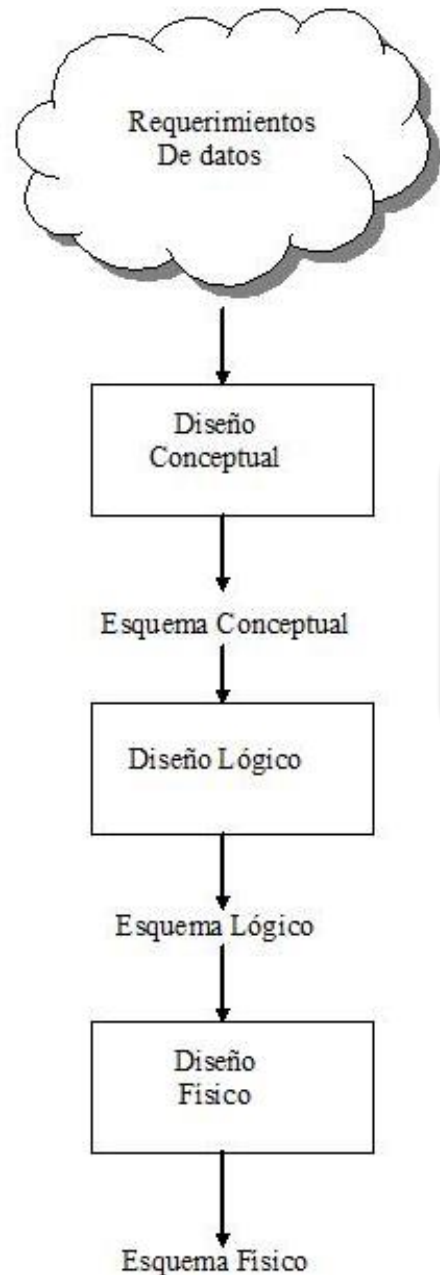


BBDD

Bases de Datos – Modelo Relacional



Modelización de BD



El modelado transcurre en tres fases:

1) Modelo conceptual: Modelo expresivo y entendible por el usuario experto en negocio (pero inexperto en informática). Representa el dominio del problema tal y como el usuario lo concibe. Nosotros trabajamos el **modelo E/R**.

2) Modelo lógico: Más técnico y que tienen traducción directa al modelo físico que entiende el futuro SGBD. Este modelo depende del tipo de BD (no es igual modelar una BD orientada a objetos que una relacional). Nosotros veremos el **Modelo relacional**.

3) Modelo físico: Resultado de aplicar el modelo lógico a un SGBD concreto. Habitualmente se expresa en un lenguaje de programación de BBDD como SQL. La transformación del Modelo Relacional al modelo físico se lleva a cabo mediante el sub-lenguaje **DDL (lenguaje de definición de datos)** de SQL.

Finalmente, a partir del modelo físico se llevará a cabo la **implementación de la BD**.

Introducción

El **modelo relacional** independiza al usuario de las estructuras de datos físicas con las que se representa la información en una base de datos, lo cual permite que la base de datos pueda implementarse en cualquier SGBD relacional (Oracle, MySQL, DB2, etc.)





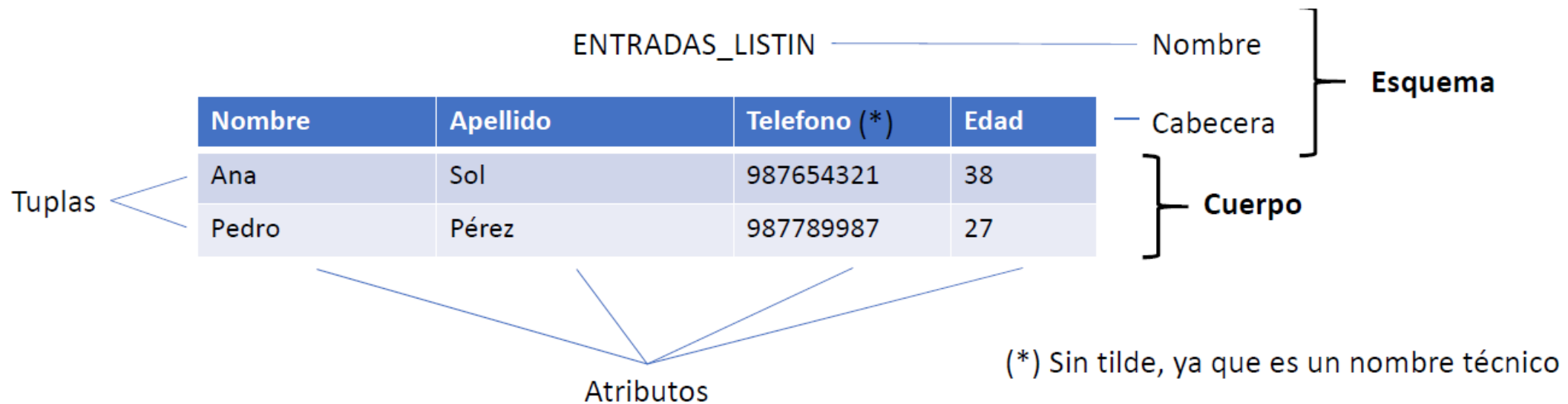
Características del Modelo Relacional

- Los datos son atómicos o monovaluados;
- Los datos de cualquier columna son de un solo tipo.
- Cada columna posee un nombre único.
- El orden de las columnas no es de importancia para la tabla.
- Las columnas de una relación se conocen como atributos.
- Cada atributo tiene un dominio.
- No existen 2 filas en la tabla que sean idénticas.
- Cada relación tiene un nombre específico y diferente al resto de las relaciones.
- Los atributos son atómicos: en cada tupla, cada atributo toma un solo valor.
- El orden de los atributos no importa: los atributos no están ordenados.
- Cada tupla es distinta de las demás: no hay tuplas duplicadas.
- El orden de las tuplas no importa: las tuplas no están ordenadas.
- Cada relación tiene un número fijo de atributos para todas las tuplas.

Las relaciones (I)

- La relación es el elemento fundamental del modelo relacional. Los usuarios ven la BD como una colección de relaciones. Estas relaciones se pueden operar mediante el **Álgebra Relacional**.
- Se define **relación** como un conjunto de atributos, cada uno de los cuales pertenece a un dominio, y que posee un nombre que identifica la relación.
- Gráficamente se representa como una **tabla** con columnas (atributos) y filas (tuplas).
- El conjunto de tuplas de una relación representa el **cuerpo** de la relación, el conjunto de atributos + nombre, el **esquema**.

