

# Tema 3. Diseño lógico de datos: el modelo relacional.

Una vez realizado el análisis conceptual de datos y obtenido el modelo entidad/relación, se pasa a realizar el denominado diseño lógico. En esta etapa, crearemos el modelo lógico de datos que soportará nuestra aplicación. El modelo más ampliamente usado por las bases de datos del mercado es el modelo relacional. A continuación, describiremos cómo pasar del esquema conceptual al diseño lógico (modelo relacional).

## 1.- El modelo relacional

En el modelo relacional la estructura básica es la relación (que se representa en forma de tabla), en la que se distinguen un conjunto de columnas llamadas atributos y un conjunto de filas denominadas tuplas, que son las ocurrencias de la relación.

### Nombre de la relación

<i>Atributo 1</i>	<i>Atributo 2</i>	<i>....</i>	<i>Atributo n</i>
Xxxxxxxxxx	Xxxxxxxxxx	....	Xxxxxxxxxx
Xxxxxxxxxx	xxxxxxxxxx	....	Xxxxxxxxxx
.....	.....	....	....
xxxxxxxxxx	xxxxxxxxxx		Xxxxxxxxxx

- Tupla 1
- Tupla 2
- .....
- Tupla m

### Ejemplo

#### Empleado

<i>DNI</i>	<i>COD_Emp</i>	<i>Nombre</i>	<i>Direcció n</i>	<i>Sueldo</i>
25242321	A456	Antonio	c/ uno	1500
23545869	A455	Francisco	c/ dos	1350
.....		.....		....
78451245	C123	Juan	c/ eme	2000

- Tupla 1
- Tupla 2
- .....
- Tupla m

Una relación suele representarse simplemente indicando su nombre y, entre paréntesis, sus atributos:

**NOMBRE\_RELACION (Atributo 1, Atributo 2, ..., Atributo n)**

Por ejemplo, la relación “Empleado” anterior se representaría de la siguiente forma:

**EMPLEADO (DNI, COD\_Emp, Nombre, Dirección, Sueldo)**

Otros conceptos importantes del modelo son los siguientes:

- **Clave candidata:** Se define como un conjunto de atributos que determinan unívoca y mínimamente cada tupla. En una relación pueden existir varias claves candidatas, por ejemplo en la relación

EMPLEADOS (COD\_EMP, DNI, NOMBRE, DIRECCIÓN, SUELDO)

Podrían ser claves candidatas tanto el COD\_EMP como el DNI. Una de estas se elige como **clave primaria** (por ejemplo el COD\_EMP), mientras que el resto se consideran **claves alternativas**.

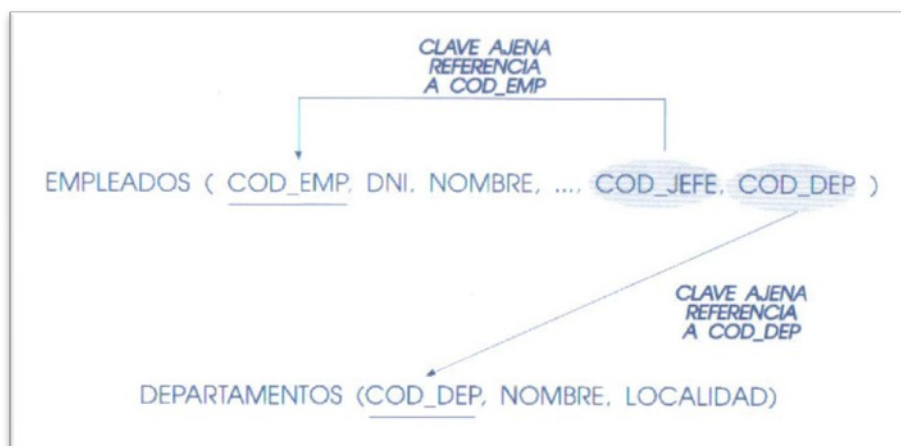
La clave primaria suele indicarse subrayándola:

EMPLEADOS (COD\_EMP, DNI, NOMBRE, DIRECCIÓN, SUELDO)

El modelo relacional presenta una regla denominada “integridad de entidad”, que establece que ningún atributo que forme parte de la clave primaria de una relación puede tomar valor nulo.

- **Clave ajena:** Conjunto de atributos cuyos valores han de coincidir con los valores de la clave primaria de otra relación (o de ella misma, en el caso de las interrelaciones reflexivas, como veremos más adelante).

En la imagen de ejemplo, se puede observar que COD\_JEFE es una clave ajena que referencia a la clave primaria de empleado COD\_EMP. Asimismo, COD\_DEP es una clave ajena que referencia a la clave primaria de la relación departamento COD\_DEP.



