

Práctica 3: Web y Bases de Datos

Práctica 3 Grupo 1361

27 de noviembre de 2019

Sistemas Informáticos

Ingeniería Informática

Universidad Autónoma de Madrid

Eric Morales eric.morales@estudiante.uam.es
Andrés Mena andres.mena@estudiante.uam.es

Introducción

En esta práctica integraremos una base de datos detrás de nuestra página web. En esta ocasión hemos elegido PostgreSQL como SGDB. En ella hemos realizado una actualización de la base de datos que se nos facilitaba , posteriormente creamos algunas funciones y procedimientos almacenados , para terminar con el tema de SQL hemos realizado dos triggers con el objetivo de mantener nuestras tablas correctamente actualizadas.

Además hemos conectado toda nuestra web con esta base de datos, actualizando los apartados de login, register, carrito y página principal.

La memoria ha sido generada utilizando L^AT_EX utilizando una clase (.cls) personalizada y propia. Aprovechando la facilidad que nos da este para añadir código en las memorias de forma clara y las numerosas ventajas de organización que este nos da.

Se adjunta junto con esta memoria todo lo solicitado en el enunciado: pagina web completa (HTML, CSS y JS), con registro, login y pagina principal actualizadas, actualización de la base de datos (actualiza.sql, setPrice.sql), procedimiento almacenado (setOrderAmount.sql) y funciones SQL (getTopVentas.sql, getTopMonths.sql) y triggers SQL (udpOrders.sql, udpInventory.sql) y por último diagrama ER.

Discusión sobre la base de datos

Antes de comenzar a realizar modificaciones en esta practica, hemos realizado un análisis extenso de la base de datos, para comprenderla bien, analizar posibles inconsistencias, y modificar ciertos aspectos para optimizar la base de datos, y hacer que se adapte mejor a las necesidades de nuestra página web.

Para analizar en profundidad cada una de las tablas, y las modificaciones que hemos llevado a cabo sobre cada una de ellas, dividiremos la memoria en secciones para cada tabla. En general, hemos cambiado la gestión de las claves primarias y foráneas, gestión de atributos multivaluados, y arreglo de inconsistencias. También se ha modificado la gestión de series para la asignación automática de ids, de forma que cuando se registre, por ejemplo, un nuevo usuario, se le asigne el id más alto disponible. Esta misma idea se aplica a las tablas de customers, actors, countries, directors, genres, languages, movies, orders y products.

Consulta setPrice

Teniendo en cuenta que los precios de las películas se han ido incrementando un 2 % anualmente, hemos elaborado la consulta setPrice.sql que completa la columna *price* de la tabla *orderdetail*, sabiendo que el precio actual es el de la tabla *products*.

Tabla actormovies

Modificamos los campos de actorid y movieid para que sean claves foraneas de las tablas actors y movies respectivamente, y hacemos que estas dos juntas sean la clave primaria de la tabla, ya que un actor puede actuar de forma única en una película.

Tabla directormovies

En la base de datos entregadas, en esta tabla se calculaba la secuencia de ids para los directores y las películas. Hemos considerado que esto es un error, ya que los ids únicos de estos campos deben de calcularse al insertar nuevas películas o directores en sus respectivas tablas (movies y directors), pero no en esta tabla que sirve únicamente como relación entre ambas. Además, hacemos que los campos movieid y directorid sean la clave primaria de la tabla, ya que un director puede actuar de forma única en una película.

Tabla orders

En esta tabla simplemente hemos asignado valores por defecto a los campos, para que a la hora de crear un pedido vacío para un cliente (lo necesitamos hacer cuando un usuario añade su primer artículo al carrito), los campos de netamount y totalamount sean 0, y el campo de impuestos (tax) sea 21.

Destacar de esta tabla, que, como podemos ver en el esquema, el campo netamount es un campo derivado de la tabla, ya que se puede obtener directamente a partir de los campos netamount y tax.

Tabla orderdetail

Al comenzar a realizar consultas en la base de datos, nos percatamos de que esta tabla no estaba bien formada. El objetivo de esta tabla era relacionar un pedido con un producto del pedido, indicando cuantas veces se había comprado ese producto en el pedido, y el precio por unidad del producto. Sin embargo, al intentar establecer como clave primaria de la tabla la unión del producto con el pedido, nos dimos cuenta de que había un error porque había repeticiones de estos campos, cuando lo que se tenía que hacer era sumar las cantidades de los productos que tuviesen el mismo orderid y prodid. Una vez solucionado este problema, mediante la actualización del campo quantity en las filas que coincidían el orderid y prodid, ya pudimos asignar a esta tabla la clave primaria previamente mencionada.

Además de las modificaciones previamente mencionadas, hemos establecido como claves foráneas de sus respectivas tablas los campos orderid y prodid.

