

# Tema 2. Diseño lógico de bases de datos

---

## 2.1. INTRODUCCIÓN

## 2.2. DISEÑO DE BD

- **2.2.1. Fase de Análisis: Especificación de requisitos Software (E.R.S.)**
- **2.2.2. Fase 1 del diseño. Diseño Conceptual: Modelo Entidad/Relación (E/R)**
- **2.2.3. Fase 2 del diseño. Diseño Lógico: Modelo Relacional**
- **2.2.4. Fase 3 del diseño. Diseño Físico: Modelo Físico**

## 2.3. MODELO ENTIDAD/RELACIÓN

- **2.3.1. Entidades**
- **2.3.2. Atributos**
- **2.3.3. Relaciones**

## 2.4. MODELO E/R EXTENDIDO

- **2.4.1. Relaciones de Jerarquía**

## 2.5. MODELO RELACIONAL

- **2.5.1. Introducción**
- **2.5.2. Estructura de datos relacional**
- **2.5.3. Elementos y propiedades del modelo relacional**
- **2.5.4. Transformación de un esquema E/R a esquema relacional**

## 2.6. NORMALIZACIÓN

- **2.6.1. Primera Forma Normal: 1FN**
- **2.6.2. Segunda Forma Normal: 2FN**
- **2.6.3. Tercera Forma Normal: 3FN**
- **2.6.4. Forma Normal de Boyce-Codd: FNBC**
- **2.6.5. Cuarta Forma Normal: 4FN**
- **2.6.6. Quinta Forma Normal: 5FN**

## 2.5. MODELO RELACIONAL

### 2.5.1. Introducción

Los SGBD se pueden clasificar de acuerdo con el modelo lógico que soportan, el número de usuarios, el número de puestos, el coste... La clasificación más importante de los SGBD se basa en el modelo lógico, siendo los principales modelos que se utilizan en el mercado los siguientes: Jerárquico, en Red, Relacional y Orientado a Objetos.

La mayoría de los SGBD comerciales actuales están basados en el modelo relacional, en el que nos vamos a centrar, mientras que los sistemas más antiguos estaban basados en el modelo de red o el modelo jerárquico.

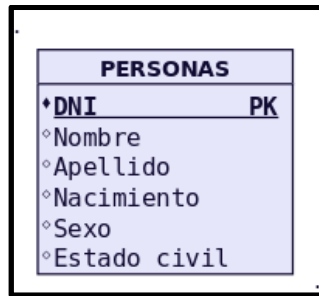
Los motivos del éxito del modelo relacional son fundamentalmente dos:

- Se basan en el álgebra relacional que es un modelo matemático con sólidos fundamentos. En esta sección se presenta el **modelo relacional**. Realizaremos la descripción de los principios básicos del modelo relacional: la estructura de datos relacional y las reglas de integridad. Ofrecen sistemas simples y eficaces para representar y manipular los datos.
- La estructura fundamental del modelo relacional es precisamente esa, la «**relación**», es decir una tabla bidimensional constituida por **filas** (registros o tuplas) y **columnas** (atributos o campos). Las relaciones o tablas representan las entidades del modelo E/R, mientras que los atributos de la relación representarán las propiedades o atributos de dichas entidades. Por ejemplo, si en la base de datos se tiene que representar la entidad PERSONA, esta pasará a ser una relación o tabla llamada «PERSONAS», cuyos atributos describen las características de las personas (tabla siguiente). Cada tupla o registro de la relación «PERSONAS» representará una persona concreta.

### 2.5.2. Estructura de datos relacional

PERSONAS					
D.N.I.	Nombre	Apellido	Nacimiento	Sexo	Estado civil
52.768.987	Juan	Loza	15/06/1976	H	Soltero
06.876.983	Isabel	Gálvez	23/12/1969	M	Casada
34.678.987	Micaela	Ruiz	02/10/1985	M	Soltera

En realidad, siendo rigurosos, una RELACIÓN del MODELO RELACIONAL es sólo la definición de la estructura de la tabla, es decir su nombre y la lista de los atributos que la componen. Una representación de la definición de esa relación podría ser la siguiente:



Para distinguir un registro de otro, se usa la «**clave primaria o clave principal**».

En una relación puede haber más combinaciones de atributos que permitan identificar unívocamente una fila (estos se llamarán «llaves o claves candidatas»), pero entre éstas se elegirá una sola para utilizar como llave primaria. **Los atributos de la llave primaria no pueden asumir el valor nulo.**

### 2.5.3. Elementos y propiedades del modelo relacional

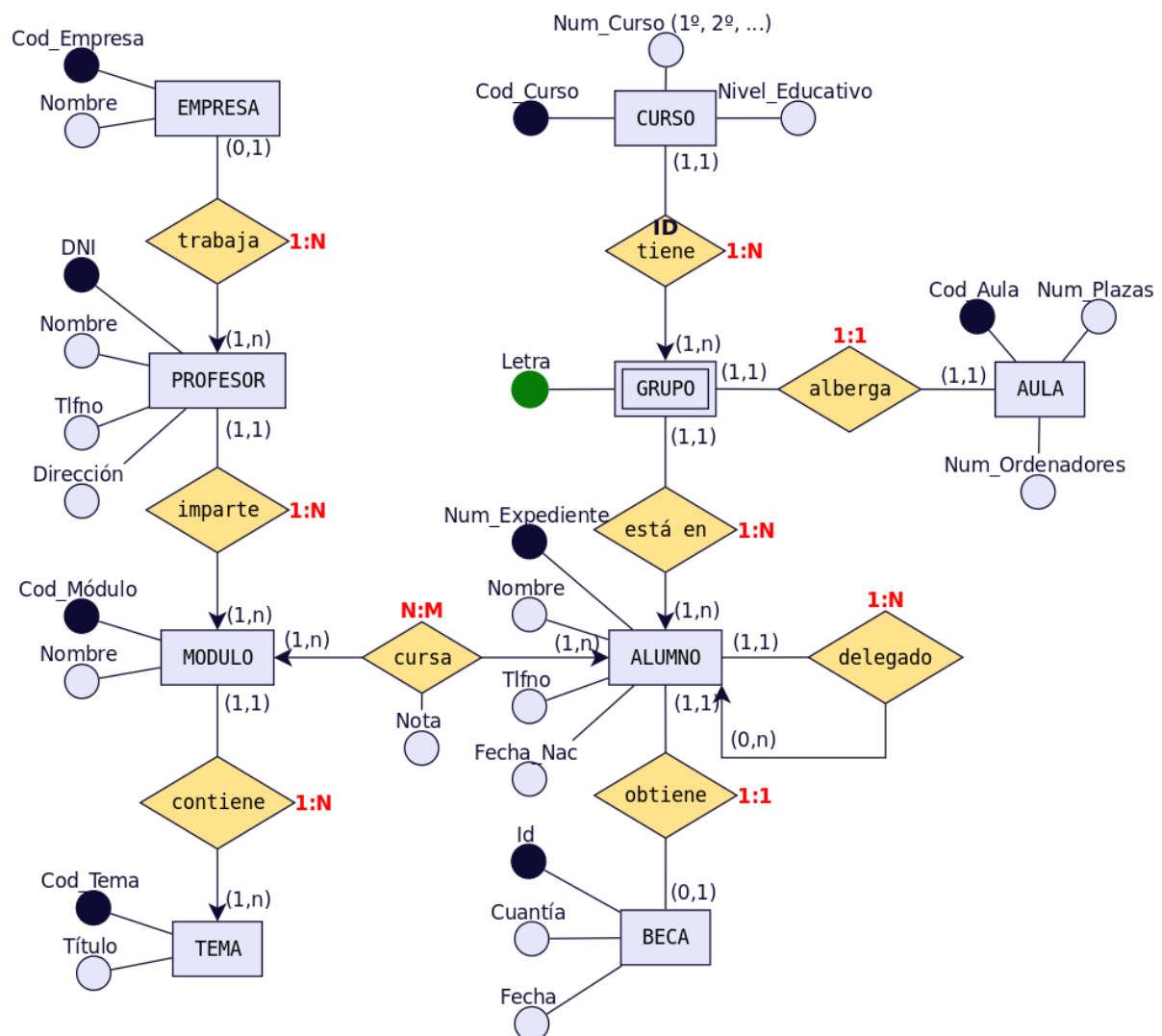
- **Atributo:** Características que describen a una entidad o relación.
- **Relación (tabla):** Representan las entidades de las que se quiere almacenar información en la BD. Está formada por:
  - **Filas (Registros o Tuplas):** Corresponden a cada ocurrencia de la entidad.
  - **Columnas (Atributos o campos):** Corresponden a las propiedades de la entidad. Siendo rigurosos una relación está constituida sólo por los atributos, sin las tuplas.
- Las relaciones tienen las siguientes **propiedades**:
  - Cada relación tiene un nombre y éste es distinto del nombre de todas las demás relaciones de la misma BD.
  - No hay dos atributos que se llamen igual en la misma relación.
  - El orden de los atributos no importa: los atributos no están ordenados.
  - Cada tupla es distinta de las demás: no hay tuplas duplicadas. (Como mínimo se diferenciarán en la clave principal)
  - El orden de las tuplas no importa: las tuplas no están ordenadas.
- **Clave:** Una clave es un conjunto de atributos que identifican de forma única una ocurrencia de entidad. En este caso, las claves pueden ser simples (atómicas) o compuestas. Además, hay varios tipos de clave:
  - **Clave Candidata:** atributo que identifica unívocamente una tupla. Cualquiera de las claves candidatas se podría elegir como clave principal.
  - **Clave Principal o Primaria:** Clave candidata que elegimos como identificador de la tuplas.
  - **Clave Alternativa:** Toda clave candidata que no es clave primaria (las que no hayamos elegido como clave principal).
  - **Clave Externa o foránea o ajena:** el atributo o conjunto de atributos que forman la clave principal de otra relación. Que un atributo sea clave ajena en una tabla significa que para introducir datos en ese atributo, previamente han debido introducirse en la tabla de origen. Es decir, los valores presentes en la

