

# Tema III: Transformación del esquema conceptual al relacional

## 3.1 Introducción. Etapas del diseño lógico

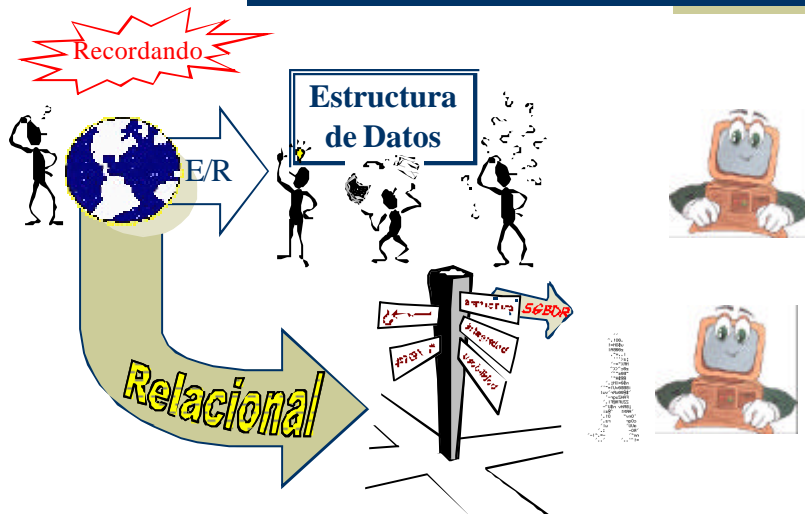
Diseño lógico estándar  
Diseño lógico específico

## 3.2 Transformación elementos básicos

## 3.3 Reglas concernientes a las extensiones del modelo E/R

Transformación de dependencias en identificación y en existencia  
Transformación de interrelaciones exclusivas  
Transformación de tipos y subtipos  
Transformación de la dimensión temporal  
Transformación de atributos derivados  
Transformación de interrelaciones de grado superior a dos

# Tema 3.1: Introducción. Etapas del diseño Lógico



# Tema 3.1: Introducción.

## Etapas del diseño Lógico

### A) Diseño lógico estándar

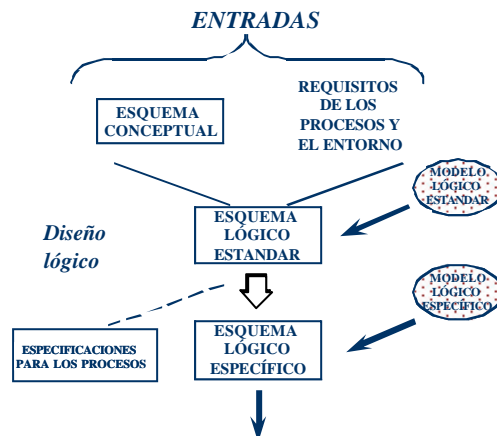
- Elaboración del Esquema Lógico Estándar que se apoya en el modelo lógico estándar **-Relacional, Codasyl, Jerárquico-**
- El Esquema Lógico Estándar se describirá utilizando el lenguaje estándar, si existe, del modelo de datos correspondiente (v.g. el SQL92)

### B) Diseño lógico específico

- Con el Esquema Lógico Estándar, y teniendo en cuenta el modelo lógico específico propio del SGBD, se elabora el esquema lógico específico, que será descrito en el lenguaje del producto comercial que estemos utilizando (p. e. **Oracle**)

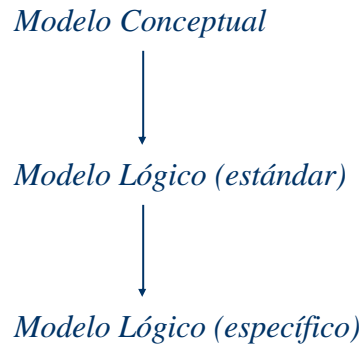
# Tema 3.1: Introducción.

## Etapas del diseño Lógico



\* Imagen inspirada en [5], página 345

## Tema 3.1: Introducción. Etapas del diseño Lógico



## Tema 3.2: Transformación Elementos Básicos

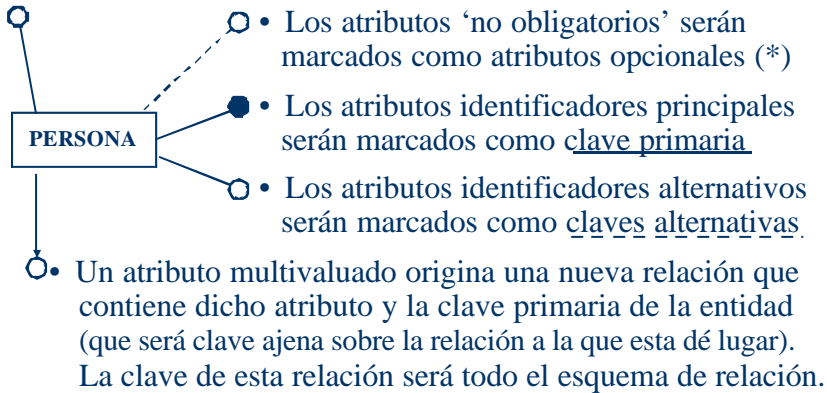


- Los dominios en E/R se mantienen como dominios en Relacional
- Las entidades en E/R se traducen en relaciones del modelo Relacional
- Las interrelaciones en E/R se traducen en
  - relaciones del modelo Relacional
  - propagación de claves (clave ajena) \*

Nota \*: aunque una clave ajena parece recoger menos semántica que una relación E/R, esta semántica se complementa con la que aporta la restricción referencial.