Para las claves ajenas el modelo relacional especifica la regla de “integridad referencial” que establece que los valores de la clave ajena o bien coinciden con los de la clave primaria a la que referencian o bien son nulos. Por tanto, para cada clave ajena se debe especificar si puede o no tomar valores nulos. Además es necesario determinar las consecuencias de ciertas

operaciones (borrado y modificación) realizadas sobre las tuplas de la relación referenciada, pudiéndose distinguir las siguientes opciones:

- Operación restringida (**RESTRICT**): Esto es, el borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada sólo se permite si no existen tuplas con dicha clave en la relación que contiene la clave ajena. Esto implicaría, por ejemplo, en el caso del ejemplo anterior, que para borrar un departamento no debe haber ningún empleado que estuviese trabajando en dicho departamento, si no, el sistema impediría su borrado.
- Operación de transmisión en cascada (**CASCADE**): El borrado o la modificación de tuplas de la relación que contiene la clave referenciada lleva consigo el borrado o la modificación en cascada de las tuplas de la relación que contiene la clave ajena. En el ejemplo, equivaldría a decir que al modificar un código de departamento en la relación DEPARTAMENTOS, se debe modificar también dicho código en todos los empleados que trabajen en dicho departamento.
- Operación con puesta a valor nulo (**SET NULL**): esto es, el borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo poner a nulos los valores de las claves ajenas de la relación que referencia. Esto implicaría que si se borra un departamento, todos los empleados tendría un valor nulo en su atributo COD\_DEP.
- Operación con puesta a valor por defecto (**SET DEFAULT**): Esto es, el borrado o modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo poner al valor por defecto a la clave ajena de la relación que referencia. El valor por defecto habrá sido definido al crear la tabla correspondiente.

## **2.- Reglas de transformación de modelo E/R a relacional**

---

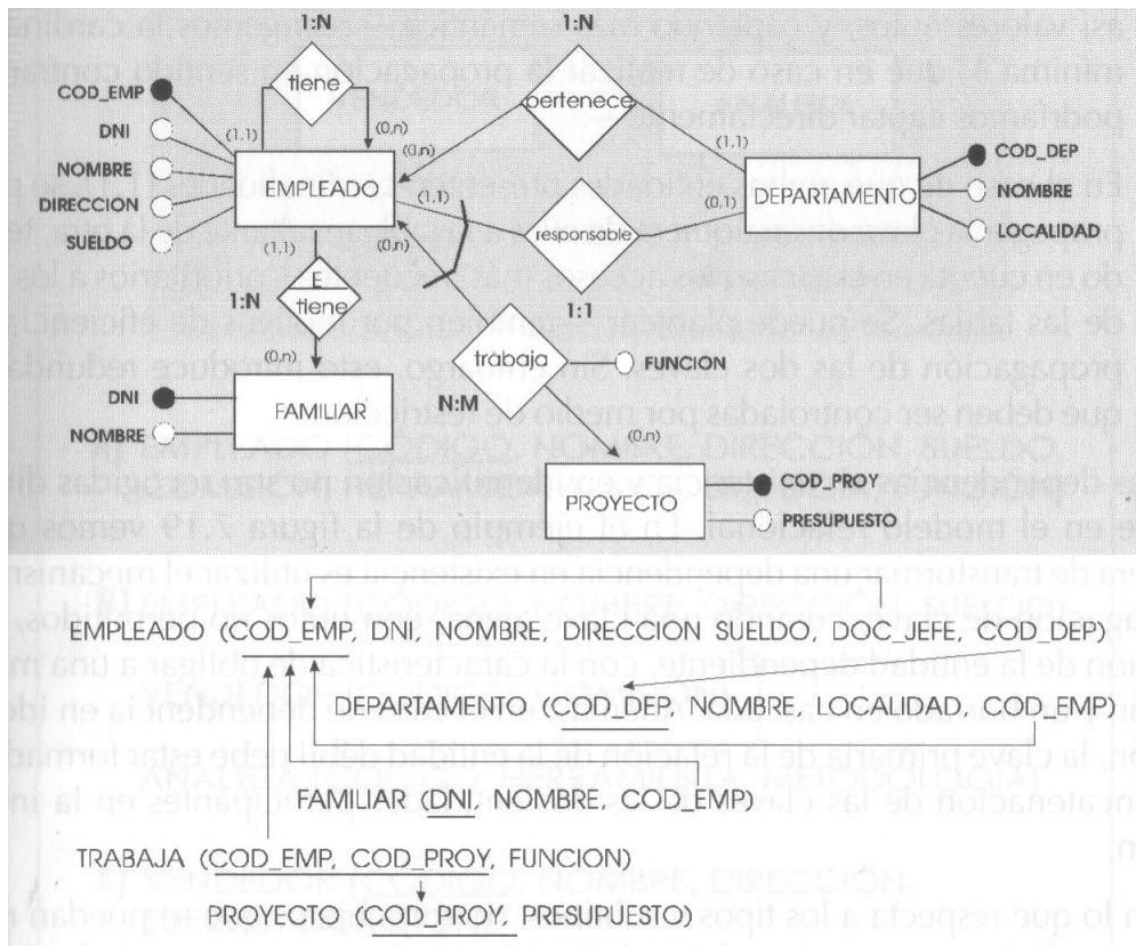
El paso de un esquema en el modelo E/R al relacional está basado en los tres principios siguientes:

1. Toda entidad se convierte en una relación con su correspondiente clave primaria.
2. Toda interrelación N:M se transforma en una relación. En principio la clave primaria de la relación es la concatenación de las claves primarias de las entidades que participan en la interrelación. Además, los atributos de la interrelación pasan a formar parte de la

relación. Las claves primarias de esta relación pasarían a ser claves ajenas que referencian a las claves primarias de las relaciones que participan en la interrelación.

3. Toda interrelación 1:N se traduce en el fenómeno de "propagación de clave", que consiste en que la clave primaria de la relación que participa con cardinalidad máxima 1 pasa a formar parte de la relación con cardinalidad máxima N, transformándose en una clave ajena.