Tabla customers

Las principales modificaciones necesarias para esta tabla, es adaptarla a las necesidades de nuestra aplicación. Para ello, hemos modificado las restricciones de qué campos debían ser NULL, para que los registros de nuestros nuevos usuarios funcionen de forma correcta. Además, en nuestra aplicación el nombre de usuario tiene que ser único, y en esta base de datos no se cumplía, por lo que para estos usuarios que estaban duplicados, hemos decidido concatenar el nombre de usuario junto con el id de ese usuario, para asegurarnos que así todos los nombres de usuario sean únicos, y poder poner este campo de la tabla como unique.

Finalmente hemos el campo de country, haciendo uso de la nueva tabla countries que hemos creado (explicada mas adelante). Esto nos permite que, en caso de tener que hacer una modificación más adelante en el nombre de un país, o querer añadir algún detalle especial a un país, no nos obligue a modificar la información de toda la base de datos.

En esta tabla podría haber atributos multivaluados, como por ejemplo el telefono, pero por no añadir más tablas al esquema de la base de datos lo hemos dejado como un único telefono para cada customer.

Tablas countries, languages, genres

Se han modificado las tablas imdb_moviecountries, imdb_moviegenres, imdb_movielanguages. Estas modificaciones se han realizado por dos motivos: Primero, para que los countries, languages y genres sean atributos multivaluados, de forma que una película pueda haber sido grabada en varios países, o pertenezca a varios géneros o haya sido interpretada en varios idiomas.

El segundo motivo son las cuestiones de mantenimiento de la base de datos, si queremos modificar, por ejemplo, el nombre de un país, simplemente tendremos que acceder a la tabla de países y modificar una entrada, mientras que si tenemos un campo con el nombre del país en todas las películas, tendríamos que ir modificando todos los campos que coincidiesen con el país que queremos modificar.

Para ello, lo primero que hemos hecho es crear 3 nuevas tablas: countries, languages y genres. Cada una de estas tablas contendrá la información que queramos guardar sobre cada país (de momento solo el nombre), cada idioma (que contendrá el nombre del idioma y un campo con información extra), y cada género (contendrá el nombre del género). Además de estos campos, cada entrada en sus respectivas tablas tendrá un identificador único (genreid, languageid y countryid), para los que hemos creado una secuencia de asignación de ids.

Una vez creadas estas tablas, para relacionar las películas con cada una de las nuevas tablas, lo que hemos hecho ha sido modificar las tablas imdb_moviecountries, imdb_moviegenres y imdb_movielanguages para que ahora solo mantenga la información del id de la movie (como clave foránea) y el id de la respectiva tabla con la que se relaciona (por ejemplo, un countryid, que también será clave foránea de su tabla). Los únicos valores que dejaremos almacenados en estas tablas serán los dos ids mencionados previamente, que juntos formarán la clave primaria. Este mismo proceso se aplica para las tres tablas. Finalmente, mencionar que hemos cambiado los nombre de esta tabla a imdb_languagemovies, imdb_genremovies y imdb_countrymovies para seguir el mismo esquema de nombres que el resto de las tablas de la base de datos.

Tabla alertas

Hemos creado esta tabla para que en caso de que nos quedemos sin stock de un producto, introduzcamos aquí la información. Esta tabla se llenará automáticamente mediante el uso de triggers (explicado mas abajo), y simplemente contendrá los prod_id de los productos de los cuales se ha agotado el stock.

1. Procedimiento almacenado, Funciones y Triggers

Como ya se ha comentado en la introducción hemos realizado el código SQL necesario para conectar nuestra página web con la base de datos solicitada.

Procedimiento almacenado

Hemos realizado un procedimiento almacenado, setOrderAmount, que completa las columnas *netamount* (suma de los precios de las películas del pedido) y *totalamount* (*netamount* más impuestos) de la tabla *orders* cuando estas no contengan ningún valor. Es invocado justo después de su declaración para realizar una carga inicial.

Funciones

Hemos realizado dos funciones, la primera es *getTopVentas*, que recibe como argumento un año y devuelve para ese año y los siguientes la película que más se ha vendido. Esta función es la mas importante ya que es la que figura en la página inicial de la web (Figura 1).

La segunda función es *getTopMonths*, que recibe unos umbrales de número de productos y de importe (*totalamount*) acumulados, y devuelve los meses en los que se ha superado alguno de los dos umbrales, junto con su importe y productos.

Triggers

Hemos programado dos triggers, el primero updOrders, que actualiza la información de la tabla *orders* cuando se añade, actualiza o elimina un artículo al carrito. Además hemos realizado una comprobación adicional, que es que si la cantidad de productos de un *orderdetail* es menor o igual que cero, se elimina esta entrada de la base de datos. (Figuras 4 y 5).

2. Pagina Web

Hemos realizado algunos cambios en la página web para poder integrar todos estas mejoras. Las cuales consisten básicamente en incluir código SQL dentro del código *Python*.

Por motivos visuales, hemos mantenido el income de los usuarios como si fuese el saldo de estos, aunque en esta práctica no tiene funcionalidad alguna. Además, la funcionalidad que se ha perdido al introducir al base de datos, sigue estando disponible de forma visual, pero no tiene efecto alguno sobre la pagina web.