clave externa tienen que corresponder a valores presentes en la clave principal correspondiente (**Integridad Referencial**).

- Una clave principal no puede asumir el valor nulo (**Integridad de la entidad**).
- **Dominio de un atributo:** Conjunto de valores que pueden ser asumidos por dicho atributo. Por ejemplo, cadenas de caracteres, números enteros, valores sí/no, etc.

#### 2.5.4. Transformación de un esquema E/R a esquema relacional

Pasamos ya a enumerar las normas para traducir del Modelo E/R al modelo relacional, ayudándonos del siguiente ejemplo:



Es muy importante tener en cuenta los siguientes puntos:

- Al pasar del esquema E/R al esquema Relacional deberemos añadir las **claves foráneas** necesarias para establecer las interrelaciones entre las tablas. Dichas claves foráneas no aparecen representadas en el esquema E/R.

- Se deben elaborar los diagramas relacionales de tal forma que, posteriormente al introducir datos, **no quede ninguna clave foránea a valor nulo (NULL)**. Para ello se siguen las reglas que se muestran a continuación.

#### 2.5.4.1. Entidades

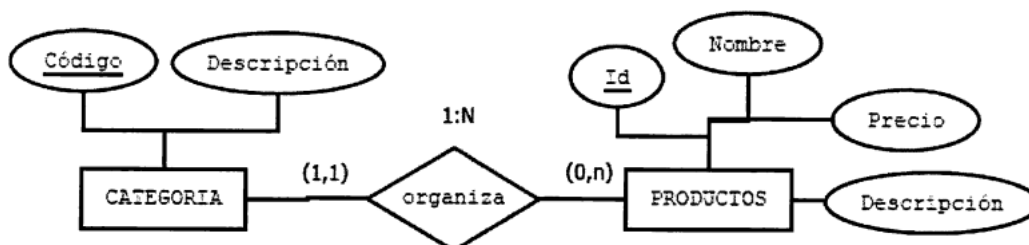
Cada entidad se transforma en una tabla. El identificador (o identificadores) de la entidad pasa a ser la clave principal de la relación y aparece subrayada o con la indicación: **PK (Primary Key)**. Si hay clave alternativa esta se pone en «negrita».

Ejemplo: Todas las entidades del ejemplo anterior generan tabla. En concreto, la entidad AULA genera la siguiente tabla: **AULA (Cod-Aula, num-plazas, num-ordenadores)**

AULA	
*Cod Aula	PK
◊Num_Plazas	
◊Num_Ordenadores	

#### Transformación de entidades fuertes

Para cada entidad fuerte, se crea una tabla con n columnas correspondientes a los atributos que tiene la entidad. La clave primaria de la tabla la forman los atributos clave de la entidad.



En este diagrama de E/R, el modelo relacional resultante de las entidades sería el siguiente:

CATEGORÍAS(Código, Descripción)  
 PRODUCTOS(Id, Nombre, Precio, Descripción)

#### Transformación de entidades débiles

Se verán más adelante en las relaciones de dependencia en existencia y en identificación.

#### 2.5.4.2. Relaciones binarias (de grado 2)

Regla general para las relaciones: se crea una tabla con todos los campos claves de las entidades relacionadas y los atributos de la relación. La clave primaria de la tabla generada es la suma de los atributos claves de las entidades relacionadas, y cada clave incorporada a la tabla, será una clave foránea que referencia a la tabla de la que se importa.

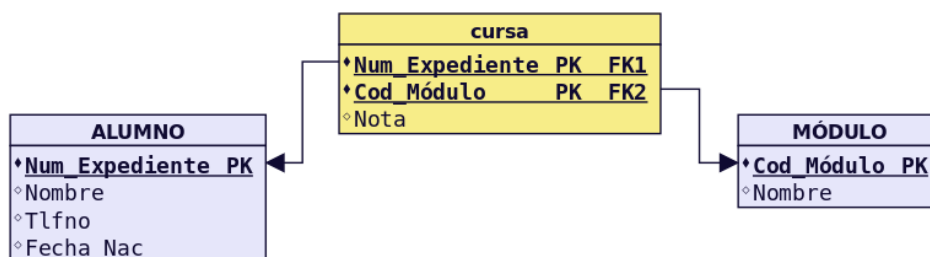
**Relaciones N:M:** Es el caso más sencillo. **Siempre generan tabla.** Se crea una tabla que incorpora como claves ajenas o foráneas **FK (Foreign Key)** cada una de las claves de las entidades que participan en la relación. La clave principal de esta nueva tabla está compuesta por dichos campos.

Es importante resaltar que no se trata de 2 claves primarias, sino de una clave primaria compuesta por 2 campos. Si hay atributos propios, pasan a la tabla de la relación. Se haría exactamente igual si hubiera participaciones mínimas 0.

Orden de los atributos en las claves compuestas: Se deben poner a la izquierda todos los atributos que forman la clave. El orden de los atributos que forman la clave vendrá determinado por las consultas que se vayan a realizar.

Las tuplas de la tabla suelen estar ordenadas siguiendo como índice la clave. Por tanto, conviene poner primero aquel o aquellos atributos por los que se va a realizar la consulta.

Ejemplo: Realicemos el paso a tablas de la relación N:M entre MÓDULO (1,n) y ALUMNO (1,n). Este tipo de relación siempre genera tabla y los atributos de la relación, pasan a la tabla que ésta genera.



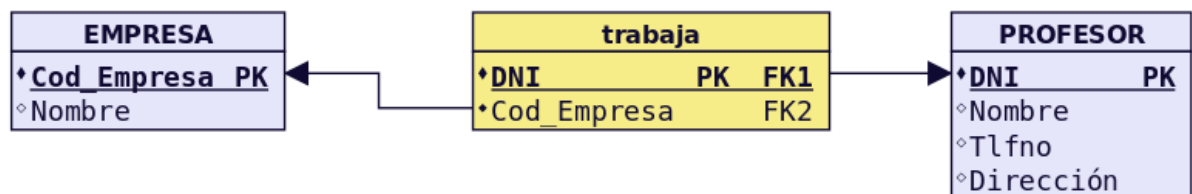
No siempre se aplica la regla general para crear una tabla por cada relación. Generalmente, se pueden encontrar las siguientes excepciones a la regla general:

**Relaciones 1:N:** Por lo general no generan tabla. Se dan 2 casos:

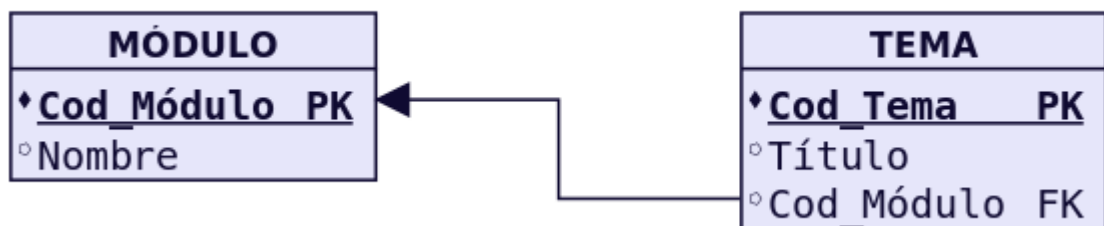
- Caso 1: Si la entidad del lado «1» presenta participación (0,1), entonces se crea una nueva tabla para la relación que incorpora como claves ajenas las claves de ambas entidades. La clave principal de la relación será sólo la clave de la entidad del lado «N».

