

# Tema 2. Bases de datos relacionales.

---

Parte II. Construcción del modelo  
relacional o Paso a Tablas

# Índice

## 1. Normas para el paso a tablas del modelo relacional.

- Traducción de las entidades.
- Traducción de las relaciones 1:1
- Traducción de las relaciones 1:N
- Traducción de las relaciones N:N
- Traducción de las relaciones de dependencia.
- Traducción de las relaciones N-arias.
- Traducción de las relaciones reflexivas.

## 2. Eliminación de las relaciones jerárquicas.

# 1. Normas para el paso a tablas del modelo relacional.

- Para realizar el paso a tablas se parte del modelo E/R. **Cada entidad de este modelo dará lugar a ninguna o una tabla del modelo relacional.** Se debe tratar de obtener el menor número de tablas posible para evitar redundancia.
- Para describir las normas es necesario conocer el concepto de **CLAVE AJENA**.
- La **CLAVE AJENA** de una tabla es el atributo o conjunto de atributos que forman la clave de otra tabla. Que un atributo sea clave ajena en una tabla significa que **para introducir datos en esa tabla, previamente han debido introducirse en la tabla origen.**

# 1. Normas para el paso a tablas del modelo relacional.

## Ejemplo:

Dadas las tablas A y B

A(X#,Y,Z,K)

B(K#,L,M)

La tabla A tiene como **clave el atributo X** y la tabla B el atributo **K**.

**La tabla A tiene como clave ajena el atributo K**, puesto que dicho atributo es clave de B. Esto significa que para introducir un dato en el atributo K de A, previamente hemos debido introducir ese dato (ocurrencia) en B.

Las claves ajenas se utilizan para establecer referencias entre las tablas. A través de la clave se referencia a toda la ocurrencia de otra tabla.

# Traducción de las entidades.

- Cada entidad se transforma en una tabla.

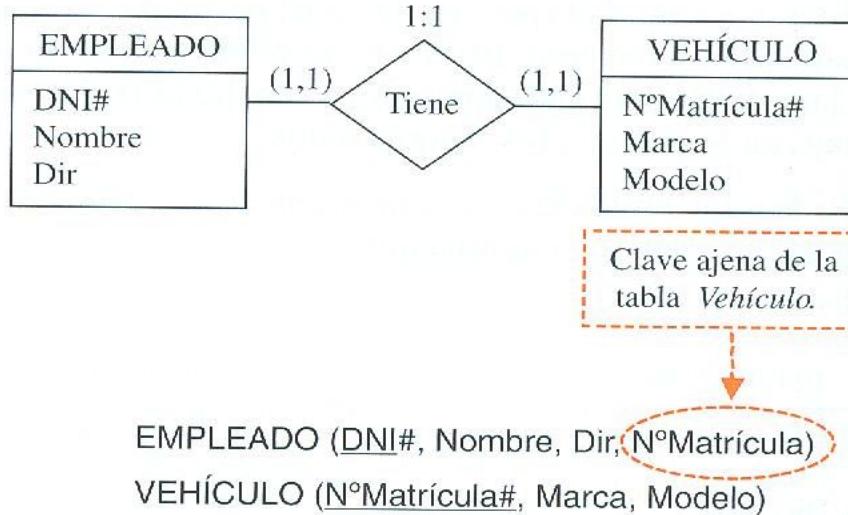


La entidad *Empleado* se transforma en una tabla con todos sus campos<sup>1</sup>, la clave aparece subrayada.

<sup>1</sup> en el modelo relacional a los atributos se les denomina campos

# Traducción de las relaciones 1:1

- Por lo general **no generan tabla**.
- La clave de una entidad **se introduce como clave ajena** en la tabla de la otra entidad.



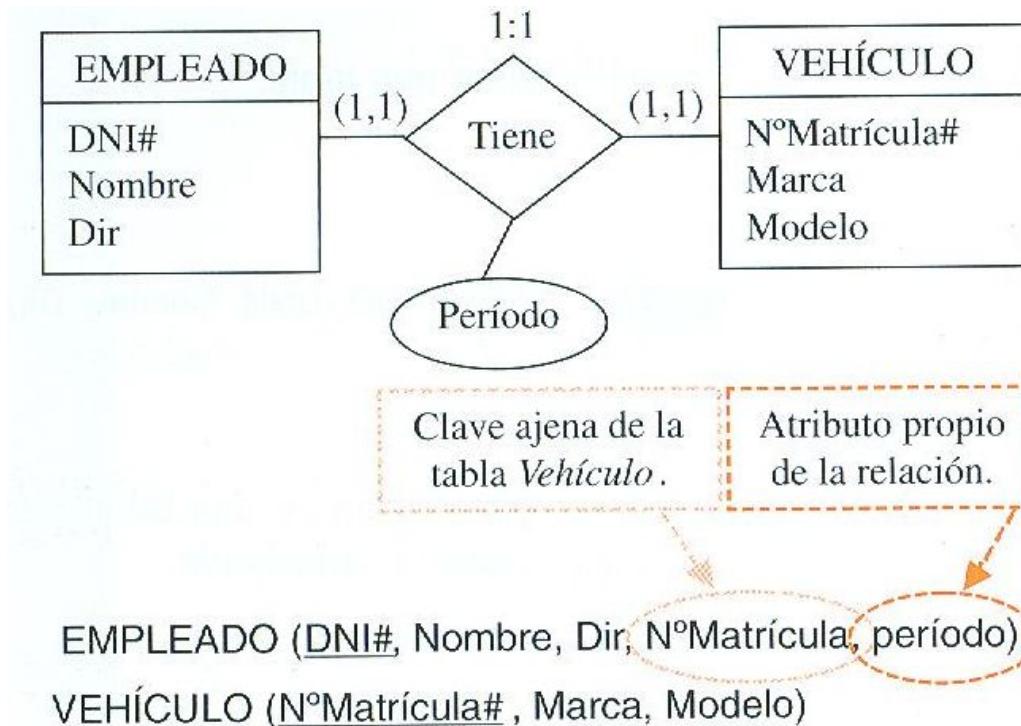
Hay diferentes posibilidades:

