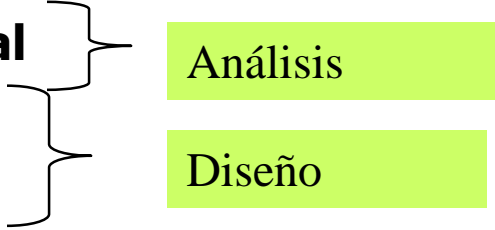


Material complementarios

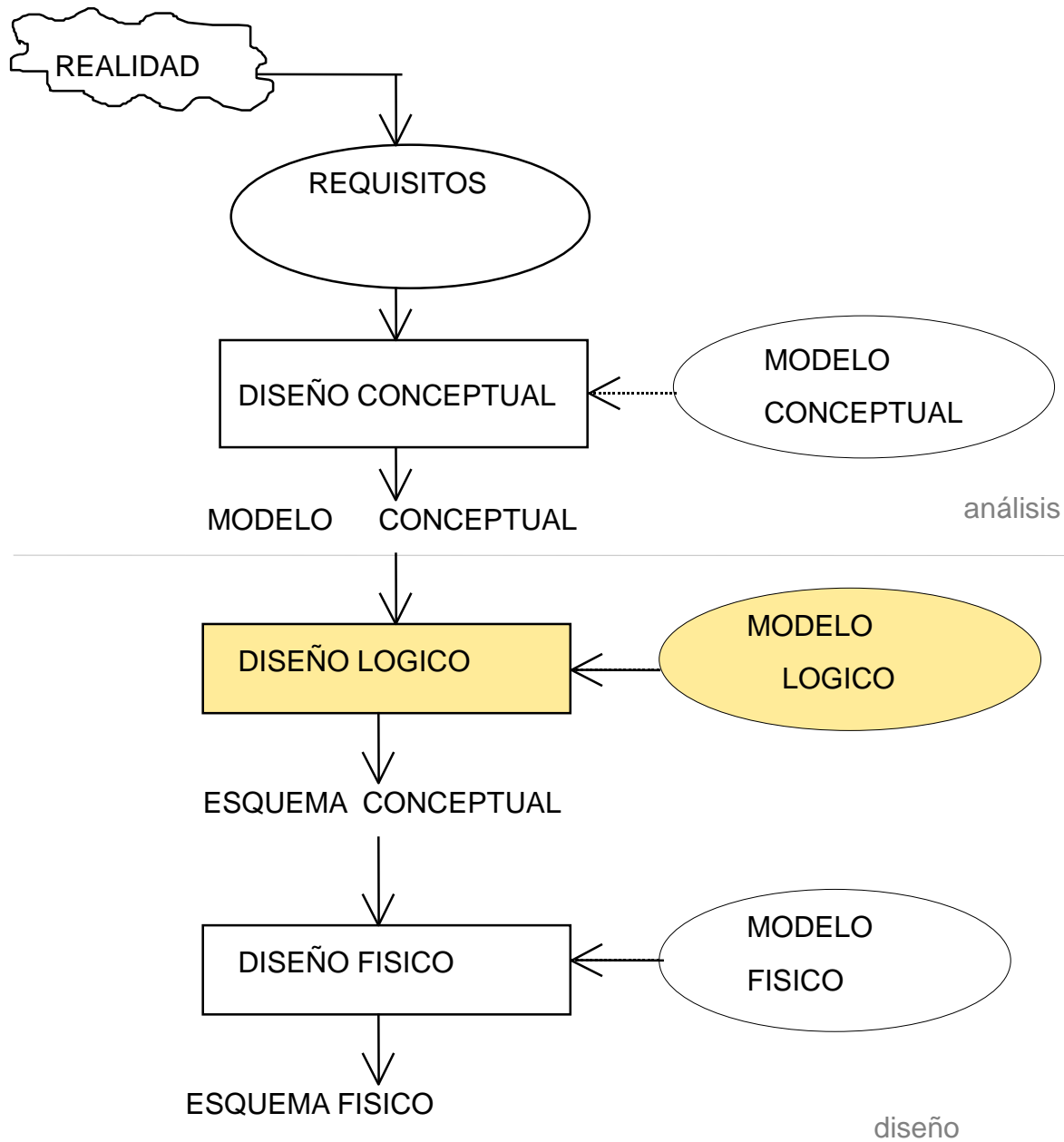
3. Transformación MER→Modelo Relacional

Contenidos:	Epígrafes unidades:
1. Diagrama entidad/relación extendido	Unidad 3 (1.,2.,3.,4.,5.,6.,7.,8.)
2. Modelo relacional	Unidad 2 (1.,2.,3.,4.,5.)
3. Transformación MER-Modelo relacional	Unidad 3 (9.,10.)
4. Normalización	Unidad 3 (11.)

Introducción

- ❑ **El diseño de datos es el proceso por el que se determina la organización de la información necesaria para el sistema.**
- ❑ **Se realiza generalmente en tres fases:**
 - **Diseño conceptual**
 - **Diseño lógico**
 - **Diseño físico**

	}	Análisis
	}	
	}	Diseño
- ❑ **Cada fase es un proceso iterativo, se van produciendo refinamientos sucesivos antes de pasar a la siguiente fase**



Objetivos de cada fase:

❑ Diseño conceptual:

Representación de los recursos de información del sistema, con independencia de usuarios, aplicaciones en particular, y fuera de consideraciones sobre dispositivos físicos

❑ Diseño lógico

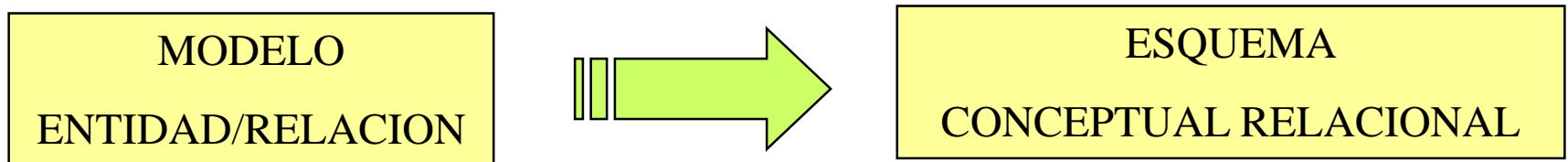
Transformación del modelo conceptual de datos obtenido en la etapa anterior, adaptándolo al sistema de gestión de datos que se vaya a utilizar y definición de especificaciones para los programas de acceso a los datos

❑ Diseño físico

Estructuración y almacenamiento del conjunto de datos del modelo lógico anterior, en un determinado dispositivo físico, optimizando la relación costes/beneficios

Transformación diseño conceptual a lógico relacional

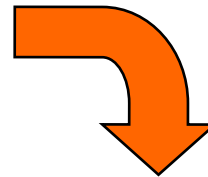
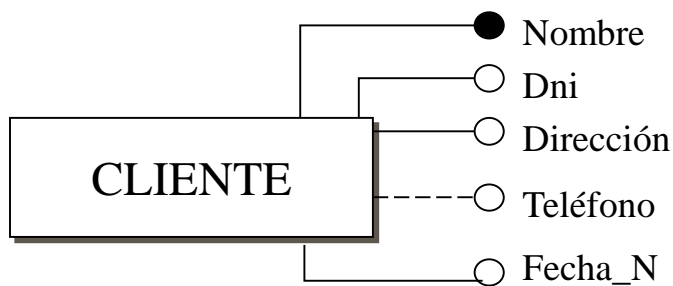
- ❑ El resultado de la etapa de diseño conceptual es el modelo Entidad Relación
- ❑ En la etapa de diseño lógico se obtiene el esquema conceptual a partir del modelo entidad-relación



- ❑ El paso del modelo entidad-relación a esquema conceptual relacional se basa en los tres principios siguientes:
 1. Toda entidad se convierte en una tabla (relación)
 2. Toda interrelación N:M se convierte en una tabla (relación)
 3. Toda interrelación 1:1 y 1:N se traduce en el fenómeno de propagación de claves

Transformación diseño conceptual a lógico relacional

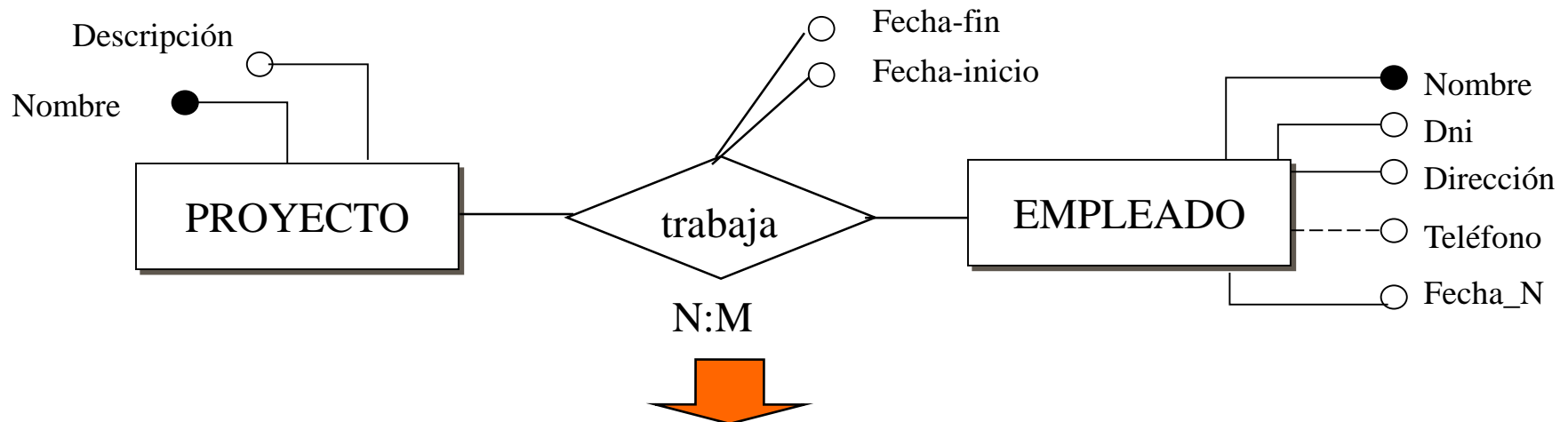
1. Toda entidad se convierte en una tabla o relación que toma el nombre de la entidad.
 - Los atributos de la entidad serán las columnas de la tabla y el atributo identificador principal será la clave primaria.
 - A menos que se indique lo contrario los atributos no identificadores no podrán tomar valores nulos.
 - Si tiene atributos compuestos o multivaluados hay que decidir cómo transformarlos, como veremos más adelante.



CLIENTE (NOMBRE, DNI, DIRECCIÓN, TELEFONO*, FECHA_N)

Transformación diseño conceptual a lógico relacional

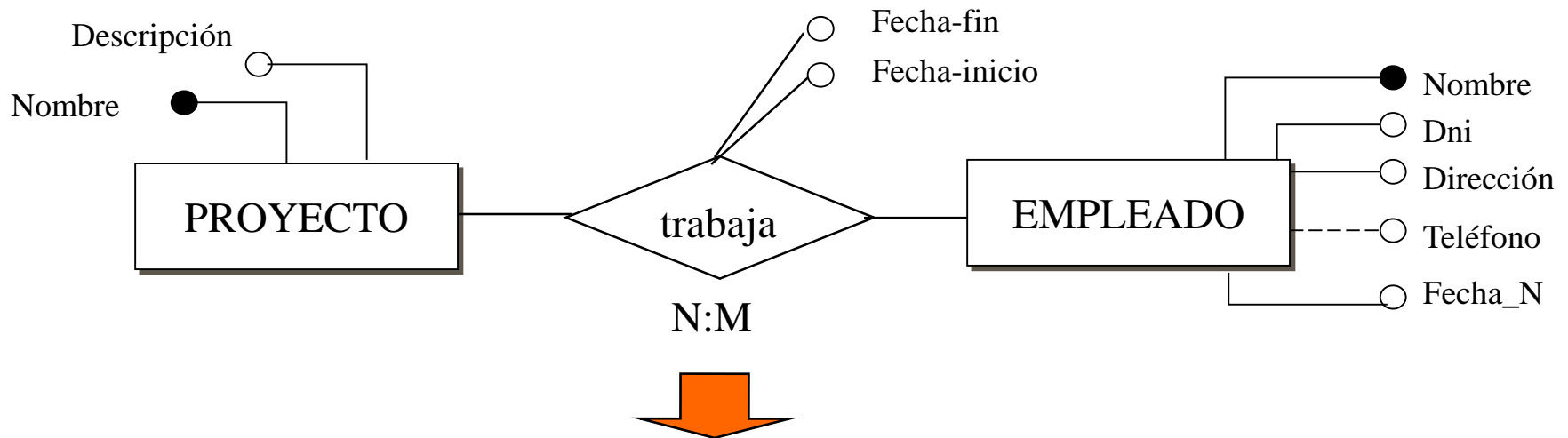
2. Toda interrelación N:M se convierte en una tabla o relación (independientemente de la cardinalidad mínima)
- La clave primaria será la concatenación de las claves primarias de las entidades que se asocia (estos atributos, a su vez, serán claves ajenas que referencian a las respectivas tablas donde son claves primarias).
 - Los atributos de la interrelación serán columnas de la tabla



TRABAJA (NOMBRE-E, NOMBRE-P, FECHA_INICIO, FECHA-FIN)

Transformación diseño conceptual a lógico relacional

2. Toda interrelación N:M se convierte en una tabla o relación (independientemente de la cardinalidad mínima)



EMPLEADO (NOMBRE, DNI, DIRECCIÓN, TELEFONO*, FECHA_N)

PROYECTO (NOMBRE, DESCRIPCION)

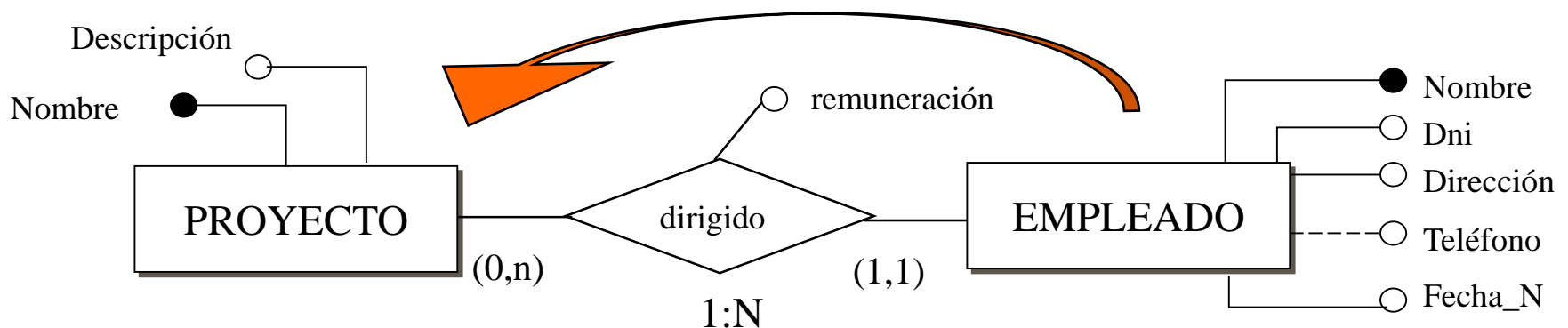
TRABAJA (NOMBRE-E, NOMBRE-P, FECHA_INICIO, FECHA-FIN)