Las relaciones (II)



Vocabulario específico del modelo

- **La relación** es un conjunto de atributos, cada uno de los cuales pertenece a un dominio, y que posee un nombre que identifica la relación.
- Las **tuplas** son cada uno de los elementos que contiene una instancia de la relación (filas).
- Los **atributos** de una relación son las características que la describen. Es decir, los elementos que pueden tomar valores (columnas)
- El **Grado** de una relación es el número de atributos que tiene.
- El **dominio** de un atributo es el conjunto de valores permitidos para ese atributo. Ej: cadena de caracteres, números enteros, valores sí o no, una fecha, etc.
- **Clave** de una relación es un conjunto de atributos que identifican de forma única cada ocurrencia. Pueden ser simples (atómicas) o compuestas y además hay varios tipos:

Superclave: Conjunto de atributos que proporcionan la funcionalidad de clave.

Clave candidata: Superclave mínima en cuanto al número de atributos utilizados.

Clave primaria: Clave candidata elegida por el diseñador como clave definitiva.

Clave foránea/extranjera: atributo de una relación externa que es clave en aquella relación.

Ejemplo de tipos de clave

Dada la entidad **Empleado** cuyos atributos son DNI, N°SS, Nombre, Edad,Codigo_Pais. Y la entidad **Países** cuyos atributos son CodPais y DescripPais.

La relación **Empleados** tendría:

Una **Superclave** que podría ser DNI + NSS + Nombre + Edad + Codigo_Pais

Como **claves candidatas** podríamos tener dos: DNI o bien NSS

Como **clave primaria** podríamos elegir: DNI por ser más representativa para un empleado

Como **clave foránea** figura: CodPais que sería clave primaria en la entidad **Países**.

EMPLEADOS (DNI, NSS, Nombre, Edad, Codigo_Pais(FK))



Analogías entre los términos

Nomenclatura modelo E/R	Nomenclatura modelo relacional	Nomenclatura tabla	Nomenclatura ficheros
Entidad, relación	Relación	Tabla	Fichero
Atributo	Atributo	Columna	Campo
Ocurrencia	Tupla	Fila	Registro
N.º de atributos	Grado	N.º de columnas	N.º de campos
Conjunto de ocurrencias	Cuerpo	Conjunto de filas	Conjunto de registros



Otros conceptos del modelo

- Las **restricciones inherentes al modelo relacional**:
 - No puede haber dos tuplas iguales en una relación.
 - Cada atributo solo puede tomar un único valor de su dominio.
- Las **restricciones semánticas** son las condiciones que deben cumplir los datos para su correcto almacenamiento:
 - **De valor único** (UNIQUE): se impide que un atributo tenga un valor repetido.
 - **De clave** (PRIMARY KEY): conjunto de atributos que identifican de forma única una tupla.
 - **De obligatoriedad** (NOT NULL): informar el atributo (darle valor) no es opcional, es obligatorio.
 - **De integridad referencial** (FOREIGN KEY): el valor referenciado debe existir en su tabla origen.
 - Borrado y modificación en cascada (CASCADE): cuando se elimina un registro de la tabla principal automáticamente se borran también los registros relacionados en las tablas secundarias. Las actualizaciones también se propagan.
 - Borrado y modificación restringidos (RESTRICT): impide realizar modificaciones o borrados que atenten contra la integridad referencial.
 - **De dominio** (CHECK, TRIGGER): el valor que pueda tomar un campo debe estar dentro del dominio. Se puede limitar el rango de valores de un atributo o ejecutar un procedimiento durante la inserción, modificación o eliminación de registros.

Notación simplificada

La **notación** que usaremos en la representación simplificada debe cumplir:

- El nombre de la relación se escribe en mayúsculas y en plural.
- Los atributos siguen al nombre de la relación, entre paréntesis y separados por comas. Los escribimos con la primera letra mayúscula y el resto en minúsculas, o bien separando cada nueva palabra con una mayúscula. Al ser nombres técnicos no usamos tildes, espacios ni caracteres raros.
- Mostramos siempre primero los atributos que forman parte de la clave primaria y deben aparecer subrayados.
- Los atributos que son claves foráneas/extranjeras aparecen seguidos de (FK).
- Ejemplos:

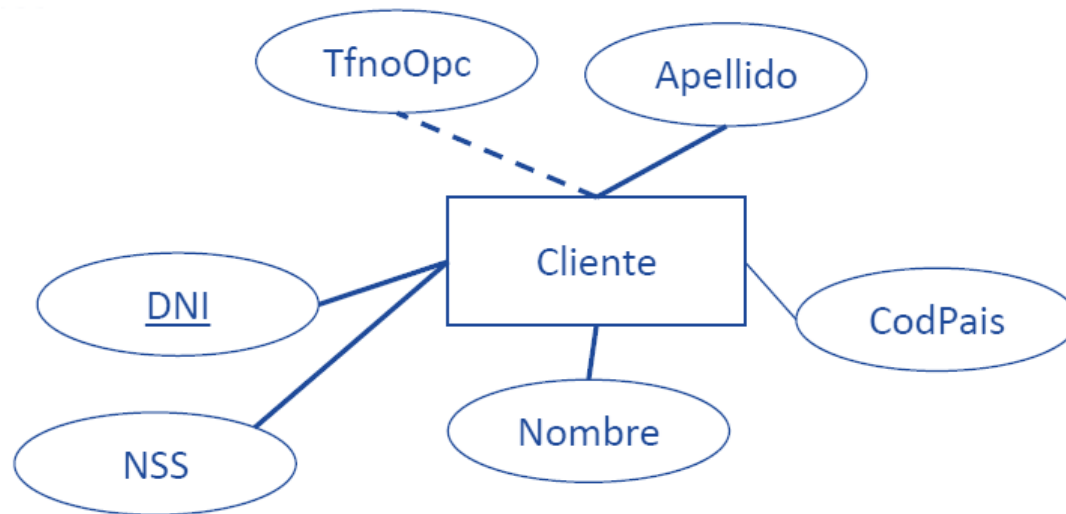
TRABAJADORES (CodTrabajador, CodEmpresa (FK), Nombre, Apellido, CodPais (FK))

COMPRAS (CodCliente (FK), Coproducto (FK), FechaHora, Cantidad, Importe)

COCHES (Matricula, CodModelo (FK), CodMarca (FK), Kilometros)

