

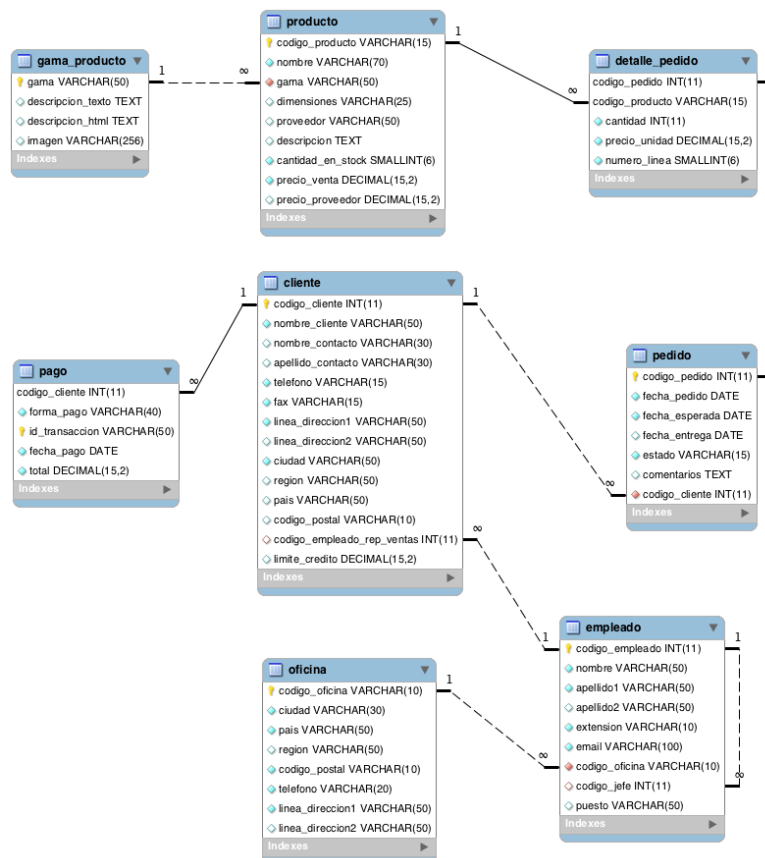
Apellidos: .....  
Nombre: ..... UVUS: .....



Bases de datos (GE/DGME)  
Segunda prueba de evaluación alternativa

19/12/2022

Ejercicio 1 (4 puntos) Considérese el siguiente esquema lógico<sup>1</sup>



Se pide lo siguiente:

- (3 puntos) Resolver las consultas siguientes utilizando el lenguaje de consultas SQL. Para ello, es necesario disponer previamente de la base de datos cargada en MySQL/MariaDB, para lo cual dispone del script `jardineria.sql`, que crea la base de datos, las tablas y los registros necesarios de la misma (no se crea la tabla `PAGO`). Una vez desplegada, resuelva las siguientes consultas:
  - (0,3 ptos) Listado con la ciudad y el teléfono de las oficinas de España, ordenadas por ciudad ascendente.
  - (0,3 ptos) Listado de países de los que se tienen clientes cuyo nombre termine en 'a' (puedes usar `LIKE` o `RLIKE`).
  - (0,4 ptos) Códigos de los empleados de la extensión 2442 junto con el nombre y primer apellido de sus jefes (en un solo campo separado por un guión, `jefe`). Puede usar la función `CONCAT (string1, string2,...)`.
  - (0,4 ptos) Empleados que no son jefes de nadie, junto con el teléfono de la oficina a la que pertenecen.

<sup>1</sup>Agradecimientos a José Juan Sánchez, a quien debemos [esta BD](#) y [algunos ejercicios propuestos](#), y a Iván López Montalbán, cuyo libro [Bases de datos](#) proporciona la fuente original de la misma.

- (v) (0,4 ptos) Oficinas donde no trabaja ningún empleado que haya sido representante de ventas de algún cliente de alguna ciudad que empiece por 'M'.
  - (vi) (0,3 ptos) Para cada país, devolver su nombre y número de clientes asociado.
  - (vii) (0,4 ptos) Productos cuya cantidad media vendida por pedido supera a 100.
  - (viii) (0,5 ptos) Listado de los 3 productos más populares (comprados por más clientes distintos) y n<sup>o</sup> total de unidades que se han vendido de cada uno. Deberá estar ordenado primero por el número de clientes (de mayor a menor) y después por el número total de unidades vendidas (de menor a mayor).
- b) (1 punto) Completar la base de datos, mostrando el código SQL necesario para la realización de las siguientes acciones (0,25 puntos/acción):
- (i) Crear la tabla **pago**.
  - (ii) Incluir en la tabla anterior la restricción de que el total del pago debe ser inferior a 1000. ¿Sería posible establecer una restricción que imponga que la fecha de pago debe ser posterior a la fecha actual (`CURDATE`), empleando `CONSTRAINT`?
  - (iii) Insertar un registro en la tabla, para el cliente 9, con forma de pago **Paypal**, id de transacción **ak-std-000013**, fecha **2009-01-06** y total 929. ¿Qué ocurre si ponemos 1100 en lugar de 929? Razone su respuesta.
  - (iv) ¿Cuál sería la expresión SQL para eliminar todos los clientes de la ciudad "Madrid"? ¿Su ejecución ocasionaría algún error? ¿A qué se debe?.

