

UNIDAD 2

BASES DE DATOS RELACIONALES

ÍNDICE

1. INTRODUCCIÓN	1
2. MODELO ENTIDAD/RELACIÓN (E/R)	1
2.1. Entidades	1
2.2. Atributos	3
2.3. Interrelaciones	7
2.4. Claves	9
2.5. Cardinalidad	10
2.6. Atributos de las interrelaciones	11
2.7. Tipos de interrelaciones con dependencia	11
3. MODELO ENTIDAD/RELACIÓN EXTENDIDO	14
3.1. Superclase y subclase	14
3.2. Herencia de atributos	15
3.3. Especialización y generalización	15
3.4. Características de la especialización y de la generalización	17
4. MODELO RELACIONAL	21
4.1. Elementos del modelo relacional	21
4.2. Terminología empleada en el modelo relacional	21
4.3. Requisitos de una tabla en una base de datos relacional	23
4.4. Vistas	23
4.5. Reglas de integridad	24
4.6. Transformación del modelo E/R al modelo relacional	26
4.6.1. Transformación de entidades	27
4.6.2. Transformación de interrelaciones	29
4.6.3. Transformación de atributos multivaluados	32
4.6.4. Transformación de especializaciones/generalizaciones	33
4.6.5. Transformación de atributos derivados	35
5. NORMALIZACIÓN: IDEAS BÁSICAS	35
5.1. Dependencias funcionales	36
5.2. Primera forma normal (1FN)	37
5.3. Segunda forma normal (2FN)	38
5.4. Tercera forma normal (3FN)	40
5.5. Resumen de las tres primeras formas normales	42

1. INTRODUCCIÓN

En el proceso de creación de una base de datos hay dos fases bien diferenciadas: el diseño y la posterior implementación. La fase de diseño consiste en analizar la realidad que se va a representar, identificar los datos que intervienen en dicha realidad y las relaciones que existen entre ellos, y estructurar toda esa información aplicando un modelo conceptual (por ejemplo, el modelo Entidad/Relación), independiente del SGBD en el que posteriormente se vaya a realizar la implementación.

Esta fase de elaboración del modelo conceptual también debe ser independiente de la fase de diseño lógico, en la cual se ha de obtener un esquema de la base de datos que responda a la estructura específica (por ejemplo, relacional) del SGBD que se aplique en cada caso concreto.

A lo largo de esta unidad, trataremos los aspectos más importantes relacionados con estos dos modelos (modelo Entidad/Relación y modelo Relacional) para el diseño de una base de datos.

2. MODELO ENTIDAD/RELACIÓN (E/R)

Fue propuesto por Peter Chen a mediados de los años setenta. Es el modelo conceptual más ampliamente conocido. A lo largo de los años ha ido experimentado una serie de extensiones, lo que ha originado un medio muy potente para la representación de los datos correspondiente a un problema.

Permite realizar el diseño conceptual de una base de datos. Es una representación lingüística y gráfica de los objetos que forman parte del mundo real.

Conceptos básicos de este modelo son: **entidades**, **interrelaciones**, **atributos** y **claves**.

2.1 Entidades

Entidad es "una persona, lugar, cosa, concepto o suceso, real o abstracto, de interés para la empresa" (ANSI - 1977). Es aquel objeto acerca del cual queremos almacenar información

en la base de datos, que es distinguible de los demás.

Una entidad puede ser un objeto con existencia física (por ejemplo, “PERSONA”, “LIBRO”, “AUTOMÓVIL”, “EMPLEADO”) o un objeto con existencia conceptual (por ejemplo, “COMPAÑÍA”, “CURSO UNIVERSITARIO”, “PRÉSTAMO”).

Una base de datos contiene grupos de entidades similares. Por ejemplo, una empresa con miles de empleados puede querer almacenar información similar de cada uno de sus empleados. Dichas entidades de empleados comparten los mismos atributos, pero cada entidad tiene sus propios valores de los atributos.

Se llama **tipo de entidad** a la estructura genérica que es compartida por un conjunto de entidades y **ocurrencia de una entidad** a cada una de las realizaciones concretas de ese tipo de entidad. Por ejemplo, el tipo de entidad AUTOR se refiere a la estructura que nos describe las características de los autores como una abstracción, mientras que una ocurrencia de autor sería cada uno de los autores en concreto, pongamos “Antonio Gala”.

Conjunto de entidades es un grupo de entidades del mismo tipo.

Para aclarar un poco esta terminología, veamos un ejemplo: “conjunto de entidades” CLIENTE sería el conjunto de personas que tienen una cuenta en un determinado banco; “tipo de entidad” CLIENTE sería los datos “NIF”, “nombre”, “calle”, “ciudad”... y una “ocurrencia” sería “11111111A”, “José Gómez”, “C/Sol”, “Pontevedra”.

Una entidad se representa gráficamente por un rectángulo etiquetado con el nombre del tipo de entidad:

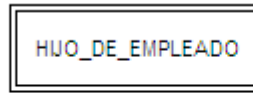


Ejemplo de representación de entidades

Entidades fuertes y débiles

- Entidades fuertes son aquellas que tienen existencia por sí mismas, es decir, no dependen de otra entidad para su existencia. Por ejemplo, la entidad EMPLEADO.
- Entidades débiles son las que dependen de otra entidad para su existencia. Por ejemplo, la entidad HIJO_DE_EMPLEADO depende de la entidad EMPLEADO, y la desaparición de la base de datos de una ocurrencia de la entidad “EMPLEADO” hace que desaparezcan también todas las ocurrencias de la entidad “HIJO_DE_EMPLEADO” asociadas.

Los tipos de entidad débil se representan con dos rectángulos concéntricos con el nombre en su interior:



Ejemplo de representación de entidad débil

A su vez, una entidad débil puede ser catalogada como:

- Débil por identificación: es una entidad débil que, además, no puede ser identificada a no ser que se indique también la entidad fuerte a la que está asociada.

Por ejemplo, sean las entidades CINE y SALA. Las diferentes salas de un cine suelen llevar el número de sala para su identificación dentro del cine al que pertenecen. Pero debe indicarse también el identificador de dicho cine para que cada sala sea identificada sin ambigüedades (por ejemplo, “sala 1 del cine A”, “sala 1 del cine B”, etc.).

- Débil por existencia: es una entidad débil que sí puede ser identificada sin necesidad de identificar la entidad fuerte a la que está asociada.

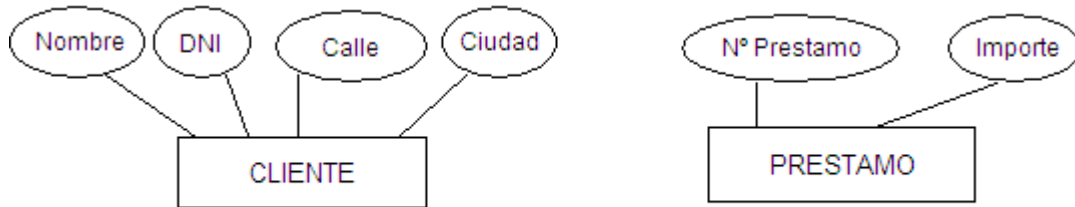
Por ejemplo, sea el tipo de entidad SOCIO, que representa el conjunto de socios de un club deportivo en el que se llevan a cabo diferentes actividades: caza, pesca, etc... Se identifica por “Num_socio”. Sea también el tipo de entidad LICENCIA, que representa el conjunto de licencias que tiene un socio para hacer diferentes actividades. LICENCIA no existirá en nuestra base de datos si no se conoce el socio al que pertenece, pero se podría identificar por “Num_licencia”, sin añadir el identificador del socio.

Una debilidad por identificación implica una debilidad por existencia, pero no a la inversa.

2.2 Atributos

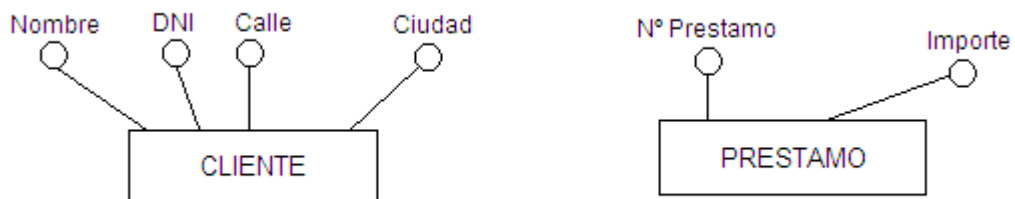
Describen las **propiedades** que tiene un tipo de entidad. Por ejemplo, una entidad EMPLEADO puede describirse por su nombre, edad, dirección, salario y puesto de trabajo, que serán sus atributos. Una ocurrencia particular tendrá un valor para cada uno de sus atributos.

La figura siguiente muestra dos tipos de entidades y sus atributos. La entidad CLIENTE tiene cuatro atributos: “Nombre”, “DNI”, “Calle”, “Ciudad”, y la entidad PRESTAMO tiene dos atributos: “Numero de préstamo” e “Importe”:



Ejemplo de representación de atributos

En ocasiones, se emplea otra simbología:



Otra notación de representación de atributos

Tipos de atributos

En el modelo E/R se manejan varios tipos distintos de atributos: simples o compuestos; monovaluados o multivaluados; elementales o derivados y opcionales u obligatorios.

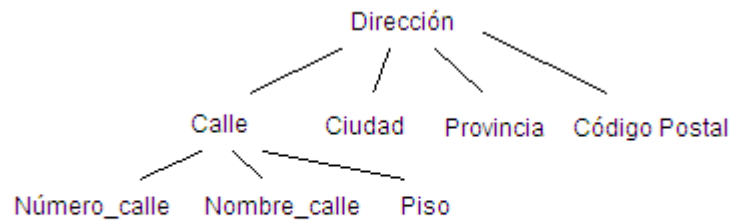
a) Atributos simples o compuestos

Los **atributos compuestos** se pueden dividir en componentes más pequeños, que representan atributos más básicos con su propio significado independiente.

Por ejemplo, el atributo “Dirección” de una entidad CLIENTE se puede subdividir en “Calle”, “Ciudad”, “Provincia” y “CP”. El atributo “Nombre_Cliente” se puede dividir en “Nombre”, “Primer apellido”, “Segundo apellido”.

Los atributos no divisibles se denominan **atributos simples** o **atómicos**.

Los atributos compuestos pueden formar una jerarquía; por ejemplo, “Calle” aún se podría subdividir en tres atributos simples, “Número_calle”, “Nombre_calle”, “Piso”, como se aprecia en la figura siguiente:

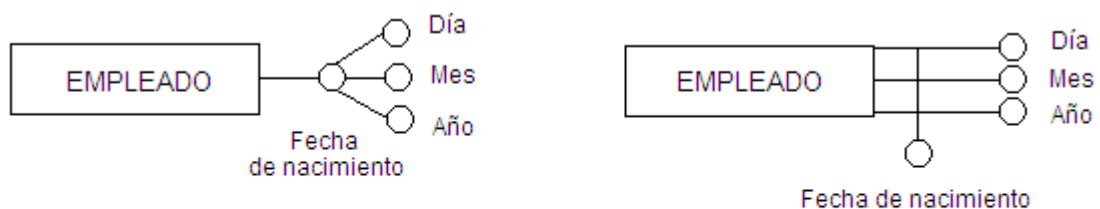


Ejemplo de jerarquía en atributo compuesto

El valor de un atributo compuesto es la concatenación de los valores de los atributos simples que lo constituyen.