- Caso 2: Para el resto de situaciones, la entidad del lado «N» recibe como clave ajena la clave de la entidad del lado «1». Los atributos propios de la relación pasan a la tabla donde se ha incorporado la clave ajena.

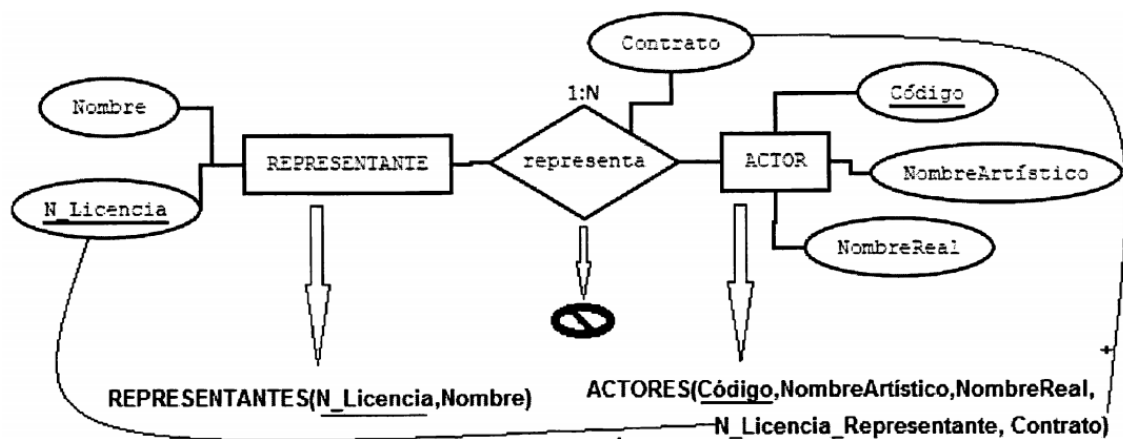
Ejemplo de caso 1: Realicemos el paso a tablas de la relación 1:N entre PROFESOR (1,n) y EMPRESA (0,1). Como en el lado «1» encontramos participación mínima 0, se generará una nueva tabla.



Ejemplo de caso 2: Realicemos el paso a tablas de la relación 1:N entre MÓDULO (1,1) y TEMA (1,n). Como no hay participación mínima «0» en el lado 1, no genera tabla y la clave principal del lado «1» pasa como foránea al lado «n».



Por norma general, en las relaciones de cardinalidad 1:N no se crea una tabla para la relación, sino que se añade a la tabla de la entidad que actúa con participación máxima N la clave de la entidad que actúa con participación máxima 1 (como clave foránea). Si además, la relación tuviera atributos se importarían también a la entidad que actúa con participación máxima N:



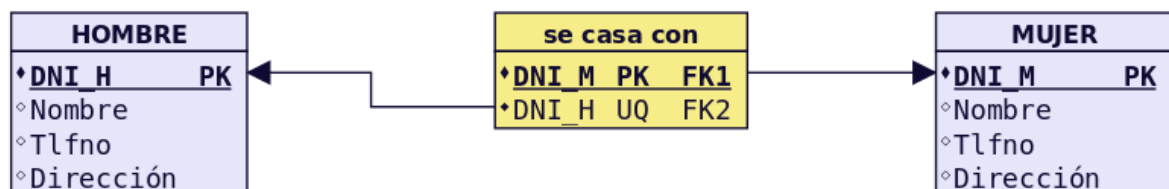
Se añade a ACTORES la clave foránea N\_Licencia\_Representante que referencia al campo N\_Licencia de la tabla REPRESENTANTES. También se ha añadido el campo Contrato, atributo de la relación, a la tabla ACTORES.

ACTORES(Codigo,NombreArtístico,NombreReal,N-Licencia-Representante,Contrato)  
 REPRESENTANTES(N-Licencia,Nombre)

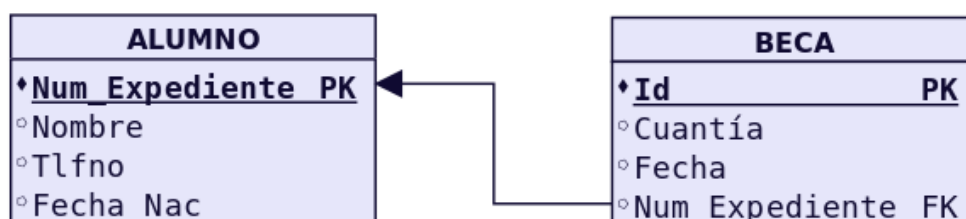
**Relaciones 1:1:** Este tipo de relaciones, por lo general, tampoco generan tabla. El paso a tablas se realiza de forma muy parecida a las relaciones 1-N. En este caso, tampoco se genera tabla para la relación y se tiene la libertad de poder incorporar la clave de una de las dos entidades a la otra. Se dan 3 casos:

- Caso 1: Si las dos entidades participan con participación (0,1), entonces se crea una nueva tabla para la relación.
- Caso 2: Si alguna entidad, pero no las dos, participa con participación mínima 0 (0,1), entonces se pone la clave ajena en dicha entidad, para evitar en lo posible, los valores nulos.
- Caso 3: Si tenemos una relación 1:1 y ninguna tiene participación mínima 0, elegimos la clave principal de una de ellas y la introducimos como clave ajena en la otra tabla. Se elegirá una u otra forma en función de cómo se quiera organizar la información para facilitar las consultas. Los atributos propios de la relación pasan a la tabla donde se introduce la clave ajena.

Ejemplo de caso 1: No se presenta ninguna situación así en el esquema estudiado. Una situación donde puede darse este caso es en HOMBRE (0,1) se casa con MUJER (0,1). Es similar al caso 1 del apartado anterior en relaciones 1:N, aunque en este caso debemos establecer una restricción de valor único para FK2.

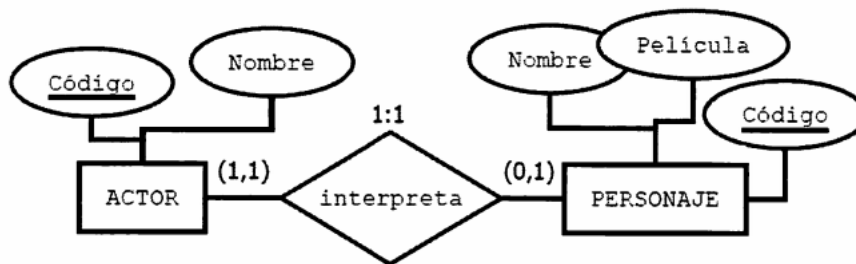


Ejemplo de caso 2 (y 3): Realicemos el paso a tablas de la relación 1:1 entre ALUMNO (1,1) y BECA (0,1). Como BECA tiene participación mínima 0, incorporamos en ella, como clave foránea, la clave de ALUMNO. Esta forma de proceder también es válida para el caso 3, pudiendo acoger la clave foránea cualquiera de las entidades.





Otro ejemplo:



En este caso existen las siguientes opciones:

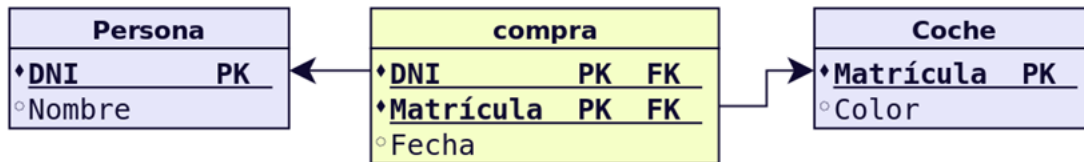
- Incorporar la clave de Personajes como clave foránea en la tabla actores:  
ACTORES(Código, Nombre, CódigoPersonaje)  
PERSONAJES(Código, Nombre, Película)
- Incorporar la clave de Actores como clave foránea en la tabla Personajes:  
ACTORES(Código, Nombre)  
PERSONAJES(Código, Nombre, Película, CódigoActor)
- Incorporar la clave de Actores como clave foránea en la tabla Personajes y la clave de Personajes a la tabla de Actores como clave foránea (Hay que tener en cuenta, que en este caso se está introduciendo una pequeña redundancia, pero puede ser de mucha utilidad para simplificar futuras consultas.):  
ACTORES(Código, Nombre, CódigoPersonaje)  
PERSONAJES(Código, Nombre, Película, CódigoActor)

**Participaciones 0,x:** Ya se han visto casos con participaciones de (0,1) y de (0,n). Normalmente las participaciones son importantes para calcular la cardinalidad de la relación, y transformar conforme a las reglas expuestas hasta ahora. Incluso en muchas ocasiones, las participaciones se omiten en los diagramas entidad-relación. 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.

