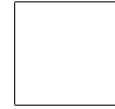


Apellidos:
Nombre: UVUS:



Bases de datos (GE/DGME)
Final oficial - Bloque primer parcial

15/07/2021

Ejercicio 1 (4 puntos)

Queremos construir una base de datos sobre un centro comercial:

- Del centro comercial nos interesa su nombre, dirección, fecha de inauguración y teléfonos de contacto ordenados. El centro dispondrá de plantas, y dentro de cada planta habrá zonas.
- El centro contará con una serie de establecimientos (tiendas, supermercados y restaurantes), con un nombre, un código y un horario.
- Entre los establecimientos destacamos las tiendas, con sus metros cuadrados, zona y tipo. Se almacenarán los clientes de las mismas (con su dni, nombre y apellidos).
- También tenemos supermercados que venden alimentos. Queremos saber su nombre, superficie y zona. De los alimentos el nombre, tipo, calorías, fecha de caducidad y precio base. Naturalmente, el mismo alimento podrá ser provisto por varios supermercados, y cada uno podrá decidir si poner el alimento en oferta.
- Tenemos los restaurantes (con nombre, num. mesas y tipo de comida), que compran alimentos a cada supermercado en una fecha y en cierta cantidad. De los restaurantes queremos saber su nombre, si tiene buffet, el número de mesas.
- Si el centro comercial tiene cine, queremos saber las películas que se proyectan cada día y hora en cada sala. En cada centro comercial solo hay un único cine, si lo hay. Las películas tienen una duración, director y género, junto a sus protagonistas.
- El centro comercial tiene trabajadores, con su DNI, nombre y apellidos, fecha de nacimiento y puesto. Un trabajador puede trabajar en varios centros comerciales. Además, se deberá conocer qué trabajador supervisa a cada uno.
- Al margen de los establecimientos, en el centro habrá instalaciones (con un código de instalación), entidades que hacen que funcione correctamente un centro comercial, cada una de un cierto tipo (como entradas, parkings, sistemas de seguridad, y otros como baños, parques, fuentes, lugares para descansar, etc.)
- De las entradas necesitamos saber de qué tipo es (exterior, de acceso desde el parking, de emergencia, etc). Cada entrada está conectada a uno o varios sistemas de seguridad.
- Cada parking tendrá que almacenar su tipo (subterráneo o exterior), zona y número de plazas de aparcamiento. También queremos controlar los coches que entran y aparcan en dicho parking. Tomaremos de los coches sus matrículas, la marca y la hora a la que accede al parking. También queremos la información de cuándo salen del mismo.
- De los sistemas de seguridad nos interesarán los problemas que detectan, etiquetados bajo un código que hace referencia al inconveniente que hay, la zona donde se sitúa y una descripción. Los problemas los solucionan los trabajadores, pudiendo necesitar varios trabajadores para resolver un problema.

Se pide lo siguiente:

- a) (2,5 puntos) Obtenga el esquema conceptual asociado al problema planteado, mediante diagrama Entidad-Interrelación, de forma que capture los requisitos detallados en los puntos anteriores.
- b) (1,5 puntos) Traslade a diseño lógico el esquema anterior, siguiendo la sistemática que hemos visto en clase.

