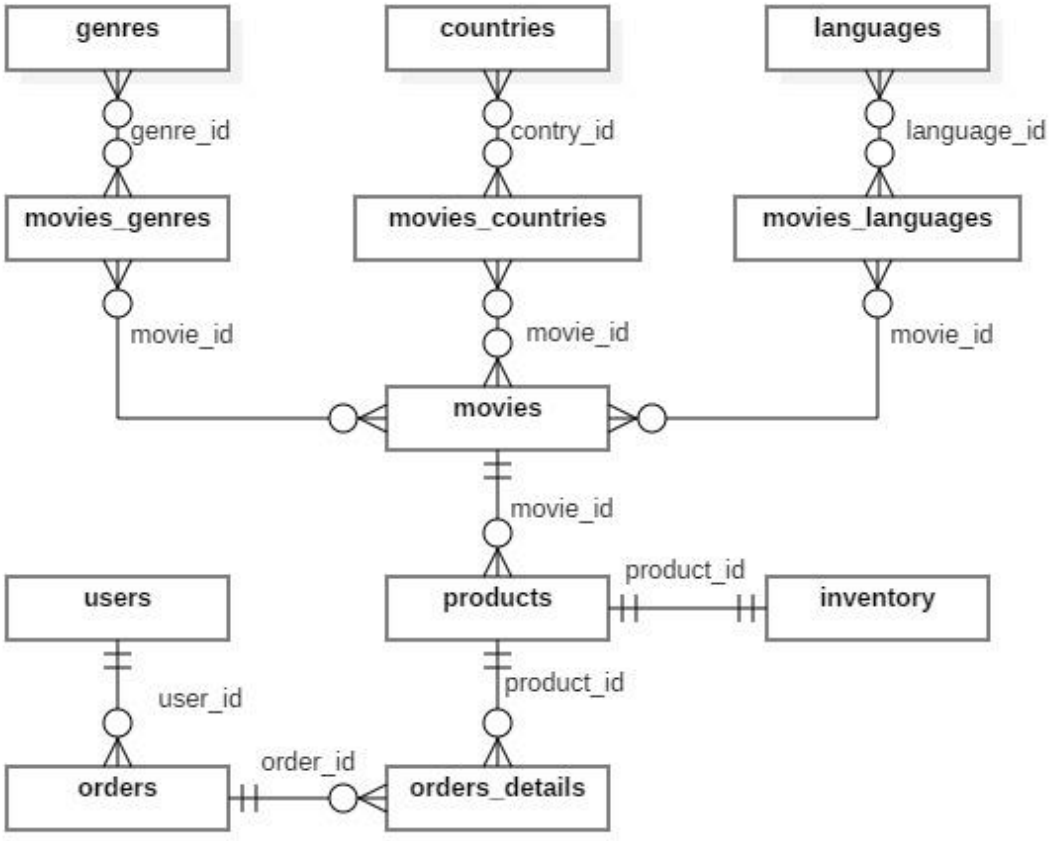
Alejandro Pascual y Víctor Yrazusta

1. Diagramas entidad-relación:

1.1 Base de datos original:



1.2 Base de datos actualizada:



2. Implementación de los requisitos en SQL:

Hemos realizado la implementación siguiendo el enunciado y la guía de estilo de la asignatura.

Hemos decidido eliminar de la base de datos todo aquello que no utilizamos en nuestra aplicación y que no forma parte de los ejercicios propuestos. Esto incluye tanto tablas enteras como columnas concretas.

Para comprobar su correcto funcionamiento hemos hecho manualmente algunas comprobaciones. Estas pruebas se pueden replicar descomentando las líneas que se han incluido en sus ficheros correspondientes.

Adicionalmente, se puede comprobar su correcto funcionamiento a través de la web de prácticas anteriores.

En particular:

- actualiza.sql: elimina toda la información que no utilizamos. Para las nuevas relaciones entre las películas y los géneros, países y lenguajes creamos una nueva tabla, copiamos todas las entradas diferenciadas (asignándoles un id de manera automática) y luego sustituimos las entradas de las tablas originales por sus ids.
- setPrice.sql: primero copia el valor original del precio de cada producto. Después reduce el precio proporcionalmente al número de años que han pasado desde la compra. Redondea los nuevos valores a dos dígitos decimales.
- setOrderAmount.sql: primero crea una vista para obtener los precios de los pedidos a partir de sus partes (almacenadas en la tabla orders_details). El procedimiento almacenado asigna el valor obtenido de esa vista a las entradas cuyo precio sea nulo. Se asegura de que el precio final cuente con los impuestos aplicados a ese pedido.
- getTopVentas.sql: utiliza una vista para obtener las ventas de cada película. El procedimiento almacenado obtiene para cada mes la película estrenada en dicho mes que más ventas ha tenido. Los resultados se dan a partir de la fecha indicada.
- getTopMonths.sql: utiliza dos vistas para obtener las ventas y beneficios de cada mes. El procedimiento almacenado filtra los que superen uno de los mínimos indicados.
- updOrders.sql: utiliza triggers para actualizar la tabla orders cuando se producen cambios en la tabla orders_details. Se utiliza el mismo procedimiento para INSERT, UPDATE y DELETE, restando la diferencia entre el precio anterior y el nuevo al carrito. En los INSERT los valores de OLD se toman como 0 y en los DELETE los valores de NEW se toman como 0.
- updInventory.sql: crea una entrada si el stock acaba en 0 e impide la compra si no hay suficiente stock.