



# Unidad Didáctica 4: Diseño de Bases de Datos Relacionales

## Parte 3: Diseño Lógico (Doc. UD4.3)

# UD 4.3.- Diseño lógico

## 1.- Introducción

## 2.- Transformación de las clases

2.1.- Clases fuertes

2.2.- Clases débiles

2.3.- Clases especializadas

## 3.- Transformación de las asociaciones

3.1.- No reflexivas

3.2.- Reflexivas

3.3.- Atributos de enlace

3.4.- Transformación de la asociación cuando se asocia con clases asociación

## 4.- Ejemplo de transformación

## 5.- Teoría de la Normalización

5.1.- Conceptos previos

5.2.- Primera forma normal (1FN)

5.3.- Segunda forma normal (2FN)

5.4.- Tercera forma normal (3FN)

5.5.- Reflexión sobre la teoría de la Normalización

# 1.- Diseño lógico

## 1ª FASE: Análisis

Investigación

Req. de  
información

Req. de  
procesos

## 2ª FASE: Diseño

Modelo semántico

Diseño conceptual

Esquema conceptual

Estática

Dinámica

Tecnología de gestión  
de datos

Diseño lógico

Esquema  
lógico

Esquemas de  
transacciones

SGBD

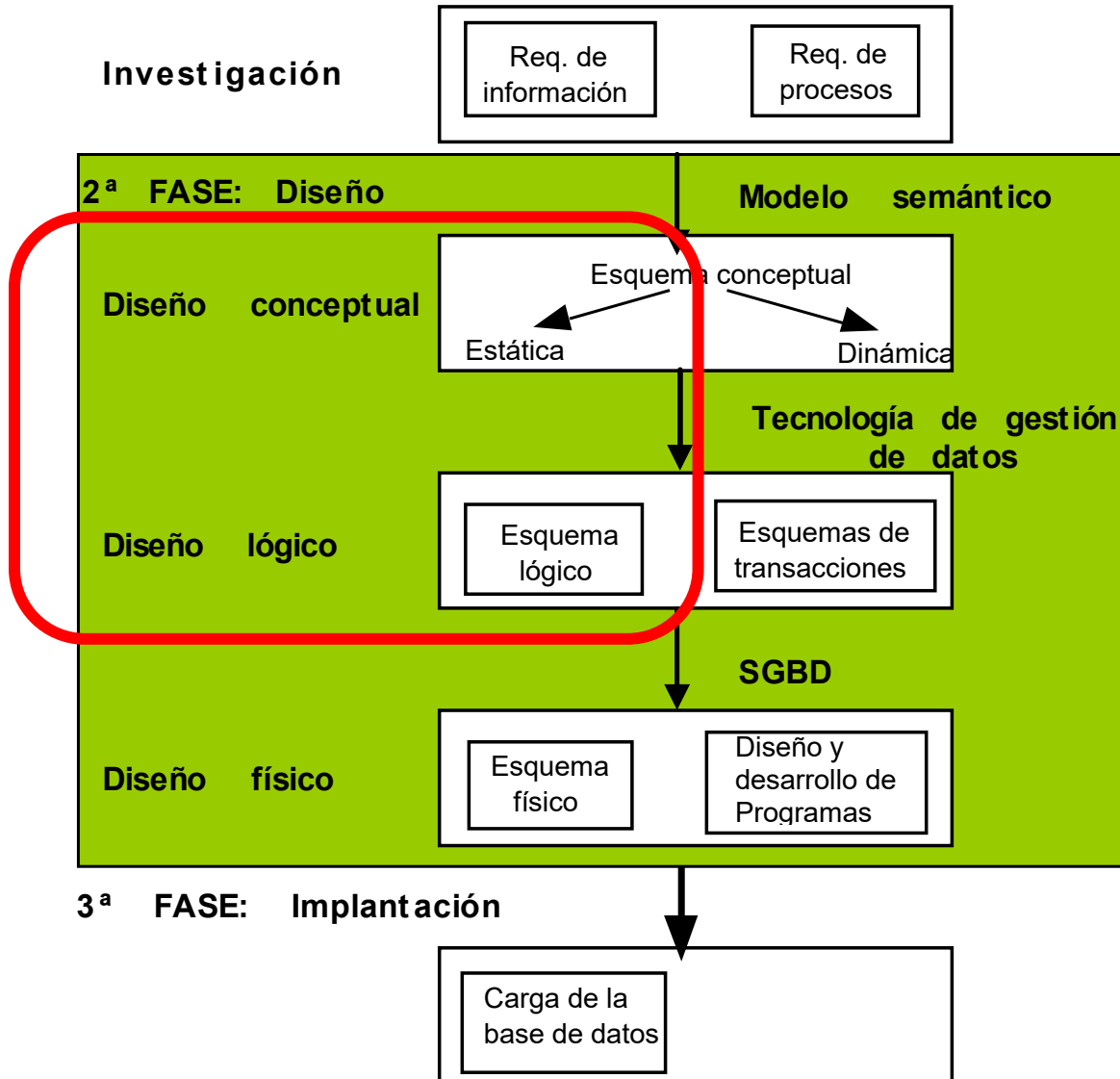
Diseño físico

Esquema  
físico

Diseño y  
desarrollo de  
Programas

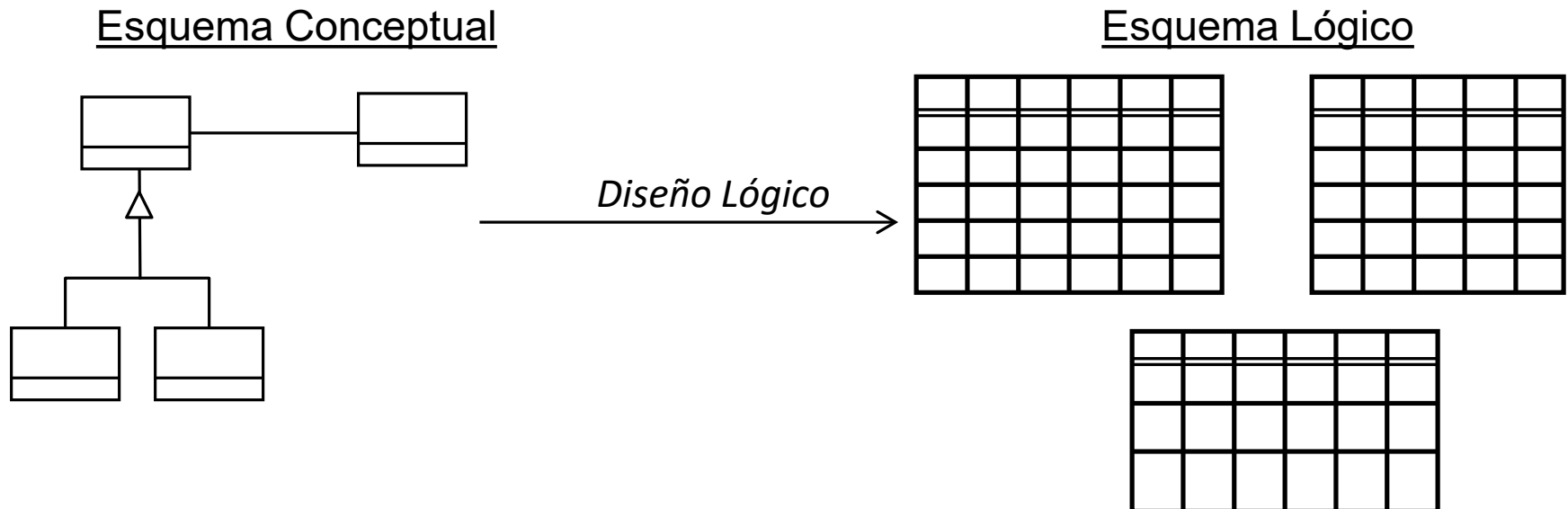
## 3ª FASE: Implantación

Carga de la  
base de datos



# 1.- Diseño lógico

**Diseño lógico:** transformación del esquema conceptual, que se encuentra descrito con un cierto modelo de datos (UML en nuestro caso), en estructuras descritas en términos del modelo de datos en el que se base el sistema de gestión de bases de datos que se vaya a utilizar.



# 1.- Diseño lógico

---

- Transformaciones: se basan en la definición de: CP, CAj, VNN, Único
- Aquellas propiedades expresadas en el diagrama que no se puedan representar en el esquema relacional deberán ser incluidas en una *lista de restricciones de integridad* para que sean controladas desde restricciones generales (*assertions*, *triggers* o programa).
- Cuando haya varios esquemas relacionales posibles:  
*“Elegir el esquema con menos restricciones de integridad añadidas. Ante igualdad de restricciones, elegir el esquema con menos relaciones”.*

# UD 4.3.- Diseño lógico

## 1.- Introducción

## 2.- Transformación de las clases

2.1.- Clases fuertes

2.2.- Clases débiles

2.3.- Clases especializadas

## 3.- Transformación de las asociaciones

3.1.- No reflexivas

3.2.- Reflexivas

3.3.- Atributos de enlace

3.4.- Transformación de la asociación cuando se asocia con clases asociación

## 4.- Ejemplo de transformación

## 5.- Teoría de la Normalización

5.1.- Conceptos previos

5.2.- Primera forma normal (1FN)

5.3.- Segunda forma normal (2FN)

5.4.- Tercera forma normal (3FN)

5.5.- Reflexión sobre la teoría de la Normalización

## 2.1.- Transformación de clases (fuertes)

Persona
DNI: {id}: char NSS: {unico <sub>1</sub> }: {1..1}: char nombre: {1..1}: propio:{1..1}: char apellidos:{1..1}: char edad: {0..1}: int teléfonos: {0..*}: char

**Persona**(DNI:char, NSS:char, nombre\_propio:char,  
    apellidos:char, edad:int,  
    CP:{DNI}  
    Único:{NSS}  
    VNN:{NSS, nombre\_propio, apellidos})

**Contacto**(DNI: char, teléfono:char)  
    CP:{DNI, teléfono}  
    CAj:{DNI}→ Persona(DNI)

## 2.1.- Transformación de clases (fuertes)

A	
$a_0$	$\{id\}: t_{a_0}$
$a_1$	$\{\text{único}_1\}: \{0..1\}: t_{a_1}$
$a_2$	$\{1..1\}: t_{a_2}$
$a_3$	$\{0..1\}: t_{a_3}$
$a_4$	$\{1..*\}: t_{a_4}$
$a_5$	$\{0..*\}: t_{a_5}$
$a_6$	$\{0..1\}: t_{a_6}$
$a_{61}$	$t_{a_{61}}$
$a_{62}$	$t_{a_{62}}$

$A(a_0:t_{a_0}, a_1:t_{a_1}, a_2:t_{a_2}, a_3:t_{a_3}, a_{61}:t_{a_{61}}, a_{62}:t_{a_{62}})$

CP: $\{a_0\}$

Único: $\{a_1\}$

VNN: $\{a_2\}$

$A4(a_0:t_{a_0}, a_4:t_{a_4})$

CP: $\{a_0, a_4\}$

CAj: $\{a_0\} \rightarrow A(a_0)$

$A5(a_0:t_{a_0}, a_5:t_{a_5})$

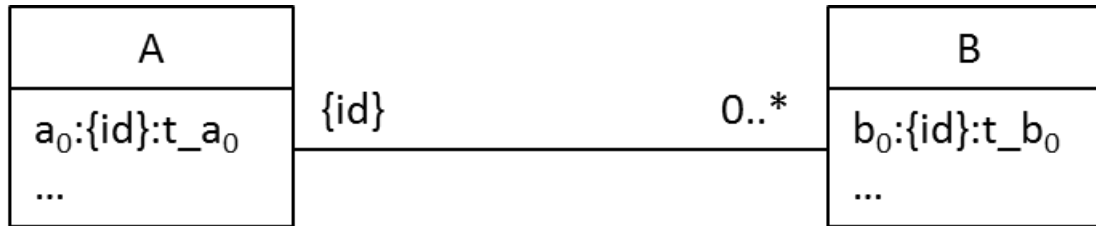
CP: $\{a_0, a_5\}$

CAj: $\{a_0\} \rightarrow A(a_0)$

**RI1:** Todo valor que aparece en el atributo  $a_0$  de  $A$  debe aparecer en el atributo  $a_0$  de  $A4$ .



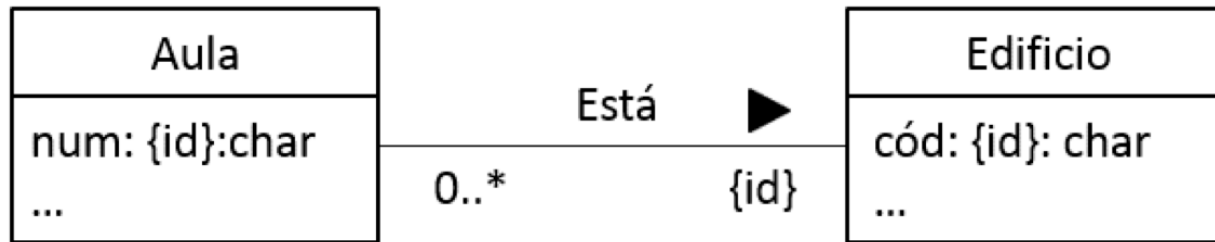
## 2.2.- Transformación de clases (débiles)



$A(a_0:t_{a_0}, \dots)$   
 $CP:\{a_0\}$

$B(b_0:t_{b_0}, a_0:t_{a_0}, \dots)$   
 $CP:\{a_0, b_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$

## 2.2.- Transformación de clases (débiles)



**Edificio**(cod:char,...)

CP:{cod}

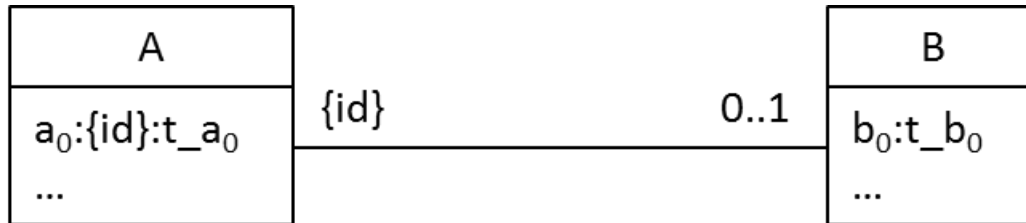
Único:{NSS}

**Aula**(cod: char, num:char)

CP:{cod, num}

CAj:{cod}→ Edificio(cod)

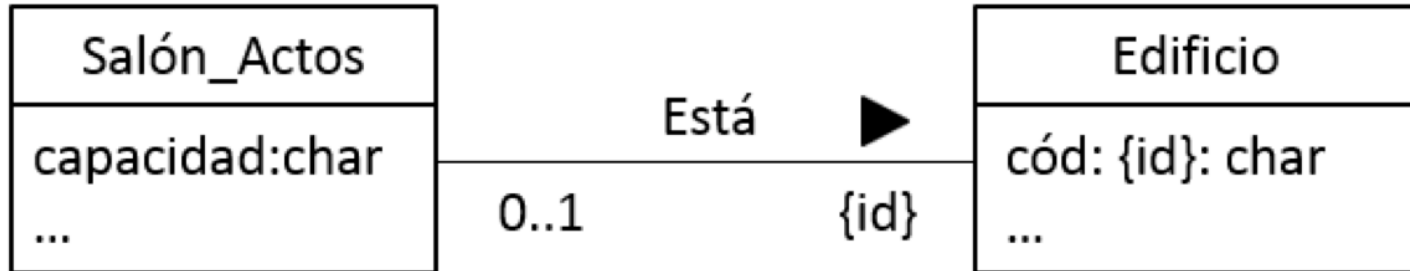
## 2.2.- Transformación de clases (débiles)



$A(a_0:t_{a_0}, \dots)$   
 $CP:\{a_0\}$

$B(b_0:t_{b_0}, a_0:t_{a_0}, \dots)$   
 $CP:\{a_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$

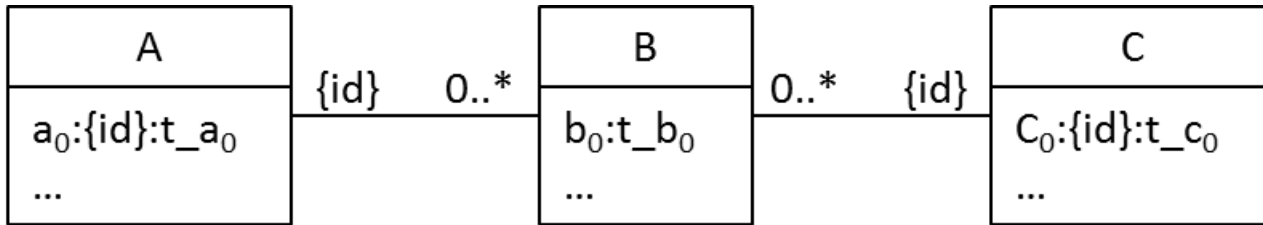
## 2.2.- Transformación de clases (débiles)