- Introducir la clave de *Empleado* como clave ajena en *Vehículo*.
- Introducir la clave de *Vehículo* como clave ajena en *Empleado*.
- Introducir ambas como clave ajena en la otra tabla. Es decir, la clave de *Vehículo* como clave

Se elegirá una u otra opción en función de cómo se quiera organizar las consultas

# Traducción de las relaciones 1:1. Atributos Propios.

Los atributos propios de la relación pasan a la tabla donde se introduce la clave ajena

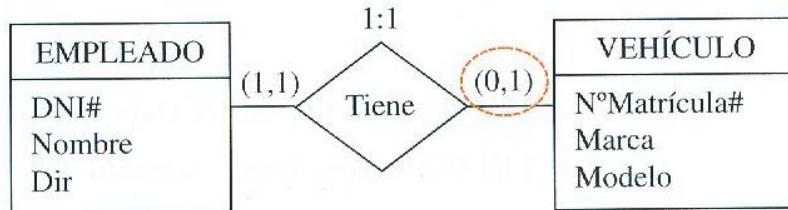


El atributo propio *periodo* pasa a la tabla *Empleado*.

# Traducción de las relaciones 1:1. Participación mínima 0.

Si alguna entidad (pero no las dos) participa con cardinalidad mínima 0 (0,1), se pone la clave ajena en la entidad que participa con cardinalidad (0,1) para evitar, en lo posible, los valores nulos.

Si las dos entidades participan con (0,1) se crea una nueva tabla para la relación.



Por ejemplo, supongamos que un empleado puede o no tener coche. En este caso se generan las siguientes tablas:

EMPLEADO (DNI#, Nombre, Dir)

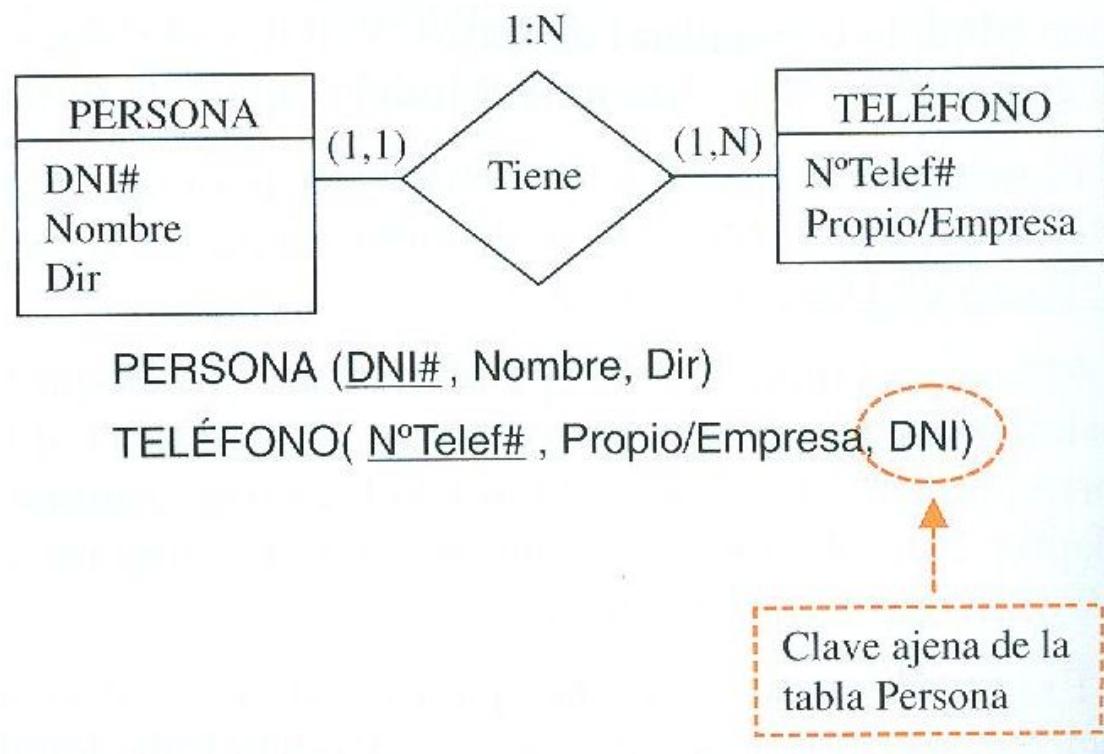
VEHÍCULO (NºMatrícula#, Marca, Modelo, DNI)

**La clave de la entidad empleado que participa con (1,1) pasa a la tabla de la entidad vehículo que participa con (0,1). De esta forma se evitan valores nulos. Como todo vehículo está asignado a un empleado, el campo DNI de vehículo se rellena siempre. Si se hiciera al revés (la clave de vehículo en la tabla Empleado), como hay empleados que no tienen vehículo a veces quedaría ese campo vacío y eso hay que tratar de evitarlo.**

# Traducción de las relaciones 1:N

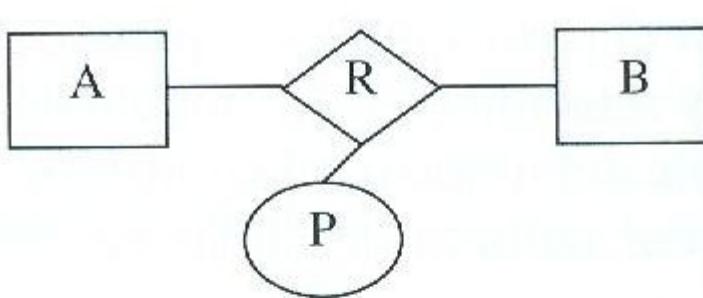
Por lo general **no generan tabla**.