La relación TRABAJA se transforma en una tabla con las claves de PROYECTO y EMPLEADO que juntas son clave primaria y a su vez por separado son claves ajenas

Transformación diseño conceptual a lógico relacional

3. Las interrelaciones 1:N o 1:1 se transforman propagando el atributo identificador principal de una entidad a otra.

- Si la relación fuese 1:1 la propagación de clave podría hacerse en cualquier sentido. El atributo propagado es una clave ajena que referencia a la otra tabla.
- Si la interrelación es 1:N se propaga la clave de la entidad de cardinalidad máxima 1 a la de N
- Los atributos de la interrelación serán columnas de la tabla a la que se ha propagado la clave



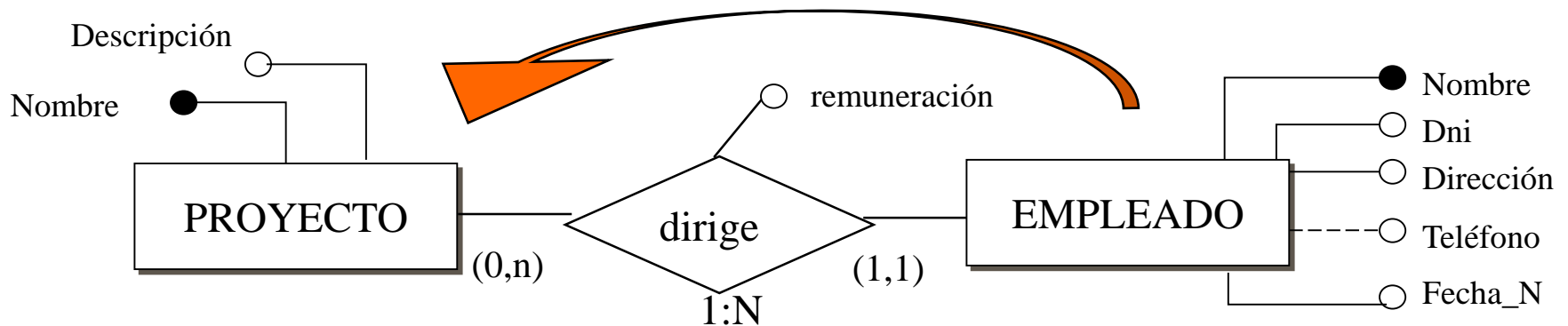
PROYECTO (NOMBRE, DESCRIPCION, DIRECTOR, REMUNERACIÓN)

EMPLEADO (NOMBRE, DNI, DIRECCIÓN, TELEFONO*, FECHA_N)

Transformación diseño conceptual a lógico relacional

3. Las interrelaciones 1:N o 1:1 se transforman propagando el atributo identificador principal de una entidad a otra.

- **Relaciones 1:N** se propaga la clave de la entidad de cardinalidad máxima 1 a la de N
- Los atributos de la interrelación serán columnas de la tabla a la que se ha propagado la clave



PROYECTO (NOMBRE, DESCRIPCION, *DIRECTOR*, REMUNERACIÓN)

EMPLEADO (NOMBRE, DNI, DIRECCIÓN, TELEFONO*, FECHA_N)

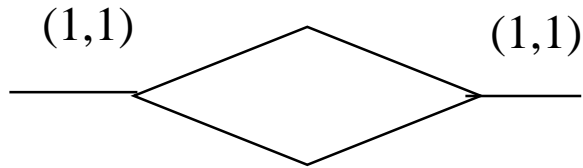
- Si la cardinalidad (0,n) fuera (1,n) habría que añadir una nota para indicar que en la tabla PROYECTO deben aparecer en la columna DIRECTOR todos los empleados (su clave primaria)

Transformación diseño conceptual a lógico relacional

3. Las interrelaciones 1:N o 1:1 se transforman propagando el atributo identificador principal de una entidad a otra, en general.
 - PERO dependiendo de las cardinalidades mínimas de la interrelación, se puede optar por otras transformaciones para evitar que las claves ajenas y los atributos tengan valores nulos

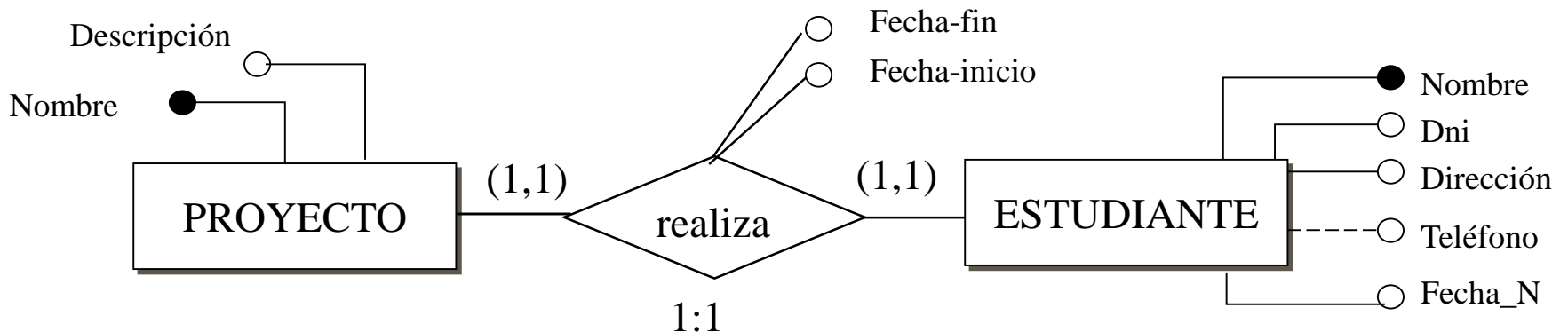
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:1



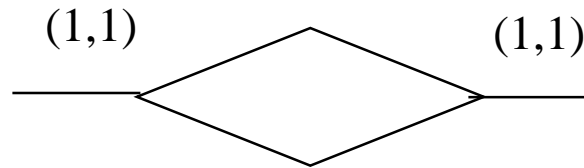
Ejemplo:

- Migrar la clave de una entidad a otra, junto con los atributos de la interrelación
- Integrar las dos entidades en una relación que contiene los atributos de las dos y de la interrelación, y de clave primaria una de las dos (al normalizar posiblemente incumpliría la 3FN y había que volver a la opción anterior)



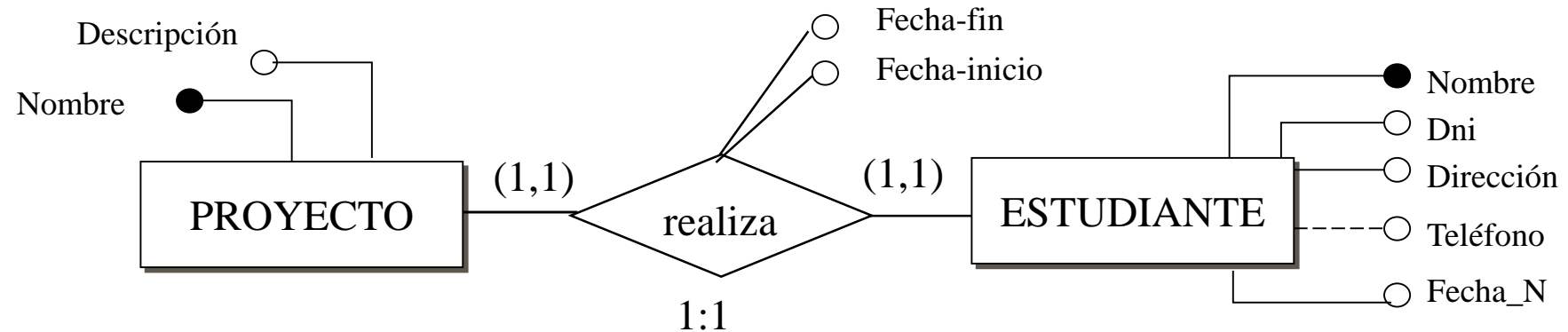
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:1



Ejemplo:

- Migrar la clave de una entidad a otra, junto con los atributos de la interrelación
- Integrar las dos entidades en una relación que contiene los atributos de las dos y de la interrelación, y de clave primaria una de las dos



- a) O se lleva la clave de proyecto a estudiante o la de estudiante a proyecto

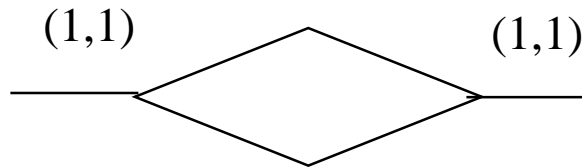
PROYECTO (NOMBRE, DESCRIPCION, *ESTUDIANTE*, FECHA_I, FECHA_F)

→ **ESTUDIANTE** (NOMBRE, DNI, DIRECCION, TELF* FECHA_NAC)

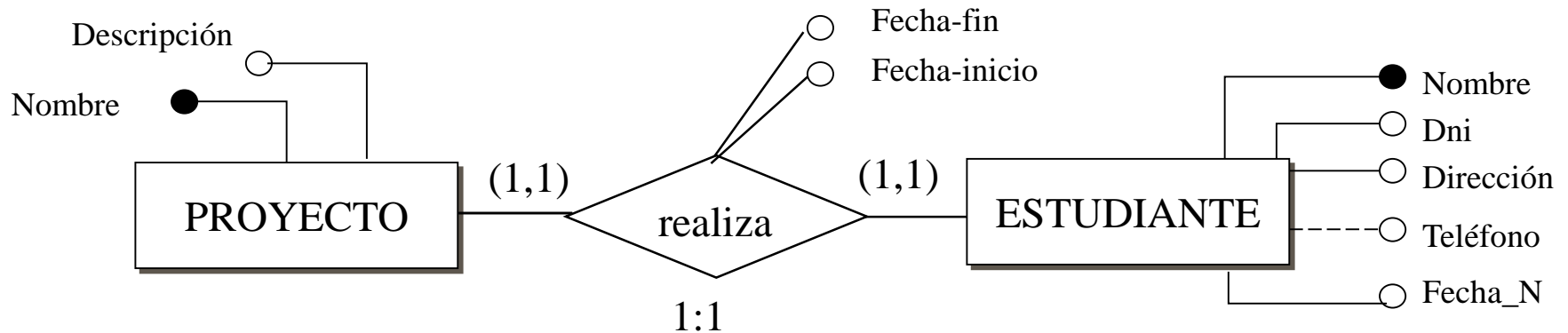
Los valores de ESTUDIANTE en PROYECTO no se pueden repetir, por lo que puede ser una clave alternativa, y tienen que estar todos

Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:1



Ejemplo:



a) Migar la clave de una entidad a otra, junto con los atributos de la interrelación

b) Integrar las dos entidades en una relación que contiene los atributos de las dos y de la interrelación, y de clave primaria una de las dos

a) O se lleva la clave de proyecto a estudiante o la de estudiante a proyecto

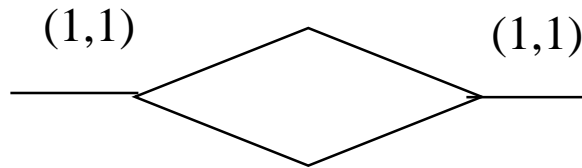
ESTUDIANTE (NOMBRE, DNI , DIRECCIÓN, TELF* FECHA_NAC,
PROYECTO, FECHA_I, FECHA_F)

→ **PROYECTO** (NOMBRE, DESCRIPCION, *ESTUDIANTE*,)

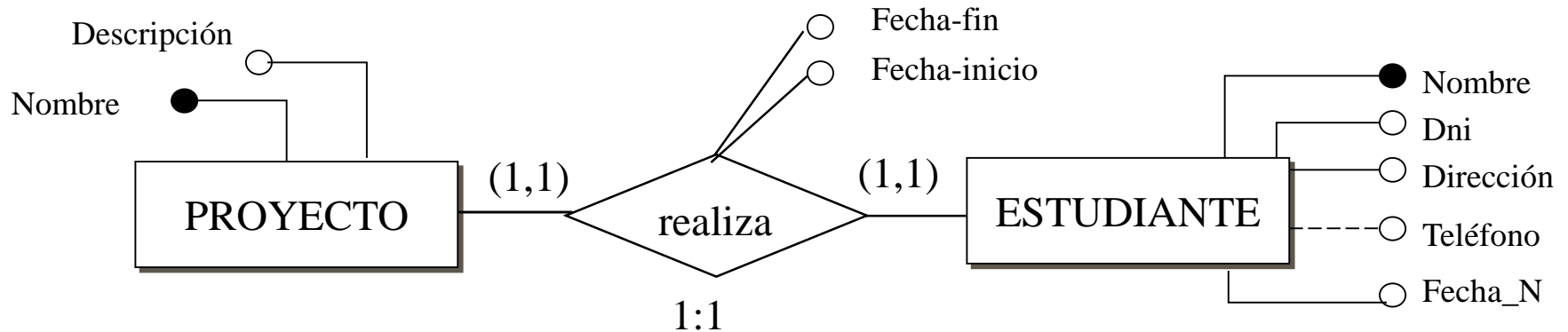
Los valores de PROYECTO en ESTUDIANTE no se pueden repetir, por lo que puede ser una clave alternativa, y tienen que estar todos

Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:1



Ejemplo:



a) Migrar la clave de una entidad a otra, junto con los atributos de la interrelación

b) Integrar las dos entidades en una relación que contiene los atributos de las dos y de la interrelación, y de clave primaria una de las dos

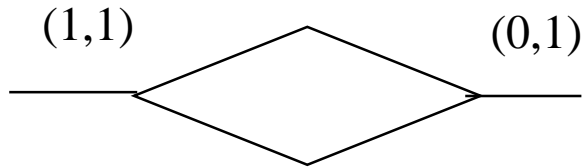
b) Se juntan en una entidad todos los atributos

ESTUDIANTE_PROYECTO (NOMBRE, DNI, DIRECCIÓN, TELF*, FECHA_NAC, NOMBREPROYECTO, DESCRIPCION, FECHA_I, FECHA_F)

Si la clave primaria es NOMBRE de estudiante, el nombre del proyecto será clave alternativa. También puede ser al revés

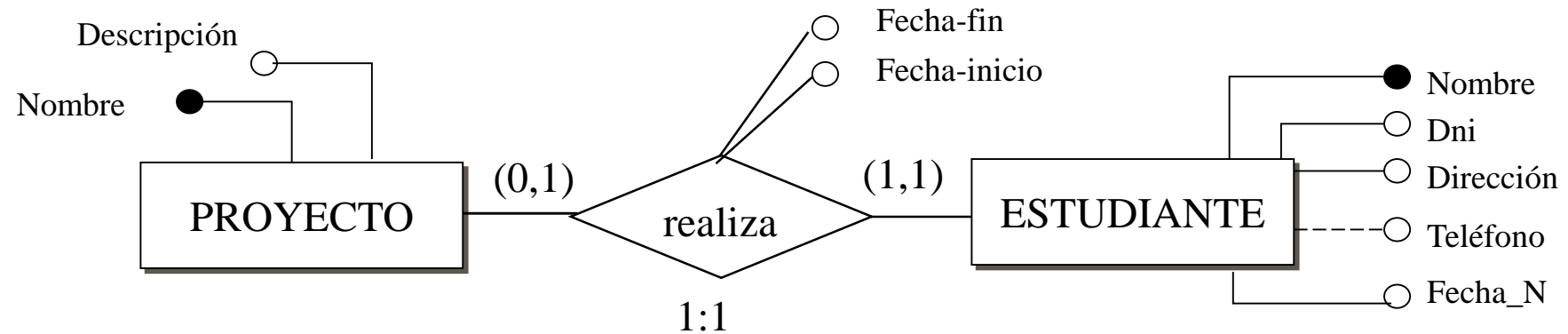
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:1



- Migrar la clave de la entidad obligatoria a la opcional, junto con los atributos de la interrelación.

