

Bases de Datos y Sistemas de Información

Grado en Ingeniería Informática

Unidad Didáctica 1: Bases de Datos Relacionales

Parte 3: Interpretación de una base de datos

(Doc. UD1.3)

Curso 2021/2022



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

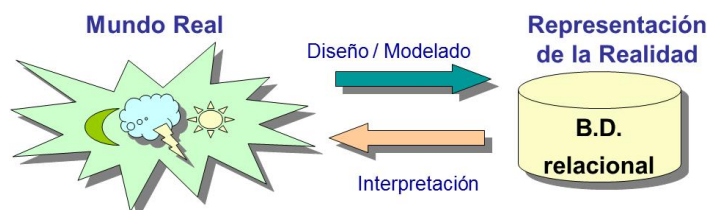
Índice

1 Introducción	1
2 Interpretación de una base de datos relacional	1
3 Base de datos Música	1
4 Identificación de objetos	2
5 Identificación de relaciones	3
6 Interpretación de las restricciones	4
6.1 Interpretación de la restricción de Valor No Nulo	4
6.2 Interpretación de la restricción de unicidad	5
7 Interpretación de la base de datos Música.....	5
8 Ejercicios propuestos	6
8.1 Base de datos Información Geográfica	6
8.2 Base de datos Biblioteca	7
8.3 Base de datos Empresa I	8
8.4 Base de datos Empresa II	9
8.5 Base de datos Agencia de Viajes	9

1 INTRODUCCIÓN

Una vez presentado el Modelo Relacional, es importante reflexionar, aunque sea de forma intuitiva, sobre la forma en que los datos resultantes de la observación de la realidad se pueden estructurar en relaciones para capturar el conocimiento que se tiene de dicha realidad.

En la figura siguiente se muestra la relación existente entre el mundo real y una base de datos:



Al proceso de obtención de un esquema relacional que represente las propiedades de interés del mundo real se le denomina *diseño de la base de datos*. Dada una base de datos, el proceso de comprensión del mundo real que modela se denomina *interpretación de una base de datos*. De los dos procesos, el primero es el más difícil y se estudiará en la segunda parte de la asignatura (unidad didáctica 4). El proceso de interpretación, sin embargo, es relativamente sencillo y se basa en el estudio de las restricciones que se han definido en el esquema.

2 INTERPRETACIÓN DE UNA BASE DE DATOS RELACIONAL

Para comprender el mundo real que representa una base de datos relacional, es necesario identificar los siguientes elementos:

- **Objetos** de los que se quiere almacenar información incluyendo las propiedades de interés para cada uno de ellos.
- **Relaciones**¹ entre los objetos y propiedades de las relaciones si es que existen.

En el apartado siguiente se presenta un esquema relacional que se utilizará a lo largo de la unidad didáctica.

3 BASE DE DATOS MÚSICA

Sea el siguiente esquema que almacena información musical:

CANCIÓN (cod:entero, título:char(30), duración:real)

CP: {cod}

VNN: {título}

COMPañÍA (cod:char(3), nombre:char(30), dir:char(30), fax:char(10), tfno:char(10))

CP: {cod}

VNN: {nombre}

DISCO (cod:char(3), nombre:char(30), fecha:date, cod_comp:char(3), cod_gru:char(3))

