

4

DISEÑO DE BASES DE DATOS RELACIONALES

<i>PROBLEMAS EN EL DISEÑO DE BASES DE DATOS RELACIONALES</i>	76
<i>FASES DEL DISEÑO DE BASES DE DATOS</i>	77
Recolección y análisis de requerimientos:	78
Diseño conceptual:	78
Diseño lógico de la base de datos (transformación de modelo de datos):	78
Diseño físico de la base de datos:	78
<i>CONCEPTOS DEL MODELO E-R</i>	79
Presentación e historia del modelo:	79
Entidades y atributos:	79
Tipos de entidades:	79
Tipos de atributos:	81
Vínculo o relación:	82
<i>MODELO E-R EXTENDIDO</i>	83
Superclase y subclases:	83
Agregado:	85
<i>REDUCCIÓN DE UN DIAGRAMA E-R A TABLAS</i>	86
<i>APROXIMACIÓN POR DESCOMPOSICIÓN</i>	91
Dependencias Funcionales	91
Claves de una relación	92
<i>PROCESO DE NORMALIZACIÓN DE UNA RELACIÓN</i>	92
Primera forma normal (1FN):	93
Segunda forma normal (2FN):	94
Tercera forma normal (3FN):	94

En general, el objetivo del diseño de una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información. Una de las técnicas para lograrlo consiste en diseñar esquemas que tengan una *forma normal* adecuada. Para determinar si un esquema de relaciones tiene una de las formas normales se requiere mayor información sobre la empresa del "mundo real" que se intenta modelar con la base de datos. La información adicional la proporciona una serie de limitantes que se denominan *dependencias de los datos*.

PROBLEMAS EN EL DISEÑO DE BASES DE DATOS RELACIONALES

Antes, de hablar de formas normales y dependencias de datos es conveniente considerar los defectos que pueden tener una base de datos mal diseñada.

Supongamos las siguientes relaciones:

PERSONA (DNI, NOMBRE, APELLIDOS)

COCHE (MATRICULA, MARCA, TIPO, POTENCIA, COLOR)

TENER (DNI, MATRICULA, FECHA, PRECIO)

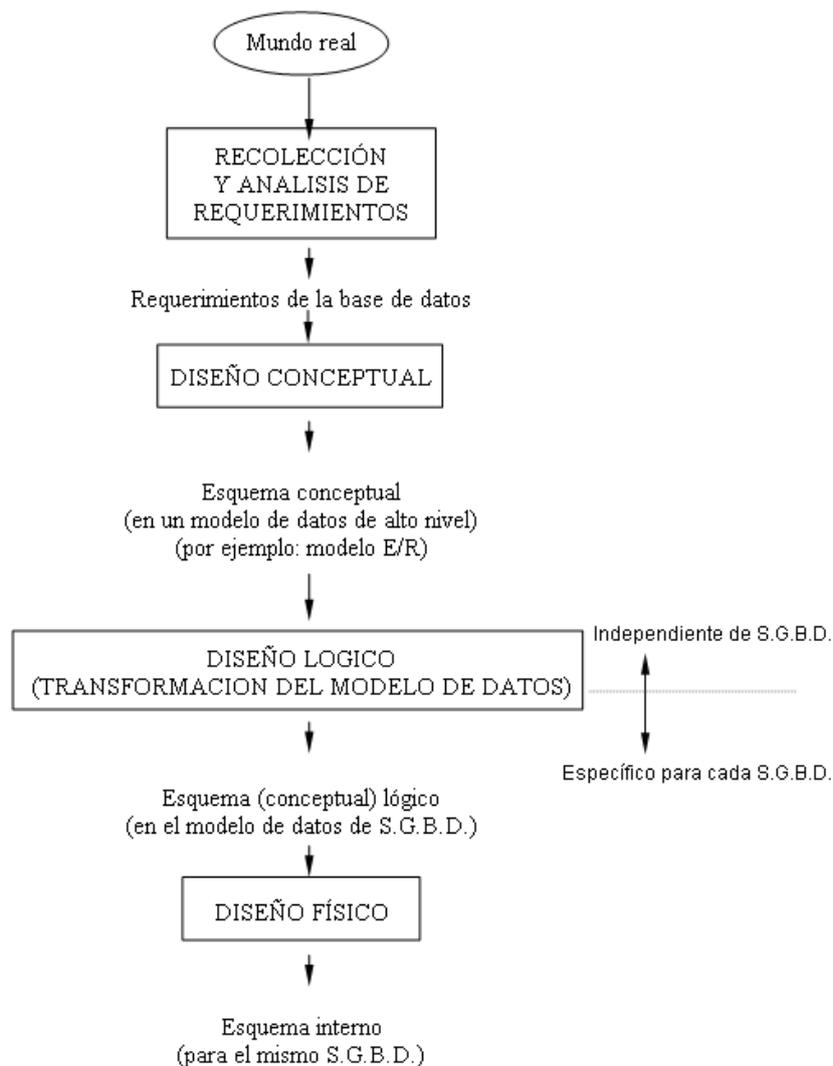
Si en lugar de las anteriores relaciones que componen la BD, optásemos por una única relación, formada por los atributos de las tres, ésta tendría los siguientes defectos:

