

# **Aclaración:**

# **COMO SE PASA del modelo**

# **ENTIDAD-RELACIÓN al**

# **RELACIONAL de TABLAS**

**BASE de DATOS**

**EMILIANO GÓMEZ VÁZQUEZ**

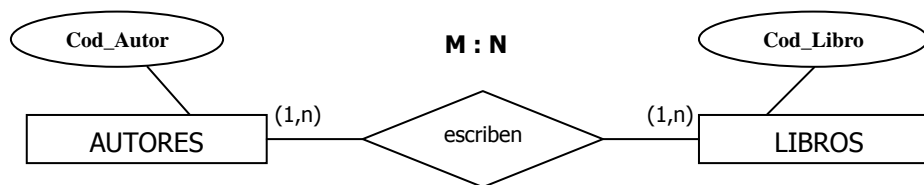
# REGLAS DE TRANSFORMACION DE E/R A RELACIONAL

## RELACIONES M:N

### TRANSFORMACIÓN DE INTERRELACIÓN M:N

**Regla general:** Se transforman en una nueva tabla cuya clave se forma, al menos, con la concatenación de las claves de las entidades que participan en la relación, que son además claves ajenas que referencian a las tablas en las que son claves primarias. El nombre asignado a la tabla es el que tenía la relación.

Ejemplo: Relación entre los autores y los libros que escriben.



escriben ( Cod\_Autor, Cod\_Libro,  
-----

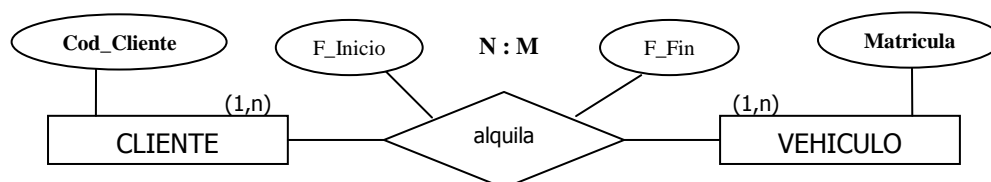
AUTORES ( Cod\_Autor, ..... )

LIBROS ( Cod\_Libro, ..... )

### TRANSFORMACION DE LA DIMENSION TEMPORAL

En algunos casos en que la relación tenga **atributos de tipo fecha**, será necesario incluir al menos una fecha como parte del atributo identificador principal para recoger la dimensión temporal del modelo. En otros casos la fecha puede ser una entidad más o solo un atributo.

Ejemplo: Relación entre los clientes que alquilan los vehículos de una empresa de alquileres. Recogemos los alquileres realizados a nuestros clientes a lo largo del tiempo.



alquila ( Cod\_Cliente, Matricula, F\_Inicio, F\_Fin  
-----

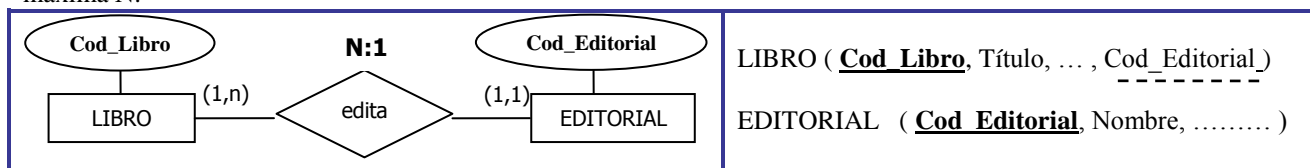
CLIENTE ( Cod\_Cliente, ..... )

VEHICULO ( Matrícula, ..... )