Veamos un ejemplo:



En el ejemplo de la figura puede observarse que las entidades EMPLEADO, DEPARTAMENTO, FAMILIAR Y PROYECTO se han transformado en otras tantas relaciones, con sus correspondientes claves primarias y atributos.

La interrelación N:M *trabaja* da lugar a una nueva relación cuya clave primaria es la concatenación de las claves primarias de las entidades que participan en ella (COD\_EMP y COD\_PROY). COD\_EMP y COD\_PROY serán claves ajenas sobre *empleado* y *Proyecto*, respectivamente.

Veamos ahora las interrelaciones 1:N:

- La interrelación *pertenece* entre *empleado* y *departamento* ha generado un fenómeno de propagación de clave, de forma que la clave primaria de la relación con cardinalidad máxima 1 (*departamento*) ha pasado a formar parte de la relación *empleado* como clave ajena.
- La interrelación *tiene* entre *empleado* y *familiar* es similar a la anterior. La clave de la relación con cardinalidad máxima 1 (*empleado*) ha pasado a formar parte de la relación con cardinalidad máxima N (*familiar*), pasando a ser COD\_EMP una clave ajena de la relación *familiar* sobre la relación *empleado*.
- La interrelación reflexiva *tiene* de *empleado*, genera el mismo fenómeno a pesar de ser reflexiva. La clave de la relación que participa con cardinalidad máxima 1 (*empleado*) pasa a formar parte de la relación que participa con cardinalidad máxima N (*empleado* también), como clave ajena. La única diferencia es que ahora tendremos que cambiar el nombre del atributo para no confundirlos. De esta forma la clave de *empleado* pasa a ser una clave ajena sobre *empleado*, denominándose DOC\_JEFE.

4. Por lo que respecta a las **interrelaciones 1:1**, se pueden recoger en el esquema relacional de distintas maneras atendiendo a las cardinalidades de las entidades que participan en la misma:

- Si las entidades que se asocian poseen cardinalidades (0,1), en este caso la interrelación 1:1 se transformará en una relación, además de las dos relaciones que representan cada una de las entidades. Por ejemplo, si tenemos la asociación MATRIMONIO entre las entidades HOMBRE y MUJER, que se transforma en una relación, evitando así los valores nulos que aparecerían en caso de propagar la clave de la entidad MUJER a la tabla HOMBRE o viceversa, ya que como reflejan las cardinalidades no todos los hombres ni todas las mujeres se encuentran casados.
- Si una de las entidades que participa en la interrelación posee cardinalidades (0,1), mientras que en la otra son (1,1), conviene propagar la clave de la entidad con cardinalidades (1,1) a la tabla resultante de la entidad de cardinalidades (0,1). En la figura anterior tenemos una interrelación que recoge el empleado que es responsable de un departamento, suponiendo que un empleado puede ser responsable como máximo de un departamento y que cada departamento tiene que tener un responsable —pero sólo uno—. En este caso propagamos la clave de EMPLEADO a la tabla de DEPARTAMENTO, evitando así valores nulos, y captando más semántica —recogemos la cardinalidad mínima 1, que en caso de realizar la propagación en sentido contrario no podríamos captar directamente—.
- En el caso de que ambas entidades presenten cardinalidades (1,1), se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra, teniendo en cuenta en este caso los accesos más frecuentes y prioritarios a los datos de las tablas.

5. Las transformaciones de **las dependencias en existencia y en identificación** se realizan de la siguiente forma:

- **Existencia:** En el ejemplo de la figura vemos que la manera de transformar una dependencia en existencia es utilizar el mecanismo de propagación de clave, creando una clave ajena, con nulos no permitidos para COD\_EMP, en la relación de la entidad dependiente FAMILIAR, con la característica de obligar a una modificación y un borrado en cascada: Si se borra un empleado, se borran todos sus familiares.
- **Identificación:** Además, en el caso de dependencia en identificación, la clave primaria de la relación de la entidad débil debe estar formada por la concatenación de las claves de las dos entidades participantes en la interrelación.

6. En lo que respecta a los **tipos y subtipos** (jerarquías o generalizaciones), caben varias soluciones de transformación en el modelo relacional. Destacamos tres principalmente:

- Opción a: Englobar todos los atributos de la entidad y sus subtipos en una sola relación. En general, adoptaremos esta solución cuando los subtipos se diferencien en muy pocos atributos y las interrelaciones que los asocian con otras entidades del esquema conceptual sean las mismas para todos los subtipos. Por ejemplo, la diferencia que existe entre un analista y un vendedor podemos considerarla como mínima desde el punto de vista de ser empleados de la empresa. Por este motivo, una solución adecuada sería la creación de una sola relación que contenga todos los atributos del supertipo y los de los subtipos, añadiendo un atributo adicional que indique el tipo que empleado al que se refiere (atributo discriminante de la jerarquía).
- Opción b: Crear una relación para el supertipo y tantas relaciones como subtipos haya, con sus atributos correspondientes, incluida la clave del supertipo como clave ajena en el subtipo. El atributo discriminante irá incluido en el supertipo. Ésta es la solución adecuada cuando existen muchos atributos distintos entre los subtipos, queriéndose mantener de todas maneras los atributos comunes a todos ellos en una relación.
- Opción c: Considerar relaciones distintas para cada subtipo que contengan además los atributos comunes. Se elegiría esta opción cuando se den las mismas condiciones que en el caso anterior (muchos atributos distintos) y los accesos realizados sobre los datos de los distintos subtipos afectan casi siempre a atributos comunes.