Los atributos compuestos son útiles para situaciones en las que un usuario, en ocasiones, hace referencia al atributo compuesto como una unidad pero, otras veces, se refiere específicamente a sus componentes. Si sólo se hace referencia al atributo compuesto como un todo, no hay necesidad de subdividirlo en sus atributos componentes. Por ejemplo, si no hay necesidad de referirse a los componentes individuales de una dirección (CP, Calle, etc.), la dirección completa se designa como atributo simple.

Se admiten diferentes formas de representación, como las que se muestran a continuación para la entidad EMPLEADO y su atributo compuesto “Fecha de nacimiento”:



Ejemplos de simbología para atributos compuestos

b) Atributos monovaluados o multivaluados

En su mayoría, los atributos tienen un solo valor para una ocurrencia en particular, y reciben el calificativo de **monovaluados**. Por ejemplo, “Edad” es un atributo

monovaluado de la entidad PERSONA.

Hay casos en los que un atributo puede tener un conjunto de valores para la misma ocurrencia. Estos atributos se llaman **multivaluados**. Por ejemplo, un atributo "Titulación" para la entidad EMPLEADO es multivaluado porque un empleado puede tener una titulación, dos, tres, etc. Los atributos multivaluados pueden tener límites inferior y superior en cuanto al número de valores para una entidad individual. Por ejemplo, el atributo "Teléfono" de la entidad CLIENTE se puede limitar entre 1 y 3 valores si, como máximo, deseamos almacenar tres números telefónicos.

Un atributo multivaluado se puede representar con una punta de flecha:



Ejemplo de representación de atributo multivaluado

c) Atributos elementales o derivados

Un atributo cuyo valor no se puede calcular en base a ningún otro atributo es **elemental**. Por ejemplo, la fecha de nacimiento.

Un atributo cuyo valor puede ser determinado a partir de valores de otros atributos se dice que es **derivado** o **calculado**. Por ejemplo, el atributo "Edad" es un atributo derivado, ya que se puede determinar a partir de la fecha actual y el valor de la fecha de nacimiento de esa persona.

Se puede representar indicando la fórmula o regla de derivación. Por ejemplo:



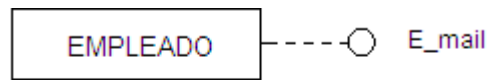
Ejemplo de representación de atributo derivado

d) Atributos opcionales u obligatorios.

Un atributo que no puede tomar valor nulo es **obligatorio**. Por ejemplo, el atributo "DNI" de la entidad EMPLEADO.

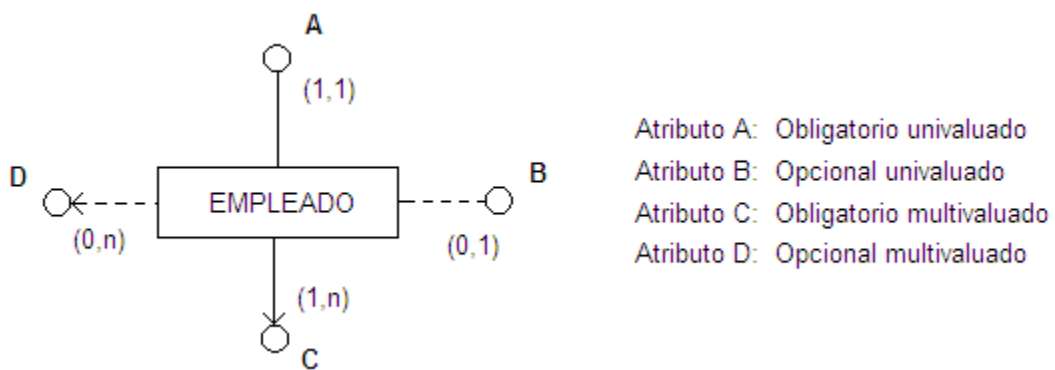
Un atributo que sí pueda quedar sin determinar (tomar valor nulo), se dice que es **opcional**. Por ejemplo, el atributo "E-mail" debería ser opcional.

Se puede representar mediante una línea discontinua, como se muestra en la figura de la página siguiente:



Ejemplo de representación de atributo opcional

A su vez, estas características de los atributos se pueden combinar entre sí, es decir, en un esquema E/R pueden existir atributos multivaluados simples opcionales, multivaluados obligatorios, etc. Por ejemplo:



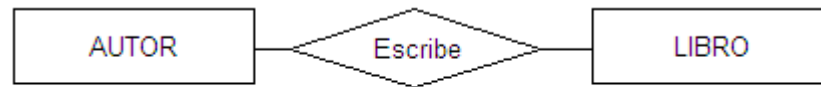
Dominios

En relación con los atributos existe el concepto de dominio. Se llama dominio al conjunto de valores que puede tomar un atributo. Por ejemplo, el atributo “Edad” de la entidad EMPLEADO puede tomar valores en un subconjunto de los números naturales. Normalmente, el modelo E/R no representa los dominios.

2.3 Interrelaciones

Una interrelación (llamada muchas veces “relación”, para mayor brevedad) es una asociación entre dos o más tipos de entidades. Suelen identificarse por los verbos que unen las entidades en la descripción lingüística de los datos. Por ejemplo, “Vende”, para EMPLEADO y COCHE; “Imparte”, para PROFESOR y ASIGNATURA; “Nacido_en”, para PERSONA y CIUDAD, etc. Otras veces se pone el nombre de las entidades: PERSONA-CIUDAD.

Las relaciones se representan por un *rombo* etiquetado con el nombre de la relación, como se muestra en la siguiente figura:



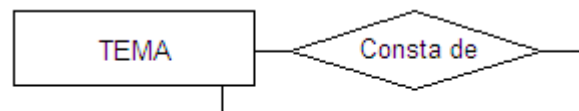
Ejemplo de representación de interrelación

- Se denominan **ocurrencias de relación** a los datos que relacionan la ocurrencia de una entidad con otra de cada una de las entidades restantes que participan en la relación. Por ejemplo, para la relación PERSONA – “Nacida_en” - CIUDAD, una ocurrencia podría ser “Carlos – Nacido_en –Albacete”.

- El **grado** es el número de tipos de entidades que participan en un tipo de relación.

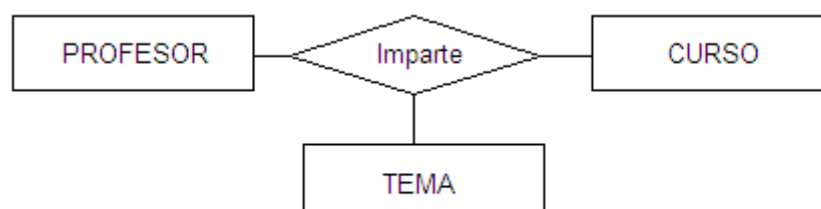
Una relación es **binaria** o de grado 2 cuando establece correspondencia entre dos entidades. Son las más comunes. Un ejemplo se ve en la figura anterior.

Se denomina relación **recursiva** o **reflexiva** a la relación que asocia una entidad consigo misma. Por ejemplo, la figura siguiente:



Ejemplo de interrelación recursiva

Se denomina relación de **grado N** a aquella que asocia más de dos entidades. Por ejemplo, la figura siguiente muestra un ejemplo de relación ternaria entre un profesor, y los temas y cursos que imparte:



Ejemplo de interrelación de grado 3

2.4 Claves

Una restricción importante que impone el modelo E/R es que cada entidad tiene que tener un atributo o atributos que permita distinguirla dentro del conjunto de entidades. Esto es lo que se llama *clave* y podemos distinguir los siguientes tipos:

- **Superclave:** es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única a una entidad en el conjunto de entidades.

Por ejemplo, el atributo “DNI” del conjunto de entidades CLIENTE es suficiente para distinguir una entidad cliente de las otras. Así, “DNI” es una superclave. Análogamente, la combinación de “Nombre-Cliente” y “DNI” es una superclave del conjunto de entidades CLIENTE.

El atributo “Nombre-Cliente” no es una superclave, porque varias personas podrían tener el mismo nombre.

El concepto de superclave no es suficiente ya que, como se ha visto, una superclave puede contener atributos innecesarios.

- **Clave candidata:** Después de ver el concepto de superclave podemos afirmar que, si K es una superclave, entonces también lo es cualquier “superconjunto” de K. No obstante, lo normal es que interesen las superclaves tales que los subconjuntos de ellas no sean también superclave. Estas superclaves mínimas se llaman claves candidatas.

Es posible que existan varias claves candidatas. Supongamos que una combinación de “Nombre-cliente” y “Dirección-cliente” es suficiente para distinguir los diferentes clientes. Entonces, los conjuntos {DNI} y {Nombre-cliente, Dirección-cliente} son claves candidatas. Sin embargo, aunque los atributos “DNI” y “Nombre-cliente” juntos puedan distinguir entidades CLIENTE, su combinación no forma una clave candidata, ya que el atributo “DNI”, por sí solo, ya es una clave candidata.

- **Clave primaria:** es una clave candidata que es elegida por el diseñador de la base de datos como elemento para identificar las entidades. Las restantes claves candidatas pasan a ser entonces **claves alternativas**.

El nombre del atributo (o atributos) que forman la clave primaria se subrayan, o bien se somborean, como se representa a continuación:



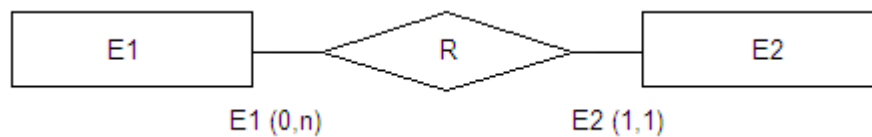
Representación de claves primarias

2.5 Cardinalidad

Se habla de cardinalidad de dos formas:

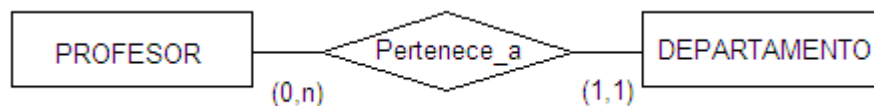
- **Cardinalidad de una entidad:** número mínimo y máximo de ocurrencias de un tipo de entidad que pueden estar relacionadas con una ocurrencia del otro (u otros) tipo de entidad que participa en la relación.

Por ejemplo, sean E1 y E2 dos entidades asociadas por una relación R:



- **E1(0,n)** significa que una ocurrencia de E2 puede estar relacionada con 0, 1, 2... n ocurrencias de E1.
- **E2(1,1)** significa que una ocurrencia de E1 está relacionada con una y sólo una ocurrencia de E2

Supongamos el siguiente esquema:

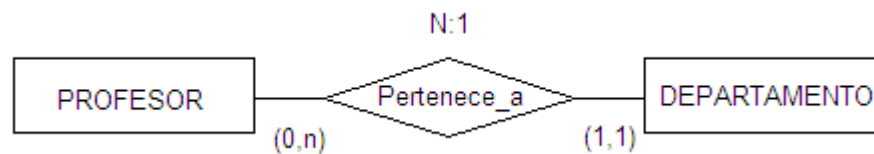


- Para la entidad PROFESOR, la cardinalidad (0,n) significa que un determinado departamento puede estar integrado por 0, 1, 2... n profesores (el valor 0 se podría justificar si en nuestro entorno educativo admitimos la existencia de

departamentos de nueva creación que, por tanto, aún no tienen asignados profesores).