➤ Transformación Modelo E/R ➡ Relacional

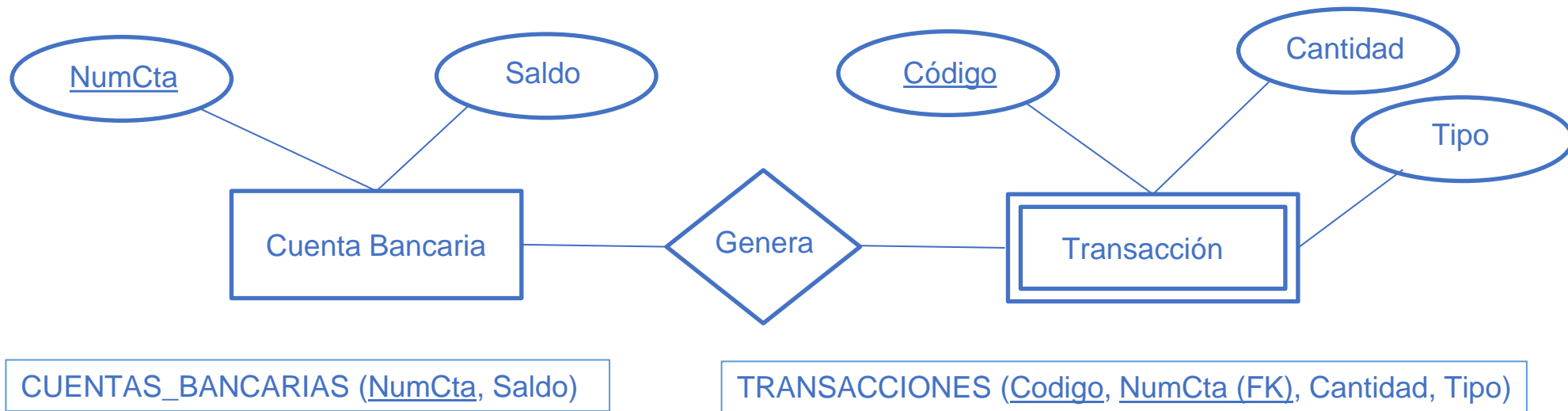
Por cada entidad fuerte, crearemos una tabla con tantas columnas como atributos tenga la entidad. La clave primaria la formarán los atributos clave de la entidad.



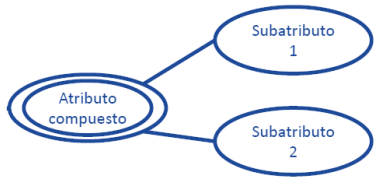
CLIENTES (DNI, NSS, Nombre, Apellido, TfnoOpc, CodPais (FK))

Transformación Modelo E/R ➡ Relacional

Por cada entidad débil, crearemos una tabla con tantas columnas como atributos tenga la entidad. La clave primaria de entidad fuerte de la que depende, migrará como clave foránea a la entidad débil. La clave primaria de la entidad débil, pasará a ser, la combinación de la clave primaria de la entidad fuerte + un discriminante (atributo propio de entidad débil que lo diferencia)



Transformación Modelo E/R ➡ Relacional



Si tengo **atributos compuestos** (constituidos por otros atributos con significado propio), todos los atributos atómicos de cada atributo compuesto, pasarán a ser atributos simples (atómicos) en la relación.



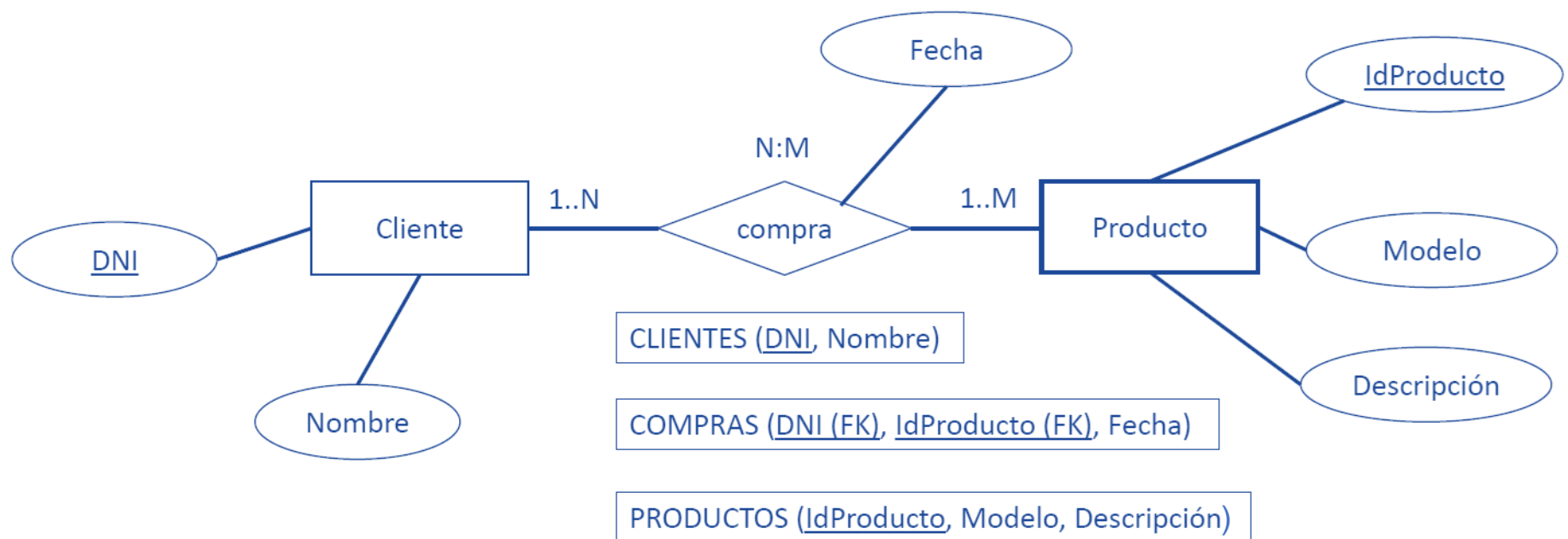
Si tengo **atributos derivados** (valores se pueden calcular a partir de valores de atributos relacionados), no se almacenarán en la base de datos por lo que no se incluyen como atributos en las relaciones.



Si tengo **atributos multivaluados** (aquellos que contienen múltiples valores por cada instancia), crearemos una tabla auxiliar en la que se migrará la clave primaria de la entidad origen que pasará a ser clave foránea y además se añadirá el atributo multivaluado. La unión de ambos será la clave primaria por norma general.

