



Invoice

Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark

UT3. DISEÑO LÓGICO. MODELO RELACIONAL

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz

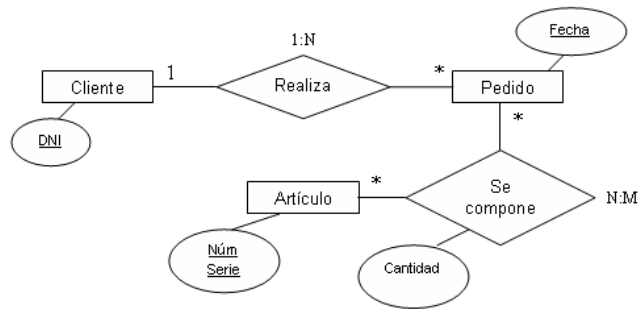


REGLAS DE TRANSFORMACIÓN DEL MODELO ENTIDAD- RELACIÓN AL MODELO RELACIONAL

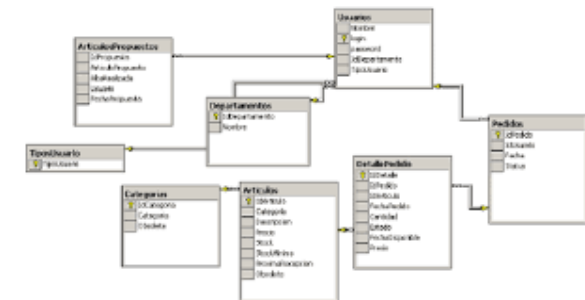


INTRODUCCIÓN

MODELO CONCEPTUAL (CHEN)



REGLAS DE TRANSFORMACIÓN



SIMPLIFICAR EL DIAGRAMA E/R

- Antes de aplicar las reglas de transformación, simplificaremos el diagrama entidad relación aplicando una serie de pautas.
- Nota: depende de cómo hayamos generado el modelo es posible que no sea necesario simplificarlo.
- Transformaciones a realizar:
 - **Transformación de los atributos compuestos.** Estos atributos deben ser descompuestos en los atributos simples por los que están formados.
Ejemplo: si en entidad EMPLEADO consideramos el atributo "Nombre completo", se descompondrá en tres atributos simples "Nombre", "Apellido1" y "Apellido2".
 - **Transformación de los atributos multievaluados** (pueden tener varios valores). Este atributo se convierte en una entidad relacionada con la entidad de la que procede. Para la nueva entidad se elegirá un nombre adecuado y tendrá un único atributo (el antiguo). Este atributo es posible que funcione correctamente como clave primaria de la entidad, pero a veces es posible que no.
Ejemplo: en la entidad EMPLEADO, el atributo "email" es multivaluado porque puede tener varios correos. Se realizará una nueva entidad EMAIL que sería una entidad débil de EMPLEADO



PASOS PARA TRANSFORMAR MODELO E/R AL RELACIONAL

1. Transformación entidades fuertes (no débiles)
2. Transformación de atributos
3. Transformación de entidades débiles
4. Transformación de relaciones
 - N:M
 - 1:N
 - 1:1
 - Atributos de relación
5. Transformación de relaciones reflexivas
 - 1:1
 - 1:N
 - M:N
6. Participaciones
7. Transformación del modelo E/R extendido



TRANSFORMAR ENTIDADES FUERTES (NO DÉBILES)

- Toda entidad se transforma en una relación o tabla.
- Para cada entidad A, entidad fuerte, con atributos (a_1, a_2, \dots, a_n) se crea una tabla A con n columnas correspondientes a los atributos de A, donde cada fila de la tabla A corresponde a una ocurrencia de la entidad A.
- Cada tipo de entidad se debe convertir en una relación, es decir será necesario crear una tabla para cada entidad que figure en el modelo entidad relación

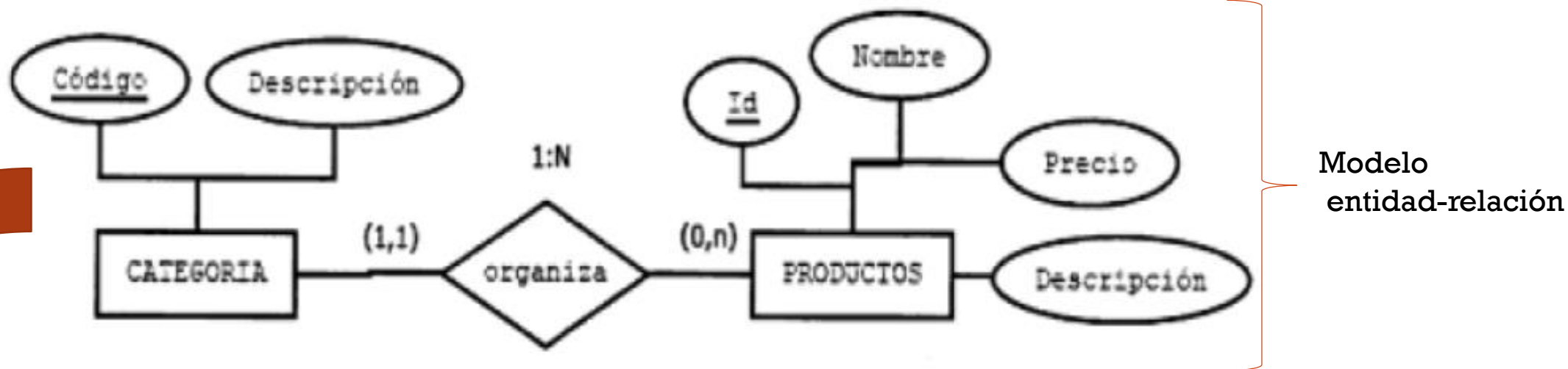


TRANSFORMAR ATRIBUTOS

- Todo atributo se transforma en columna dentro de una tabla.
- El atributo o atributos principales (los que actúan como identificador de la entidad) pasa a ser la clave principal de la relación y se subrayarán para identificarla como la clave principal de la relación (tabla). Además, deben ser no nulos.
- El resto de atributos pasan a ser otras columnas de la tabla pudiendo ser nulos o no nulos.
- Si hay atributos que se pueden calcular a partir de otros, no los almacenaremos. Por ejemplo, si en persona tenemos la edad y la fecha de nacimiento, solo almacenaremos la fecha de nacimiento. Procuramos guardar el atributo que no cambie con el tiempo



EJEMPLO



Paso a tablas de entidades fuertes

CATEGORÍA
Pk código
descripción

PRODUCTO
Pk id
nombre
precio
descripción

CATEGORÍA (Código, Descripción)