De todas estas opciones, **la más habitual es la opción b**. Será la que utilizaremos en clase.

7. **Atributos compuestos:** Cada atributo de este tipo se transformará en tantos atributos simples como componentes tenga el atributo compuesto. Así, un atributo compuesto

*Dirección*, cuyos componentes son *calle*, *número* y *puerta*, se transformará en tres atributos.

8. **Atributos multivaluados:** dan lugar a una nueva relación cuya clave primaria es la concatenación de la clave primaria de la entidad en la que se sitúa el atributo multivaluado más el nombre del atributo multivaluado. Por ejemplo, tenemos un entidad RECAMBIO, con un atributo multivaluado denominado Proveedor. Para transformar el atributo multivaluado Proveedor obtenemos una relación denominada PROVEEDOR cuya clave primaria es la concatenación de la clave primaria de RECAMBIO más el propio atributo Proveedor.
9. **Atributos Nulos:** Si un atributo puede tomar valor nulo, lo etiquetaremos en la relación con un asterisco \*. Por ejemplo, supongamos que el atributo sueldo admite valores nulos:

EMPLEADO (COD\_EMPLEADO, NOMBRE, APELLIDOS, SUELDO\*)

#### 10. Interrelaciones 1:N - Casos especiales

- Cardinalidad mínima 0 en la entidad que participa con cardinalidad máxima 1:



La clave ajena propagada será opcional (habrá ocurrencias de la otra entidad cuya clave ajena sea NULL):

PERSONA (DNI, NOMBRE, APELLIDOS)

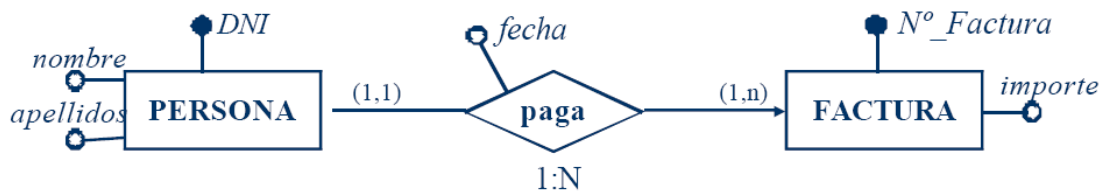
FACTURA (N\_FACTURA, DNI\*, IMPORTE, FECHA)

- Cardinalidad mínima 0 en la entidad que participa con cardinalidad máxima n:



Habrán ocurrencias de persona que no tengan asociada ninguna factura:  
NO HAY QUE HACER NADA.

- Cardinalidad mínima 1 en cualquiera de las dos entidades:



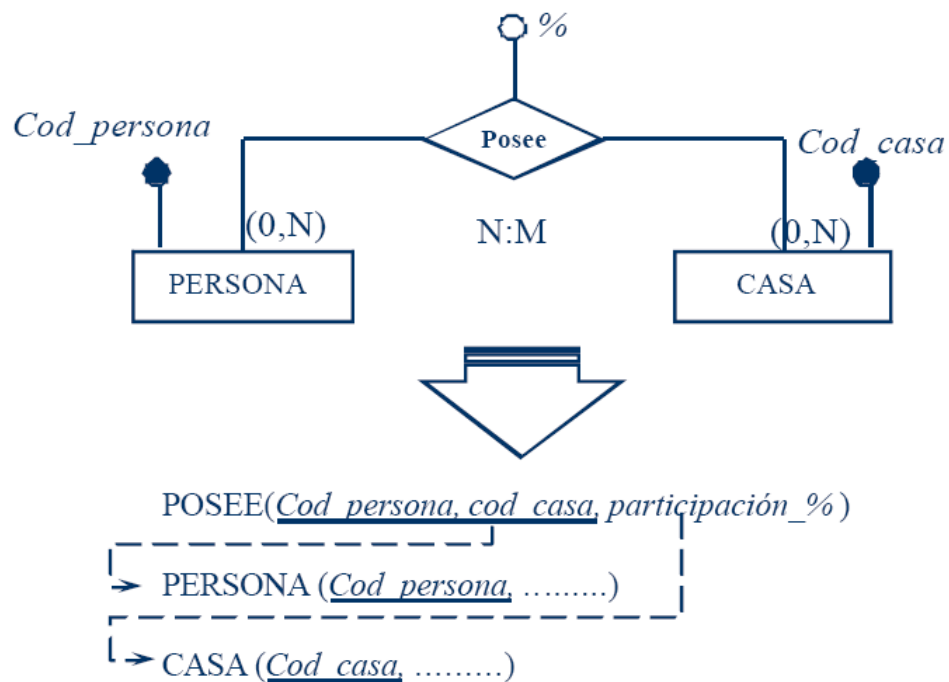
La clave ajena propagada (DNI) no podrá tomar valores nulos en ningún caso (siempre debe haber una persona para cada factura).

PERSONA (DNI, NOMBRE, APELLIDOS)  
 FACTURA(N\_FACTURA, DNI, IMPORTE,FECHA)

El modelo relacional no puede garantizar que todo valor de la clave propagada (DNI) debe aparecer en alguna ocurrencia de la otra relación. Esto habrá que controlarlo desde la aplicación que hace uso de la base de datos.