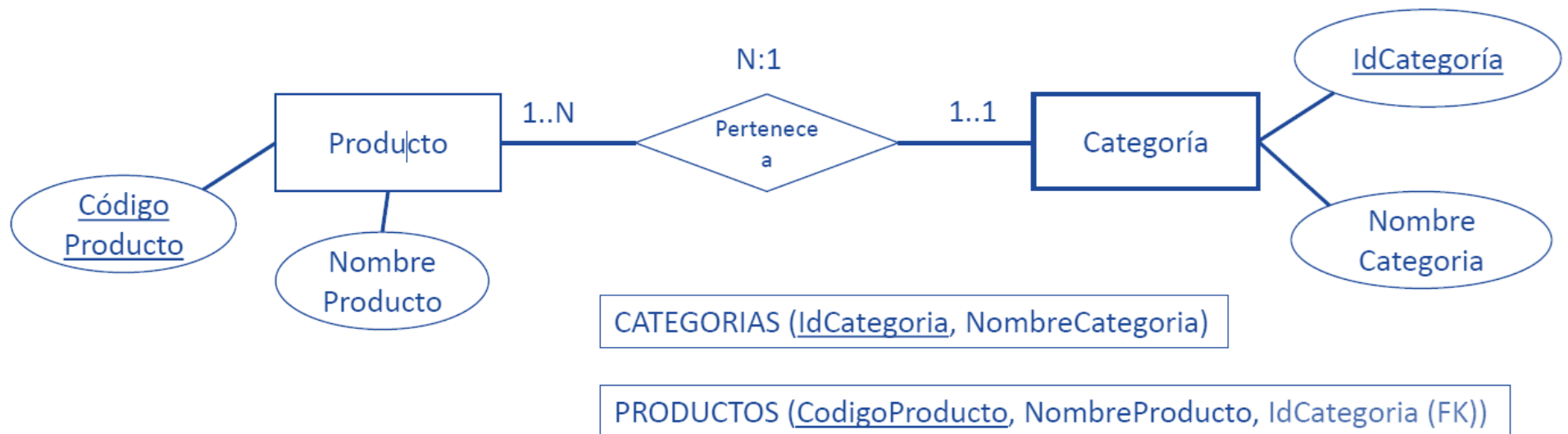
Transformación Modelo E/R ➡ Relacional

Por cada relación N:M, crearemos una nueva tabla para la relación cuya clave primaria será la unión de las claves primarias de las tablas relacionadas.



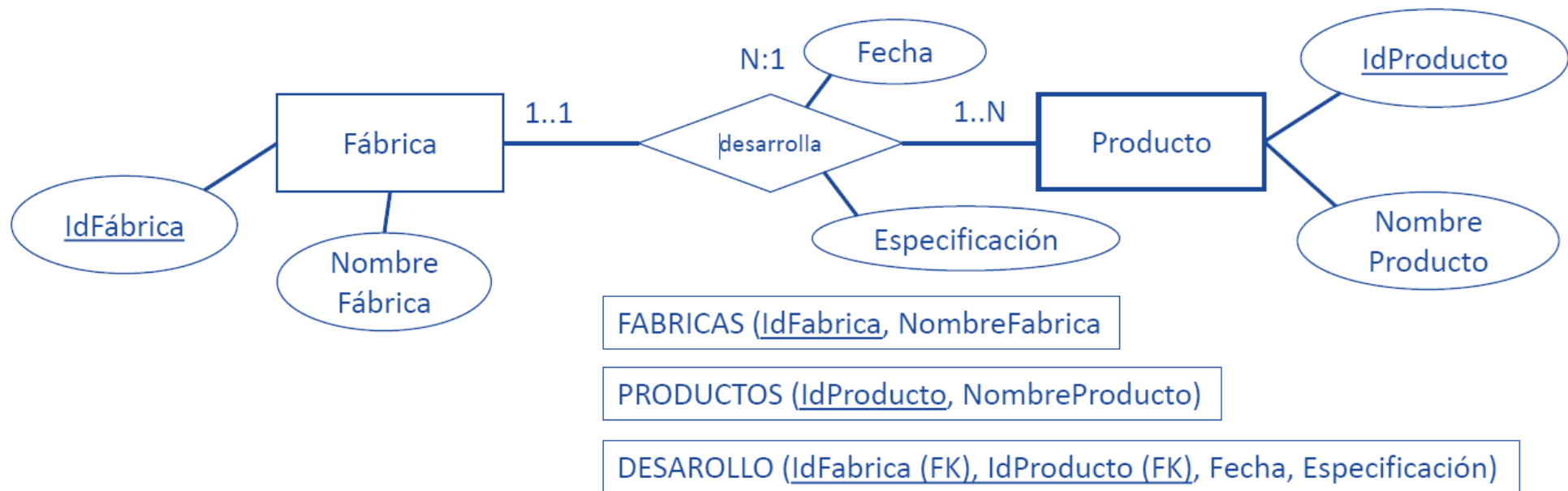
Transformación Modelo E/R ➡ Relacional

Por cada relación 1:N, lo normal será NO crear una nueva tabla, pero la tabla correspondiente de la entidad del lado de la participación mayor pasará a incluir la clave de la otra entidad como clave foránea.