Ejemplo:



PROYECTO (NOMBRE, DESCRIPCION, *ESTUDIANTE*, FECHA_I, FECHA_F)

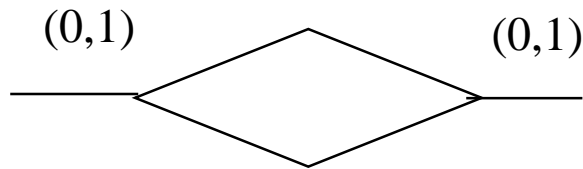
→ **ESTUDIANTE** (NOMBRE, DNI, DIRECCION, TELF* FECHA_NAC)

Los valores de ESTUDIANTE en PROYECTO no se pueden repetir, por lo que puede ser una clave alternativa.

PERO ahora no es obligatorio que tengan que estar todos los estudiantes en PROYECTO por lo que no hay que indicar nada al respecto

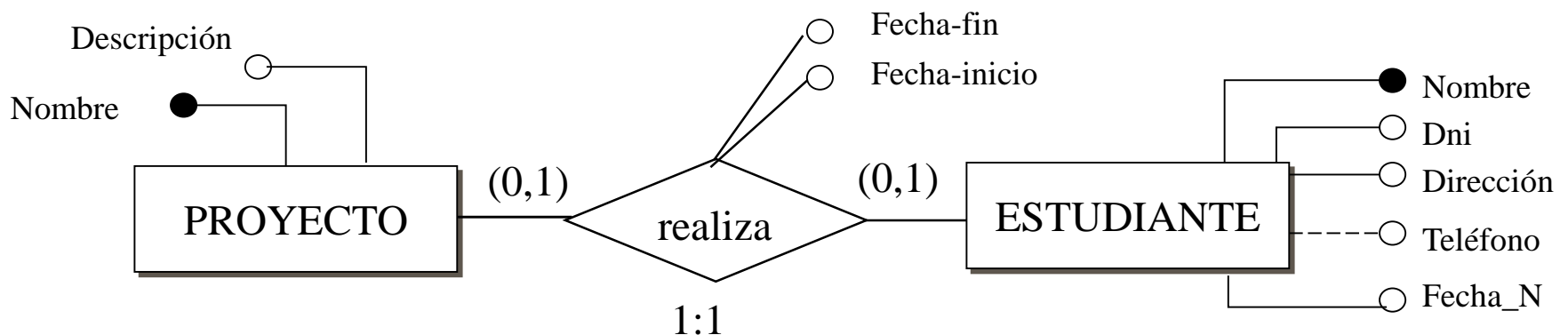
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:1



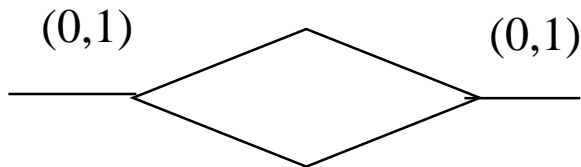
- a) Migrar la clave de una entidad a la otra, junto con los atributos de la interrelación
- b) Transformar la interrelación en una relación que contiene los atributos de la interrelación y las claves primarias de las dos entidades (una de ellas será la clave primaria)

Ejemplo:



Transformación diseño conceptual a lógico relacional

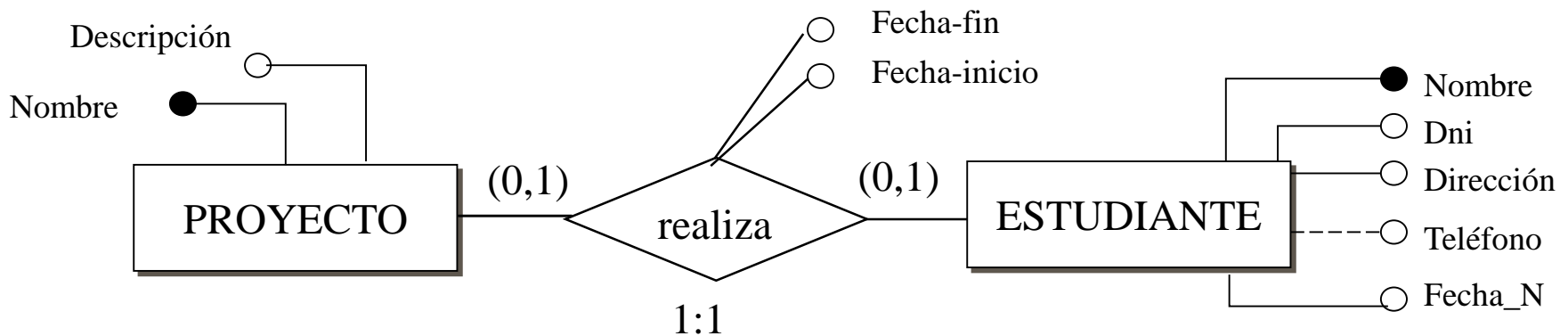
Interrelaciones binarias 1:1



a) Migrar la clave de una entidad a la otra, junto con los atributos de la interrelación

b) ...

Ejemplo:



a) O migramos la clave de ESTUDIANTE a PROYECTO y será opcional al igual que los atributos de la relación, o al revés

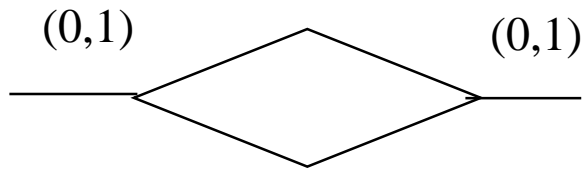
PROYECTO (NOMBRE, DESCRIPCION, *ESTUDIANTE**, FECHA_I*, FECHA_F*)

→ **ESTUDIANTE** (NOMBRE, DNI, DIRECCION, TELF* FECHA_NAC)

Los valores de ESTUDIANTE en PROYECTO no se pueden repetir, PERO al ser opcional no podría ser una clave alternativa y tampoco es obligatorio que tengan que estar todos los estudiantes en PROYECTO por lo que no hay que indicar nada al respecto

Transformación diseño conceptual a lógico relacional

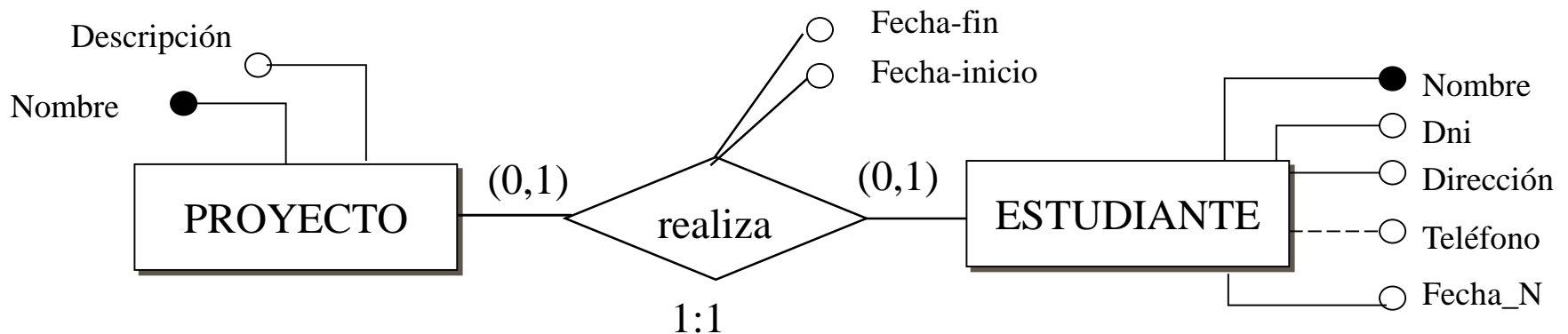
Interrelaciones binarias 1:1



a)

b) Transformar la interrelación en una relación que contiene los atributos de la interrelación y las claves primarias de las dos entidades (una de ellas será la clave primaria)

Ejemplo:



PROYECTO (NOMBRE, DESCRIPCION)

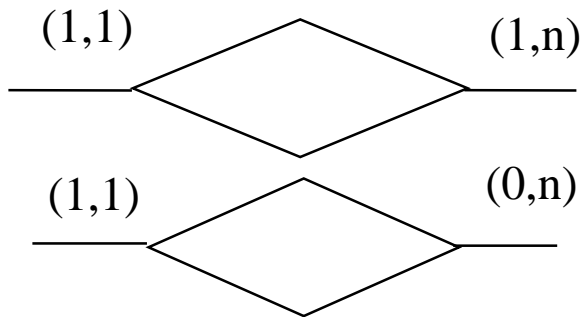
ESTUDIANTE (NOMBRE, DNI, DIRECCION, TELF* FECHA_NAC)

ESTUD-PROY (PROYECTO, *ESTUDIANTE*, FECHA_I, FECHA_F)

Si la clave primaria es el nombre de proyecto, el de estudiante será clave alternativa, o al revés

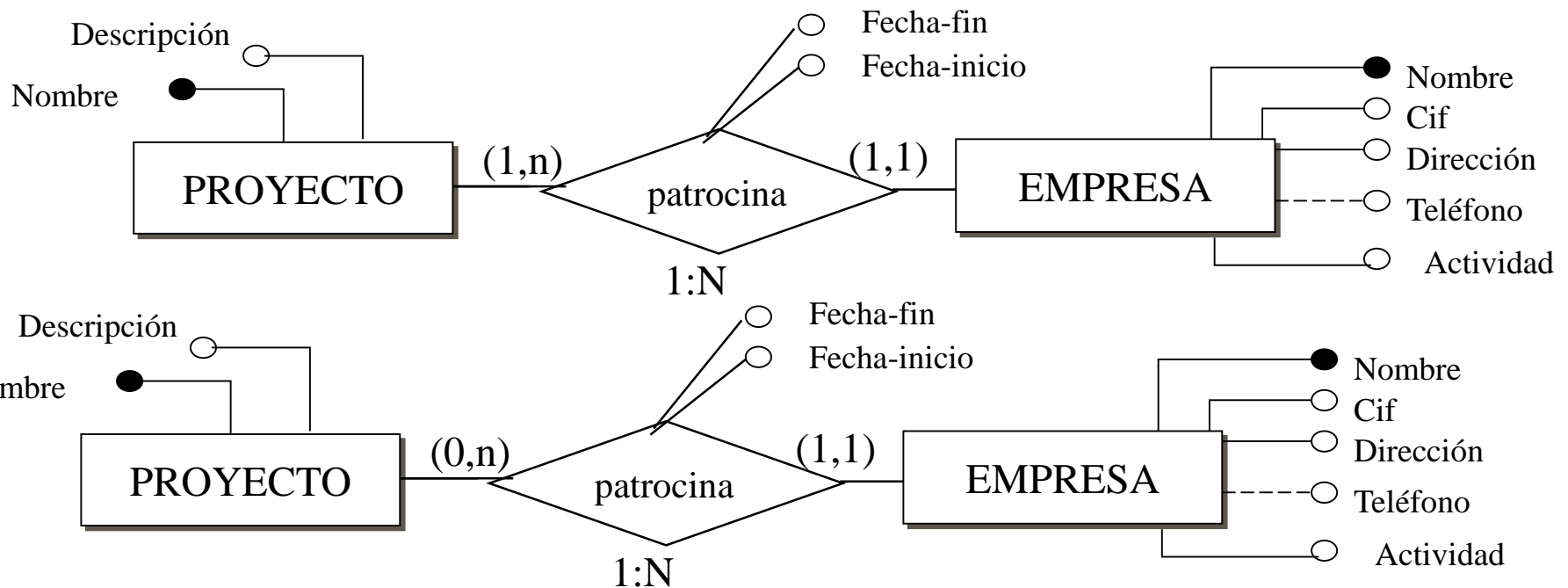
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:N



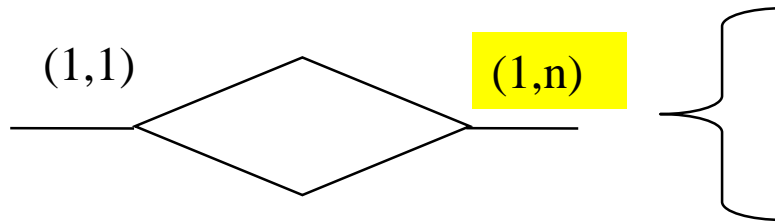
- Migrar la clave junto con los atributos de la interrelación

Ejemplo:



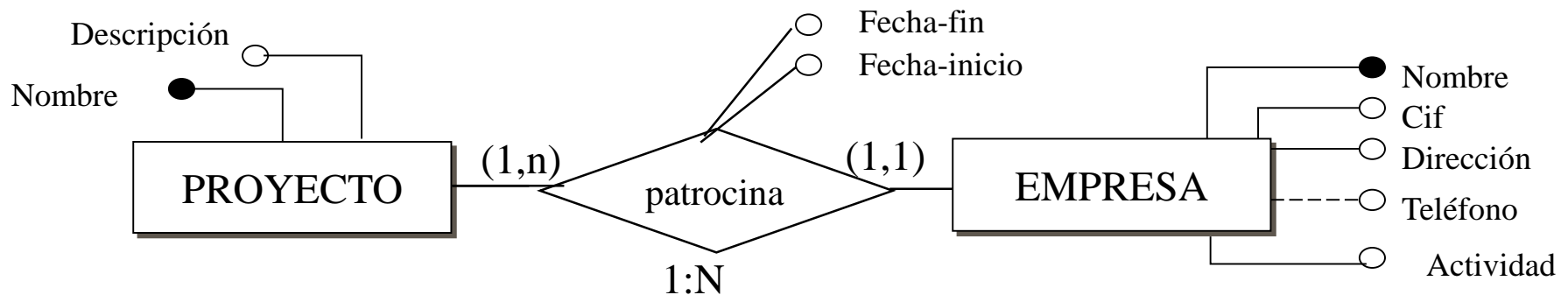
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:N



- Migrar la clave junto con los atributos de la interrelación

Ejemplo:



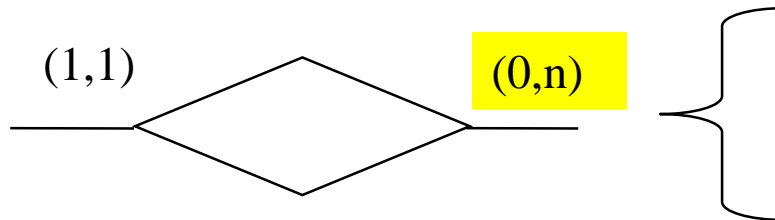
PROYECTO (NOMBRE, DESCRIPCION, *EMPRESA*, FECHA_I, FECHA_F)

EMPRESA (NOMBRE, CIF, DIRECCION, TELF*, ACTIVIDAD)

En la tabla PROYECTO deben aparecer todas las empresas, ya que todas tienen como mínimo 1 proyecto.