La clave de la entidad 1 ( la que participa con cardinalidad máxima 1) pasa como clave ajena a la tabla de la entidad N (la que participa con cardinalidad máxima N).



## Traducción de las relaciones 1:N. Atributos propios.

Pasan a la tabla donde se ha incorporado la clave ajena.



A (a1#, a2, a3, ...)

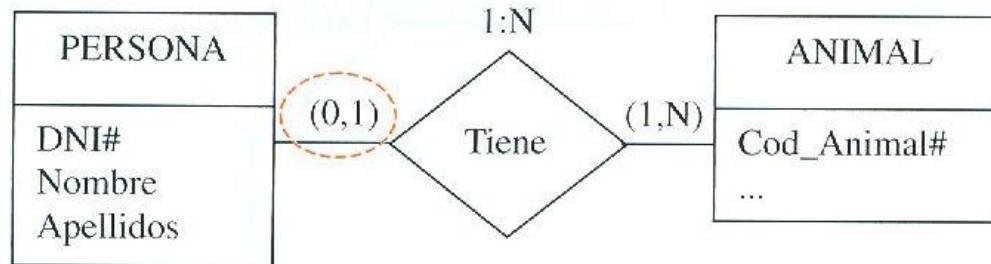
B (b1#, b2, b3, ..., a1, P)

El atributo propio de la relación  
pasa junto con la clave ajena.

# Traducción de las relaciones 1:N. Participación mínima 0.

Hay dos posibilidades.

- Si esta en el lado N , (0,N), se hace como se ha dicho antes.
- Si está en el lado 1, (0,1) , se crea la tabla nueva para la relación, con las claves de ambas entidades. La clave de la relación será solo la clave de la entidad N



PERSONA (DNI#, Nombre, Apellidos)

ANIMAL (Cod\_Animal#,...)

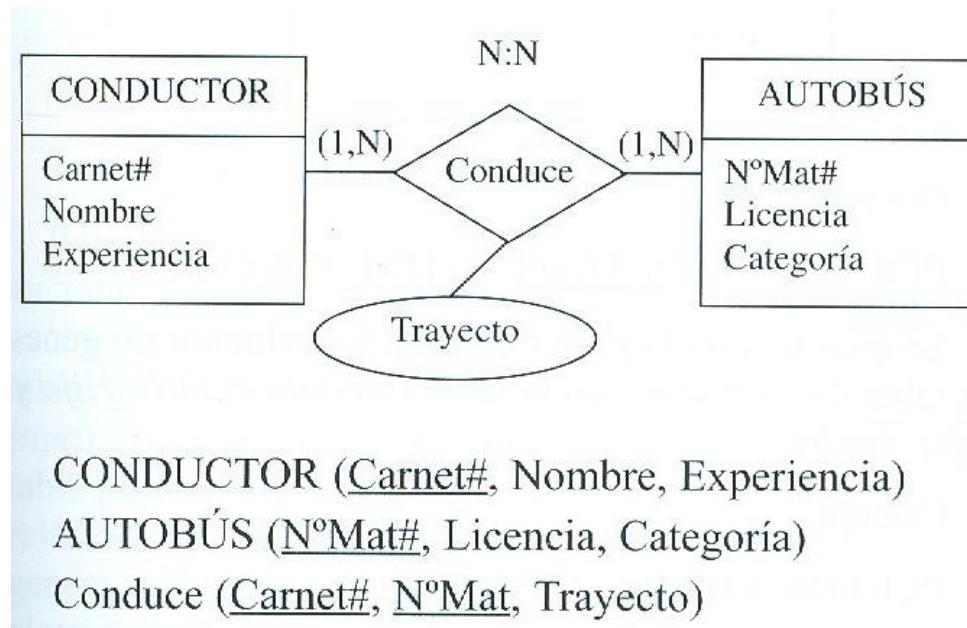
Tiene (Cod\_Animal#, DNI)

Las entidades *Persona* y *Animal* generan tabla con todos sus atributos. La relación *Tiene* genera tabla puesto que la participación de *Persona* es (0,1).

# Traducción de las relaciones N:N

Por lo general **siempre generan tabla**.

Se crea una tabla cuya clave está formada por las claves de las entidades que participan en la relación. Si hay atributos propios pasan a la tabla de la relación.



Si hubiera participaciones mínimas 0 se haría igual.

## Traducción de las relaciones N:N. Orden de los atributos en las claves compuestas.

- Se deben poner a la izquierda todos los atributos que forman la clave.
- El orden de los atributos que forman la clave vendrá determinado por las consultas que se vaya a realizar. Las ocurrencias suelen estar ordenadas como índice la clave. Por tanto conviene poner primero aquel o aquellos atributos por los que se vaya a realizar la consulta.
- Por ejemplo, supongamos que queremos hacer consultas para el ejemplo anterior.
  - Obtener los trayectos del conductor de carnet 111
  - Obtener los nº de matrícula de todos los autobuses del conductor 555.