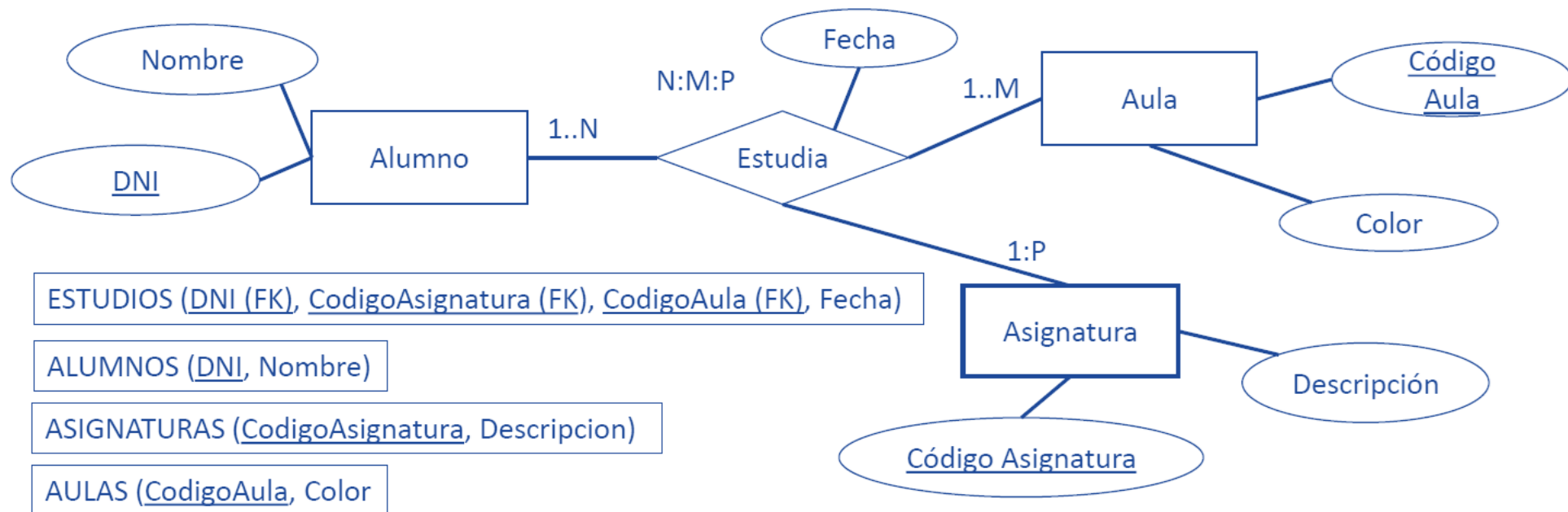
Transformación Modelo E/R ➡ Relacional

Si la relación **1:N**, tiene varios atributos y se prevé que pueda convertirse en N:M a futuro, podremos aplicar la misma lógica que una relación N:M.



Transformación Modelo E/R ➡ Relacional

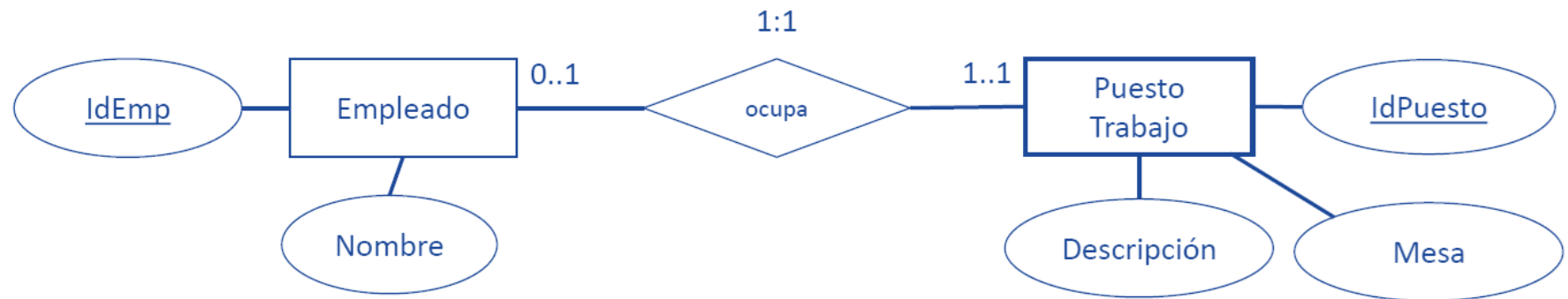
Por cada relación N:M:P, crearemos una nueva tabla, en plural, cuya clave está formada por todas las claves de las entidades que participan en la relación más los atributos propios de la entidad.



Transformación Modelo E/R ➡ Relacional

Por cada relación 1:1, tendremos varias opciones:

a) Si tenemos una participación 0:1 de un lado y 1:1 del otro, propagaremos la clave de la entidad del lado 1:1 a la del lado 0:1



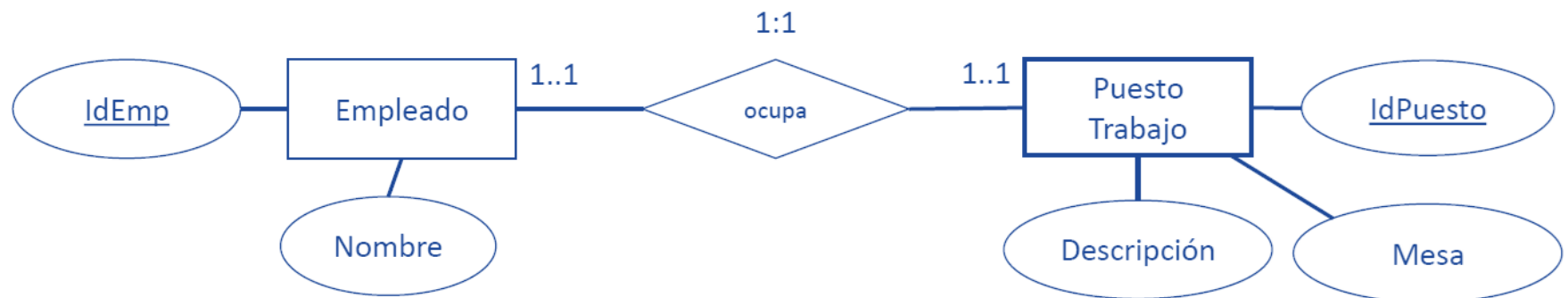
EMPLEADOS (IdEmp, Nombre, IdPuesto (FK))

PUESTOS (IdPuesto, Descripción, Mesa)

Transformación Modelo E/R ➔ Relacional

Por cada relación 1:1, tendremos varias opciones:

b) Si tenemos una participación 1:1 de ambos lados, podemos, o bien propagar una de las claves (como en la opción a), o bien crear una tabla como si se tratara de una relación N:M, o en última opción (la menos recomendable), fusionar las dos tablas en una única.



EMPLEADOS (IdEmp, Nombre, IdPuesto (FK))

PUESTOS (IdPuesto, Descripción, Mesa)

EMPLEADOS (IdEmp, Nombre)

PUESTOS (IdPuesto, Descripción, Mesa, IdEmp (FK))

EMPLEADOS (IdEmp, Nombre)

PUESTOS (IdPuesto, Descripción, Mesa)

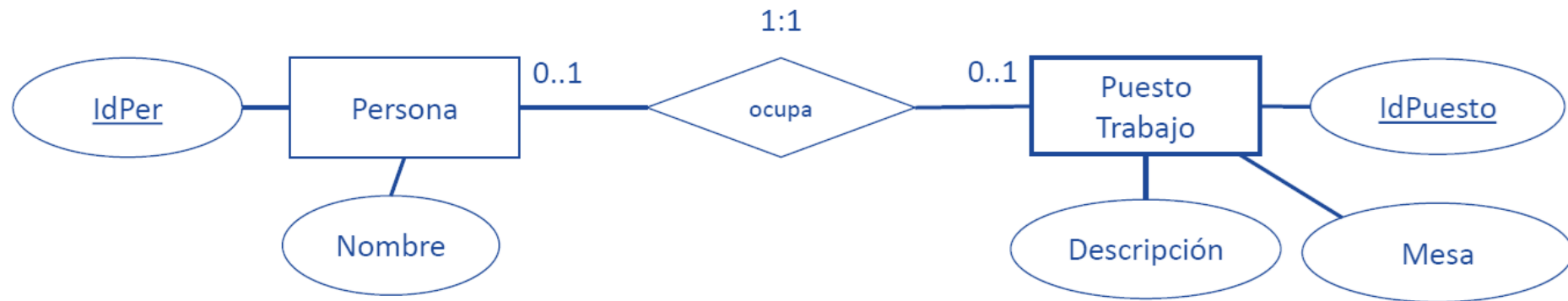
EMPLEADOS_PUESTOS (IdEmp(FK), IdPuesto(FK))