CP: {cod}

CAj: {cod_comp} → COMPañÍA

VNN: {cod_comp}

CAj: {cod_gru} → GRUPO

VNN: {cod_gru}

¹ La palabra *relación* en el mundo de las bases de datos es polisémica ya que hace referencia, por una parte a la estructura del modelo relacional y, por otra, a la conexión entre objetos de la realidad que se está modelando. Esta confusión no existe en inglés donde la estructura se denomina *relation* y la conexión *relationship*.

```

ESTÁ (can:entero, cod:char(3))
  CP: {can, cod}
  CAj: {can} → CANCIÓN
  CAj: {cod} → DISCO

GRUPO (cod:char(3), nombre:char(30), fecha:date, país:char(10))
  CP: {cod}
  VNN: {nombre}
  Único: {nombre}

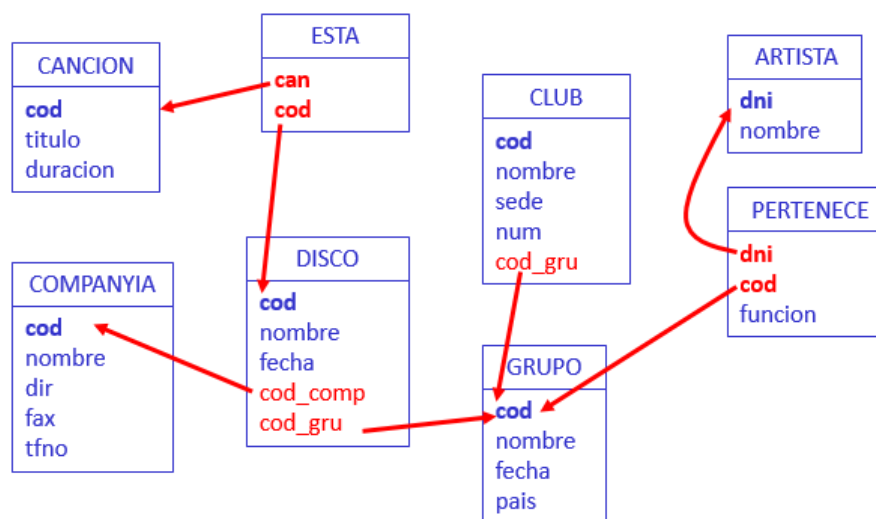
ARTISTA (dni:char(10), nombre:char(30))
  CP: {dni}
  VNN: {nombre}

CLUB (cod:char(3), nombre:char(30), sede:char(30), num:entero, cod_gru:char(3))
  CP: {cod}
  CAj: {cod_gru} → GRUPO
  VNN: {cod_gru}
  Único: {cod_gru}
  VNN: {nombre}

PERTENECE (dni:char(10), cod:char(3), función:char(15))
  CP: {dni, cod}
  CAj: {dni} → ARTISTA
  CAj: {cod} → GRUPO

```

Un esquema relacional también se puede presentar gráficamente, de forma que cada relación se representa en un rectángulo donde se incluyen sus atributos (la CP en negrita) y donde las claves ajenas se destacan mediante flechas que conectan las relaciones. La representación gráfica del esquema de música sería el siguiente:



4 IDENTIFICACIÓN DE OBJETOS

Para comprender la realidad para la que se ha diseñado la base de datos lo primero que debe hacerse es buscar los esquemas de relación que almacenan información sobre objetos de interés en el sistema de información. En muchos casos para ello basta con localizar esquemas de relación cuya clave primaria no contenga ninguna clave ajena a otra relación. En general, cada una de estas relaciones se corresponde con un objeto del que se quiere almacenar información y cada atributo de la relación que no es una clave ajena a otra relación representa una propiedad de ese objeto.

En el esquema ejemplo se pueden detectar los siguientes objetos de información con las siguientes propiedades:

- **Canción:** que tiene como propiedades el título, la duración y el código (que la identifica).

- *Compañía*: que tiene como propiedades el nombre, la dirección postal, el número de fax, el teléfono y el código (que la identifica).
- *Disco*: que tiene como propiedades el nombre, una fecha y el código (que lo identifica).
- *Grupo*: que tiene como propiedades el nombre, una fecha, un país y el código (que lo identifica).
- *Artista*: que tiene como propiedades el nombre y el DNI (que lo identifica).
- *Club*: que tiene como propiedades el nombre, la dirección postal de la sede, un número y el código (que la identifica).

