

Caso T5

1) Una tienda de uniformes ha encargado un sistema de información en base a los siguientes requerimientos limitados a la parte que vamos a tratar (se supone que el sistema de información es mucho más grande, ya que normalmente se manejan cosas como nóminas, facturas, inventario, etc.):

- La tienda vende solo cuatro tipos de producto: pantalones, camisas, chaquetas y jerséis.
- Cuando un cliente compra un producto, se genera un pedido o bien al propio almacén de la tienda o bien al fabricante, si no hay stock. En caso de que haya stock, pero no el suficiente, se generan los dos pedidos simultáneamente. Cuando la tienda ha recibido todo el producto, avisa al cliente para que vaya a recogerlo. Las compras solo pueden ser de un determinado producto, pero del número de unidades que se quiera.
- La tienda guarda la información de los clientes normales y también de los llamados "VIP", que son los que, por diversas circunstancias, tienen acceso a descuentos, promociones, ventajas, etc. Esta información no se utiliza habitualmente a menos que el cliente se identifique con su tarjeta de cliente VIP.

Así pues, con estos requerimientos, una empresa consultora ha diseñado una base de datos para que funcione con la aplicación que van a desarrollar. Las tablas de la BD son las siguientes (en algún caso se hace referencia a códigos/claves de otras tablas que no aparecen porque quedan fuera del ejercicio):

Tabla "clientes"		
Columna	Tipo	Descripción
cod_cliente	integer unique not null	Código de cliente
tipo_doc	char(1) not null	"N"=NIF, "E"=NIE, "P"=pasaporte
num_doc	char(16) not null	Número del documento identificador
cod_pais	char(2) not null	Código de dos letras del país ("ES"=España)
apellidos	char(40) not null	Apellidos
nombre	char(40) not null	Nombre(s) de pila

Tabla “clientesvip”		
Columna	Tipo	Descripción
cod_clientevip	integer unique not null	Código de cliente
tipo_doc	char(1) not null	“N”=NIF, “E”=NIE, “P”=pasaporte
num_doc	char(16) not null	Número del documento identificador
cod_pais	char(2) not null	Código de dos letras del país (“ES”=España)
apellidos	char(40) not null	Apellidos
nombre	char(40) not null	Nombre(s) de pila
fechainicio	date not null	Fecha de inicio como cliente VIP
fechafin	date not null	Fecha de caducidad de la tarjeta VIP
descuento	decimal(5,2) not null	Descuento fijo por ser cliente VIP
productos	text not null	Explicación de a qué productos se aplica el descuento
motivo	text not null	Motivo de alta como cliente VIP
recomendadopor	char(40) not null	Persona que lo recomendó
fechanac	date	Fecha de nacimiento
promocumple	text not null	Promoción que se aplica en su cumpleaños

Tabla “ventas”		
Columna	Tipo	Descripción
cod_venta	integer unique not null	Código de venta
unidades	integer not null	Número de unidades del producto
fechaventa	date not null	Fecha de la venta
cliente	integer	Código del cliente que hace la compra (*)
clientevip	integer	Código del cliente VIP que hace la compra (*)
pantalon	integer	Código del pantalón (**)
camisa	integer	Código de la camisa (**)
chaqueta	integer	Código de la chaqueta (**)
jersey	integer	Código del jersey (**)

(*) Uno de los dos códigos de cliente debe ser NOT NULL y el otro NULL.

(**) Uno de los cuatro códigos de producto debe ser NOT NULL. y los otros NULL.

Tabla “pantalones”		
Columna	Tipo	Descripción
cod_pantalon	char(10) unique not null	Código de pantalón
marca	char(40) not null	Marca
modelo	char(40) not null	Modelo
talla	char(10) not null	Talla
descripcion	char(40) not null	Texto descriptivo

Tabla “chaquetas”		
Columna	Tipo	Descripción
cod_chaqueta	char(10) unique not null	Código de chaqueta
marca	char(40) not null	Marca
modelo	char(40) not null	Modelo
talla	char(10) not null	Talla
descripcion	char(40) not null	Texto descriptivo

Tabla “camisas”		
Columna	Tipo	Descripción
cod_camisa	char(10) unique not null	Código de camisa
marca	char(40) not null	Marca
modelo	char(40) not null	Modelo
talla	char(10) not null	Talla
descripcion	char(40) not null	Texto descriptivo

Tabla “jerseys”		
Columna	Tipo	Descripción
cod_jersey	char(10) unique not null	Código de jersey
marca	char(40) not null	Marca
modelo	char(40) not null	Modelo
talla	char(10) not null	Talla
descripcion	char(40) not null	Texto descriptivo