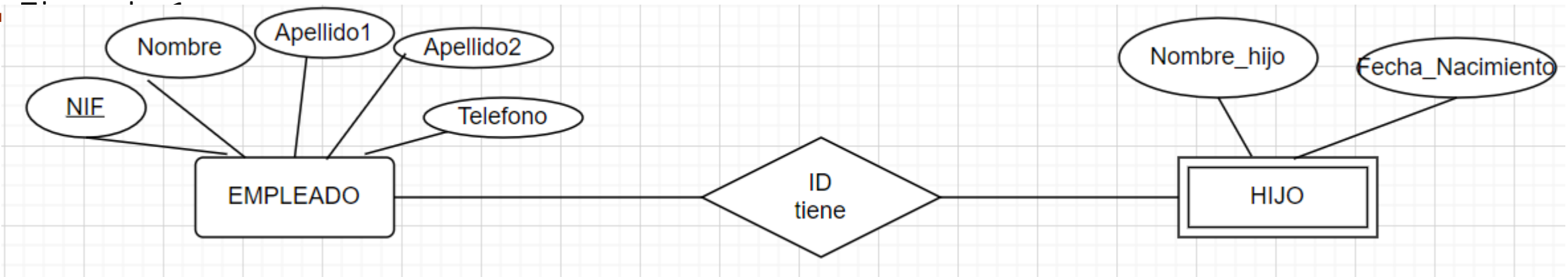
PRODUCTOS (Id, Nombre, Precio, Descripción)

Modelo lógico



TRANSFORMAR ENTIDADES DÉBILES

- Convertir toda entidad débil a fuerte.
- Convertir una entidad débil a fuerte consiste en hacer que cada entidad débil genere una tabla que incluirá todos sus atributos, añadiéndose a esta los atributos que son clave primaria de la entidad fuerte con la que está relacionada. Estos atributos añadidos serán clave foránea que referencia a la entidad fuerte.
- A continuación, se escogerá una clave primaria para la tabla creada.



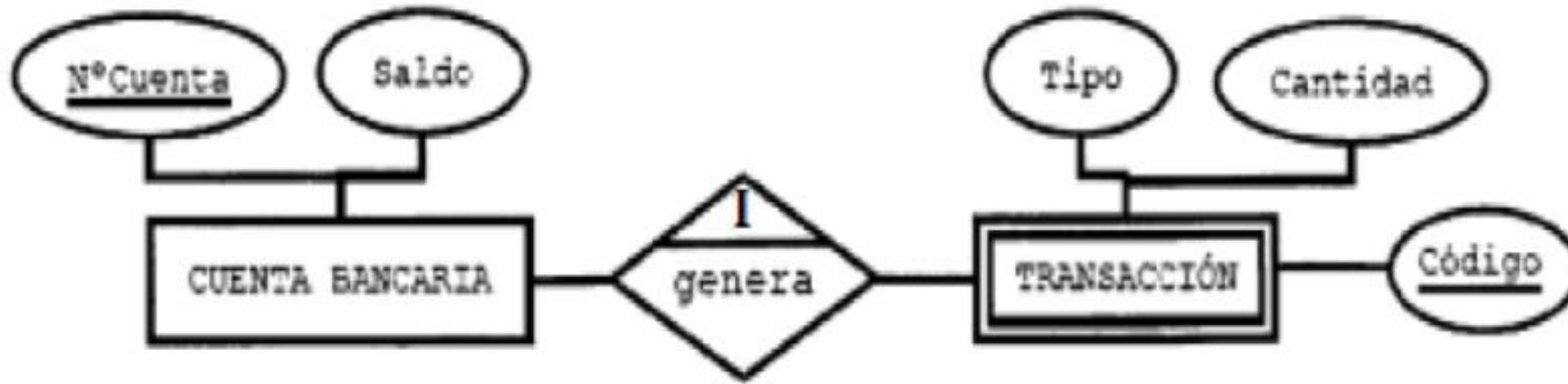
EMPLEADO(NIF, Nombre, Apellido1, Apellido2, Telefono)

HIJO (NIF, Nombre_Hijo, Fecha_Nacimiento)



TRANSFORMAR ENTIDADES DÉBILES

- Ejemplo



Modelo
entidad-relación

CUENTAS_BANCARIA
Pk <u>nº cuenta</u>
saldo

TRANSACCION
Pk Fk <u>nº cuenta</u>
Pk <u>código</u>
tipo
cantidad

CUENTA_BANCARIA (N°Cuenta, saldo)

TRANSACCIÓN (Código, Nº Cuenta, Tipo, Cantidad)

Modelo lógico



TRANSFORMAR RELACIONES

Recordamos que teníamos tres tipos de relaciones.

- Relación 1:1 → Esta cardinalidad especifica que una entidad A puede estar vinculada mediante una relación a una y sola una ocurrencia de otra entidad B. A su vez, una ocurrencia de la entidad B solo puede estar vinculada a una ocurrencia de la entidad A
 - Ejemplo: Un empleado solo puede ser jefe de un departamento y un departamento solo puede tener un jefe
- Relación 1:N → Especifica que una entidad A puede estar vinculada mediante una relación a varias ocurrencias de otra entidad B. Sin embargo, una de las ocurrencias de la entidad B solo puede estar vinculada a una ocurrencia de la entidad A
 - Ejemplo: Una persona puede tener varios vehículos pero un vehículo pertenece solo a una persona
- Relación N:M → Especifica que una entidad A puede estar vinculada mediante una relación a varias ocurrencias de la entidad B. A su vez una ocurrencia de la entidad B puede estar vinculada a varias de la entidad A.
 - Ejemplo: Un empleado puede trabajar para varios proyectos al mismo tiempo y en un mismo proyecto pueden trabajar varios empleados.