Es importante darse cuenta de que con la mera observación y estudio del esquema relacional no siempre será posible “entender” qué información se almacena en la base de datos; en ocasiones habrá que recurrir a los usuarios de la misma para comprender todo su significado. En el caso de la base de datos anterior hemos sacado algunas conclusiones como que el atributo *dir* de la relación *Compañía* representa la dirección postal de la misma, al igual que el atributo *sede* de la relación *Club* también es una dirección, sin embargo, ¿qué representa el atributo *fecha* de la relación *Disco*? y ¿qué significa el atributo *num* de la relación *Club*? Para poder manipular la base de datos, realmente no sería necesario ir más allá de saber que cada disco del que se almacene información tiene asociada una fecha y que cada club tiene un número pero siempre facilita la manipulación posterior un mayor conocimiento de las propiedades de cada objeto.

5 IDENTIFICACIÓN DE RELACIONES

Los objetos de los que se quiere almacenar información en un sistema, raramente están aislados unos de otros, en muchos casos se conectan mediante relaciones. Para identificar estas relaciones es necesario estudiar cada una de las claves ajenas del esquema relacional. En general, se pueden distinguir dos situaciones:

- Clave ajena en una relación que representa un objeto. Estas claves ajenas representan la conexión entre el objeto representado en la relación que contiene la clave ajena y el objeto representado en la relación a la que apunta la clave ajena. El significado de esta relación no siempre será evidente ya que en el esquema relacional no tiene un nombre definido; de nuevo la ayuda de los usuarios de la base de datos podrá clarificar las dudas que se planteen. Se pueden distinguir dos casos:

- La clave ajena no tiene restricción de unicidad en la relación que la contiene. En el esquema ejemplo tenemos:
 - Clave ajena de la relación *Disco* a la relación *Compañía*. Esta clave ajena conecta cada disco con una compañía.
 - Clave ajena de la relación *Disco* a la relación *Grupo*. Esta clave ajena conecta cada disco con un grupo.

Este tipo de relaciones se llaman “1:M” o “Uno a Muchos” ya que sólo hay que reflexionar un poco para concluir que en cada tupla de la relación que contiene la clave ajena, ésta sólo puede tomar un valor (1) mientras que nada prohíbe que en varias tuplas de la relación la clave ajena tenga el mismo valor (M). En el ejemplo:

- Clave ajena de la relación *Disco* a la relación *Compañía*. Un disco sólo se relaciona con una compañía (aquella cuyo código aparezca en el atributo *cod_comp* de *Disco*). La misma compañía puede relacionarse con muchos discos ya que puede haber varias tuplas en *Disco* con el mismo valor en el atributo *cod_comp*.
- Clave ajena de la relación *Disco* a la relación *Grupo*. Un disco sólo se relaciona con un grupo (aquél cuyo código aparezca en el atributo *cod_gru* de *Disco*). El mismo grupo puede relacionarse con muchos discos ya que puede haber varias tuplas en *Disco* con el mismo valor en el atributo *cod_gru*.
- La clave ajena tiene restricción de unicidad en la relación que la contiene. En el esquema ejemplo tenemos:
 - Clave ajena de la relación *Club* a la relación *Grupo*. Esta clave ajena conecta cada club

con un grupo.

Este tipo de relaciones se llaman “1:1” o “Uno a Uno” ya que en cada tupla de la relación que contiene la clave ajena, ésta sólo puede tomar un valor (1) y por otro lado, dada la restricción de unicidad, un valor en la clave ajena sólo puede aparecer como mucho una vez en la relación que contiene la clave ajena(1).

- Clave ajena de la relación *Club* a la relación *Grupo*. Un club sólo se relaciona con un grupo (aquel cuyo código aparezca en el atributo *cod_gru* de *Club*). El mismo grupo sólo puede relacionarse con un club como mucho ya que no puede haber varias tuplas en *Club* con el mismo valor en el atributo *cod_gru* porque se violaría la restricción de unicidad.
- Existen relaciones cuya clave principal está formada por la unión de dos claves ajenas a otras dos relaciones. Estas relaciones representan una conexión entre los objetos representados en las relaciones a las que apuntan las claves ajenas. Estas conexiones suelen quedar descritas por el nombre de la relación que las contiene, además, si la relación entre los objetos tiene alguna propiedad, ésta se representa con un atributo de la relación. En el ejemplo:
 - La relación *Está* representa una conexión entre el objeto representado en la relación *Canción* y el representado en la relación *Disco*.
 - La relación *Pertenece* representa una conexión entre el objeto representado en la relación *Artista* y el representado en la relación *Grupo* con la propiedad *función*.