A continuación se muestra un resumen de los casos disponibles en relaciones N:M, 1:N y 1:1.

## N : M

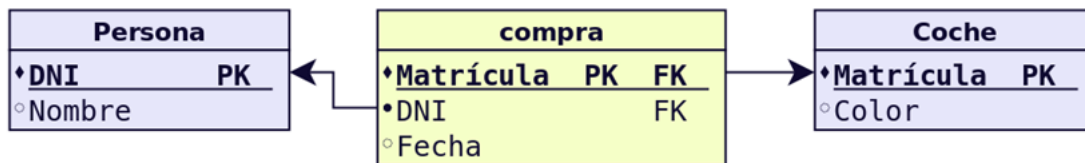
Cualquier tipo de participación



## 1 : N

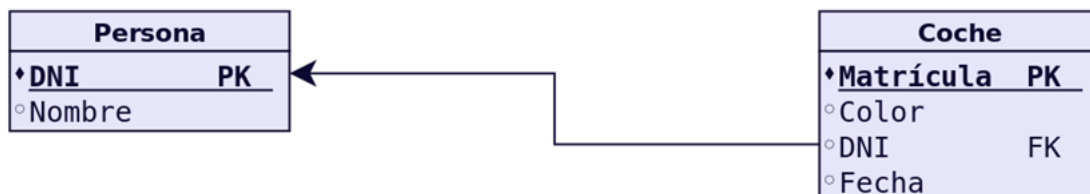
**(0,1)**

Persona (0,1) -- (0,n) Coche  
 Persona (0,1) -- (1,n) Coche



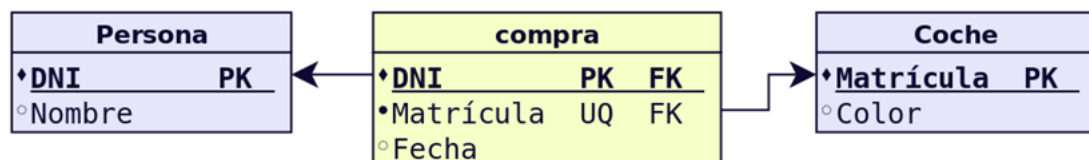
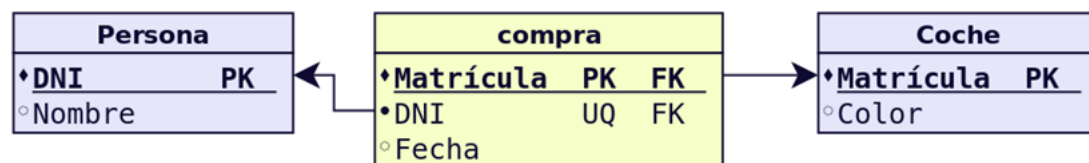
**(1,1)**

Persona (1,1) -- (0,n) Coche  
 Persona (1,1) -- (1,n) Coche

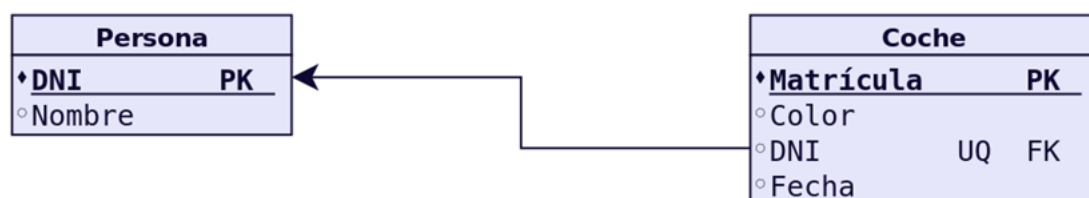


**1 : 1**

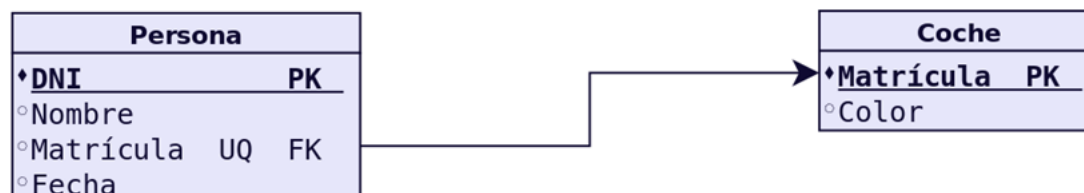
**(0,1)**                      Persona (0,1) -- (0,1) Coche                      **(0,1)**



**(1,1)**                      Persona (1,1) -- (0,1) Coche                      **(0,1)**  
**(1,1)**                      Persona (1,1) -- (1,1) Coche                      **(1,1)**



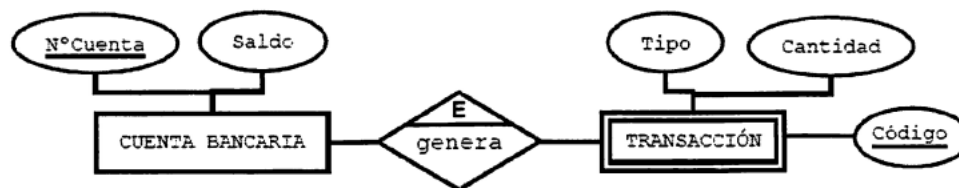
**(0,1)**                      Persona (0,1) -- (1,1) Coche                      **(1,1)**  
**(1,1)**                      Persona (1,1) -- (1,1) Coche                      **(1,1)**



#### 2.5.4.3. Relaciones de dependencia (Siempre de grado 2 y cardinalidad 1:N)

**Relaciones de dependencia en existencia:** Se comportan como una 1:N normal. La clave principal del lado 1 pasa al lado «N» como foránea (hacia donde apunta la flecha) Ejemplo: No encontramos ningún ejemplo, reseñado como tal, en el supuesto anterior. Ahora bien, se comportan en el paso a tablas como cualquier otra relación 1:N. Sólo se tendría en cuenta, el hecho de ser débil en existencia para en el momento de creación de la BD, imponer que al borrar una ocurrencia en el lado «1», se borren las asociadas en el lado «n».

Un ejemplo alternativo de entidades sería el siguiente:

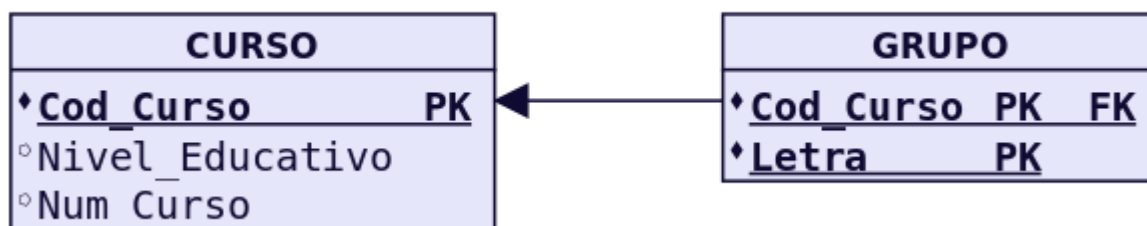


Las tablas generadas de las entidades serían las siguientes:

CUENTAS\_BANCARIAS(Num-cuenta,Saldo)

TRANSACCIONES(Código,Tipo,Cantidad,**Num-cuenta**)

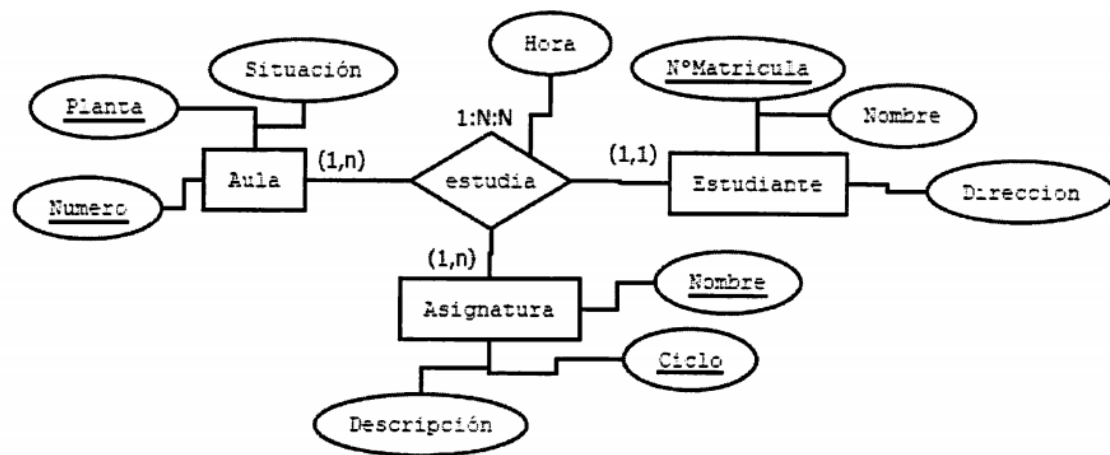
**Relaciones de dependencia en identificación:** Por lo general no generan tablas, porque suelen ser 1:1 o 1:N. Como en toda relación 1:N, La clave de la entidad fuerte debe introducirse en la tabla de la entidad débil como foránea y, además en este caso, formar parte de la clave de ésta. En las entidades débiles, la clave de la entidad fuerte debe ir la primera y, a continuación, los discriminadores de la débil. Ejemplo: Realicemos el paso a tablas de la relación débil en identificación entre CURSO Y GRUPO.