### Nota.

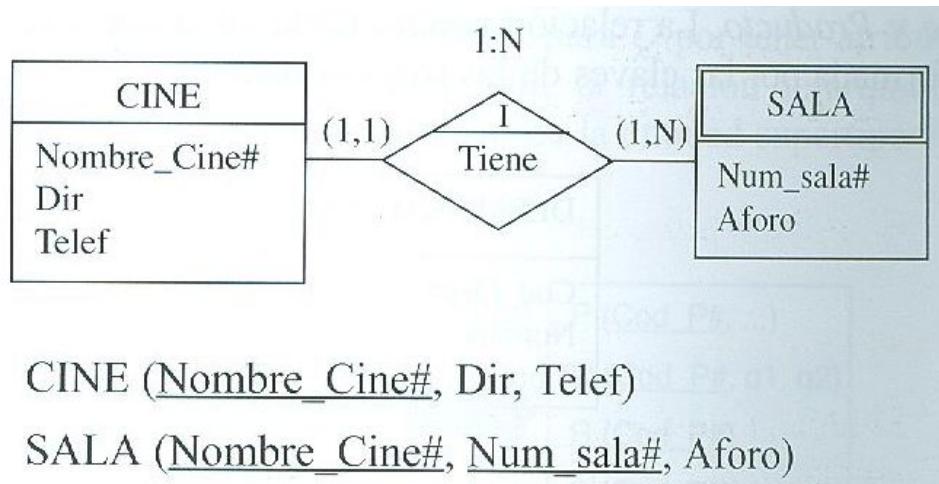
No siempre se pueden saber cuales van a ser todas las consultas que se hagan en la base de datos, pero si se puede tener alguna idea.

# Relaciones de dependencia

## ■ Relación de dependencia en identificación.

Por lo general suelen ser del tipo 1:1 o 1:N, por lo que no generan tabla.

La clave de la entidad fuerte debe introducirse en la tabla de la entidad débil y formar parte de clave de esta. Además debe ir la primera ( la de la entidad fuerte).

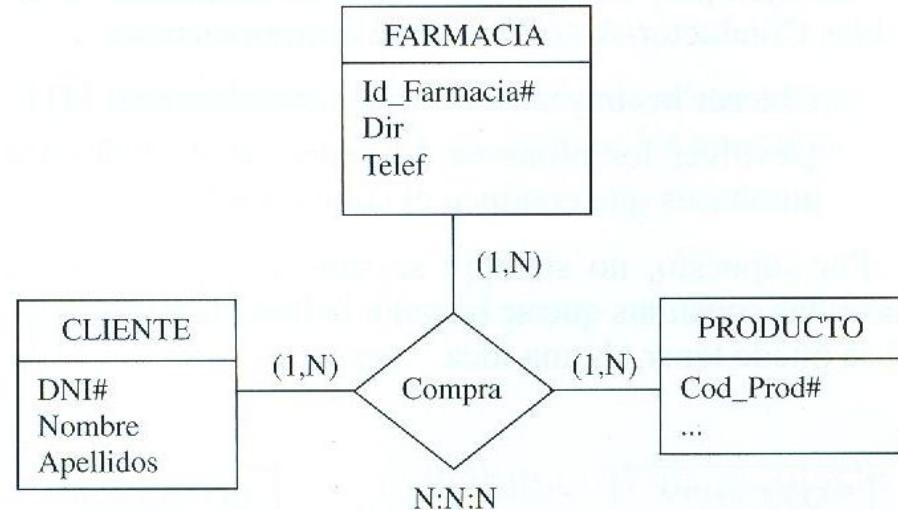


## ■ Relación de dependencia en existencia.

Generan tabla o no dependiendo de la cardinalidad. Se actúa como se indica en cada caso.

# Relaciones N-arias

- Por lo general siempre generan tabla.
- Cuando hay más de dos entidades participando en la relación se crea una nueva tabla, que tenga por **clave las claves de las entidades que participan en la relación**. Y si hay atributos propios se incluyen en esa tabla.
- Si una entidad participa con cardinalidad máxima 1, puede que no tenga que formar parte de la clave de la tabla de la relación N-aria.



FARMACIA (Id\_Farmacia#, Dir, Telef)

CLIENTE (DNI#, Nombre, Apellidos)

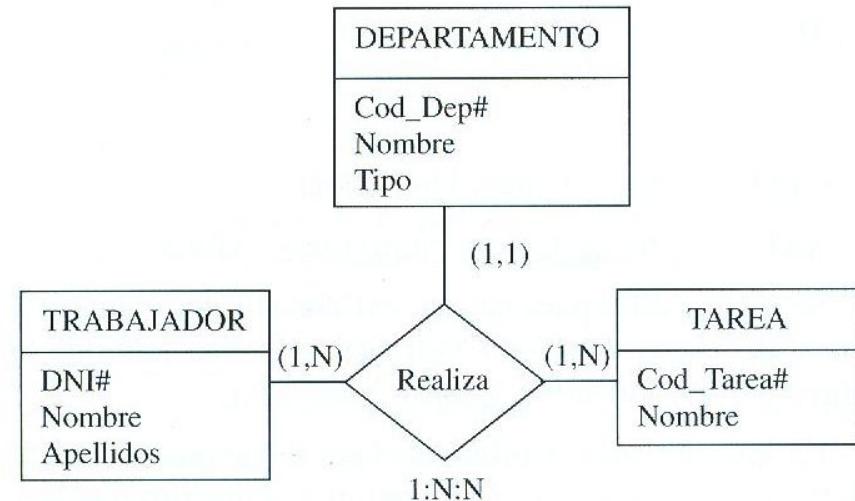
PRODUCTO (Cod\_Prod#, ...)