**Edificio**(cod:char,...)  
CP:{cod}

**Salón\_Actos**(cod:char, capacidad:char, ...)  
CP:{cod}  
CAj:{cod} → Edificio(cod)

## 2.2.- Transformación de clases (débiles)

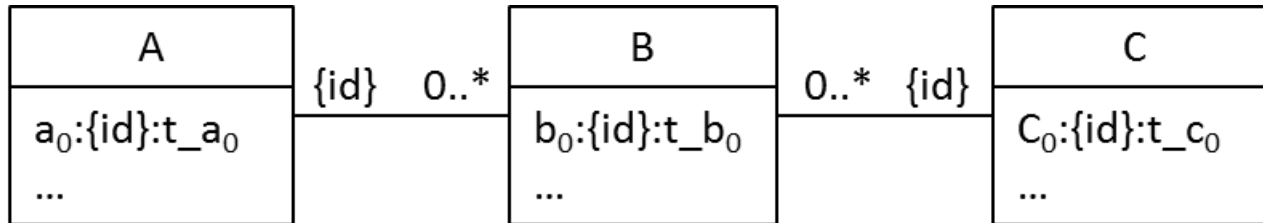


$A(a_0:t_{a_0}, \dots)$   
CP:{a<sub>0</sub>}

$C(c_0:t_{c_0}, \dots)$   
CP:{c<sub>0</sub>}

$B(a_0:t_{a_0}, c_0:t_{c_0}, b_0:t_{b_0}, \dots)$   
CP:{a<sub>0</sub>, c<sub>0</sub>}  
CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)  
CAj:{c<sub>0</sub>} → C(c<sub>0</sub>)

## 2.2.- Transformación de clases (débiles)

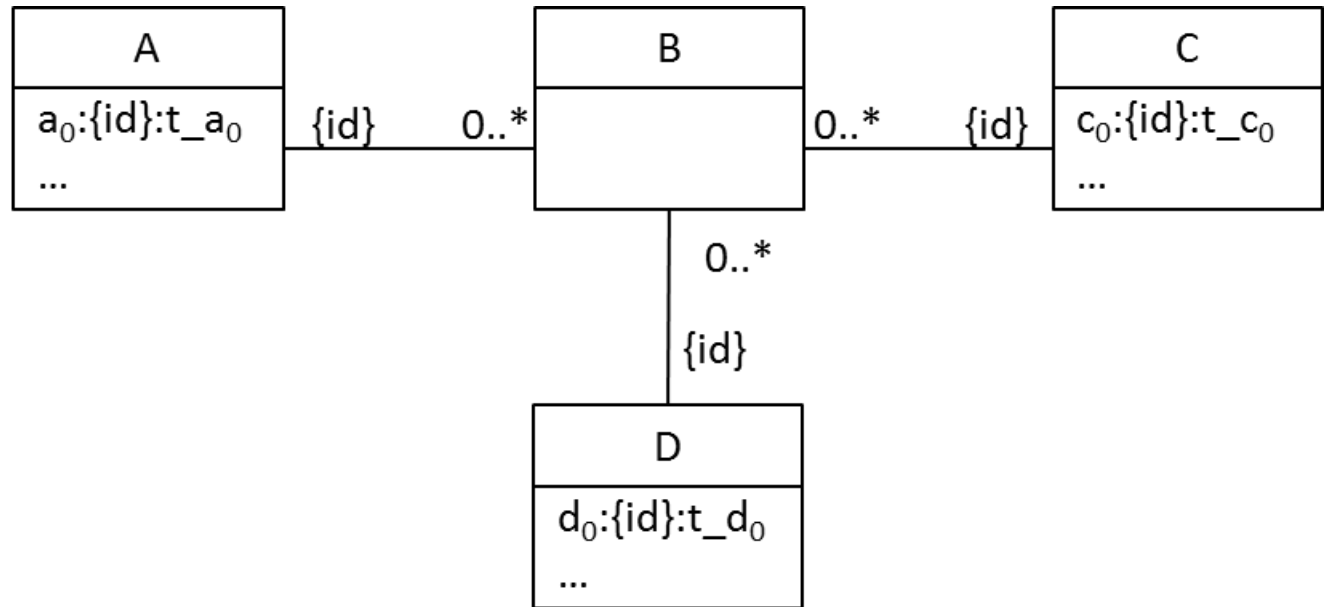


$A(a_0:t_{a_0}, \dots)$   
CP:{a<sub>0</sub>}

$C(c_0:t_{c_0}, \dots)$   
CP:{c<sub>0</sub>}

$B(a_0:t_{a_0}, c_0:t_{c_0}, b_0:t_{b_0}, \dots)$   
CP:{a<sub>0</sub>, c<sub>0</sub>, b<sub>0</sub>}  
CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)  
CAj:{c<sub>0</sub>} → C(c<sub>0</sub>)

## 2.2.- Transformación de clases (débiles)



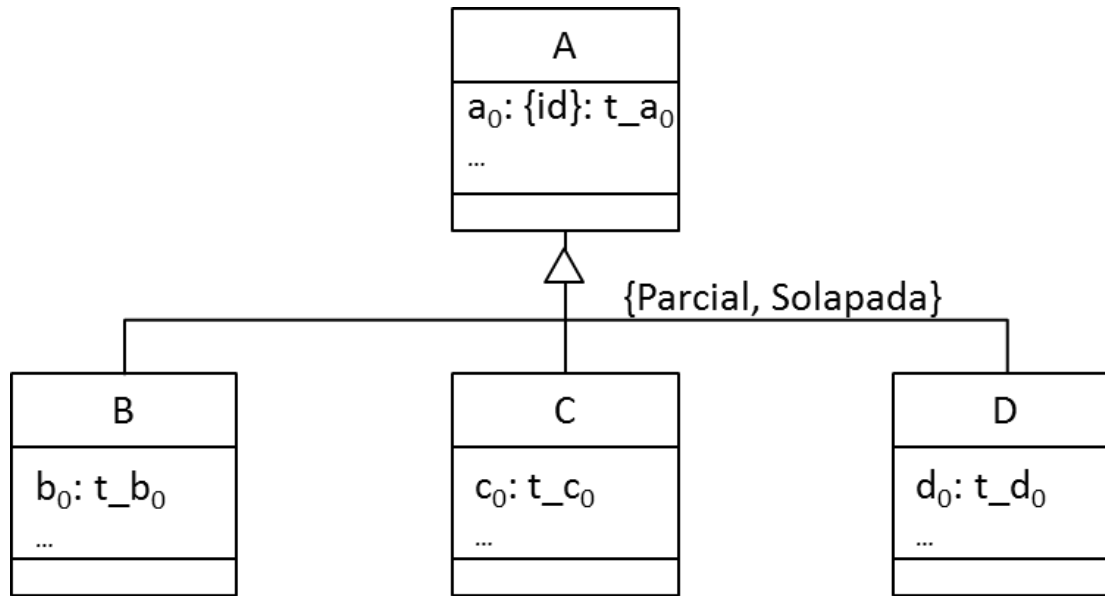
$A(a_0:t_{a_0}, \dots)$   
 $CP:\{a_0\}$

$C(c_0:t_{c_0}, \dots)$   
 $CP:\{c_0\}$

$D(d_0:t_{d_0}, \dots)$   
 $CP:\{d_0\}$

$B(a_0:t_{a_0}, c_0:t_{c_0}, d_0:t_{d_0})$   
 $CP:\{a_0, c_0, d_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$   
 $CAj:\{c_0\} \rightarrow C(c_0)$   
 $CAj:\{d_0\} \rightarrow D(d_0)$

## 2.3.- Transformación de clases (Gen/Esp)



$A(a_0:t_{a_0}, \dots)$   
 $CP:\{a_0\}$

$C(a_0:t_{a_0}, c_0:t_{c_0}, \dots)$   
 $CP:\{a_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$

$B(a_0:t_{a_0}, b_0:t_{b_0}, \dots)$   
 $CP:\{a_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$

$D(a_0:t_{a_0}, d_0:t_{d_0}, \dots)$   
 $CP:\{a_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$



## 2.3.- Transformación de clases (Gen/Esp)

$RI_{Total}$ : Todo valor que aparece en el atributo  $a_0$  de  $A$  debe aparecer en el atributo  $a_0$  de  $B$ , de  $C$  o de  $D$ .

$RI_{Disjunta}$ : No puede haber un mismo valor en el atributo  $a_0$  de  $B$  y en el  $a_0$  de  $C$ ; ni en el  $a_0$  de  $B$  y en el  $a_0$  de  $D$ ; ni en el atributo  $a_0$  de  $C$  y en el  $a_0$  de  $D$ .

$A(a_0:t_{a_0},...)$   
 $CP:\{a_0\}$

$C(a_0:t_{a_0},c_0:t_{c_0},...)$   
 $CP:\{a_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$

$B(a_0:t_{a_0},b_0:t_{b_0},...)$   
 $CP:\{a_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$

$D(a_0:t_{a_0},d_0:t_{d_0},...)$   
 $CP:\{a_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$

# UD 4.3.- Diseño lógico

## 1.- Introducción

## 2.- Transformación de las clases

2.1.- Clases fuertes

2.2.- Clases débiles

2.3.- Clases especializadas

## 3.- Transformación de las asociaciones

3.1.- No reflexivas

3.2.- Reflexivas

3.3.- Atributos de enlace

3.4.- Transformación de la asociación cuando se asocia con clases asociación

## 4.- Ejemplo de transformación

## 5.- Teoría de la Normalización

5.1.- Conceptos previos

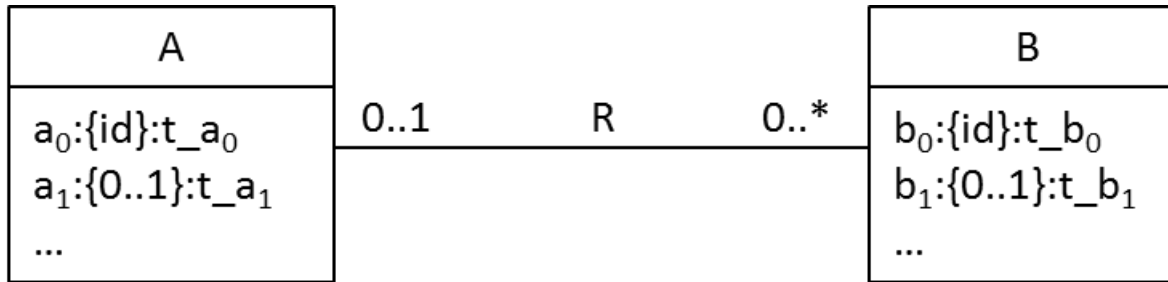
5.2.- Primera forma normal (1FN)

5.3.- Segunda forma normal (2FN)

5.4.- Tercera forma normal (3FN)

5.5.- Reflexión sobre la teoría de la Normalización

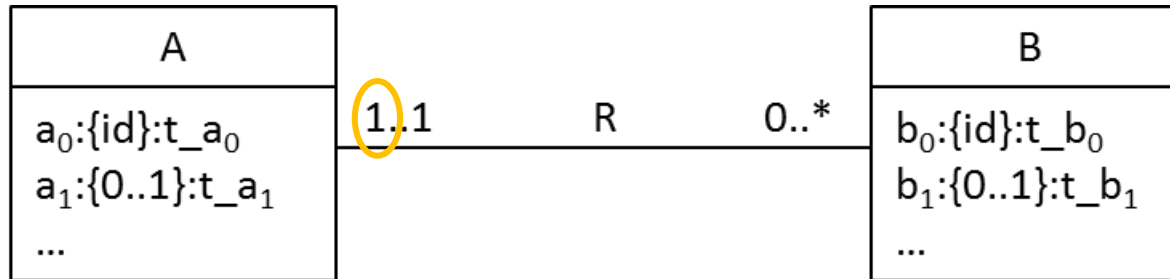
## 3.1.- Transf. de asociaciones binarias 0..1:0..\*



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
CP:{a<sub>0</sub>}

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, \mathbf{a_0:t_{a_0}})$   
CP:{b<sub>0</sub>}  
 $\mathbf{CAj:\{a_0\} \rightarrow A(a_0)}$

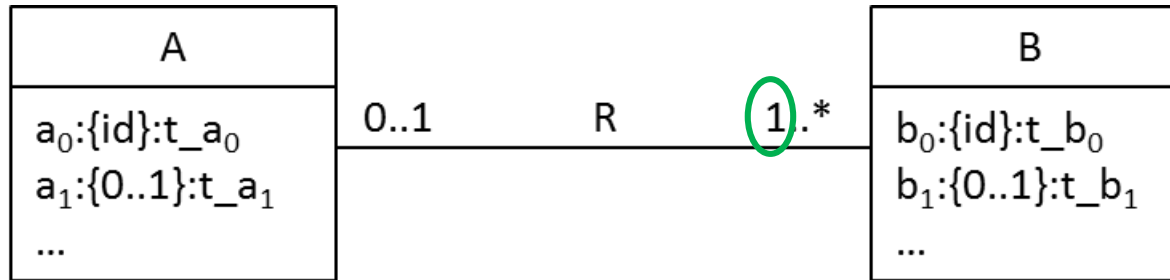
## 3.1.- Transf. de asociaciones binarias 1..1:0..\*



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
CP:{a<sub>0</sub>}

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0})$   
CP:{b<sub>0</sub>}  
CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)  
VNN:{a<sub>0</sub>}

## 3.1.- Transf. de asociaciones binarias 0..1:1..\*

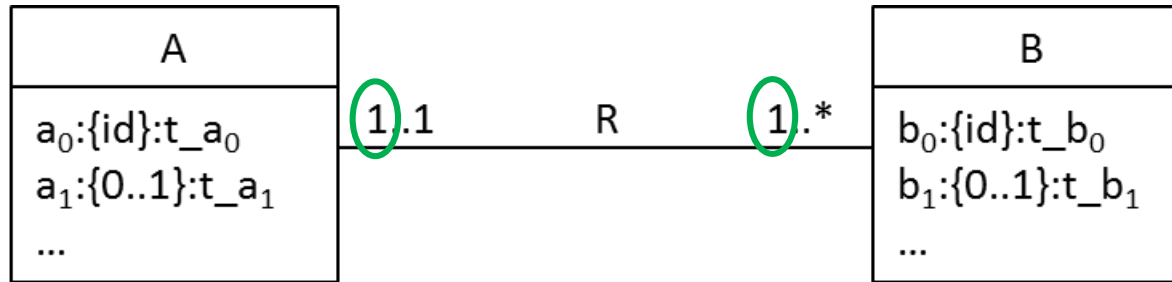


$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
CP: $\{a_0\}$

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0})$   
CP: $\{b_0\}$   
CAj: $\{a_0\} \rightarrow A(a_0)$

**RI1:** Todo valor que aparece en el atributo  $a_0$  de  $A$  debe aparecer en el atributo  $a_0$  de  $B$ .

## 3.1.- Transf. de asociaciones binarias 1..1:1..\*

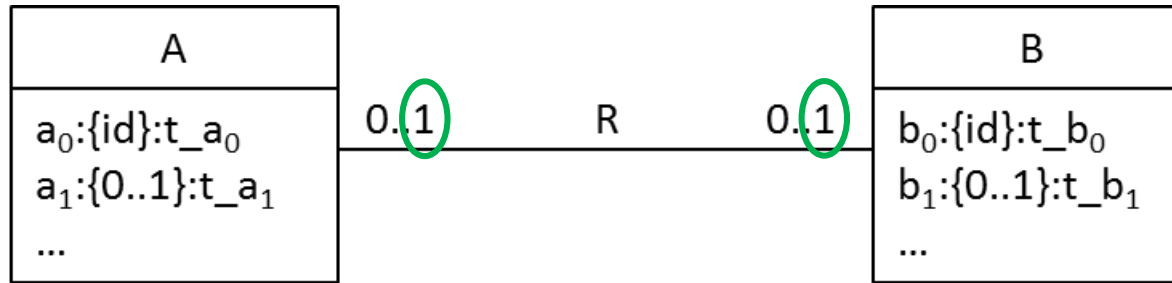


$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
CP: $\{a_0\}$

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0})$   
CP: $\{b_0\}$   
CAj: $\{a_0\} \rightarrow A(a_0)$   
VNN: $\{a_0\}$

**RI1:** Todo valor que aparece en el atributo  $a_0$  de  $A$  debe aparecer en el atributo  $a_0$  de  $B$ .

## 3.1.- Transf. de asociaciones binarias 0..1:0..1



### Esquema 1

$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
CP: $\{a_0\}$

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0})$   
CP: $\{b_0\}$   
Único: $\{a_0\}$   
CAj: $\{a_0\} \rightarrow A(a_0)$

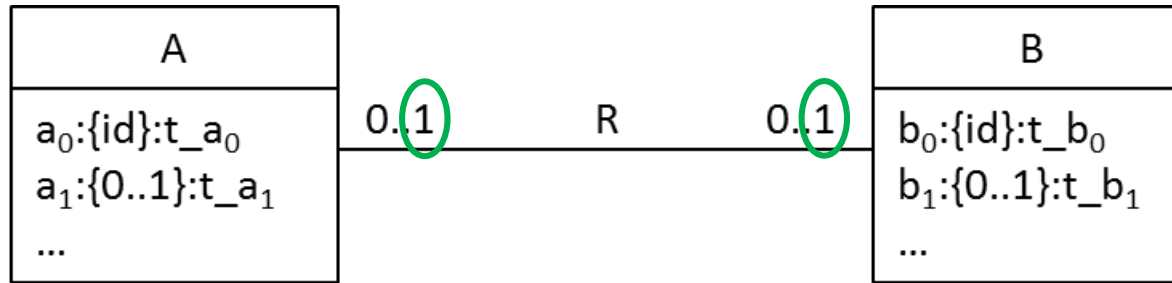
### Esquema 2

$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots, b_0:t_{b_0})$   
CP: $\{a_0\}$