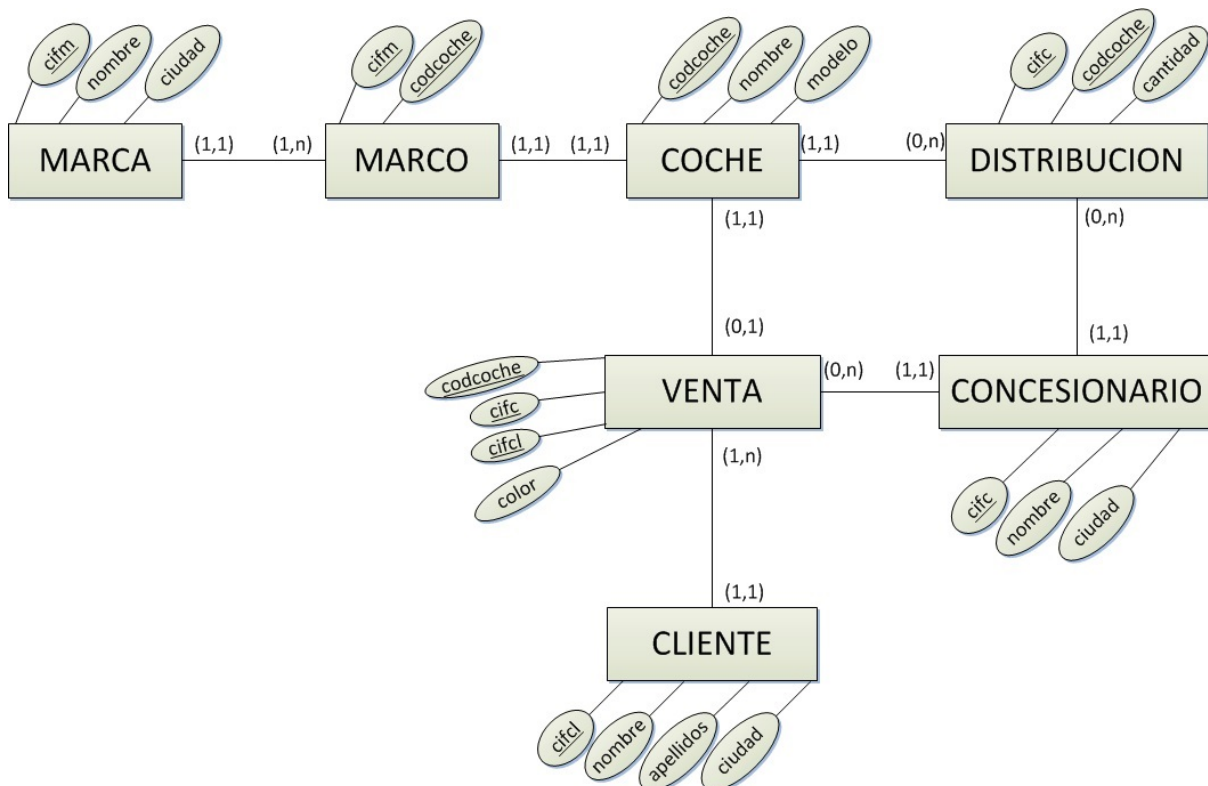
Ejercicio 2 (1 punto)

Conteste razonadamente a las siguientes cuestiones:

- a) ¿Cuáles son a su juicio las principales ventajas de las bases de datos con respecto a otros sistemas de información? Justifique su respuesta.
- b) ¿Qué tipos de redundancia pueden darse en los elementos de un diseño conceptual? Ilustre algunos posibles escenarios a través de ejemplos (no es necesario que contemple toda posible situación).
- c) ¿En qué consiste la integridad referencial? Comente cómo podemos tratar de garantizarla en las distintas acciones que se podrían realizar sobre una base de datos relacional.

Ejercicio 3 (2,5 puntos) SQL

Consideremos una base de datos con las siguientes relaciones, como vimos en clase:



Se pide obtener una sentencia SQL para cada una de las necesidades indicadas:

- a) Creación de una tabla relacional para almacenar la distribución, considerando la información sobre las claves y tomando tipos de datos adecuados (los cifs contienen cadenas de 9 caracteres, los códigos de coches son numéricos, etc.)
- b) Actualización del nombre del concesionario como Reina Mercedes y la ciudad como Sevilla, para el concesionario cuyo cife es A00000000.
- c) Obtener las consultas siguientes:
 - Nombre y color de los coches vendidos ayer.
 - Cantidad total de coches vendidos por cada concesionario.
 - Datos de los concesionarios situados en ciudades que empiecen por la letra S.
 - Pares de código de concesionario y cliente para los que existen más de 5 ventas.
 - Clientes que no han comprado ningún coche de la marca Skoda.

Ejercicio 4 (1,5 puntos) Álgebra Relacional

Sea este esquema relacional, donde **CLIENTE** almacena información sobre clientes, **PRODUCTO** sobre productos a vender y **VENTA** sobre ventas, con **Cod-Producto** indicando el producto vendido y **CodCliente** el cliente al que vendimos el producto.