## RELACIONES 1:N

### TRANSFORMACION DE INTERRELACION 1:N

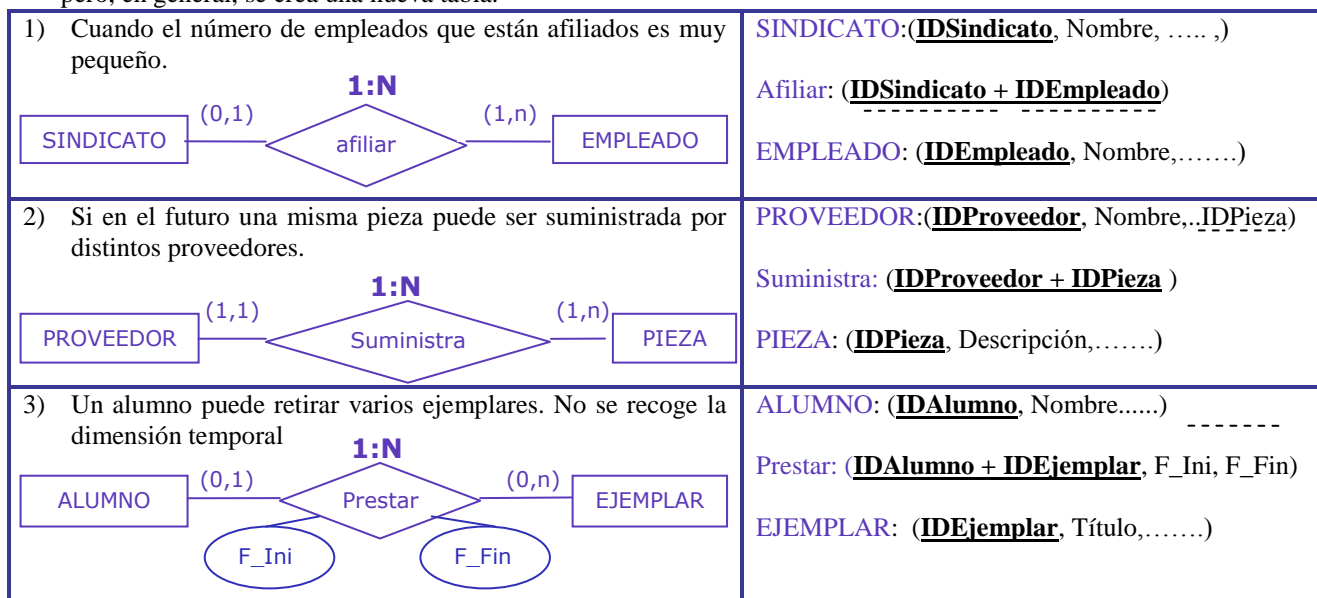
Como **norma general** se propaga la clave de la entidad que tiene cardinalidad máxima 1 a la que tiene cardinalidad máxima N.



### EXCEPCIONES:

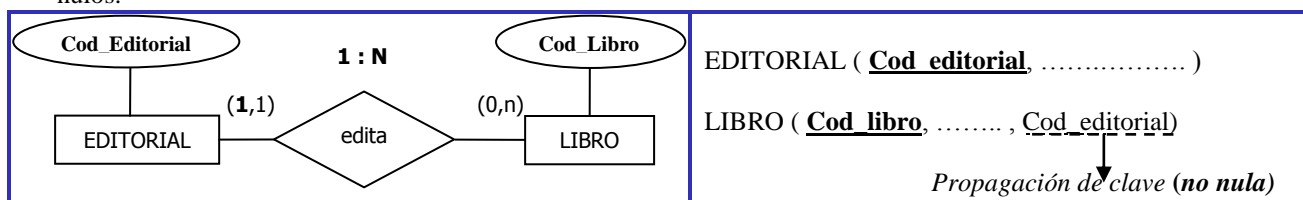
En los siguientes casos interesa más *crear una nueva tabla* a partir de la relación como en el caso de correspondencias M:N

- 1) Cuando el número de ocurrencias de la entidad que propaga la clave es muy pequeño y cabe la posibilidad de que al propagar la clave quedan muchos **valores repetidos o nulos**.
- 2) Cuando se prevea que en el futuro se puede convertir en una **relación N:M**
- 3) Cuando la relación tenga **atributos propios**. En algunos casos se pueden migrar estos atributos junto con la clave pero, en general, se crea una nueva tabla.