## Tema 3.2: Transformación Elementos Básicos

Los **ATRIBUTOS** de una entidad serán atributos de la relación correspondiente, con ciertas salvedades:



## Tema 3.2: Transformación Elementos Básicos

### Interrelaciones N:M

Se traducen en una relación.

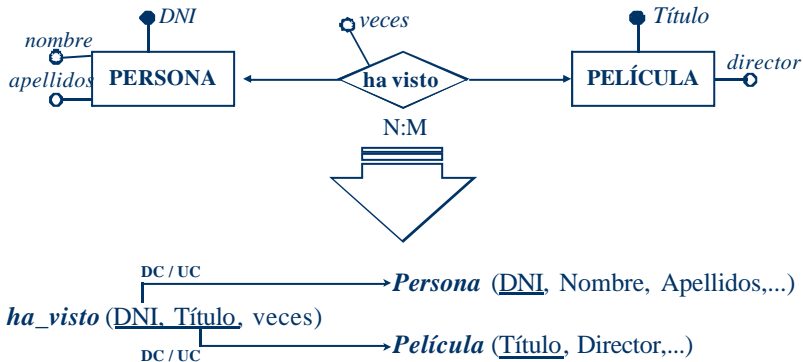


- Esta relación contendrá las claves de las relaciones asociadas, que en conjunto serán clave de la nueva relación.
- También incluirá los atributos de la interrelación original.
- Las opciones de borrado y modificación dependerán del cada caso particular (si bien, en general, se escogerá en ambas la opción *casca*)

## Tema 3.2: Transformación Elementos Básicos

### Interrelaciones N:M

Ejemplo:



## Tema 3.2: Transformación Elementos Básicos

### Interrelaciones 1:N

Dos posibilidades:

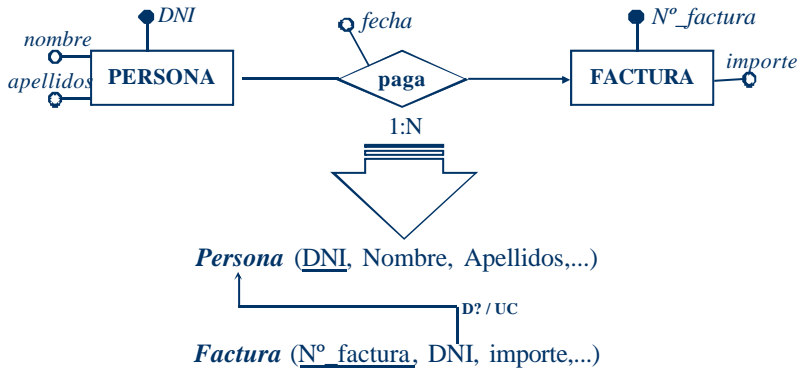


- Propagar la clave de la entidad que interviene con cardinalidad 1 (en la relación correspondiente a la otra entidad aparecerá esta clave como clave ajena; junto a ella, irán los atributos de la interrelación si los hubiera)
- Transformarla en una nueva relación  
Dicha relación tendría como atributos las claves de ambas entidades (y los atributos propios de la interrelación). Su clave sería la clave de la entidad que interviene en la interrelación con N ocurrencias).

# Tema 3.2: Transformación Elementos Básicos

## Interrelaciones 1:N

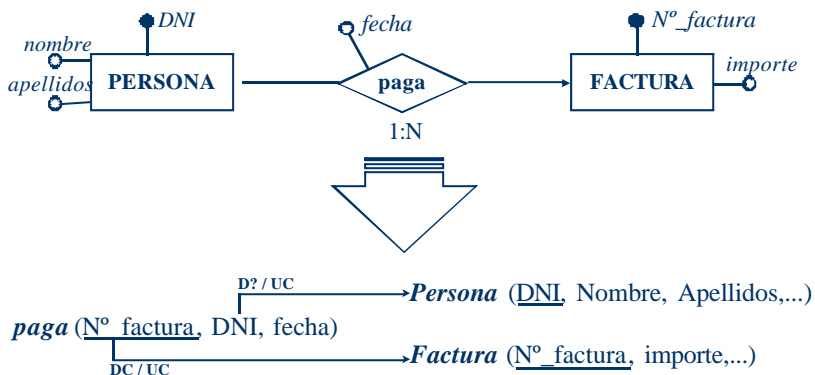
Ejemplo: propagar clave



# Tema 3.2: Transformación Elementos Básicos

## Interrelaciones 1:N

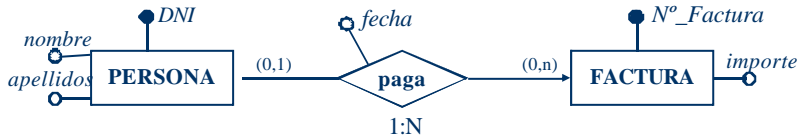
Ejemplo: crear nueva relación



# Tema 3.2: Transformación Elementos Básicos

## Interrelaciones 1:N

¿Qué hacer con las cardinalidades mínimas 'cero'?



- (0,1): la entidad que interviene con una ocurrencia es opcional (pueden existir ocurrencias de la otra entidad no relacionadas).