Register y Login

Hemos creado de nuevo las funciones internas del apartado de registro y login, con la finalidad de que estas interactúen correctamente con la tabla *customers* de la base de datos.

Página principal

Como se puede ver en la Figura 1, hemos añadido la tabla resultado de la función `getTopVentas(3)` para de esta forma obtener las películas más vendidas de los últimos 3 años, esta tabla además se va actualizando a medida que jugamos con nuestra web, como se puede ver en la imagen, la película más vendida de 2019 tiene 785 ventas, las cuales han sido realizadas desde el apartado de carrito de la página.

Además hemos modificado las imágenes por una genérica, ya que por desgracia estas no venían incluidas en la base de datos y además hemos modificado las películas mostradas en la página inicial, para que coincidan con las que contiene la base de datos.

Carrito

Aprovechando la utilidad de los triggers programados anteriormente, el carrito funciona de una forma bastante sencilla. Permite añadir películas y aumentar la cantidad de las mismas, además, si la cantidad llega a 0, se elimina la entrada de la base de datos. Al finalizar la compra se comprueba si el saldo es suficiente y se muestra el resultado por pantalla, si este no es afirmativo el carrito se mantiene intacto. (Figuras 2, 3, 4 y 5).

The screenshot shows the homepage of the Memo Video website. At the top, there is a dark header bar with the word "MEMO" in white. On the right side of the header, there are links for "Añadir Saldo: 22323 €", "Cerrar sesión", and a user profile icon labeled "benton". Below the header, there is a sidebar on the left with links for "Inicio", "Compra Actual", and "Historial pedidos". The main content area features a large banner with the text "MEMO VIDEO" and "Bienvenido a Memo, tu plataforma de contenido audiovisual de confianza". Below the banner is a search bar with the placeholder "Buscar película..." and a "Buscar" button. There is also a "Filtrar por Categorías" button. In the background of the banner, there is a blurred image of a person's hand reaching for a DVD or Blu-ray disc. At the bottom of the page, there is a section titled "Top ventas últimos 3 años" with a table. The table has three columns: "Año", "Película", and "Ventas". The data from the table is as follows:

Año	Película	Ventas
2017	Wizard of Oz, The (1939)	134
2018	Life Less Ordinary, A (1997)	134
2019	'Night, Mother (1986)	785

At the very bottom of the page, there is a footer bar with the text "Prácticas de Sistemas Informáticos - Usuarios Actuales: 909 - Eric Morales y Andrés Mena".

Figura 1: Pagina principal con tabla getTopVentas

M E M O

Añadir Saldo: 22323 € Cerrar sesión  benton

	Producto	Precio	Cantidad	Operaciones carrito
Inicio				
Compra Actual				
Historial pedidos				
	 'Breaker' Morant (1980) (Standard)	13€	 1 	
	Resumen pedido	Precio total: 13.00€		

Sobre nosotros

Términos de uso

Prácticas de Sistemas Informáticos - Usuarios Actuales: 925 - Eric Morales y Andrés Mena

Figura 2: Carrito con cantidad de películas aceptado

The screenshot shows a user interface for a shopping cart. At the top, there is a dark header bar with the text "MEMO" on the left and "Añadir Saldo: 22323 €" and "Cerrar sesión" on the right. Below this, a user profile icon and the name "benton" are visible. The main area has a light gray background. On the left, there is a sidebar with links: "Inicio", "Compra Actual" (which is highlighted in blue), and "Historial pedidos". In the center, there is a table with four columns: "Producto", "Precio", "Cantidad", and "Operaciones carro". Below the table, the text "Carrito vacío" is displayed. At the bottom of the page, there are two links: "Sobre nosotros" and "Términos de uso". A thin horizontal bar at the very bottom contains the text "Prácticas de Sistemas Informáticos - Usuarios Actuales: 943 - Eric Morales y Andrés Mena".

Figura 3: Compra finalizada con éxito

M E M O

Añadir Saldo: 22323 € Cerrar sesión  benton

	Producto	Precio	Cantidad	Operaciones carrito
Inicio				
Compra Actual				
Historial pedidos				
	 'Breaker' Morant (1980) (Standard)	5499€	423	 
	Resumen pedido	Precio total: 6653.79€		

Sobre nosotros

Términos de uso

Prácticas de Sistemas Informáticos - Usuarios Actuales: 924 - Eric Morales y Andrés Mena

Figura 4: Compra con cantidad que supera al stock

M E M O

Añadir Saldo: 22323 € Cerrar sesión  benton

Inicio Alguna de las películas del carrito se ha quedado sin stock. Lo sentimos, no se ha realizado la compra

Compra Actual

Historial pedidos

Producto	Precio	Cantidad	Operaciones carro
 'Breaker' Morant (1980) (Standard)	5499€	423	 
Resumen pedido	Precio total: 6653.79€		

Sobre nosotros

Términos de uso

Prácticas de Sistemas Informáticos - Usuarios Actuales: 955 - Eric Morales y Andrés Mena

Figura 5: Intento de compra fallida por falta de stock