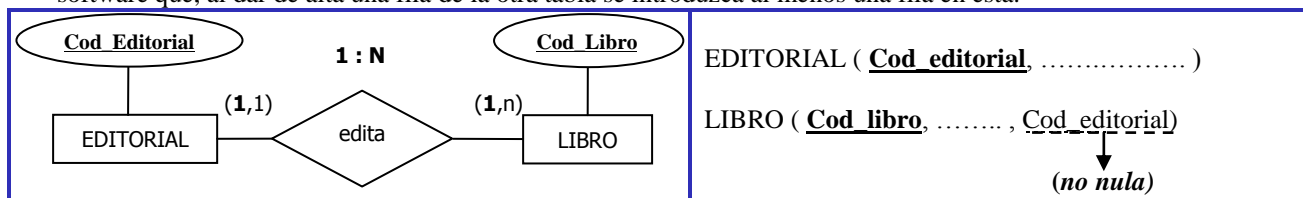


### CARDINALIDADES:

- 1) Cuando la entidad que tiene cardinalidad máxima 1, tiene también 1 de cardinalidad mínima, tendremos que tener en cuenta al propagar la clave que en la tabla que recibe la clave, como clave extranjera, no pueda tener valores nulos.



- 2) Cuando la entidad que tiene cardinalidad máxima n, tiene de cardinalidad mínima 1, tendremos que controlar por software que, al dar de alta una fila de la otra tabla se introduzca al menos una fila en esta.

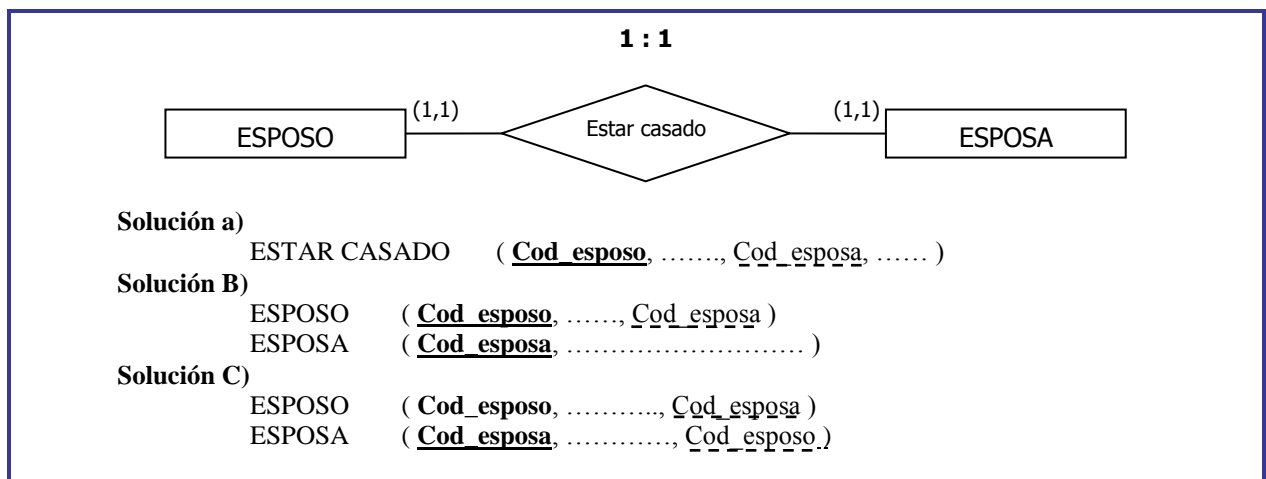


## RELACIONES 1:1

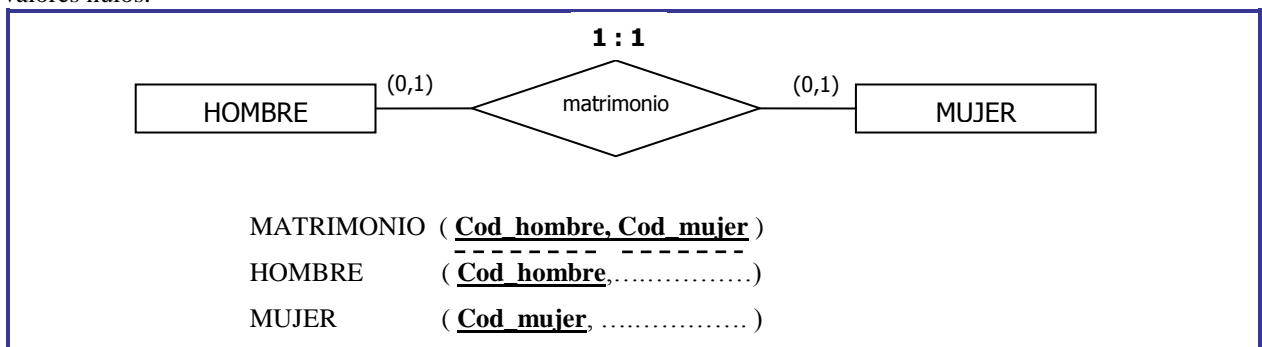
No hay una regla fija, puede optarse por la solución basada en:

- Recoger la mayor cantidad de semántica posible.
- Tener en cuenta las cardinalidades mínimas
- Evitar los valores nulos.
- Motivos e eficiencia.