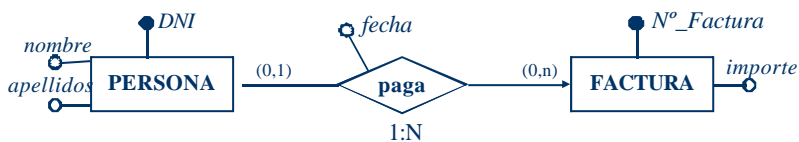
Resultado:

- si se propaga, la clave ajena propagada será opcional (\*) (habrá ocurrencias de la otra entidad cuya clave ajena sea NULL)
- si se crea una nueva relación, habrá identificadores de la entidad opcional que no aparezcan en ninguna ocurrencia de esta relación. (no hay que hacer nada)

# Tema 3.2: Transformación Elementos Básicos

## Interrelaciones 1:N

¿Qué hacer con las cardinalidades mínimas 'cero'?



- (0,n): la entidad que interviene con varias ocurrencias es opcional. (pueden existir ocurrencias de la otra entidad no relacionadas).

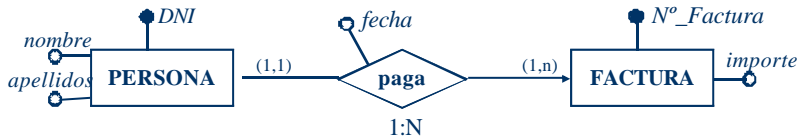
Resultado: no hay que tomar ninguna medida especial

- si se propaga, habrá ocurrencias de la clave de identificación propagada que no aparezcan en ninguna tupla de la otra relación.
- si se crea una nueva relación, habrá identificadores de la entidad opcional que no aparezcan en ninguna ocurrencia de esta relación.

# Tema 3.2: Transformación Elementos Básicos

## Interrelaciones 1:N

¿Qué hacer con las cardinalidades mínimas 'uno'?



- (1,1): la entidad interviene con una y solo una ocurrencia.

Resultado:

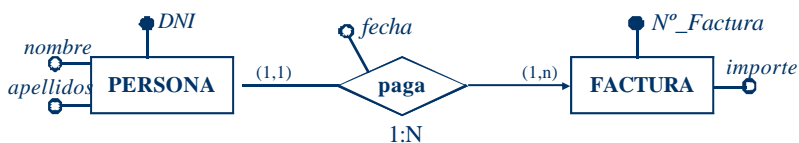
- si se propaga, la clave ajena tomará siempre un valor (obligatoriedad).
- si se crea una nueva relación, en ella debería haber una tupla por cada ocurrencia de la relación que interviene con N (pero esto no quedará garantizado a priori; habría una pérdida de semántica).

Nota: estas pérdidas de semántica se suplirán con la inclusión de restricciones semánticas; en concreto, se añadirán aserciones que vigilen que estas condiciones se cumplen.

# Tema 3.2: Transformación Elementos Básicos

## Interrelaciones 1:N

¿Qué hacer con las cardinalidades mínimas 'uno'?



- (1,n): la entidad interviene con una o varias ocurrencias.

Resultado: no se toma ninguna medida especial (pérdida de semántica).

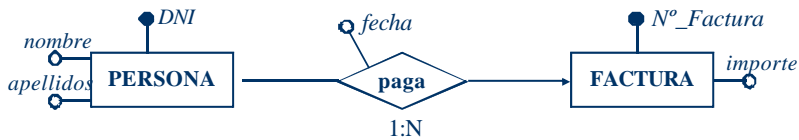
- si se propaga, toda valor de la clave propagada debería aparecer en alguna ocurrencia de la otra relación (pero no se garantiza)
- si se crea una nueva relación, en ella debería haber al menos una tupla por cada ocurrencia de la relación que interviene con una ocurrencia (pero tampoco esto está garantizado a priori)



## Tema 3.2: Transformación Elementos Básicos

### Interrelaciones 1:N

¿Cuál de las opciones es más conveniente?

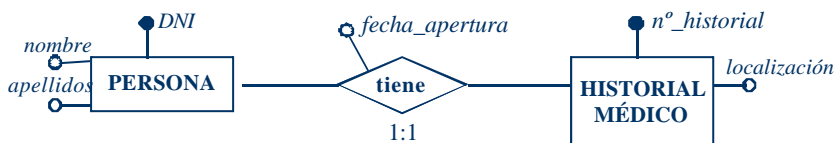


- En general, es preferible propagar la clave
- Se creará una nueva relación si:
  - a) la interrelación tiene caracterización propia (atributos propios)
  - b) se prevé que la interrelación podría ser N:M en el futuro
  - c) la cardinalidad mínima de la interrelación para la entidad que propaga es 0 (opcional), y en la otra entidad el número de ocurrencias no relacionadas es elevado (la clave ajena sería NULL)

## Tema 3.2: Transformación Elementos Básicos

### Interrelaciones 1:1

Se puede considerar un caso particular de las anteriores, y por tanto las soluciones anteriores son válidas también en este caso.