Único: $\{b_0\}$   
CAj: $\{b_0\} \rightarrow B(b_0)$

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$   
CP: $\{b_0\}$

# 3.1.- Transf. de asociaciones binarias 0..1:0..1



Esquema 3

$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
 CP:{a<sub>0</sub>}

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$   
 CP:{b<sub>0</sub>}

$R(b_0:t_{b_0}, a_0:t_{a_0})$   
 CP:{b<sub>0</sub>}  
 Único:{a<sub>0</sub>}  
 VNN:{a<sub>0</sub>}  
 CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)  
 CAj:{b<sub>0</sub>} → B(b<sub>0</sub>)

Esquema 4

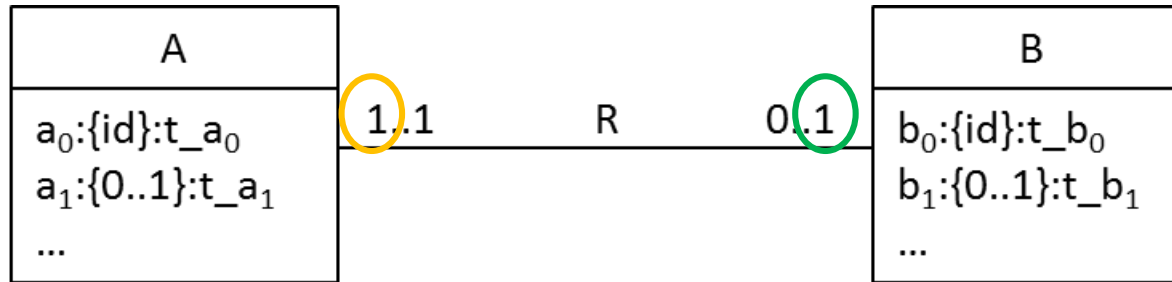
$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
 CP:{a<sub>0</sub>}

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$   
 CP:{b<sub>0</sub>}

$R(b_0:t_{b_0}, a_0:t_{a_0})$   
 CP:{a<sub>0</sub>}  
 Único:{b<sub>0</sub>}  
 VNN:{b<sub>0</sub>}  
 CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)  
 CAj:{b<sub>0</sub>} → B(b<sub>0</sub>)



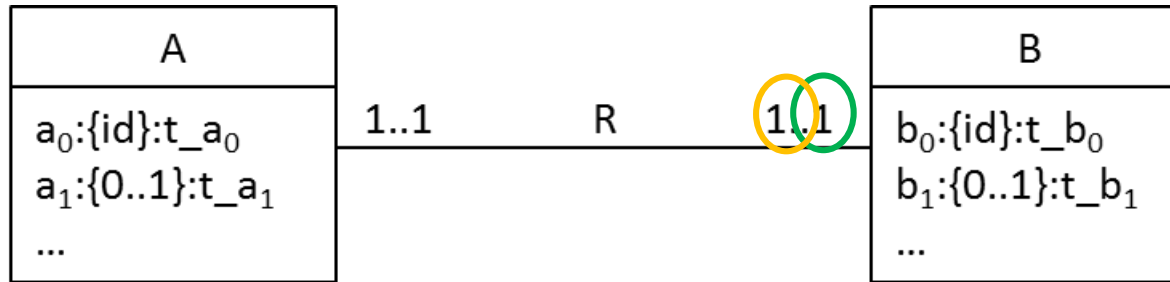
## 3.1.- Transf. de asociaciones binarias 0..1:1..1



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
CP: $\{a_0\}$

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0})$   
CP: $\{b_0\}$   
Único: $\{a_0\}$   
VNN: $\{a_0\}$   
CAj: $\{a_0\} \rightarrow A(a_0)$

## 3.1.- Transf. de asociaciones binarias 1..1:1..1



### Esquema 1

A-B( $a_0:t_{a_0}, a_1:t_{a_1}, \dots, b_0:t_{b_0}, b_1:t_{b_1}, \dots$ )

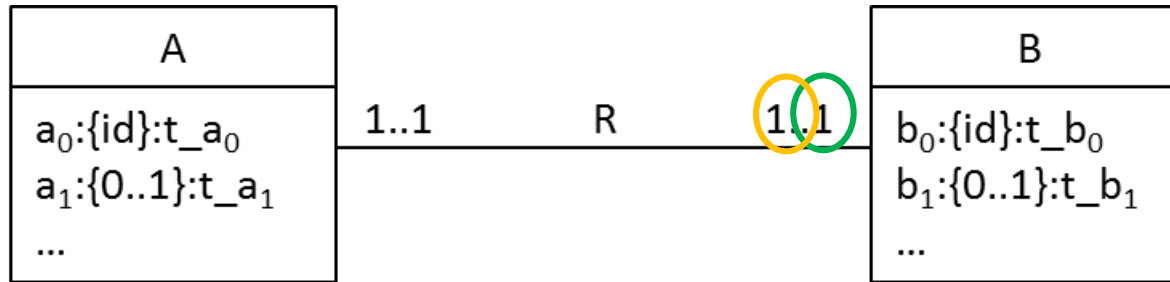
CP: $\{a_0\}$

Único: $\{b_0\}$

VNN: $\{b_0\}$

$b_0$  es una clave alternativa

# 3.1.- Transf. de asociaciones binarias 1..1:1..1



## Esquema 2

$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots, b_0:t_{b_0})$

CP: $\{a_0\}$

Único: $\{b_0\}$

VNN: $\{b_0\}$

CAj: $\{b_0\} \rightarrow B(b_0)$

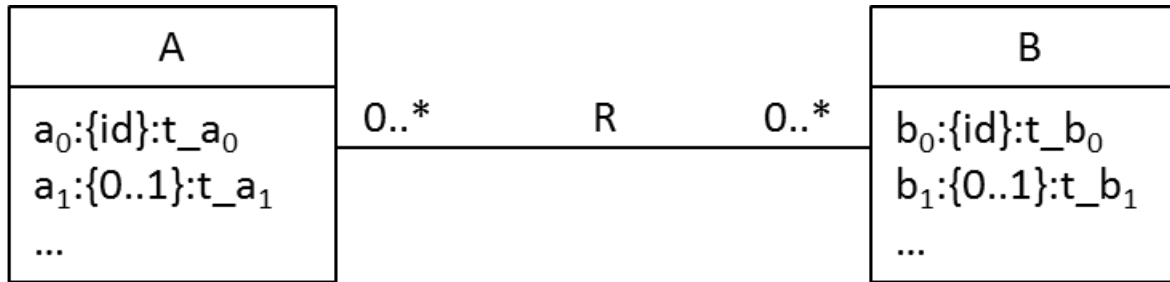
$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$

CP: $\{b_0\}$

CAj: $\{b_0\} \rightarrow A(b_0)$

Esto es posible porque  $b_0$  tiene unicidad en A

## 3.1.- Transf. de asociaciones binarias 0..\*:0..\*

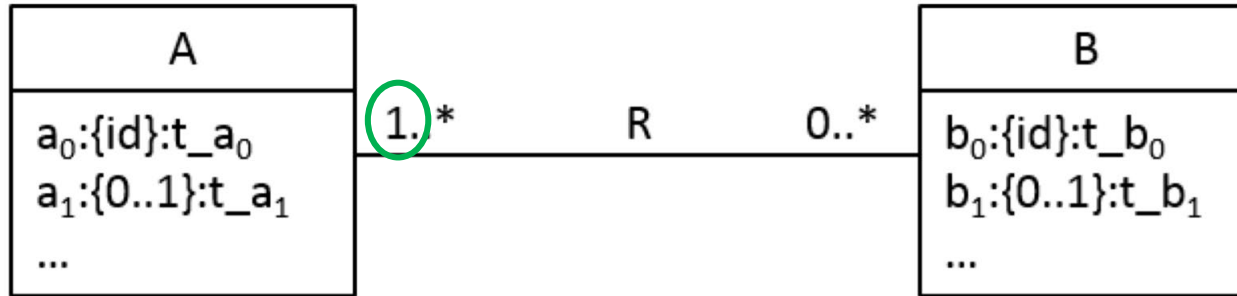


$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
CP:{a<sub>0</sub>}

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$   
CP:{b<sub>0</sub>}

$R(a_0:t_{a_0}, b_0:t_{b_0})$   
CP:{a<sub>0</sub>, b<sub>0</sub>}  
CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)  
CAj:{b<sub>0</sub>} → B(b<sub>0</sub>)

## 3.1.- Transf. de asociaciones binarias 1..\*:0..\*



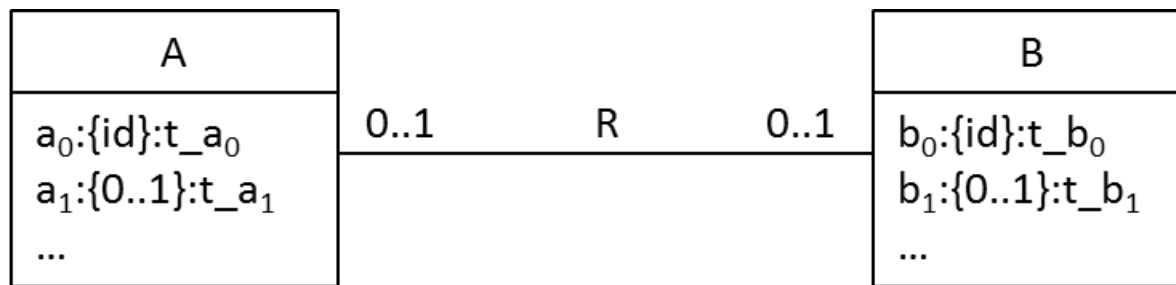
$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
CP:  $\{a_0\}$

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$   
CP:  $\{b_0\}$

$R(a_0:t_{a_0}, b_0:t_{b_0})$   
CP:  $\{a_0, b_0\}$   
 $CAj:\{a_0\} \rightarrow A(a_0)$   
 $CAj:\{b_0\} \rightarrow B(b_0)$

**RI1:** Todo valor que aparece en el atributo  $b_0$  de B debe aparecer en el atributo  $b_0$  de R

## 3.2.- Transf. de asociaciones reflexivas 0..1:0..1



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$

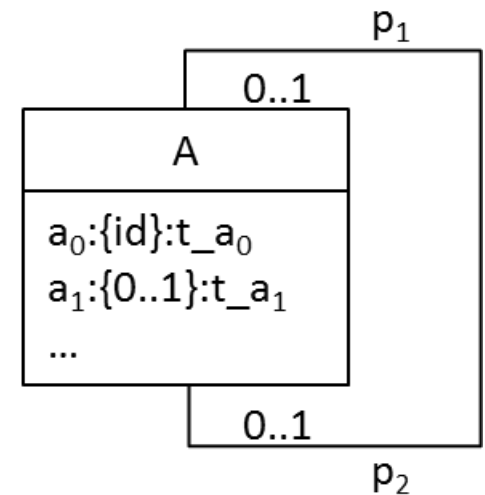
CP:{a<sub>0</sub>}

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0})$

CP:{b<sub>0</sub>}

Único:{a<sub>0</sub>}

CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)



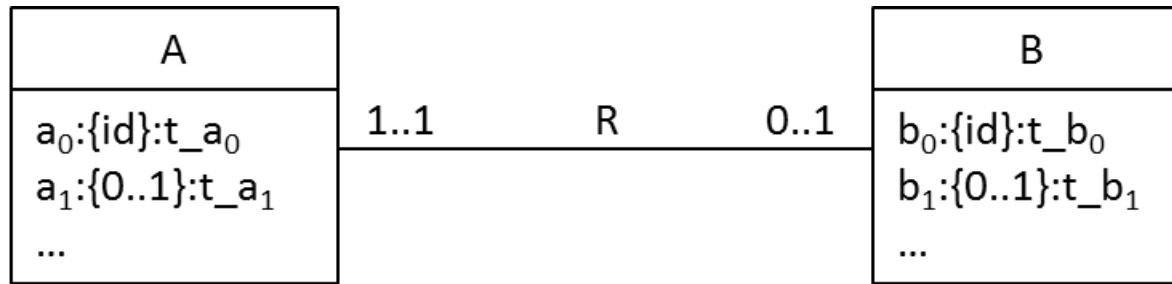
$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots, a_{0\_p_2}:t_{a_0})$

CP:{a<sub>0</sub>}

Único:{a<sub>0\_p2</sub>}

CAj:{a<sub>0\_p2</sub>} → A(a<sub>0</sub>)

## 3.2.- Transf. de asociaciones reflexivas 0..1:1..1



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$

CP:{a<sub>0</sub>}

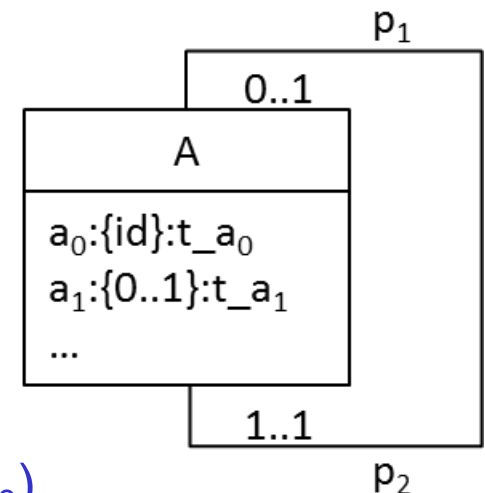
$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0})$

CP:{b<sub>0</sub>}

Único:{a<sub>0</sub>}

VNN:{a<sub>0</sub>}

CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots, a_{0\_p2}:t_{a_0})$

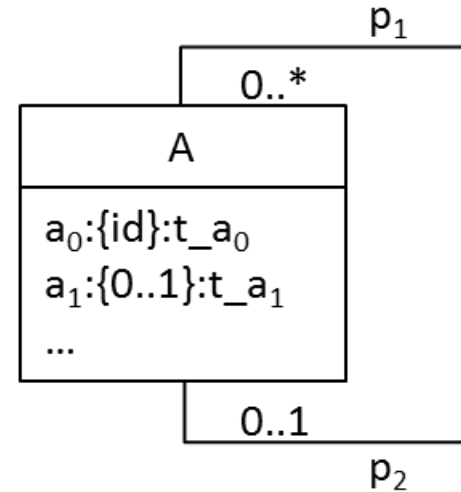
CP:{a<sub>0</sub>}

Único:{a<sub>0\_p2</sub>}

VNN:{a<sub>0\_p2</sub>}

CAj:{a<sub>0\_p2</sub>} → A(a<sub>0</sub>)

## 3.2.- Transf. de asociaciones reflexivas 0..\*:0..1



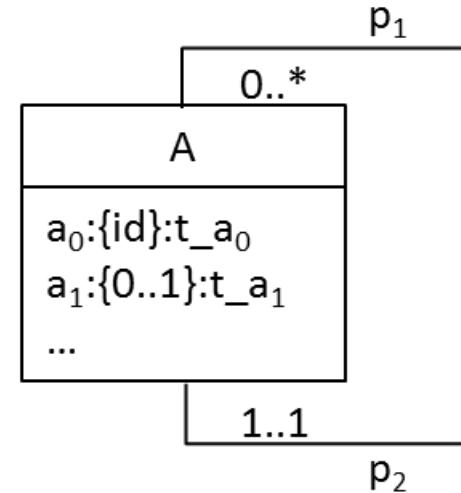
$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots, a_{p_2}:t_{a_0})$

$CP:\{a_0\}$

$CAj:\{a_{p_2}\} \rightarrow A(a_0)$



## 3.2.- Transf. de asociaciones reflexivas 0..\*:1..1



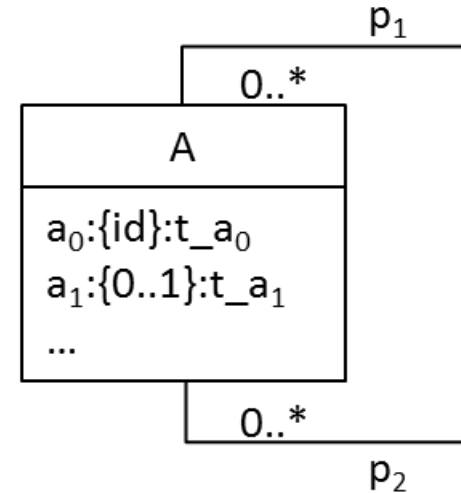
$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots, a_{p_2}:t_{a_0})$

$CP:\{a_0\}$

$CAj:\{a_{p_2}\} \rightarrow A(a_0)$

$VNN:\{a_{p_2}\}$

## 3.2.- Transf. de asociaciones reflexivas 0..\*:0..\*



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots, a_{p_2}:t_{a_0})$

CP:{ $a_0$ }

$R(a_{p_1}:t_{a_0}, a_{p_2}:t_{a_0})$

CP:{ $a_{p_1}, a_{p_2}$ }

$CA_j:\{a_{p_1}\} \rightarrow A(a_0)$

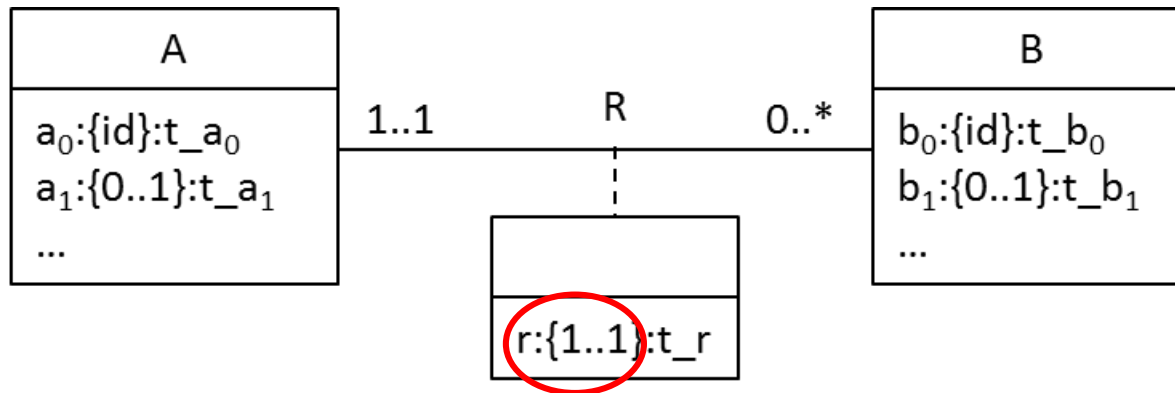
$CA_j:\{a_{p_2}\} \rightarrow A(a_0)$

## 3.3.- Asociaciones con atributos de enlace

---

- Los atributos de enlace se incluyen en la tabla donde está representada la asociación que describen.
- La presencia de atributos de enlace puede hacer que el esquema aplicado deje de ser adecuado:
  - Añadir Restricciones de Integridad (R.I.)
  - Añadir nueva tabla