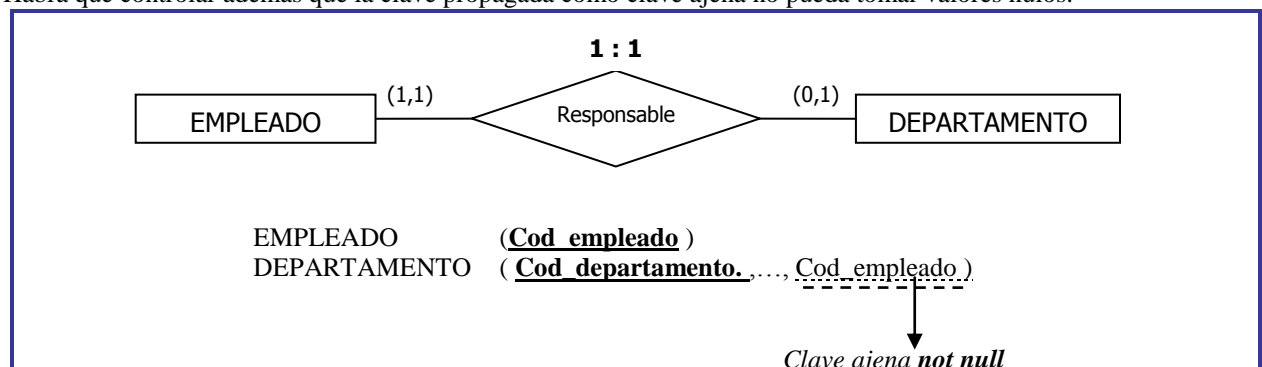
- 1) Cuando las cardinalidades de **ambas entidades son (1,1)** se pueden adoptar distintas soluciones:
  - a) No se necesitan 2 tablas, se puede crear **una única tabla** en la que se incluyan los atributos de las dos entidades, cuya clave principal será cualquiera de los atributos identificadores principales.
  - b) **Propagar la clave** de cualquiera de ellas a la otra tabla, teniendo en cuenta a cual de ellas se le efectúan los accesos mas frecuentes.
  - c) **Propagar las dos claves**, introduce redundancias que se controlarán mediante restricciones



- 2) Cuando las cardinalidades de **ambas entidades son (0,1)**, la relación se transforma en una tabla para evitar los valores nulos.



- 3) **Si una tiene cardinalidad (1,1) y la otra (0,1)**, se propaga la clave de la entidad que tiene cardinalidad (1,1) a la que tiene (0,1).  
 Habrá que controlar además que la clave propagada como clave ajena no pueda tomar valores nulos.



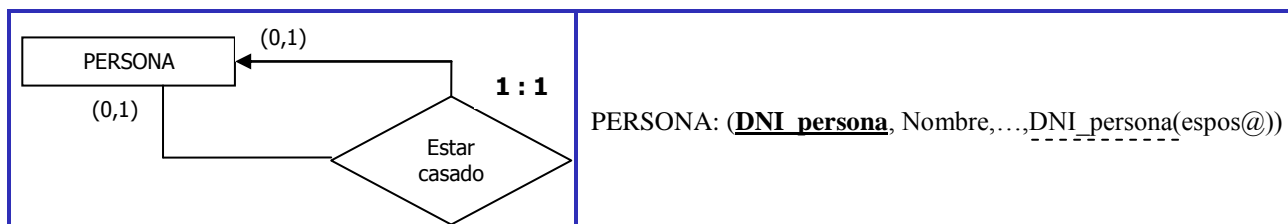
## TRANSFORMACION DE UNA INTERRELACION REFLEXIVA

Se trata de relaciones en las que solo participa una entidad. Como **regla general** toda relación reflexiva se convierte en dos tablas: una para la entidad y otra para la relación.