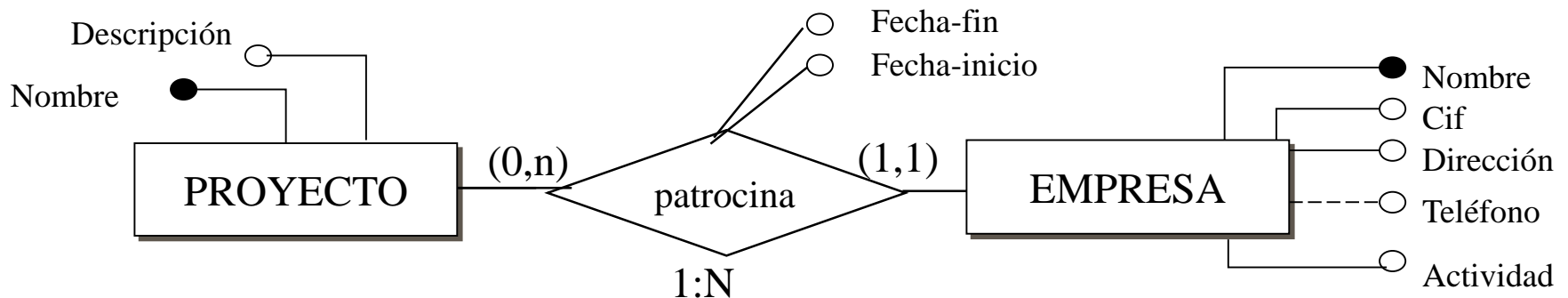
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:N



- Migrar la clave junto con los atributos de la interrelación

Ejemplo:



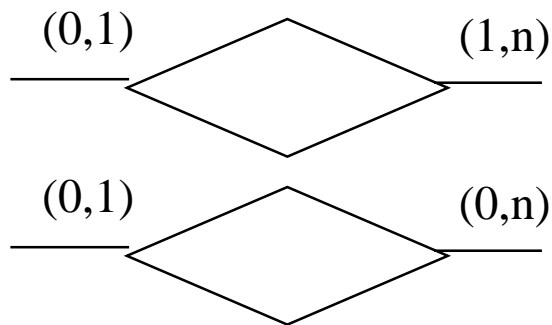
PROYECTO (NOMBRE, DESCRIPCION, *EMPRESA*, FECHA_I, FECHA_F)

→ **EMPRESA** (NOMBRE, CIF, DIRECCION, TELF*, ACTIVIDAD)

Ahora en la tabla PROYECTO no tienen por qué aparecer todas las empresas, ya que tienen como mínimo 0 proyecto, por tanto no hace falta decir nada al respecto.

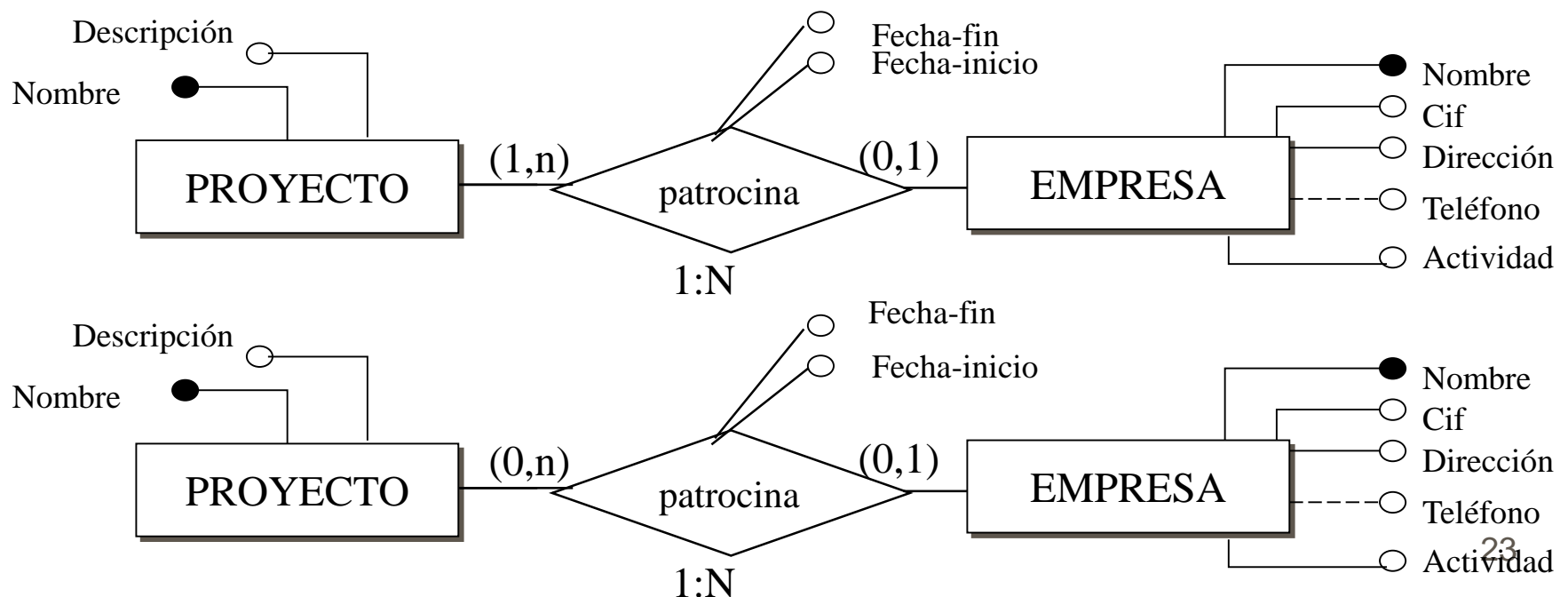
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:N



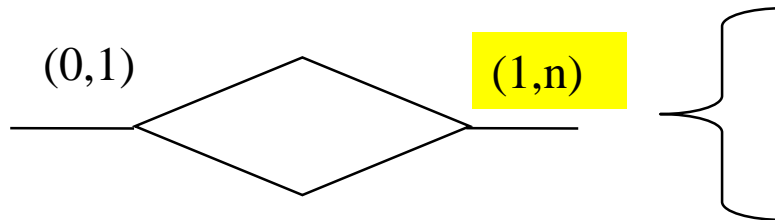
- Migrar la clave junto con los atributos de la interrelación
- La interrelación se transforma en una relación que contiene los atributos de la interrelación y las claves de las dos entidades

Ejemplo:



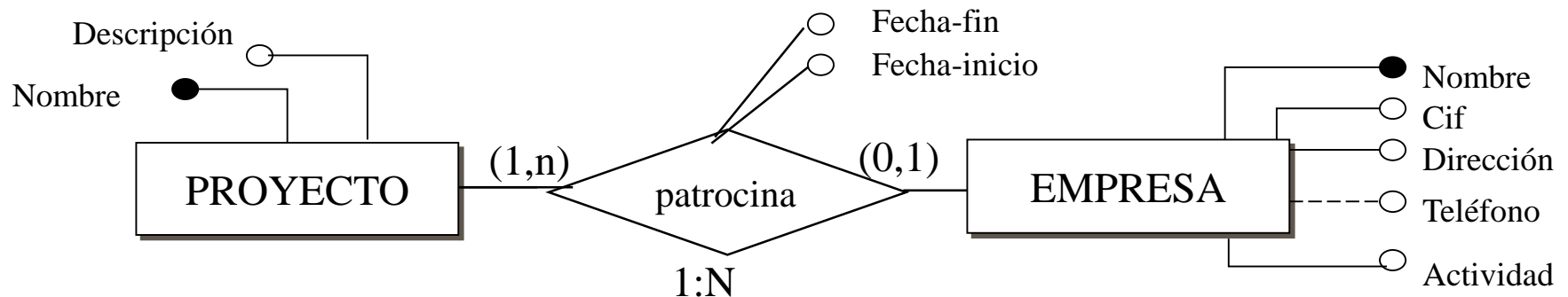
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:N



- Migrar la clave junto con los atributos de la interrelación **y serán opcionales por (0,1)**

Ejemplo:



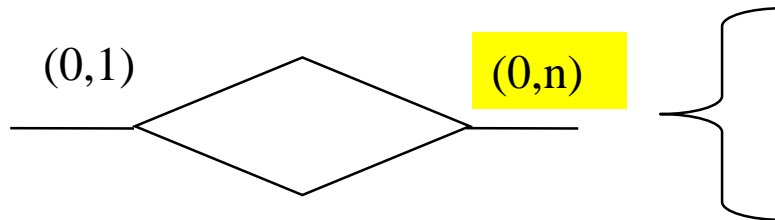
PROYECTO (NOMBRE, DESCRIPCION, *EMPRESA**, *FECHA_I**, *FECHA_F**)

EMPRESA (NOMBRE, CIF, DIRECCION, *TELF**, ACTIVIDAD)

En la tabla PROYECTO deben aparecer todas las empresas, ya que todas tienen como mínimo 1 proyecto.

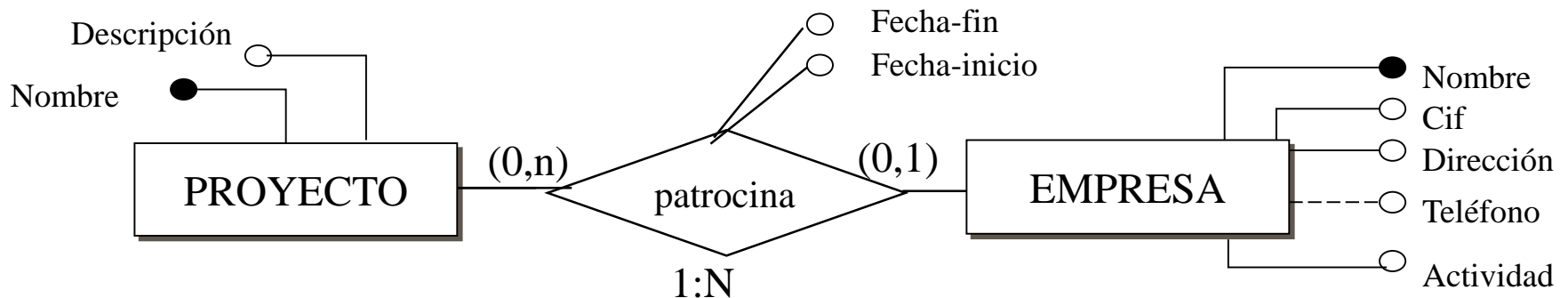
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:N



- Migrar la clave junto con los atributos de la interrelación y serán opcionales por (0,1)

Ejemplo:



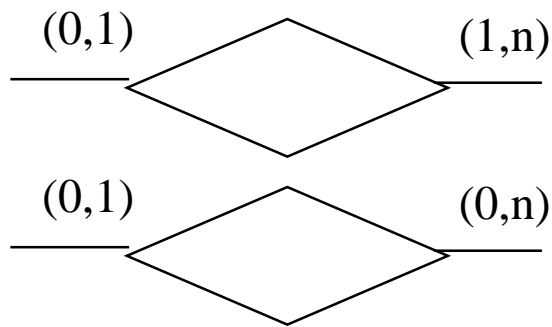
PROYECTO (NOMBRE, DESCRIPCION, *EMPRESA**, FECHA_I*, FECHA_F*)

→ **EMPRESA** (NOMBRE, CIF, DIRECCION, TELF*, ACTIVIDAD)

Ahora en la tabla PROYECTO no tienen por qué aparecer todas las empresas, ya que tienen como mínimo 0 proyecto, por tanto no hace falta decir nada al respecto.