TRANSFORMAR RELACIONES

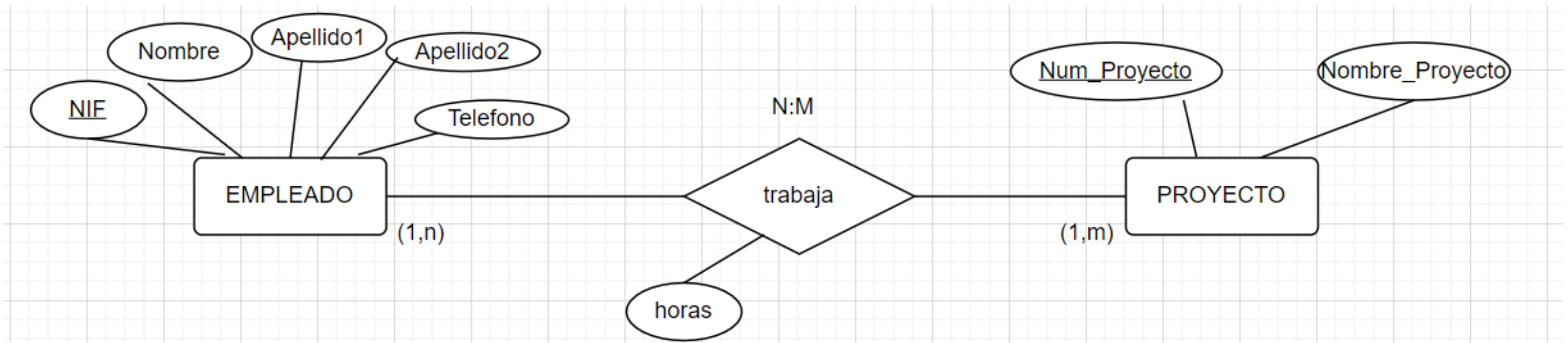
- Como norma general, se añade una tabla para la relación con los atributos de ésta y como clave primaria la composición de las claves de las otras entidades. Las relaciones N:M o las ternarias se comportan así.
- Hay que estudiar qué ocurre con el BORRADO y MODIFICACIÓN de la PK REFERENCIADA. Operaciones permitidas:
 - Operación Restringida: Por defecto no se permite ni borrar ni modificar.
 - Operación puesta a nulos: Borrado o Modificado SET NULL
 - Operación de valor por defecto: Borrado o Modificado SET DEFAULT
 - Operación en cascada: Borrado o Modificado en cascada. CASCADE.
- No siempre se aplica la regla general para crear una tabla por cada relación. Generalmente, se pueden encontrar excepciones a la regla general



TRANSFORMAR RELACIONES N:M

Sigue la regla general

- Se transforma en una relación (tabla) que tendrá como clave primaria la concatenación de los atributos principales de cada una de las entidades que relacionan que **serán clave ajena respecto** a cada una de las tablas donde ese atributo es clave primaria,
- Nota: habrá que considerar qué sucede en los casos en los que se borre o modifique la clave primaria referenciada.



EMPLEADO(NIF, Nombre, Apellido1, Apellido2, Telefono)

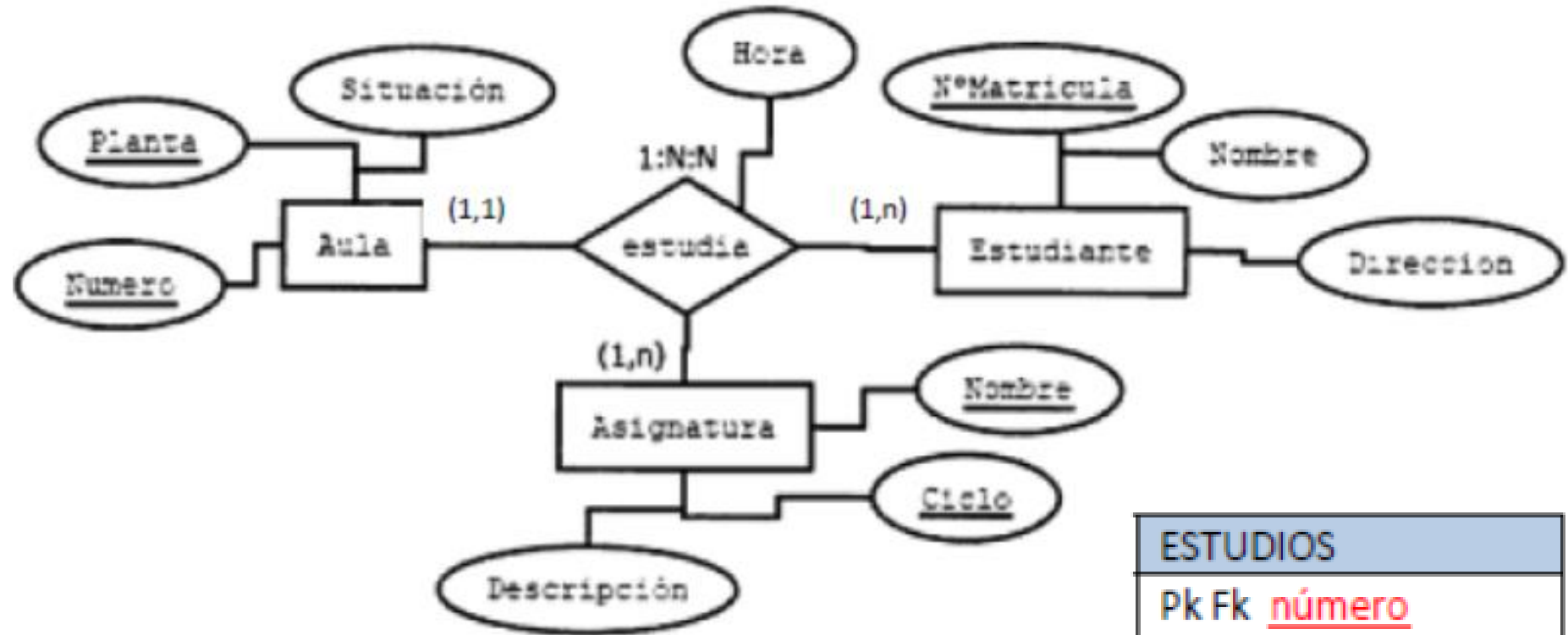
Trabaja_en(NIF, Num Proyecto, Horas)

PROYECTO(Num Proyecto, Nombre_Proyecto)



TRANSFORMAR RELACIONES N:M

■ Ejemplo 2:



AULAS
Pk <u>número</u>
Pk <u>planta</u>
situación

ESTUDIANTES
Pk <u>nº matrícula</u>
nombre
dirección

ASIGNATURAS
Pk <u>nombre</u>
Pk <u>ciclo</u>
descripción

ESTUDIOS
Pk Fk <u>número</u>
Pk Fk <u>planta</u>
Pk Fk <u>nº matrícula</u>
Pk Fk <u>nombre</u>
Pk Fk <u>ciclo</u>
hora



TRANSFORMAR RELACIONES N:M

- Ejemplo 2:

AULAS
Pk <u>número</u>
Pk <u>planta</u>
situación

ESTUDIANTES
Pk <u>nº matrícula</u>
nombre
dirección

ASIGNATURAS
Pk <u>nombre</u>
Pk <u>ciclo</u>
descripción

ESTUDIOS
Pk Fk <u>número</u>
Pk Fk <u>planta</u>
Pk Fk <u>nº matrícula</u>
Pk Fk <u>nombre</u>
Pk Fk <u>ciclo</u>
hora