CLIENTE(Num-Cliente, Nombre, Apellidos, Dir, Tfno, Pobl)

PRODUCTO(Cod-Producto, NombreProd, Desc, Precio, LocalidadFabricacion)

VENTA(Id-Venta, CodCliente, Cod-Producto, Cantidad, Fecha)

Se pide emplear Álgebra Relacional para responder a las operaciones de recuperación:

- a) Devolver la descripción y el precio del producto fabricado en Cáceres y cuyo nombre es "Mascarilla FFP2".
- b) Obtener el nombre de producto y la cantidad, para las ventas realizadas a clientes de Palencia.
- c) Mostrar el código de los clientes que han comprado todos los productos fabricados en Tarragona.
- d) Nombre y apellidos de los clientes que no han hecho ninguna compra cuya cantidad sea mayor a 10.

Ejercicio 5 (1 punto)

Sea $R = (AT, DEP)$, con $AT = \{A, B, C, D, E, F, G, H, I\}$ y

$DEP = \{ADE \rightarrow F, AF \rightarrow B, E \rightarrow ACD, C \rightarrow D, BF \rightarrow CD, EF \rightarrow B, EG \rightarrow I, G \rightarrow H\}$

Se pide responder razonadamente a las siguientes cuestiones:

1. Hallar un recubrimiento minimal a partir del conjunto de dependencias funcionales de partida, siguiendo el procedimiento sistemático que hemos visto en clase, y apoyándose en cualquier paso previo que pueda necesitar (cierres transitivos, claves candidatas, etc.)
2. Determine el grado de normalización de R .

Notas:

- Escriba su nombre, apellidos y UVUS en la hoja del examen.
- Cumplimente los ejercicios en los folios proporcionados en el aula.

Apellidos:
Nombre: UVUS:



Bases de datos (GE/DGME)

15/07/2021

Final oficial - Bloque segundo parcial - Primera parte

Ejercicio 1 (1 punto) Responda a las siguientes cuestiones acerca del desarrollo web:

- a) (0,5) Indique y describa brevemente los programas y lenguajes que hemos empleado para el desarrollo web. Relacione estos elementos con conceptos como edición, pruebas, contenido, estilo, dinámica en cliente y dinámica en servidor.
- b) (0,5) Ilustre gráficamente y explique el proceso que ocurre desde que escribimos una dirección web en un navegador hasta que visualizamos la respuesta renderizada, incluyendo información de nuestra base de datos.

Ejercicio 2 (2,5 puntos) Desarrolle el código HTML correspondiente a:

- a) (0,5) Una lista no ordenada con sus deportes favoritos, incluyendo para cada uno de ellos una lista que ordene sus deportistas favoritos para cada deporte, incluyendo un enlace a sus webs.
- b) (0,75) Una tabla con sus bloques principales y varias filas en las distintas secciones, usando las etiquetas adecuadas para resaltar lo que necesite.
- c) (0,75) Un formulario con controles básicos usando *distintos tipos de etiquetas* para varios controles de entrada, incluyendo algunos propios de HTML 5 (no versiones previas). Incorpore atributos interesantes, además de los obligatorios.
- d) (0,5) Un fragmento de página con elementos puramente semánticos como los vistos en clase.

Ejercicio 3 (1 punto) Indique la expresión Emmet necesaria para generar:

- a) Un elemento de bloque con un identificador “miContenedor”, que contenga tres párrafos, cada uno de ellos 25 palabras generadas automáticamente.
- b) Una lista ordenada de clase “miLista” con 5 elementos, cada uno con un id secuencial, y conteniendo un enlace a una página también de nombre secuencial, con el texto del enlace componiéndose de 3 palabras generadas aleatoriamente.

Ejercicio 4 (2,5 puntos) Desarrolle el código CSS necesario para lo siguiente:

- a) (0,5) Aplicar a los párrafos hijos directos de un `div` con id “mi_div” un espaciado interno de 20 píxels. Asignarles la mitad del ancho, centrados horizontalmente.
- b) (0,5) Dejar a todos los tipos de celdas de toda tabla, al pasar sobre ellos, un espaciado interior con una medida que dependa del tamaño de la fuente y un color de fondo.
- c) (0,5) Aplique un estilo al `div` con id “barra-fija”, de modo que quede posicionado en la parte superior de la pantalla, separada 2 píxels de la parte superior y 3 del lado izquierdo, independientemente de dónde lo hayamos incluido en nuestro código html, pero de forma que al hacer scroll hacia abajo suficientemente dejemos de verlo.
- d) (0,5) Aplicar a todo elemento con la clase “divide” un grid que tenga 2 columnas de la misma anchura y 3 filas de la misma altura, ajustable al tamaño de la página. Aplicar una pequeña separación entre las filas y columnas del grid.