#### 11. Interrelaciones con atributos normales que no denotan fechas

- Si se crea una nueva relación, esos atributos se incluyen en esta relación.



- Si se propaga una clave (caso de las 1:N), los atributos acompañan a la clave.



## 12. Interrelaciones con un atributo multivaluado que no denota fechas:

Tenemos un aprendiz que asiste a cursos de formación en los que puede obtener distintas calificaciones.



Tenemos las siguientes relaciones:

APRENDIZ (COD\_AP)

CURSO\_FORMACION (COD\_CURSO)

ASISTE (COD\_AP, COD\_CURSO, CALIFICACION)

Dado que aprendiz puede obtener varias calificaciones a lo largo del curso (calificación es multivaluado), CALIFICACION tiene que formar parte de la clave.

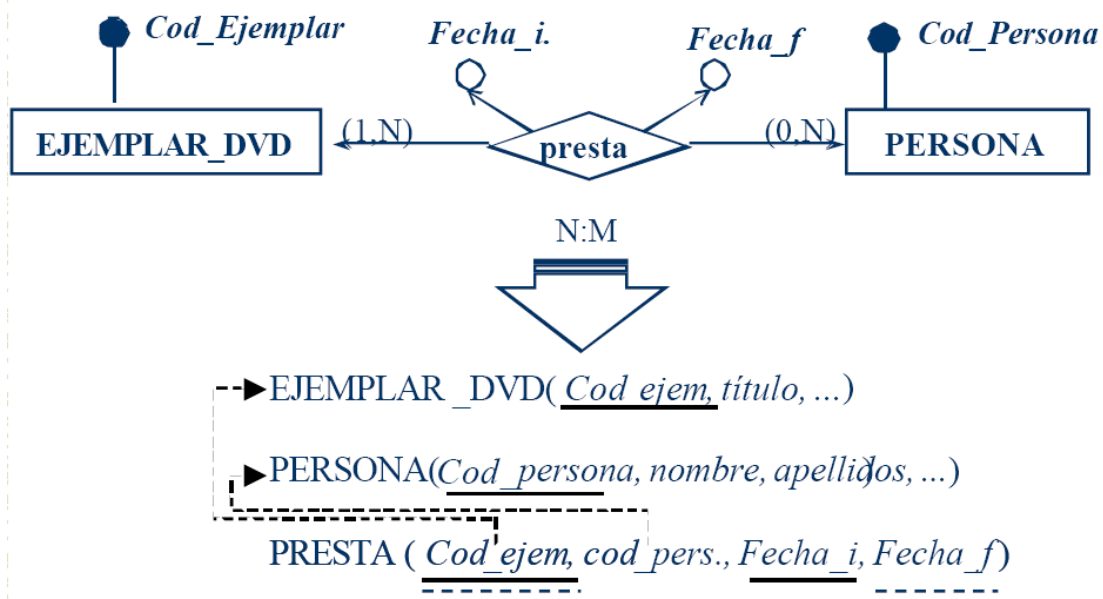
ASISTE (COD\_AP, COD\_CURSO, CALIFICACION)

Además, el aprendiz puede obtener la misma calificación durante el mismo curso, con lo cual hay que añadir un atributo fecha, que indique en qué fecha ha obtenido esas calificaciones. La fecha, además, debe formar parte de la relación.

ASISTE (COD\_AP, COD\_CURSO, CALIFICACION, FECHA)

## 13. Interrelaciones con un atributo multivaluado que denota fechas

- 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 de las dos, dependiendo de la semántica del problema).



Una persona puede tener en préstamo un determinado DVD. Si la clave de la relación hubiera sido (Cod\_Ejem, Cod\_pers), una persona no podría alquilar el mismo DVD nunca pues se repetirían tuplas.

PRESTA ('001', 'Antonio', '15/10/2009', '17/10/2009')

PRESTA ('001', 'Antonio', '22/10/2009', '25/10/2009')

No es lógico: Una misma persona no puede alquilar dos veces el mismo ejemplar de DVD, ya que se repiten tuplas (misma clave) y eso no está permitido.

Una solución adecuada puede ser escoger como clave (Cod\_Ejem, fecha\_i), esto permite que un ejemplar para una fecha de inicio de alquiler no se pueda repetir (lo cual es lógico).

PRESTA ('001', 'Antonio', '15/10/2009', '17/10/2009')

PRESTA ('001', 'Antonio', '22/10/2009', '25/10/2009')

Una misma persona se puede llevar el mismo ejemplar en fechas distintas. Pero dos personas no pueden llevarse el mismo ejemplar en la misma fecha.

Si incluimos cod\_pers como parte de la clave estaríamos representando el hecho de que varias personas pueden llevarse el mismo ejemplar el mismo día (a horas distintas y una vez devuelto el ejemplar, claro está).