Las soluciones aplicables son:

- a) crear una nueva relación ←
- b) propagar una de las claves ←
- c) propagar las claves de las dos entidades (mutuamente)
- d) fusionar ambas entidades (interrelacionadas) en una sola relación

# Tema 3.2: Transformación Elementos Básicos

## Interrelaciones 1:1

Crear una nueva relación: (justificación similar al caso 1:n)

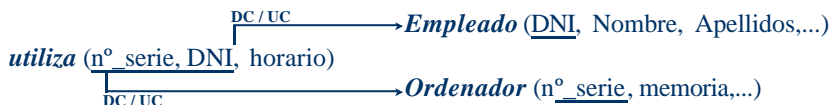
- a) si las cardinalidades mínimas son cero (ambas), esto evitará valores nulos en las claves ajenas y mantendrá la simetría natural (entidades mantienen su independencia en relaciones separadas)
- b) si la interrelación tiene caracterización propia (atributos) o
- c) si se prevé que posteriormente puedan variarse las cardinalidades



# Tema 3.2: Transformación Elementos Básicos

## Interrelaciones 1:1

Ejemplo: (crear una nueva relación)



Nota: observar la pérdida de eficiencia, ya que muchas consultas implican combinar dos relaciones, e incluso hay consultas que implican combinar las tres relaciones.

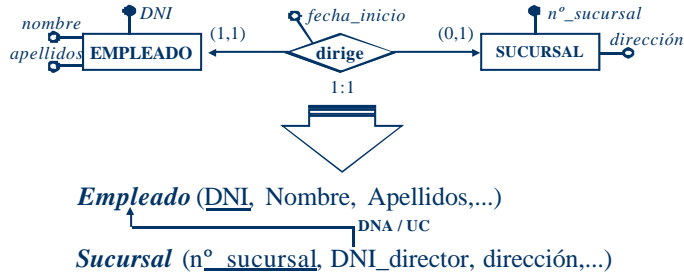
## Tema 3.2: Transformación Elementos Básicos

### Interrelaciones 1:1

Propagar una clave: (*justificación similar a la anterior*)

- si **una** de las cardinalidades mínimas es cero y la otra no (será 1,1), conviene propagar la clave de esta última (la obligatoria).

**Ejemplo:**



## Tema 3.2: Transformación Elementos Básicos

### Interrelaciones 1:1

Propagar una clave:

- Inconvenientes:
  - se pierde la simetría
  - consultas a la información de la entidad que interviene con (1,1) suponen combinación natural (por ejemplo, empleados que no dirigen sucursal)
- Ventajas:
  - no pierde semántica (sobre la cardinalidad mínima 1)
  - se evitan valores nulos
  - algunas consultas no precisan combinación de relaciones
- **NOTA:** observar que la opción de borrado debe ser restringido o en cascada

## Tema 3.2: Transformación Elementos Básicos

### Interrelaciones con atributos:

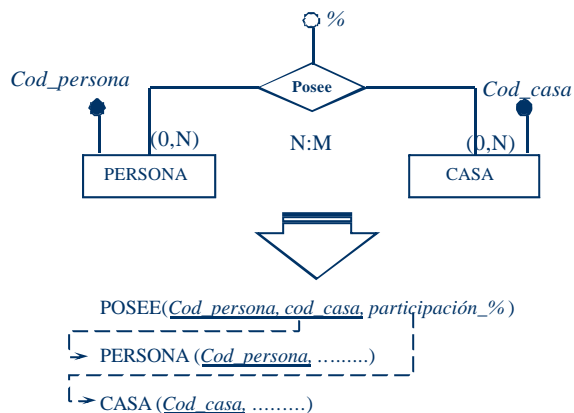
- si se crea una nueva relación, esos atributos se incluyen en esta relación.
- si se propaga una clave, los atributos acompañan a la clave.
- si se propagan ambas claves, los atributos se incluyen en una de las entidades interrelacionadas.
- si se fusionan en una relación, esta también contendrá esos atributos.

### Interrelaciones con un atributo multivaluado:

- la interrelación habrá de transformarse en una relación, y su clave deberá contener ese atributo (además de la clave de una de las entidades o las dos)

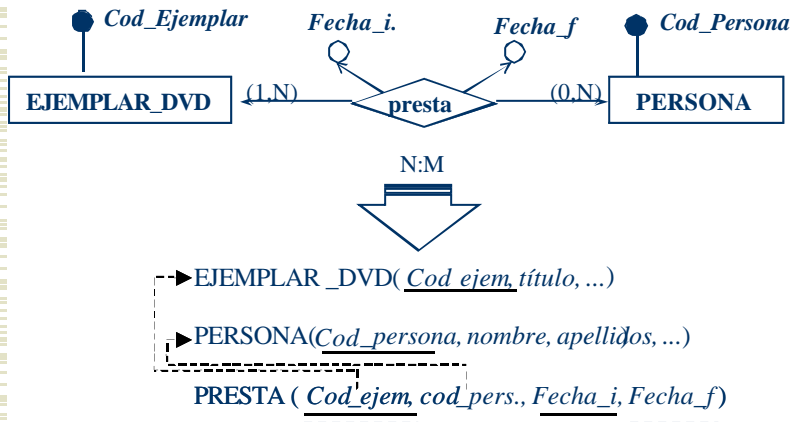
## Tema 3.2: Transformación Elementos Básicos