Este tipo de relaciones se llaman “M:M” o “Muchos a Muchos”. En el ejemplo:

- En la relación *Está* la clave primaria es {*can, cod*} por lo que se puede repetir el valor del atributo *can* (que es una clave ajena a *Canción*) siempre y cuando aparezca con diferentes valores del atributo *cod* (que es una clave ajena a *Disco*) y viceversa así pues una canción se puede relacionar con muchos discos y un disco se puede relacionar con muchas canciones.
- En la relación *Pertenece* se puede repetir el valor del atributo *dni* (que es una clave ajena a *Artista*) siempre y cuando aparezca con diferentes valores del atributo *cod* (que es una clave ajena a *Grupo*) y viceversa así pues un artista se puede relacionar con muchos grupos y un grupo se puede relacionar con muchos artistas.

6 INTERPRETACIÓN DE LAS RESTRICCIONES

En los dos apartados anteriores se ha presentado como identificar los objetos de información y las conexiones existentes entre ellos, sin embargo, aún hay propiedades de la realidad que no se han detectado y que pueden descubrirse estudiando las restricciones de *unicidad* y de *valor no nulo* que puedan haberse definido en el esquema relacional.

6.1 Interpretación de la restricción de Valor No Nulo

Si la restricción de valor no nulo afecta a un atributo que no es clave ajena a otra relación su significado es evidente: ‘el valor de la propiedad representada en el atributo es necesario para que se introduzca información del objeto que describe’. Es el caso, por ejemplo, del atributo *título* en la relación *Canción* representa la siguiente propiedad: ‘es imprescindible conocer el título de todas las canciones’.

Sin embargo, si la restricción afecta a una clave ajena, la propiedad que representa es diferente ya que significa que cada objeto representado en la relación debe relacionarse obligatoriamente con algún objeto representado en la relación a la que apunta la clave ajena. Así, el valor no nulo del atributo *cod_gru* en la relación *Club* representa la siguiente propiedad: ‘todos los clubes están asociados a un grupo al que admiran’.

6.2 Interpretación de la restricción de unicidad

Si la restricción de unicidad afecta a un atributo que no es clave ajena a otra relación su significado es evidente: ‘el valor de la propiedad representada en el atributo no se puede repetir para distintos objetos representados en la relación’. En el ejemplo, la unicidad del atributo *nombre* de la relación *Grupo* representa la siguiente propiedad: ‘no puede haber dos grupos con el mismo nombre’.

Sin embargo, si la restricción afecta a una clave ajena, como ya se ha visto en el apartado 5, la propiedad que representa es diferente ya que significa que cada objeto representado en la relación a la que apunta la clave ajena puede relacionarse como mucho con un objeto representado en la relación que la contiene. Así, la unicidad del atributo *cod_gru* en la relación *Club* representa la siguiente propiedad: ‘un grupo puede tener como mucho un club de fans’.

7 INTERPRETACIÓN DE LA BASE DE DATOS MÚSICA

Una vez estudiado el esquema de la base de datos, con un poco de imaginación y un poco de ayuda, podríamos describir la realidad que representa como se describe a continuación.

La base de datos almacena información sobre una discoteca referente a qué discos contiene, qué compañías discográficas los han editado, qué canciones hay grabadas y por qué grupo; la información almacenada en cada una de las relaciones del esquema es la siguiente:

- *Canción*: almacena información de las canciones grabadas en los discos. Propiedades de interés:
 - *cod*: código de la canción. (Identificador)
 - *título*: cómo se llama la canción. (Valor obligatorio)
 - *duración*: cuánto dura aproximadamente la canción.
- *Compañía*: almacena información de las compañías discográficas que han editado los discos. Propiedades de interés:
 - *cod*: código de la compañía editora de discos. (Identificador)
 - *nombre*: cómo se llama la compañía. (Valor obligatorio)
 - *dir*: dirección postal de la compañía.
 - *fax*: número de fax de contacto.
 - *tfno*: número de teléfono de las oficinas.
- *Grupo*: almacena información de grupos musicales. Propiedades de interés:
 - *cod*: código del grupo. (Identificador)
 - *nombre*: cómo se llama el grupo. (Valor obligatorio y único)
 - *fecha*: cuándo se constituyó el grupo.
 - *país*: dónde surgió el grupo.
- *Disco*: almacena información de los discos. Propiedades de interés:
 - *cod*: código que identifica cada disco. (Identificador)
 - *nombre*: título del disco.
 - *fecha*: fecha en que se editó el disco.
 - *cod_comp*: código de la compañía que ha editado el disco. (Valor obligatorio). Representa una relación con las compañías con las siguientes características: todo disco ha sido editado por una compañía y nada más que una; una compañía puede editar cualquier cantidad de discos (0, 1, 2, ...).
 - *cod_gru*: código del grupo que ha grabado el disco. (Valor obligatorio). Representa una relación con los grupos con las siguientes características: todo disco ha sido grabado por un grupo y nada más que uno; un grupo puede grabar cualquier cantidad de discos.
- *Artista*: almacena información de cantantes. Propiedades de interés:

- *dni*: nº de dni del artista. (Identificador)
- *nombre*: cómo se llama el artista. (Valor obligatorio)
- *Club*: almacena información de clubes de fans. Propiedades de interés:
 - *cod*: código del club de fans. (Identificador)
 - *nombre*: cómo se llama el club. (Valor obligatorio)
 - *sede*: dónde tiene la sede.
 - *num*: cuántas personas pertenecen al club.
 - *cod_gru*: código del grupo que se admira. (Valor obligatorio). Representa una relación con los grupos con las siguientes características: todo club admira exactamente a un grupo; un grupo puede tener como mucho un club de fans (es decir, o no tiene o tiene uno).
- *Está*: almacena información sobre qué canciones hay grabadas en cada disco. Es decir, *can* es el código de la canción que está grabada en el disco de código *cod*. Representa una relación entre las canciones y los discos con las siguientes características: una canción puede aparecer en cualquier número de discos y un disco puede contener cualquier número de canciones.
- *Pertenece*: almacena información sobre quién forma parte de cada grupo. Es decir, el artista con *dni* forma parte del grupo de código *cod* y realiza en éste la tarea especificada en *función*. Representa una relación entre los artistas y los grupos con las siguientes características: un artista puede pertenecer a cualquier número de grupos y a un grupo pueden pertenecer cualquier número de artistas.

8 EJERCICIOS PROPUESTOS²

En los apartados siguientes se proponen una serie de esquemas relacionales para que se realice su interpretación de forma similar a cómo se ha realizado con el esquema de la discoteca. Para ello, para cada esquema deberá:

- Descubrir los objetos de interés y las propiedades que los describen.
- Descubrir las relaciones existentes entre los objetos y sus características.

8.1 Base de datos Información Geográfica

```
RÍO(rcod:char(3),nombre:char(30),longitud:real,mcod:char(3))
  CP:{rcod}
  CAj:{mcod}→MAR
```

```
MAR(mcod:char(3),nombre:char(30),detalles:char(150))
  CP:{mcod}
```

