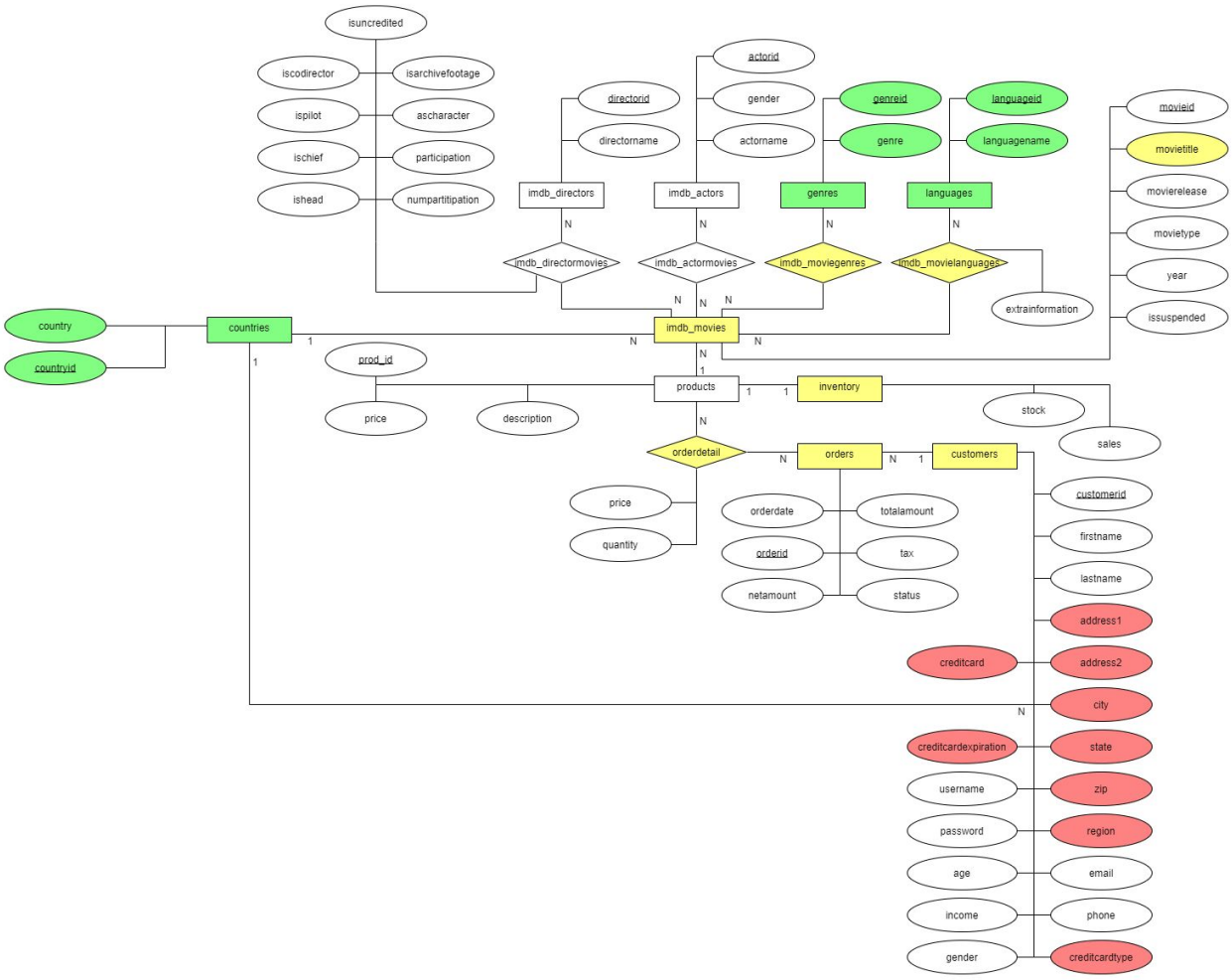
Miguel Herrera y José Manuel Freire

SI - Práctica 2

Group 1391

Parte 1

En esta parte hemos tenido que hacer el diagrama de la base de datos, rediseñarla y hacer un nuevo diagrama. El diagrama final de como nos ha quedado la base de datos es:



En verde están los elementos nuevos que hemos creado nosotros. En amarillo están los elementos que hemos modificado nosotros. Los elementos en rojo son

los que queríamos actualizar, pero que no hemos podido por cuestión de tiempo y que serán actualizados en la entrega final de ser posible.

Los cambios que hemos hecho en la base de datos son:
- Hemos creado una entidad para el idioma y modificado la relación para unirla con las películas.
- Hemos creado una entidad para el género y modificado la relación para unirla con las películas.
- Hemos creado una entidad para el país y quitado su relación. Ahora el país se usa tanto en el cliente como en las películas.
- Hemos modificado orderdetail para añadirle claves foráneas de products y orders.
- Hemos modificado orders para añadirle la clave foránea de customer.
- Hemos modificado inventory para añadirle la clave foránea de product.
- Hemos modificado imdb_actormovies para añadirle las claves foráneas de actor y de movie.
- Queremos añadir la tarjeta de crédito a su propia entidad, para que así una persona pueda tener varias tarjetas de crédito.
- Queremos añadir la dirección a su propia entidad, para que así una persona pueda tener varias direcciones guardadas.

Después de hacer el diagrama y el actualiza.sql pasamos a hacer las queries.

setPrice.sql: En esta consulta el objetivo era poner el precio a cada venta en la tabla orderdetails, reduciendo un 2% por cada año desde la venta de la película. Para resolverlo primero reamos una view para obtener una tabla con el precio y año de cada producto de todos los pedidos. Después creamos una view para obtener el precio de las películas reducidas un 2% anual. Y, por último, completamos la columna price de la tabla orderdetail.

setOrderAmount.sql: El objetivo de esta consulta era ponerle un precio final al pedido, y añadirle el impuesto para conseguir un precio para el cleinte. Lo primero que hacemos es crear una view para obtener la suma de los precios de la película del pedido. Despúes creamos el procedimiento que completa las columnas netamount y totalamount de la tabla orders.

getTopVentas.sql: En este procedimiento almacenado el objetivo era obtener las películas más vendidas cada año junto con su número de ventas y el año el que es la más vendida entre dos años dados. Hemos hecho que en caso de empate se muestren dos ese año. Primero creamos una view para obtener todas las películas vendidas y el año en el que se vendieron. Segundo creamos una view que cuenta cuántas veces una película fue vendida cada año. Tercero creamos una view para obtener el número de veces que ha sido vendida la película más vendida de cada año. Por último terminamos creando el procedimiento que consigue las películas más vendidas de cada año y varias en caso de empate.

getTopMonths.sql: En este procedimiento almacenado el objetivo era obtener los meses que superasen el mínimo de ventas o de dinero que le dieramos como argumentos. Lo primero que hicimos fue crear una view con las ventas totales, ganancias, mes y año. Lo siguiente y último fue hacer el procedimiento que separa de la tabla anterior los meses que cumplen la condición. Devuelve el mes, el año, el dinero y las ventas.

La mayor complicación que hemos tenido con la práctica fue hacer los procedimientos, ya que no conseguíamos que funcionasen correctamente y a veces se quedaban haciendo la query eternamente.