PRESTA ('001', 'Antonio', '15/10/2009', '17/10/2009')

PRESTA ('001', 'Juan', '15/10/2009', '17/10/2009')

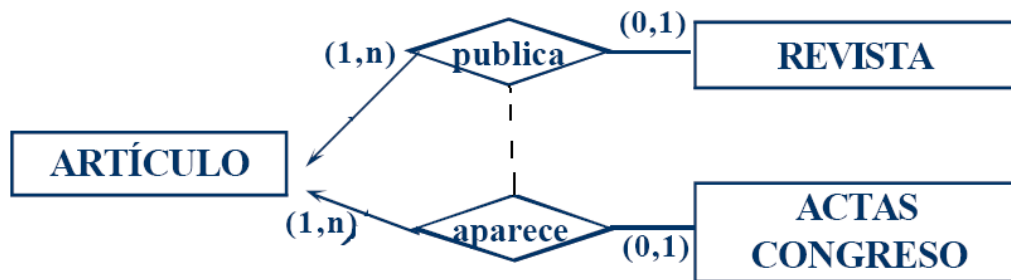
PRESTA ('001', 'Antonio', '15/10/2009', '17/10/2009')

No sería válido (se repite la clave).

Dos personas se llevan el mismo ejemplar el mismo día (a horas distintas). Sin embargo, Una misma persona no puede llevarse el mismo ejemplar el mismo día (habría que incluir las horas como atributo).

La solución depende de la **SEMÁNTICA** que se quiera representar.

#### 14. Interrelaciones exclusivas



Además de crear las relaciones

REVISTA (COD\_RV)

ACTAS\_CONGRESO (COD\_AC)

ARTICULO (COD\_ART, COD\_REV\*, COD\_AC\*)

Hay que incluir algún tipo de verificación cuando creamos la tabla en la base de datos. Esta verificación será de este tipo:

CHECK ((*Cod\_Rv* IS NULL AND *Cod\_Ac* IS NOT NULL)

OR

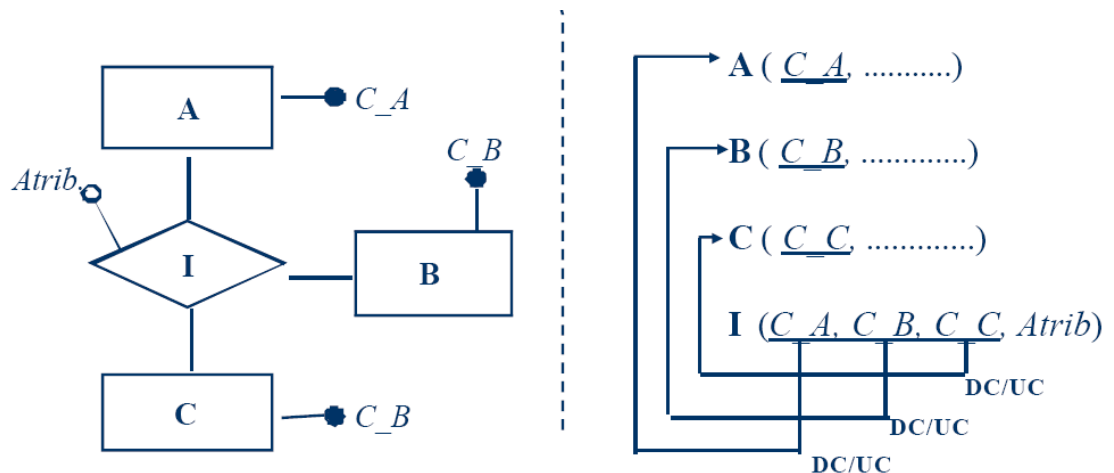
(*Cod\_Rv* IS NOT NULL AND *Cod\_Ac* IS NULL))

Indica que *Cod\_RV* es nulo y *cod\_ac* no lo es o lo contrario. Esto hará que la interrelación sea exclusiva: ambos campos *Cod\_RV* y *Cod\_ac* no puede ser no nulos a la vez.

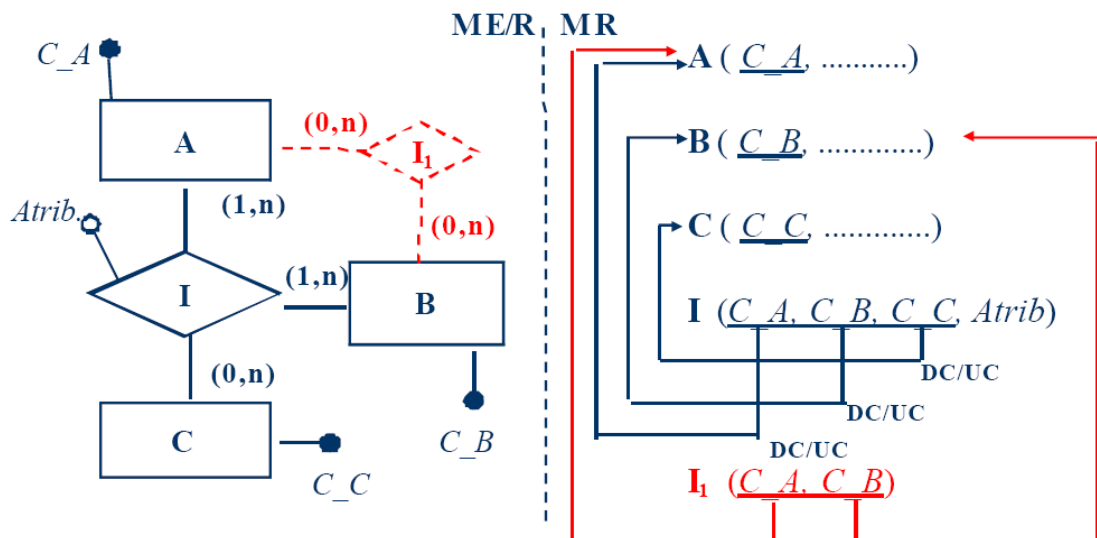
## 15. Interrelaciones de grado mayor de 2 (ternarias)

- Caso general: Cardinalidad mínima 1 y máxima n en todas las entidades

Se crean las relaciones correspondientes y una nueva relación correspondiente a la interrelación, incluyendo como clave la concatenación de las claves de las tres entidades participantes. Además se incluirán los atributos de la interrelación.