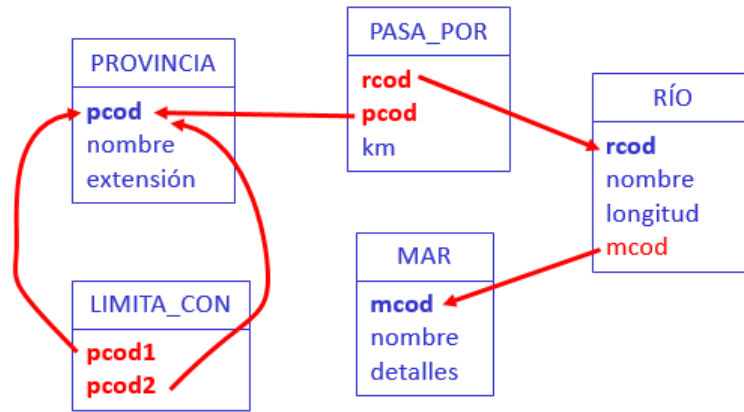
```
PROVINCIA(pcod:char(2),nombre:char(30),extensión:real)
  CP:{pcod}
```

```
PASA_POR(rcod:char(3),pcod:char(2),km:real)
  CP:{pcod,rcod}
  CAj:{pcod}→PROVINCIA
  CAj:{rcod}→RÍO
```

```
LIMITA_CON(pcod1:d_pcod,pcod2:d_pcod)
  CP:{pcod1,pcod2}
  CAj:{pcod1}→PROVINCIA
  CAj:{pcod2}→PROVINCIA
```

² Con lo que se ha presentado hasta aquí es posible interpretar la mayoría de los esquemas, sin embargo, a veces la realidad modelada es más compleja y tiene otro tipo de objetos y relaciones (especializaciones, generalizaciones, ...) que se podrán comprender sin dificultad cuando se estudie el diseño de bases de datos (UD4) pero que ahora es prematuro presentar.

Gráficamente, el esquema es el siguiente:



- Realizar la interpretación del esquema asumiendo que a la relación entre mar y río representa la información de en qué mar desemboca el río.
- Responder las siguientes cuestiones teniendo en cuenta el esquema de la base de datos (no en la realidad que se pueda conocer):
 - ¿Puede un río desembocar en dos mares?
 - ¿Puede un río pasar por dos provincias?
 - ¿Puede un río pasar dos veces por la misma provincia?
 - ¿Puede una provincia limitar con ella misma?
 - ¿En cuántos mares, como máximo, puede desembocar un río?, ¿y cómo mínimo?
 - ¿Qué propiedades de la realidad que modela la base de datos le parece que no están reflejados en el esquema relacional?

8.2 Base de datos Biblioteca

Se desea mantener información de una biblioteca doméstica; para ello se ha definido una base de datos relacional cuyo esquema se muestra a continuación:

AUTOR(autor_id:char(4), nombre:char(35), nacionalidad:char(20))

CP:{autor_id}

VNN:{nombre}

LIBRO(id_lib: char(10), título³:char(80), año:entero, num_obras:entero)

CP:{id_lib}

TEMA(temática:char(20), descripción:char(50))

CP:{temática}

OBRA(cod_ob:entero, título:char(80), temática:char(20))

CP:{cod_ob}

CAj:{temática}→TEMA

VNN:{título}

AMIGO(num:entero, nombre:char(60), teléfono:char(10))

CP:{num}

VNN:{nombre}

LEER(num:entero, cod_ob:entero)

CP:{num, cod_ob}

CAj:{num}→AMIGO

CAj:{cod_ob}→OBRA

³Algunos SGBD no aceptan caracteres especiales, como p. ej. letras acentuadas, en los nombres de atributos o relaciones. Para evitar problemas hemos decidido no acentuar ninguna palabra que figure en la definición de una relación.

```

ESTA_EN(cod_ob:entero,id_lib:char(10))
  CP:{cod_ob,id_lib}
  CAj:{cod_ob}→OBRA
  CAj:{id_lib}→LIBRO

```

```

ESCRIBIR(cod_ob:entero,autor_id:char(4))
  CP:{cod_ob,autor_id}
  CAj:{cod_ob}→OBRA
  CAj:{autor_id}→AUTOR

```

- Representar gráficamente el esquema de la base de datos.
- Realizar la interpretación del esquema.
- Responder las siguientes cuestiones:
 - ¿Puede un amigo haber leído la misma obra más de una vez?
 - ¿Puede una obra tener más de una temática?
 - ¿Un libro puede contener más de una vez la misma obra? ¿Y una obra puede estar incluida en más de un libro?
 - ¿Cuántas obras debe escribir, como mínimo, un autor? ¿Y cómo máximo?
 - ¿Puede haber obras que no aparezcan en ningún libro?
 - ¿Puede haber obras con el mismo título?