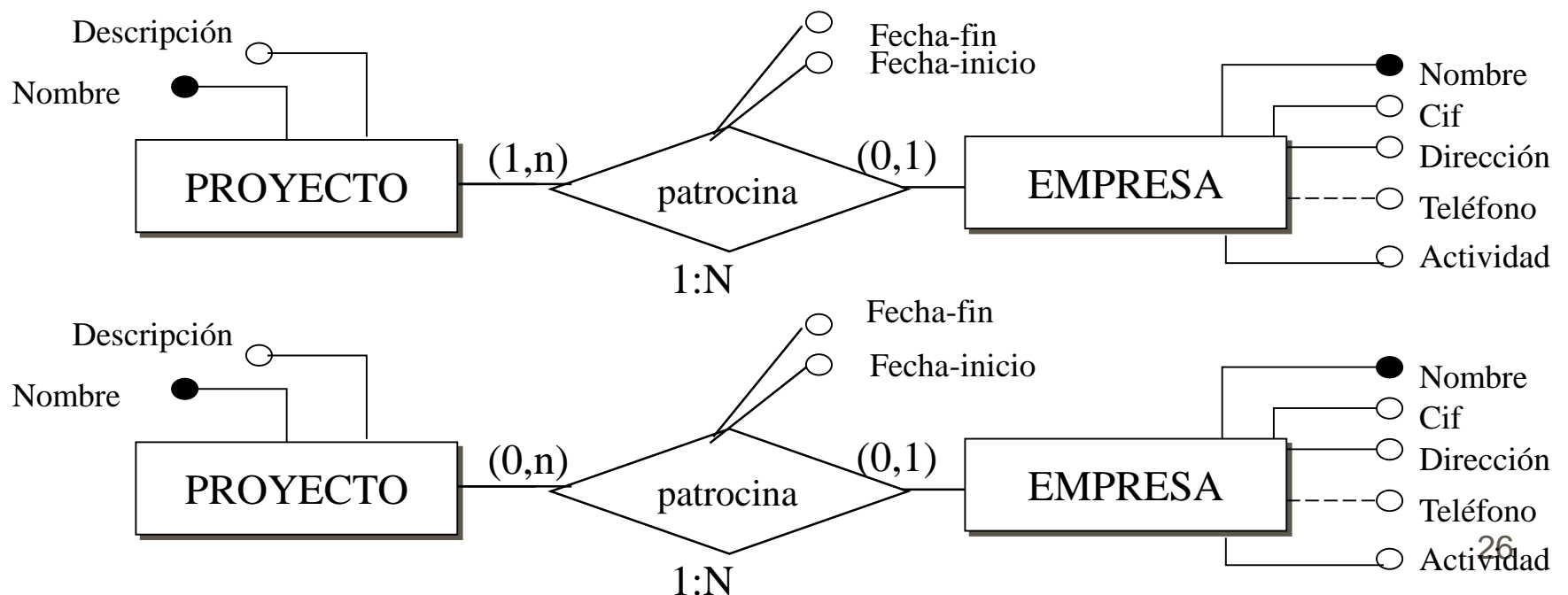
Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:N



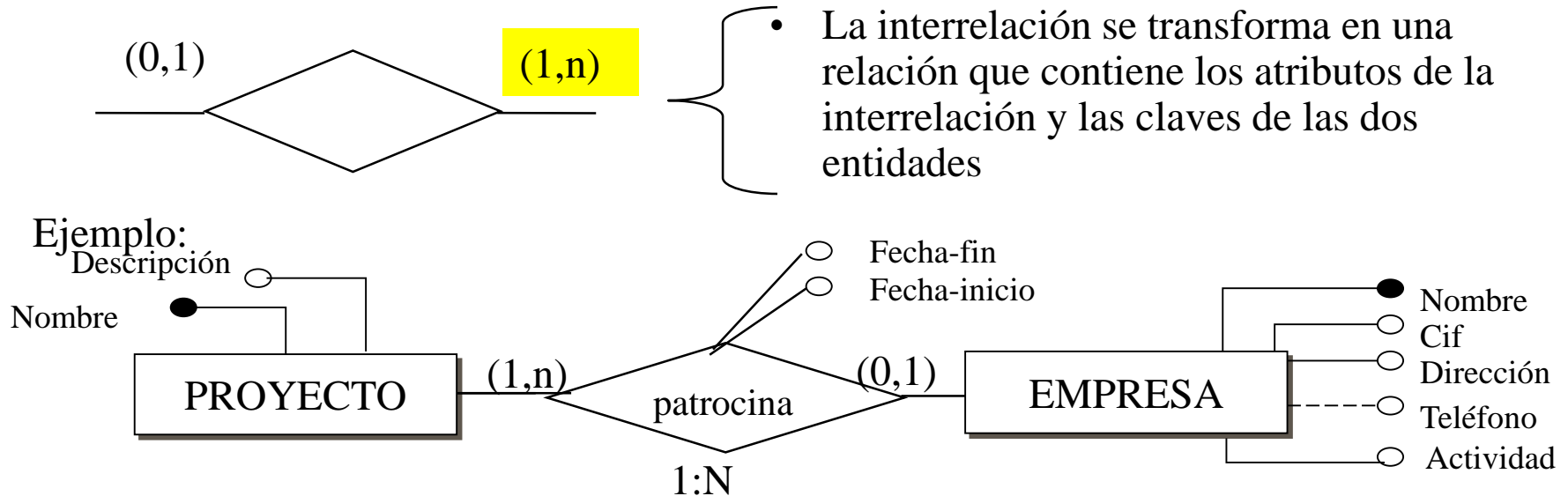
- La interrelación se transforma en una relación que contiene los atributos de la interrelación y las claves de las dos entidades

Ejemplo:



Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:N



→ **PROYECTO** (NOMBRE, DESCRIPCION)

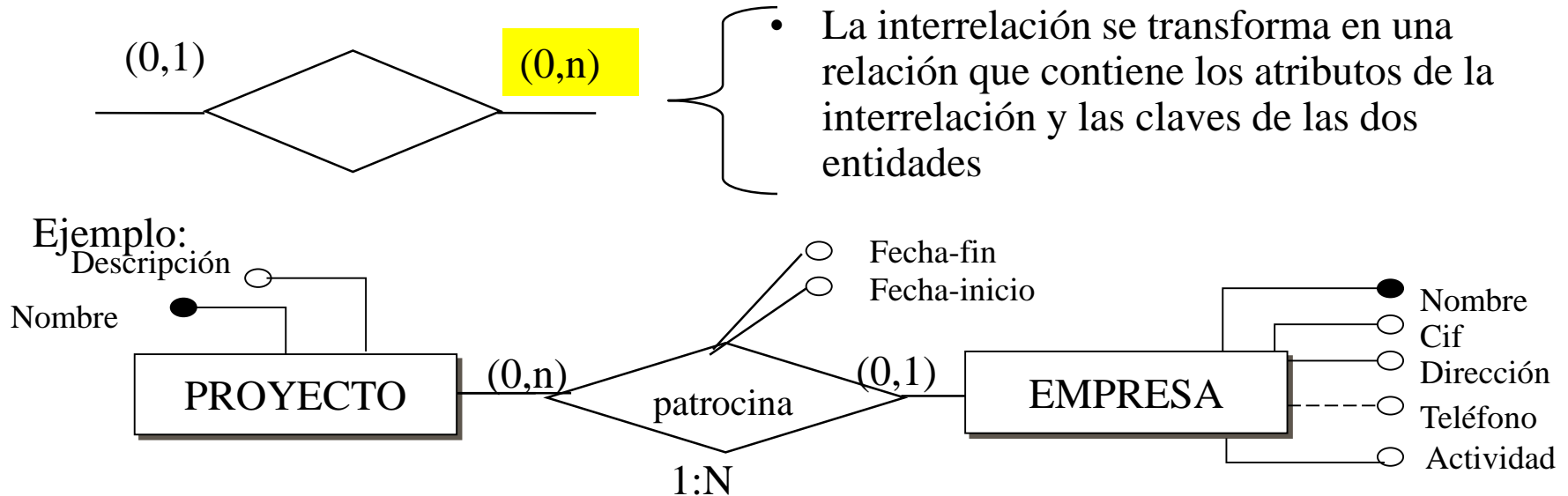
→ **EMPRESA** (NOMBRE, CIF , DIRECCION, TELF* , ACTIVIDAD)

PATROCINA (NOMPROY, NOMEMPR, FECHA_I, FECHA_F)

En la tabla PATROCINA deben aparecer todas las empresas, ya que todas tienen como mínimo 1 proyecto (Los proyectos no tienen que aparecer todos porque su mínimo es 0, por lo que no es necesario indicarlo)

Transformación diseño conceptual a lógico relacional

Interrelaciones binarias 1:N



→ **PROYECTO** (NOMBRE, DESCRIPCION)

→ **EMPRESA** (NOMBRE, CIF , DIRECCION, TELF* , ACTIVIDAD)

PATROCINA (NOMPROY, NOMEMPR, FECHA_I, FECHA_F)

No es necesario indicar nada más puesto que ni Proyecto ni Empresa tienen que estar todas en la tabla PATROCINA puesto que las dos tienen participación de mínimo 0

Transformación diseño conceptual a lógico relacional

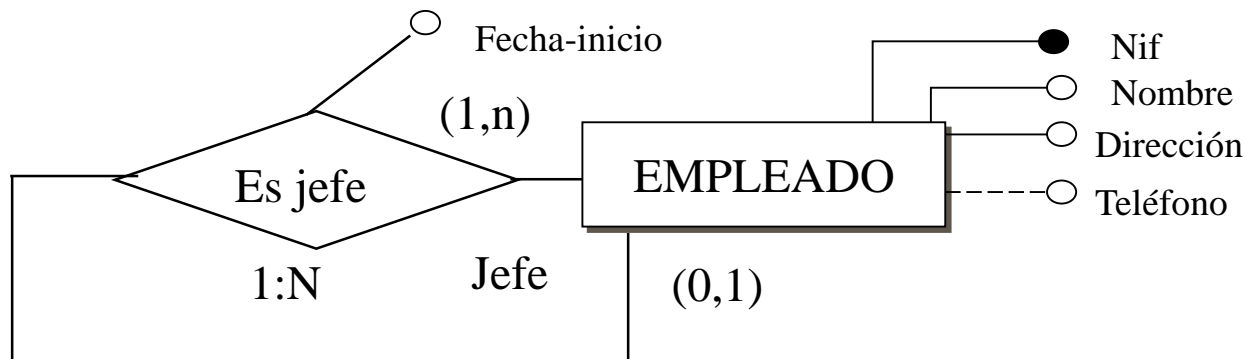
Interrelaciones **unarias** 1:N (igual que si son binarias 1:N)

- Migrar la clave junto con los atributos de la interrelación
- La interrelación se transforma en una relación que contiene los atributos de la interrelación y dos veces la clave de la entidad (juegan un rol distinto). Una de ellas será la clave primaria de la relación



No conviene si la interrelación tiene atributos o si se generan muchos valores nulos por ser opcional la relación

Ejemplo:

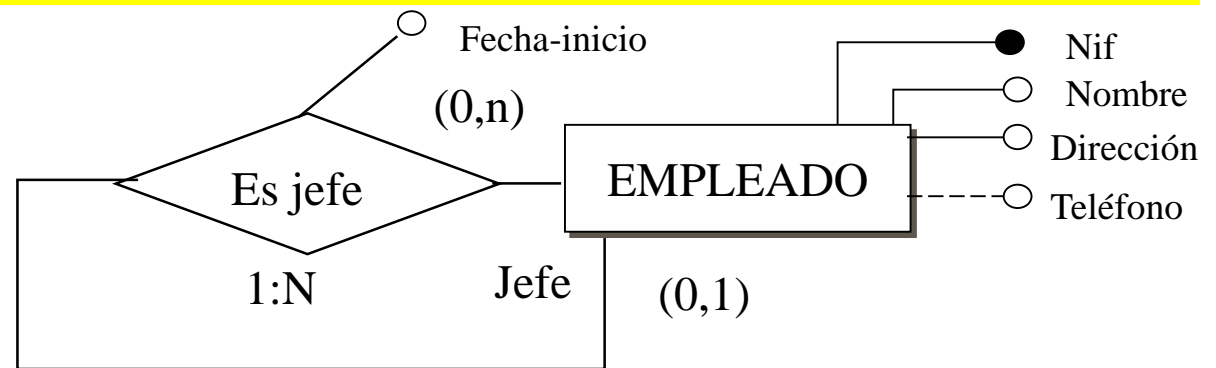


Transformación diseño conceptual a lógico relacional

Interrelaciones **unarias** 1:N (igual que si son binarias 1:N)

- Migrar la clave junto con los atributos de la interrelación

Ejemplo:



EMPLEADO (NIF, NOMBRE, DIRECCION, TELF*, *MIJEFE**, FECHA_I*)

- La interrelación se transforma en una relación que contiene los atributos de la interrelación y dos veces la clave de la entidad (juegan un rol distinto). Una de ellas será la clave primaria de la relación (la que participa con N)

EMPLEADO (NIF, NOMBRE, DIRECCION, TELF)

EMP-JEFE (NIF-EMP, *MIJEFE*, FECHA_I)

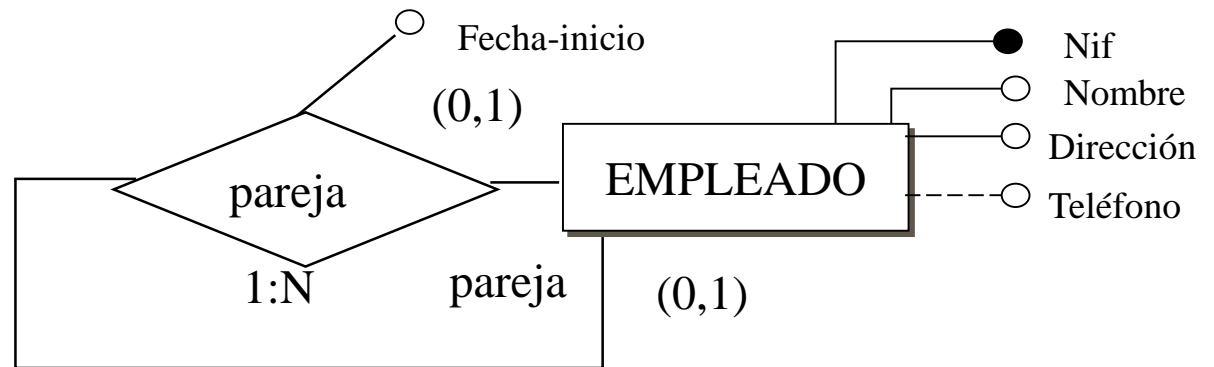
Ahora MIJEFE no es opcional porque en esta tabla solo estarán los empleados que tienen jefe.