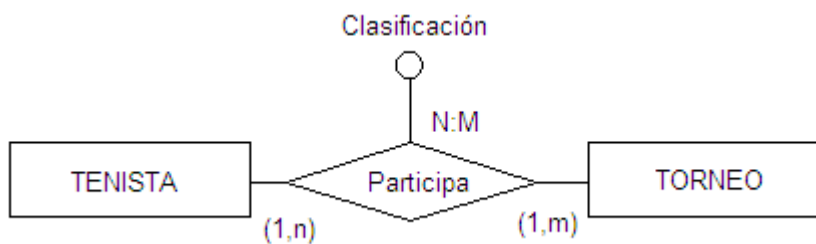
- Para la entidad DEPARTAMENTO, la cardinalidad (1,1) significa que un determinado profesor pertenece a uno y sólo un departamento.
- **Cardinalidad de una interrelación:** se obtiene de las cardinalidades máximas de cada entidad.

En el ejemplo anterior, la cardinalidad de la relación “Pertenece_a” es (N:1), lo que se podría interpretar como “N profesores pertenecen a 1 departamento”.



2.6 Atributos de las interrelaciones

Las interrelaciones pueden tener atributos igual que si fueran entidades. Aportan información a la relación entre dos tipos de entidades. Por ejemplo, si queremos registrar la clasificación que ha obtenido un tenista en un torneo, el atributo “Clasificación” no aporta información del tenista en sí, ni tampoco del torneo, sino que corresponde a un tenista en un determinado torneo:



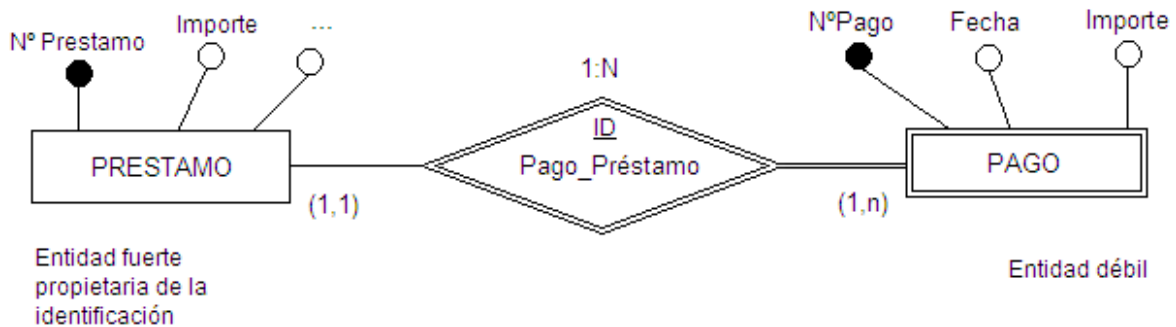
2.7 Tipos de interrelaciones con dependencia

Al igual que existen entidades fuertes y débiles, también existen tipos de relaciones fuertes y débiles, según estén asociando dos tipos de entidades fuertes o un tipo de entidad débil con otro tipo de entidad. En los diagramas E/R, los tipos de entidades débiles y la

correspondiente relación se distinguen rodeando los rectángulos y rombos con líneas dobles.

Es interesante distinguir, dentro del tipo de relación débil, la dependencia en existencia y la dependencia en identificación:

Ejemplo 1: Dependencia en identificación



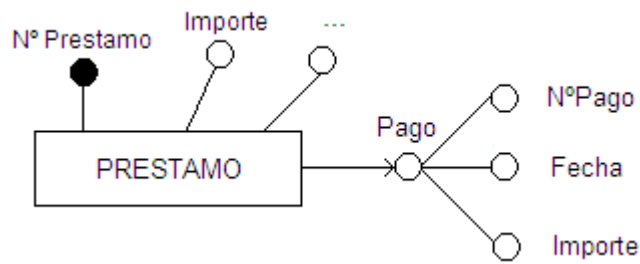
El tipo de entidad PRÉSTAMO representa el conjunto de préstamos de un banco y el tipo de entidad PAGO representa los pagos que se realizan de los préstamos. PAGO tiene una dependencia en identificación respecto a la entidad PRESTAMO pues no existirá ni se podrá identificar si no se conoce el préstamo al que está asociado.

Si existe dependencia en identificación, el rombo que representa la interrelación suele ir etiquetado con ID.

Los tipos de entidades débiles tienen una **clave parcial**, que es el conjunto de atributos que pueden identificar de manera única las entidades débiles relacionadas con la misma entidad propietaria. Cuando se emplea el símbolo de la elipse para representar los atributos, la clave parcial se subraya con una línea punteada o interrumpida. En el ejemplo, la entidad PAGO tiene como clave parcial al atributo “N°Pago”.

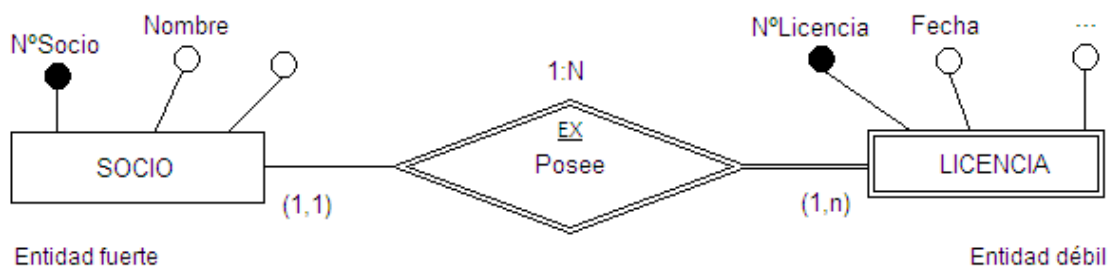
La **clave primaria** de un tipo de entidad débil se forma mediante la clave primaria del tipo de entidad fuerte más la clave parcial del tipo de entidad débil. En el ejemplo, la clave primaria de la entidad PAGO está formada por los atributos “N°Préstamo” y “N°Pago”, conjuntamente.

En ocasiones, los tipos de entidades débiles se representan en forma de atributos compuestos multivaluados del tipo de entidad propietario. El siguiente esquema refleja esta nueva interpretación del ejemplo anterior:



Un tipo de entidad débil se puede modelar más adecuadamente como un atributo si sólo participa en la relación de identificación y si tiene pocos atributos. Por su parte, un tipo de entidad débil será más adecuada para modelar una situación en la que participe en otras relaciones además de la relación de identificación y en donde tenga muchos atributos. El diseñador de la base de datos elegirá cuál representación se utilizará.

Ejemplo 2: Dependencia en existencia



El tipo de entidad **SOCIO** representa el conjunto de socios de una entidad deportiva y el tipo de entidad **LICENCIA** representa el conjunto de licencias que posee un socio para hacer determinadas actividades. **LICENCIA** tiene una dependencia en existencia respecto a la entidad **SOCIO**, pues no existirá si no existe el socio al que pertenece, pero no en identificación, pues tiene un atributo clave: "N°Licencia" que permitiría identificar a cada ocurrencia de la entidad **LICENCIA**.

Si existe dependencia en existencia, el rombo que representa la interrelación puede etiquetarse con **EX**.

3. MODELO ENTIDAD/RELACIÓN EXTENDIDO

Aunque los conceptos del modelo E/R pueden modelar la mayoría de las características de las bases de datos, algunos aspectos de una base de datos pueden ser expresados más adecuadamente mediante ciertas extensiones del modelo E/R básico.

El modelo E/R extendido incluye los conceptos de **subclase**, **superclase**, **especialización**, **generalización**, **herencia**, etc...

3.1 Superclase y subclase

Un tipo de entidad se utiliza para representar un conjunto de entidades (ocurrencias) del mismo tipo. En muchos casos, un tipo de entidad puede contener varios subgrupos significativos de sus entidades, los cuales necesitan ser representados explícitamente por tener significado específico en los tratamientos de información.

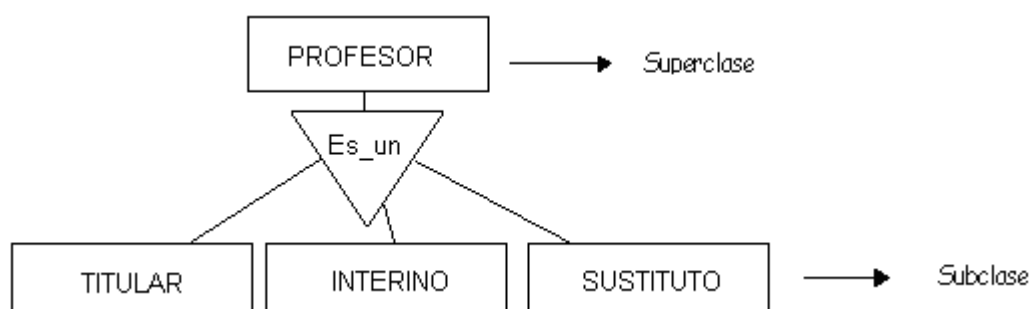
Por ejemplo, el tipo de entidad PROFESOR podría desglosarse en los siguientes subgrupos: PROF. TITULAR, INTERINO y SUSTITUTO.

Cada uno de los subgrupos se llama **subclase** (o **subtipo**) y el tipo de entidad PROFESOR es una **superclase** (o **supertipo**).

La interrelación que se establece entre los subtipos y el supertipo se llama **relación clase/subclase** y corresponde a la noción de "es-un" o, más precisamente, "es_un_tipo_de". En el ejemplo anterior, se diría que "un interino es un profesor", "un sustituto es un profesor"...

Esta clase de interrelación tiene la característica de que todo ejemplar de un subtipo es también un ejemplar del supertipo, aunque no sucede lo contrario.

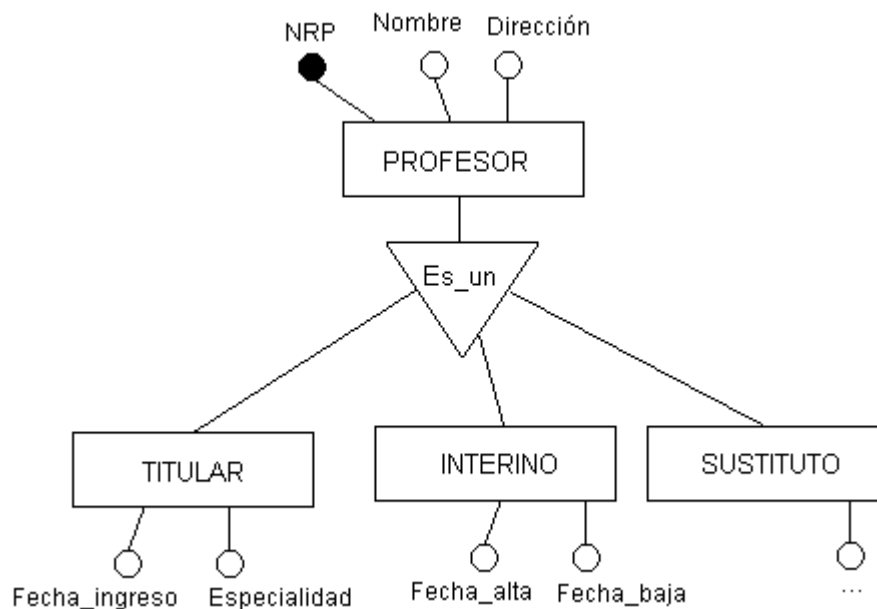
Para representar estas estructuras, se utiliza un triángulo que conecta la entidad que representa el supertipo con las entidades que representan los subtipos, tal como se muestra en la siguiente figura.