8.3 Base de datos Empresa I

```

PROVEEDOR(vcod:char(3),nombre:char(30),ciudad:char(30))
  CP:{vcod}

```

```

PIEZA(zcod:char(4),nombre:char(30),color:char(15),peso:real,
      ciudad:d_ciu)
  CP:{zcod}

```

```

PROYECTO(ycod:char(5),nombre:char(50),ciudad:char(30))
  CP:{ycod}

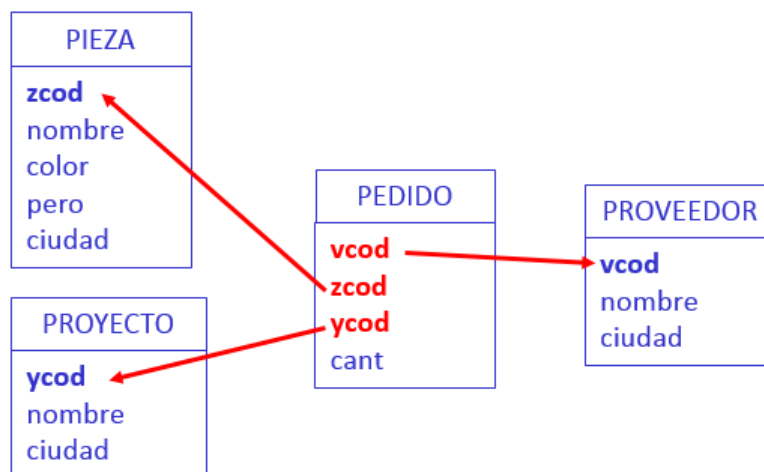
```

```

PEDIDO(vcod:char(3),zcod:char(4),ycod:char(5),cant:entero)
  CP:{vcod,zcod,ycod}
  CAj:{vcod}→PROVEEDOR
  CAj:{zcod}→PIEZA
  CAj:{ycod}→PROYECTO

```

Gráficamente, el esquema es el siguiente:



En este esquema relacional la relación *Pedido* tiene la clave primaria compuesta por la unión de tres claves ajenas siendo por lo tanto una relación *ternaria*

- Realizar la interpretación del esquema, para ello, generalice lo que se ha visto para las relaciones

binarias.

- Responder las siguientes cuestiones:
 - ¿Puede haber dos piezas con el mismo código *zcod*?
 - ¿Puede una pieza tener dos colores?
 - ¿Puede un proveedor servir dos pedidos con la misma pieza y proyecto?
 - Se podría añadir a la relación “*Pedido*” el atributo *fecha*, e incluso podría formar parte de la clave primaria. ¿Qué realidad representaría entonces la relación *Pedido*?

8.4 Base de datos Empresa II

DEPARTAMENTO(código:char(5),ubicación:char(40),director:entero)

CP:{código}

Único:{director}

CAj:{director}→EMPLEADO

EMPLEADO(dni:entero,nombre:char(50),dirección:char(30),cod_dep:char(5))

CP:{dni}

CAj:{cod_dep}→DEPARTAMENTO

VNN:{cod_dep}

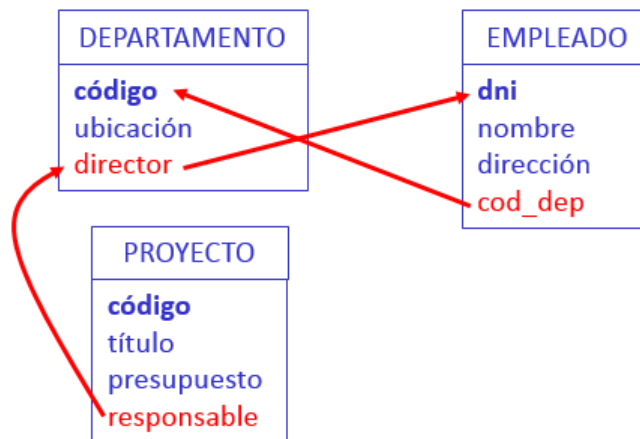
PROYECTO(código:char(5),título:char(100),presupuesto:real,responsable:entero)