Tabla “ped_almacen”		
Columna	Tipo	Descripción
cod_pedalm	integer not null	Código del pedido al almacén
pantalon	integer	Código del pantalón (*)
camisa	integer	Código de la camisa (*)
chaqueta	integer	Código de la chaqueta (*)
jersey	integer	Código del jersey (*)
unidades	integer not null	Unidades del producto
operario	integer	Código del operario que se encargará del pedido (**)
estado	char(40) not null	Notas sobre el estado del pedido
incidencias	char(80) not null	Descripción de incidencias
entregado	timestamp	Fecha y hora de la entrega en la tienda
conductor	integer	Conductor que ha transportado el pedido (**)

(*) Uno de los cuatro códigos de producto debe ser NOT NULL. y los otros NULL.

(**) Claves ajenas de tablas fuera del ámbito del ejercicio

Tabla "ped_fabric"		
Columna	Tipo	Descripción
cod_pedfabric	integer not null	Código del pedido al fabricante
pantalón	integer	Código del pantalón (*)
camisa	integer	Código de la camisa (*)
chaqueta	integer	Código de la chaqueta (*)
jersey	integer	Código del jersey (*)
unidades	integer not null	Unidades del producto
fabricante	integer not null	Código del fabricante (**)
fechaprevista	date	Fecha de entrega prevista
transportista	char(60)	Notas transportista y número de seguimiento
precio	decimal(6,2)	Precio del pedido

(*) Uno de los cuatro códigos de producto debe ser NOT NULL. y los otros NULL.

(**) Claves ajenas de tablas fuera del ámbito del ejercicio

A partir de la información expuesta, se debe revisar el diseño de las tablas que ha hecho la empresa consultora y explicar razonadamente y de manera breve qué cambios habría que hacer, juntando, separando o reorganizando tablas, para mejorar el modelo (no es necesario escribir todas las tablas modificadas y sus campos). Finalmente, se debe dibujar el modelo físico de la BD modificada.

Valor de la pregunta: 50% de la nota del caso

2) Explicar qué función realizan los siguientes programas en Java:

a) (Valor: 12,5%)

```
public class X {
    static int calc (int c) {
        int s = 0;
        while (c != 0) {
            s = s + c%10;
            c = c/10;
        }
        return s;
    }

    public static void main(String[] args) throws Exception {
        int s=0;

        for (int i=0;i<args.length;i++) {
            try {
                int p=Integer.parseInt(args[i]);
                s=s+calc(p);
            } catch (NumberFormatException e) {
                s=s+args[i].length();
            }
        }
        System.out.print("Result: ");
        System.out.println(s);
    }
}
```

b) (Valor: 12,5%)

```
public class X {  
  
    public static void fn(String[] a) {  
  
        if (a == null || a.length < 2) {  
            return;  
        }  
  
        for (int i = 0; i < a.length / 2; i++) {  
            String t = a[2*i];  
            a[2*i] = a[2*i+1];  
            a[2*i+1] = t;  
        }  
  
    }  
  
    public static void main(String[] args) throws Exception {  
        System.out.println(String.join(", ", args));  
        fn(args);  
        System.out.println(String.join(", ", args));  
    }  
}
```

c) (Valor: 12,5%)

```
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
  
public class X {  
    public static void main(String[] args) throws Exception {  
        BufferedReader r = null;  
        BufferedWriter w = null;  
        r = new BufferedReader(new FileReader(args[0]));  
        w = new BufferedWriter(new FileWriter(args[1]));  
  
        String p = null;  
        String c = r.readLine();  
        int n = 0;  
        while (c != null) {  
            if (c.equals(p)) {  
                n++;  
            } else {  
                if (n > 0) {  
                    w.write("* ");  
                    w.write(Integer.toString(n));  
                    w.newLine();  
                    n = 0;  
                }  
                w.write(c);  
                w.newLine();  
                p = c;  
            }  
            c = r.readLine();  
        }  
        if (n > 0) {  
            w.write("* ");  
            w.write(Integer.toString(n));  
            w.newLine();  
        }  
        r.close();  
        w.close();  
    }  
}
```

d) (Valor: 12,5%)

```
import java.util.HashSet;

public class X {

    public static void main(String[] args) throws Exception {
        HashSet<String> set = new HashSet<String>();
        System.out.print("Result:");
        for (int i=0;i<args.length;i++) {
            // set.add(X) -> true si el conjunto aún no contenía X
            if (set.add(args[i])) {
                System.out.print(" "+args[i]);
            }
        }
        System.out.println();
    }
}
```

Valor de la pregunta: 50% de la nota del caso