



## **EJERCICIOS 4-2.**

### **EJERCICIOS. DISEÑO FISICO. DDL**

Base de Datos  
CFGs DAW

Francisco Aldarias Raya  
[paco.aldarias@ceedcv.es](mailto:paco.aldarias@ceedcv.es)

2019/2020

Fecha 30/11/19

Versión:191130.0946


## Licencia



**Reconocimiento - NoComercial - CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

## Nomenclatura

20/11/2019. Ejercicio3. Corregida errata en Recurso.

# ÍNDICE DE CONTENIDO

<b>1. EJERCICIO 1. ESCALA.....</b>	<b>3</b>
1.1 Ejercicio 1.1.....	3
1.2 Ejercicio 1.2.....	4
1.3 Ejercicio 1.3.....	4
1.4 Ejercicio 1.4.....	4
1.5 Ejercicio 1.5.....	4
1.6 Ejercicio 1.6.....	4
1.7 Ejercicio 1.7.....	4
1.8 Ejercicio 1.8.....	4
1.9 Ejercicio 1.9.....	4
1.10 Ejercicio 1.10.....	4
1.11 Ejercicio 1.11.....	5
<b>2.EJERCICIO 2 SUCURSALES.....</b>	<b>5</b>
<b>3.EJERCICIO 3 VENTAS.....</b>	<b>7</b>

## UD04-2. EJERCICIOS. DISEÑO FÍSICO. DDL

Realiza los ejercicios de manera individual y comenta en el foro de la unidad aquellos en los que tengas dudas. Realizar las operaciones en DDL de MySQL, Oracle y Postgres (si existe alguna diferencia).

### 1. EJERCICIO 1. ESCALA

Traduce el siguiente modelo relacional al SQL :

Asociación (código, nombre, ubicación)

CP:{código}

Alpinistas (nif, nombre, fecha\_nacimiento, {teléfonos}, código, fecha\_ingreso)

CP:{nif}

CA:{código} → Asociación

Teléfonos (nif, telefono)

CP:{nif,telefono}

CA:{nif} → Alpinistas

Picos (nombre, altura, coordenadas, país)

CP: {nombre}

Escala (nif, nombre, fecha\_inicio, tiempo, oxígeno, cara)

CP{nif, nombre}

CA{nif} → alpinista

CA{nombre} → pico

#### 1.1 Ejercicio 1.1

Antes de crear las tablas, vamos a crear el usuario alpinista, de manera que las tablas que vamos a crear se inserten en el esquema alpinista.

## 1.2 Ejercicio 1.2

Creamos las tablas, hazlo ejecutando el script alpinista.sql

## 1.3 Ejercicio 1.3

Añade un campo 'observaciones' sobre la tabla PICOS (decide tú el tipo y tamaño del mismo).

## 1.4 Ejercicio 1.4

Modifica el campo código de la tabla asociaciones para que pueda contener números hasta el 9999999.

## 1.5 Ejercicio 1.5

Añade un campo 'foto' sobre la tabla ALPINISTAS (decide tú el tipo y tamaño del mismo).

## 1.6 Ejercicio 1.6

Modifica la columna fecha\_ingreso de la table alpinistas para que por defecto tenga la fecha del sistema.

## 1.7 Ejercicio 1.7

El valor del campo 'coordenadas' no debe repetirse nunca, a pesar de no ser clave primaria. Introduce la correspondiente restricción sobre PICOS.

## 1.8 Ejercicio 1.8

Si se borra un alpinista de la base de datos ¿qué debería hacerse con sus correspondientes entradas en la tabla TELEFONOS? Indica la sentencia DDL necesaria para aplicar esta restricción.

## 1.9 Ejercicio 1.9

¿Y si se borra una asociación de la base de datos?

## 1.10 Ejercicio 1.10

Desactivar la clave foránea de la tabla alpinistas.

### 1.11 Ejercicio 1.11

Volver a activar la clave foránea desactivada en la pregunta anterior.

