

SISTEMAS INFORMÁTICOS

Memoria práctica 3

Muuuvie
Web de venta de películas

Marta García Marín
Jesús Daniel Franco López

Índice

Introducción	3
Discusión acerca de la DB suministrada	3
Cambios realizados con actualiza.sql	3
Diagrama ER resultante	4
Consultas solicitadas	5
Evidencias de resultados obtenidos	6
Cambios en el código	8
Ficheros	9
Referencias	10

Introducción

Esta práctica ha consistido en la incorporación de una base de datos en nuestra página web dinámica. Para ello, hemos utilizado HTML, Python, JavaScript , CSS, JQuery y SQL, centrándonos en este principalmente.

Discusión acerca de la DB suministrada

La base de datos proporcionada en el enunciado constaba de algunos fallos, entre los que se incluían la ausencia de tablas para relaciones con atributos multievaluados.

En general nos ha parecido que la información dada era bastante completa, pero hemos encontrado errores y carencia de atributos necesarios para nuestra Web.

Aparte de todo esto, la mayoría de tablas no tenía claves foráneas asociadas a los id que referenciaba, por lo que también las hemos añadido.

En un principio se nos ocurrió crear una db desde cero y poblarla a partir de los datos suministrados, pero nos dimos cuenta de que no era necesario. En vez de esto, añadimos las columnas importantes para nuestra aplicación, como el campo cvc para los customers.

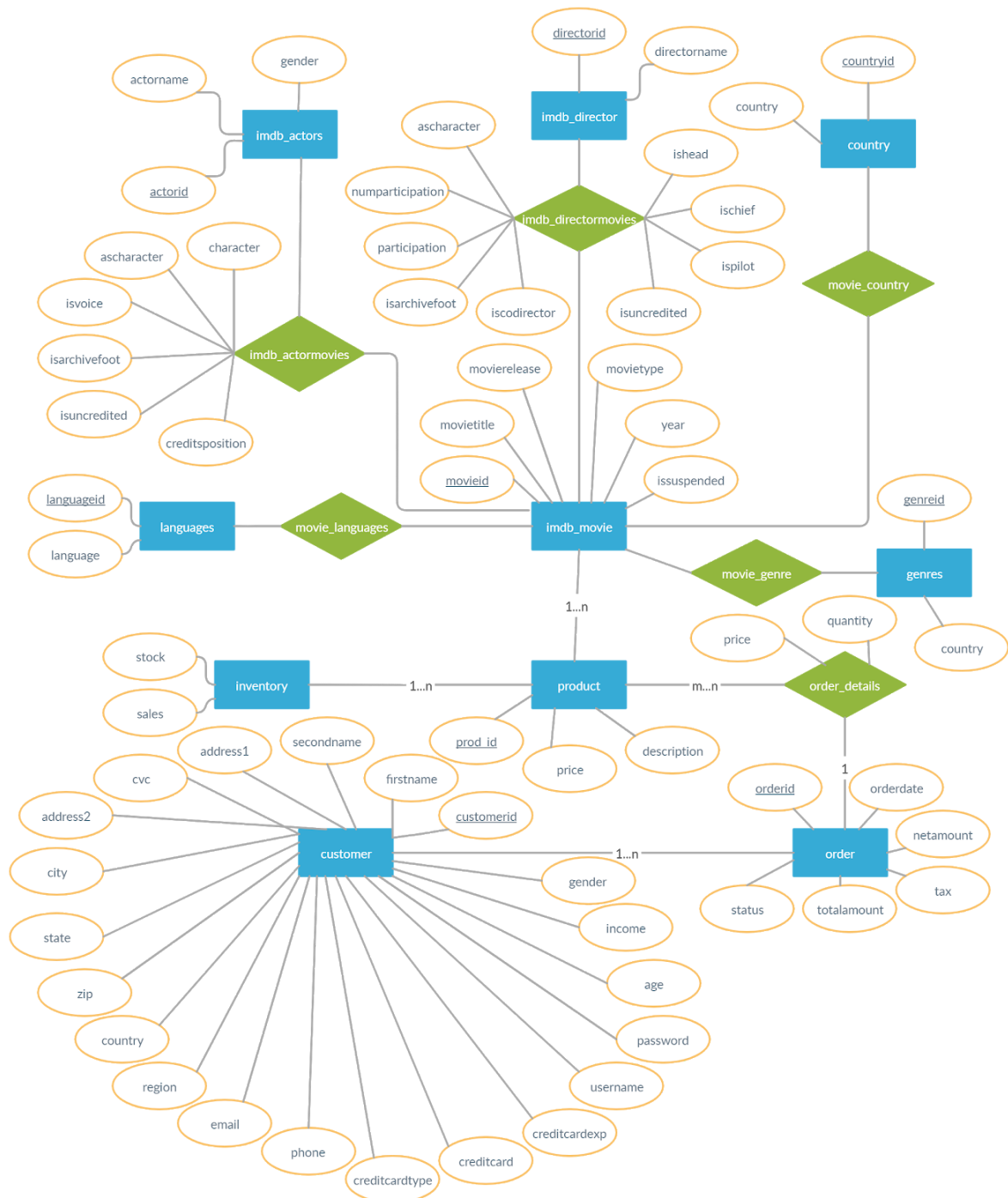
Algo que nos ha chocado bastante y afecta a la estética de nuestra aplicación web es la ausencia de referencias a las imágenes de cada una de las películas, a los póster.