## 3.3.- Asociaciones con atributos de enlace



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$

CP:{ $a_0$ }

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0}, r:t_r)$

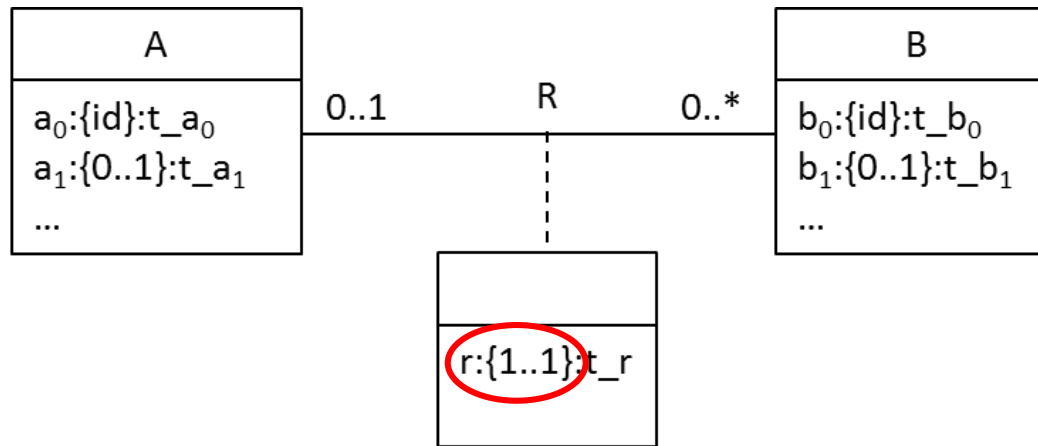
CP:{ $b_0$ }

VNN:{ $a_0$ }

VNN:{ $r$ }

CAj:{ $a_0$ }  $\rightarrow A$

## 3.3.- Asociaciones con atributos de enlace



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$

CP: $\{a_0\}$

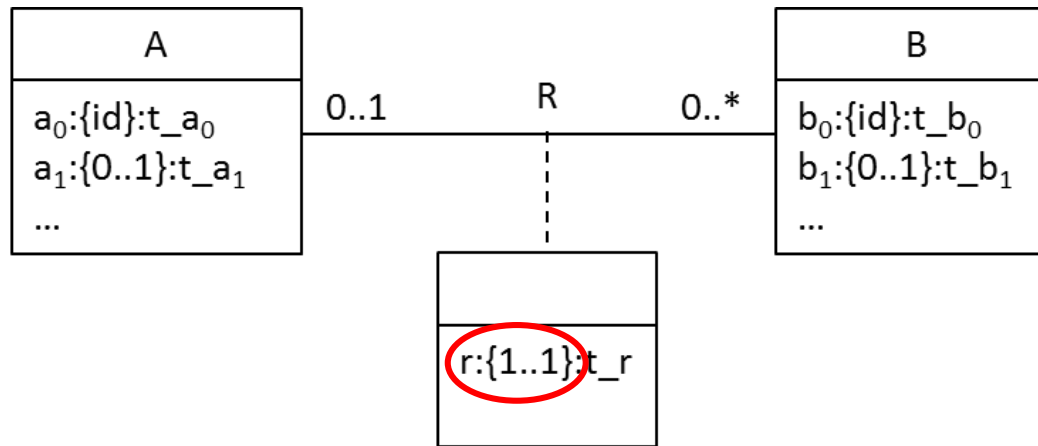
$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0}, r:t_r)$

CP: $\{b_0\}$

CAj: $\{a_0\} \rightarrow A(a_0)$

**RI1:** En toda tupla de  $B$  se debe cumplir que o bien  $a_0$  y  $r$  son no nulo o  $a_0$  y  $r$  son nulos.

## 3.3.- Asociaciones con atributos de enlace



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$

CP:{ $a_0$ }

$R(b_0:t_{b_0}, a_0:t_{a_0}, r:t_r)$

CP:{ $b_0$ }

VNN:{ $a_0$ }

VNN:{ $r$ }

CAj:{ $a_0$ }  $\rightarrow$  A ( $a_0$ )

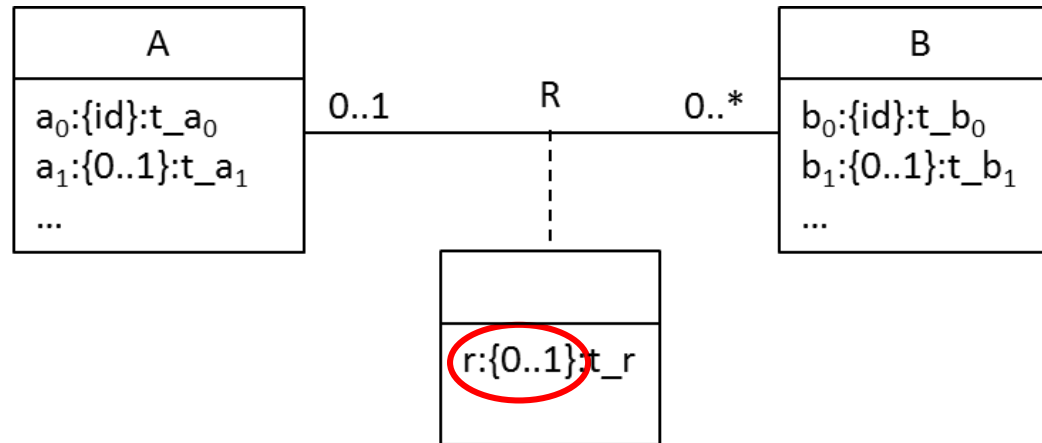
CAj:{ $b_0$ }  $\rightarrow$  B ( $b_0$ )

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$

CP:{ $b_0$ }

**Solución mejor  $\rightarrow$  no incluye RI**

## 3.3.- Asociaciones con atributos de enlace



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$

CP: $\{a_0\}$

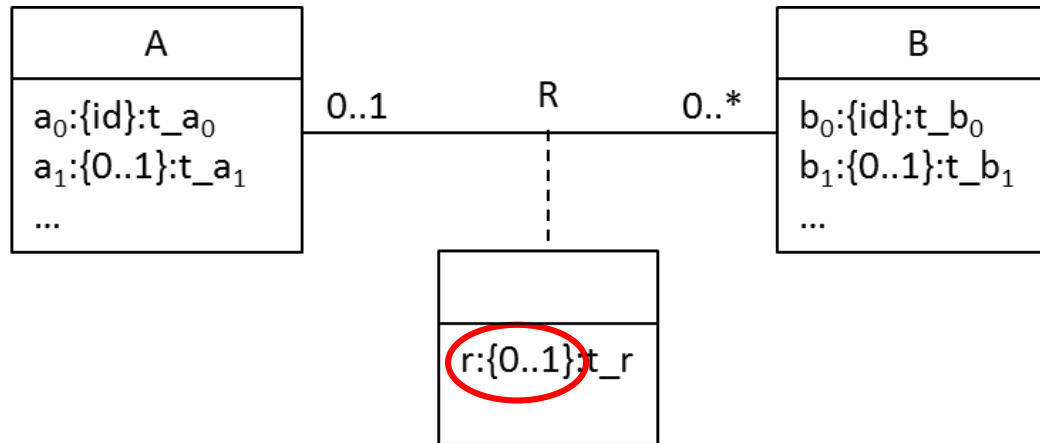
$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0}, r:t_r)$

CP: $\{b_0\}$

CAj: $\{a_0\} \rightarrow A(a_0)$

**RI1:** No puede existir una tupla en  $B$  que tenga el atributo  $a_0$  nulo y el atributo  $r$  distinto de nulo

## 3.3.- Asociaciones con atributos de enlace



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$

CP:{a<sub>0</sub>}

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$

CP:{b<sub>0</sub>}

$R(b_0:t_{b_0}, a_0:t_{a_0}, r:t_r)$

CP:{b<sub>0</sub>}

VNN:{a<sub>0</sub>}

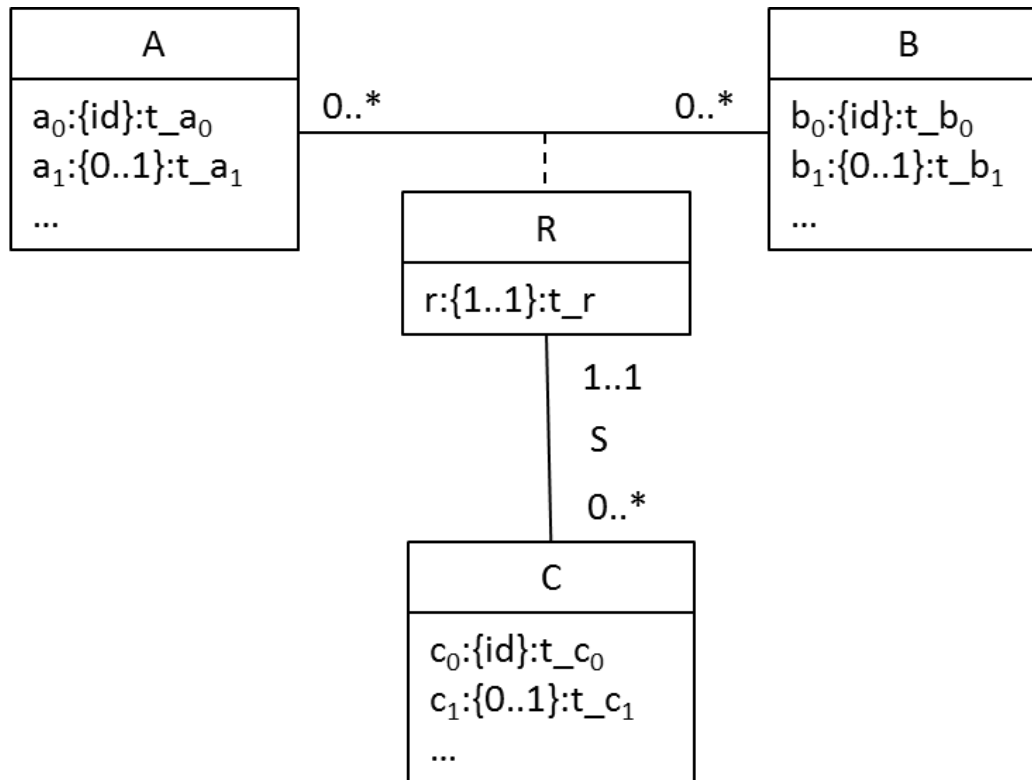
CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)

CAj:{b<sub>0</sub>} → B(b<sub>0</sub>)

**Solución mejor → no incluye RI**



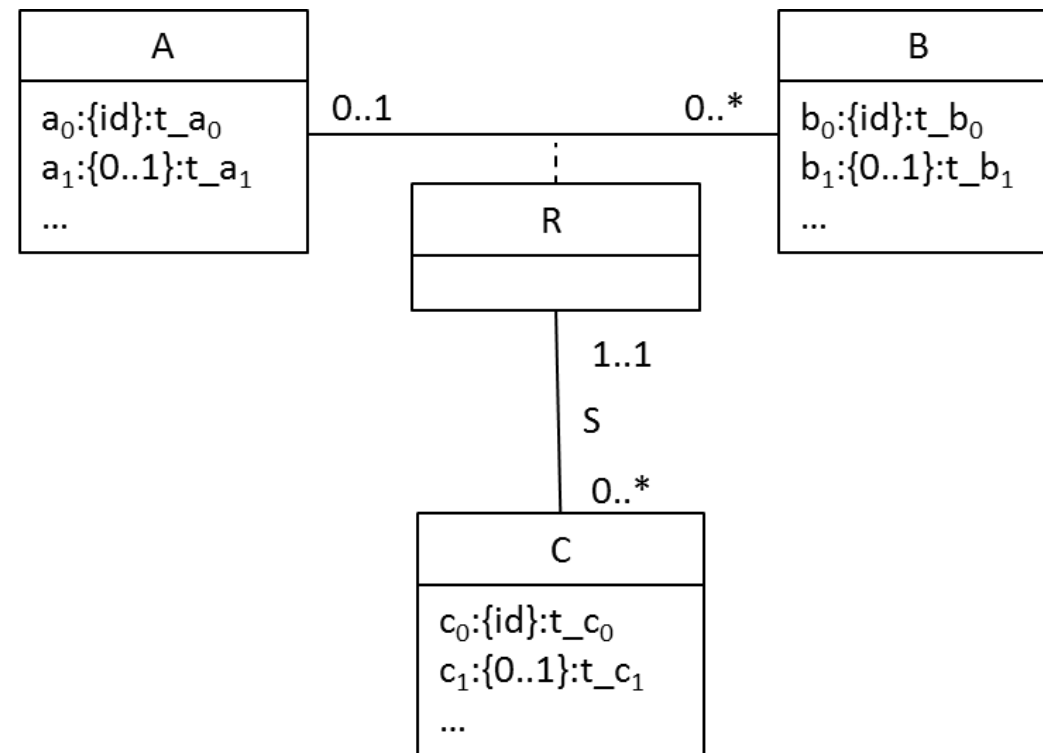
## 3.4.- Transf. Asociación (cuando se asocia con otras clases)



$C(c_0:t_{c_0}, c_1:t_{c_1}, \dots, a_0:t_{a_0}, b_0:t_{b_0})$   
 CP:{c<sub>0</sub>}  
 CAj:{a<sub>0</sub>, b<sub>0</sub>} → R(a<sub>0</sub>, b<sub>0</sub>)  
 VNN:{a<sub>0</sub>, b<sub>0</sub>}

$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$   
 CP:{a<sub>0</sub>}  
 $B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$   
 CP:{b<sub>0</sub>}  
 $R(a_0:t_{a_0}, b_0:t_{b_0}, r:t_r)$   
 CP:{a<sub>0</sub>, b<sub>0</sub>}  
 CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)  
 CAj:{b<sub>0</sub>} → B(b<sub>0</sub>)

## 3.4.- Transf. Asociación (cuando se asocia con otras clases)



$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$

CP:{a<sub>0</sub>}

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots, a_0:t_{a_0})$

CP:{b<sub>0</sub>}

CAj:{a<sub>0</sub>}→A

$C(c_0:t_{c_0}, c_1:t_{c_1}, \dots, b_0:t_{b_0})$

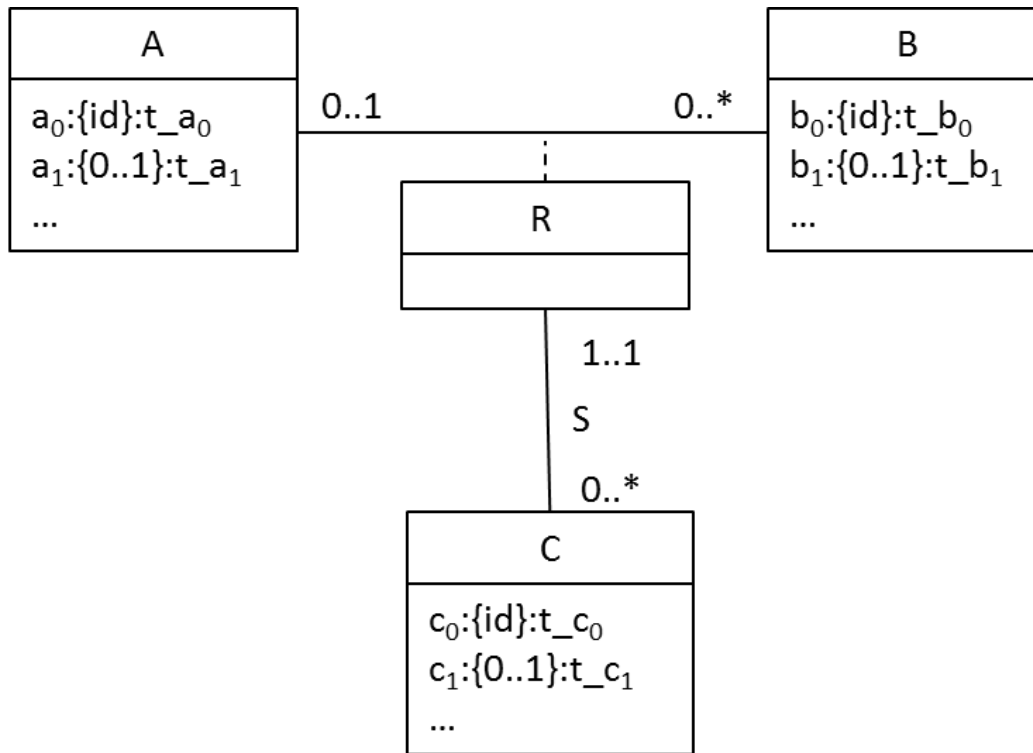
CP:{c<sub>0</sub>}

CAj:{b<sub>0</sub>}→B

VNN:{b<sub>0</sub>}

**RI1:** No puede existir una tupla en C con un valor en  $b_0$  que aparezca en B con un  $a_0$  nulo (que no se relaciones con A).