Compra (Id\_Farmacia#, DNI#, Cod\_Prod#)

# Relaciones N-arias

## Ejemplo 2.

Cada trabajador realiza cada tarea en un único departamento. No es necesario que el departamento forme parte de la clave de la tabla Realiza. Tan sólo con DNI y Cod\_tarea se puede identificar de manera única la ocurrencia de la tabla Realiza.



DEPARTAMENTO (Cod\_Dep#, Nombre, Tipo)

TRABAJADOR (DNI#, Nombre, Apellidos)

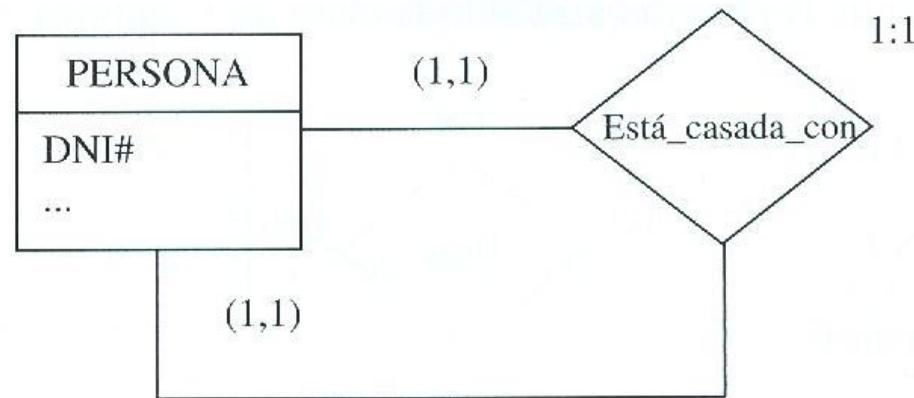
TAREA (Cod\_Tarea#, Nombre)

Realiza (DNI#, Cod\_Tarea#, Cod\_Dep)

# Relaciones reflexivas o recursivas.

- Generan tabla o no según la cardinalidad.

**Cardinalidad 1:1**

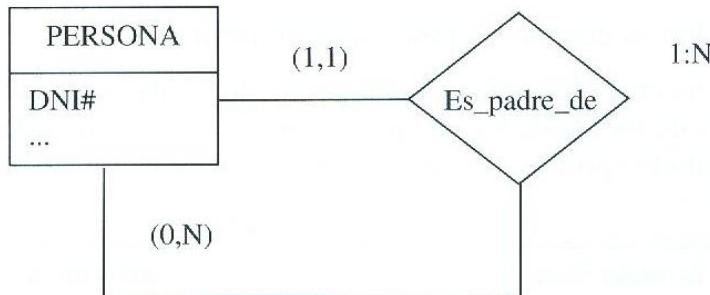


PERSONA (DNI#, ..., DNI\_Cónyuge)

Se crea una tabla para la entidad *Persona*. La relación no genera tabla por tener cardinalidad 1:1. En la tabla *Persona* se introducen dos veces el *DNI*, uno es clave y el otro no. Para distinguirlos, uno se llama *DNI* y el otro *DNI\_Cónyuge*.

# Relaciones reflexivas o recursivas.

## Cardinalidad 1:N



Opción 1:

PERSONA (DNI\_Hijo#, ..., DNI\_Padre)

Se crea una tabla para *Persona*. La relación no genera tabla. Se introduce en la tabla *Persona* el *DNI\_Hijo* y *DNI\_Padre*.

Opción 2:

PERSONA (DNI#, ...)

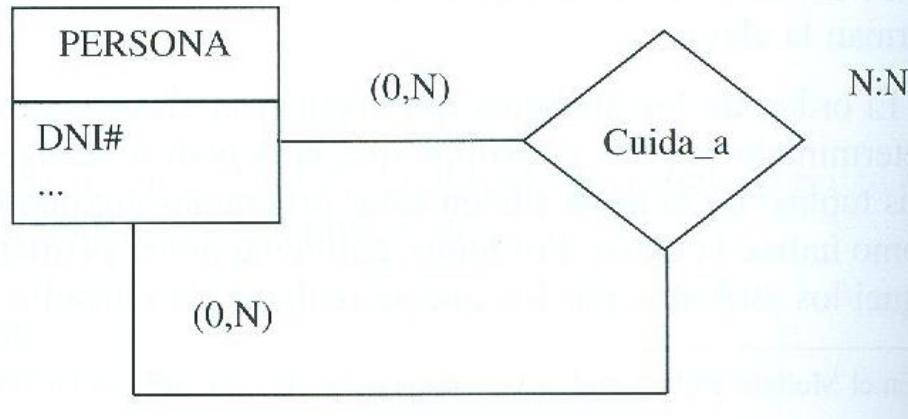
Es\_padre\_de (DNI\_Hijo#, DNI\_Padre)

Se crea una tabla para *Persona*. La relación sí genera tabla con *DNI\_Hijo* y *DNI\_Padre*. Sólo es clave el *DNI\_Hijo*, puesto que todo hijo tiene como padre a una y sólo una persona.

Si en lugar de la participación (1,1), fuera (0,1), obligatoriamente se tendría la segunda opción.

# Relaciones reflexivas o recursivas.

Cardinalidad N:N



**PERSONA** (DNI#, ...)

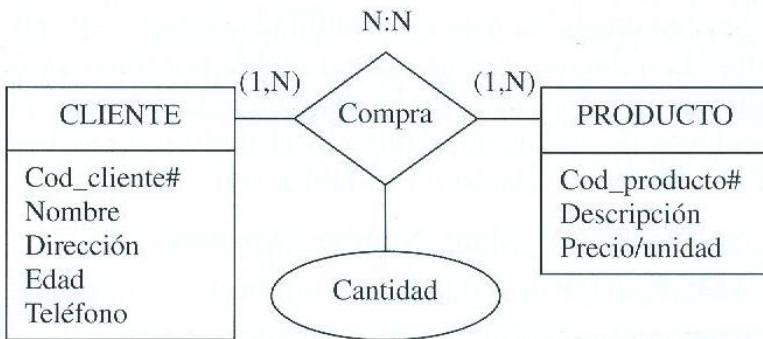
**Cuida\_a** (DNI\_Cuidador#, DNI\_Paciente#)

Se crea una tabla para la entidad *Persona* y otra para la relación. La clave de la relación es *DNI\_Cuidador* y *DNI\_Paciente*.

