

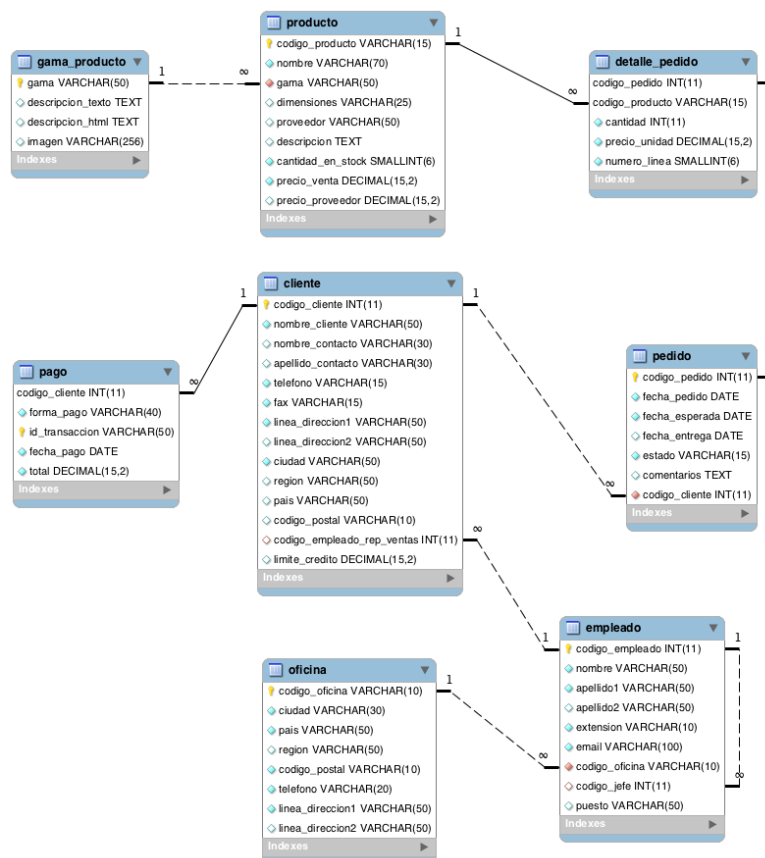
Apellidos: .....  
 Nombre: ..... UVUS: .....



Bases de datos (GE/DGME)  
 Primer parcial oficial - Parte 2

10/01/2023

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:
  - (0,5 ptos) Listado con los países de los que se tienen clientes, ordenados alfabéticamente.
  - (0,6 ptos) Listado con el código de pedido y ciudad cuya fecha de entrega ha sido al menos dos días antes de la fecha esperada. Puede usar la función `DATEDIFF(expr1, expr2)` devuelve el número de días entre las fechas.
  - (0,6 ptos) Empleados que no son jefes de nadie, junto con el teléfono de la oficina a la que pertenecen.
  - (0,6 ptos) Productos cuya cantidad máxima vendida en pedido supera a 200.

<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) (0,7 ptos) Listado de los 3 clientes que han comprado más productos distintos, mostrando además la cantidad total de unidades compradas. Deberá estar ordenado primero por el número de productos (de mayor a menor) y después por la cantidad total de unidades compradas (de menor a mayor).
- b) (1 punto) Completar la base de datos, mostrando el código SQL necesario para la realización de las siguientes acciones (1/3 punto por acción):
  - (i) Crear la tabla **pago**, incluyendo la restricción de que el total del pago debe estar entre 100 y 900.
  - (ii) 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 950. ¿Qué ocurre al intentar ejecutarlo? Razone su respuesta.
  - (iii) ¿Cuál sería la expresión SQL para eliminar todas las oficinas de España? ¿Ocasionaría algún error su ejecución? ¿A qué se debe?

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

$$\begin{aligned}
 AT &= \{C, D, E, F, G, H, I, J, K, L, M\}, \\
 DEP &= \{C \rightarrow H, CF \rightarrow I, F \rightarrow CEHD, HI \rightarrow J, G \rightarrow KL, L \rightarrow M, E \rightarrow FC\}
 \end{aligned}$$

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

- a) Obtener paso a paso el recubrimiento minimal.
- b) Obtener las claves candidatas (recuerde, puede apoyarse en el cálculo de los conjuntos de *imprescindibles*, *descartados* y *resto* de campos).
- c) Determinar el grado de normalización de la relación.
- d) Normalizar hasta 3FN.

**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 (1/3 por consulta):
  - (i) Datos de los libros cuyo **status** sea distinto de PUBLISH, ordenados descendentemente por título.
  - (ii) Número de libros que sean posteriores a 2010 o tengan una categoría distinta a Java.
  - (iii) Título, año y número de páginas del libro cuya descripción corta termina en **change.**, sin distinguir mayúsculas de minúsculas.
- b) (1 punto) Realizar las siguientes operaciones de manipulación de datos (1/3 op.):
  - (i) Actualizar el libro **Specification by Example** para que su imagen (campo **thumbnailUrl**) tome el valor **gojkoAdzic.jpg**.
  - (ii) Incorporar un nuevo libro que le guste, con los datos que conozca del mismo, incluyendo categorías. Identifíquelo con el valor 2000.

---

<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) 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:
- (i) (0,2) Número de páginas totales (entre todos los libros) de cada año, mostrando primero los años más recientes.
  - (ii) (0,5) Restringiendo a los años posteriores a 2010 y las iniciales A, C y E, calcular para cada valor de año e inicial, el título y el número de páginas del libro con mayor número de páginas. Finalmente, ordénalo descendientemente por año, y ascendientemente por inicial dentro del año.
  - (iii) (0,3) 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 una base de datos de un Sistema de Información Geográfica, **esbozando con documentos** de ejemplo, teniendo en cuenta los siguientes requisitos:
- *Se almacenará información sobre ríos, con un código de río, nombre, descripción y longitud total. Además, para cada río se almacenarán las provincias por las que pasa y la longitud del tramo del río para cada provincia bañada.*
  - *De las provincias se almacenará un código administrativo de provincia, el nombre y el número de habitantes, así como la capital de la provincia y la Comunidad Autónoma a la que pertenece.*
  - *Los ríos pueden ser afluentes de otros ríos. En este caso, para cada afluente se desea conocer de qué río lo es, en qué provincia se une a dicho río y en qué punto kilométrico lo hace.*
  - *En cuanto a los sistemas montañosos, se almacenará un código identificativo, el nombre, la orientación y la longitud, así como la altura máxima y las provincias que ocupa.*
  - *Los sistemas están formados por montes con código, nombre, descripción y altura. Un monte sólo pertenecerá a un sistema. De los montes también se quiere conocer la provincia o provincias en las que se encuentra (hay casos en los que un monte es compartido por varias provincias). Los montes además pueden tener un origen volcánico o de plegamiento. En el caso de que su origen sea volcánico, se desea almacenar el tipo de volcán.*
  - *Algunos ríos y montes son elementos geológicos monitorizados por satélite. De dichos elementos se desea almacenar la fecha en la que se comienza su monitorización y el satélite que realiza el seguimiento. Un elemento monitorizado sólo puede serlo por un satélite y un satélite podrá monitorizar varios elementos.*