Se pueden presentar los siguientes casos:

### Relación 1:1

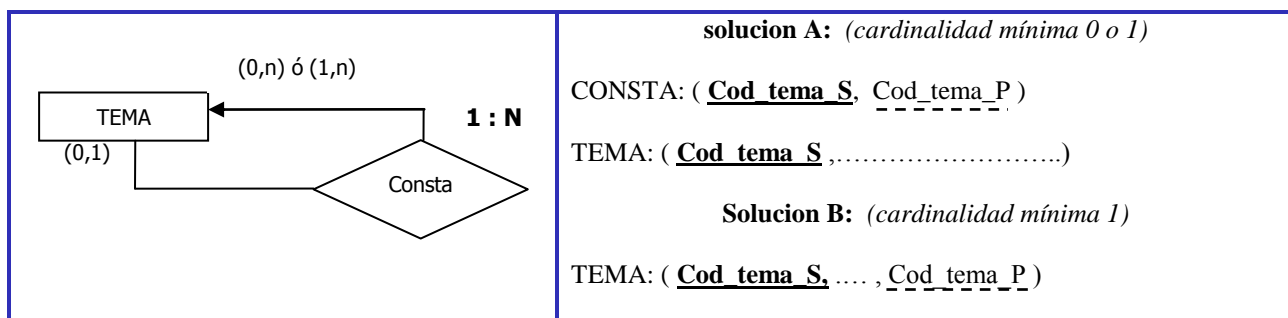
No se crea una tabla para la relación. La clave de la entidad se repite, con lo que la tabla resultante tendrá ese atributo dos veces, una como clave primaria y otra como clave ajena de ella misma.



### Relación 1:N

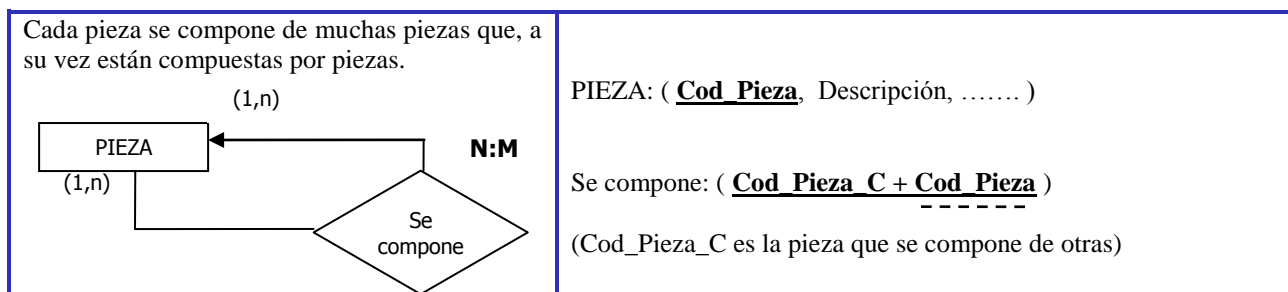
En este caso hay que tener en cuenta la cardinalidad del lado muchos:

- Si no es obligatoria se crea una nueva tabla cuya clave será la de la entidad del lado muchos (Cod\_tema\_S) y se propaga la clave a la nueva tabla como clave ajena.
- Si es siempre obligatoria no se crea una nueva tabla



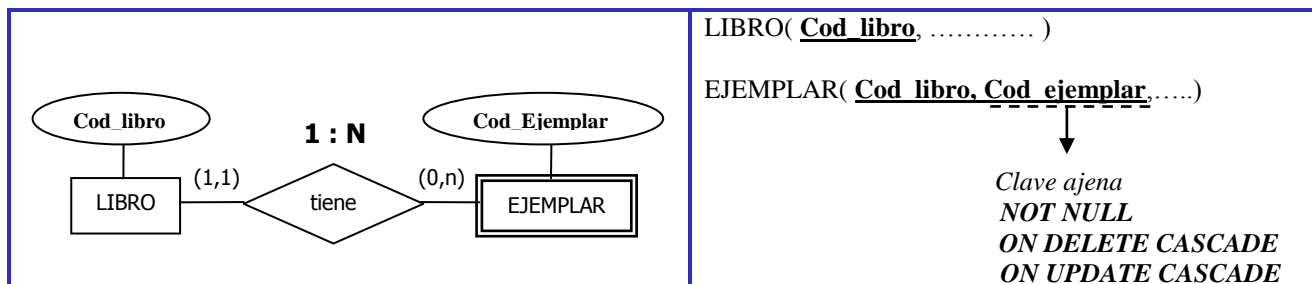
### Relación M:N

La tabla que resulta de la relación contendrá dos veces la clave primaria de la entidad del lado muchos, más los atributos de la relación, si los hay. La clave de esta nueva tabla será la combinación de las dos.