EMPLEADOS_PUESTOS (IdEmp, IdPuesto, Nombre, Mesa, DescPuesto)

Transformación Modelo E/R ➡ Relacional

Por cada relación 1:1, tendremos varias opciones:

c) Si tenemos una participación 0:1 de ambos lados, podemos, o bien propagar una de las claves (como en la opción a), o bien crear una tabla como si se tratara de una relación N:M.



PERSONAS (IdPer, Nombre, IdPuesto (FK))

PUESTOS (IdPuesto, Descripción, Mesa)

PERSONAS (IdPer, Nombre)

PUESTOS (IdPuesto, Descripción, Mesa, IdPer (FK))

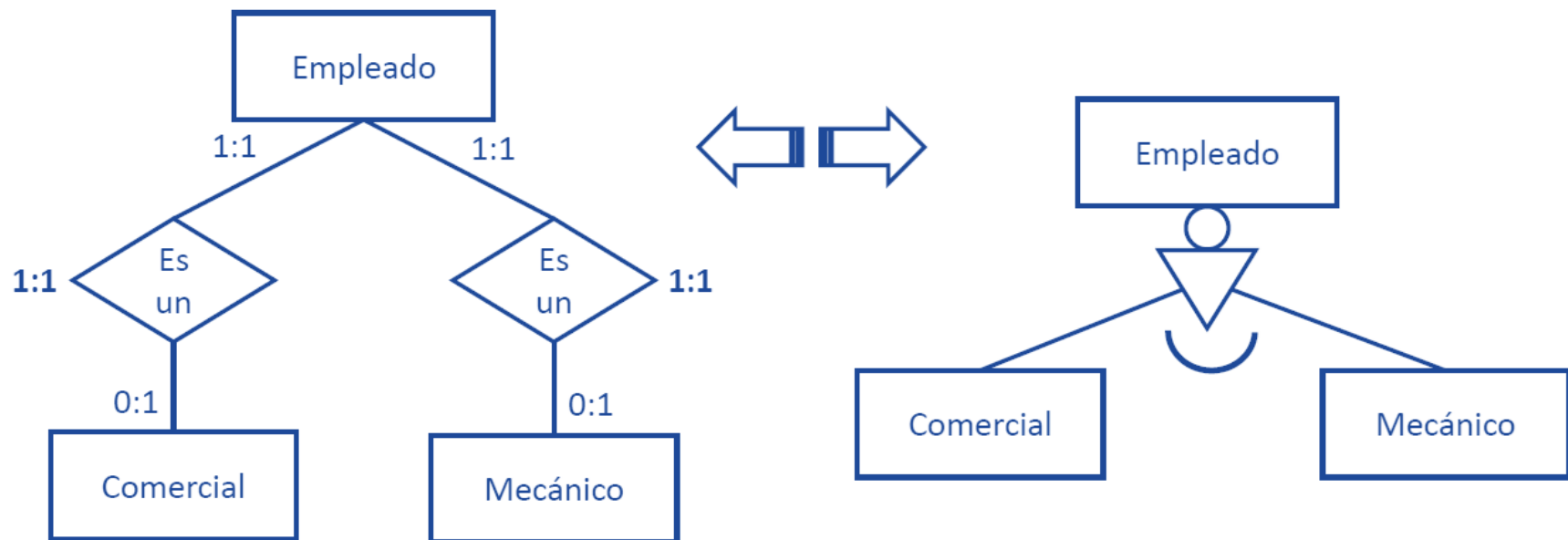
PERSONAS (IdPer, Nombre)

EMPLEADOS_PUESTOS (IdPer (FK), IdPuesto (FK))

PUESTOS (IdPuesto, Descripción, Mesa)

Transformación Modelo E/R ➡ Relacional

Nota: Con las **especializaciones** (las vemos posteriormente), sabemos que aparecen muchas relaciones con **cardinalidad 1:1**

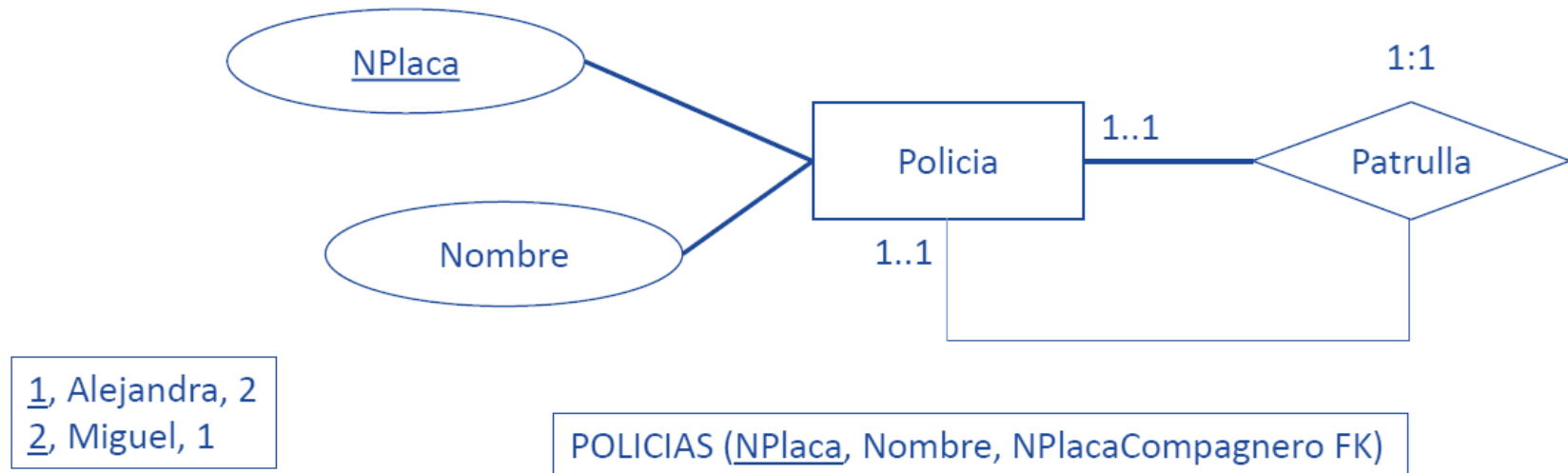


Transformación Modelo E/R ➡ Relacional

Por cada relación reflexiva, tendremos varias opciones:

a) Si la cardinalidad es 1:1, NO crearemos una nueva tabla sino que añadiremos sobre la misma tabla, con otro nombre y como clave foránea, la clave primaria.