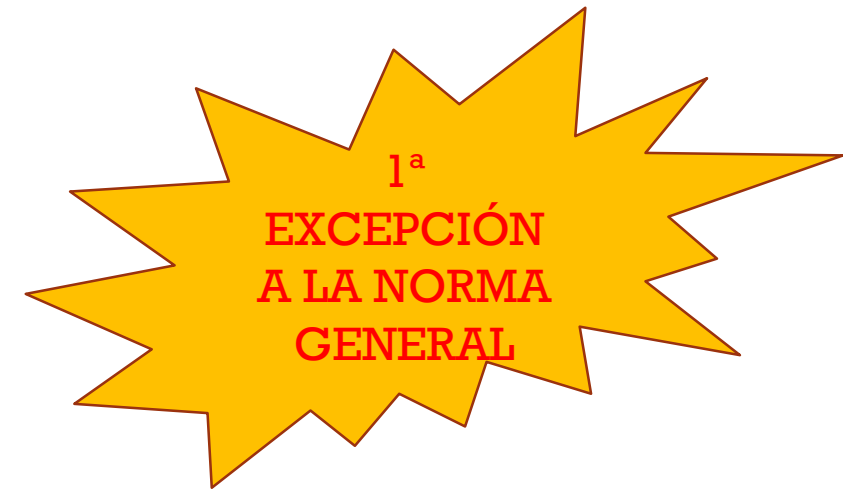
- Observación: aunque en teoría, la tabla ESTUDIOS tiene como clave primaria la suma de las claves primarias de las tablas que relaciona, tener en una base de datos tablas con claves tan complejas, hace que el sistema pueda funcionar más lento de lo esperado debido a la multitud de comprobaciones que el gestor debe realizar cuando se inserta o modifica un dato.
Si es un sistema cuyo funcionamiento se basa en la inserción o modificación constante de datos, más que en la consulta de los mismos, quizá, en estos casos, se pueda saltar la teoría y crear un campo sencillo adicional, identificador de la fila, y sustituirlo por la clave primaria compuesta original. De esta forma, se simplifica enormemente la clave primaria en pos de un funcionamiento más eficiente.



TRANSFORMAR RELACIONES 1:N

Existen dos opciones:

1. Propagar el atributo principal de la entidad que tiene cardinalidad máxima 1 a la que tiene N, y hacer desaparecer la relación como tal
2. Transformar en relación tipo N:M. Esto se hará en tres casos:
 - a. Si es posible que aparezcan muchos nulos porque existen pocos elementos relacionados
 - b. Cuando se prevé que dicha relación pasará en un futuro a ser tipo N:M
 - c. Cuando la relación tiene atributos propios



COMO NORMA GENERAL Y PARA EL EXAMEN, PRIORIZAMOS LA EFICIENCIA Y UTILIZAMOS LA SOLUCIÓN 1

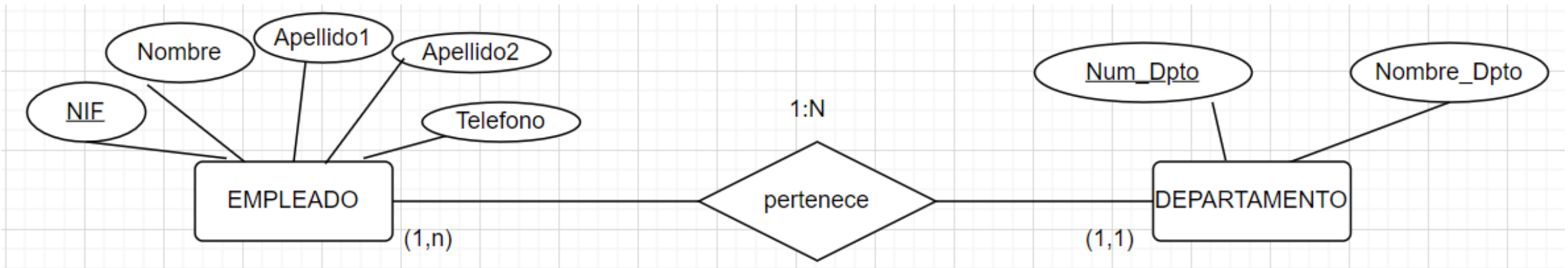


TRANSFORMAR RELACIONES 1:N

Existen dos opciones:

1. Propagar el atributo principal de la entidad que tiene cardinalidad máxima 1 a la que tiene N, y hacer desaparecer la relación como tal

Ej:



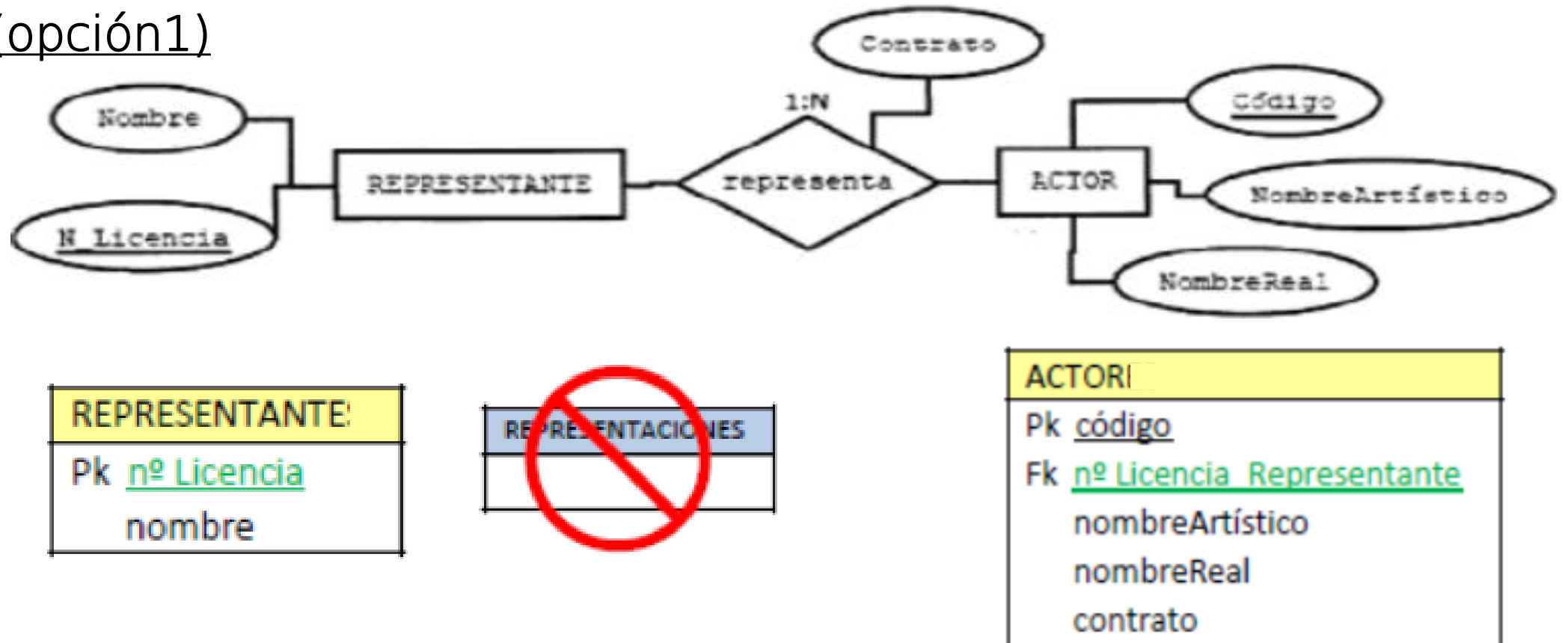
EMPLEADO(NIF, Nombre, Apellido1, Apellido2, Telefono, Num_Dpto)