### Ejemplo Interrelación con atributo



## Tema 3.2: Transformación Elementos Básicos

### Ejemplo Interrelación con atributos multivaluados

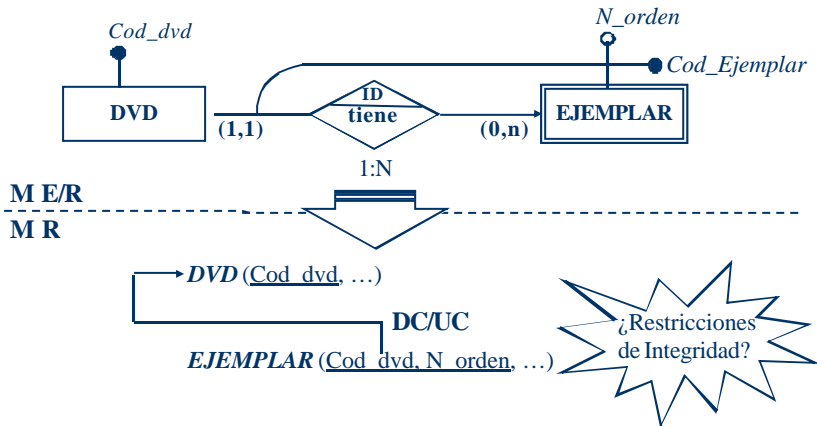


## Tema 3.3: Transformación de las extensiones del E/R

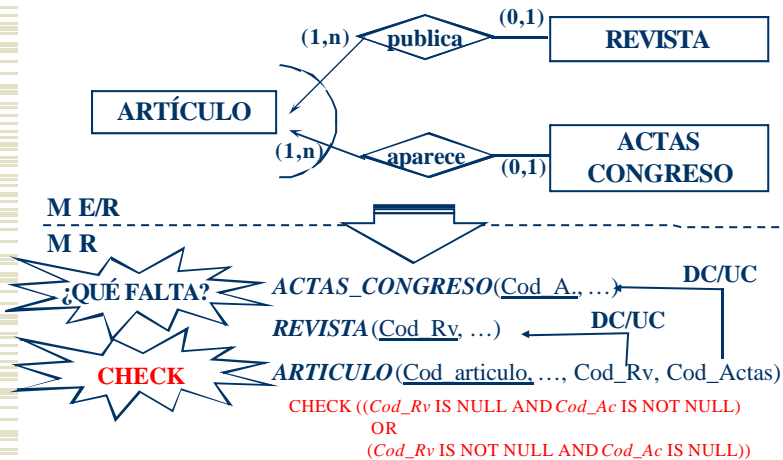
- 5.3.1 Dependencias en Existencia/Identificación
- 5.3.2 Interrelaciones Exclusivas
- 5.3.2 Generalizaciones
- 5.3.4 Dimensión Temporal
- 5.3.5 Atributos Derivados
- 5.3.6 Interrelaciones de grado superior a 2

## Tema 3.3.1: Dependencias en Existencia/Identificación

Se transforman de la misma forma que las interrelaciones 1:N

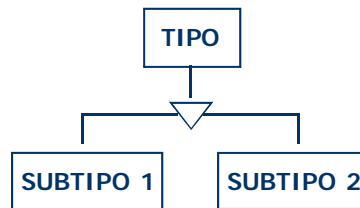


## Tema 3.3.2: Interrelaciones Exclusivas



## Tema 3.3.3: Generalizaciones

- ♦ El Modelo Relacional no dispone de instrumentos que permitan representar tipos y subtipos.
- ♦ Se definen distintos métodos de transformación, dependiendo de los objetivos perseguidos:
  - Información semántica representada en el modelo
  - Eficiencia de acceso a los datos



## Tema 3.3.3: Generalizaciones

### ♦ Método 1

Utilizar una **única relación** para representar un tipo y todos sus subtipos, añadiendo un atributo que indique el tipo de entidad al que se hace referencia.

Puede hacerse cuando:

- los **atributos** de los subtipos son **similares**
- las **interrelaciones** que involucren a los subtipos son las mismas (o no existen)

Será necesario implementar las restricciones semánticas necesarias a través de CHECKS o DISPARADORES.

## Tema 3.3.3: Generalizaciones

### ♦ Método 1

Según los parámetros de la jerarquía, ¿qué restricciones hay que definir?

#### Solapamiento

Sí ? El atributo discriminante puede tomar varios valores combinados.

No ? Verificar que sólo los atributos adecuados al subtipo toman valores.

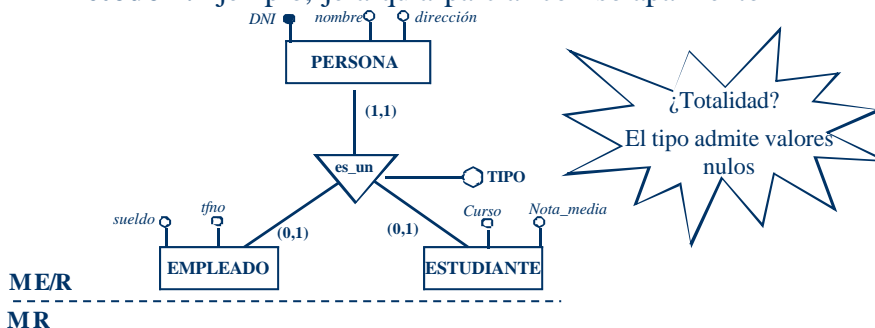
#### Totalidad

Sí (Total) ? El atributo discriminante no puede tomar valores nulos.

No (Parcial) ? El atributo discriminante debe admitir valores nulos.