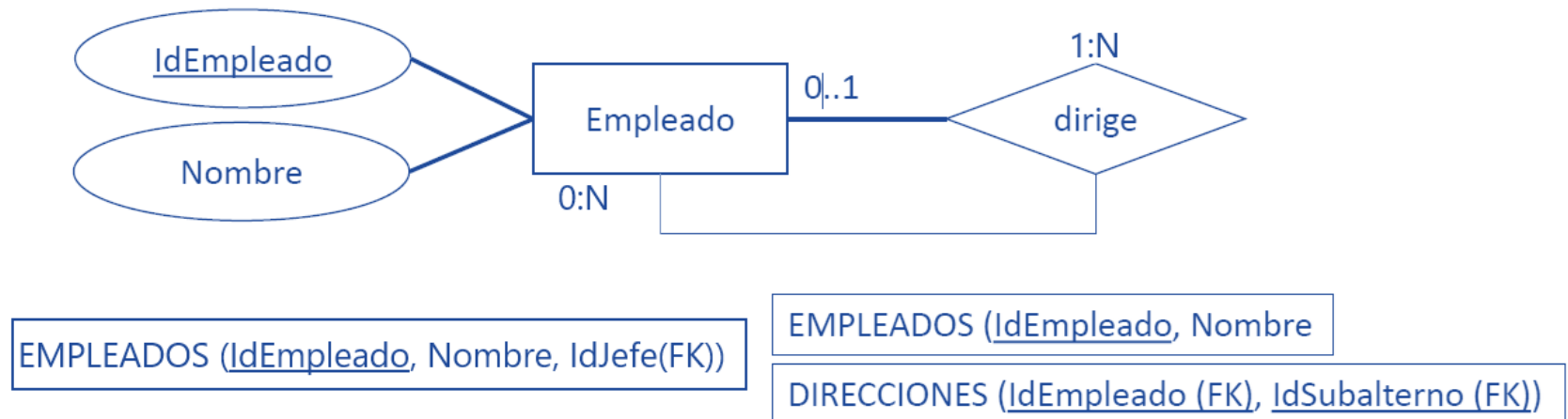
Nota: Se asume que las parejas siempre son las mismas



Transformación Modelo E/R ➡ Relacional

Por cada relación reflexiva, tendremos varias opciones:

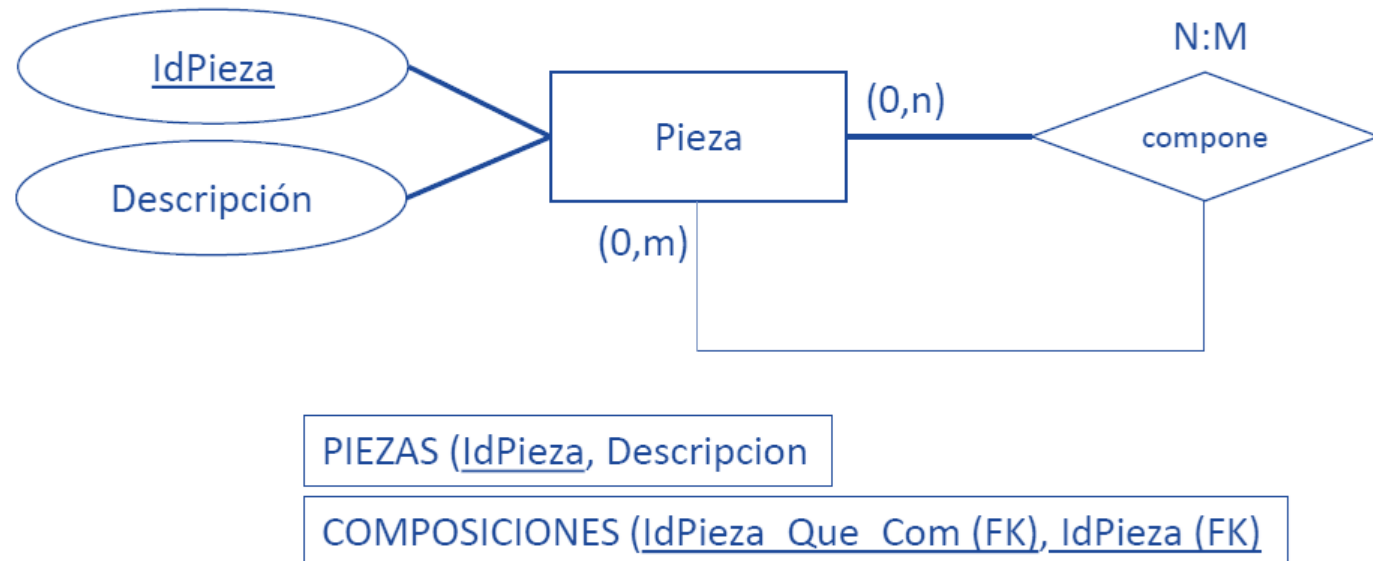
b) Si la cardinalidad es 1:N, NO crearemos una nueva tabla sino que añadiremos sobre la misma tabla, con otro nombre y como clave foránea, la clave primaria. Aunque también se podría crear una nueva tabla cuya PK sea la unión de la PK consigo misma con distinto nombre.



