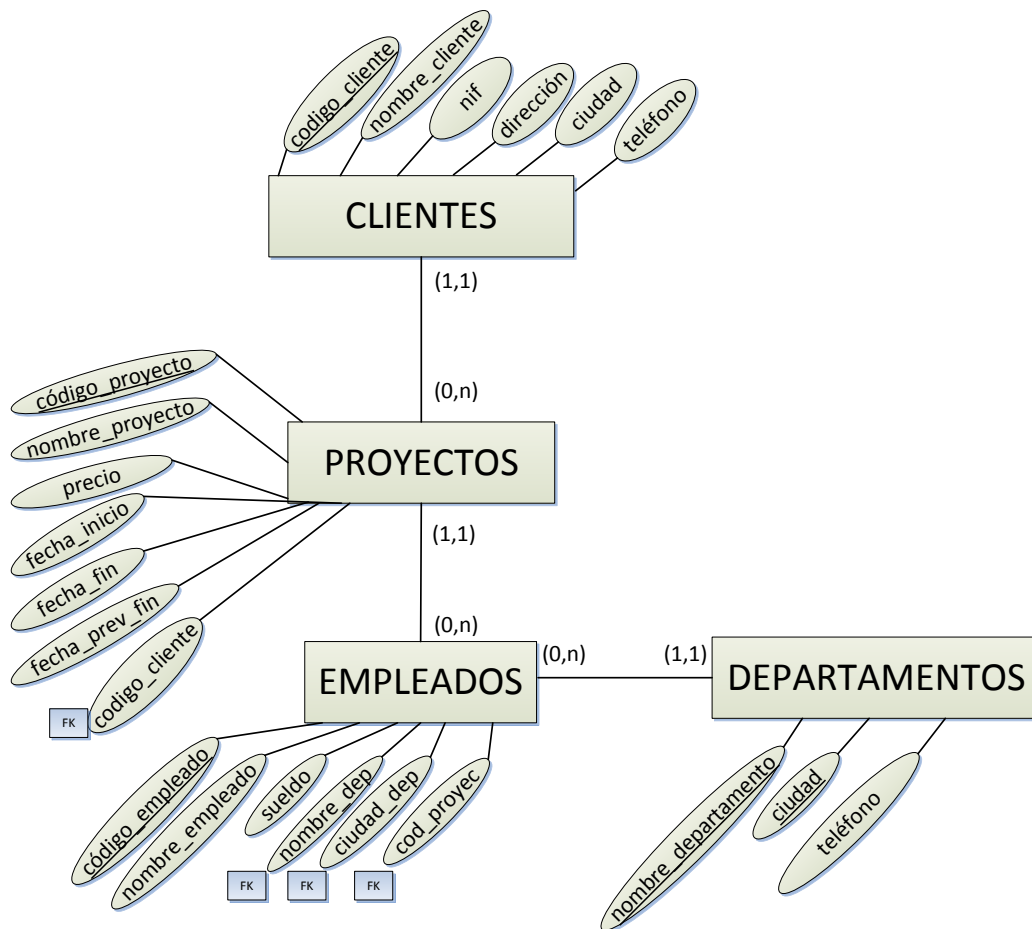


## **Primer cuatrimestre**

### **Se desea diseñar una base de datos para almacenar información acerca de fórmula 1. Se dispone inicialmente de los siguientes requisitos:**