#### 2.5.4.4. Relaciones de grado mayor que 2

Relaciones n-arias (solo veremos hasta grado 3): Siempre generan tabla. Las claves principales de las entidades que participan en la relación pasan a la nueva tabla como claves foráneas. Y solo las de los lados «n» forman la principal. Si hay atributos propios de la relación, estos se incluyen en esa tabla. Ejemplo: No encontramos ningún ejemplo de relación de más de grado 2 en el supuesto anterior. Se va a ver un ejemplo alternativo de una relación ternaria:

En este ejemplo hay dos entidades con clave compuesta, aula y asignatura, y otra, estudiante, que tiene una clave simple. La transformación al modelo relacional exige la creación de una tabla para la relación. La tabla ESTUDIOS, tendrá como columnas los atributos clave del aula, los de asignatura y el atributo clave de estudiante, todos ellos formando la clave primaria y, al mismo tiempo, actuando como claves foráneas de sus respectivas tablas. Además, se incorpora el atributo de relación "hora".

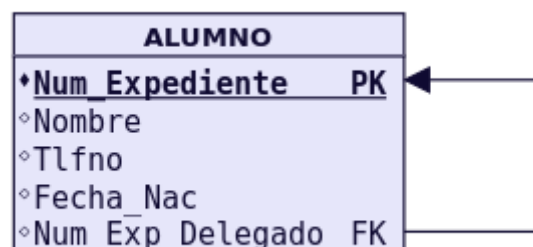


AULAS(Numero,Planta,Situación)  
 ESTUDIANTES(Num-Matricula,Nombre,Dirección)  
 ASIGNATURAS(Nombre,Ciclo,Descripción)  
 ESTUDIOS(Numero,Planta,Num-Matricula,Nombre,Ciclo,Hora)

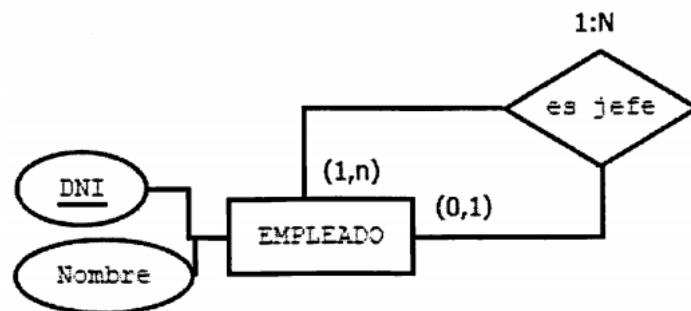
#### 2.5.4.5. Relaciones reflexivas

Relaciones Reflexivas o Recursivas: Generan tabla o no en función de la cardinalidad. Si es 1:1, no genera tabla. En la entidad se introduce dos veces la clave, una como clave principal y otra como clave ajena. Se suele introducir una modificación en el nombre por diferenciarlas. Si es 1:N, se puede generar tabla o no. Si hubiese participación 0 en el lado 1, obligatoriamente se generaría tabla. Si es N:N, la relación genera tabla. En las relaciones reflexivas con cardinalidad N:M, se aplica la regla general para la transformación de relaciones.

Ejemplo: Realicemos el paso a tablas de la relación reflexiva de ALUMNO. Como no tiene participación mínima «0» en el lado 1, no genera tabla. La clave principal de ALUMNOS, volverá a aparecer en ALUMNOS como clave foránea, igual que en cualquier relación 1:N. Ahora bien, como no puede haber dos campos con el mismo nombre en la misma tabla, deberemos cambiar un poco el nombre de la clave principal, para que haga referencia al papel que ocupa como clave foránea.



Entonces, en relaciones 1:N, no tiene porqué crearse una tabla para la relación. Hay que crear una tabla con el nombre de la entidad, añadiendo otra vez la clave con el nombre cambiado:



En este ejemplo, el empleado sólo puede tener un jefe, por tanto, se incorpora el DNI del jefe del empleado (DNIJefe) como clave foránea:

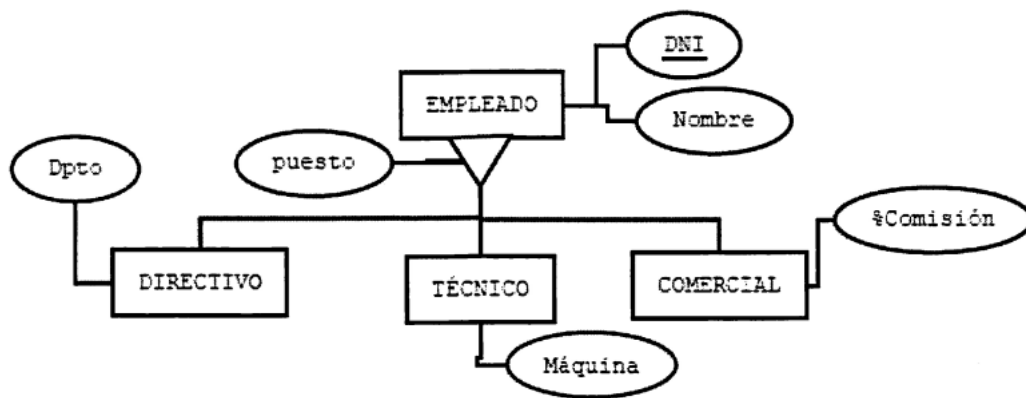
EMPLEADOS(DNI,Nombre,DNIJefe)

#### 2.5.4.6. Jerarquías

Eliminación de las relaciones jerárquicas: Las relaciones jerárquicas son un caso especial. Se pueden dar algunas guías que sirvan de referencia, pero en la mayoría de los casos, va a depender del problema concreto. Estas guías son:

- Se creará una tabla para la entidad supertipo. A no ser que tenga tan pocos atributos que dejarla sea una complicación.
- Si la entidad subtipo no tiene atributos y no está relacionada con ninguna otra entidad, desaparece.
- Si la entidad subtipo tiene algún atributo, se crea una tabla. Si no tiene clave propia, hereda la de la entidad supertipo.
- Si la relación es exclusiva, el atributo que genera la jerarquía se incorpora en la tabla de la entidad supertipo.
- Si se ha creado una tabla por cada una de las entidades subtipo, no es necesario incorporar dicho atributo a la entidad supertipo.

Ejemplo: Para transformar las generalizaciones se puede optar por 4 opciones. Cada opción se adaptará mejor o peor a los diferentes tipos de especialización (Exclusiva, Inclusiva, Total, Parcial).



1. Se puede crear una tabla para la superclase y otras tantas para cada subclase, incorporando el campo clave de la superclase a las tablas de las subclases.

EMPLEADOS(DNI, Nombre, Puesto)  
 DIRECTIVOS(DNI, Dpto)  
 TECNICOS(DNI, Máquinas)  
 COMERCIALES(DNI, Comisión)

2. Se puede crear una tabla para cada subclase incorporando todos los atributos de la clase padre, y no crear una tabla para la superclase.

DIRECTIVOS(DNI, Nombre, Puesto, Dpto)  
 TÉCNICOS(DNI, Nombre, Puesto, Máquinas)  
 COMERCIALES (DNI, Nombre, Puesto, Comisión)

3. Se puede crear una sola tabla para la superclase, incorporando los atributos de todas las subclases y añadir, para distinguir el tipo de la superclase, un campo llamado "tipo", que contendrá el tipo de subclase al que representa cada tupla. Este tipo de opción se adapta muy bien a las especializaciones exclusivas.

EMPLEADOS(DNI, Nombre, Puesto, Dpto, Máquinas, Comisión, Tipo)

4. Se puede crear una sola tabla para la superclase como en la opción anterior, pero en lugar de añadir un solo campo "tipo", se añaden varios campos que indiquen si cumple un perfil, de este modo se soportan las especializaciones inclusivas.