## DEPENDENCIAS

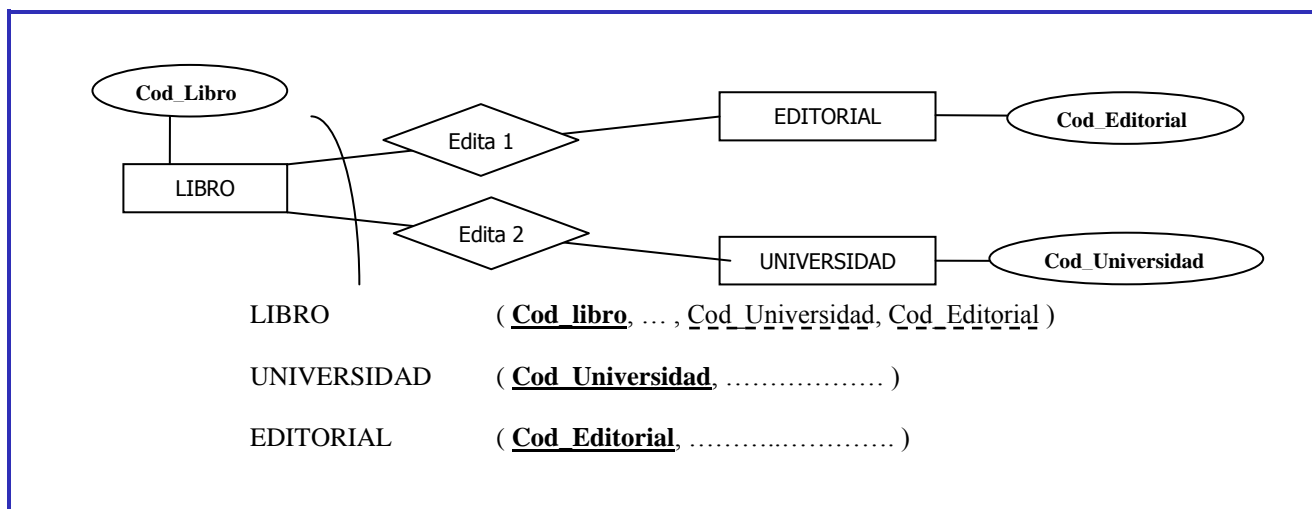
- ✓ **En existencia:** Se propaga la clave, creando una **clave ajena** con las características de:
  - No permitir nulos
  - Modificación en cascada
  - Borrado en cascada
- ✓ **En existencia e identificación: (Relaciones fuertes: IDENTIFICATIVAS)** La propagación de la clave se hace de forma que la clave primaria de la entidad débil estará formada por la **concatenación de las dos claves**.