DEPARTAMENTO(Num_Dpto, Nombre_Dpto)



TRANSFORMAR RELACIONES 1:N

Ejemplo 2 (opción1)



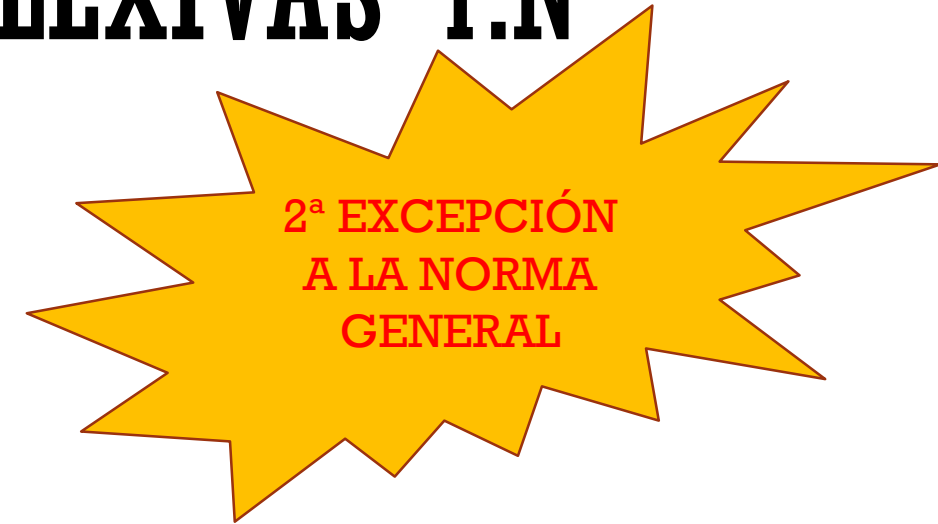
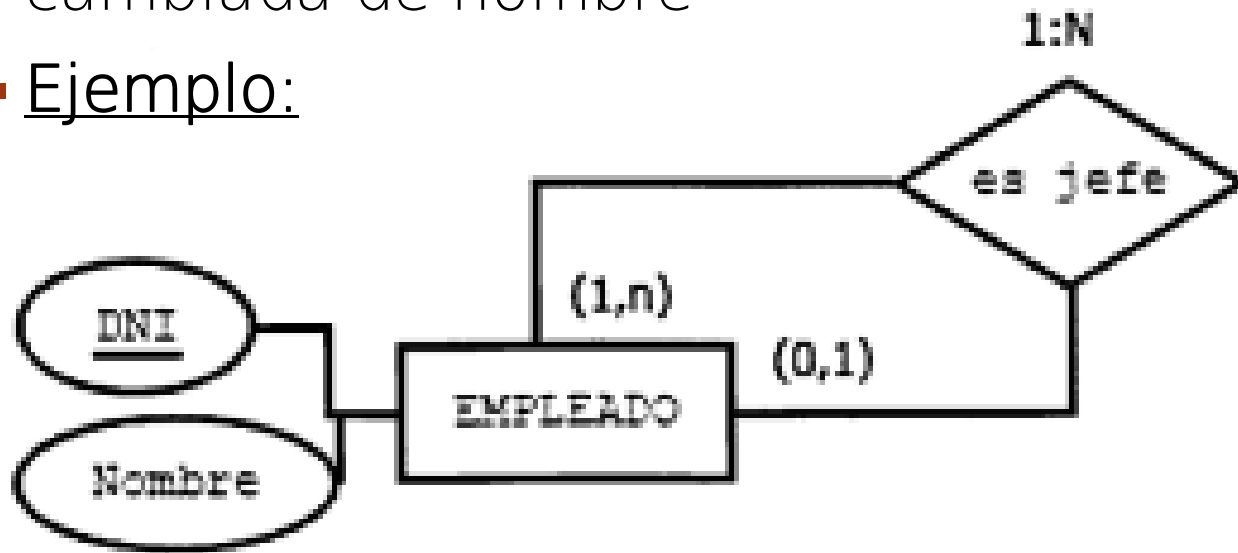
REPRESENTANTE(NºLicencia, Nombre,)

ACTOR(Codigo, NombreArtistico, NombreReal, NºLicencia_Representante, contrato)

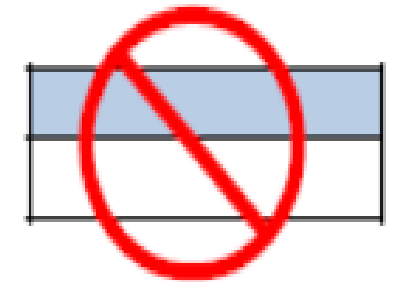


TRANSFORMAR RELACIONES REFLEXIVAS 1:N

- En este caso tampoco se crea una tabla para la relación.
- Se crea una tabla con el nombre de la entidad, añadiendo otra vez la clave cambiada de nombre
- Ejemplo:



EMPLEADOS
Pk <u>DNI</u>
nombre
Fk dniJefe



EMPLEADO (DNI, Nombre, DNIjefe)



TRANSFORMAR RELACIONES 1:1

- Este tipo de relaciones **tampoco** generan tabla.
- El paso a tablas se realiza de forma parecida a las relaciones 1:N
- Se tiene la libertad de poder incorporar la clave de una de las dos entidades a la otra