CP:{código}

CAj:{responsable}→DEPARTAMENTO

f(responsable) = director

Gráficamente, el esquema es el siguiente:



- Responder las siguientes cuestiones:
 - Dé una interpretación a cada una de las tres asociaciones que existen entre los objetos de este sistema de información.
 - ¿Puede ser el director de un departamento miembro de un departamento distinto?
 - ¿Puede un empleado ser director de dos departamentos distintos? Y ¿puede ser responsable de dos proyectos distintos?
 - ¿Cuántos directores puede tener como máximo un departamento? Y ¿cuántos debe tener como mínimo?

8.5 Base de datos Agencia de Viajes

VIAJE(cod:char(3),fecha:date,precio:real,dni_g:char(11))

CP:{cod}

CAj:{dni_g}→GUÍA

VNN:{fecha}

GUÍA(dni_g:char(11),nombre:char(30),idioma:char(15),edad:entero)
 CP:{dni_g}

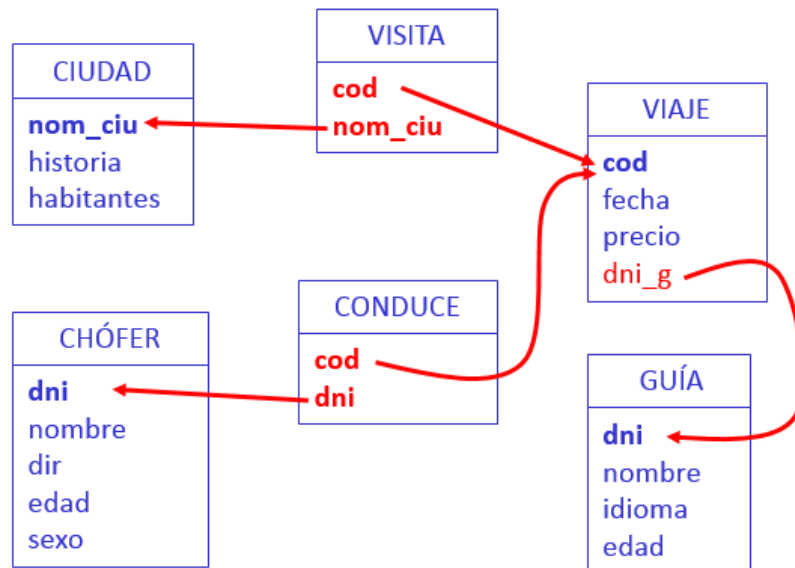
CHÓFER(dni:char(11),nombre:char(30),dir:char(50),edad:entero,sexo:char(1))
 CP:{dni}

CIUDAD(nom_ciu:char(30),historia:char(150),habitantes:entero)
 CP:{nom_ciu}

VISITA(cod:char(3),nom_ciu:char(30))
 CP:{cod,nom_ciu}
 CAj:{cod}→VIAJE
 CAj:{nom_ciu}→CIUDAD

CONDUCE(dni:char(11),cod:char(3))
 CP:{dni,cod}
 CAj:{cod}→VIAJE
 CAj:{dni}→CHÓFER

Gráficamente, el esquema es el siguiente:



- Realizar la interpretación del esquema.
- Responder las siguientes cuestiones:
 - ¿Se puede visitar en el mismo viaje dos veces la misma ciudad?
 - ¿Puede un guía hablar dos idiomas?
 - ¿Puede un chofer estar en dos viajes en una misma fecha?
 - ¿Puede un chofer ser guía también?
 - ¿Cuántos chóferes hay, como mínimo, en cada viaje?
 - ¿Puede un guía participar en más de un viaje?
 - ¿Puede haber alguien que sea chófer y guía a la vez? Si es así, ¿qué problemas podrían aparecer en ese esquema? ¿hay algún esquema alternativo que los resuelva?