Cambios realizados con actualiza.sql

El fichero actualiza.sql consta de las siguientes partes:

- Adición de las foreign keys a todas las tablas, para poder relacionarlas entre sí.
- Creación de la tabla de países (con un countryid y un país), así como la tabla relación de los países con su película correspondiente (con un countryid y un movieid), y sus foreign keys.
- Creación de la tabla de géneros (con un genreid y un género), así como la tabla relación de los géneros con su película correspondiente (con un countryid y un genreid), y sus foreign keys.
- Creación de la tabla de idiomas (con un languageid y un idioma), así como la tabla relación de los idiomas con su película correspondiente (con un languageid y un movieid), y sus foreign keys.
- Alteración de la tabla customers, donde hemos añadido la columna cvc para la tarjeta de crédito, estableciendo los valores de los clientes ya registrados a '000'

Diagrama ER resultante



Consultas solicitadas

getTopVentas

Probamos esta función obteniendo las películas más vendidas desde 2003

```
SELECT * FROM getTopVentas('2003')
```

	year text	movieid integer	movietitle character varying	sales integer
1	2003	232941	Living Dead Girl (2003)	184
2	2004	318168	Princess Mononoke: Making of (2004)	161
3	2005	171324	Heavyweights (2005)	167
4	2006	279714	Night Fliers (2006)	187

La consulta devuelve la película más comprada para cada uno de los años siguientes al dado como argumento.

Para ello hemos unido las tablas imdb_movies, products e inventory para:

- Conocer en inventory las que tienen más sales
- Tomar el producto que es
- Saber las película a la que se refiere
- Group by el año, ya que sólo vamos a coger uno de ellos

getTopMonths

Realizando la consulta con los parámetros aportados en el enunciado, `SELECT * FROM getTopMonths(320000,19000)`, obtenemos como resultado todos los meses menos los del 2013, ya que estos umbrales se superan siempre a partir del 2014.

Para comprobar mejor su funcionamiento, hemos realizado esta consulta con umbrales más altos: 2350000 euros y 19000 artículos

```
SELECT * FROM getTopMonths(2350000,19000)
```

	anno double precision	mes double precision	total numeric	productos bigint
1	2015	9	2328868.6470660673301140	19133
2	2015	10	2296022.5886119283108690	19086
3	2017	7	2439135.5199923103938850	19280
4	2018	1	2375297.0392156856245625	18598
5	2018	9	2404409.3911764700148675	18387
6	2018	10	2441166.7627450974002480	18576

Evidencias de resultados obtenidos

Capturas de pantalla de la Web

Aunque no lo solicitaba el enunciado, también hemos hecho la opción de novedades, muuuvies más vistas y todas las muuuvies, como se muestra a continuación. Debido a que no hay imágenes de los posters en la base de datos, hemos decidido mostrar estas con un poster común para todas.



Al acceder a una película, observamos que se muestran todos sus campos y te da la opción de añadirla al carrito



Al añadirlos al carrito se muestran en él como en la práctica anterior, permitiendo que se finalice la compra, que se añaden más pelis y que se eliminen estas, haciendo así uso de los triggers.