## Tema 3.3.3: Generalizaciones

### ♦ Método 1. Ejemplo, jerarquía parcial con solapamiento



*PERSONA* (DNI, nombre, dirección, tipo\*, sueldo\*, teléfono\*, Curso\*, Nota\_media\*)



## Tema 3.3.3: Generalizaciones

### ♦ Método 1. Ejemplo, Restricción exclusividad

```
Check (( Tipo= 'Emp'  
        AND curso IS NULL  
        AND nota_media IS NULL  
        AND sueldo IS NOT NULL  
        AND tfno IS NOT NULL  
  
OR  
        ( Tipo = 'Est'  
        AND sueldo IS NULL  
        AND tfno IS NULL  
        AND curso IS NOT NULL  
        AND nota_media IS NOT NULL ))
```

## Tema 3.3.3: Generalizaciones

### ♦ Método 2

Utilizar una **relación** para representar al **supertipo** y **tantas relaciones como subtipos** haya. Como antes habrá que añadir un atributo que indique el tipo de entidad al que se hace referencia.

Puede hacerse cuando:

- los subtipos tienen **atributos dispares y/o interrelaciones diferentes**
- Incorporar **mayor semántica** en el grafo relacional

Será necesario implementar las restricciones semánticas necesarias a través de CHECKS o DISPARADORES.

## Tema 3.3.3: Generalizaciones

### ♦ Método 2

Según los parámetros de la jerarquía, ¿qué restricciones hay que definir?

#### Solapamiento

Sí ? El atributo discriminante puede tomar varios valores combinados.

No ? Verificar que sólo aparecen entradas en la relación del subtipo correspondiente.

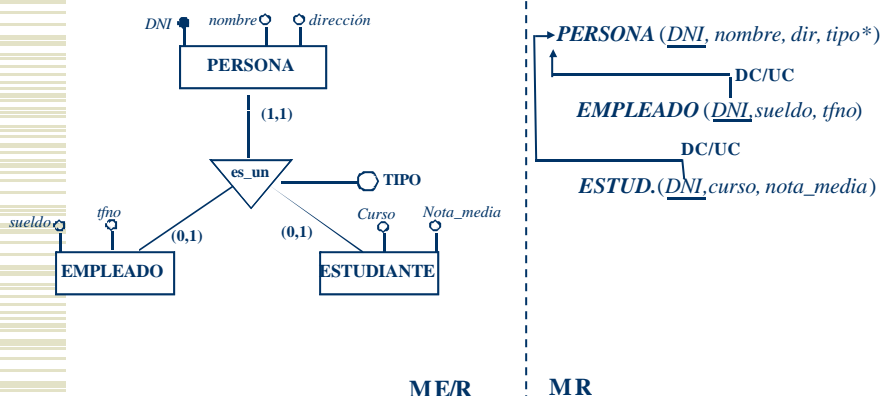
#### Totalidad

Sí (Total) ? El atributo discriminante no puede tomar valores nulos y es necesario verificar que hay entradas para todas las tuplas del tipo.

No (Parcial) ? El atributo discriminante debe admitir valores nulos.

## Tema 3.3.3: Generalizaciones

### ♦ Método 2. Ejemplo, jerarquía parcial con solapamiento



## Tema 3.3.3: Generalizaciones

### ♦ Método 3

Se emplea una relación para cada subtipo; cada una de ellas incluye los atributos comunes asociados al tipo.

Puede hacerse cuando:

- los subtipos tienen **atributos dispares y/o interrelaciones diferentes**
- La mayoría de los accesos a los datos de los subtipos involucran en mayor medida a los atributos comunes? **Eficiencia**

Será necesario implementar las restricciones semánticas necesarias a través de CHECKS o DISPARADORES.

## Tema 3.3.3: Generalizaciones

### ♦ Método 3

Según los parámetros de la jerarquía, ¿qué restricciones hay que definir?

#### Solapamiento

Sí? Nada que controlar.

No? Verificar que sólo aparecen entradas en la relación del subtipo correspondiente.

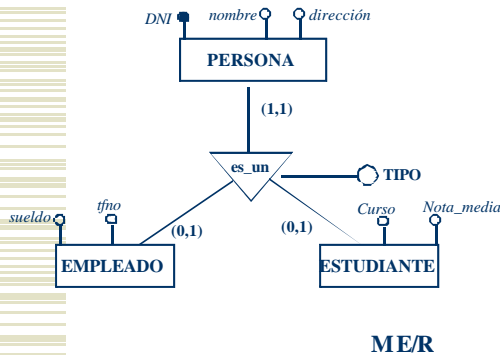
#### Totalidad

Sí (Total)? Nada que controlar.