Transformación diseño conceptual a lógico relacional

Interrelaciones **unarias** 1:1 (igual que si son binarias 1:1)

- Migrar la clave junto con los atributos de la interrelación

Ejemplo:



→ **EMPLEADO** (NIF, NOMBRE, DIRECCION, TELF*, MIPAREJA*, FECHA_I*)

MIPAREJA es el NIF del empleado que es pareja y es OPCIONAL

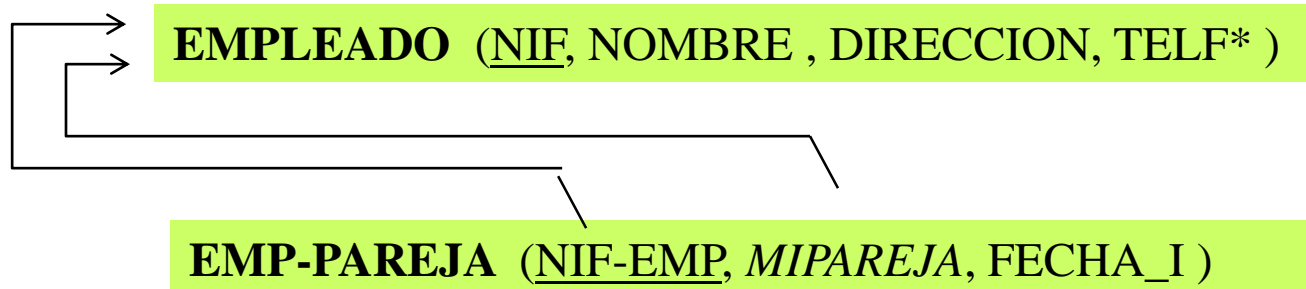
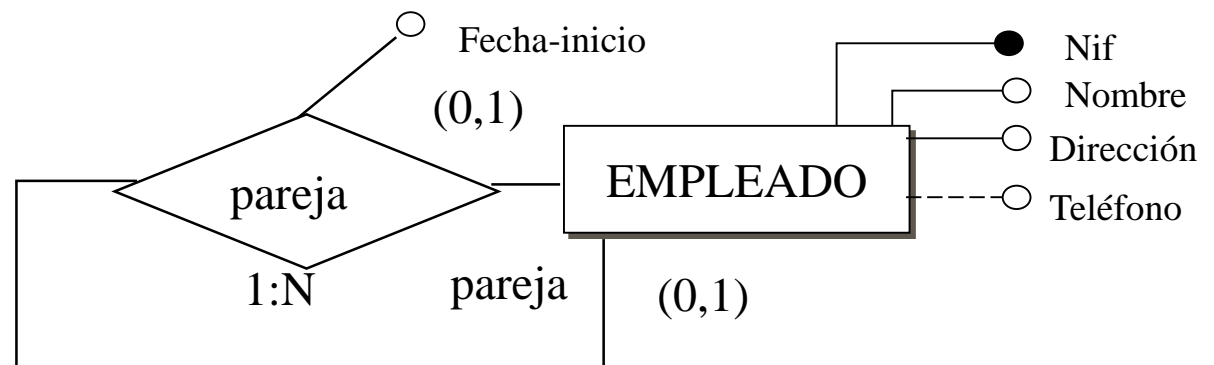
Hay que anotar que el NIF de MIPAREJA no se puede repetir, PERO no puede ser Clave alternativa al ser opcional

Transformación diseño conceptual a lógico relacional

Interrelaciones **unarias** 1:1 (igual que si son binarias 1:1)

- La interrelación se transforma en una relación que contiene los atributos de la interrelación y dos veces la clave de la entidad (juegan un rol distinto). Una de ellas será la clave primaria de la relación (la que participa con N)

Ejemplo:



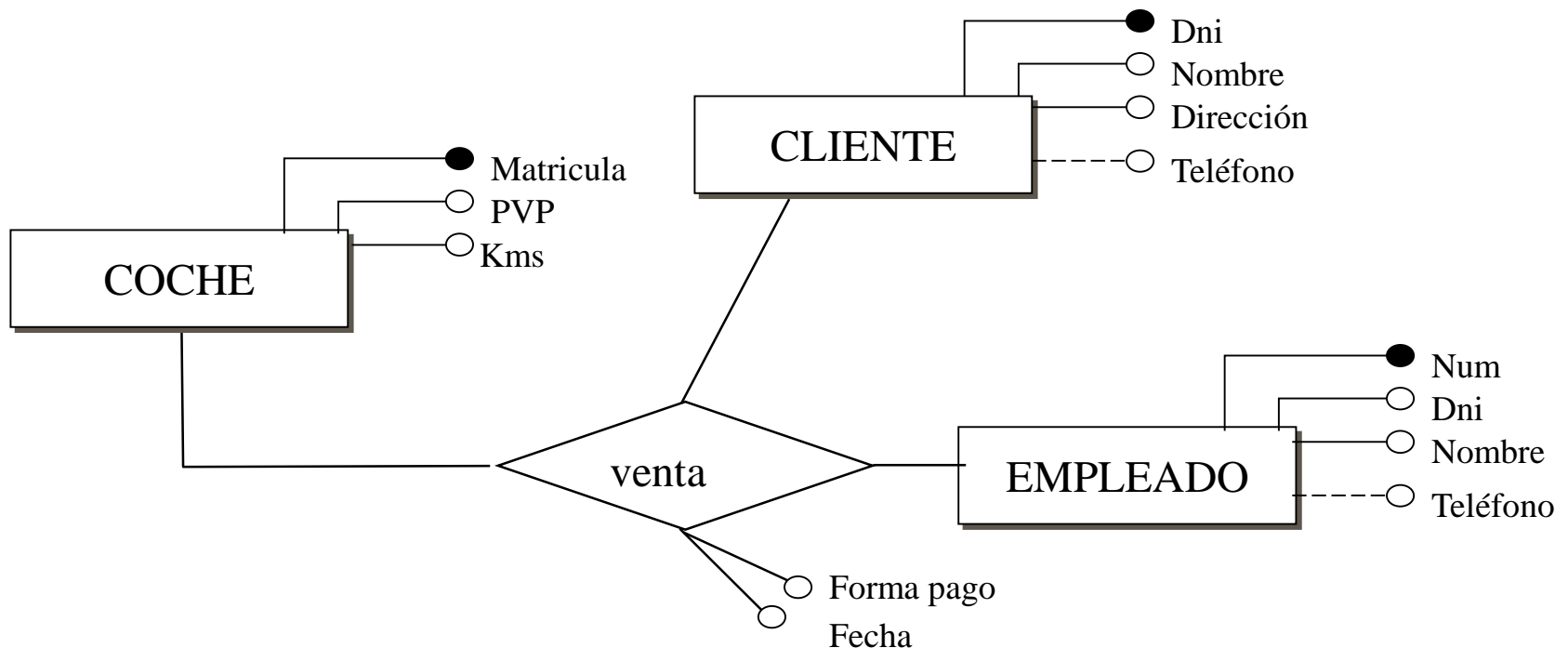
Ahora MIPAREJA no es opcional porque en esta tabla solo estarán los empleados que tienen otro como pareja. Los valores de MIPAREJA no se pueden repetir, por lo que puede ser Clave alternativa en este caso

Transformación diseño conceptual a lógico relacional

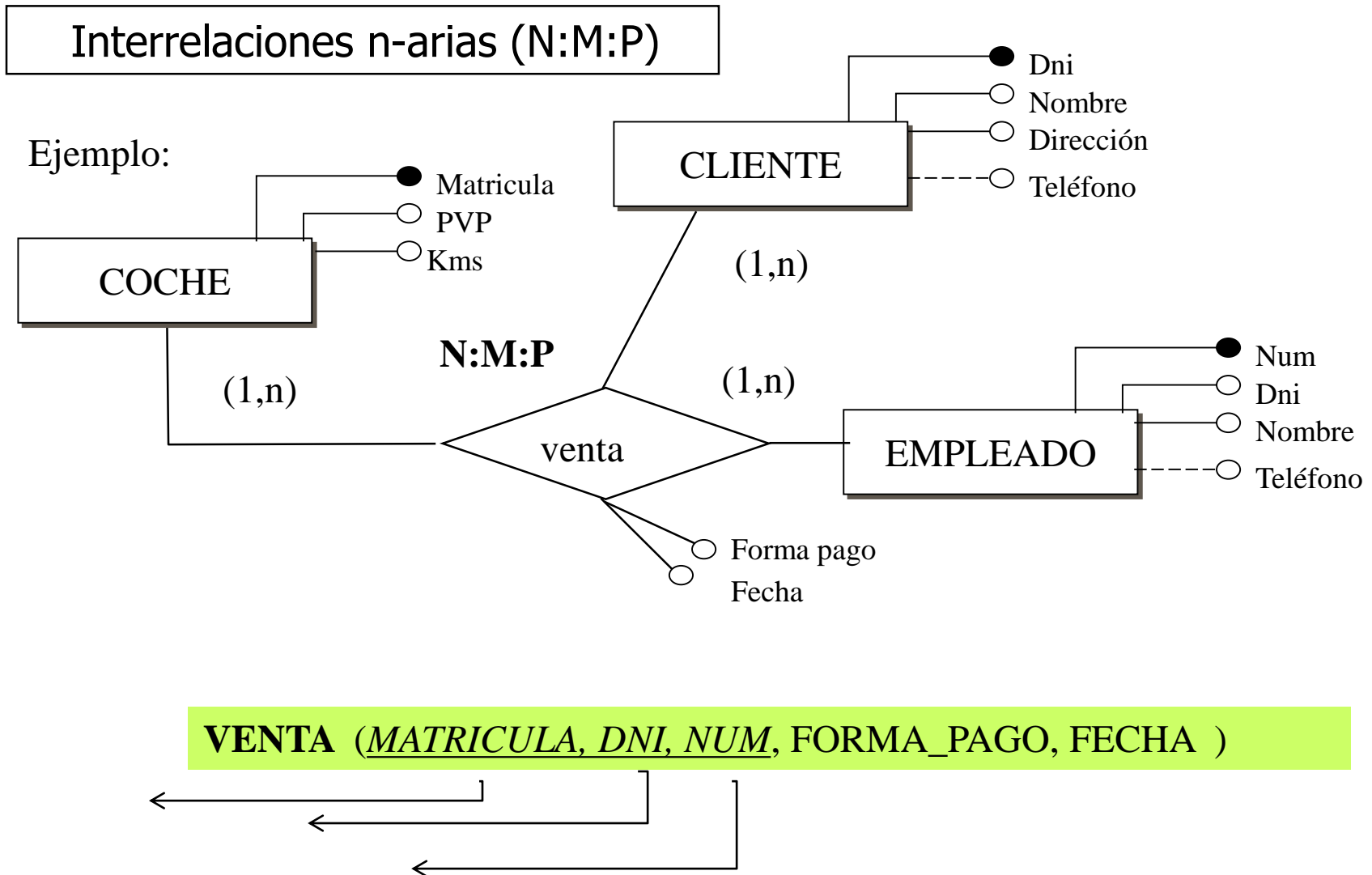
Interrelaciones n-arias

- La interrelación se transforma en una relación que contiene los atributos de la interrelación y las claves primarias de todas las entidades participantes. Generalmente, la clave primaria estará compuesta por las claves primarias de las entidades participantes (todas o no según las cardinalidades de la relación).

Ejemplo:

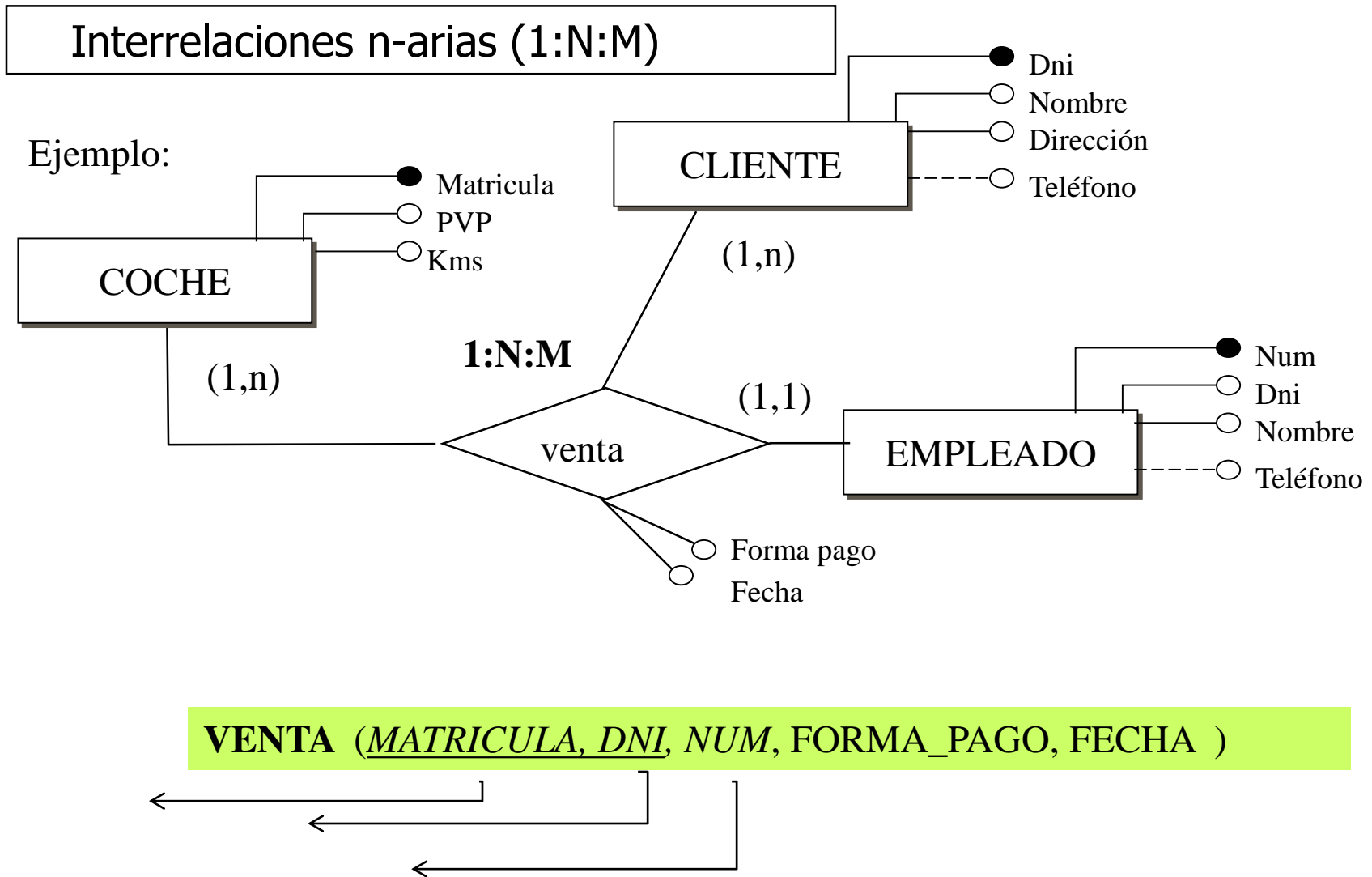


Transformación diseño conceptual a lógico relacional



La clave primaria está compuesta por las tres claves migradas y cada una es una clave ajena que apunta a la tabla de la que viene (COCHE, CLIENTE o EMPLEADO)

Transformación diseño conceptual a lógico relacional



Si por cada par de COCHE y CLIENTE solo hay un empleado relacionado, la clave primaria está compuesta solo por las claves de COCHE y CLIENTE.