# Ejercicios.

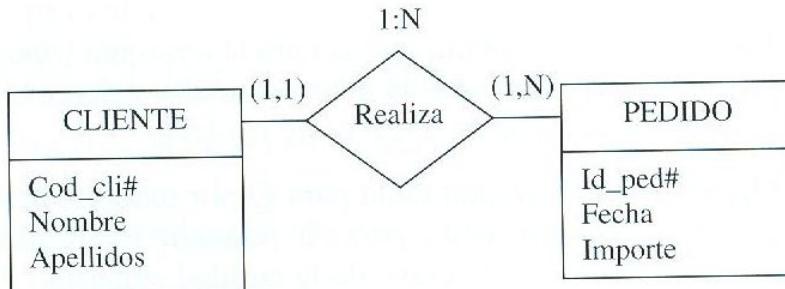
## Ejercicio nº 1:

Realizar el paso a tablas del siguiente modelo E/R:



## Ejercicio nº 2:

Realizar el paso a tablas del siguiente modelo E/R:



## Solución:

Se crea una tabla para cada entidad con todos sus atributos. La relación también genera tabla por tener cardinalidad N:N. La clave de la relación está formada por las claves de ambas entidades. El atributo *cantidad* se pone en la tabla de la relación. Las tablas son:

CLIENTE (Cod\_cliente#, Nombre, Dirección, Edad, Teléfono)

PRODUCTO (Cod\_producto#, Descripción, Precio/unidad)

Compra (Cod\_cliente#, Cod\_producto#, cantidad)

## Solución:

Se crea una tabla para cada entidad con todos sus atributos. La relación no genera tabla por tener cardinalidad 1:N. La clave de *Cliente* (entidad 1) pasa como clave ajena a la tabla de la entidad *Pedido* (entidad N). Las tablas son:

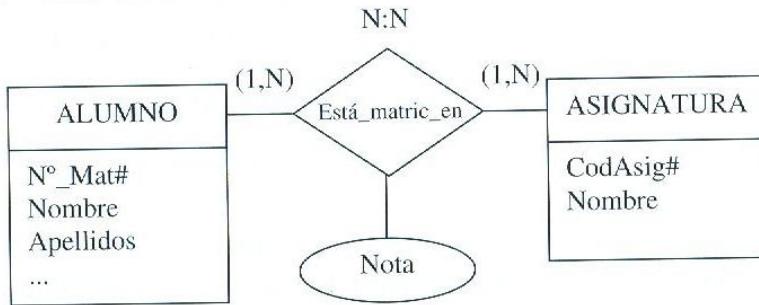
CLIENTE (Cod\_cliente#, Nombre, Apellidos)

PEDIDO (Cod\_pedido#, Fecha, Importe, Cod\_cliente)

# Ejercicios

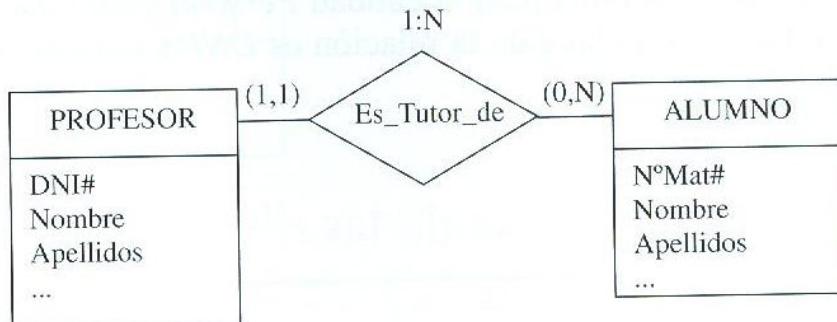
## Ejercicio nº 3:

Realizar el paso a tablas del siguiente modelo E/R:



## Ejercicio nº 4:

Realizar el paso a tablas del siguiente modelo E/R:



## Solución:

Se crea una tabla para cada entidad con todos sus atributos. La relación también genera tabla por tener cardinalidad N:N. La clave de la relación está formada por las claves de ambas entidades. El atributo *nota* se pone en la tabla de la relación. Las tablas son:

ALUMNO (Nº\_Mat#, Nombre, Apellidos,...)

ASIGNATURA (CodAsig#, Nombre)

Está\_matric\_en (Nº\_Mat#, CodAsig#, nota)

## Solución:

Se crea una tabla para cada entidad con todos sus atributos. La relación no genera tabla por tener cardinalidad 1:N y la participación mínima cero se da en el lado de la entidad N (*Alumno*). La clave de la entidad 1 (*Profesor*) pasa a la tabla de la entidad N (*Alumno*), como clave ajena. Las tablas son:

PROFESOR (DNI#, Nombre\_P, Apellidos\_P,...)

