

Práctica 3. Programación Web y Bases de Datos

Versión 1.1 29/10/2014

Objetivos docentes

Tras la realización de la Práctica 3, el estudiante debe ser capaz de:

- Entender los conceptos involucrados en el diseño de una base de datos relacional, en especial las relaciones que se deben establecer entre las distintas tablas.
- Realizar consultas de diferente complejidad sobre bases de datos relacionales.
- Entender y manejar con agilidad los Triggers en una base datos.
- Entender y ser capaz de implementar Funciones y Procedimientos Almacenados
- Acceder a bases de datos desde el lenguaje PHP, realizando actualizaciones y consultas bajo demanda del usuario

Enunciado

Se solicitan a continuación una serie de consultas de cara a elaborar informes sobre la base de datos. Si los datos existentes en la base de datos no son suficientes para comprobar las consultas adecuadamente el alumno deberá incorporar a la misma la información extra necesaria.

Trabajo a Realizar:

- a) En primer lugar, se deberá efectuar un proceso de ingeniería reversa sobre la base de datos suministrada. En particular, se deberá:
 - Obtener el diagrama entidad-relación correspondiente. Para su representación se debe utilizar la notación vista en teoría. Tablas (arrays) y tablas asociativas de PHP.
 - Discutir el diseño de la base de datos, ventajas y desventajas de las decisiones implícitas en el mismo.
 - Completar aquellos aspectos que se consideren necesarios, tales como restricciones, claves externas, etc.
- b) Para garantizar la integridad de los datos, crear las tablas correspondientes y convertir los atributos multivaluados 'moviecountries', 'moviegenres' y 'movielanguages' en relaciones entre la tabla 'movies' y las tablas creadas.
- c) Sabiendo que los precios de las películas se han ido incrementando un 2% anualmente, elaborar una consulta, **setItemPrice.sql**, que complete la columna 'price' de la tabla 'orderdetail', sabiendo que el precio actual es el de la tabla 'products'.
- d) Una vez se disponga de esta información, realizar un procedimiento almacenado, **setOrderAmount**, que complete las columnas netamount (suma de los precios de las películas del pedido) y totalamount (netamount más impuestos) de la tabla orders cuando éstas no contengan ningún valor. Invocadlo para realizar una carga inicial.

- e) Realizar una función PostgreSQL, **getTopVentas**, que reciba como argumento un año y devuelva para ese año y los siguientes la película que mas se ha vendido

AÑO	PELICULA	VENTAS
1897	El Gran Dictador	12
1912	Superman	4
....		

(Sólo un ejemplo, no son resultados reales)

Incorporar la tabla resultante para los últimos 3 años a la página inicial de bienvenida.

- f) Realizar una función PostgreSQL, **getTopMonths**, que reciba unos umbrales de número de productos y de importe (totalamount) acumulados, y devuelva los meses en los que se ha superado alguno de los dos umbrales. Probadla con umbrales de 19.000 artículos y 320.000 euros.
- g) Realizar un trigger, **updOrders**, que actualice la información de la tabla 'orders' cuando se añada o elimine un artículo al carrito.
- h) Realizar un trigger, **updInventory**, que actualice la tablas 'inventory' y 'orders' cuando se finalice la compra. El trigger también deberá crear una alerta en una nueva tabla de alertas si la cantidad en stock llega a cero. Realizar los cambios necesarios en la base de datos para incluir dicha tabla.
- i) Implementar en el sitio web el registro y la validación de usuario (login) usando la tabla customers
- j) Implementar en el sitio web la funcionalidad de carrito, usando las tablas apropiadas

Trabajo a Realizar:

Como **resultado** de la práctica se entregará:

- Código fuente: Un script, **actualiza.sql**, para realizar los cambios considerados necesarios sobre la base de datos suministrada. Dado su tamaño, dicha base de datos no formará parte de la entrega. Discutir el diseño de la base de datos, ventajas y desventajas de las decisiones implícitas en el mismo.
 - o El código fuente sql requerido.
 - o PHP y auxiliares (CSS, JS, HTML, etc.)
- Memoria en la que se incluirá:
 - o El diagrama entidad-relación y los cambios introducidos, con su justificación
 - o El análisis de la solución dada a las consultas solicitadas.
 - o Las comprobaciones de resultados.

- Cualquier mejora o propuesta de implantación que el alumno considere.

IMPORTANTE: Si es necesario incorporar datos adicionales para completar las tablas del modelo y para hacer pruebas, estos cambios deben ser documentados debidamente en la memoria entregada.

Base de Datos de Trabajo

En esta práctica se suministra el contenido de una Base de Datos (fichero **dump_vx.x.sql.-zip**) con la que se realizarán las pruebas de lo solicitado en la práctica. La base de datos se ha de crear con el nombre **si1**.

Algunas características de esta base de datos son:

- Un pedido en curso (cesta o carrito), se caracteriza por tener un valor NULL en la columna 'status' de la tabla 'orders'
- El pedido pasa sucesivamente por los siguientes valores de 'status': NULL, 'Paid', 'Processed', 'Shipped'.
- La columna 'sales' de la tabla inventory contiene el número acumulado de artículos vendidos de un producto.

Entrega

La fecha y normas de entrega de las prácticas se encuentran en Moodle.

Evaluación

Se valorará muy positivamente cualquier mejora o añadido adicional que aporte el alumno, discusión sobre la eficiencia del método empleado, etc.

Revisad que el formato de las sentencias SQL de consulta se adecúe a las Normas de Prácticas.