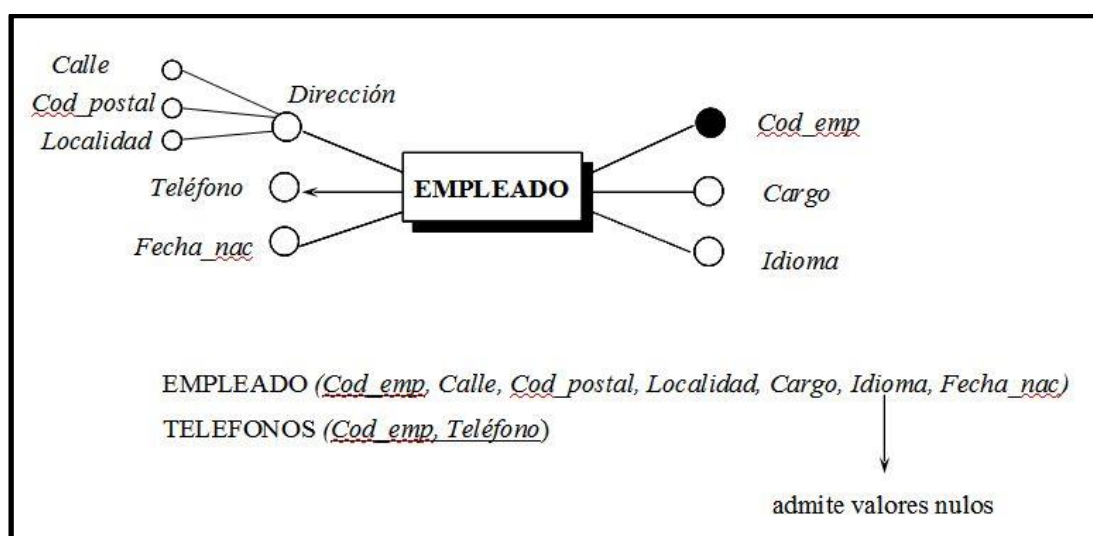
EMPLEADOS(DNI, Nombre, Puesto, Dpto, Máquinas, Comisión, EsDirectivo, EsTécnico, EsComercial)

#### 2.5.4.7. Atributos multivaluados y compuestos

Generalmente se siguen los siguientes pasos, aunque depende del contexto del ejercicio y del diseño que queramos realizar:

- Atributos multivaluados: Se crea una nueva relación formada con la clave primaria de la entidad y el atributo multivaluado, siendo ambos clave primaria de la nueva relación (hay otras posibilidades).
- Atributos compuestos: Se transforma en los atributos simples (campos) que componen el atributo compuesto, desapareciendo este como tal de la relación.

Ejemplo de transformación de una entidad con atributos compuestos y multivaluados:





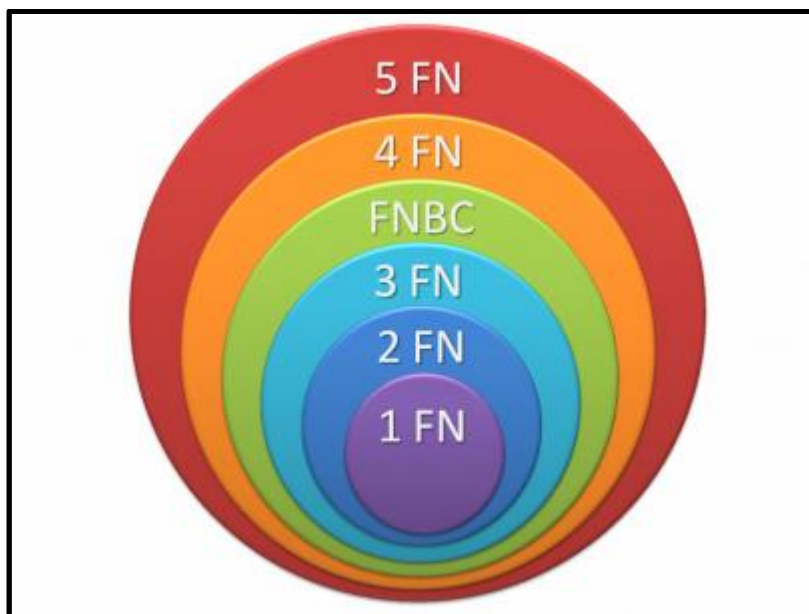
## 2.6. NORMALIZACIÓN

El diseño de una BD relacional se puede realizar aplicando al mundo real, en una primera fase, un modelo como el modelo E/R, a fin de obtener un esquema conceptual; en una segunda fase, se transforma dicho esquema al modelo relacional mediante las correspondientes reglas de transformación. También es posible, aunque quizás menos recomendable, obtener el esquema (modelo) relacional sin realizar ese paso intermedio que es el esquema conceptual (E/R). En ambos casos, es conveniente (obligatorio en el modelo relacional directo) aplicar un conjunto de reglas, conocidas como “Teoría de normalización”, que nos permiten asegurar que un esquema relacional cumple unas ciertas propiedades, evitando:

- La redundancia de los datos: repetición de datos en un sistema.
- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

En la práctica, si la BD se ha diseñado haciendo uso de modelos semánticos como el modelo E/R no suele ser necesaria la normalización. Por otro lado **si nos proporcionan una base de datos creada sin realizar un diseño previo, es muy probable que necesitemos normalizar.**

En la teoría de bases de datos relacionales, las **formas normales (FN)** proporcionan los criterios para determinar el grado de vulnerabilidad de una tabla a inconsistencias y anomalías lógicas. Cuanto más alta sea la forma normal aplicable a una tabla, menos vulnerable será a inconsistencias y anomalías. Edgar F. Codd originalmente definió las tres primeras formas normales (**1FN, 2FN, y 3FN**) en 1970. Estas formas normales se han resumido como requiriendo que **todos los atributos sean atómicos, dependan de la clave completa y en forma directa (no transitiva)**. La forma normal de Boyce-Codd (**FNBC**) fue introducida en 1974 por los dos autores que aparecen en su denominación. Las cuarta y quinta formas normales (**4FN y 5FN**) se ocupan específicamente de la representación de las relaciones muchos a muchos y uno a muchos entre los atributos y fueron introducidas por Fagin en 1977 y 1979 respectivamente. Cada forma normal incluye a las anteriores.



Antes de dar los conceptos de formas normales veamos unas definiciones previas:

- **Dependencia funcional:**  $A \rightarrow B$ , representa que B es funcionalmente dependiente de A. Para **un valor de A** siempre aparece **un valor de B**.
  - Ejemplo: Si A es el D.N.I., y B el Nombre, está claro que para un número de D.N.I, siempre aparece el mismo nombre de titular.
- **Dependencia funcional completa:**  $A \rightarrow B$ , si B depende de A en su totalidad.
  - Ejemplo: Tiene sentido plantearse este tipo de dependencia cuando A está compuesto por más de un atributo. Por ejemplo, supongamos que A corresponde al atributo compuesto: D.N.I.\_Empleado + Cod.\_Dpto. y B es Nombre\_Dpto. En este caso B depende del Cod\_Dpto., pero no del D.N.I.\_Empleado. Por tanto no habría dependencia funcional completa.
- **Dependencia transitiva:**  $A \rightarrow B \rightarrow C$ . Si  $A \rightarrow B$  y  $B \rightarrow C$ , Entonces decimos que C depende de forma transitiva de A.
  - Ejemplo: Sea A el D.N.I. de un alumno, B la localidad en la que vive y C la provincia. Es un caso de dependencia transitiva  $A \rightarrow B \rightarrow C$ .
- **Determinante funcional:** todo atributo, o conjunto de ellos, de los que depende algún otro atributo.
  - Ejemplo: El D.N.I. es un determinante funcional pues atributos como nombre, dirección, localidad, etc., dependen de él.
- **Dependencia multivaluada:**  $A \twoheadrightarrow B$ . Son un tipo de dependencias en las que un determinante funcional no implica un único valor, sino un conjunto de ellos. **Un valor de A** siempre implica **varios valores de B**.
  - Ejemplo: CursoBachillerato  $\twoheadrightarrow$  Modalidad. Para primer curso siempre va a aparecer en el campo Modalidad uno de los siguientes valores: Ciencias, Humanidades/Ciencias Sociales o Artes. Igual para segundo curso.

### 2.6.1. Primera Forma Normal: 1FN

Una Relación está en 1FN si y sólo si cada atributo es atómico.

Ejemplo: Supongamos que tenemos la siguiente tabla con datos de alumnado de un centro de enseñanza secundaria.