## 3.4.- Transf. Asociación (cuando se asocia con otras clases)



$C(c_0:t_{c_0}, c_1:t_{c_1}, \dots, b_0:t_{b_0})$   
 CP:{c<sub>0</sub>}  
 CAj:{b<sub>0</sub>} → R  
 VNN:{b<sub>0</sub>}

$A(a_0:t_{a_0}, a_1:t_{a_1}, \dots)$

CP:{a<sub>0</sub>}

$B(b_0:t_{b_0}, b_1:t_{b_1}, \dots)$

CP:{b<sub>0</sub>}

$R(b_0:t_{b_0}, a_0:t_{a_0})$

CP:{b<sub>0</sub>}

VNN:{a<sub>0</sub>}

CAj:{a<sub>0</sub>} → A(a<sub>0</sub>)

CAj:{b<sub>0</sub>} → B(b<sub>0</sub>)

**Solución mejor → no incluye RI**

# UD 4.3.- Diseño lógico

---

## 1.- Introducción

## 2.- Transformación de las clases

2.1.- Clases fuertes

2.2.- Clases débiles

2.3.- Clases especializadas

## 3.- Transformación de las asociaciones

3.1.- No reflexivas

3.2.- Reflexivas

3.3.- Atributos de enlace

3.4.- Transformación de la asociación cuando se asocia con clases asociación

## 4.- Ejemplo de transformación

## 5.- Teoría de la Normalización

5.1.- Conceptos previos

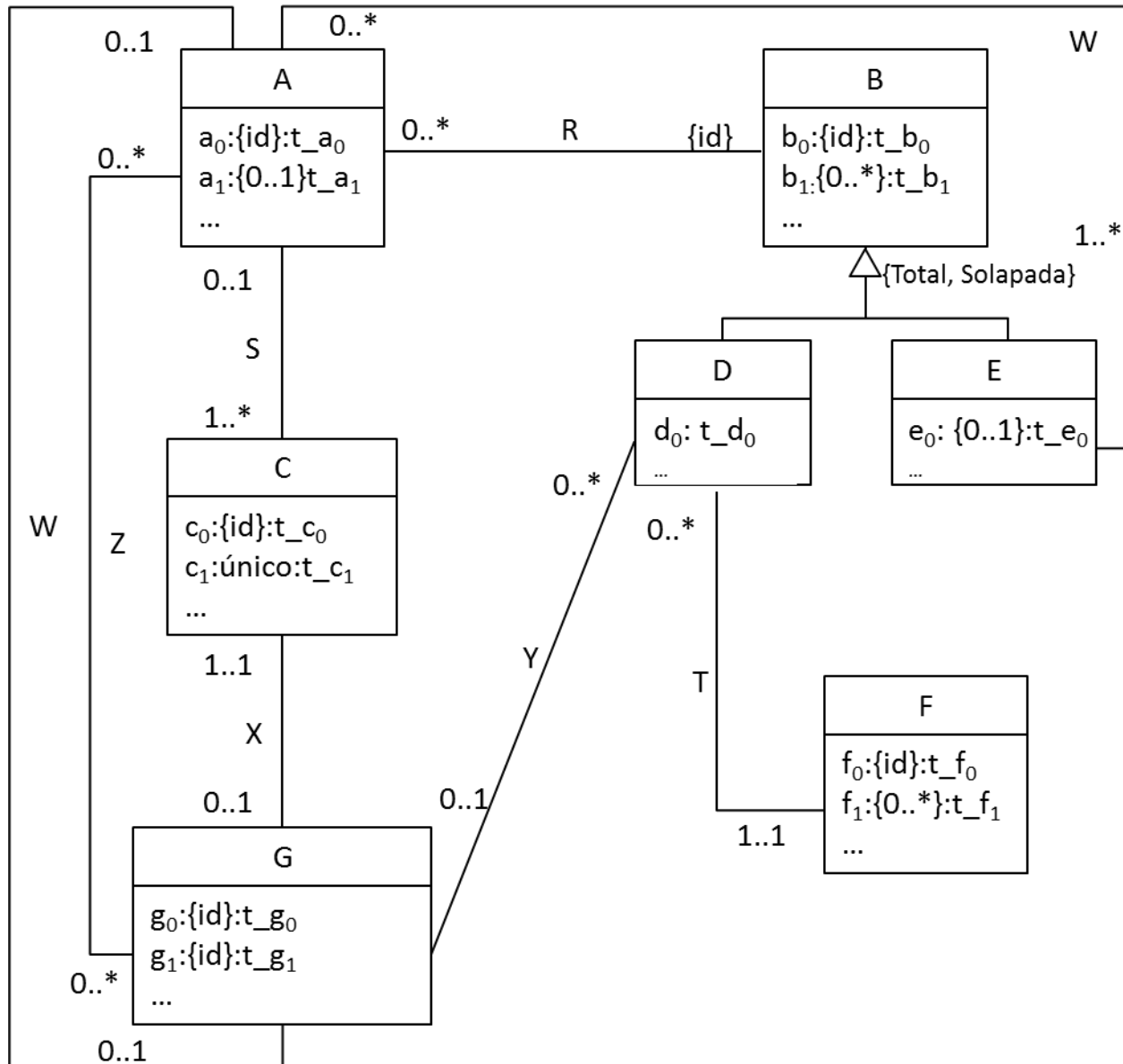
5.2.- Primera forma normal (1FN)

5.3.- Segunda forma normal (2FN)

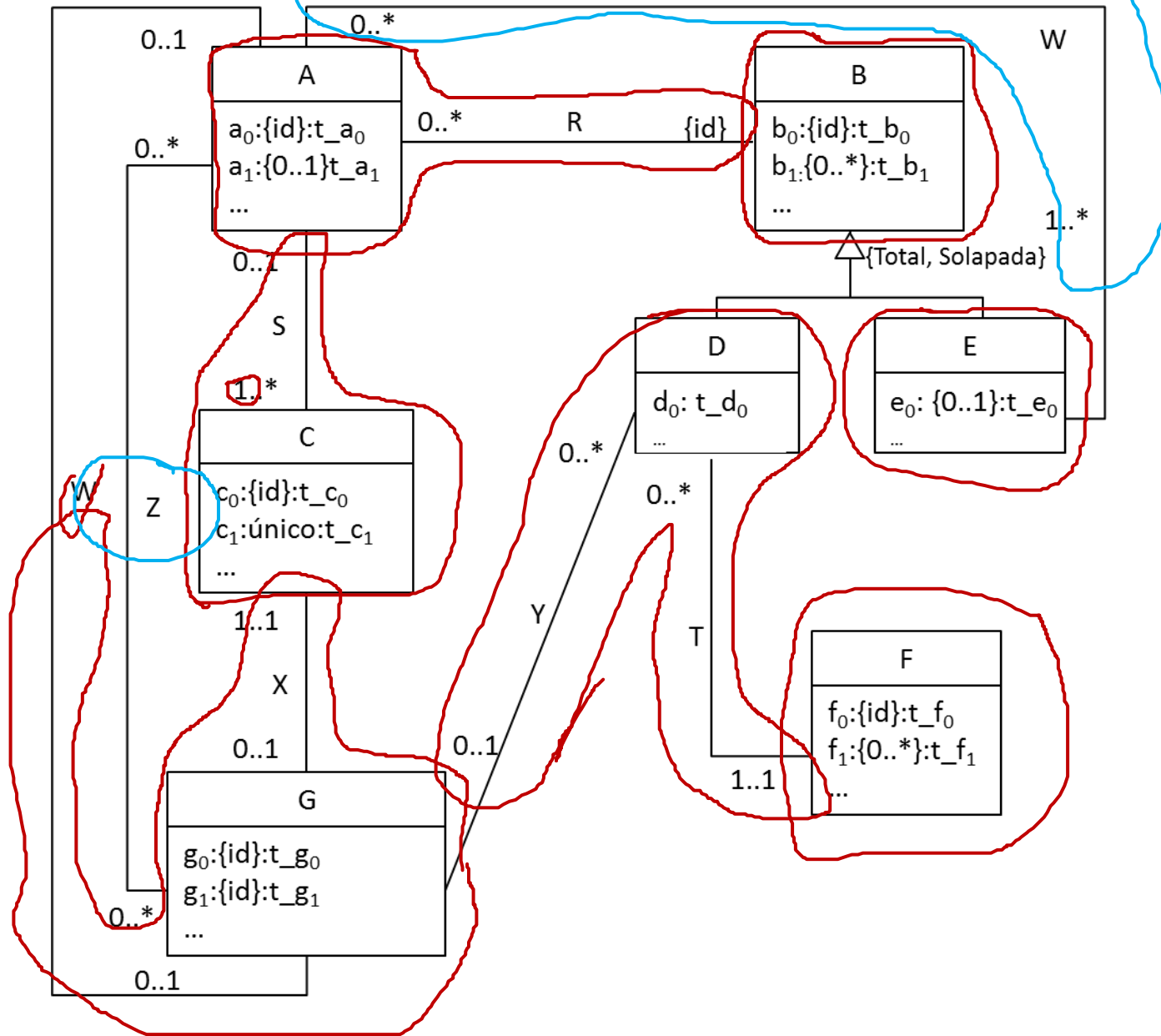
5.4.- Tercera forma normal (3FN)

5.5.- Reflexión sobre la teoría de la Normalización

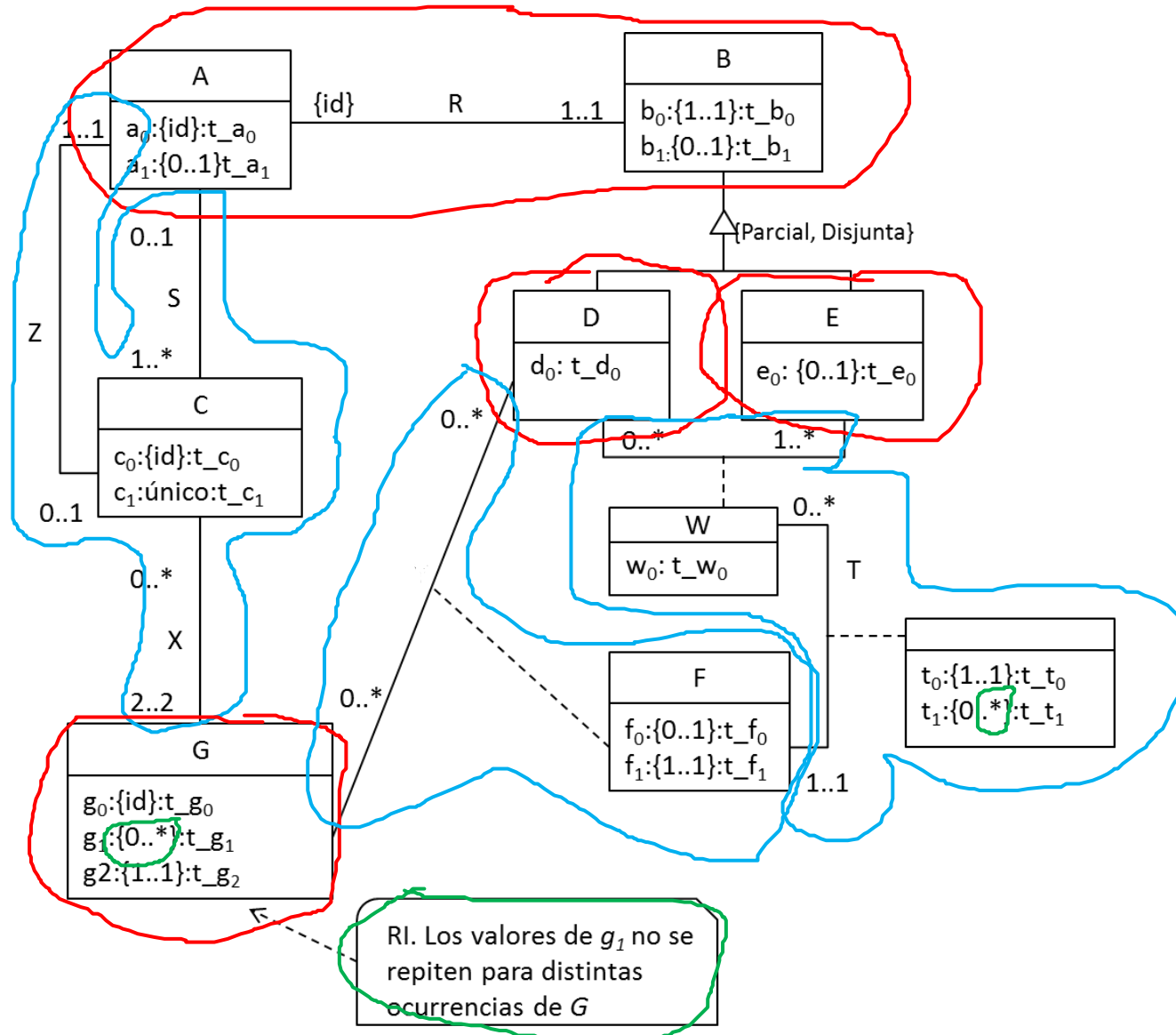
## 4.- Ejemplo de transformación



## 4.- Ejemplo de transformación



## 4.- Ejemplo de transformación



# UD 4.3.- Diseño lógico

## 1.- Introducción

## 2.- Transformación de las clases

2.1.- Clases fuertes

2.2.- Clases débiles

2.3.- Clases especializadas

## 3.- Transformación de las asociaciones

3.1.- No reflexivas

3.2.- Reflexivas

3.3.- Atributos de enlace

3.4.- Transformación de la asociación cuando se asocia con clases asociación

## 4.- Ejemplo de transformación

## 5.- Teoría de la Normalización

5.1.- Conceptos previos

5.2.- Primera forma normal (1FN)

5.3.- Segunda forma normal (2FN)

5.4.- Tercera forma normal (3FN)

5.5.- Reflexión sobre la teoría de la Normalización



## 5.- Teoría de la Normalización

---

- Tiene como objetivo garantizar que las relaciones obtenidas a partir del proceso de diseño lógico cumplan las propiedades del modelo relacional.
- Nosotros utilizaremos las formas normales como mecanismo de validación/corrección de la transformación obtenida en el diseño lógico.

# 5.- Teoría de la Normalización

---

- **Conceptos Previos:**

- Dependencia Funcional (completa o no)
- Diagrama de Dependencias Funcionales
- Clave de una relación
- Atributo Primo

# 5.- Teoría de la Normalización

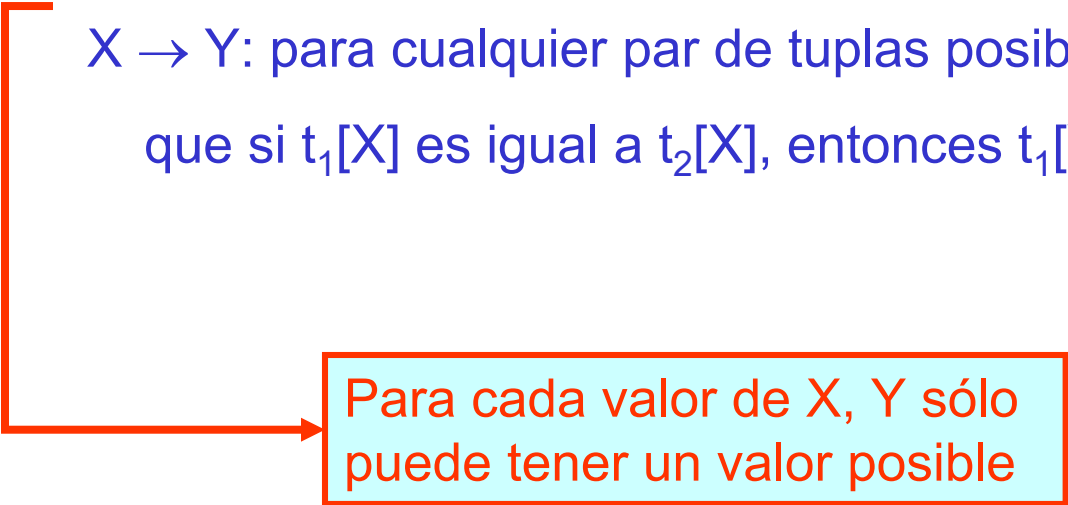
- **Conceptos Previos:**

- Dependencia Funcional

$A = \{A_1, \dots, A_n\}$  conjunto de atributos del esquema R

Una dependencia funcional entre X y Y ( $X \subseteq A$ ,  $Y \subseteq A$ ,  $X \neq Y$ )

$X \rightarrow Y$ : para cualquier par de tuplas posibles de R,  $t_1$  y  $t_2$ , se cumple que si  $t_1[X]$  es igual a  $t_2[X]$ , entonces  $t_1[Y]$  es igual a  $t_2[Y]$ .