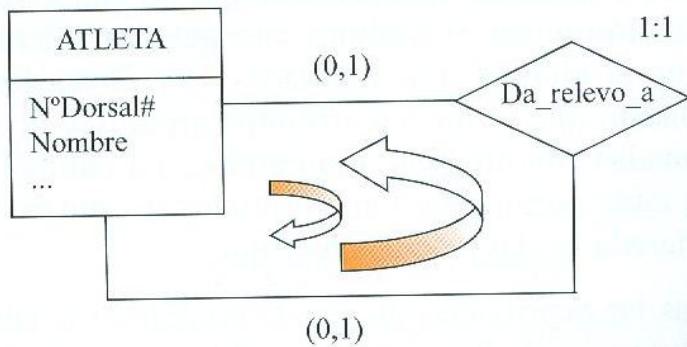
ALUMNO (Nº\_Mat#, Nombre\_A, Apellidos\_A,..., DNI)

En la Tabla *Profesor* están todos los profesores. En la tabla del Alumno están todos los alumnos y el profesor que es su tutor. Si un profesor no es tutor de ningún alumno, aparecerán sus datos en la tabla *Profesor* y no aparecerá en ninguna ocurrencia de la tabla *Alumno*.

# Ejercicios

## Ejercicio nº 5:

Realizar el paso a tablas del siguiente modelo E/R:



## Solución:

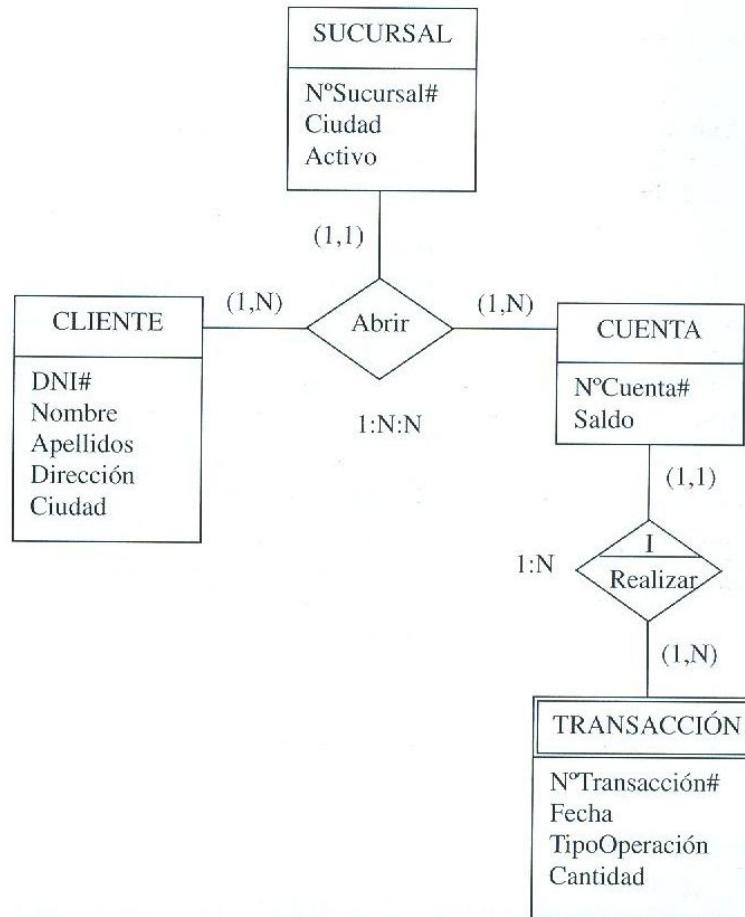
Se crea una tabla para la entidad con todos sus atributos. La relación sí genera tabla puesto que, aunque tiene cardinalidad 1:1, pero la participación mínima en ambos lados es cero. La tabla de la relación tiene como clave cualquiera de los dos atributos, pero no es necesario que sean los dos.

ATLETA (NºDorsal#, Nombre, ...)

Da\_relevo\_a (NºDorsal\_Origen#, NºDorsal\_Destino)

# Ejercicios

## Ejercicio 6



**Solución:**

Las tablas son:

**SUCURSAL** (NºSucursal#, Ciudad, Activo)

**CLIENTE** (DNI#, Nombre, Apellidos, Dirección, Ciudad)

**CUENTA** (NºCuenta#, Saldo)

**Abrir** (NºSucursal#, NºCuenta#, DNI#)

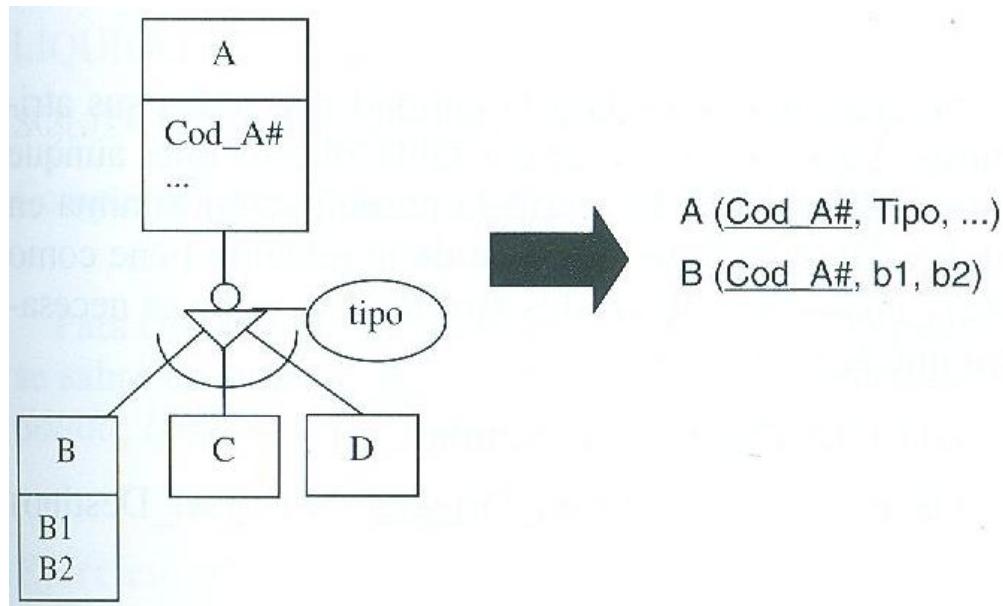
## 2. Eliminación de las relaciones jerárquicas.