3.2 Herencia de atributos

Una de las características más importantes de este tipo de relaciones es la herencia, es decir, todo atributo de la superclase pasa a ser un atributo de las subclases. Y si la superclase participa en una interrelación, las subclases también participarán.

Por ejemplo, en el esquema anterior, tanto los profesores titulares como los interinos y los sustitutos, son profesores, por lo que heredarán todos los atributos de la entidad PROFESOR. Como vemos en la siguiente figura, la subclase TITULAR, aparte de sus atributos propios ("Fecha_ingreso" y "Especialidad"), hereda los atributos de PROFESOR ("NRP", "Nombre", "Dirección"...), y lo mismo podríamos decir para las otras dos subclases.



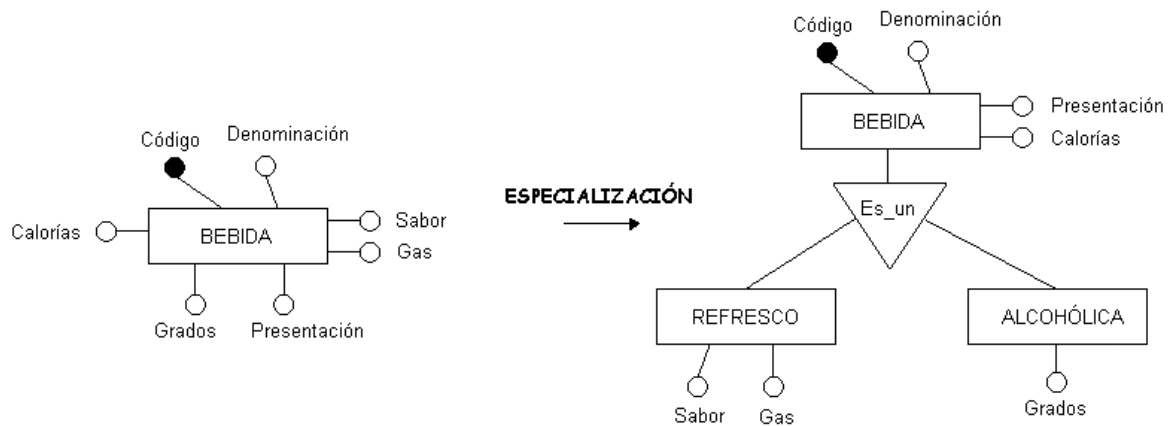
3.3 Especialización y generalización

La aparición de las jerarquías vistas en el apartado anterior puede surgir de dos formas diferentes en el modelado de bases de datos:

a) Especialización

Cuando se observa que un tipo de entidad tiene ciertos atributos y/o tipos de interrelación que tienen sentido para unos ejemplares pero no para otros, es conveniente definir uno o varios subtipos que contengan estos atributos específicos, dejando en el supertipo los que son comunes. Se utiliza para resaltar diferencias entre las subclases.

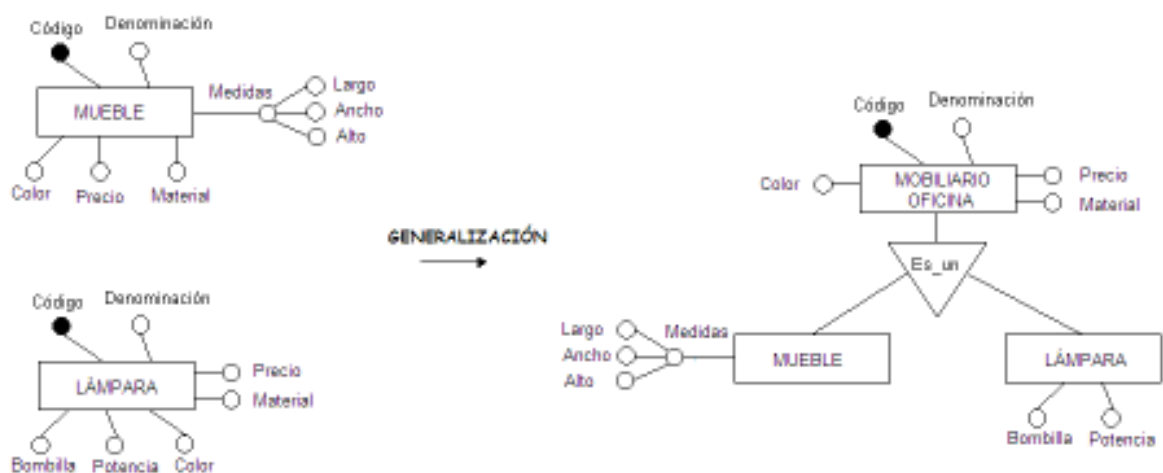
Por ejemplo, recogiendo información sobre bebidas, podemos modelar el esquema izquierdo de la figura siguiente. Pero, si analizamos los datos con un poco más de detalle, podemos llegar a la conclusión de que el atributo “Grados” sólo nos interesa para las bebidas alcohólicas, mientras que los datos sobre el sabor y si es gaseada o no, sólo nos interesan para las bebidas refrescantes. Con este razonamiento, podemos efectuar una especialización que nos llevará al modelo representado en el esquema derecho de la figura:



b) Generalización

Es el proceso inverso de la especialización. Consiste en unir conjuntos de entidades de bajo nivel (subclases) con atributos en común, para llegar a una clase generalizada (superclase). Se utiliza para resaltar semejanzas entre las subclases.

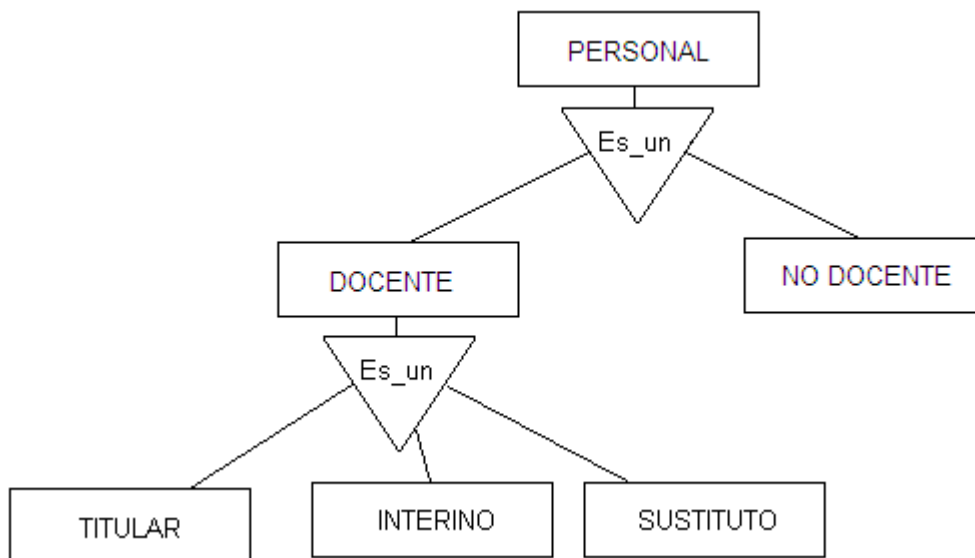
Por ejemplo, recogiendo información en un comercio de mobiliario de oficina, podemos modelar el esquema izquierdo de la figura siguiente. Analizando los datos, vemos que existen muchos atributos en común a las dos entidades, y podemos efectuar una generalización que se recoge en el modelo representado en la parte derecha:



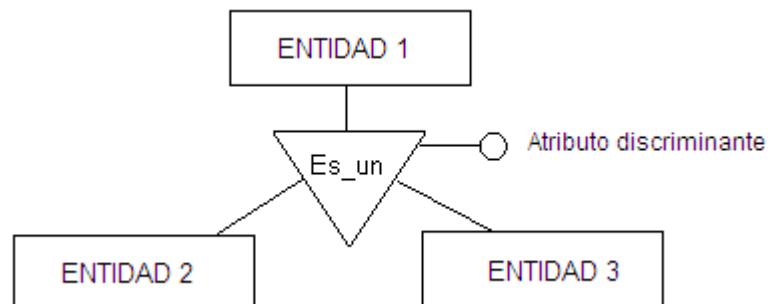
En resumen, la generalización y la especialización son procesos inversos que son usados para llegar al mismo punto final. Si nos movemos desde los subtipos hacia el supertipo, se trata de una generalización; si primero identificamos el supertipo y, a partir de él, llegamos a los subtipos, se trata de una especialización.

3.4 Características de la especialización y de la generalización

- Puede ocurrir que se formen jerarquías de más de un nivel, es decir, que un subtipo sea a su vez supertipo de otros. Por ejemplo:



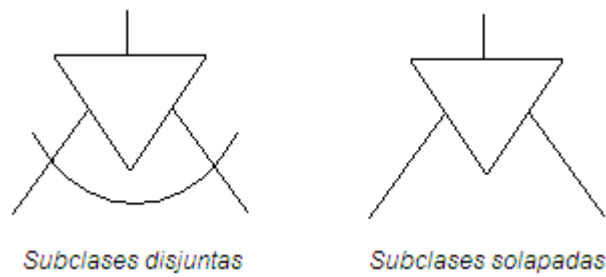
- La división en subtipos (especialización) puede venir determinada por una condición predefinida (por ejemplo, en función de los valores de un atributo), de manera que se representará la condición (o atributo discriminante) asociada al triángulo que representa la relación, como se muestra en la figura siguiente:



Cuando no interesa considerar ninguna condición para determinar la pertenencia a las subclases, será el usuario, en el momento de insertar un ejemplar en la base de datos, quien especifique a qué subtipo pertenece.

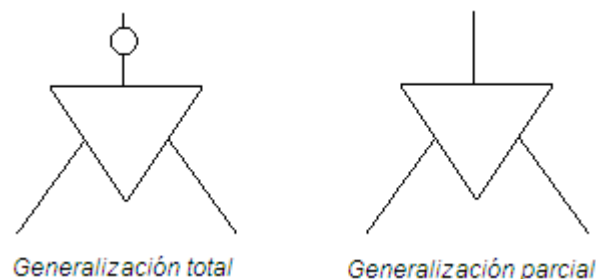
- En ocasiones, una entidad puede pertenecer a más de una subclase en la especialización o en la generalización. Según que esto sea o no posible, la generalización (o especialización) puede ser:
 - a) **Exclusiva o disjunta:** Una entidad es miembro de una única subclase.
 - b) **Solapada o superpuesta:** Una misma entidad puede ser miembro de más de una subclase.