- e) (0,5) Emplear a todo elemento que tenga la clase “relaja” un diseño flexible a nivel de columna, de modo que todos los elementos incluidos queden dispuestos equitativamente. Además, aplique estilo a uno de esos elementos hijos de modo que, independientemente de su posición en el código, este elemento aparezca el último.

Ejercicio 5 (2 puntos) Haga uso de Bootstrap para crear la estructura necesaria para:

- a) (1) Crear un elemento de navegación (pestañas, pills o menús) con 2 opciones, cada una de ellas habilitando una sección. Cree los elementos origen de navegación y código mínimo del destino, marcando adecuadamente la opción por defecto.
- b) (1) Generar un formulario Bootstrap con dos campos, y dos botones de distinto tipo, usando las clases adecuadas. Para dispositivos de tamaño pequeño o superior debemos garantizar que aparezca un campo por fila, incluyendo etiqueta y control, debidamente alineados, quedando el texto de las etiquetas a la misma distancia del comienzo del control de formulario para todas las filas.

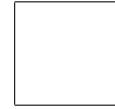
Ejercicio 6 (1 punto) Emplee código SCSS visto en clase para lo siguiente:

- a) (0,25) Declarar variables SASS para establecer algunos datos por defecto, y emplearlos en alguna regla de estilo básica.
- b) (0,375) Usando el mecanismo de anidamiento visto, proporcione estilo a ciertos elementos jerárquicamente organizados con varios niveles de profundidad. Puede usar ejemplos de html de ejercicios anteriores. Debe incluir selectores para elementos hijos, descendientes, y el operador para referirnos al propio elemento dentro del anidamiento, añadiéndole además una pseudo-clase.
- c) (0,375) Definir un `mixin` con parámetros, valores declarados por defecto, y asignando varias propiedades de estilo. Utilice este `mixin` en un par de reglas de estilo, en una pasando datos específicos y en la otra dejando valores por defecto.

Notas:

- Escriba su nombre, apellidos y UVUS en la hoja del examen.
- Cumplimente los ejercicios en los folios proporcionados en el aula.

Apellidos:
Nombre: UVUS:



Bases de datos (GE/DGME)

15/07/2021

Final oficial - Bloque segundo parcial - Segunda parte

Ejercicio 1 (0,75 puntos) Responda a las siguientes cuestiones:

- a) (0,5 pts) Indique y describa brevemente los programas que hemos empleado para:
a) editar código; b) servir archivos; c) lanzar peticiones a servidor; d) ejecutar programas en servidor; e) servir bases de datos; f) actuar como cliente de estas; g) visualizar sitios; h) testear aspectos del funcionamiento en web.
- b) (0,25 pts) ¿Qué protocolo hemos visto para hacer peticiones web y recibir resultados? ¿De qué dos formas principales podemos transmitir en este protocolo los datos entre cliente y servidor? ¿Qué método suele emplearse para cada forma?

Ejercicio 2 (3 puntos) Desarrolle el código HTML y JavaScript necesario para:

- a) (1 pts) Terminada de cargar la página, deberá pedir en mensaje emergente al usuario un texto a repetir, y un segundo mensaje donde indique el número de veces a repetir. A continuación, deberá imprimir el mensaje por consola tantas veces como se haya indicado. Modifique el código para mostrar el mensaje en un elemento de bloque del DOM con un cierto id, generando un párrafo por cada repetición.
- b) (1 pts) Crear una función que reciba un array, con objetos del tipo `[{nombre: "Luis", ape: "Pérez"}, {nombre: "Elena", ape: "López"}]`, y devuelva otro array con la suma de caracteres (ej: `[9, 10]`). Asociar a un botón un manejador de evento en cuyo cuerpo se llame a esta función, pasándole el array del ejemplo, y lo asocie a una variable/constante. Mostrar un mensaje emergente con un texto que indique cuál es la suma de todos los números del array devuelto.
- c) (1 pts) Crear una función que reciba un array `a` (del tipo del anterior) y un entero `ncar`, y devuelva un array con los objetos de `a` cuyo nombre tenga menos de `ncar` caracteres. Definir fuera de la función un array de ejemplo. Suponiendo que existe un elemento `<select>` con números, crear un manejador de evento tal que al cambiar la selección, se llame a la función anterior, pasándole el array definido y el valor del elemento seleccionado.