- Se debe almacenar información sobre las escuderías que existen o han existido en fórmula 1, incluyendo su nombre identificativo, año de debut, país y url de su web. (Ejemplo: Ferrari, Red Bull, etc.)
- Se deben almacenar además fotos asociadas a la escudería, indicando la posición de la foto dentro de la escudería por si queremos mostrarlas ordenadas en una web. La información deberá incluir el año de la foto y su fichero.
- Se deben almacenar los empleados (no pilotos) de cada escudería, identificados por su dni, incluyendo nombre y apellidos y fecha de inicio de contrato.
- Se deberá conocer si el empleado es personal técnico (ingenieros y ayudantes), de los que se almacenará su cargo y nivel de estudios, o si es personal administrativo.
- Además, se deberá conocer para cada empleado quién es su jefe, pudiendo éste ser jefe de varios empleados pero pudiendo tener éstos como máximo un jefe.
- Se debe almacenar información sobre pilotos de fórmula 1 (en general), almacenando su nombre y apellidos, nombre artístico (único), fecha de nacimiento, país, localidad, y url de su web si la tiene.
- Se deben almacenar noticias asociadas a pilotos, identificadas por su fecha y titular, y acompañadas del cuerpo de la noticia.
- Se deben almacenar las temporadas, identificadas por su año (ejemplo: 2014), e incluyendo la fecha del primer gran premio y la fecha de fin.
- Se debe almacenar la información sobre los participantes, es decir, la participación de cada piloto en cada temporada en cada escudería. Además de estos datos, deberá almacenarse el rol del piloto ese año en esa escudería (piloto principal o probador). Por ejemplo, un piloto puede ser participante como probador en una escudería en 2013 y pasar a titular en 2014.
- Se deberá almacenar información sobre las sedes de GP. Además de su nombre identificativo, se deberá almacenar el país, la ciudad, el año de debut del GP y el último año en que ha sido sede.
- Se deberán almacenar los grandes premios (GP), identificados por la sede y la temporada. Un gran premio tiene una sede (por ejemplo, China, Valencia o Mónaco), y una misma sede puede albergar varios grandes premios (por ejemplo, GP China 2013, GP China 2014, GP Valencia 2011). Además de la sede y la temporada, se deberá almacenar la extensión del circuito y el número de vueltas, de modo que se pueda conocer la distancia total a recorrer en la carrera.
- Se deberá conocer qué participante corre en cada GP, indicando la posición que alcance finalmente en el mismo y el tiempo invertido.
- Se deben almacenar las tandas de clasificación, indicando el identificador de la tanda (Q1, Q2 o Q3), la descripción de la tanda y el número de pilotos que pueden tomar parte en la misma.
- Se debe conocer para cada participante, GP y tanda, la posición que participación del participante en dicha tanda de dicho GP, junto con la posición ocupada y el tiempo invertido en su mejor vuelta.

1. (3,75 puntos) Construya un diagrama Entidad-Relación que se ajuste a la descripción de requisitos anterior.
2. (2,5 puntos) A partir del DER propuesto, construye el esquema relacional asociado, explicando las decisiones tomadas en caso de que tenga varios posibles enfoques para resolver lo mismo. No olvide indicar las claves primarias y ajenas.
3. (2,5 puntos) Dada una base de datos sobre sobre clientes, proyectos, departamentos y empleados (datos en pág. siguiente):



**Genere el código SQL correspondiente a las siguientes instrucciones:**

- Creación de una tabla relacional para almacenar los departamentos.
- Creación de una tabla relacional para almacenar los empleados, suponiendo que las tres restantes han sido ya creadas.
- Modificación de la tabla PROYECTOS para modificar el tipo del campo precio de entero a real.
- Actualización a la ciudad Sevilla de los clientes que tengan como ciudad Sevilla.
- Eliminación de los departamentos de Córdoba.
- Inserción de un departamento CCIA de Sevilla, de teléfono 954557947.

- Selección de:
  - a. Información sobre los clientes de Sevilla con teléfono 954999999.
  - b. Código y nombre de los proyectos de Sevilla con fecha de inicio hoy o mañana.
  - c. Teléfono de los departamentos cuyo nombre comience por L.
  - d. Media de sueldo de los empleados por ciudad de su departamento.
  - e. Número de proyectos con precio mayor que 100.000 por cliente, siempre que el número de proyectos sea al menos 3.
  - f. Código y nombre de proyecto, para los proyectos de clientes de Sevilla.
  - g. Ciudad del cliente, nombre del proyecto, nombre del empleado, sueldo y teléfono del departamento, para todos los empleados.
  - h. Datos de los empleados cuyo proyecto tiene un precio mayor a 2.000.000.
  - i. Datos de los departamentos que no tienen ningún empleado asociado.

**4. (1,25 puntos) El diseño de una base de datos relacional se compone de la siguiente relación:**

$$T = \{A, B, C, D, E, F\}$$

$$L = \{E \rightarrow C; AC \rightarrow D; D \rightarrow A; A \rightarrow BF; CD \rightarrow E\}$$

**Se pide:**

**a) Determinar el nivel de normalización, respondiendo a las siguientes cuestiones previamente:**

- Describir un conjunto de dependencias funcionales elementales mínimo.
- ¿Hay conjuntos de campos equivalentes? ¿Cuáles son las claves candidatas?
- ¿Cuáles son los campos principales (P) y no principales (Q) de la tabla? ¿Hay dependencias transitivas?