La siguiente figura muestra como se representa cada caso:



- Cada entidad de la superclase puede o no pertenecer al menos a una subclase de la especialización o generalización. Según eso, la generalización puede ser:
 - a) **Total:** Una entidad en la superclase debe ser miembro de alguna subclase.
 - b) **Parcial:** Alguna entidad en la superclase no corresponde a ninguna subclase. Una especialización o generalización parcial describe un conocimiento incompleto del problema.

La siguiente figura muestra como se representa cada caso:



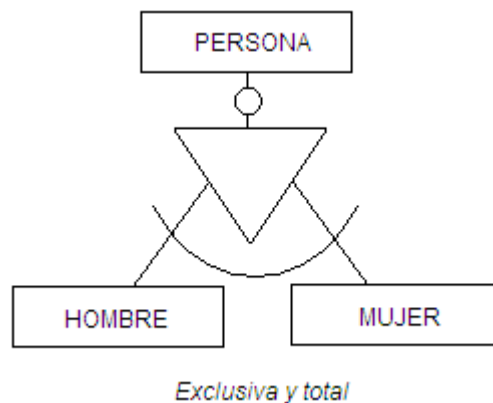
En resumen, hay cuatro tipos de posibilidades en la especialización o en la generalización:

- a) Exclusiva y total
- b) Exclusiva y parcial
- c) Solapada y total
- d) Solapada y parcial

Veamos un ejemplo y la representación de cada una:

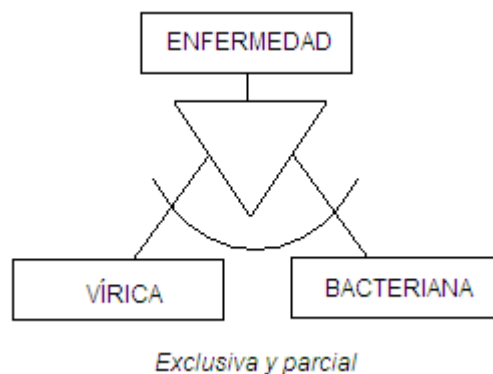
- a) Exclusiva y total

El tipo de entidad PERSONA puede ser especializada en dos subclases: HOMBRE y MUJER, de forma **total y sin solapamiento**. Una entidad PERSONA pertenecerá a la subclase HOMBRE o a la subclase MUJER; es decir, no existirá una entidad PERSONA que no sea de alguna de estas dos subclases y, además, de forma exclusiva (pertenecerá a una y sólo a una de estas subclases).



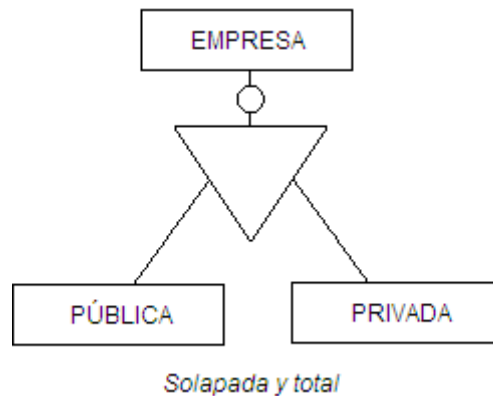
- b) Exclusiva y parcial

El tipo de entidad ENFERMEDAD puede ser especializada en dos subclases: VÍRICA y BACTERIANA. Pero también puede suceder que tengamos entidades que no puedan ser clasificadas como ninguno de los dos tipos debido, posiblemente, al desconocimiento del valor del atributo utilizado como condición:



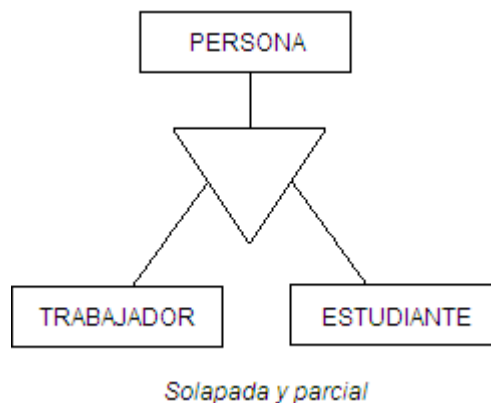
c) Solapada y total

Un tipo de entidad EMPRESA se especializa en dos subclases: PÚBLICA y PRIVADA. Se puede producir el hecho de que existan entidades que puedan ser consideradas tanto del tipo PÚBLICA como PRIVADA, o bien de ambos tipos al mismo tiempo y, además, el hecho de que no existan entidades que no puedan ser especializadas en alguna de estas dos subclases.



d) Solapada y parcial

En este caso, podemos considerar un tipo de entidad PERSONA que puede ser especializado en dos subclases: TRABAJADOR y ESTUDIANTE, de forma parcial con solapamiento. Es decir, una entidad PERSONA puede ser del tipo TRABAJADOR y/o del tipo ESTUDIANTE y, además, pueden existir entidades PERSONA que no puedan clasificarse en ninguna de estas dos subclases.



4. MODELO RELACIONAL

El modelo relacional fue desarrollado inicialmente por E. F. Codd en los años setenta y se ha ido divulgando progresivamente hasta ocupar el primer lugar en los sistemas de bases de datos utilizados. El modelo se ha refinado con el tiempo para cubrir las necesidades reales de los usuarios y, sobre todo, para aumentar su potencia sin afectar a su simplicidad.

4.1 Elementos del modelo relacional

El elemento principal del modelo relacional de datos es la **relación**. No debe confundirse con el término “relación” visto en el modelo E/R. Recordamos que, en el modelo E/R, una relación es una asociación o correspondencia entre entidades, y que también podía llamarse interrelación.

En el modelo relacional, cada relación se representa mediante una **tabla bidimensional**, y se utilizarán relaciones (tablas) para representar tanto las entidades como las interrelaciones estudiadas en el modelo conceptual E/R. Por lo tanto, el esquema de una base de datos relacional es un conjunto de relaciones que cumplen unas determinadas propiedades.

La figura siguiente es un ejemplo de tabla para la relación COCHE-USADO:

Matrícula	Modelo	KM	PVP	Garantía
1234-CGF	Renault Laguna	5.000	3.000	6
7890-AUP	Ford Focus	60.000	1.800	6
5599-PMC	Opel Astra	2.500	6.000	12
3344-BNC	Audi A3	30.000	9.500	12

4.2 Terminología empleada en el modelo relacional

De las tablas se derivan los siguientes conceptos:


- **Tupla o registro.** Es cada una de las filas de la tabla.
- **Atributo o campo.** Es cada una de las columnas de la tabla. Todos los registros o tuplas tienen igual número de atributos o campos.
- **Cardinalidad.** Es el número de tuplas de la tabla.

- **Grado.** Es el número de atributos de la tabla.
- **Dominio de un atributo.** Es el conjunto de valores que puede tomar dicho atributo. Por ejemplo, el dominio del atributo “*Garantía*” es un número comprendido entre 3 y 18, que representa el número de meses mínimo y máximo que ofrece el concesionario como garantía del coche.
- **Clave candidata.** Conjunto mínimo de atributos que identifican de forma unívoca cada tupla de una relación.
- **Clave principal o primaria.** Clave candidata elegida para identificar las tuplas de una relación. Las restantes claves candidatas que no han sido elegidas como claves primarias de una relación son **claves alternativas**.
- **Clave foránea o externa.** Conjunto de atributos en una relación que es una clave primaria en otra (o incluso en la misma). Dicho de un modo más formal, si una relación R1 está relacionada con otra R2, una clave foránea de la relación R2 es un conjunto de atributos cuyos valores han de coincidir con los valores de la clave primaria de la relación R1. Por ejemplo:

DEPARTAMENTOS (R1)	
Código	Nombre
D001	Informática
D002	Inglés
D003	Ciencias

LIBROS (R2)				
ISBN	Título	Autor	...	CódigoDepto
012345678	SQL Server	De Miguel, A.	...	D001
876543210	La célula	Ibarra, E.	...	D003
234567890	El cuerpo humano	Salgado, M.	...	D003
...

Clave foránea



El atributo “CódigoDepto” es un ejemplo de clave foránea: los valores del atributo “CódigoDepto” de la relación LIBROS deben coincidir con los de la clave primaria, “Código”, de la relación DEPARTAMENTOS.

Ejemplo

En la tabla del apartado 4.1 se puede observar una relación llamada COCHE-USADO, con las siguientes características:

- Grado: cinco.
- Cardinalidad: cuatro.
- Atributos: “Matrícula”, “Modelo”, “KM”, “PVP” y “Garantía”.
- Una tupla es, por ejemplo, la formada por los datos {3344-BNC, Audi A3, 30.000, 9.500, 12}.
- El dominio de “Garantía” es el conjunto de números enteros positivos comprendidos entre 3 y 18.
- Clave candidata: “Matrícula”.
- Clave primaria: “Matrícula” (no hay claves alternativas).

4.3 Requisitos de una tabla en una base de datos relacional

En general, una **tabla** debe reunir los siguientes requisitos para ser considerada como una relación:

- Debe tener un número fijo de atributos para todas las tuplas.
- Cada atributo tiene un único dominio.
- El orden de las tuplas y de los atributos no es relevante.
- No puede haber dos tuplas iguales.
- Cada intersección fila-columna debe contener un valor único perteneciente al dominio de la columna correspondiente (no hay grupos repetitivos).

4.4 Vistas

Los conceptos que se han expuesto hasta el momento hacen referencia al esquema conceptual del modelo relacional. Para describir el esquema externo se utilizan las vistas.

Una vista es una tabla que el usuario puede crear y manejar. Es una tabla virtual que no tiene que corresponderse con ningún archivo del nivel interno. Las tuplas que pertenecen a una vista se obtienen como resultado de consultas a las tablas.

Las vistas pueden formarse eliminando atributos de una tabla, uniendo tablas por atributos comunes y de ambas formas. También pueden definirse a partir de otras vistas.

4.5 Reglas de integridad

Las operaciones que pueden afectar a la integridad de los datos son la inserción, la modificación y el borrado de registros en la base de datos.

Existen algunas restricciones que deben cumplirse para mantener la base de datos íntegra (aunque no todos los sistemas gestores de bases de datos relacionales facilitan mecanismos para especificar tales restricciones).

Las tres restricciones principales que forman parte del modelo relacional de datos son las siguientes:

- a) **Integridad de entidad**. Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo o desconocido.

Esto es lógico ya que, por definición, la clave primaria permite distinguir las tuplas de una relación. Si alguno de los atributos que forman la clave primaria tomase valores nulos, algunas de las tuplas de la relación podrían no ser identificadas.

Esta restricción se consigue comprobando en cada inserción y modificación que ningún valor de los introducidos para la clave primaria es nulo o no forma parte de su dominio.

- b) **Integridad de clave**. Los valores de claves candidatas en una relación deben ser únicos para cada tupla.

El razonamiento es análogo al anterior.

Para evitar violar esta regla, se comprobará en todas las inserciones y modificaciones que el valor de todas las claves candidatas no existe ya en alguna tupla de esa relación.

- c) **Integridad referencial.** Nos indica que los valores de la clave foránea en la relación “hijo” deben corresponderse con los valores de la clave primaria en la relación “padre”, o bien ser nulos, si se admiten nulos. (Conviene fijarse en que es la clave foránea la que admitiría valores nulos, nunca la clave primaria en la tabla referenciada o tabla “padre”).