## 2. EJERCICIO 2 SUCURSALES

Tenemos las siguientes tablas creadas, pero no se ha creado ninguna clave ajena. Se trata de que vosotros analicéis cuáles son las claves ajenas necesarias para el correcto funcionamiento del sistema y las añadáis a la base de datos por medio de modificaciones de las tablas (ALTER TABLE .....).

```
CREATE TABLE articulos(  
  id_art INTEGER,  
  nombre VARCHAR(30) default "",  
  precio NUMBER default 0.0,  
  codigo VARCHAR(7) default "",  
  id_gru INTEGER default 0 NOT NULL,  
  CONSTRAINT ART_ID_PK PRIMARY KEY(id_art)  
);
```

```
CREATE TABLE clientes (  
  id_cli INTEGER NOT NULL ,  
  nombre VARCHAR(60) default "" NOT NULL ,  
  direccion VARCHAR(80) default "",  
  constraint cli_idc_pk PRIMARY KEY (id_cli)  
);
```

```
CREATE TABLE facturas (  
  id_fac INTEGER NOT NULL ,  
  serie VARCHAR(1) default "" NOT NULL,  
  numero INTEGER default 0 NOT NULL,  
  fecha date,  
  id_cli INTEGER default 0 NOT NULL,  
  id_ven INTEGER NOT NULL,  
  CONSTRAINT FAC_IDF_PK PRIMARY KEY (id_FAC)
```

```
);
```

```
CREATE TABLE grupos (  
  id_gru INTEGER,  
  descripcion VARCHAR(15) default '' NOT NULL,  
  CONSTRAINT GRU_IDG_PK PRIMARY KEY (id_gru)  
);
```

```
CREATE TABLE lineas_factura (  
  id_lin INTEGER NOT NULL,  
  id_fac INTEGER default 0 NOT NULL ,  
  importe FLOAT default 0 NOT NULL ,  
  id_art INTEGER NOT NULL,  
  cantidad INTEGER NOT NULL,  
  fecha date default '01/01/2000' NOT NULL ,  
  id_suc INTEGER default 0 NOT NULL ,  
  constraint LIN_PK PRIMARY KEY (id_lin, factura)  
);
```

```
CREATE TABLE sucursales (  
  id_suc INTEGER NOT NULL,  
  descripcion VARCHAR(15) default '' NOT NULL ,  
  CONSTRAINT SUC_IDS_PK PRIMARY KEY (id_suc)  
);
```

```
CREATE TABLE vendedores (  
  id_ven INTEGER NOT NULL,  
  nombre VARCHAR(50) default '' NOT NULL ,  
  fecha_ingreso date default '01-01-2000' NOT NULL ,  
  salario float default 0 NOT NULL ,  
  CONSTRAINT VEN_IDV_PK PRIMARY KEY (id_ven)  
);
```

### 3. EJERCICIO 3 VENTAS

1. USUARIO (dni, nombre, fecha\_nac)

CP: DNI

2. CLIENTE (dni, descuento )

CP: dni

CA: dni → Usuario

Restricción CA dni : Un usuario no puede borrarse si es cliente.

3. PEDIDO (codigo, fecha, dni\_cliente )

CP:codigo

CA: dni\_cliente → Cliente

VNN: dni\_cliente

Restricción CA dni\_cliente: Un cliente no puede borrarse si tiene lineas de pedido.

4. LINEA ( numero\_linea, precioventa, codigo\_pedido, codigo\_producto, cantidad)

CP : { numero\_linea, codigo\_pedido}

CA: Codigo\_pedido → Pedido

CA: codigo\_producto→Producto

VNN: codigo\_producto

Restricción CA Codigo\_pedido: Si se borra o actualiza un pedido se borra todas las lineas de ese pedido en cascada.

Restricción CA Codigo\_producto: Un producto no podrá borrarse de existen en lineas del pedido.

4. PRODUCTO ( codigo\_producto, nombre, precioactual)

CP: codigo\_producto

Realiza el diseño físico y su diagrama.