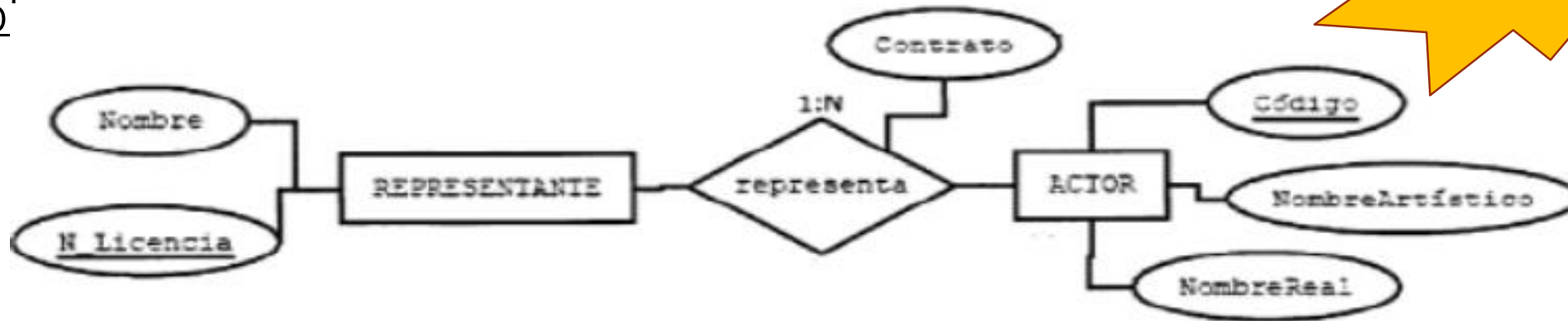
**3ª EXCEPCIÓN
A LA NORMA
GENERAL**



TRANSFORMAR RELACIONES 1:1

3^a
EXCEPCIÓN
A LA NORMA
GENERAL

■ Ejemp'



REPRESENTANTE
Pk <u>nº Licencia</u>
nombre

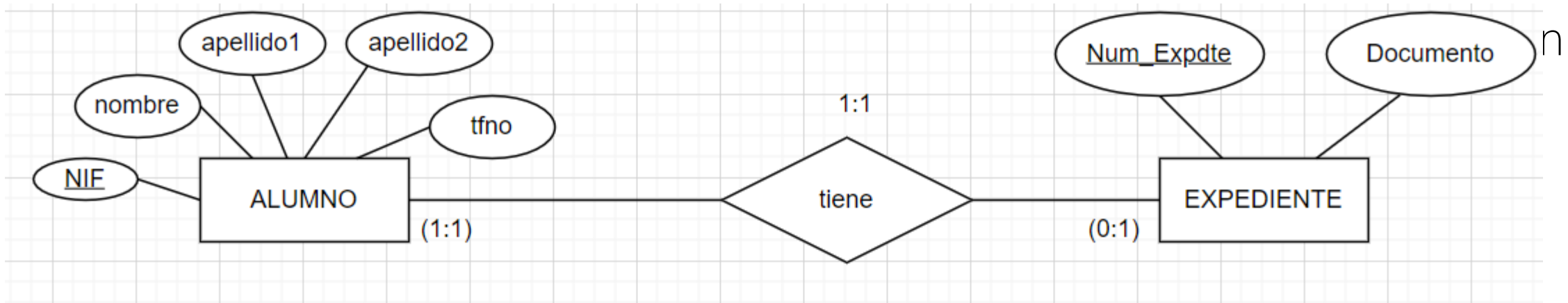
REPRESENTACIONES

ACTOR
Pk <u>código</u>
Fk <u>nº Licencia Representante</u>
nombreArtístico
nombreReal
contrato



TRANSFORMAR RELACIONES 1:1

- Si tenemos cardinalidades $(1,1) : (1,1) \rightarrow$ propagamos la clave en la dirección que queramos y no pasamos a tabla la relación
- Si tenemos cardinalidades $(0,1) : (1,1) \rightarrow$ propagamos la clave hacia el $(0,1)$ y no pasamos a tabla la relación
- Si tenemos cardinalidades $(1,1) : (0,1) \rightarrow$ propagamos la clave hacia el $(0,1)$ y no pasamos a tabla la relación



ALUMNO(NIF, Nombre, Apellido1, Apellido2, Tfno)

EXPEDIENTE (Num_Expdte, Documento, NIF)



TRANSFORMAR LOS ATRIBUTOS DE LAS RELACIONES

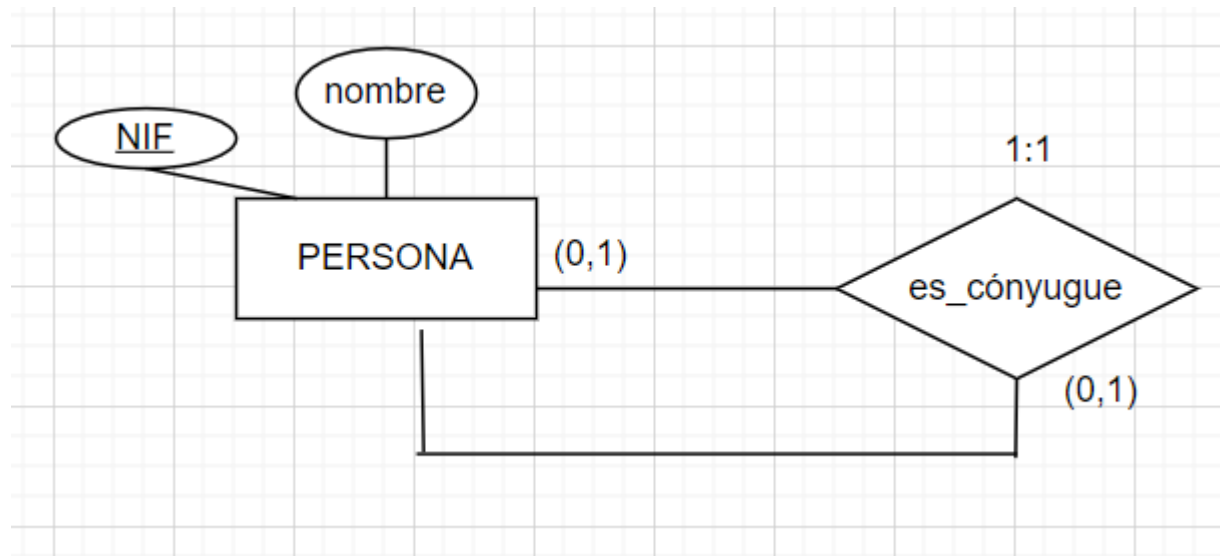
- Si la relación se transforma en una tabla, todos sus atributos pasan a ser columnas de la tabla.
- Si alguno de los atributos de la relación es principal, entonces deberá ser incluido como parte de la clave principal en dicha tabla.



TRANSFORMAR RELACIONES REFLEXIVAS

En función de la cardinalidad de la relación:

- Relación 1:1 → La clave de la entidad se repite, con lo que la tabla resultante tendrá dos veces ese atributo, una como clave primaria y otra como clave ajena de la misma.