Alumnos							
DNI	Nombre	Curso	FechaMatrícula	Tutor	Localidad Alumno	ProvinciaAlumno	Teléfonos
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla	660111222
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla	660222333 660333444 660444555
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla	
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba	
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba	661000111 661000222

Como se puede observar, esta tabla no está en 1FN puesto que el campo Teléfonos contiene varios datos dentro de una misma celda y por tanto no es un campo cuyos valores sean atómicos. La solución sería la siguiente (separar las tablas):

Alumnos						
DNI	Nombre	Curso	FechaMatrícula	Tutor	Localidad Alumno	ProvinciaAlumno
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba

Teléfonos	
DNI	Teléfono
11111111A	660111222
22222222B	660222333
22222222B	660333444
22222222B	660444555
55555555E	661000111
55555555E	661000222

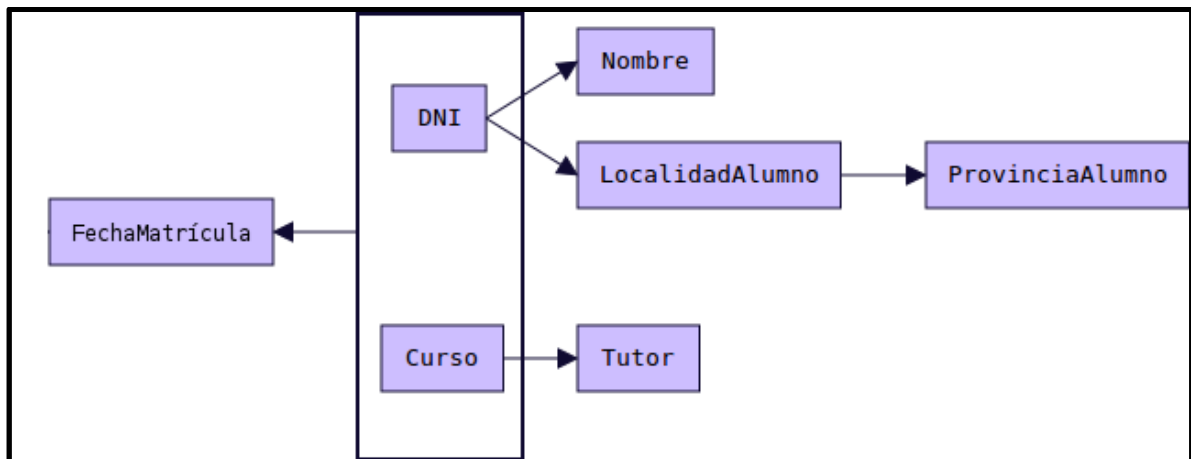
### 2.6.2. Segunda Forma Normal: 2FN

Una Relación está en 2FN si y sólo si está en 1FN y todos los atributos que no forman parte de la Clave Principal tienen dependencia funcional completa de ella.

Ejemplo: Seguimos con el ejemplo anterior. Trabajaremos con la siguiente tabla:

Alumnos						
DNI	Nombre	Curso	FechaMatrícula	Tutor	LocalidadAlumno	ProvinciaAlumno
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba

Vamos a examinar las dependencias funcionales. El gráfico que las representa es el siguiente:



- Siempre que aparece un DNI aparecerá el Nombre correspondiente y la LocalidadAlumno correspondiente. Por tanto  $DNI \rightarrow Nombre$  y  $DNI \rightarrow LocalidadAlumno$ . Por otro lado siempre que aparece un Curso aparecerá el Tutor correspondiente. Por tanto  $Curso \rightarrow Tutor$ . Los atributos Nombre y LocalidadAlumno no dependen funcionalmente de Curso, y el atributo Tutor no depende funcionalmente de DNI.
- El único atributo que sí depende de forma completa de la clave compuesta DNI y Curso es FechaMatrícula:  $(DNI, Curso) \rightarrow FechaMatrícula$ .

A la hora de establecer la Clave Primaria de una tabla debemos escoger un atributo o conjunto de ellos de los que dependan funcionalmente el resto de atributos. Además debe ser una dependencia funcional completa. Si escogemos DNI como clave primaria, tenemos un atributo (Tutor) que no depende funcionalmente de él. Si escogemos Curso como clave primaria, tenemos otros atributos que no dependen de él.

Si escogemos la combinación (DNI, Curso) como clave primaria, entonces sí tenemos todo el resto de atributos con dependencia funcional respecto a esta clave. Pero es una dependencia parcial, no total (salvo FechaMatrícula, donde sí existe dependencia completa). Por tanto esta tabla no está en 2FN. La solución sería la siguiente (dividir en 3 tablas):

Alumnos			
DNI	Nombre	Localidad	Provincia
11111111A	Eva	Écija	Sevilla
22222222B	Ana	Écija	Sevilla
33333333C	Susana	El Villar	Córdoba
44444444D	Juan	El Villar	Córdoba
55555555E	José	Écija	Sevilla

Matrículas		
DNI	Curso	FechaMatrícula
11111111A	1ESO-A	01-Julio-2016
22222222B	1ESO-A	09-Julio-2016
33333333C	1ESO-B	11-Julio-2016
44444444D	2ESO-A	05-Julio-2016
55555555E	2ESO-A	02-Julio-2016

Cursos	
Curso	Tutor
1ESO-A	Isabel
1ESO-B	Roberto
2ESO-A	Federico

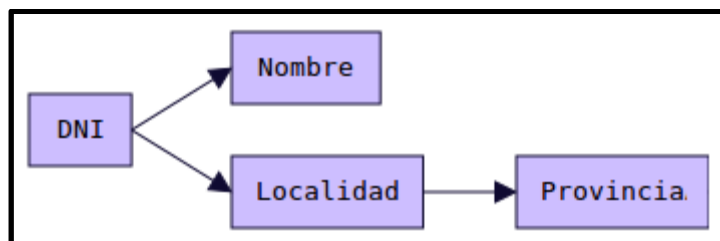
### 2.6.3. Tercera Forma Normal: 3FN

Una Relación esta en 3FN si y sólo si está en 2FN y no existen dependencias transitivas. Todas las dependencias funcionales deben ser respecto a la clave principal.

Ejemplo: Seguimos con el ejemplo anterior. Trabajaremos con la siguiente tabla:

Alumnos			
DNI	Nombre	Localidad	Provincia
11111111A	Eva	Écija	Sevilla
22222222B	Ana	Écija	Sevilla
33333333C	Susana	El Villar	Córdoba
44444444D	Juan	El Villar	Córdoba
55555555E	José	Écija	Sevilla

Las dependencias funcionales existentes son las siguientes. Como podemos observar existe una dependencia funcional transitiva: DNI → Localidad → Provincia



Para que la tabla esté en 3FN, no pueden existir dependencias funcionales transitivas. Para solucionar el problema deberemos crear una nueva tabla. El resultado es:

Alumnos		
DNI	Nombre	Localidad
11111111A	Eva	Écija
22222222B	Ana	Écija
33333333C	Susana	El Villar
44444444D	Juan	El Villar
55555555E	José	Écija

Localidades	
Localidad	Provincia
Écija	Sevilla
El Villar	Córdoba

### RESULTADO FINAL

Alumnos		
DNI	Nombre	Localidad
11111111A	Eva	Écija
22222222B	Ana	Écija
33333333C	Susana	El Villar
44444444D	Juan	El Villar
55555555E	José	Écija

Localidades	
Localidad	Provincia
Écija	Sevilla
El Villar	Córdoba

Teléfonos	
DNI	Teléfono
11111111A	660111222
22222222B	660222333
22222222B	660333444
22222222B	660444555
55555555E	661000111
55555555E	661000222

Matrículas		
DNI	Curso	FechaMatrícula
11111111A	1ESO-A	01-Julio-2016
22222222B	1ESO-A	09-Julio-2016
33333333C	1ESO-B	11-Julio-2016
44444444D	2ESO-A	05-Julio-2016
55555555E	2ESO-A	02-Julio-2016

Cursos	
Curso	Tutor
1ESO-A	Isabel
1ESO-B	Roberto
2ESO-A	Federico



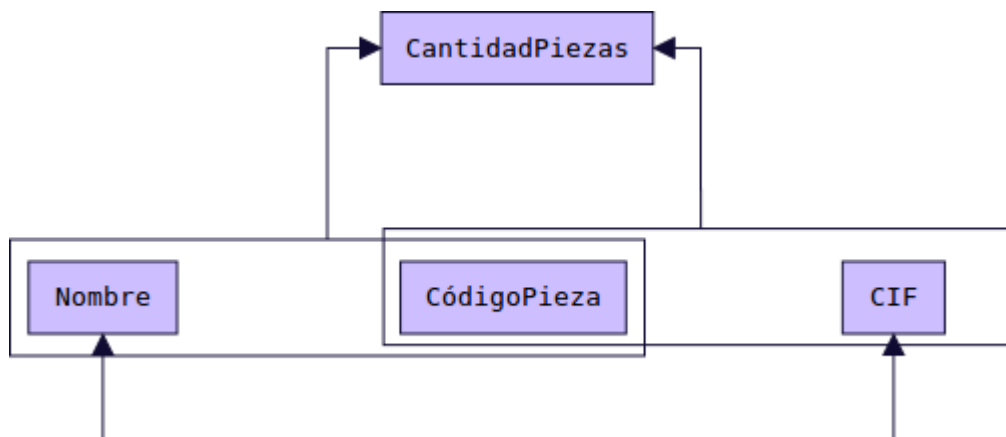
#### 2.6.4. Forma Normal de Boyce-Codd: FNBC

Una Relación está en FNBC si está en 3FN y no existe solapamiento de claves candidatas. Solamente hemos de tener en cuenta esta forma normal cuando tenemos varias claves candidatas compuestas y existe solapamiento entre ellas. Pocas veces se da este caso.