Transformación diseño conceptual a lógico relacional

❑ TRANSFORMACIONES PREVIAS:

- ❑ También es necesario transformar elementos del modelo E-R no representables en el modelo conceptual relacional, como son:
- ❑ Algunas de estas transformaciones deben hacerse antes de transformar las relaciones, porque estos cambios pueden implicar cambios en las relaciones existentes en el MER
 - ❑ Atributos compuestos
 - ❑ Dependencias en existencia y en identificación
 - ❑ Jerarquías de generalización/especialización
 - ❑ Atributos multivaluados

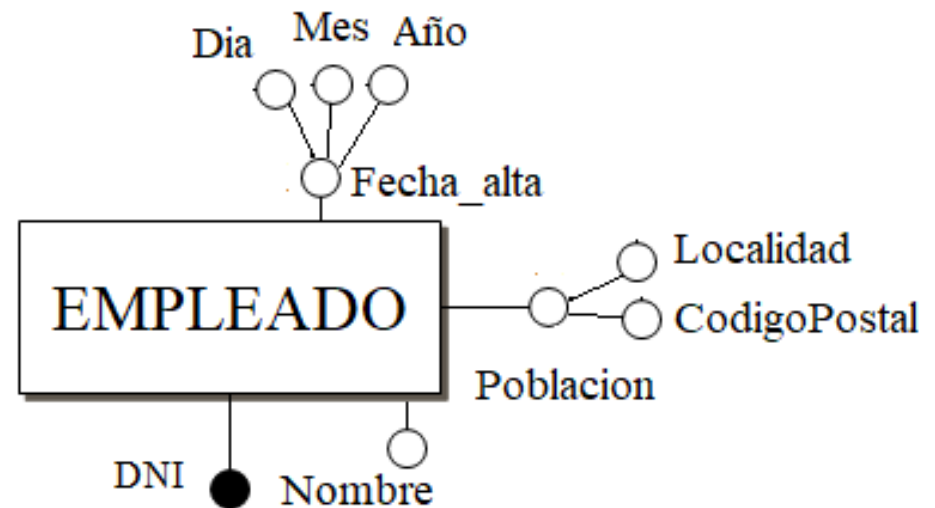
Transformación diseño conceptual a lógico relacional

Atributos compuestos

Dos opciones:

- 1.- Considerar el atributo compuesto como simple. (Si se quiere identificar cada parte del atributo se tiene que hacer por programa)
- 2.- Considerar cada componente del atributo compuesto como un atributo simple

Ejemplo



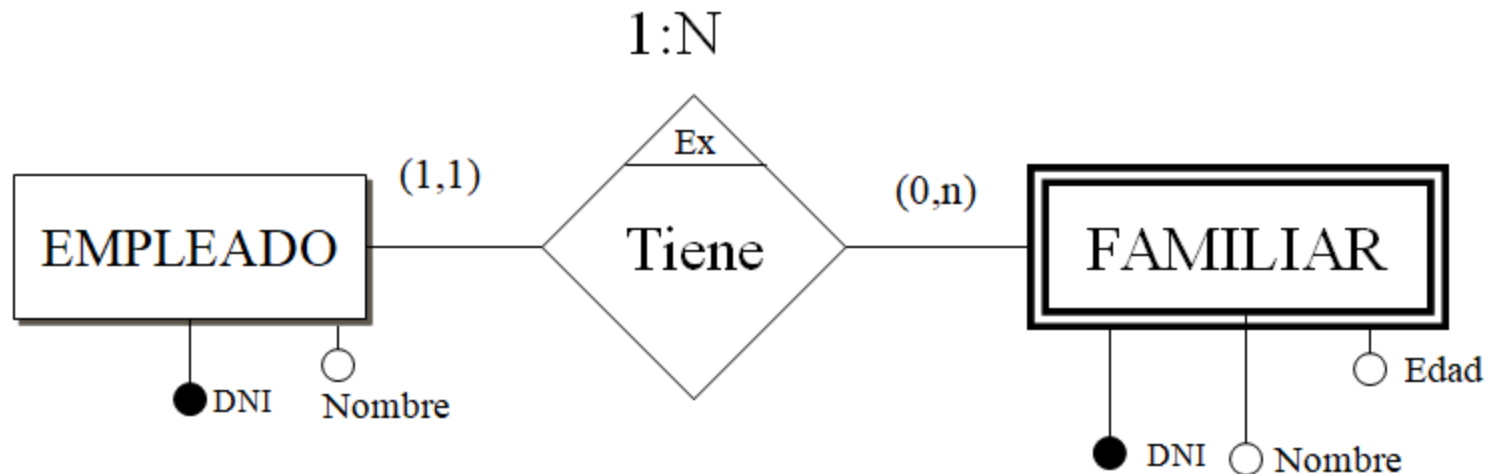
EMPLEADO (DNI, NOMBRE, FECHA_ALTA, LOCALIDAD, CODIGOPOSTAL)

Transformación diseño conceptual a lógico relacional

Dependencia en existencia y en identificación

1. Dependencia en existencia:

Se transforma como una relación 1:N, teniendo en cuenta que la clave ajena propagada de la entidad fuerte a la entidad débil tiene que tener la opción de borrado en cascada



→ EMPLEADO (DNI, NOMBRE)

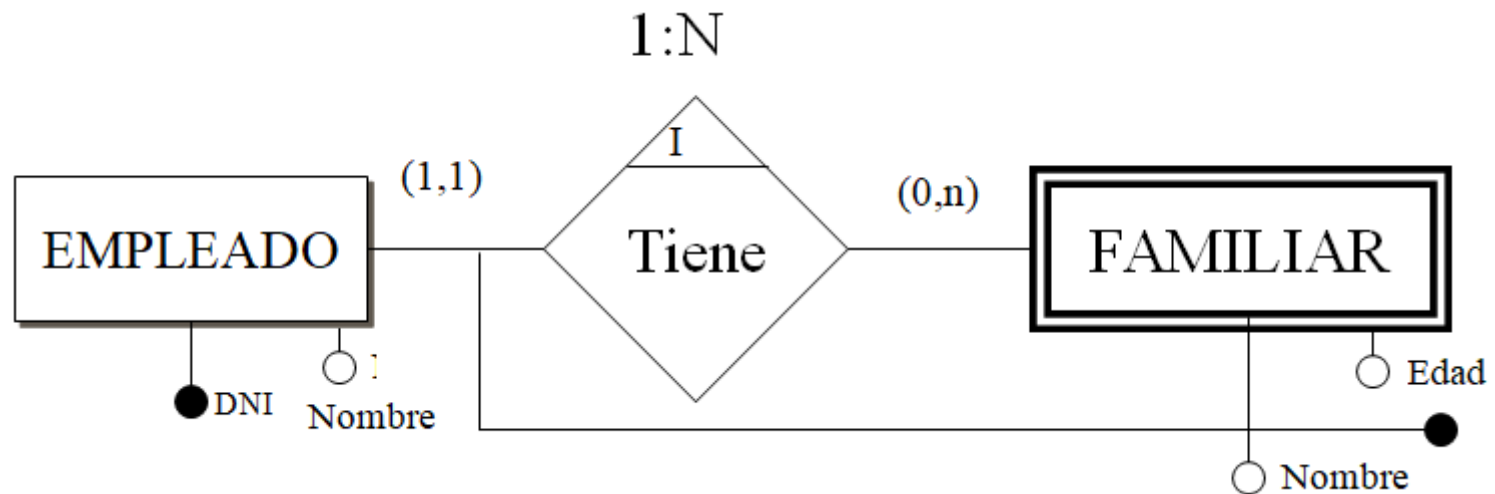
FAMILIAR (DNI, NOMBRE, *DNI-EMPLEADO*, EDAD)

Transformación diseño conceptual a lógico relacional

Dependencia en existencia y en identificación

2. Dependencia en identificación:

Además, la clave ajena propagada de la entidad fuerte a la entidad débil forma parte de su clave primaria.



→ EMPLEADO (DNI, NOMBRE)

FAMILIAR (DNI-EMPLEADO, NOMBRE, EDAD)

Transformación diseño conceptual a lógico relacional

Jerarquías de generalización/especialización

Tres opciones:

- a) Englobar todas las entidades (supertipo y subtipos) en una
- b) Crear una entidad para cada una
- c) Crear una entidad por cada subtipo

Transformación diseño conceptual a lógico relacional

Jerarquías de generalización/especialización

Las restricciones semánticas de totalidad/parcialidad y exclusividad/solapamiento hay que anotarlas al pasar al modelo relacional, para poder controlarlas.

Totalidad:

Un disparador para forzar una inserción en un subtipo si se inserta una tupla en el supertipo. El atributo discriminante tampoco puede tomar valor nulo

Parcialidad:

El atributo discriminante puede tomar valor nulo

Exclusividad:

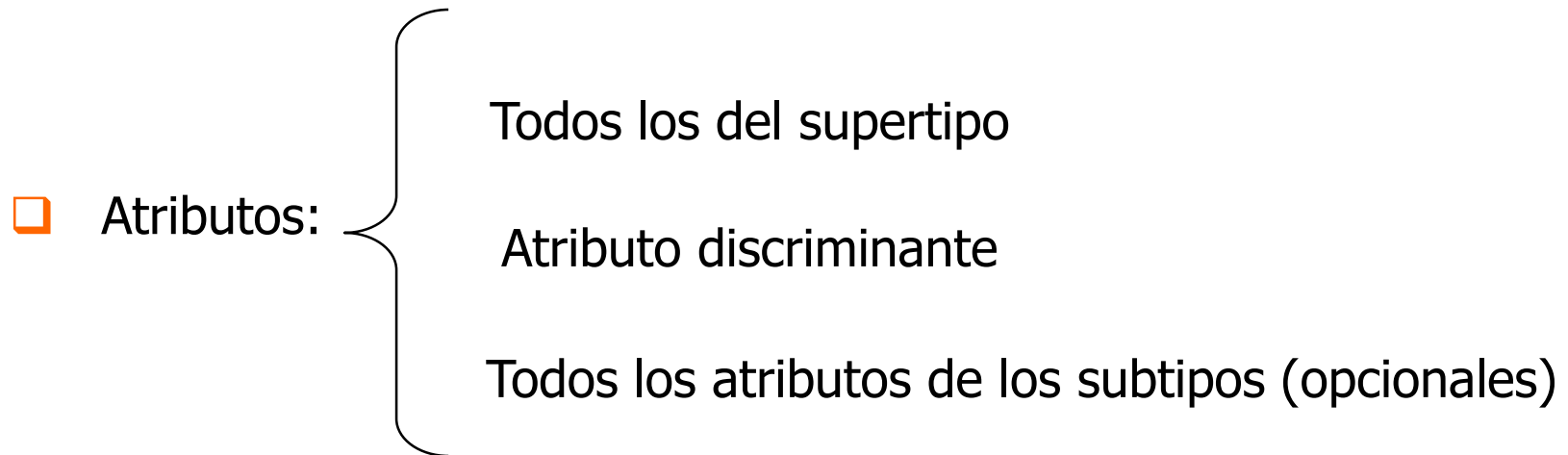
Una aserción que compruebe que si un ejemplar pertenece a uno de los subtipos, no puede pertenecer a los demás

Solapamiento:

El discriminante debe permitir nuevos valores que representen el solapamiento y con una aserción comprobar que los ejemplares están en los subtipos adecuados

Transformación diseño conceptual a lógico relacional

a) Englobar todas las entidades (supertipo y subtipos) en una

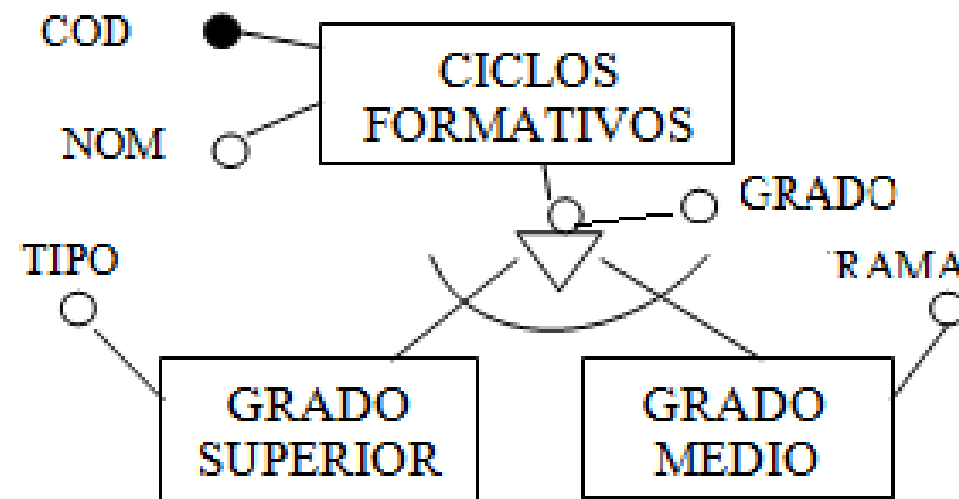


- Conviene cuando los subtipos se diferencian en pocos atributos y las interrelaciones con el resto de entidades son las mismas para todos los subtipos
- Genera valores nulos (en los atributos opcionales)
- Ralentiza el proceso de búsqueda (hay que tener en cuenta todas las tuplas en vez de las de una sola subentidad)
- Restricciones semánticas (verificaciones) para representar exclusividad,...

Transformación diseño conceptual a lógico relacional

a) Englobar todas las entidades (supertipo y subtipos) en una

Ejemplo:



CF (**COD**, NOM, GRADO, TIPO*, RAMA*)

En este caso hay que aclarar que: en cada fila hay que llenar obligatoriamente uno de los dos atributos, o tipo o rama (según el valor de atributo discriminante GRADO sea Superior o Medio).

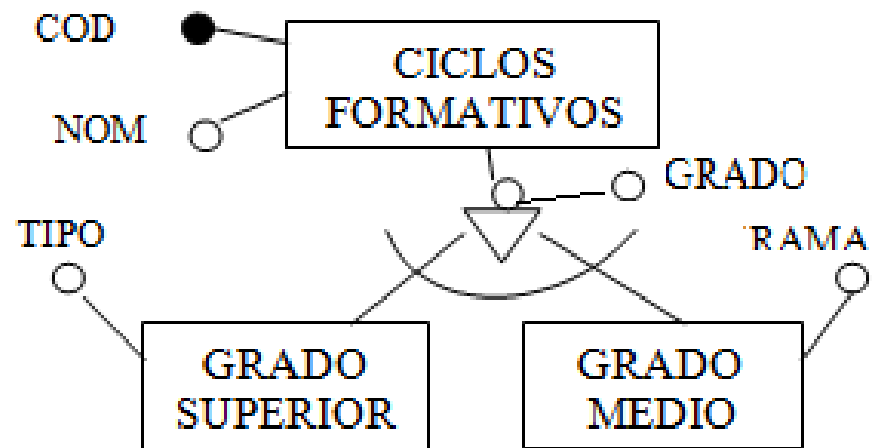
Transformación diseño conceptual a lógico relacional

- b) Crear una entidad para cada una
 - ❑ Entidad supertipo: atributos y clave de la supertipo
 - ❑ Entidades subtipo: atributos del subtipo y la clave primaria del supertipo (que será la clave primaria de la entidad)
 - ❑ Conviene cuando existen muchos atributos distintos entre los subtipos y se quieren mantener de todas maneras los atributos comunes a todos ellos en una relación
 - ❑ Restricciones semánticas oportunas

Transformación diseño conceptual a lógico relacional

b) Crear una entidad para cada una

Ejemplo:



→ CF (COD, NOM, GRADO)
 → CFGS(COD, TIPO)
 → CFGM(COD, RAMA)

En este caso hay que aclarar , para que se cumpla la totalidad y exclusividad, que: todos los ciclos de la tabla CF tienen que estar obligatoriamente en una (sólo en una) de las de los subtipos, o en CFGS o CFGM

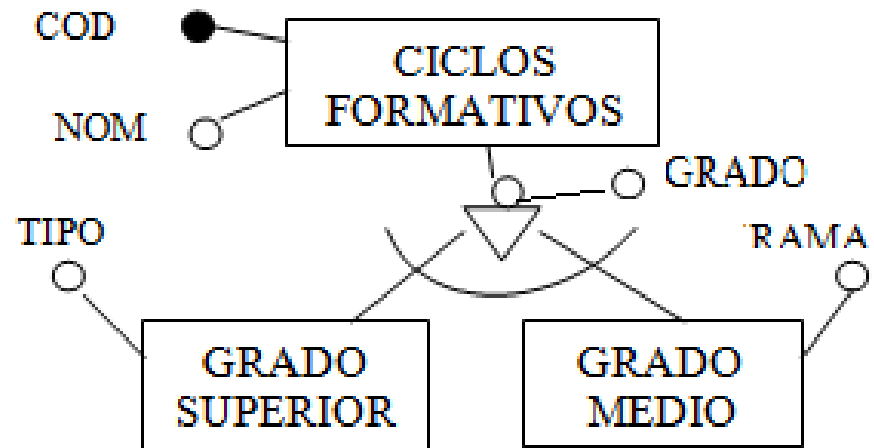
Transformación diseño conceptual a lógico relacional

- c) Crear una entidad por cada subtipo
- ☐ Contiene los atributos propios del subtipo más los atributos comunes (del supertipo). La clave primaria será la clave del supertipo
- ☐ Conviene cuando tienen muchos atributos distintos y los accesos realizados sobre los datos de los distintos subtipos afectan siempre a los comunes, y no hay interrelaciones entre el supertipo y otras entidades
- ☐ Inconvenientes:
 - ☐ Los interrelaciones con la entidad supertipo deben convertirse en interrelaciones con todas las subentidades
 - ☐ Los atributos de la entidad supertipo se repiten en cada subentidad
 - ☐ Los accesos a la entidad supertipo deben convertirse en accesos a todas las subentidades

Transformación diseño conceptual a lógico relacional

c) Crear una entidad por cada subtipo

Ejemplo:



CFGs (**COD**, NOM, TIPO)

CFGM (**COD**, NOM, RAMA)

En este caso la totalidad se cumple, solo hay dos grupos de CF, pero habría que aclarar la exclusividad: no puede existir en las dos tablas el mismo ciclo.