**Ejercicio 2 (2 puntos)** Sea el esquema de relación  $R(AT, DEP)$ , cuyos campos son todos monovaluados, y tal que:

$$\begin{aligned}
 AT &= \{A, B, C, D, E, F, G, H, I, J, K\}, \\
 DEP &= \{A \rightarrow F, AD \rightarrow G, D \rightarrow ACFB, FG \rightarrow H, E \rightarrow IJ, J \rightarrow K, C \rightarrow DA\}
 \end{aligned}$$

**Se pide** lo siguiente (0,5 puntos/apartado):

- a) Obtener paso a paso el conjunto de dependencias funcionales mínimo (recubrimiento minimal), como se ha visto en clase, y el cierre transitivo de cada campo individual.
- b) Obtener las claves candidatas, **indicando el razonamiento seguido** (recuerde, puede apoyarse en el cálculo de los conjuntos de *imprescindibles*, *descartados* y *resto* de campos). Indicar los conjuntos P y Q (campos principales y no principales), y si hay campos o conjuntos de campos equivalentes.
- c) Determinar el grado de normalización de la relación, justificando el cumplimiento o no de 1FN, 2FN y 3FN, en base a la naturaleza de los campos y la presencia de dependencias funcionales no plenas o de dependencias transitivas.
- d) Normalizar hasta 3FN, aplicando análisis o síntesis, descomponiendo en un conjunto de relaciones en 3FN y preservando las dependencias originales.

**Ejercicio 3 (4 puntos)** **Se pide** lo siguiente en relación con MongoDB<sup>2</sup>:

- a) (1 punto) Realizar las siguientes operaciones de consulta básicas (0,25/consulta):
  - (i) Datos de los libros cuyo **status** sea **PUBLISH**, ordenados por título.
  - (ii) Distintos valores de año que tenemos en nuestros libros de más de 500 páginas.

---

<sup>2</sup>Los siguientes apartados emplean la BD que puede importar desde `books.json`, cortesía de Hakan Özler en su repositorio público [mongodb-json-files](#).

- (iii) Número de libros que sean del 2001 o tengan la categoría PHP.
  - (iv) Título, año y número de páginas del libro cuya descripción corta comienza por MongoDB, sin distinguir mayúsculas de minúsculas.
- b) (1 punto) Realizar las siguientes operaciones de manipulación de datos (0,25/op.):
- (i) Actualizar el libro **Specification by Example** para que su imagen (campo `thumbnailUrl`) tome el valor `gojkoAdzic.jpg`.
  - (ii) Añadir al libro **Flex 3 in Action** la categoría **CSS**, sin incluir repetidos.
  - (iii) Incorporar un nuevo libro que le guste, con los datos que conozca del mismo. Identifíquelo con el valor 1000.
  - (iv) Para todo libro en que aparezca en sus categorías el valor **Object-Oriented Programming**, sustituir sus apariciones en el array por **OOP**.
- c) (1 punto) Realizar las siguientes operaciones empleando el *pipeline* de agregaciones (0,25/operación):
- (i) Número de libros de cada año, mostrando primero los más antiguos.
  - (ii) Número total de páginas y promedio, para cada año e inicial. Limite su resultado a los 5 años con mayor promedio de páginas.
  - (iii) Restringiendo a los años 2010 al 2012 y las iniciales A, B y C, calcular para cada valor de año e inicial, el título y el número de páginas del libro con mayor número de páginas. Finalmente, ordénelo descendientemente por año, y ascendientemente por inicial dentro del año.
  - (iv) Número de libros disponible para cada categoría, ordenados por número de libros. Ayuda: puesto que cada libro puede tener varias categorías, recuerde el uso del operador *unwind*.
- d) (1 punto) Usando los principios de diseño vistos, **plantear** las colecciones para el problema planteado a continuación, **esbozando con documentos** de ejemplo:
- *Existen líneas de metro, cada una con un  $n^o$  identificador y un nombre. Una línea pasa por una serie de estaciones en un orden determinado. Es muy importante recoger la información sobre este orden.*
  - *Hay estaciones de metro, con nombre único. Cada estación pertenece al menos a una línea, pudiendo pertenecer a varias. A su vez cada estación puede tener varios accesos (con un nombre y una calle a la que dan acceso), pero un acceso sólo puede pertenecer a una única estación.*
  - *Hay dos tipos de estaciones: simples y mixtas. Las primeras sólo tiene servicio de metro, mientras que las segundas ofrecen conexión con otros servicios (estación de tren, estación de autobuses, aeropuerto,...). Para las estaciones mixtas interesa conocer con qué otros servicios ofrece conexión.*
  - *Cada tren tiene un código interno que lo identifica. Asimismo interesa conocer el modelo del tren, el número de vagones, la antigüedad, el color, la empresa y la capacidad. Un tren está asignado a una línea, pero puede ir cambiando la línea a la que está asignada.*