“MUUVIE”



maca@re.na

Buscar

Novedades

Muuuvies + vistas

Todas

Elegir categoría

Mi carrito

Película	Precio	Mantener?
Dick Tracy (1990)	11 ubres	eliminar
Thelma & Louise (1991)	10 ubres	eliminar

Finalizar compra

© 2019, muuuvie [Términos y condiciones](#)

Observamos que la compra se realiza con éxito



“MUUVIE”



maca@re.na

Buscar

Novedades

Muuuvies + vistas

Todas

Elegir categoría

Mi carrito

Película	Precio	Mantener?
----------	--------	-----------

Compra realizada con éxito

© 2019, muuuvie [Términos y condiciones](#)

Cambios en el código

Los principales cambios en el código se han basado en el acoplamiento de las funciones relacionadas con la base de datos. Esta nos ha permitido prescindir de los archivos json del usuario y de las películas, realizando todos los accesos mediante consultas a la base de datos.

La más significativa de las modificaciones ha sido la incorporación del archivo database.py, donde hemos definido todas y cada de las funciones necesarias para hacer consultas a nuestra base de datos.

Las funciones de este archivo se encargan de, mediante `connection.execute(query)`, obtener o modificar datos de la db.

Aparte, hemos modificado los templates correspondientes a index, carrito, login_register y película. Esto ha sido necesario ya que ahora no accedemos a las películas desde el json, sino que lo hacemos de la db. Por este motivo, el nombre de los campos a los que se accede desde los templates cambia.

Por último, hemos modificado el archivo routes.py para satisfacer los requerimientos pedidos por el enunciado. Los cambios más importantes los hemos llevado a cabo en los servicios de acceso y modificación de carrito y autenticación y registro de los usuarios.

Cabe destacar que, de ahora en adelante, un usuario podrá acceder a su cuenta únicamente con su mail, y no con su username. ¿Por qué hemos tomado esta decisión? Porque en la base de datos suministrada existían diferentes usuarios con el mismo username, y hemos llegado a la conclusión de que el login mediante el mail era el único procedimiento posible para garantizar la no pérdida de información. Las otras alternativas eran agrupar la tabla customers por el username y coger el primero que encontrásemos.

Ficheros

enumeración de TODOS los ficheros

```
/public_html/app/sql
    /public_html/app/sql/actualiza.sql
    /public_html/app/sql/dump_v1.3.sql
    /public_html/app/sql/getTopMonths.sql
    /public_html/app/sql/getTopVentas.sql
    /public_html/app/sql/setOrderAmount.sql
    /public_html/app/sql/setPrice.sql
    /public_html/app/sql/updInventory.sql
    /public_html/app/sql/updOrders.sql
/public_html/app/catalogue
    /public_html/app/catalogue/catalogue.json
/public_html/app/static
    /public_html/app/static/files
        /public_html/app/static/files/terms_and_cond.md
    /public_html/app/static/images
        /public_html/app/static/images/disenio
            mismo contenido que en la entrega anterior
        /public_html/app/static/images/pelis
            mismo contenido que en la entrega anterior
    /public_html/app/static/scripts
        /public_html/app/static/scripts/cambiar_contrasena.js
        /public_html/app/static/scripts/validar_login.js
        /public_html/app/static/scripts/validar_registro.js
    /public_html/app/static/styles.css
/public_html/app/templates
    /public_html/app/templates/ayuda.html
    /public_html/app/templates/base.html
    /public_html/app/templates/cambiar_contrasena.html
    /public_html/app/templates/carrito.html
    /public_html/app/templates/historial.html
    /public_html/app/templates/index.html
    /public_html/app/templates/login.html
    /public_html/app/templates/login_registro.html
    /public_html/app/templates/pelicula.html
    /public_html/app/templates/perfil.html
    /public_html/app/templates/registro.html
/public_html/app/thesessions
/public_html/app/usuarios
/public_html/app/__init__.py
/public_html/app/__main__.py
/public_html/app/routes.py
```

/public_html/app/database.py

/public_html/start.wsgi

(imágenes utilizadas en el diseño y las carátulas de las películas)

Referencias

1. Documentación de PostgreSQL:
<https://www.postgresql.org/docs/9.3/static/>
<https://www.postgresql.org/docs/9.3/static/sql-commands.html>
<https://www.w3schools.com/sql/>
2. Acceso a bases de datos en Python usando SQLAlchemy:
<https://www.sqlalchemy.org/>
3. Diapositivas de moodle
4. Stack Overflow para resolver algunas dudas: <https://es.stackoverflow.com/>
5. Guías básicas de Python, JavaScript, HTML y JQuery