Ejemplo: Tenemos una tabla con información de proveedores, códigos de piezas y cantidades de esa pieza que proporcionan los proveedores. Cada proveedor tiene un nombre único. Los datos son:

Suministros			
CIF	Nombre	CódigoPieza	CantidadPiezas
S-11111111A	Ferroman	1	10
B-22222222B	Ferrotex	1	7
M-33333333C	Ferropet	3	4
S-11111111A	Ferroman	2	20
S-11111111A	Ferroman	3	15
B-22222222B	Ferrotex	2	8
B-22222222B	Ferrotex	3	4

El gráfico de dependencias funcionales es el siguiente:



El atributo CantidadPiezas tiene dependencia funcional de dos claves candidatas compuestas, que son:

- (NombreProveedor, CódigoPieza)
- (CIFProveedor, CódigoPieza)

Existe también una dependencia funcional en doble sentido (que no nos afecta): NombreProveedor <-> CIFProveedor.

Para esta tabla existe un solapamiento de 2 claves candidatas compuestas. Para evitar el solapamiento de claves candidatas dividimos la tabla. La solución es:

Proveedores	
CIF	Nombre
S-11111111A	Ferroman
B-22222222B	Ferrotex
M-33333333C	Ferropet

Suministros		
CIF	CódigoPieza	CantidadPiezas
S-11111111A	1	10
B-22222222B	1	7
M-33333333C	3	4
S-11111111A	2	20
S-11111111A	3	15
B-22222222B	2	8
B-22222222B	3	4

#### 2.6.5. Cuarta Forma Normal: 4FN

Una Relación está en 4FN si y sólo si está en 3FN (o FNBC) y las únicas dependencias multivaluadas son aquellas que dependen de las claves candidatas.

Ejemplo: Tenemos una tabla con la información de nuestros alumnos y las asignaturas que cursan así como los deportes que practican.

Alumnado		
Estudiante	Asignatura	Deporte
11111111A	Matemáticas, Lengua	Baloncesto, Natación
22222222B	Matemáticas	Fútbol, Natación

Alumnado		
Estudiante	Asignatura	Deporte
11111111A	Matemáticas	Natación
11111111A	Matemáticas	Baloncesto
11111111A	Lengua	Natación
11111111A	Lengua	Baloncesto
22222222B	Matemáticas	Fútbol
22222222B	Matemáticas	Natación

Para normalizar esta tabla, debemos darnos cuenta que la oferta de asignaturas está compuesta por un conjunto de valores limitado. Igual sucede con los deportes. Por tanto existen dos dependencias multivaluadas:

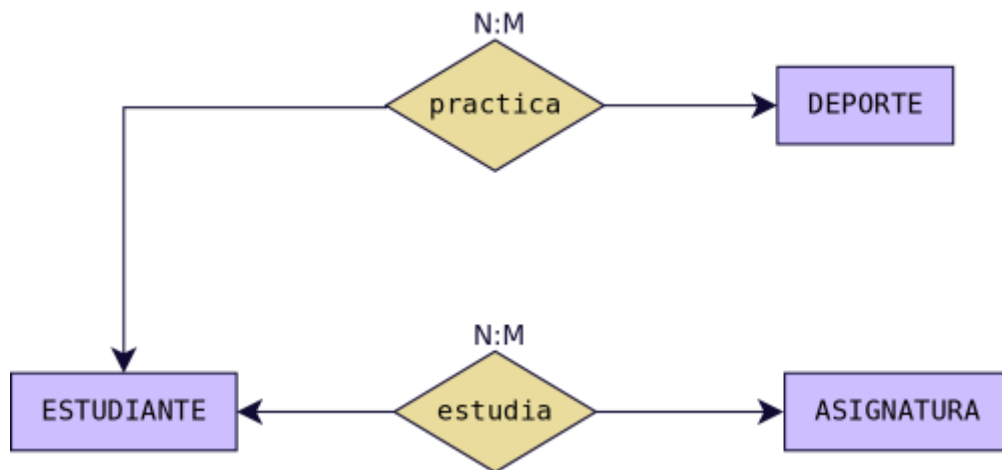
- Estudiante →→ Asignatura
- Estudiante →→ Deporte

Por otro lado no existe ninguna dependencia entre la asignatura cursada y el deporte practicado. Para normalizar a 4FN creamos 2 tablas:

EstudiaAsignatura	
Estudiante	Asignatura
11111111A	Matemáticas
11111111A	Lengua
22222222B	Matemáticas

PracticaDeporte	
Estudiante	Deporte
11111111A	Natación
11111111A	Baloncesto
22222222B	Fútbol
22222222B	Natación

Diagrama E/R equivalente



### 2.6.6. Quinta Forma Normal: 5FN

La quinta forma normal (5FN), es una generalización de la anterior. También conocida como forma normal de proyección-únión (PJ/NF). Una tabla se dice que está en 5NF si y sólo si está en 4NF y cada dependencia de unión (join) en ella es implicada por las claves candidatas.

Ejemplo: Tenemos una tabla con varios proveedores que nos proporcionan piezas para distintos proyectos. Asumimos que un Proveedor suministra ciertas Piezas en particular, un Proyecto usa ciertas Piezas, y un Proyecto es suplido por ciertos Proveedores, entonces tenemos las siguientes dependencias multivaluadas:

- Proveedor  $\twoheadrightarrow$  Pieza
- Pieza  $\twoheadrightarrow$  Proyecto
- Proyecto  $\twoheadrightarrow$  Proveedor

Se puede observar cómo se produce un ciclo:

- Proveedor  $\twoheadrightarrow$  Pieza  $\twoheadrightarrow$  Proyecto  $\twoheadrightarrow$  Proveedor (nuevamente)

Suministros		
Proveedor	Pieza	Proyecto
E1, E4, E6	PI3, PI6	PR2, PR4
E2, E5	PI1, PI2	PR1, PR3
E3, E7	PI4, PI5	PR5, PR6

Suministros		
Proveedor	Pieza	Proyecto
E1	PI3	PR2

Suministros		
Proveedor	Pieza	Proyecto
E1	PI3	PR4
E1	PI6	PR2
E1	PI6	PR4
E4	PI3	PR2
E4	PI3	PR4
E4	PI6	PR2
E4	PI6	PR4
E6	PI3	PR2
E6	PI3	PR4
E6	PI6	PR2
E6	PI6	PR4
E2	PI1	PR1
E2	PI1	PR3
E2	PI2	PR1
E2	PI2	PR3
E5	PI1	PR1
E5	PI1	PR3
E5	PI2	PR1
E5	PI2	PR3
E3	PI4	PR5
E3	PI4	PR6
E3	PI5	PR5
E3	PI5	PR6
E7	PI4	PR5
E7	PI4	PR6
E7	PI5	PR5
E7	PI5	PR6

Descomponemos la tabla en 3 tablas nuevas: Proveedor-Pieza, Pieza-Proyecto, Proyecto-Proveedor.

Proveedor-Pieza	
Proveedor	Pieza
E1	PI3
E1	PI6
E4	PI3
E4	PI6
E6	PI3
E6	PI6
E2	PI1
E2	PI2
E5	PI1
E5	PI2
E3	PI4
E3	PI5
E7	PI4
E7	PI5

Pieza-Proyecto	
Pieza	Proyecto
PI3	PR2
PI3	PR4
PI6	PR2
PI6	PR4
PI1	PR1
PI1	PR3
PI2	PR1
PI2	PR3
PI4	PR5
PI4	PR6
PI5	PR5
PI5	PR6

Proyecto-Proveedor	
Proyecto	Proveedor
PR2	E1
PR4	E1
PR2	E4
PR4	E4
PR2	E6
PR4	E6
PR1	E2
PR3	E2
PR1	E5
PR3	E5
PR5	E3
PR6	E3
PR5	E7
PR6	E7

El producto natural de estas 3 tablas nos da la tabla original. Proveedor-Pieza  $\times$  Pieza-Proyecto  $\times$  Proyecto-Proveedor = Suministros

Diagrama E/R equivalente