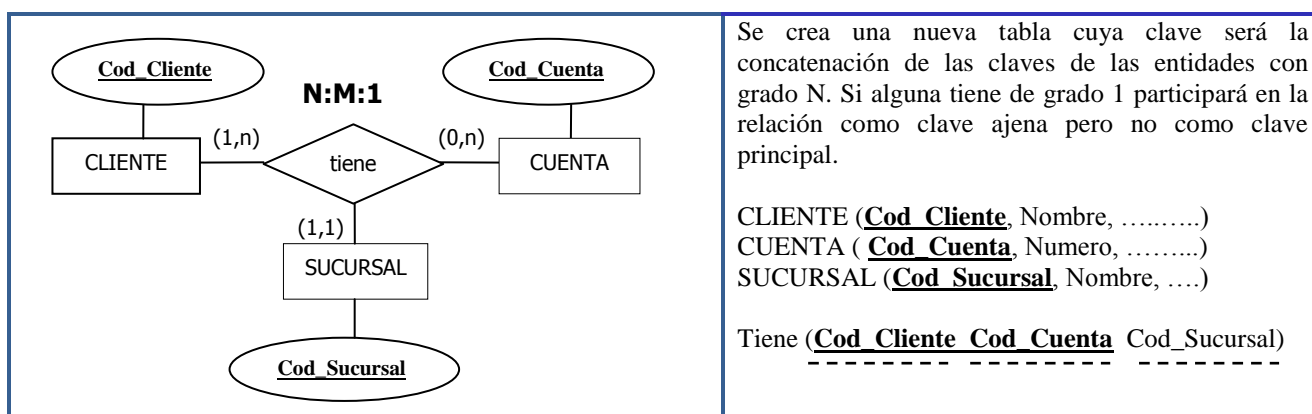
## RELACIONES EXCLUSIVAS

Se **propaga la clave** de las entidades que se excluyen, a la otra entidad, como claves ajenas.

Para que se cumpla la exclusividad es necesario definir **restricciones** en cada caso. Por ejemplo, chequeando que si una de las claves ajenas tiene datos, la otra sea nula y viceversa.



## RELACIONES TERNARIAS



## TRANSFORMACIÓN DE JERARQUÍAS

Pueden darse 3 opciones:

a) Crear **una sola tabla** con todos los atributos de la entidad y de los subtipos, añadiendo como un atributo más el atributo discriminante. Esto se aplica cuando:

- Los subtipos se diferencian en muy pocos atributos
- Las relaciones que los asocian al resto de las entidades sean las mismas para los subtipos.

*Si la jerarquía es:*

- **Total:** el atributo discriminante no admitirá nulos
- **Parcial:** el atributo discriminante si admitirá nulos

*Si entre los subtipos puede haber:*

- **Solapamiento:** Se forman grupos repetitivos, por tanto será necesario crear una nueva tabla que asocie el atributo discriminante con el supertipo.
- **Exclusividad:** No es necesaria una tabla nueva

b) Crear **una tabla para cada tipo y subtipos** que haya. Esto se aplica cuando:

- Existen muchos atributos distintos entre los subtipos.
- Se quieren mantener los atributos comunes en una tabla.

c) Crear **una tabla por cada subtipo**, incluyendo los atributos comunes en cada una. Esto se aplica cuando:

- Existen muchos atributos distintos entre los subtipos
- Los accesos a los datos de los subtipos siempre afectan a los atributos comunes.

### Ventajas e inconvenientes:

Opción a) Es la más rápida por tener que acceder a una sola entidad.

Opción b) La menos eficiente

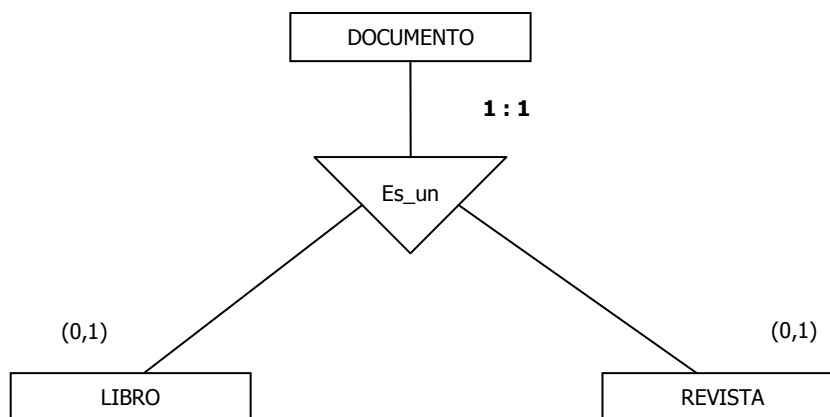
La mejor desde un punto de vista semántico

Opción c) Más eficiente en consultas que afecten a todos los atributos de un subtipo

Menos eficiente en consultas que afecten a los atributos comunes

Introduce redundancias

Es la que pierde más semántica



a) DOCUMENTO ( Código, título, idioma, ..... , tipo)

b) DOCUMENTO ( Código, título, idioma, ..... )

LIBRO ( Código, ..... , )

REVISTA ( Código, ..... , )

c) LIBRO ( Código, título, idioma, ... )

REVISTA ( Código, título, idioma, ... )