**b) Normalizar hasta 3FN, describiendo el proceso realizado.**

**c) (Opcional) Dibujar los esquemas conceptual y lógico correspondientes.**

## Segundo cuatrimestre

## Ejercicio 1 (3,5 puntos):

- (0,75 pts) Describa los siguientes procesos (puede emplear términos como ip, dns, http, html, css, js, jQuery, ajax, php, mysql):
  - a. El usuario teclea en el navegador la página “http://www.servidor.com/hola.html” y se muestra el resultado en la web.
  - b. El usuario teclea “http://www.unservidor.com/cuenta.php?nombre=Pepe” y se muestra una lista de movimientos en la cuenta de Pepe.
  - c. El usuario hace click en un botón de tipo button de un formulario y se muestra en la misma pantalla una tabla.
- (2,25 pts) Escriba el código HTML asociado a los siguientes elementos (se evaluará como indica la puntuación de cada apartado):
  - a. (0,5) Esqueleto de una página HTML5 mínima incluyendo elemento base para rutas relativas, y conteniendo una cabecera de tercer nivel alineada a la derecha con el texto “Comienza lo bueno”, un párrafo con un elemento destacado con el texto “claro que sí” y un elemento con una fuente de tamaño inferior al actual en una unidad indicando “¿o no?”.
  - b. (0,5) Un texto incorporando algunos de los elementos siguientes: inserciones o borrados, acrónimos, abreviaturas, citas, definiciones, texto literal respetando el formato tal como lo escribimos, elementos de línea con su texto, saltos de línea, líneas horizontales, caracteres codificados (para espacio en blanco, tilde, etc.), código de algún lenguaje de programación, ampliar detalles. Indique un mínimo de 4.
  - c. (0,25) Una lista no ordenada de imágenes actuando como enlaces a páginas de Internet (por ejemplo, con iconos de facebook, twitter y Google+).
  - d. (0,5) Una tabla con su título “Libros más vendidos”, con borde de tamaño 2, anchura interior de celda de 4, secciones de cabecera y cuerpo de la tabla, quedando de la forma siguiente:

Libros más vendidos	
Libros	
Libro	Precio
Las gafas de la felicidad	15,90 €
Las tres bodas de Manolita	22,90 €
Divergente	19,00 €

- e. (0,75) Un formulario indicando acción y método de envío, conteniendo etiqueta, control de texto, email, fecha/hora, radio, selección mediante desplegable incluyendo grupos de opciones, y botón de envío de formulario, en ese orden. Opcional: reemplazar el select por otro mecanismo similar mediante input (nuevo de html5).
- (0,5 pts) Dibuje el árbol del DOM del apartado e anterior, incorporando además de las etiquetas: los tipos de controles, botones y los nombres.

## Ejercicio 2 (2 puntos) CSS. Aplique los siguientes estilos mediante CSS. [Incluya el código mediante alguna de las tres formas indicadas en clase]:

- Quite el subrayado que suele aparecer por defecto en los hiperenlaces de nuestras páginas.
- En la misma instrucción CSS aplique algún color de la escala de grises a los elementos de las clases *destacado* e *importante*, y establezca su tamaño al doble del tamaño de la letra M.
- Ponga un borde de línea continua de 2 píxels azul al elemento identificado como *soyyo*.
- Ponga un color de fondo amarillo a cada control de texto contenido en un formulario al tomar el foco.
- Cambie el color a rojo de los enlaces hijos directos de elementos de lista en el momento de pasar por encima de ellos.
- Establezca un margen izquierdo en píxels y un ancho a los párrafos hermanos de una cabecera de primer nivel.
- Cambie el color del segundo hijo de tipo td de una fila de tabla.
- Establezca un relleno porcentual a todo elemento que sea último hijo de una lista ordenada cuyo tipo termina en e.

## Ejercicio 3. Bootstrap (1 punto):

- ¿Qué debemos incluir en el código html de nuestra página para comenzar a trabajar con Bootstrap y poder así aplicar las clases correspondientes?
- Cree un formulario de tipo horizontal ocupando las 10 columnas centrales de una fila n un contenedor (es decir, desplazándose 1 columna desde la izquierda). Dentro del mismo, coloque algunos grupos conteniendo etiquetas ocupando 4 columnas y controles ocupando 8.
- Incluya un botón por defecto para reset, uno primario para submit y uno de aviso para validar mediante una acción JavaScript.
- ¿Qué cambio debemos hacer a la tabla definida en el ejercicio 1 para que se presente con el estilo de Bootstrap? ¿Y para que aparezca con las filas de colores alternos?
- Ponga algún ejemplo de código html empleando Bootstrap para generar contenido de navegación, pestañas, píldoras, menús de navegación, grupos de botones, listas de elementos, mensajes de alerta, paneles, pozos, desplegados, barras de progreso o algunos otros elementos no contemplados en otros apartados y que demuestren su empleo de Bootstrap.