Para cada valor de X, Y sólo puede tener un valor posible

# 5.- Teoría de la Normalización

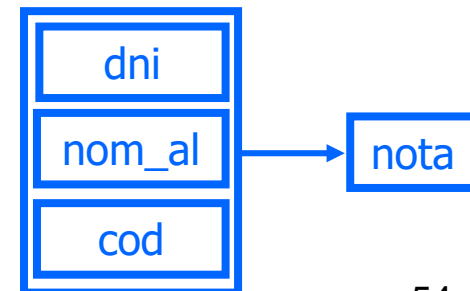
- **Conceptos Previos:**
  - Dependencia Funcional

exp	nom_al	cod	nom_as	nota
1	Pep	DBD	Diseño de Bases de Datos	6
1	Pep	BDA	Bases de Datos	7
2	Joana	DBD	Diseño de Bases de Datos	7
2	Joana	BDA	Bases de Datos	5

Ejemplos de dependencia funcional:



- {nom\_as} depende funcionalmente de {cod}
- {nota} depende funcionalmente de {exp, nom\_al, cod}



# 5.- Teoría de la Normalización

---

- **Conceptos Previos:**
  - Dependencia Funcional **completa**

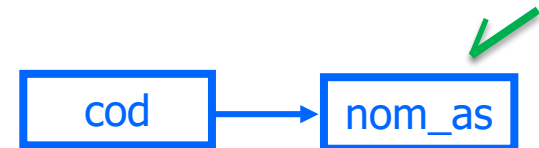
Una dependencia funcional entre dos conjuntos de atributos  $X \rightarrow Y$  es completa si la eliminación de cualquier atributo  $A_i$  de  $X$  hace que la dependencia deje de existir, es decir, si  $\forall A_i / A_i \in X$  se cumple que  $Y$  no depende funcionalmente de  $(X - \{A_i\})$ .

# 5.- Teoría de la Normalización

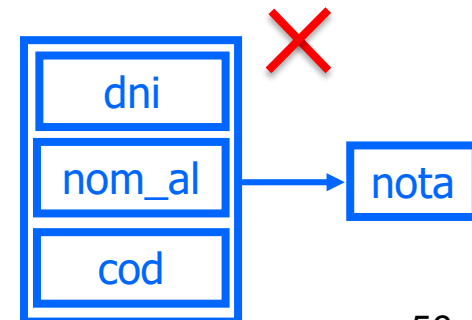
- **Conceptos Previos:**
  - Dependencia Funcional **completa**

exp	nom_al	cod	nom_as	nota
1	Pep	DBD	Diseño de Bases de Datos	6
1	Pep	BDA	Bases de Datos	7
2	Joana	DBD	Diseño de Bases de Datos	7
2	Joana	BDA	Bases de Datos	5

Ejemplos de dependencia funcional:



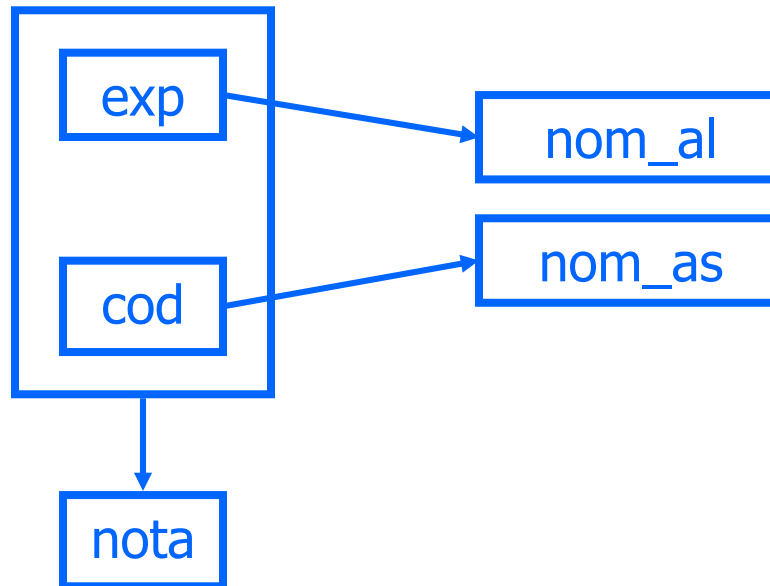
- {nom\_as} depende funcionalmente de {cod}
- {nota} depende funcionalmente de {exp, nom\_al, cod}



# 5.- Teoría de la Normalización

- **Conceptos Previos:**
  - **Diagrama de Dependencias Funcionales**

Representación gráfica de las dependencias. Se utilizan cajas para enmarcar los atributos o conjuntos de atributos y flechas para denotar la dependencia funcional. Normalmente sólo se representan las dependencias funcionales completas.



# 5.- Teoría de la Normalización

- **Conceptos Previos:**
  - Clave de una relación

Sea  $R$  un esquema de relación con el conjunto de atributos  $A = \{A_1, A_2, \dots, A_n\}$ , y sea  $C$  un subconjunto de atributos de ese esquema ( $C \subseteq A$ ); se dice que  $C$  es una clave de  $R$  si  $C$  es la clave primaria de  $R$  o bien si  $C$  tiene una restricción de unicidad.



Todos los atributos de la relación que no formen parte de una clave dependen funcionalmente de cada clave de la relación



# 5.- Teoría de la Normalización

---

- **Conceptos Previos:**
  - **Atributo Primo**

Sea  $R$  un esquema de relación con el conjunto de atributos  $A = \{A_1, A_2, \dots, A_n\}$ , diremos que un atributo  $A$  es primo si forma parte de alguna clave de  $R$

# 5.- Teoría de la Normalización

## 1ª Forma Normal

Una Relación está en 1FN si sus atributos sólo tienen valores atómicos (simples e indivisibles).

**Problemas** cuando se utilizan relaciones que no están en 1FN: Se debe utilizar operadores asociados a los tipos de datos complejos (listas, conjuntos, registros,...)

Conjunto

Registro

CP: {vcod}

vcod	nom	teléfonos	dir
V1	Pep	(96 3233258, 964 523844, 979 568987, 987 456123)	Pau 7, Valencia
V2	Joan	(96 3852741, 910 147258)	Eolo 3, Castellón
V3	Eva	(987 456 312)	F. Lorca 2, Utiel

# 5.- Teoría de la Normalización

---

- **1ª Forma Normal**

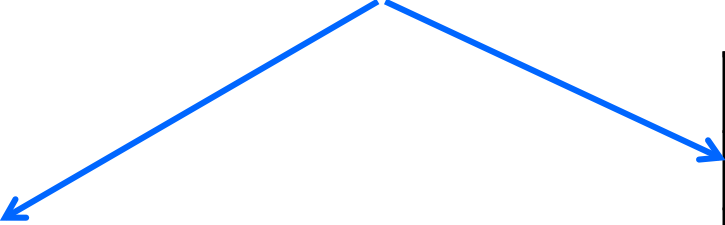
Paso a 1FN:

- si R tiene un atributo que es un conjunto, lo eliminaremos de la relación y definiremos una nueva relación con ese atributo y buscaremos la clave primaria.

# 5.- Teoría de la Normalización

- 1ª Forma Normal

vcod	nom	teléfonos	dir
V1	Pep	(96 3233258, 964 523844, 979 568987, 987 456123)	Pau 7, València
V2	Joan	(96 3852741, 910 147258)	Eolo 3, Castelló
V3	Eva	(987 456 312)	F. Lorca 2, Utiel



vcod	nom	dir
V1	Pep	Pau 7, València
V2	Joan	Eolo 3, Castelló
V3	Eva	F. Lorca 2, Utiel

vcod	telefono
V1	96 3233258
V2	96 3852741
V3	987 456 312
V1	964 523844
V1	979 568987
V1	987 456123
V2	910 147258

# 5.- Teoría de la Normalización

- **1ª Forma Normal**

Proveedor(vcod, nom, teléfonos, dir)

CP: {vcod}

Proveedor(vcod, nom, dir)

CP: {vcod}

Llistín(vcod, teléfono)

CP: {teléfono}

CAj: {vcod} → Proveedor

VNN: {vcod}

Si los teléfonos no se pueden  
compartir

# 5.- Teoría de la Normalización

- **1ª Forma Normal**

Proveedor(vcod, nom, teléfonos, dir)

CP: {vcod}

Proveedor(vcod, nom, dir)

CP: {vcod}

Llistín(vcod, teléfono)

CP: {teléfono, vcod}

CAj: {vcod} → Proveedor

Si los teléfonos se pueden  
compartir

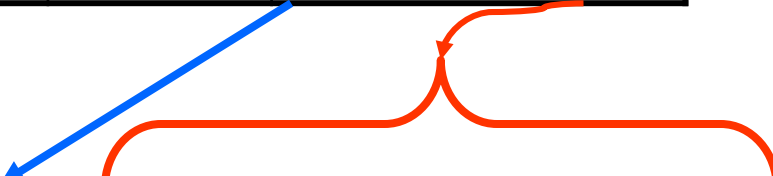
# 5.- Teoría de la Normalización

## 1ª Forma Normal

□ Paso a 1FN:

- si R tiene un atributo que es un registre se debe sustituir por los campos del registro.

<b>vcod</b>	<b>nom</b>	<b>dir</b>
V1	Pep	Paz 7, València
V2	Joan	Eolo 3, Castelló
V3	Eva	F. Lorca 2, Utiel



<b>vcod</b>	<b>nom</b>	<b>calle</b>	<b>número</b>	<b>ciudad</b>
V1	Pep	Paz	7	València
V2	Joan	Eolo	3	Castelló
V3	Eva	F. Lorca	2	Utiel

# 5.- Teoría de la Normalización

---

## 1ª Forma Normal

Proveedor(vcod, nom, dir)

CP: {vcod}



Proveedor(vcod, nom, calle, número, ciudad)

CP: {vcod}



# 5.- Teoría de la Normalización

---

## 2ª Forma Normal

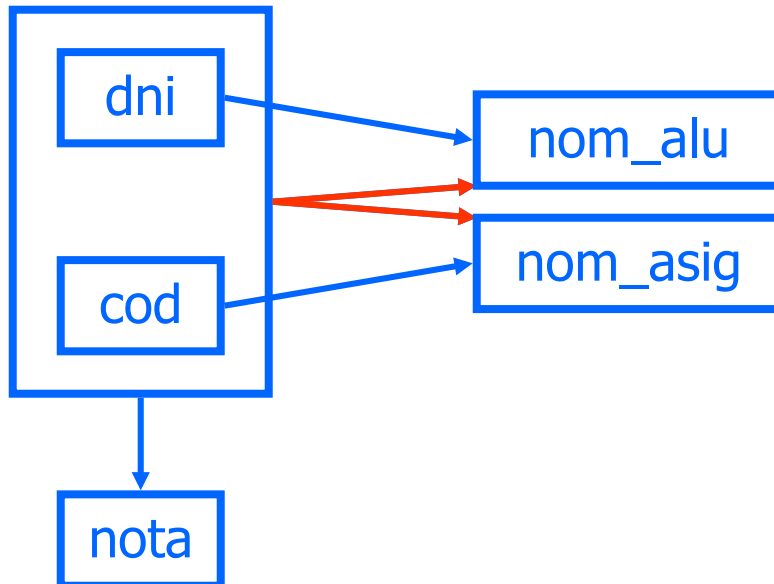
- Una Relación está en 2FN si está en 1FN y todo atributo no-primo depende funcionalmente de forma completa de **toda clave de R**.
- Problemas cuando utilizamos relaciones que no están en 2FN:
  - Existen redundancias
  - La manipulación de la información es complicada
  - No es fácil insertar ni borrar

# 5.- Teoría de la Normalización

## 2ª Forma Normal

CP: {dni, cod}

<b>dni</b>	<b>nom_alu</b>	<b>cod</b>	<b>nom_asig</b>	<b>nota</b>
1	Pep	DBD	Diseño de Bases de Datos	6
1	Pep	BDA	Bases de Datos	7
2	Joana	DBD	Diseño de Bases de Datos	7
2	Joana	BDA	Bases de Datos	5



→ D.F. Completa

→ D.F. No Completa

# 5.- Teoría de la Normalización

---

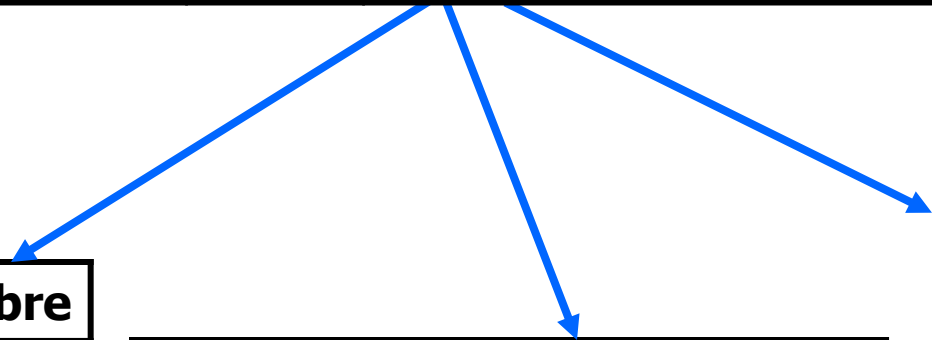
## 2ª Forma Normal

- Paso a 2FN:
  - La clave consta de más de un atributo y existe algún atributo no-primo que no depende completamente de la clave primaria.
  - Se debe dividir la relación original en varias relaciones para eliminar las dependencias no completas.

# 5.- Teoría de la Normalización

## 2ª Forma Normal

<b>dni</b>	<b>nom_alu</b>	<b>cod</b>	<b>nom_asig</b>	<b>nota</b>
1	Pep	DBD	Diseño de Bases de Datos	6
1	Pep	BDA	Bases de Datos	7
2	Joana	DBD	Diseño de Bases de Datos	7
2	Joana	BDA	Bases de Datos	5



<b>dni</b>	<b>nombre</b>
1	Pep
2	Joana

<b>cod</b>	<b>nombre</b>
DBD	Diseño de Bases de Datos
BDA	Bases de Datos

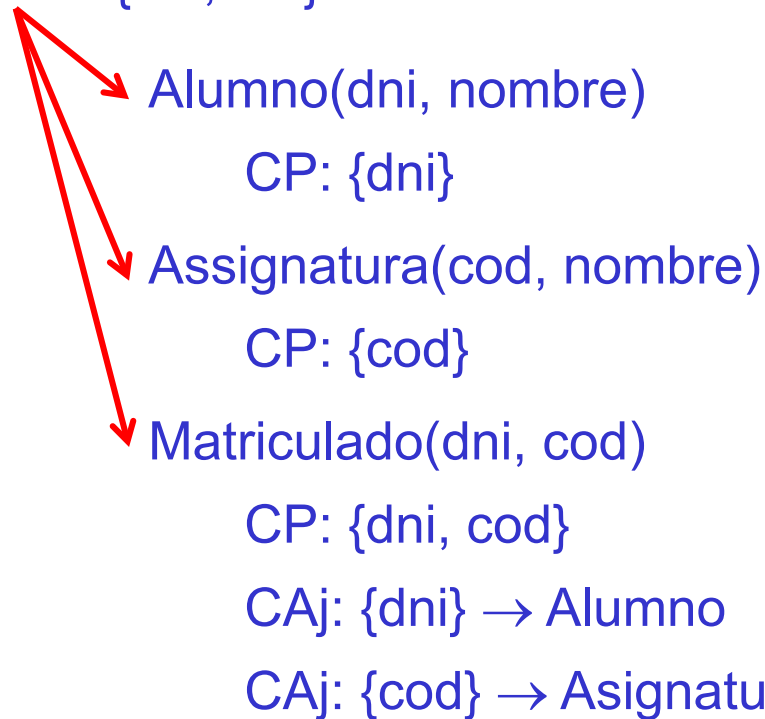
<b>dni</b>	<b>cod</b>	<b>nota</b>
1	DBD	6
2	BDA	7
1	DBD	7
2	BDA	5

# 5.- Teoría de la Normalización

## 2ª Forma Normal

Matriculado(dni, cod, nom\_alu, nom\_asig)

CP: {dni,cod}



# 5.- Teoría de la Normalización

---

## 3ª Forma Normal

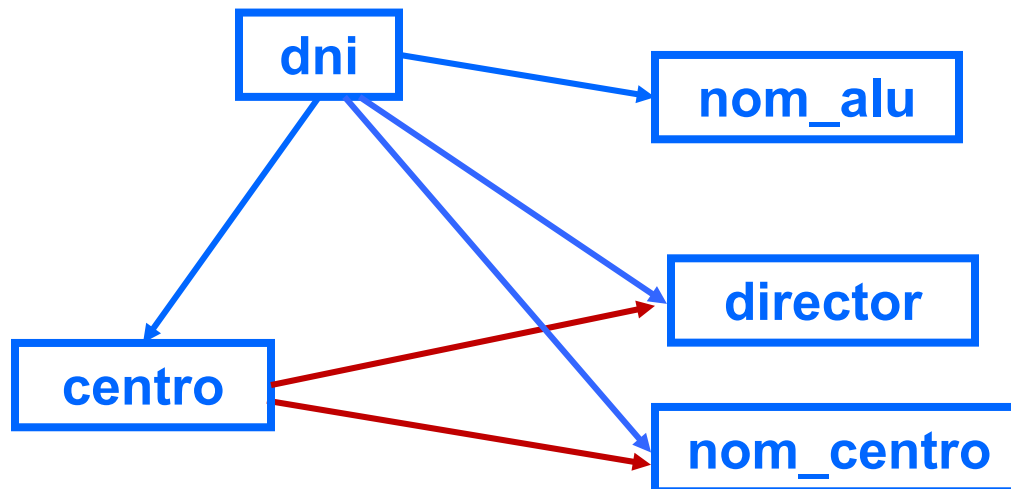
- Una Relación está en 3FN si está en 2FN y no existen dependencias funcionales entre atributos no-primos.
- Problemas cuando se utilizan relaciones que no están en 3FN
  - Existen redundancias
  - La manipulación de la información es complicada
  - No es fácil insertar ni borrar

# 5.- Teoría de la Normalización

## 3ª Forma Normal

<b>dni</b>	<b>nom_alu</b>	<b>centro</b>	<b>nom_centro</b>	<b>director</b>
1	Olga	EUI	Escuela Universitaria de Informática	Pep
2	Jona	EUI	Escuela Universitaria de Informática	Pep
3	Anna	FI	Facultad de Informática	Eva
4	Marc	FI	Facultad de Informática	Eva

CP: {dni}



D.F. entre atributos

No primos

(Dep. Transitivas)

# 5.- Teoría de la Normalización

---

## 3ª Forma Normal

- Paso a 3FN:
  - Existen al menos un par de atributos no-primos que son dependientes.
  - Quitar el atributo de la relación y crear una nueva relación que incluya ese atributo y que tenga como clave primaria el atributo del que depende.