Esta regla puede verse afectada en la inserción, modificación y borrado de tuplas:

- Para las dos primeras operaciones, el SGBD relacional debe asegurarse de que, al introducir un valor de una clave foránea, ese valor sea nulo (si se permiten) o exista en todas las relaciones en las que la clave foránea sea clave primaria.
- En cuanto a la eliminación, no se debe permitir el borrar una tupla de una relación cuyo valor de clave primaria exista como valor de clave foránea de otra relación.

Para entender esta regla, veamos la siguiente figura:

PROFESOR			ASIGNATURA		
Nombre	Dirección	Teléfono	Nombre	Horas	Tipo
Pepe	C/Real ...	999 999999	Inglés	4	Optativa
Juan	C/Nueva...	444 444444	Lengua	5	Obligatoria
Ana	F. y Q.
Luís	Matemát.
María	Historia

IMPARTIR	
Nombre_Profesor	Nombre_Asignatura
Pepe	Inglés
Ana	Matemát.
María	Lengua

Si en la relación IMPARTIR apareciera un profesor llamado “Eduardo” y éste no estuviera en la relación PROFESOR, se violaría la regla de integridad referencial, ya que en la relación IMPARTIR los atributos “Nombre_Profesor” y “Nombre_Asignatura” son claves foráneas.

Ocurriría algo similar si en la relación IMPARTIR apareciera una asignatura llamada “Base de datos” y ésta no estuviera en la relación ASIGNATURA.

También se perdería la integridad si se borrara la tupla {Pepe, C/Real..., 999 999999}, pues "Pepe" es el valor de la clave primaria en PROFESOR y existe como valor de clave foránea para la tupla {Pepe, Inglés} en la relación IMPARTIR.

Además de estas restricciones formales, es conveniente que los SGBD relacionales tengan en cuenta lo siguiente:

- No se permitirá introducir valores de atributos que no pertenezcan a sus dominios o sean de distinto tipo.
- Se podrán definir atributos obligatorios y opcionales. Los obligatorios, sean claves o no, deberán contener siempre valores válidos.
- Se podrán formular reglas de gestión de la base de datos (reglas semánticas), tales como "la fecha de devolución será mayor que la de préstamo", "el mínimo de unidades suministradas por un proveedor será tres", "todos los valores numéricos deberán ser positivos", "un árbitro no podrá participar en dos partidos consecutivos de un mismo equipo", etc.

4.6 Transformación del modelo E/R al modelo relacional

Una vez obtenido el esquema conceptual mediante el modelo E/R, los pasos a seguir son:

- Transformar el diagrama E/R (esquema conceptual) en el modelo relacional.
- Aplicar reglas de normalización (para diseñar relaciones bien estructuradas).

Las tres reglas básicas para transformar el diagrama E/R al modelo relacional son:

1. Toda entidad se transforma en una relación.
2. Las interrelaciones N:M se transforman en una relación.
3. Las interrelaciones 1:N dan lugar a una relación o bien a una propagación de clave.

Veamos las reglas de forma más detallada.

4.6.1 Transformación de entidades

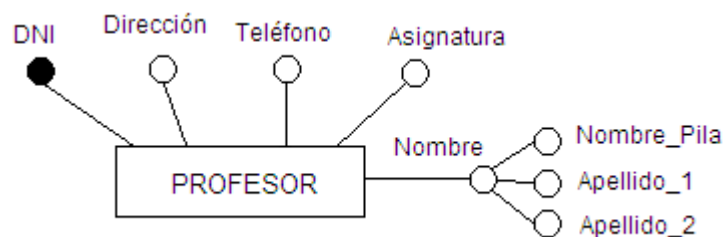
Cada tipo de entidad se transforma en una relación o tabla. La tabla tendrá el mismo nombre que el tipo de entidad del que procede.

Cada atributo de una entidad se transforma en una columna de la tabla a la que la entidad a dado lugar. Hay que tener en cuenta que existen atributos identificadores, otros que son identificadores alternativos y atributos no identificadores. Para los dos últimos tipos, si se desea que no puedan tomar valores nulos, hay que indicarlo.

Los atributos compuestos dan lugar a una columna para cada uno de sus atributos simples.

Por ejemplo, veamos el siguiente modelo E/R y su correspondencia con el modelo relacional:

Modelo E/R



Modelo relacional

PROFESOR						
DNI	Nombre_Pila	Apellido_1	Apellido_2	Asignatura	Dirección	Teléfono
32320000	Manuel	Leal	Blanco	Lengua	Real 22	986111111
36101010	Juan	Gómez	Ríos	Inglés	Badajoz 11	650777777
33999999	María	Rodríguez	Sánchez	Física	Pez 77	567888888

También se puede indicar de la forma:

PROFESOR (DNI, Nombre_Pila, Apellido_1, Apellido_2, Asignatura, Dirección, Teléfono)

La clave primaria se señala mediante el atributo subrayado.

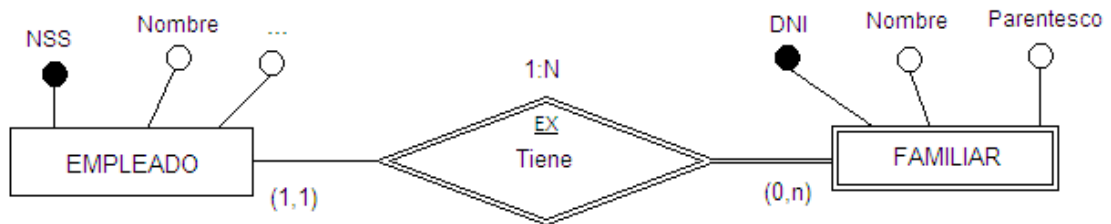
4.6.1.1 Tipo de entidades débiles

Cada atributo de una entidad débil se transforma en una columna de la relación, igual que en el caso anterior pero, además, hay que incluir como clave foránea el/los atributos que

formen la clave primaria en la tabla principal. Además, en el caso de que la entidad sea débil en identificación, la clave principal de esta relación o tabla estará formada por los atributos de la clave foránea más los atributos que formen la clave parcial de la entidad débil.

Por ejemplo:

Modelo E/R



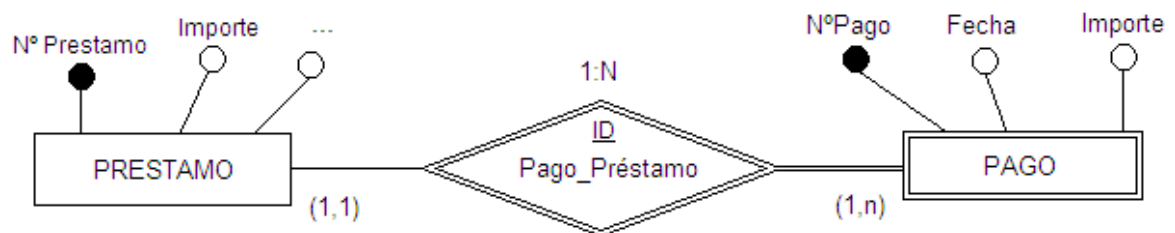
Modelo Relacional

FAMILIAR (NSS, DNI, Nombre_Familiar, Parentesco)

La clave primaria es sólo el atributo “DNI”, porque se trata de un caso de debilidad en existencia.

Retomemos el ejemplo del apartado 2.7, que representaba un caso de debilidad en identificación, y veamos cómo sería su conversión al modelo relacional:

Modelo E/R



Modelo Relacional

PAGO (N°Prestamo, N°Pago, Fecha, Importe_Pago)

En este caso, la clave primaria de la relación PAGO sería la concatenación de “N°Prestamo” y “N°Pago”.

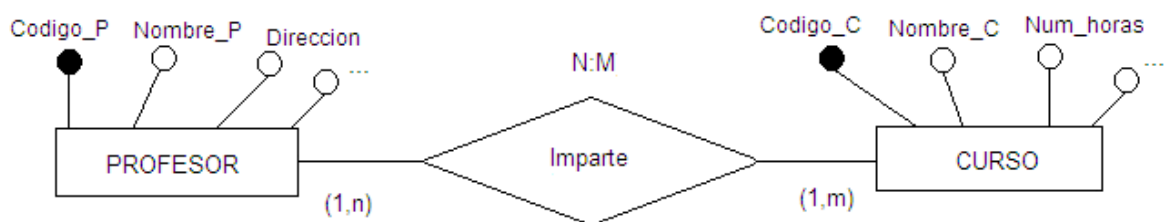
4.6.2 Transformación de interrelaciones

4.6.2.1 Interrelaciones N:M

Una interrelación N:M se transforma en una relación o tabla que tendrá como clave primaria la concatenación de las claves primarias de los tipos de entidad que asocia, y como atributos los propios de la interrelación.

Por ejemplo:

Modelo E/R:



Modelo relacional:

PROFESOR (Codigo_P, Nombre_P, Direccion...)

CURSO (Codigo_C, Nombre_C, NumHoras...)

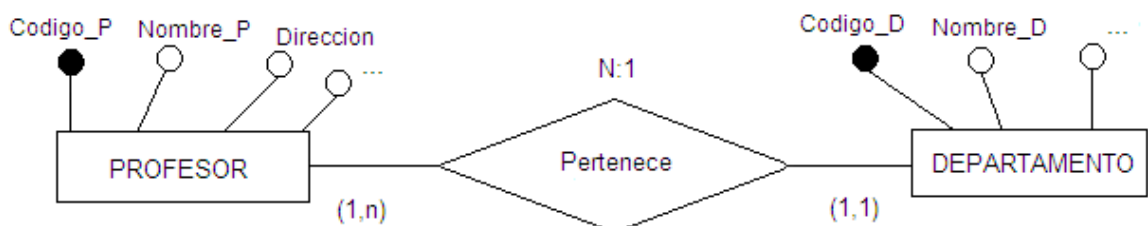
IMPARTE (Codigo_P, Codigo_C)

4.6.2.2 Interrelaciones 1:N

Existen dos posibilidades:

- a) Propagación de clave: Propagar los atributos que forman la clave principal del tipo de entidad de cardinalidad 1 a la que tiene cardinalidad N, desapareciendo el nombre de la interrelación. Por ejemplo:

Modelo E/R:



Modelo relacional:

PROFESOR (Codigo_P, Nombre_P, Dirección..., Codigo_D)

DEPARTAMENTO (Codigo_D, Nombre_D...)

Si la interrelación tuviese atributos propios, estos se propagarían en el mismo sentido que la clave.

- b) Transformar la interrelación 1:N como si fuese una interrelación N:M. Es decir, se crea una nueva relación (tabla) cuya clave primaria es, en este caso, sólo la clave primaria de la tabla a la que le corresponde la cardinalidad N. Por ejemplo, para el modelo E/R anterior, la transformación al modelo relacional daría lugar a las relaciones siguientes:

PROFESOR (Codigo_P, Nombre_P, Dirección...)

DEPARTAMENTO (Codigo_D, Nombre_D...)