Transformación diseño conceptual a lógico relacional

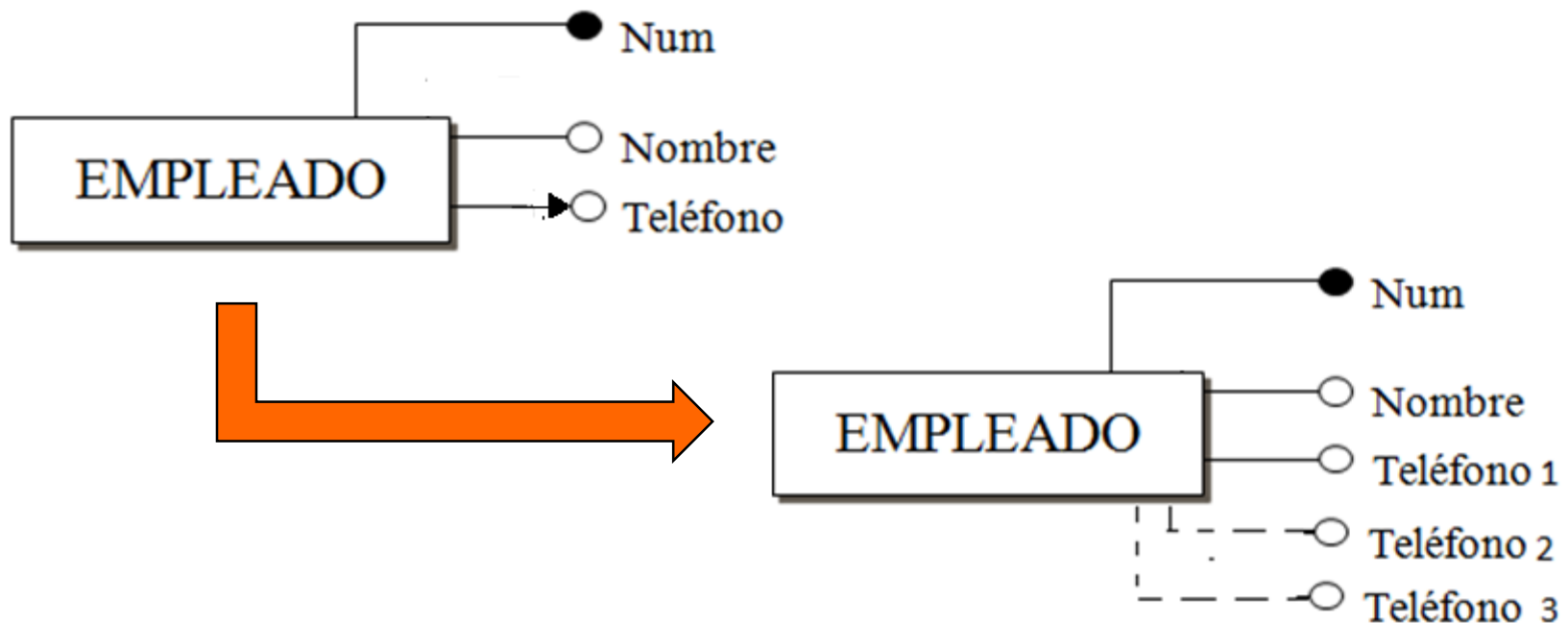
Atributos multivaluados en entidades

Opciones:

1.- Si el atributo se repite un número pequeño de veces.

Añadir un atributo por cada vez que se pueda repetir

Ejemplo: se guardarán hasta tres teléfonos



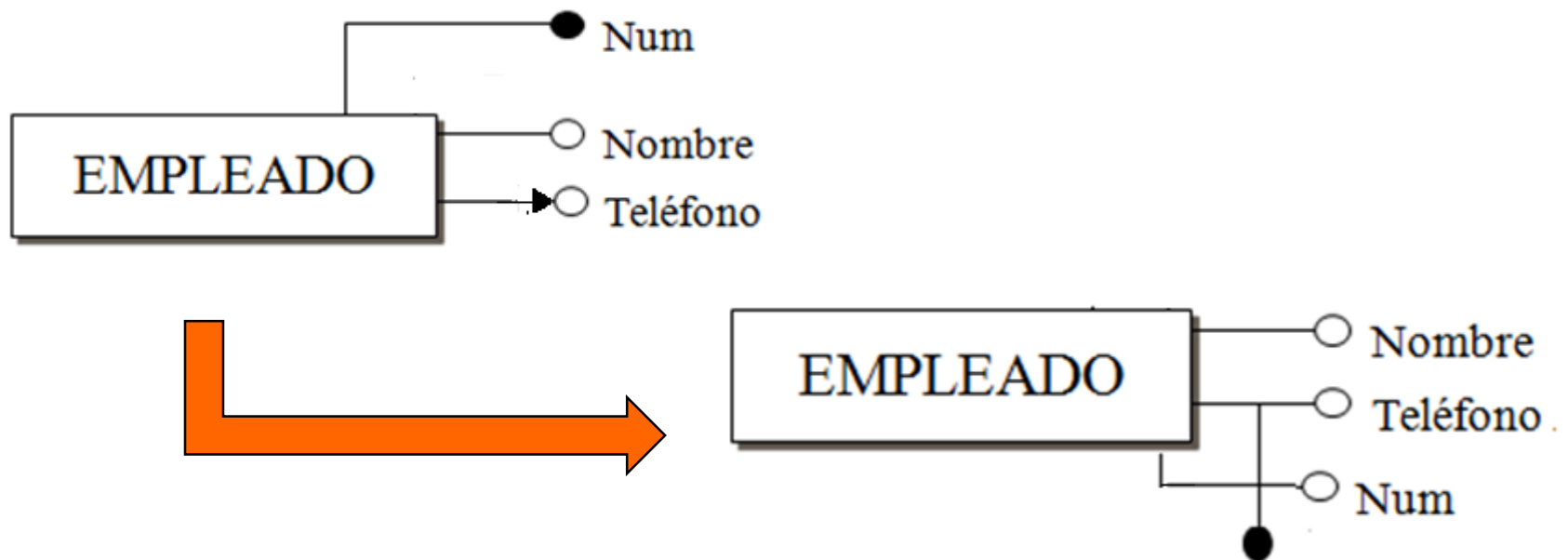
Transformación diseño conceptual a lógico relacional

Atributos multivaluados en entidades

Opciones:

2.- Si el atributo se repite un número indefinido de veces.

Añadir una ocurrencia para cada valor del atributo



El atributo ahora monovaluado forma parte de la clave primaria

No puede ser clave primaria NUM porque si el un empleado tiene 2 teléfonos aparecerá 2 veces en esta entidad (una vez con todos sus datos y el primer teléfono y otra vez con todos sus datos y el 2º teléfono)

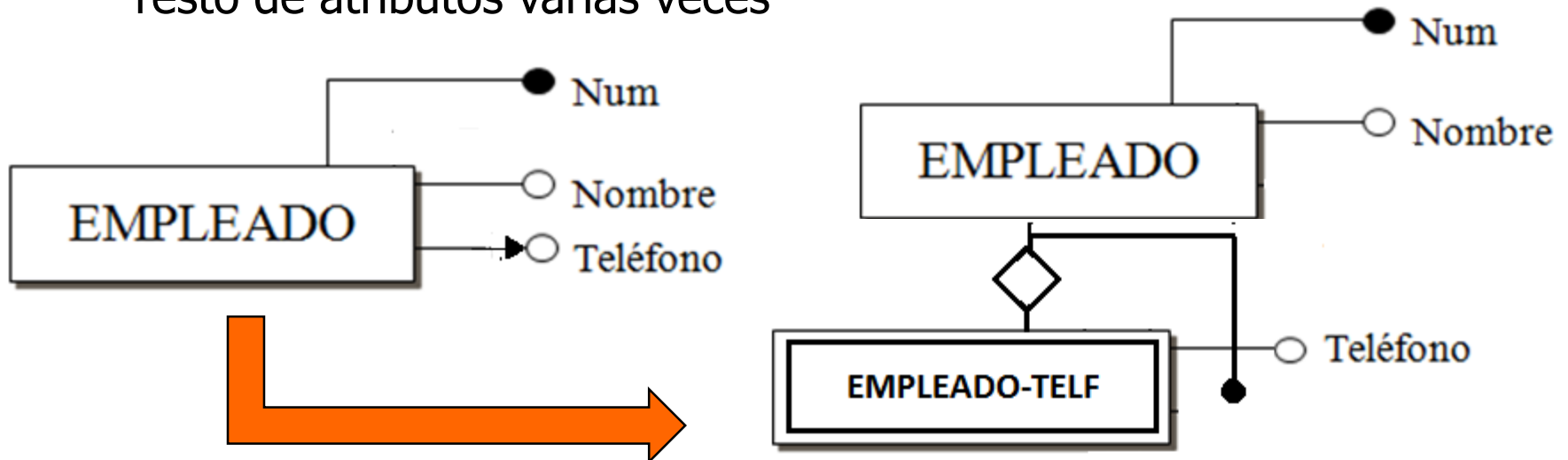
Transformación diseño conceptual a lógico relacional

Atributos multivaluados en entidades

Opciones:

3.- Si el atributo se repite un número indefinido de veces.

Crear una nueva entidad débil con la clave primaria y el atributo multivaluado (ahora monovaluado) pero así no hay que repetir el resto de atributos varias veces

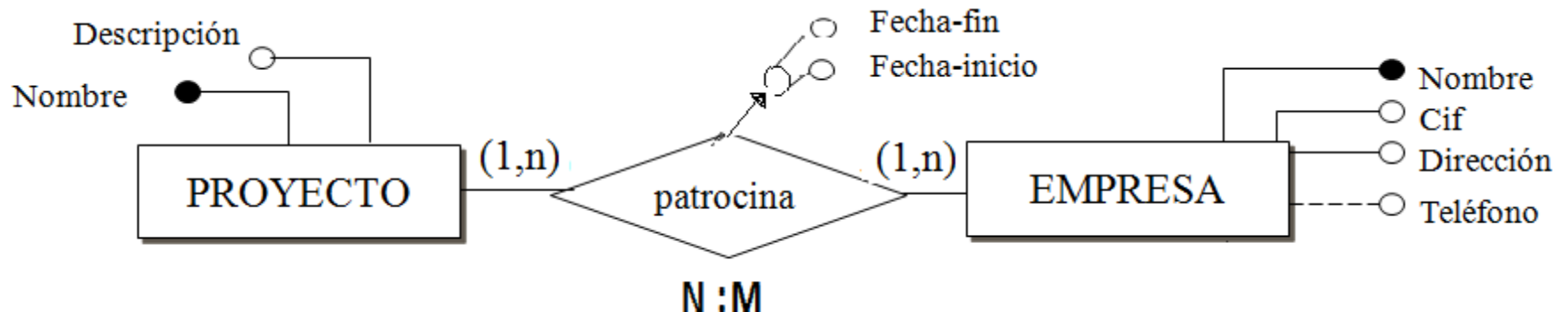


Transformación diseño conceptual a lógico relacional

Atributos multivaluados en interrelaciones

Al transformar la relación se habrá migrado el atributo a alguna de las entidades relacionadas o se habrá creado una nueva entidad (tabla) y en ella estará el atributo.

Una vez situado el atributo en una entidad (o tabla) se elegirá una de las tres opciones anteriores, según convenga



PATROCINA (PROY, EMPR, FECHA-INICIO, FECHA-FIN)

→ PROYECTO (NOMBRE, DESCRIPCION)

→ EMPRESA (NOMBRE, CIF, DIRECCION, TELEFONO*)

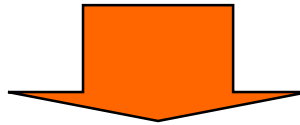
Transformación diseño conceptual a lógico relacional

Algunas restricciones de los esquemas E-R no tienen representación en el esquema relacional por lo que **es necesario realizar anotaciones adicionales** para que se controlen después mediante verificaciones, aserciones o disparadores:

- ☐ Cardinalidades mínimas obligatorias en interrelaciones (excluyendo las que se controlan con clave ajena NO NULL)
- ☐ Cardinalidades máximas conocidas en interrelaciones
- ☐ Exclusividad, totalidad, solapamiento en generalizaciones
- ☐ Inserciones y borrado en generalizaciones
- ☐ Atributos derivados
- ☐ Exclusividad entre interrelaciones

Transformación diseño conceptual a lógico relacional

- ❑ Existen algunas restricciones que es necesario controlar con mecanismos externos al modelo relacional.
- ❑ Los SGBD comerciales no suelen implantar el modelo relacional completo por lo que no siempre están disponibles los mecanismos de verificación, aserción o disparadores



Recurrir a otros medios

(procedimientos almacenados, aplicaciones externas, etc...)