Las relaciones jerárquicas son un caso especial. En la mayoría de los casos dependen del problema en concreto. Se pueden dar algunas guías que sirvan como referencia.

1. Se creará una tabla para cada entidad supertipo, a no ser que tenga tan pocos atributos que dejarla sea una complicación.
2. Si la entidad subtipo no tiene atributos y no está relacionada con ninguna otra entidad, desaparece.
3. Si la entidad subtipo tiene algún atributo, se crea una tabla. Si no tiene clave propia hereda la de la entidad supertipo.

## 2. Eliminación de las relaciones jerárquicas.

4. Si la relación es **exclusiva**, el atributo que genera la jerarquía se incorpora en la tabla de la entidad supertipo. Si se ha creado una tabla para cada una de las entidades subtipo no hace falta incorporar dicho atributo en la entidad supertipo.



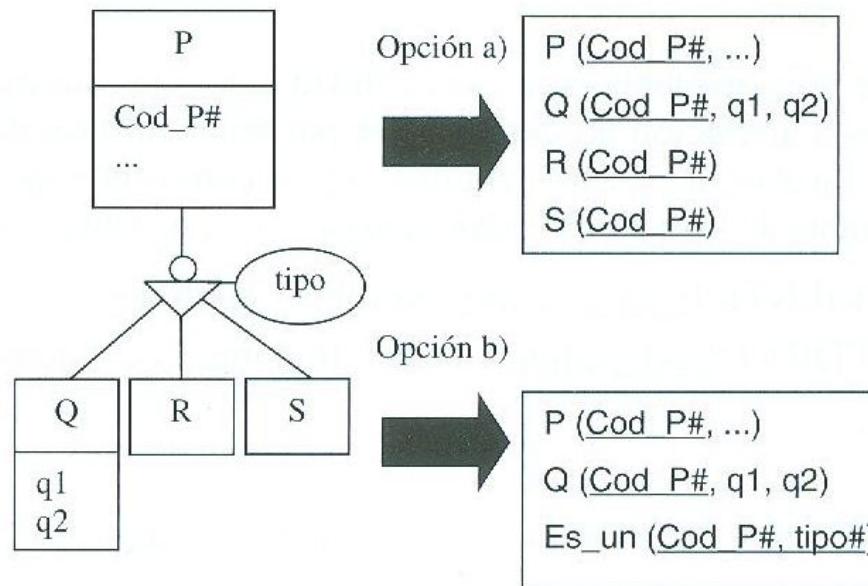
Las ocurrencias de C y D estarán en la tabla de A.

Las ocurrencias de B estarán en la tabla A y en la tabla de B.

## 2. Eliminación de las relaciones jerárquicas.

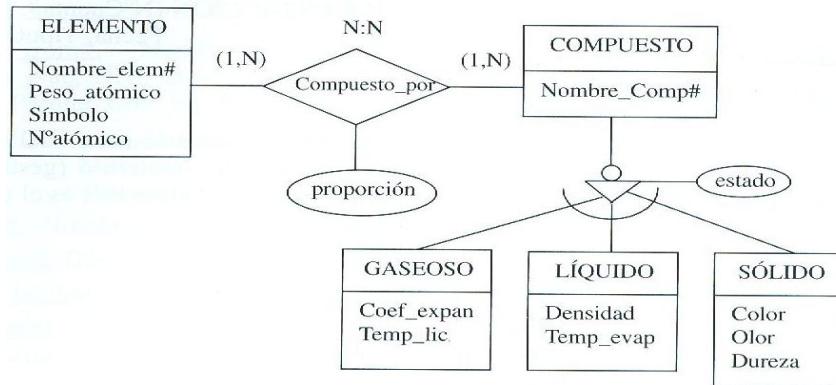
5. Si la relación es **inclusiva** se pueden realizar 2 opciones:

- a) Se creará una tabla para cada entidad subtipo. Si se ha creado una tabla para la entidad supertipo no se incorpora en ella el atributo que genera la jerarquía.
- b) Se crea una tabla para la relación jerárquica que contenga la clave de la entidad supertipo y el atributo que genera la jerarquía. Ambos forman parte de la clave de dicha tabla.



## 2. Eliminación de las relaciones jerárquicas.

Ejemplo. Realizar el paso a tablas del modelo E/R siguiente:



Las tablas son:

ELEMENTO (Nombre\_elem#, Peso\_atómico, Símbolo, N°atómico)  
GASEOSO (Nombre\_comp#, Coef\_expan, Temp\_lic)  
LÍQUIDO (Nombre\_comp#, Densidad, Temp\_evap)  
SÓLIDO (Nombre\_comp#, Color, Olor, Dureza)  
Compuesto\_por (Nombre\_comp#, Nombre\_elem#, proporción)

**La entidad Elemento genera tabla.**

**La relación compuesto\_por genera tabla por ser N:N.**

**Para la relación jerárquica:**

**Como toda entidad subtipo genera tabla, con clave Nombre\_comp#, no es necesario que COMPUESTO genere tabla, ya que el único atributo de COMPUESTO está en las tres entidades subtipo. Además como es exclusiva total, todas las ocurrencias de compuesto estarán en los subtipos.**