Ejercicio 3 (3,25 puntos) Desarrolle el código PHP necesario para:

- a) (1,5 pts) Crear una biblioteca con las siguientes funciones:
 - Una función que reciba como parámetros un nombre de base de datos, un usuario y un password, construya la cadena de conexión para acceder a la misma y se conecte como hemos visto en clase, pasándole url, usuario y password. Devuelva la conexión creada.
 - Una función que reciba dos parámetros (una conexión a BD y un valor numérico), y que lance una consulta preparada que, haciendo uso de la conexión pasada, emplee el valor recibido para filtrar los datos del SELECT sobre una tabla de su base de datos, obteniendo los registros cuyo valor para un cierto campo superen el valor numérico dado por el parámetro. La función debe devolver a quien la llame las filas devueltas por la consulta, sin más procesamiento.

- Una función que reciba tres parámetros (conexión a BD, id y nombre), y que lance una inserción preparada, que inserte en los dos campos correspondientes de una tabla un nuevo registro con los datos recibidos. La función deberá devolver un array asociativo con un campo “mensaje” con un valor “Ok”.
- b) (1,75 pts) Crear una página que reciba peticiones GET/POST del cliente y responda según lo recibido, usando las funciones de la biblioteca anterior, y devolviendo una salida JSON en todos los casos:
- En primer lugar, intentar recibir un dato “tipoAccion”, con posibles valores *consulta* o *inserta*. Si el tipo recibido no es ninguno de los dos, debe responder con un objeto con un campo “mensaje”, indicando que no es válido. En otro caso, comprobar en la petición si se recibieron **bd**, **user** y **pass**. Si no es así, se deben asignar los valores por defecto “prueba”, “root” y “”, respectivamente. Con estos datos, establecer la conexión con la BD, usando la función creada.
 - Si la acción es *consulta*, comprobar si se ha recibido **num**, y en su caso emplearlo para lanzar la consulta desarrollada del ejercicio anterior. Además, enviar a la salida como JSON un array asociativo con un campo *mensaje* que indique “Ok” y otro *resultado* que incluya la respuesta de la función anterior.
 - Si la acción es *inserta*, debe comprobarse que se han proporcionado en la petición los datos **id** y **nombre**, y a continuación llamarse a la función desarrollada. Se deberá finalmente emitir a la salida un objeto JSON con el contenido que se haya recibido de la llamada a la función.

Ejercicio 4 (3 puntos) Desarrolle el código HTML y Javascript necesario para responder a los siguientes apartados:

- a) (0,5 pts) Crear un formulario con un select con dos opciones con valores **bd1** y **bd2** y nombres inventados, con dos campos de texto para usuario y password, un control de número “val”, otro “id” y un control para proporcionar en “nombre”.
- b) (1,5 pts) Definir una función que lance con Javascript una petición GET al programa de servidor creado, solicitando la acción de tipo *consulta*, pasándole los datos que se necesiten para la misma. Al recibir el resultado del servidor, deberá comprobarse si el resultado fue “Ok”, y en ese caso deberá generar en la misma página un elemento de tipo lista que visualice un elemento por cada registro recibido del servidor, mostrando el valor de algunos campos. Asociar dicha función como manejador de evento tanto al cambiar el select como al modificar el valor del control “val”.
- c) (1 pto) Incorporar al formulario anterior un botón tal que, al hacer click sobre el mismo, lance desde Javascript una petición de tipo POST al servidor, solicitando la acción de tipo *inserta*, pasándole los valores de los elementos necesarios. Al recibir la respuesta del servidor, bastará con emitir un mensaje emergente que diga “¡Excelente!” en caso de que el mensaje contenido en el resultado de la operación haya sido “Ok”, y otro mensaje “¡Algo ha ido mal!” en caso contrario.

Notas:

- Escriba su nombre, apellidos y UVUS en la hoja del examen.
- Cumplimente los ejercicios en los folios proporcionados en el aula.
- Se valorará la variedad de técnicas empleadas, como se planteó en clase.