Transformación Modelo E/R ➡ Relacional

Por cada relación reflexiva, tendremos varias opciones:

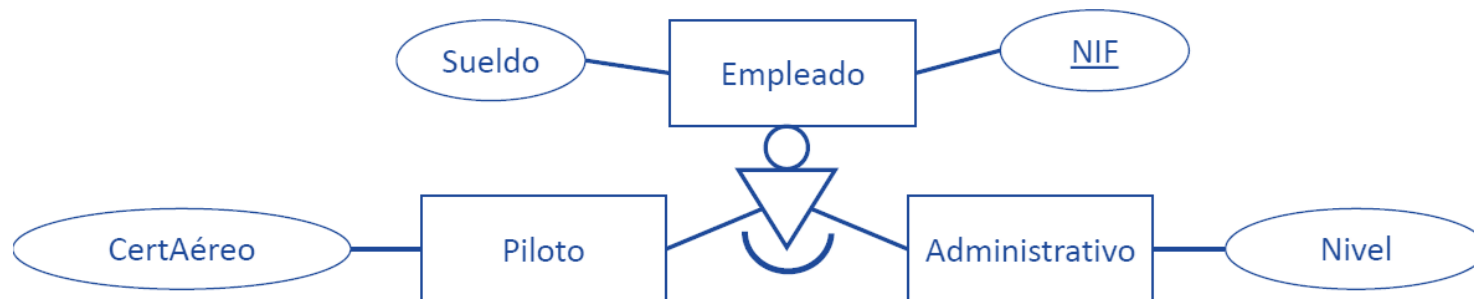
c) Si la cardinalidad es N:M, crearemos una nueva tabla para la relación cuya clave primaria será la unión de las claves primarias de las tablas relacionadas.



Transformación Modelo E/R ➡ Relacional

Por cada especialización, tendremos varias opciones:

- ★ a) Podremos crear una tabla para la superclase y otra por cada subclase: La clave primaria de la superclase debe aparecer como clave primaria de las subclases. Los atributos comunes aparecen en la tabla de la superclase. Los atributos específicos de cada subclase en sus tabla correspondiente.



EMPLEADOS (NIF, Sueldo)

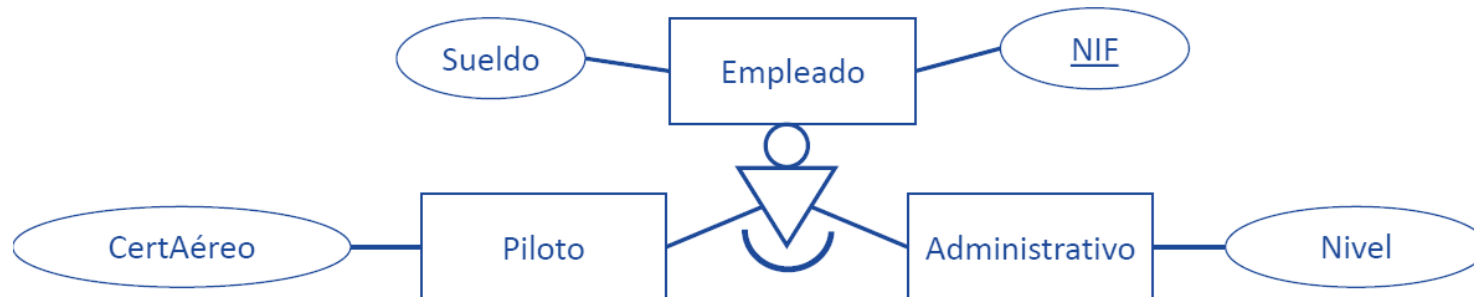
PILOTOS (NIF(FK), CertAereo)

ADMINISTRATIVOS (NIF(FK), Nivel)

Transformación Modelo E/R ➡ Relacional

Por cada especialización, tendremos varias opciones:

b) Podremos crear una tabla por cada subclase únicamente: La clave primaria de las subclases será la que identificaba a la superclase. Los atributos de la superclase aparecen repetidos en la tabla de cada subclase. Los atributos específicos de cada subclase en sus tabla correspondiente.



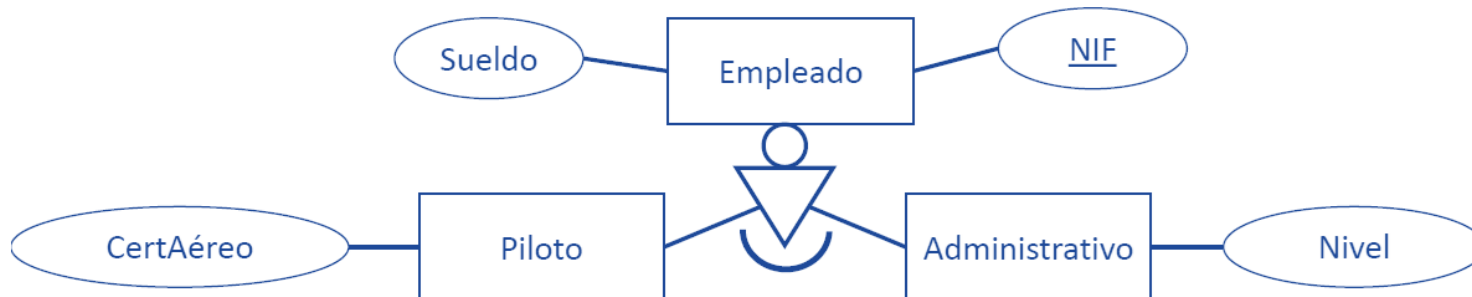
PILOTOS (NIF, CertAereo, Sueldo)

ADMINISTRATIVOS (NIF, Nivel, Sueldo)

Transformación Modelo E/R ➡ Relacional

Por cada especialización, tendremos varias opciones:

c) Podremos crear una única tabla para la superclase y luego fusionar los atributos propios y de las subclases y creando un atributo nuevo llamado tipo o bien fusionar los atributos propios y de las subclases pero creando un atributo de validez para cada atributo específico de las subclases.



EMPLEADOS (NIF, CertAereo, Sueldo, Nivel, Tipo)

EMPLEADOS (NIF, CertAereo, Sueldo, Nivel, EsPiloto, EsAdministrativo)

Pérdida de semántica

Debemos tener en cuenta que en el paso del modelo E/R al modelo relacional, algunas características representadas en el modelo conceptual pueden no representarse bien en el modelo lógico suponiendo así una **perdida de semántica** en el proceso.

Esto sucede por ejemplo en cuanto tenemos generalizaciones o especializaciones y debemos controlarlas mediante las restricciones semánticas (ej. check)



Resumen

Modelo E/R		Modelo relacional
Relación		Resolución
Binaria	1:1	Propagar clave: clave ajena / otra tabla / uniendo las dos entidades en una nueva tabla.
	1:N	Propagar clave: clave ajena / excepcionalmente otra tabla.
	M:N	Tabla nueva con clave primaria = unión de las claves primarias de las dos tablas.
Reflexiva	1:1	Clave ajena.
	1:N	Clave ajena / excepcionalmente otra tabla.
	M:N	Otra tabla con clave ajena duplicada.
Ternaria		Tabla nueva con clave primaria = unión de las claves primarias de entidades de lado 1:N.