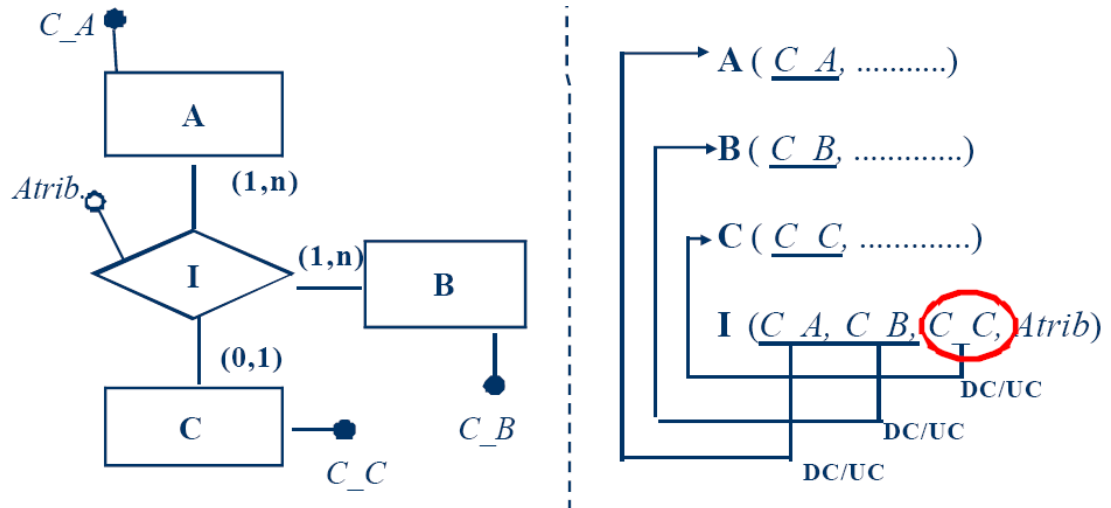


- Cardinalidad máxima n en todas las entidades y 0 en alguna de las ramas. Se aplica el caso general y además se crea otra interrelación ( $I_1$ ). La relación derivada de esta interrelación ( $I_1$ ) tiene como clave primaria la concatenación de las claves primarias de las entidades participantes.



- Cardinalidad máxima 1 en una de las ramas.

Se crea la relación derivada de la interrelación y la clave de ésta será, como mínimo, la concatenación de las claves de las dos relaciones que participan con cardinalidad máxima n.



### 3.- DEPENDENCIAS FUNCIONALES Y TEORIA DE LA NORMALIZACION

La teoría de la normalización se basa en el concepto de dependencias, que son propiedades inherentes al contenido semántico de los datos. La existencia de una dependencia no se puede demostrar, pero si afirmar, por observación del mundo real. Existen muchos tipos de dependencias (funcionales, multivaluadas, de combinación, etc.), pero en este libro solo trataremos las funcionales, ya que son las que se encuentran asociadas a las tres primeras formas normales.

#### 3.1.- Primera forma normal (1FN)

Una relación está en primera forma normal si todo atributo tiene un valor atómico. Supongamos la siguiente relación:

EMPLEADO (COD\_EMPLEADO, NOMBRE\_EMPLEADO, TELEFONOS)

Con los siguientes valores:

COD_EMPLEADO	NOMBRE_EMPLEADO	TELEFONOS
001	Antonio	955123456,954987654
002	Juan	954456789
003	Pepe	955789887

Esta relación no estaría en forma normal, pues como vemos, el campo TELEFONOS del empleado con COD\_EMPLEADO '001' no tiene un valor atómico. Para que esta relación pasase a estar en 1ª forma normal debería tener esta forma:

COD_EMPLEADO	NOMBRE_EMPLEADO	TELEFONOS
001	Antonio	955123456
001	Antonio	954987654
002	Juan	954456789
003	Pepe	955789887

Pasando a ser clave tanto COD\_EMPLEADO como TELEFONOS.

En el modelo relacional, todos los valores de los atributos deben ser atómicos, por lo que toda relación del modelo relacional estará siempre en 1ª forma normal.

### 3.2.- Definición de dependencia funcional

Decimos que un descriptor Y (conjunto de campos) depende funcionalmente del descriptor X, si, y sólo si, cada valor de X tiene asociado en todo momento un único valor de Y, lo que se representa como:

$X \rightarrow Y$

Así, por ejemplo, podemos decir que:

$COD\_ALMACEN \rightarrow DIRECCION\_ALMACEN$

ya que para cada código (suponiendo que no se repiten) existe una sola dirección, en esta dependencia se dice que el COD\_ALMACEN es el implicante y DIRECCION\_ALMACEN el implicado.