PERSONA(NIF, nombre, NIF_conyugue)

NIF_conyugue es clave ajena en la tabla PERSONA

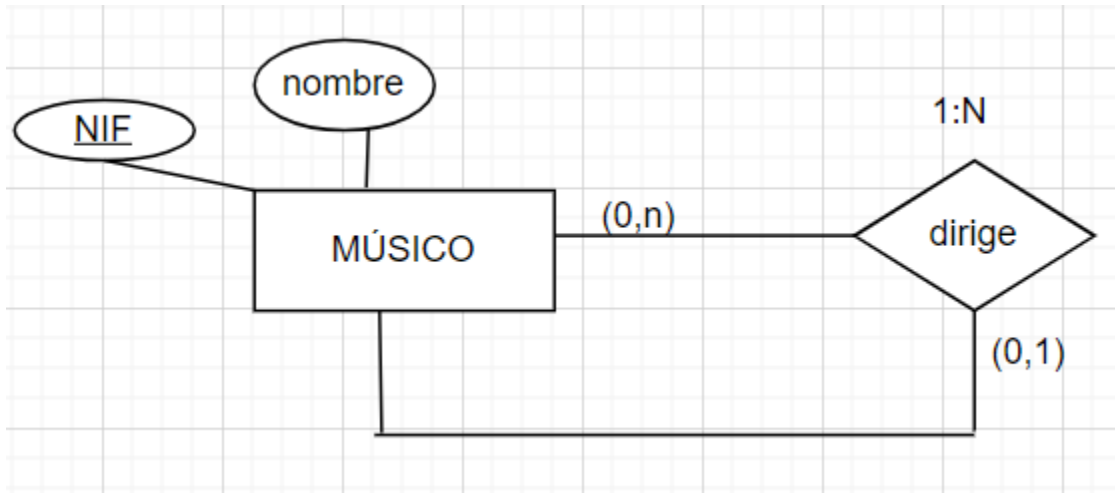


TRANSFORMAR RELACIONES REFLEXIVAS

En función de la cardinalidad de la relación:

- Relación 1:N. Tendríamos dos casos

1. Si la entidad muchos es siempre obligatoria, actuaremos con en el caso 1:1
2. Si la entidad muchos no es obligatoria se creará una nueva tabla cuya clave será la de la entidad del lado muchos, y se propaga la clave hacia



MÚSICO (NIF, nombre)

Dirige(NIF, NIF_director)

NIF_director es clave ajena

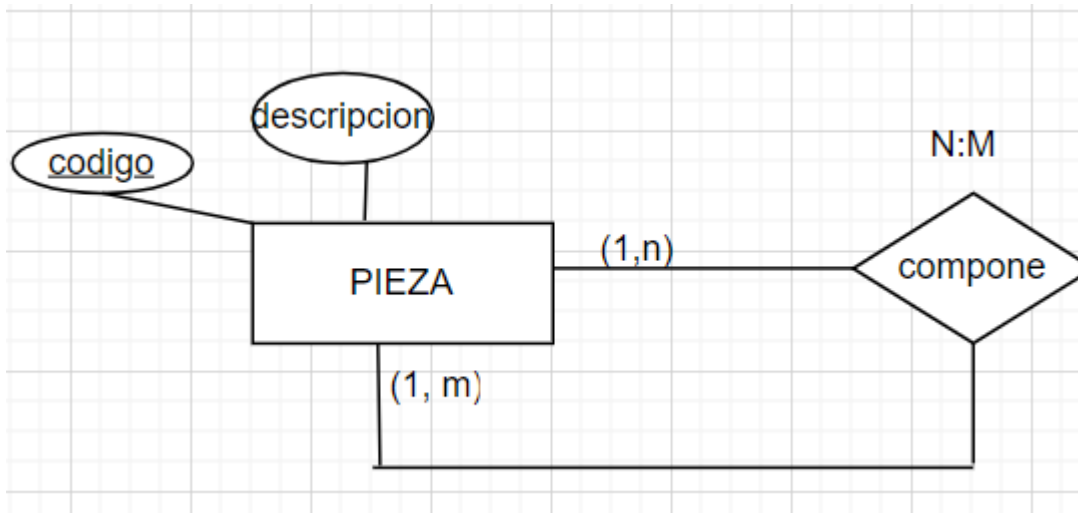


TRANSFORMAR RELACIONES REFLEXIVAS

En función de la cardinalidad de la relación:

- **Relación N:M**. Se trata como una relación binaria entre dos entidades por lo que se crea una nueva tabla que tendrá dos veces la clave primaria de la entidad. La clave de esta tabla será la combinación de ambas claves primarias.

Si hubiera atributos asociados a la relación, se asociarían a esta nueva tabla



PIEZA (código, descripción)

PIEZA_COMPUESTA (código, código componente)



PARTICIPACIONES

- Son importantes para calcular la cardinalidad de la relación y aplicar las transformaciones que correspondan.
- En algunos diagramas entidad-relación las participaciones se omiten. No obstante es necesario tener en cuenta cuándo la participación tiene un mínimo de 0, para adoptar un campo de una tabla como opcional NULL u obligatorio NOT NULL



TRANSFORMACIÓN MOELO E/R EXT

- Como los TIPOS y los SUBTIPOS no tienen una representación directa en el Modelo Relacional hay varias soluciones de transformación
 1. Crear una sola relación. → R (supertipo + subtipos)
 2. Crear dos relaciones. → R1(subtipo1)
R2(subtipo2)
 3. Crear tres relaciones. → R1(supertipo)
R2(subtipo1)
R3(subtipo2)