No (Parcial)? NO PUEDE UTILIZARSE ESTA TRANSFORMACIÓN

## Tema 3.3.3: Generalizaciones

- ♦ **Método 3.** Ejemplo, jerarquía parcial con solapamiento



*EMP(DNI, nombre, dir, sueldo, tfno)*

*EST(DNI, nombre, dir, curso, nota\_media)*

ME/R

MR

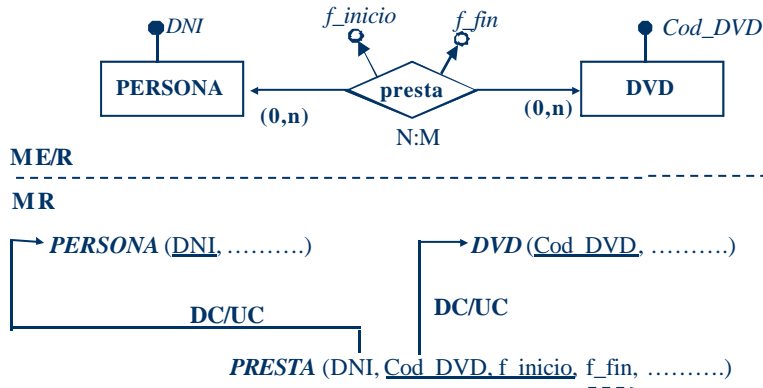
## Tema 3.3.3: Generalizaciones

### EJEMPLOS

- ♦ Una empresa de estudios forestales desea almacenar en una base de datos información sobre sus empleados, que pueden ser administrativos u operarios de campo. Estos datos serán DNI, nombre, apellidos, fecha de contrato y fecha de baja. Además, para el caso de los operarios es necesario almacenar el coste por hora, así como el número de horas trabajadas.
- ♦ Los operarios de campo tienen la misión de tomar medidas sobre determinadas parcelas y los administrativos serán los encargados de grabar los datos de los formularios rellenados por los operarios de campo.

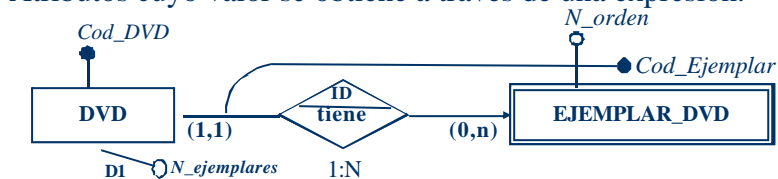
## Tema 3.3.4: Dimensión Temporal

Se emplean las reglas de transformación de atributos multivaluados.



## Tema 3.3.5: Atributos Derivados

Atributos cuyo valor se obtiene a través de una expresión.



ME/R

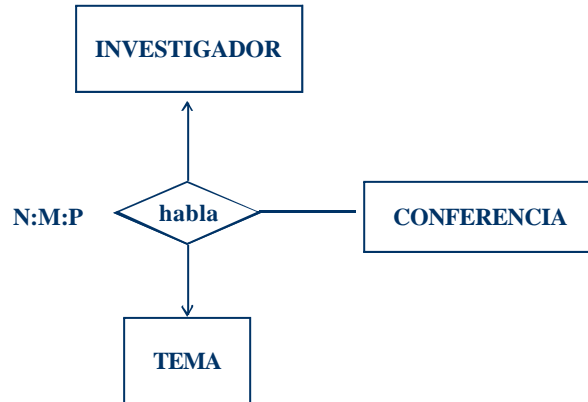
MR

**DVD** (Cod\_DVD, Título, N\_ejemplares)

Será necesario incluir un disparador para el cálculo del atributo

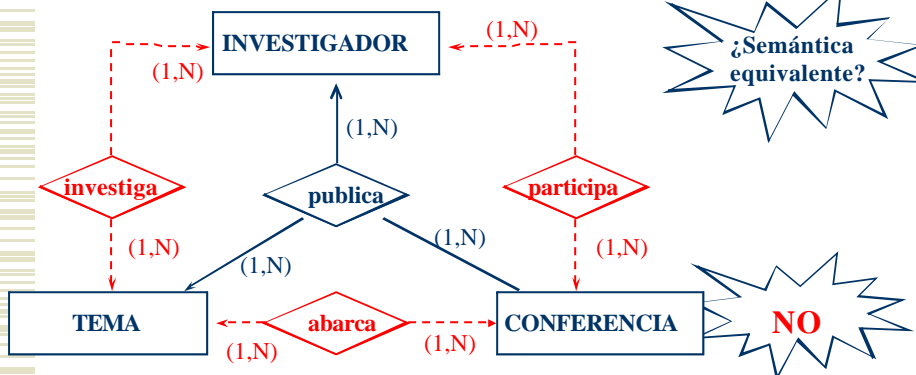
## Tema 3.3.6: Interrelaciones Ternarias

En el modelo E/R se permiten interrelaciones entre más de dos entidades.



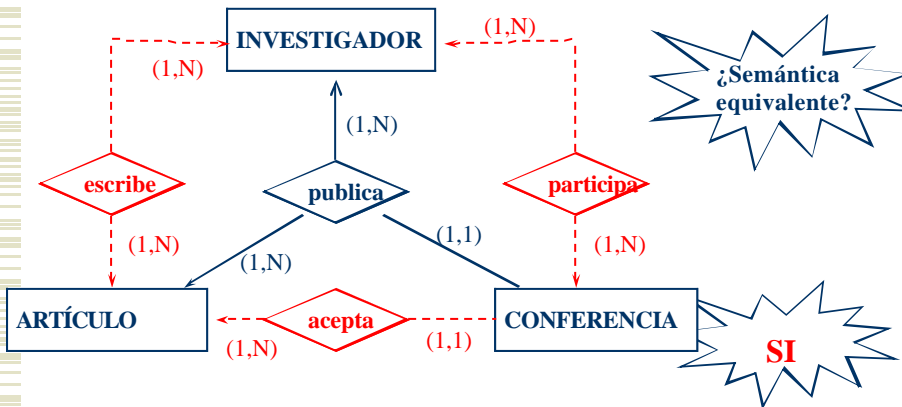
## Tema 3.3.6: Interrelaciones Ternarias

Siempre que sea posible, este tipo de interrelaciones se representarán a través de varias interrelaciones binarias.