# 5.- Teoría de la Normalización

## 3ª Forma Normal

<b>dni</b>	<b>nom_alu</b>	<b>centro</b>	<b>nom_centro</b>	<b>director</b>
1	Olga	EUI	Escuela Universitaria de Informática	Pep
2	Joana	EUI	Escuela Universitaria de Informática	Pep
3	Anna	FI	Facultad de Informática	Eva
4	Marc	FI	Facultad de Informática	Eva

CP: {dni}



<b>dni</b>	<b>nom_alu</b>	<b>centro</b>
1	Olga	EUI
2	Joana	EUI
3	Anna	FI
4	Marc	FI

<b>centre</b>	<b>nom_centro</b>	<b>director</b>
EUI	Escuela Universitaria de Informática	Pep
FI	Facultad de Informática	Eva

# 5.- Teoría de la Normalización

## 3ª Forma Normal

Alumno(dni, nom\_alu, centro, nom\_centro, director)

CP: {dni}

→ Alumno(dni, nombre, centro)

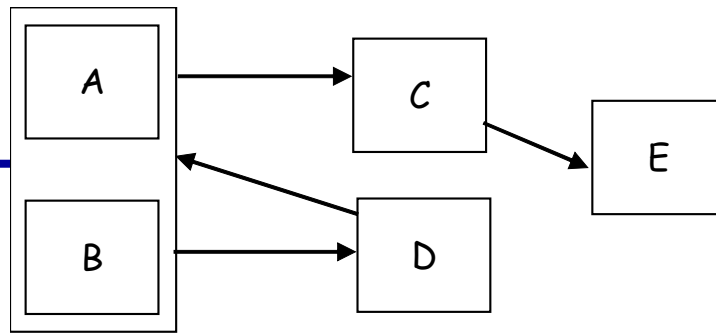
CP: {dni}

CAj: {centro} → Centro

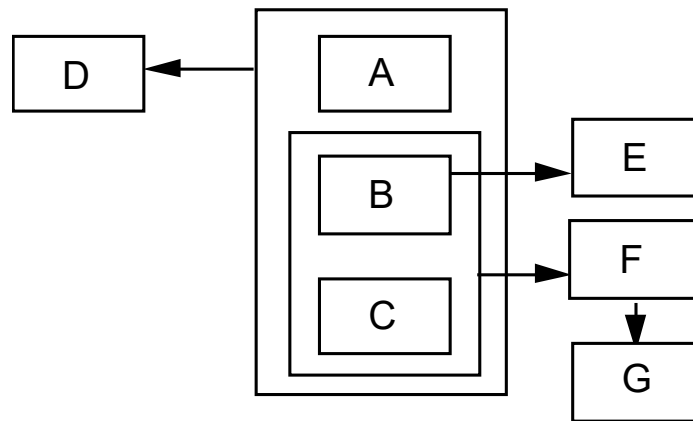
→ Centro(centro, nombre, director)

CP: {centro}

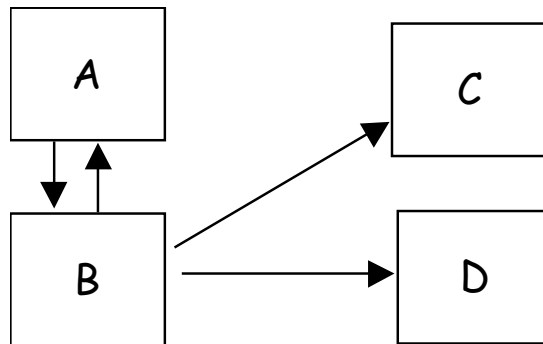
**E1**



**E2**

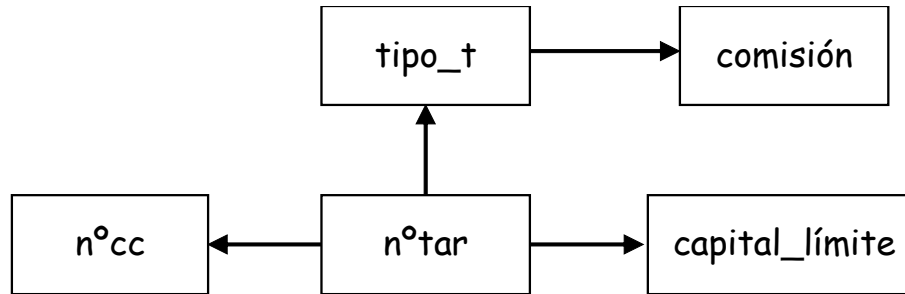


**E3**



# E4

**Tarjeta(n°cc: cadena(15), n°tar: cadena(15), tipo\_t: cadena(10), comisión: real, capital\_límite: real)**  
CP: {n°tar}  
CAj: {n°cc} → Cuenta  
VNN: { n°cc, tipo\_t, comisión, capital\_límite }



**Tarjeta(n°cc: cadena(15), n°tar: cadena(15), tipo\_t: cadena(10), capital\_límite: real)**  
CP: {n°tar}  
CAj: {n°cc} → Cuenta  
VNN: {n°cc, tipo\_t, capital\_límite}  
**Comisión(tipo\_t: cadena(10), comisión: real)**  
CP: {tipo\_t}  
VNN: {comisión}

**Sea el siguiente esquema de relación:**

R(A: entero, B: texto, C: entero, D: texto, E: texto, F: texto, G: texto)

CP: {A, B}

VNN: {C, D, E, F, G}

**A partir de las dependencias que aparecen a continuación, transformad la relación R en un conjunto de relaciones en tercera forma normal.**

$\{G\} \rightarrow \{E\}$

$\{A\} \rightarrow \{D\}$

$\{G\} \rightarrow \{F\}$

$\{A\} \rightarrow \{G\}$

# Ejercicio

---

Sea el siguiente esquema de relación:

R(A: entero, B: texto, C: entero, D: texto, E: texto, F: texto, G: texto)

CP: {A, B}

VNN: {C, D, E, F, G}

Teniendo en cuenta las dependencias que se exponen a continuación, transfórmala a un conjunto de relaciones en tercera forma normal.

$\{G\} \rightarrow \{E\}$

$\{G\} \rightarrow \{F\}$

$\{A\} \rightarrow \{D\}$

$\{A\} \rightarrow \{G\}$

# Ejercicio

R(A: entero, B: texto, C: entero, D: texto, E: texto, F: texto, G: texto)

CP: {A, B}

VNN: {C, D, E, F, G}

$\{G\} \rightarrow \{E\}$

$\{G\} \rightarrow \{F\}$

$\{A\} \rightarrow \{D\}$

$\{A\} \rightarrow \{G\}$

Dependencias transitivas:

Si  $\{A\} \rightarrow \{G\}$  y  $\{G\} \rightarrow \{E\}$

$\{A\} \rightarrow \{E\}$

Si  $\{A\} \rightarrow \{G\}$  y  $\{G\} \rightarrow \{F\}$

$\{A\} \rightarrow \{F\}$

## 2FN

$\{G\} \rightarrow \{E\}$

$\{G\} \rightarrow \{F\}$

$\{A\} \rightarrow \{D\}$

$\{A\} \rightarrow \{G\}$

$\{A\} \rightarrow \{E\}$

$\{A\} \rightarrow \{F\}$

**R1**(A: entero, B: texto, C: entero)

CP: {A, B}

CAj: {A}  $\rightarrow$  R21

VNN: {C}

**R21** (A: entero, D: texto, G: texto, E: texto, F: texto)

CP: {A}

VNN: {D, G, E, F}

Todo valor de {A} de la relación R21 aparece en R1.

# Ejercicio

**R21** (A: entero, D: texto, G: texto, E: texto, F: texto)

CP: {A}

VNN: {D, G, E, F}

Todo valor de {A} de la relación R21 aparece en R1.

**R1** (A: entero, B: texto, C: entero)

CP: {A, B}

CAj: {A}  $\rightarrow$  R21

VNN: {C}

## 3FN

$\{G\} \rightarrow \{E\}$      $\{G\} \rightarrow \{F\}$      $\{A\} \rightarrow \{D\}$      $\{A\} \rightarrow \{G\}$      $\{A\} \rightarrow \{E\}$      $\{A\} \rightarrow \{F\}$

**R1** (A: entero, B: texto, C: entero)

CP: {A, B}

CAj: {A}  $\rightarrow$  R21

VNN: {C}

**R21** (A: entero, D: texto, G: texto)

CP: {A}

CAj: {G}  $\rightarrow$  R22

VNN: {D, G}

**R22** (G: entero, E: texto, F: texto)

CP: {G}

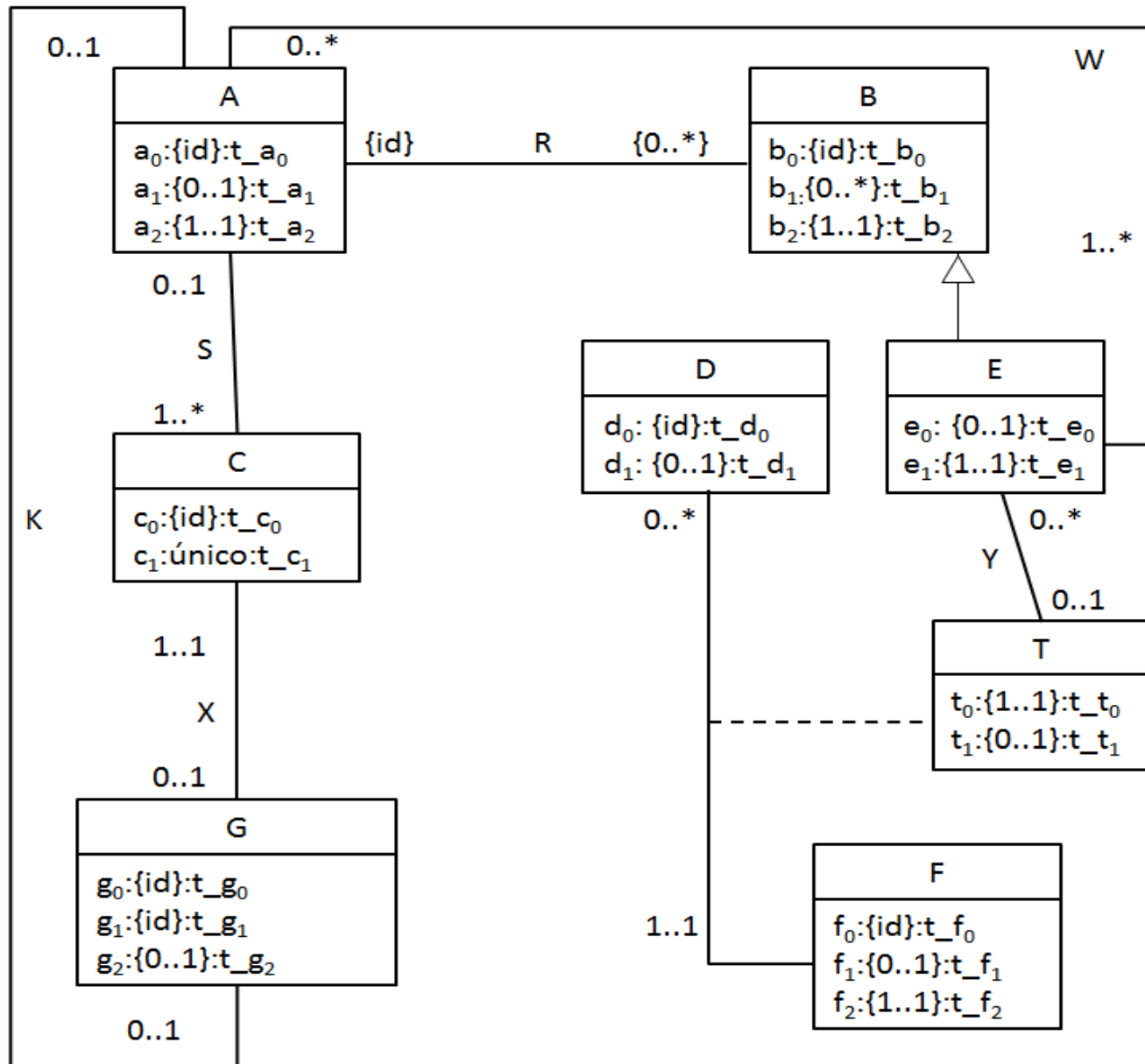
VNN: {E, F}

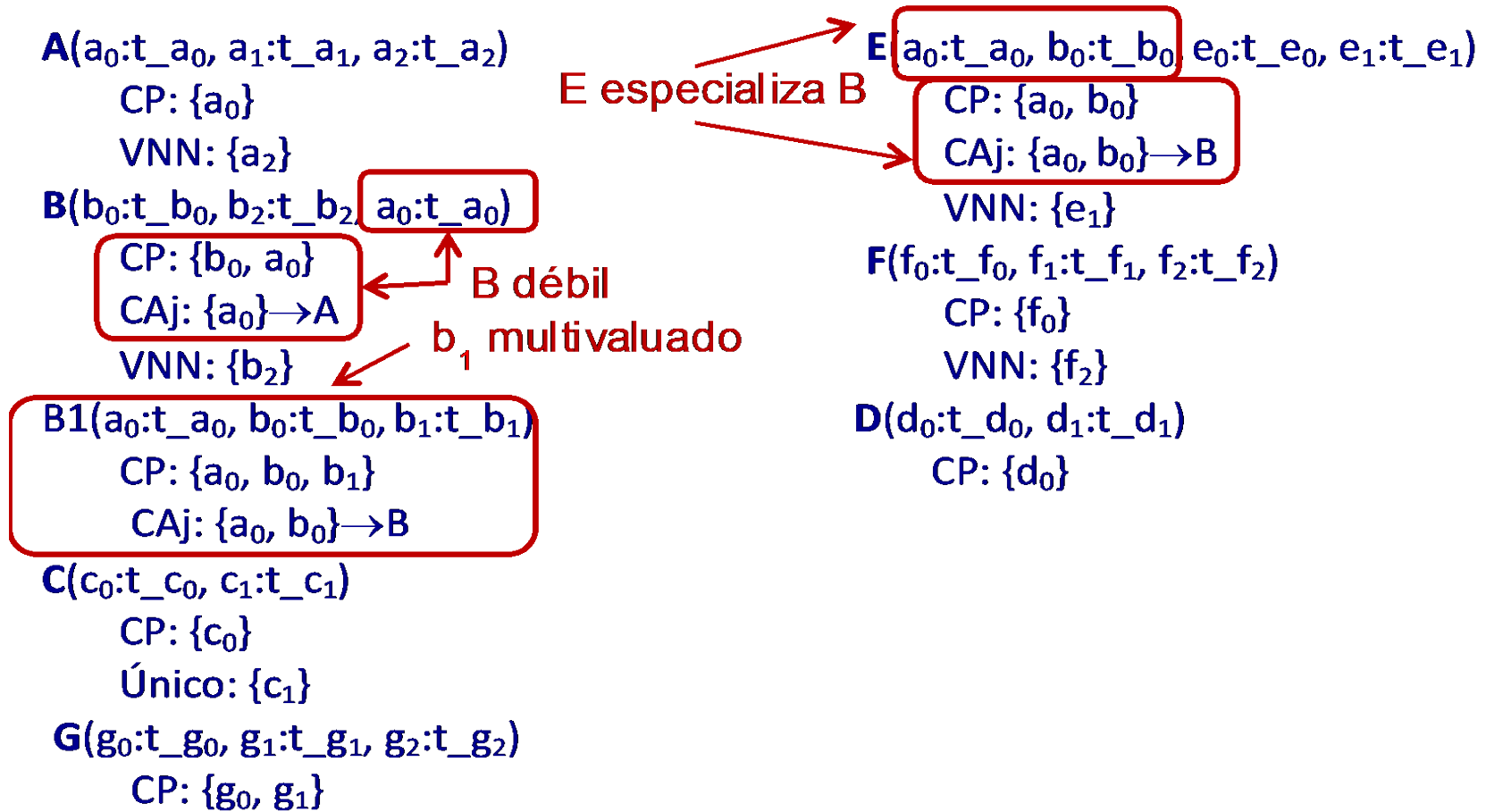
Todo valor de {A} de la relación R21 aparece en R1.