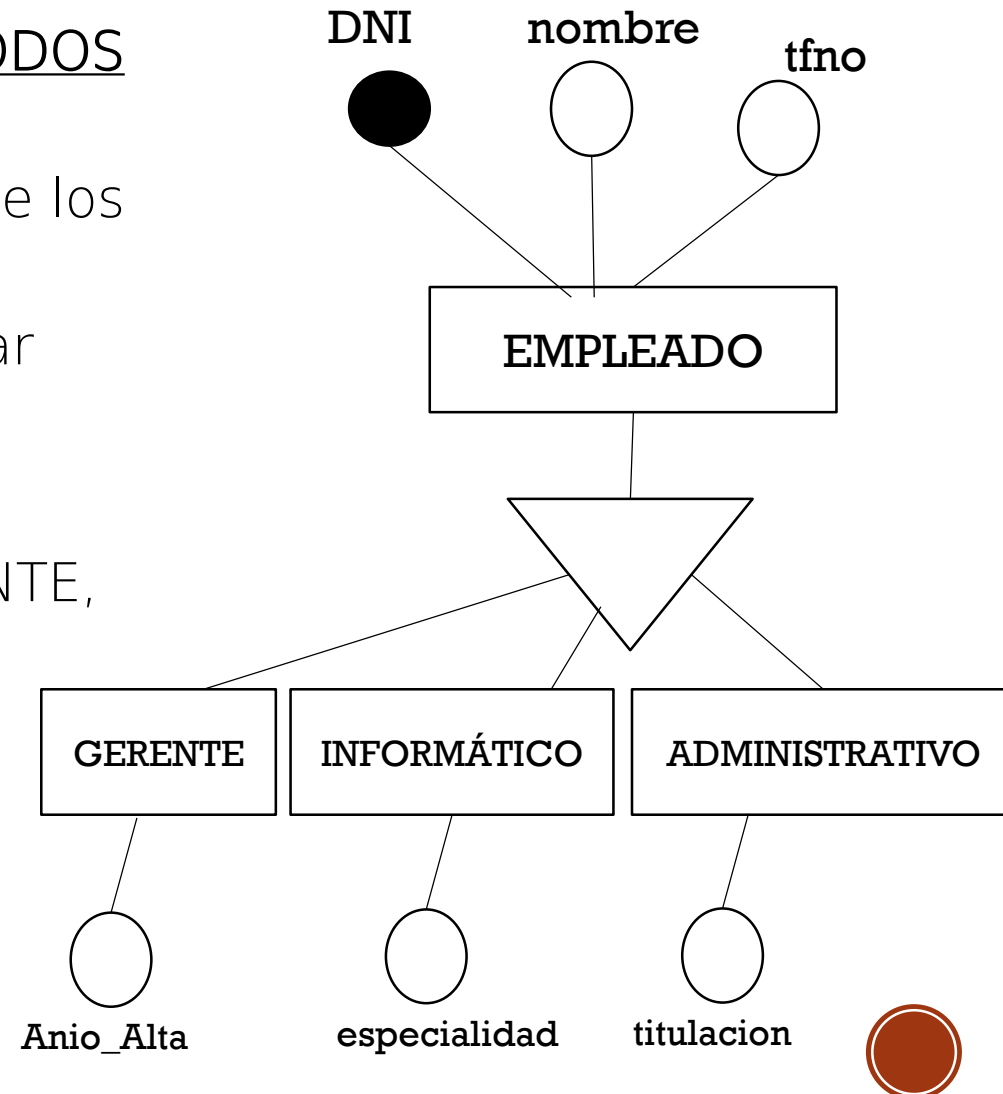
TRANSFORMACIÓN MODELO E/R EXT

CREAR UNA ÚNICA RELACIÓN O TABLA QUE AGLUTINE TODOS LOS SUBTIPOS

- En ella se incluirán todos los atributos del supertipo y de los subtipos.
- Esto permite mayor simplicidad, aunque puede provocar valores nulos en los atributos propios de cada subtipo.
- Por este motivo no es una opción que suela utilizarse.
- Ejemplo: Entidad EMPLEADO, como supertipo de GERENTE, INFORMÁTICO Y ADMINISTRATIVO como subtipos.

EMPLEADO (DNI, Nombre, Tfno, Año_Alta, especialidad, titulo, tipo)

Donde “tipo” se podría representar como 1 para “GERENTE”, 1 para “INFORMÁTICO” y 2 “ADMINISTRATIVO”



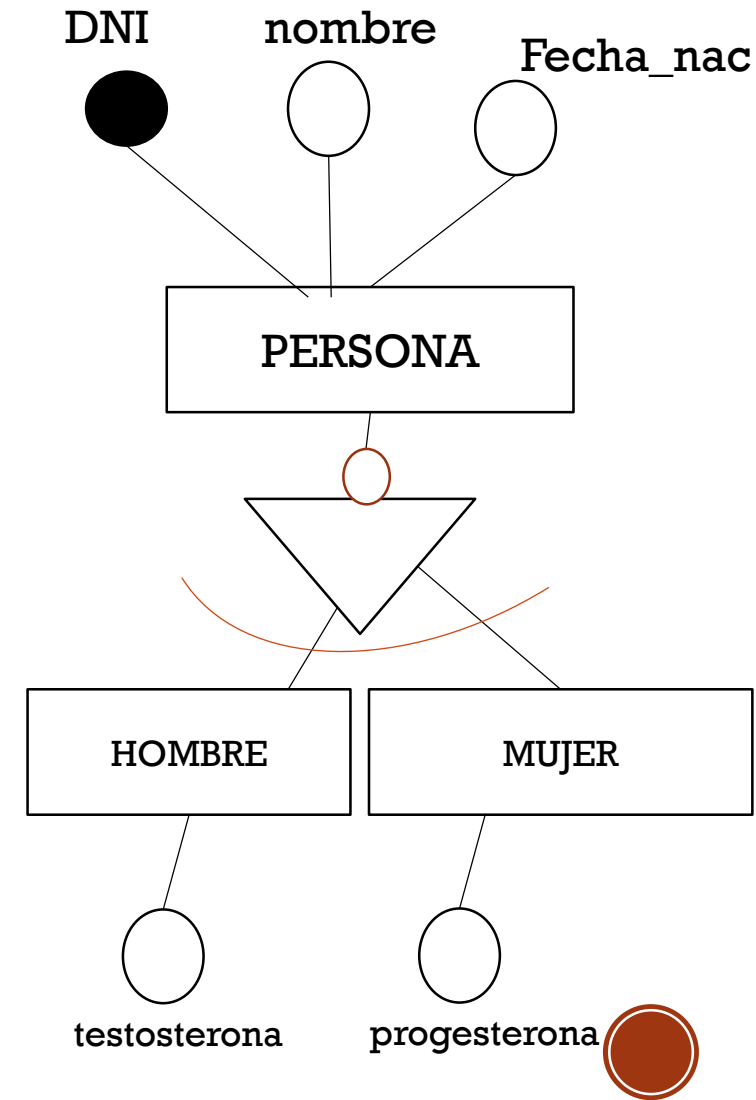
TRANSFORMACIÓN MODELO E/R EXT

CREAR RELACIÓN O TABLA SOLO PARA LOS SUBTIPOS

- En este caso los subtipos heredan todos los atributos del supertipo.
- Esta opción también se utiliza cuando la especialización es total y exclusiva y, sobre todo, si el supertipo o padre no tiene relaciones y con los hijos los que sí tienen relaciones.
- Siguiendo con mismo ejemplo anterior

HOMBRE (DNI, Nombre, Fecha_nac ,Testosterona)

MUJER (DNI, Nombre, Fecha_nac ,Progesterona)





Invoice

Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark

UT3. DISEÑO LÓGICO. MODELO RELACIONAL

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz