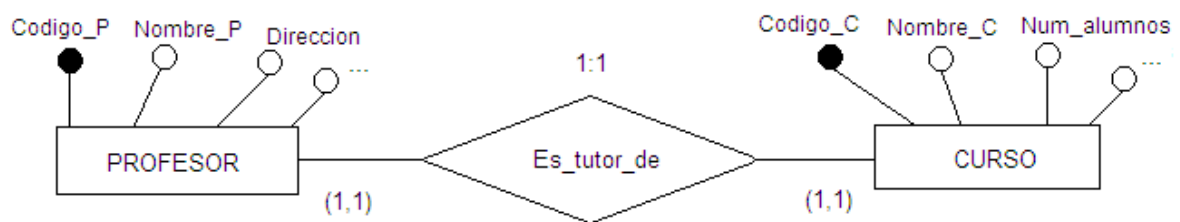
PERTENECE (Codigo_P, Codigo_D)

4.6.2.3 Interrelaciones 1:1

Una interrelación de tipo 1:1 es un caso particular de una N:M o también de una 1:N, por lo que no hay regla fija para la transformación: puede crearse una nueva relación o bien efectuar una propagación de clave.

El siguiente ejemplo se ha resuelto mediante la creación de una nueva relación (aunque no sea la solución más eficiente, por el número de tablas que genera):

Modelo E/R:



Modelo relacional:

PROFESOR (Codigo_P, Nombre_P, Dirección...)

CURSO (Codigo_C, Nombre_C, Num_alumnos...)

ES_TUTOR_DE (Codigo_P, Codigo_C)

La clave de la nueva relación (ES_TUTOR_DE) puede ser cualquiera de sus dos atributos.

En el caso de la propagación de clave, dicha propagación podría realizarse en ambos sentidos, aunque debemos tener en cuenta dos posibilidades:

- a) Si una entidad de las que participan en la interrelación posee cardinalidad (0,1) mientras que la otra posee (1,1), es mejor propagar la clave de la entidad con cardinalidad (1,1). Por ejemplo:

Modelo E/R



Modelo relacional

PROFESOR (Codigo_P, Nombre_P, Dirección...)

DEPARTAMENTO (Codigo_D, Nombre_D..., Codigo_P)

Se ha propagado la clave de la entidad PROFESOR ("Codigo_P") porque es la entidad que tiene cardinalidad (1,1). Esta solución evita los valores nulos que se producirían de haber hecho la propagación en sentido contrario. Es decir, si el paso a modelo relacional se hubiera hecho propagando la clave de la entidad DEPARTAMENTO a la relación PROFESOR, el resultado hubiera sido:

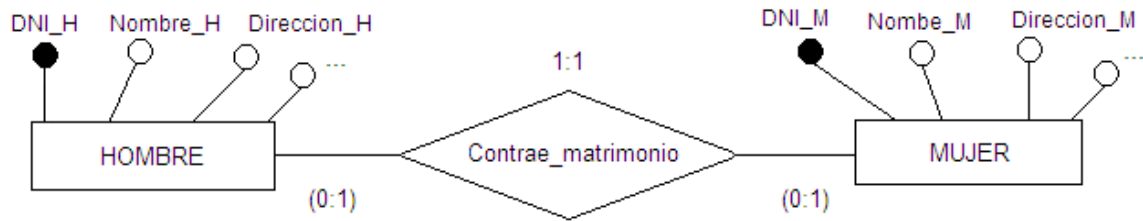
PROFESOR (Codigo_P, Nombre_P, Dirección..., Codigo_D)

DEPARTAMENTO (Codigo_D, Nombre_D...)

Como hay profesores que no dirigen ningún departamento, lo que viene indicado por la cardinalidad (0,1), habrá tuplas en la relación PROFESOR en las que el campo "Codigo_D" no tenga ningún valor, o lo que es igual, tenga valor nulo.

- b) Si las dos entidades que participan en la interrelación poseen cardinalidad (0,1), puede ser más adecuado realizar la transformación en una nueva relación. Por ejemplo:

Modelo E/R



Modelo relacional

HOMBRE (DNI_H, Nombre_H, Dirección_H...)

MUJER (DNI_M, Nombre_M, Dirección_M...)

CONTRAE_MAT. (DNI_H, DNI_M)

La clave de la nueva relación (CONTRAE_MATRIMONIO) puede ser cualquiera de sus dos atributos.

Al crear una nueva relación, se evitan los valores nulos que se producirían de haber propagado la clave de la relación MUJER a la relación HOMBRE o viceversa, ya que, según refleja la cardinalidad, no todos los hombres ni todas las mujeres están casados.

4.6.3 Transformación de atributos multivaluados

Para los atributos multivaluados se creará una relación o tabla que incluya:

- La clave primaria (K) de la relación correspondiente al tipo de entidad (o al tipo de interrelación) que contiene el atributo multivaluado.
- Un atributo (A) que corresponda al propio atributo multivaluado.

La clave primaria de la nueva relación estará formada por K y A.

Por ejemplo, supongamos en el siguiente esquema el atributo multivaluado "Ubicación", referente a la entidad DEPARTAMENTO:



Al transformarlo al modelo relacional, obtenemos las relaciones que se indican a continuación:

PROFESOR (Codigo_P, Nombre_P, Dirección...)
DEPARTAMENTO (Codigo_D, Nombre_D..., Codigo_P)
UBICACIÓN (Codigo_D, Ubicación)

Se ha creado una nueva relación llamada UBICACIÓN, que tiene como atributos:

- La clave primaria de la relación DEPARTAMENTO, correspondiente a la entidad del mismo nombre, la cual contiene el atributo multivaluado.
- Un atributo llamado “Ubicación”, correspondiente al propio atributo multivaluado.

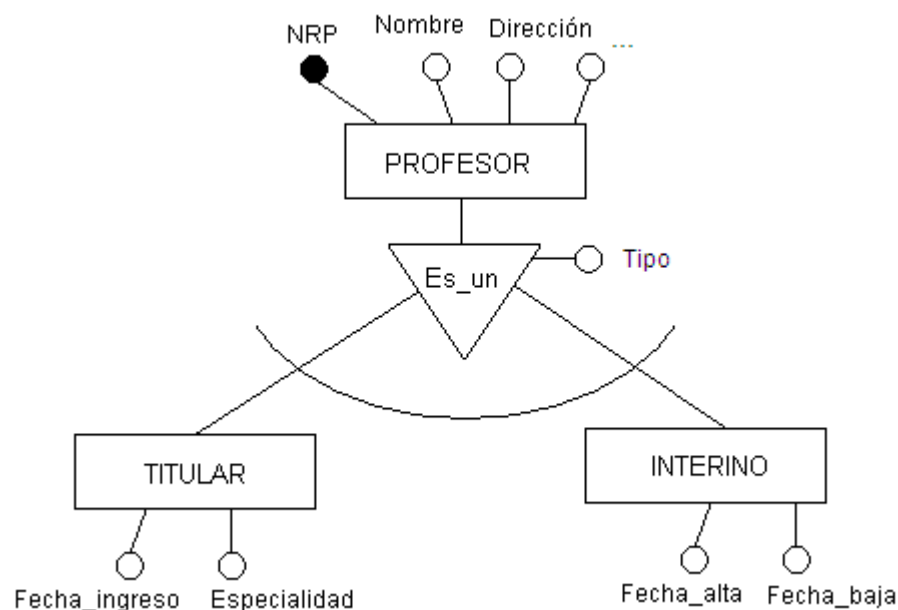
Ambos atributos conforman la clave primaria.

4.6.4 Transformación de especializaciones/generalizaciones

El modelo relacional no dispone de mecanismos explícitos para la representación de esquemas jerárquicos de tipos y subtipos. Por tanto, en la transformación de estas relaciones caben varias soluciones:

- Crear una relación para la superclase y tantas relaciones como subclases haya, que contengan sus atributos propios y los atributos que conformen la clave primaria de la superclase.

Por ejemplo, retomemos el esquema del apartado 3.2:



PROFESOR (NRP, Nombre, Dirección..., Tipo)

TITULAR (NRP, Fecha_ingreso, Especialidad)

INTERINO (NRP, Fecha_alta, Fecha_baja)

En general, esta solución es buena cuando existen muchos atributos distintos entre los subtipos y se quieren mantener los comunes a todos ellos en una relación. Además, como los atributos exclusivos de cada subtipo pertenecen a su correspondiente relación, no genera valores nulos. Por el contrario, es la solución que da lugar a más relaciones y, por tanto, la menos eficiente en aquellos casos en los que haya que recurrir a la combinación de tablas para obtener la información que se necesite.

- b) No crear una relación para la superclase y crear tantas relaciones como subclases haya, que contengan, además de sus propios atributos, los atributos de la superclase, o atributos comunes.

El resultado sería:

TITULAR (NRP, Nombre, Dirección..., Fecha_ingreso, Especialidad)

INTERINO (NRP, Nombre, Dirección..., Fecha_alta, Fecha_baja)

Esto es adecuado si las subclases son disjuntas y la especialización es total. Tampoco genera valores nulos.

- c) Englobar todos los atributos de la entidad superclase y sus subclases en una sola relación, añadiendo el atributo discriminante que indica el tipo de subclase.

El resultado sería:

PROFESOR (NRP, Nombre, Dirección... Tipo, Fecha_ingreso, Especialidad, Fecha_alta, Fecha_baja)

En general, esta solución es buena cuando las subclases se diferencian en muy pocos atributos y las interrelaciones que los asocian con el resto de las entidades del esquema son las mismas para todos. Además, desde el punto de vista de la eficiencia, el acceso a toda la información de una determinada ocurrencia de una entidad (a todos los datos de un determinado profesor, por ejemplo), es mucho más rápido, ya que no hace falta combinar varias relaciones. Por el contrario, es inevitable la generación de valores nulos en los atributos específicos de cada subtipo.

4.6.5. Transformación de atributos derivados

Los atributos derivados se pueden tratar como atributos normales, con lo que pasarán a ser columnas de la relación que corresponda. En este caso, deberá arbitrarse algún mecanismo que calcule automáticamente su valor a partir del atributo del cual se derivan, para evitar posibles inconsistencias.

Otra solución es no almacenar las columnas que provengan de atributos derivados, y crear procedimientos que calculen sus valores cada vez que se necesiten, aunque esto puede generar problemas de eficiencia.

5. NORMALIZACIÓN: IDEAS BÁSICAS

En los apartados anteriores se ha descrito la forma de expresar gráficamente los datos de una base de datos utilizando el modelo E/R y la forma de trasladar el diagrama resultante al modelo relacional de datos. En este apartado se estudia el siguiente paso en el diseño de una base de datos: convertir el esquema relacional en un buen esquema de base de datos que permita almacenar toda la información con un mínimo de redundancia y que a la vez facilite su recuperación.

Si se realiza un buen diseño en los pasos anteriores, en general se obtendrá un esquema exento de errores. Sin embargo, ante las posibles dudas respecto a si un determinado esquema relacional es o no correcto, será preferible aplicar a dicho esquema un método formal de análisis que determine lo que pueda estar equivocado en el mismo y nos permita llegar a otro esquema mejor. Este método formal es la **teoría de la normalización**.

La teoría de la normalización se centra en lo que se conoce con **formas normales**. Se dice que un esquema de relación está en una determinada forma normal si satisface un conjunto específico de restricciones.