## Tema 3.3.6: Interrelaciones Ternarias

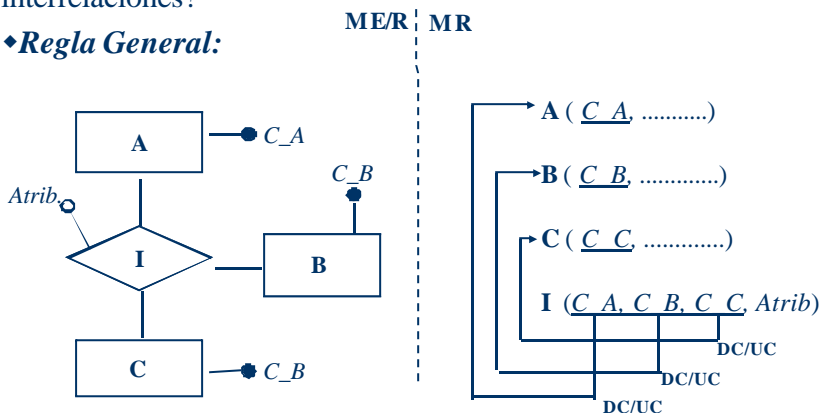
Otro ejemplo:



## Tema 3.3.6: Interrelaciones Ternarias

¿Cómo se transforma al Modelo Relacional este tipo de interrelaciones?

♦ **Regla General:**



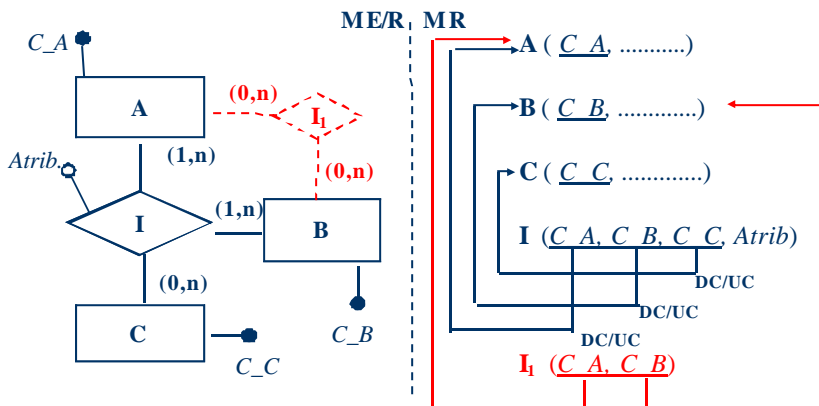
## Tema 3.3.6: Interrelaciones Ternarias

- ♦ Es necesario un análisis más profundo teniendo en cuenta las cardinalidades de la interrelación.
- ♦ Caso 1. Cardinalidad máxima  $n$  y mínima  $1$  en todas las ramas de la interrelación:

Es aplicable la regla general.

## Tema 3.3.6: Interrelaciones Ternarias

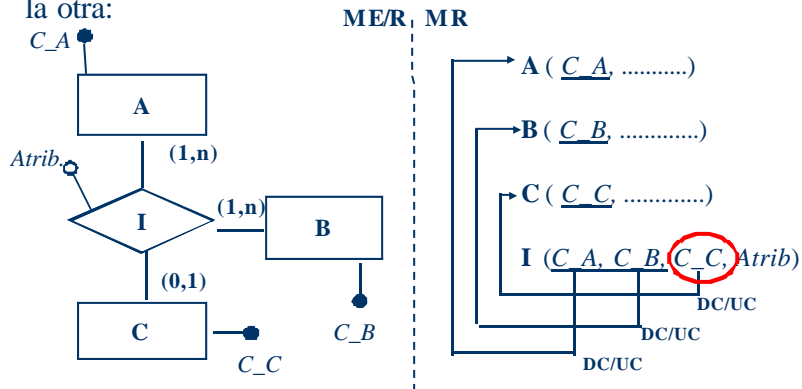
- ♦ Caso 2. Cardinalidad máxima  $n$  y mínima  $0$  en una rama:





## Tema 3.3.6: Interrelaciones Ternarias

- ◆ Caso 3. Cardinalidad máxima n en dos ramas y máxima 1 en la otra:



## Bibliografía

- ◆ **BÁSICA:**

- [1] D. Cuadra, E. Castro, A. Iglesias, P. Martínez, F.J. Calle, C. de Pablo, H. Al-Jumaily y L. Moreno. Desarrollo de Bases de Datos: casos prácticos desde el análisis a la implementación. Capítulo 2. RA-MA. 2007.
- [2] M. Piattini, E. Marcos, C. Calero y B. Vela. Tecnología y Diseño de Bases de Datos. Capítulos 6 y 16. RA-MA 2006.

- ◆ **RECOMENDADA:**

- [3] A. Silberschatz, H. Korth & S. Sudarskhan. Fundamentos de Bases de Datos. 5ª Edición. Capítulo 7. McGraw Hill. 2006.
- [4] R. Elmasri and S.B. Navathe. Fundamentos de Sistemas de Bases de Datos. Capítulo 3. Addison Wesley. 2007.
- [5] A. de Miguel, M. Piattini y E. Marcos. Diseño de Bases de Datos Relacionales. Capítulo 3. RA-MA. 1999.
- [6] A. de Miguel, P. Martínez, E. Castro, J.M. Cavero, D. Cuadra, A. Iglesias y C. Nieto. Diseño de Bases de Datos: Problemas Resueltos. Capítulo 2. RA-MA. 1999.