#### Ejercicio 4. JavaScript (1,5 puntos):

- Escriba una función que se ejecute al cargarse la página y que muestre al usuario el mensaje emergente “¡Página cargada!”. Llame a la función desde un botón al hacer click.
- Escriba una función que reciba un número, compruebe si es par; si lo es escriba en el documento html un párrafo indicando que el número es par, seguido de un texto indicando el resultado de dividirlo por 2.
- Genere el código JavaScript que haga lo siguiente:
  - a. Asigne a una variable el elemento con id “marcado” (supongamos que se trata de un div de nuestra página).
  - b. Cree un elemento de texto que diga “Texto añadido”.
  - c. Cree un elemento de tipo párrafo.
  - d. Asigne el texto al párrafo.
  - e. Asigne al atributo de alineación del párrafo el valor necesario para que aparezca centrado.
  - f. Añada el párrafo al elemento con id “marcado” del apartado a.

#### Ejercicio 5. jQuery (1 punto):

- Escribir el código necesario para efectuar lo siguiente cuando se haya cargado la página empleando jQuery: mostrar en un mensaje emergente el texto “Buenas noches”.
- Disponer de una función llamada fibonacci que reciba un número n y devuelva el término enésimo de la sucesión de Fibonacci (si n vale 0 o 1, devolverá 1, y en caso contrario la suma de los dos términos anteriores). Llamar a fibonacci(8) cuando se haga click en el botón con id “calcularTerminoN”.
- Acceder mediante jQuery a elementos del DOM por id, por nombre de clase y por nombre de etiqueta.

#### Ejercicio 6. PHP (1 punto):

- Escriba una página PHP completa que reciba tres parámetros a, b y c de una petición que admita tanto GET como POST, compruebe cuál es el mayor, y devuelva como respuesta en una nueva página un control de área de texto con la respuesta, de forma que podamos visualizarlo en nuestro navegador.
- Escriba el código PHP para hacer lo siguiente:
  - Conectarse a la base de datos ‘docencia’.
  - Preparar una consulta en una nueva variable, incluyendo el código necesario para recuperar todos los datos de la tabla Exámenes cuyo aula (un texto) sea la que hemos recibido y cuyo curso (un número) sea el que hemos recibido.
  - Ejecutar una consulta para recuperar los datos de la tabla Exámenes.
  - Recorrer los resultados, y devolver en la salida en forma de tabla con los datos fecha, hora, aula y asignatura.
  - Cerrar la conexión con la base de datos.

`mysql_affected_rows` — Obtiene el número de filas afectadas en la anterior operación de MySQL

`mysql_close` — Cierra una conexión de MySQL

`mysql_connect` — Abre una conexión al servidor MySQL. Requiere servidor, usuario y password

`mysql_create_db` — Crea una base de datos MySQL

`mysql_db_name` — Recupera el nombre de la base de datos desde una llamada a `mysql_list_dbs`

`mysql_db_query` — Selecciona una base de datos y ejecuta una consulta sobre la misma

`mysql_fetch_array` — Recupera una fila de resultados como un array asociativo, un array numérico o como ambos

`mysql_fetch_assoc` — Recupera una fila de resultados como un array asociativo

`mysql_field_len` — Devuelve la longitud del campo especificado

`mysql_field_name` — Obtiene el nombre del campo especificado de un resultado

`mysql_field_type` — Obtiene el tipo del campo especificado de un resultado

`mysql_free_result` — Libera la memoria del resultado

`mysql_num_fields` — Obtiene el número de campos de un resultado

`mysql_num_rows` — Obtener el número de filas de un conjunto de resultados

`mysql_pconnect` — Abre una conexión persistente a un servidor MySQL

`mysql_ping` — Efectuar un chequeo de respuesta (ping) sobre una conexión al servidor o reconectarse si no hay conexión

`mysql_query` — Enviar una consulta MySQL

`mysql_real_escape_string` — Escapa caracteres especiales en un string para su uso en una sentencia SQL

`mysql_result` — Obtener datos de resultado

`mysql_select_db` — Seleccionar una base de datos MySQL, pasándole como parámetros el sistema de bases de datos al que nos conectamos y la base de datos objetivo

`mysql_set_charset` — Establece el conjunto de caracteres del cliente

`mysql_stat` — Obtiene el estado actual del sistema

`mysql_tablename` — Obtiene el nombre de la tabla de un campo