### 3.3.- Dependencia funcional completa y segunda forma normal (2FN)

Si el descriptor X es compuesto:

$X(X1, X2)$

se dice que Y tiene dependencia funcional completa o plena respecto de X, si depende funcionalmente de X, pero no depende de ningún subconjunto del mismo, esto es:

$X \rightarrow Y$

$X1 \not\rightarrow Y$

$X2 \not\rightarrow Y$

Así, por ejemplo, si suponemos que en una empresa un empleado puede trabajar en varios proyectos, realizando una sola función en cada uno de ellos (consultor, analista, programador, etc.), aunque pueda ser distinta según el proyecto, tendríamos que:

$(DNI\_EMPLEADO, COD\_PROYECTO) \rightarrow FUNCION$

es una dependencia completa, ya que ninguno de los elementos del descriptor por separado determina el implicado, al poder tener un empleado muchas funciones, lo mismo que un proyecto.

Sin embargo, la dependencia:

$(\text{COD\_PIEZA}, \text{COD\_ALMACEN}) \rightarrow \text{DIRECCION\_ALMACEN}$

no es completa, ya que  $\text{COD\_ALMACEN} \rightarrow \text{DIRECCION\_ALMACEN}$  y no es necesario conocer el COD\_PIEZA para saber junto al COD\_ALMACEN cuál es la DIRECCION\_ALMACEN.

Una relación está en 2FN si:

- Esta en 1FN.
- Todo campo no clave tiene una dependencia funcional completa respecto de las claves candidatas.

Por esta razón la relación:

VENTAS (COD\_PIEZA, COD.ALMACEN, CANTIDAD, DIRECCION\_ALMACEN)

no se encuentra en 2FN, mientras que las relaciones:

N\_VENTAS (COD\_PIEZA, COD\_ALMACEN, CANTIDAD)  
ALMACENES (COD\_ALMACEN, DIRECCION\_ALMACEN)

Si, ya que en la primera relación,  $(\text{COD\_PIEZA}, \text{COD\_ALMACEN}) \rightarrow \text{CANTIDAD}$  es una dependencia completa, y en el segundo  $\text{COD\_ALMACEN} \rightarrow \text{DIRECCION\_ALMACEN}$  también lo es, ya que en el caso de que la clave sea simple (este formada por un solo campo) siempre que se encuentre en 1FN estará también en 2 FN.

### 3.4.- Dependencia funcional transitiva y tercera forma normal (3FN)

Supongamos que existe una relación R, con tres descriptores X, Y, Z entre los que existen las siguientes dependencias:

$X \rightarrow Y$

$Y \not\rightarrow X$

$Y \rightarrow Z$

Se dice que Z tiene una dependencia transitiva respecto de X a través de Y, lo que se representa como

$X \twoheadrightarrow Z$ .

Por ejemplo, si suponemos que se dan las siguientes dependencias:

$\text{COD\_EMPLEADO} \rightarrow \text{COD\_DEPARTAMENTO}$

$\text{COD\_DEPARTAMENTO} \rightarrow \text{NOMBRE\_DEPARTAMENTO}$

$\text{COD\_DEPARTAMENTO} \not\rightarrow \text{COD\_EMPLEADO}$



Podemos afirmar que COD\_EMPLEADO → NOMBRE\_DEPARTAMENTO transitivamente a través de COD\_DEPARTAMENTO.

Se dice que un registro se encuentra en 3FN si:

- Esta en 2FN.
- Ningún campo no clave depende transitivamente de ninguna clave.

Es por esto que la relación:

EMPLEADOS (COD\_EMPLEADO, COD\_DEPARTAMENTO, NOMBRE\_DEPARTAMENTO)

no se encuentra en 3FN, ya que la clave es el COD\_EMPLEADO, mientras que NOMBRE\_DEPARTAMENTO, que es un campo que no forma parte de la clave, depende transitivamente de esta.

### 3.5.- Forma normal de Boyce-Codd (FNBC)

Una relación se encuentra en FNBC si, y sólo si, todo determinante es clave", donde por determinante entendemos cualquier conjunto de campos del que otro campo depende funcionalmente de forma completa.

Dada la siguiente relación:

VENTAS (COD\_PIEZA, COD\_ALMACEN, NOMBRE\_ALMACEN, CANTIDAD)

Donde suponemos que los almacenes se identifican unívocamente tanto por el código como por el nombre; existen dos claves, la formada por el conjunto (COD\_PIEZA, COD\_ALMACEN) y la que componen (COD\_PIEZA, NOMBRE\_ALMACEN). Sin embargo, existe cuatro determinantes; además de las claves anteriores tanto el campo COD\_ALMACEN como NOMBRE\_ALMACEN son determinantes, ya que uno implica al otro y viceversa, por lo que este registro no se encuentra en FNBC, ya que no todo determinante es clave.

Si queremos cumplir la FNBC deberemos descomponer el registro, por ejemplo, la siguiente manera:

ALMACENES (COD\_ALMACEN, NOMBRE\_ALMACEN)

N\_VENTAS (COD\_PIEZA, COD\_ALMACEN, CANTIDAD)

Donde en la primera relación existen dos claves: COD\_ALMACEN y NOMBRE\_ALMACEN, y ambos campos son también determinantes, por lo que está en FNBC. Mientras que en la segunda existe una clave compuesta (COD\_PIEZA, COD\_ALMACEN) y un único determinante formado por esos dos campos (COD\_PIEZA, COD\_ALMACEN) que determinan a CANTIDAD, por lo que también se encuentra en FNBC.