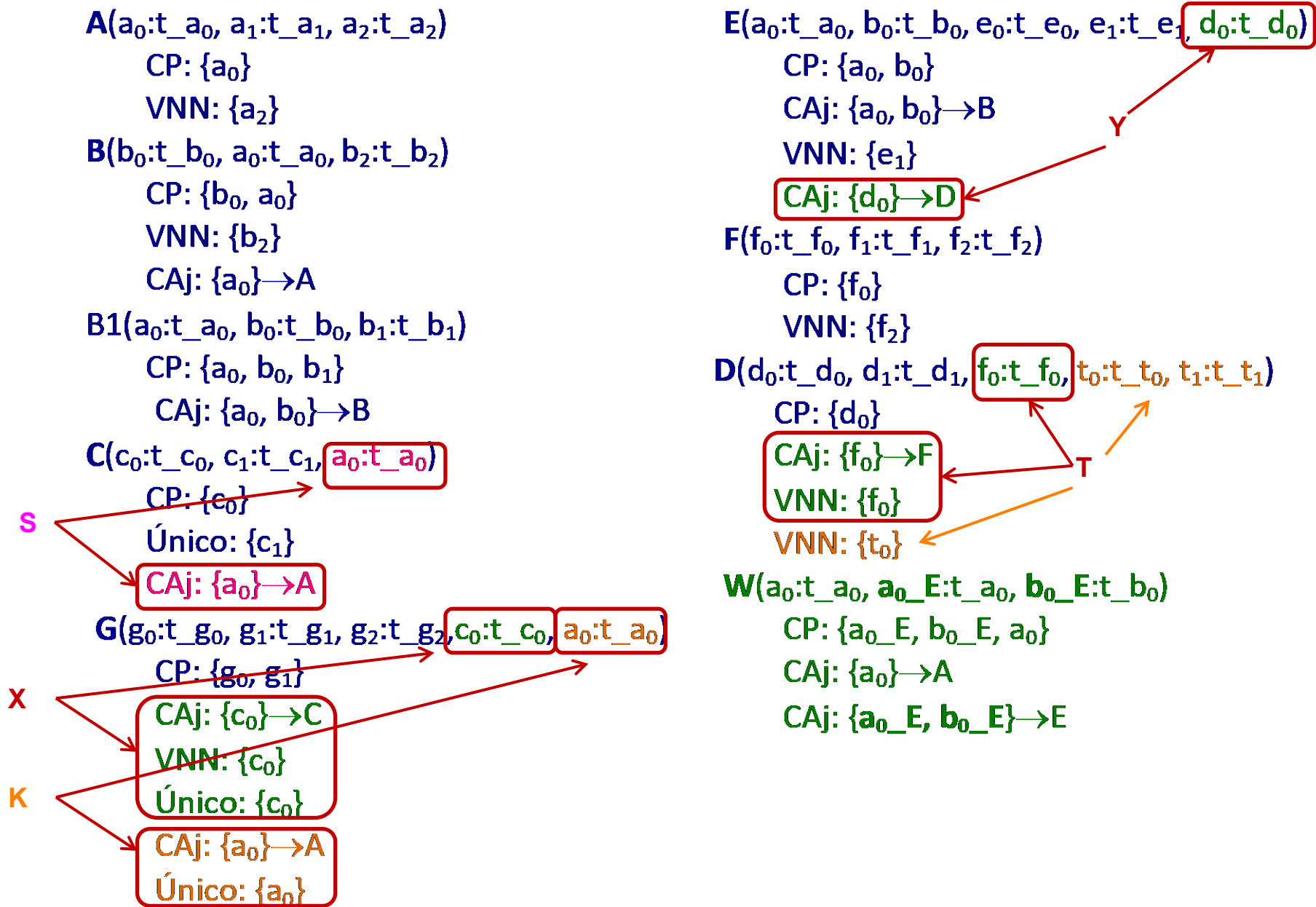
Todo valor de {G} de la relación R22 aparece en R21.



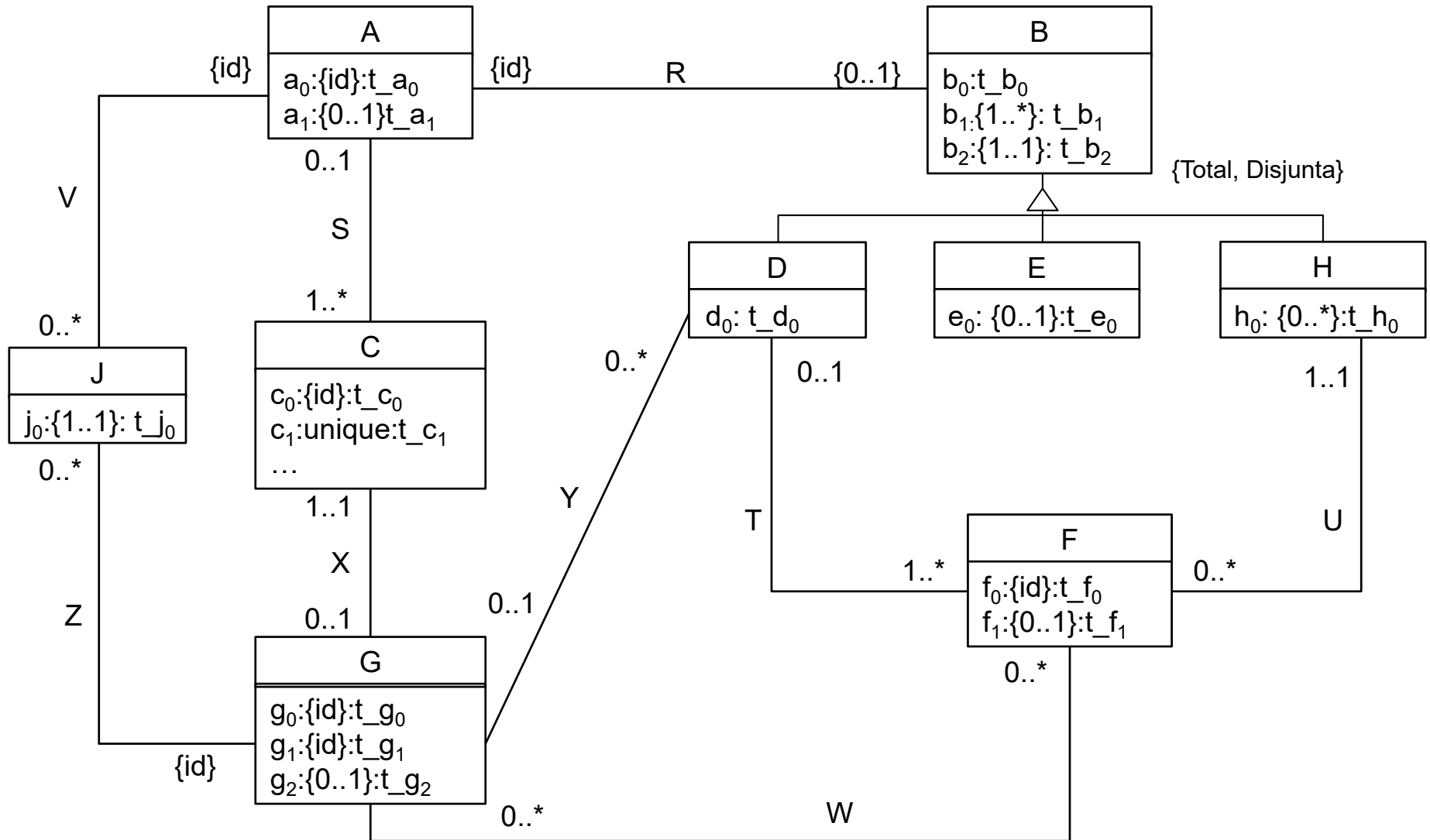
# Ejemplo. Diseño lógico



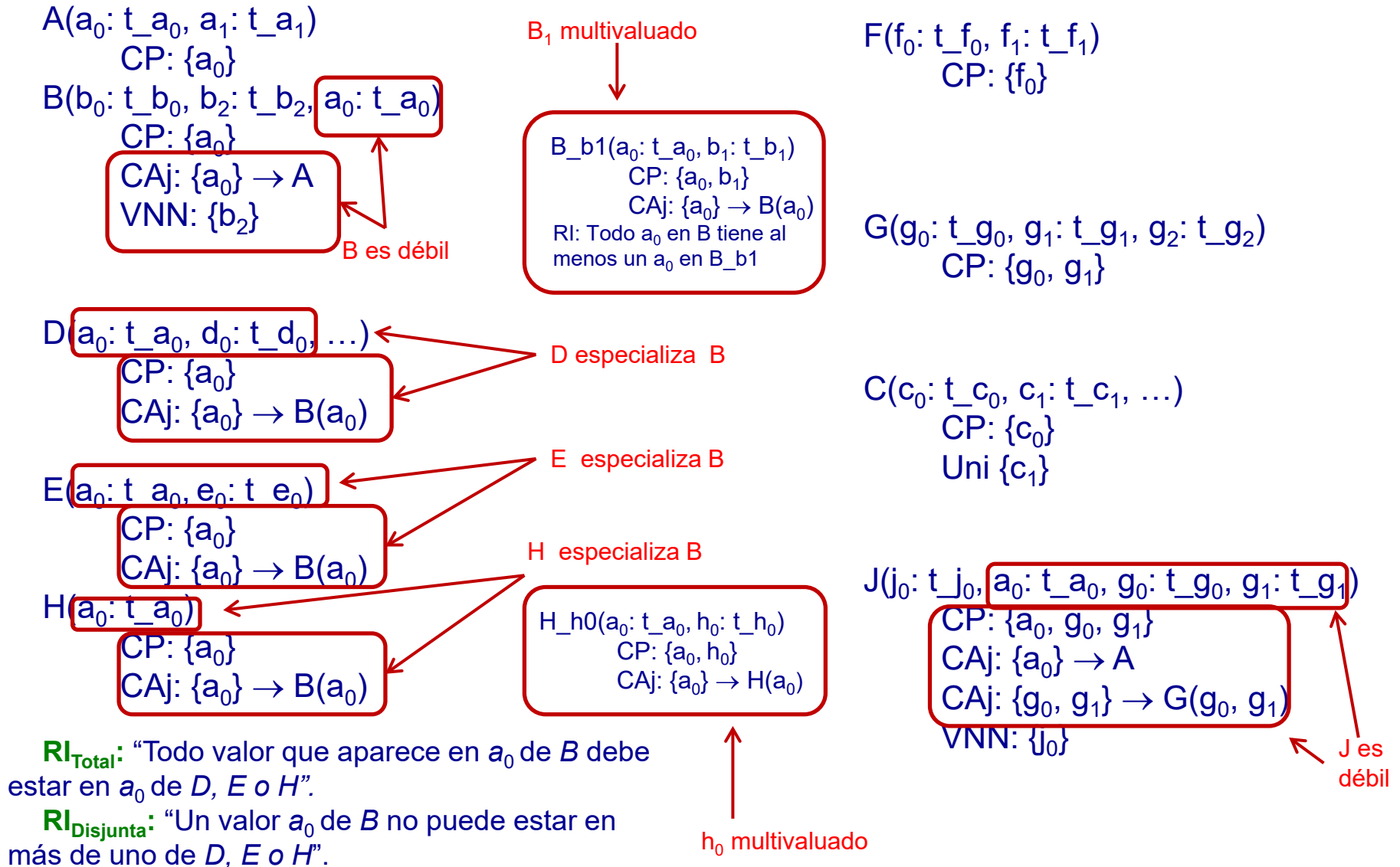




# Ejemplo. Diseño lógico



# Ejemplo. Transformación de clases



$A(a_0: t\_a_0, a_1: t\_a_1)$

CP:  $\{a_0\}$

$B(b_0: t\_b_0, b_2: t\_b_2, a_0: t\_a_0)$

CP:  $\{a_0\}$

CAj:  $\{a_0\} \rightarrow A$

VNN:  $\{b_2\}$

$D(a_0: t\_a_0, d_0: t\_d_0, g_0: t\_g_0, g_1: t\_g_1 \dots)$

CP:  $\{a_0\}$

CAj:  $\{a_0\} \rightarrow B(a_0)$

CAj:  $\{g_0, g_1\} \rightarrow G(g_0, g_1)$  ← Y

$E(a_0: t\_a_0, e_0: t\_e_0)$

CP:  $\{a_0\}$

CAj:  $\{a_0\} \rightarrow B(a_0)$

$H(a_0: t\_a_0)$

CP:  $\{a_0\}$

CAj:  $\{a_0\} \rightarrow B(a_0)$

RI<sub>Total</sub>: "Todo valor que aparece en  $b_0$  de  $B$  debe estar en  $b_0$  de  $D$ ,  $E$  o  $H$ ".

RI<sub>Disjunta</sub>: "Un valor  $b_0$  de  $B$  no puede estar en más de uno de  $D$ ,  $E$  o  $H$ ".

$H\_h0(a_0: t\_a_0, h_0: t\_h_0)$

PK:  $\{a_0, h_0\}$

FK:  $\{a_0\} \rightarrow H(a_0)$

$B\_b1(a_0: t\_a_0, b_1: t\_b_1)$

PK:  $\{a_0, b_1\}$

FK:  $\{a_0\} \rightarrow B(a_0)$

RI: Todo  $a_0$  en  $B$  tiene al menos un  $a_0$  en  $B\_b1$

$F(f_0: t\_f_0, f_1: t\_f_1, a_0D: t\_a_0, a_0H: t\_a_0)$

CP:  $\{f_0\}$

CAj:  $\{a_0D\} \rightarrow D(a_0)$

RI: "Todo valor  $a_0$  en  $D$  debe estar en  $F$ "

CP:  $\{a_0H\} \rightarrow H(a_0)$  VNN:  $\{a_0H\}$

$G(g_0: t\_g_0, g_1: t\_g_1, g_2: t\_g_2, c_0: t\_c_0)$

CP:  $\{g_0, g_1\}$

UNI:  $\{c_0\}$  NNV:  $\{c_0\}$

CAj:  $\{c_0\} \rightarrow C(c_0)$  ← X

$C(c_0: t\_c_0, c_1: t\_c_1, \dots, a_0: t\_a_0)$

CP:  $\{c_0\}$

UNI:  $\{c_1\}$

CAj:  $\{a_0\} \rightarrow A$

RI: "Todo valor  $a_0$  en  $A$  debe estar en  $C$ " ← S

$J(j_0: t\_j_0, a_0: t\_a_0, g_0: t\_g_0, g_1: t\_g_1)$

CP:  $\{a_0, g_0, g_1\}$

CAj:  $\{a_0\} \rightarrow A$

CAj:  $\{g_0, g_1\} \rightarrow G(g_0, g_1)$

VNN:  $\{j_0\}$

$W(g_0: t\_g_0, g_1: t\_g_1, f_0: t\_f_0)$

CP:  $\{g_0, g_1, f_0\}$

CAj:  $\{g_0, g_1\} \rightarrow G(g_0, g_1)$

CAj:  $\{f_0\} \rightarrow F$

U

T

X

S

# Ejercicio normalización

---

Sea el siguiente esquema de relación:

R (A: entero, B: carácter, C: carácter, D: conjunto de entero, E: entero, F: texto, G: entero, H: entero)

CP: {A, B, C}

VNN: {D, E, F, G, H}

A partir de las dependencias que aparecen a continuación, transforme la relación a un conjunto de relaciones en tercera forma normal.

$\{B\} \rightarrow \{G\}$   $\{A, B\} \rightarrow \{F\}$   $\{G\} \rightarrow \{H\}$

# Solución

R (A: entero, B: carácter, C: carácter, E: entero)

CP: {A, B, C}

CAj: {A, B}  $\rightarrow$  R3

CAj: {B}  $\rightarrow$  R2

CAj: {A,B}  $\rightarrow$  R3

VNN: {E}

**Todo (A, B, C) de R1 debe existir en R**

R2 (B: carácter, G: entero)

CP: {B}

CAj: {G}  $\rightarrow$  R22

VNN: {G}

R1 (A: entero, B: carácter, C: carácter, D: entero)

CP: {A, B, C, D}

CAj: {A, B, C}  $\rightarrow$  R

R3 (A: entero, B: carácter, F: texto)

CP: {A,B}

CAj: {B}  $\rightarrow$  R2

VNN: {F}

R22 (G: entero, H: entero)

CP: {G}

VNN: {H}



# Ejercicio normalización

---

Sea el siguiente esquema de relación: Sea el siguiente esquema de relación:

$R(A: \text{entero}, B: \text{texto}, C: \text{entero}, D: \text{texto}, E: \text{texto}, F: \text{texto}, G: \text{texto}, H: \text{conjunto de enteros})$

CP:  $\{A, B\}$  VNN:  $\{C, D, E, F, G\}$

Teniendo en cuenta las dependencias que se exponen a continuación, transforma la relación R en un conjunto de relaciones en tercera forma normal.

$\{A\} \rightarrow \{C\}, \quad \{E\} \rightarrow \{F\}, \quad \{F\} \rightarrow \{G\}$

# Solución

R (A: entero, B: texto, D: texto, E: texto)

CP: {A, B}

CAj: {A}  $\rightarrow$  R3

CAj: {E}  $\rightarrow$  R2

VNN: {D,E}

R2 (E: texto, F: texto)

CP: {E}

CAj: {F}  $\rightarrow$  R4

VNN: {F}

R1 (A: entero, B: texto, H: entero)

CP: {A, B, H}

CAj: {A, B}  $\rightarrow$  R

R3 (A: entero, C: texto)

CP: {A}

VNN: {C}

R4(F: texto, G: texto)

CP: {F}

VNN: {G}