En el proceso de normalización, según la propuesta original de Codd, se somete al esquema de relación a una serie de pruebas para “certificar” si pertenece o no a una cierta forma normal. En un principio, Codd propuso tres formas normales, a las cuales llamó “Primera Forma Normal” (**1FN**), “Segunda Forma Normal” (**2FN**) y “Tercera Forma Normal” (**3FN**). Posteriormente, Codd y Boyce propusieron una definición más estricta de la 3FN, a la que se conoce como **forma Normal de Boyce-Codd**. Todas estas formas normales se basan en el concepto de dependencias funcionales. Más adelante se propusieron otras formas normales: **4FN** y **5FN**, con el fundamento de las dependencias multivaluadas.

En esta unidad se tratará el estudio de las tres primeras formas normales.

5.1 Dependencias funcionales

Las dependencias funcionales se refieren a la capacidad de determinar el valor de un atributo conociendo otro(s).

Por ejemplo, sea la relación **TRABAJADOR** (CodTrabajador, Nombre, TipoOficio). En cada tupla, "CodTrabajador" determina a "Nombre" y, también, "CodTrabajador" determina a "TipoOficio". Dicho de otro modo, conociendo el valor del atributo "CodTrabajador", podemos determinar sin ambigüedades el valor de "Nombre" y también el valor de "TipoOficio".

Estas son dos dependencias funcionales y su notación es la siguiente:

DF: CodTrabajador \rightarrow Nombre

DF: CodTrabajador \rightarrow TipoOficio

Formalmente, si X e Y son atributos en una relación R, se dice que **X determina funcionalmente a Y** o, lo que es lo mismo, que **Y depende funcionalmente de X**, y se representa $X \rightarrow Y$ si, para toda ocurrencia de R, el valor de X determina unívocamente el valor de Y (es decir, si para cada valor de X existe un único valor de Y asociado a ese valor de X).

En el ejemplo anterior, se dice que:

- "CodTrabajador" determina funcionalmente a "Nombre".
- "CodTrabajador" determina funcionalmente a "TipoOficio".

Al conjunto de atributos de la parte izquierda de la dependencia se le denomina **determinante** ("CodTrabajador" es el atributo determinante).

La clave de una relación es siempre un determinante, ya que su valor determina unívocamente el valor de todos los atributos de una tupla, pero no todo determinante es clave.

Dependencia funcional total

Se dice que Y tiene una dependencia funcional total de X si y sólo si no existe un subconjunto Z contenido en X, tal que $Z \rightarrow Y$; es decir, si Y depende funcionalmente de X y no depende funcionalmente de ningún subconjunto propio de X. Dicho de otro modo, Y tiene una dependencia funcional total de X, si de la colección de atributos de que está formado X no podemos eliminar ninguno sin que se rompa la dependencia funcional.

Por ejemplo:

- DF: $A \rightarrow B$, B tiene una dependencia total de A.
- DF: $AC \rightarrow B$ y $A \rightarrow B$, B no tiene dependencia funcional total de AC, porque depende funcionalmente de A (podemos eliminar el atributo C, y B sigue dependiendo funcionalmente de A).

De lo anterior se deduce que, si $X \rightarrow Y$, y ocurre que X tiene un solo atributo, entonces la dependencia funcional es total.

Atributos primos

Un atributo de una relación R se denomina **atributo primo** o **principal** si es miembro de alguna clave de R.

Un atributo es **no primo** si no es miembro de ninguna clave de R.

5.2 Primera forma normal (1FN)

Un esquema de relación R está en Primera Forma Normal cuando todos sus atributos son atómicos (o indivisibles). No se aceptan atributos multivaluados ni compuestos.

La 1FN es una restricción inherente al modelo relacional. Recordamos que, para que una tabla pueda ser considerada una relación, no debe admitir grupos repetitivos, esto es, debe estar en 1FN.

Por ejemplo:

Supongamos la relación **LIBRO** (Código, Título, Autor) que se muestra a continuación:

CODIGO	TITULO	AUTOR
12345	Guía de viajes	Ángel Romero Santiago Bermúdez
23456	Las violetas	Elvira Santurce
34567	Turismo rural	Juan Segovia Marta Leal

No está en 1FN porque en el campo “Autor” hay grupos repetitivos: hay libros que tienen más de un autor. En realidad, se trata de una tabla, pero no es una relación según el modelo relacional.

Para pasar esta tabla a 1FN habrá que deshacer los grupos repetitivos formando nuevas tuplas, tal como aparece a continuación:

CODIGO	TITULO	AUTOR
12345	Guía de viajes	Ángel Romero
12345	Guía de viajes	Santiago Bermúdez
23456	Las violetas	Elvira Santurce
34567	Turismo rural	Juan Segovia
34567	Turismo rural	Marta Leal

De este modo, la relación está en 1FN, y la clave primaria estaría compuesta por “Código” y “Autor”. Es decir, la relación ahora se representaría como **LIBRO** (Código, Título, Autor).

5.3 Segunda forma normal (2FN)

Un esquema de relación R está en Segunda Forma Normal cuando está en 1FN y, además, todo atributo no primo de R tiene dependencia funcional completa respecto de cada una de las claves.

La 2FN no se cumple cuando algún atributo no principal depende funcionalmente de algún subconjunto de la clave.

Por ejemplo:

Retomemos la anterior relación **LIBRO** (Código, Título, Autor), que está en 1FN, y veamos si está en 2FN. Para ello, vamos a estudiar las dependencias funcionales existentes:

DF: Código, Autor \rightarrow Título

DF: Código \rightarrow Título

El atributo no primo "Título" no depende funcionalmente de manera total de la clave primaria ("Código", "Autor") porque se cumple que Código \rightarrow Título. Por tanto, esta relación no está en 2FN.

Si una relación no está en 2FN, se puede normalizar en varias relaciones 2FN en las que los atributos no primos estén asociados sólo a la parte de la clave primaria de la que dependen funcionalmente de manera total. Para ello, extraemos de la relación los atributos que hacen fallar la 2FN, y creamos una nueva relación con cada uno de ellos, acompañado de la parte de la clave primaria de la que dependen de forma total.

Por ejemplo:

La relación anterior **LIBRO** (Código, Autor, Título) se descompondría en dos relaciones:

LIBRO1 (Código, autor)

LIBRO2 (Código, Título)

- La relación que llamamos LIBRO1 es la resultante de extraer el atributo "Título" (por el cual no se cumplía la 2FN) de la relación inicial.
- La relación que llamamos LIBRO2 es la creada con el atributo extraído: "Título", más su determinante: "Código".

De todo lo visto podemos extraer la siguiente conclusión: una relación estará en 2FN si todas las claves son simples o si todos los atributos son primos o principales ya que, en cualquiera de los casos, la dependencia de la clave será total.

5.4 Tercera forma normal (3FN)

Una relación está en Tercera Forma Normal si y sólo si está en 2FN y no existe ningún atributo no primo de R que dependa transitivamente de alguna clave de R.

La 3FN no se cumple cuando existen atributos no principales que dependen funcionalmente de otros atributos no principales.

Por ejemplo:

Supongamos la relación **TRABAJADOR** (CodTrabajador, TipoOficio, Prima), y consideremos que se cumplen las siguientes dependencias funcionales:

DF: CodTrabajador → TipoOficio

DF: CodTrabajador → Prima

Es lógico que ambas sean dependencias funcionales para esa relación, porque “CodTrabajador” es una clave. Sin embargo, también existe otra dependencia:

DF: TipoOficio → Prima

Y, en este caso, el determinante “TipoOficio” no es una clave, sino otro atributo no principal. En conclusión, la anterior relación no está en 3FN.

Siempre es posible transformar un esquema que no está en 3FN en esquemas en 3FN sin perder información ni dependencias funcionales. Para ello, debemos descomponer la relación extrayendo de ella los atributos que figuren en la parte derecha de cualquier dependencia funcional que haga fallar el criterio de la 3FN, y creando una nueva relación con cada uno de ellos, acompañado de su atributo determinante.

Por ejemplo:

La relación anterior **TRABAJADOR** (CodTrabajador, TipoOficio, Prima) se descompondría en dos relaciones:

TRABAJADOR1 (CodTrabajador, TipoOficio)

TRABAJADOR2 (TipoOficio, Prima)

- La relación que llamamos TRABAJADOR1 es la resultante de extraer el atributo “Prima” (por el cual no se cumplía la 3FN) de la relación inicial.

- La relación que llamamos TRABAJADOR2 es la creada con el atributo extraído: “Prima”, más su determinante: “TipoOficio”.

Por todo lo anterior, podemos decir que una relación estará en 3FN si todos sus atributos son primos o bien si tiene un único atributo no primo ya que, en ambos casos, no es posible la existencia de dependencias transitivas.

La 3FN no maneja satisfactoriamente el caso de una relación que posea dos o más claves candidatas compuestas y solapadas.

Atributos derivados y 3FN

Con los atributos derivados se produce un caso especial de dependencias transitivas debido a que todo atributo derivado depende funcionalmente del atributo (o atributos) del que se deriva.

Si el atributo Y es un atributo derivado del atributo X por cualquier expresión aritmética o lógica, el conocimiento de un valor de X nos lleva a conocer el correspondiente valor de Y, y se cumpliría la dependencia funcional $X \rightarrow Y$.

Sin embargo, este tipo de dependencias no plantea problemas de inconsistencia (si realmente el atributo derivado se calcula automáticamente) y, por tanto, no es necesario tenerlo en cuenta al aplicar la teoría de normalización.

Por ejemplo:

Supongamos la relación **ARTÍCULO** (CodArtículo, Nombre, Precio, Precio_IVA), para una serie de artículos del mismo ramo, y para los que se aplica el mismo tipo de IVA: 7%. El atributo “Precio_IVA” es un atributo derivado de “Precio”, y la regla de derivación sería: $\text{Precio} * 1.07$.

Si estudiamos las dependencias funcionales, vemos que:

DF₁: CodArtículo \rightarrow Nombre

DF₂: CodArtículo \rightarrow Precio

DF₃: CodArtículo \rightarrow Precio_IVA

DF₄: Precio \rightarrow Precio_IVA

En la DF₄ se pone de manifiesto una dependencia transitiva: el determinante “Precio” no es una clave, sino otro atributo no principal. Según esto, la relación no cumpliría la 3FN. Pero, como a quien determina es a un atributo derivado de él, no habría que hacer ninguna transformación y la relación se considera que sí está en 3FN.

5.5 Resumen de las tres primeras formas normales

En muchas ocasiones, el estudio y posterior aplicación de la teoría de la normalización supone ciertas dificultades. Quizás sea debido al formalismo con el que están definidas las formas normales, lo que hace que aparente ser un tema más farragoso de lo que en realidad es.

Por eso, intentando conseguir un enfoque más didáctico, podemos resumir las tres primeras formas normales en una sola oración, parafraseando un dicho que, seguramente, es conocido por todos:

*Una relación está en 3FN si los atributos que no forman parte de la clave
“**dependen de la clave, de toda la clave y nada más que de la clave**”.*

Depender “*de toda la clave*” es lo que viene a decir la 2FN, que supone que exista dependencia funcional completa. Y depender “*nada más que de la clave*” supone que no haya dependencias transitivas, lo que expresa la 3FN.