- En primer lugar, algunos datos serán *redundantes*; en general en esta relación una persona aparecerá tantas veces como coches posea.
- Esta redundancia conlleva unos *riesgos de incoherencia* durante las actualizaciones: por ejemplo, si resulta que el nombre de López no es Pedro sino Juan, hay que tener cuidado y actualizar todas las tuplas en las que aparece López.
- Es preciso admitir la presencia de *valores nulos* en una relación de este tipo para poder mantener en la base, coches sin propietarios o personas que no tienen coches. Si muchos de los atributos no se aplican a todas las tuplas de la relación, acabaremos con un gran número de nulos en esas tuplas. Esto puede originar un considerable desperdicio de espacio de almacenamiento Ej: Si sólo el 10% de los empleados tiene oficinas.

individuales, no se justificará incluir un atributo NUM_OFIC en la relación EMPLEADO; más bien, podríamos crear una relación OFICINAS_EMPL (DNIEMP, NUM_OFIC) contenga exclusivamente tuplas para los empleados con oficinas individuales).

Por lo tanto además de hacerse más complicada la actualización (inserción, eliminación y modificación), se desperdicia espacio. Uno de los objetivos en el diseño de esquemas es *minimizar el espacio de almacenamiento* que ocupan las relaciones base (archivos). La agrupación de atributos en esquemas de relación tiene un efecto significativo sobre el espacio de almacenamiento, se requiere más.

FASES DEL DISEÑO DE BASES DE DATOS



Recolección y análisis de requerimientos:

Los diseñadores entrevistan a los futuros usuarios de la base de datos para recoger y documentar sus necesidades de información. En paralelo, conviene definir los *requerimientos funcionales* que consisten en operaciones (transacciones) que se aplicarán a la base de datos, e incluyen la obtención de datos y la actualización.

Diseño conceptual:

Una vez recogidos todos los requerimientos, el siguiente paso es crear un *esquema conceptual* para la base de datos mediante un modelo de datos conceptual de alto nivel.

El esquema conceptual contiene una descripción detallada de los requerimientos de información de los usuarios, y contiene descripciones de los tipos de datos, relaciones entre ellos y restricciones.

Nosotros utilizaremos para el diseño de esquemas conceptuales el **modelo E-R** (entidad-relación), que describe los datos como entidades, vínculos (relaciones) y atributos.

Diseño lógico de la base de datos (transformación de modelo de datos):

El siguiente paso en el proceso de diseño consiste en implementar de hecho la base de datos con un S.G.B.D. comercial, transformando el modelo conceptual al modelo de datos empleados por el S.G.B.D. (jerárquico, red o relacional).

En nuestro módulo haremos la implementación con un S.G.B.D. relacional, por ser el modelo más utilizado por las empresas en la actualidad.

Diseño físico de la base de datos:

En este paso se especifican las estructuras de almacenamiento internas y la organización de los archivos de la base de datos.

CONCEPTOS DEL MODELO E-R

Presentación e historia del modelo:

El modelo E-R fue propuesto por Peter P. Chen entre los años 1976-1977. Posteriormente otros muchos autores han investigado y escrito sobre el modelo, proporcionando importantes aportaciones, por lo que realmente no se puede considerar que exista un único modelo E-R.

El modelo E-R describe los datos como entidades, relaciones (vínculos) y atributos y permite representar el esquema conceptual de una base de datos de forma gráfica mediante los **diagramas E-R**.

Entidades y atributos:

El objeto básico que se representa en el modelo E-R es la **entidad** que es "cualquier objeto del mundo real con existencia propia, sobre el cual queremos tener información en una base de datos". Una entidad puede ser un objeto con existencia física (una cierta persona, una casa, un empleado, un coche,..) o un objeto con existencia conceptual (una empresa, un puesto de trabajo, un curso universitario,...).

Conjunto de entidades es la totalidad de las entidades del mismo tipo que comparten las mismas propiedades o atributos. En los diagramas E-R se representan mediante un *rectángulo* y dentro del mismo se pone el nombre. Por ejemplo: CLIENTE, PROVEEDOR, ARTICULO, COCHE, etc. Debemos elegir nombres que comuniquen, hasta donde sea posible, el significado de cada entidad. Normalmente se utilizan nombres en singular y no en plural.



EMPLEADO

Tipos de entidades:

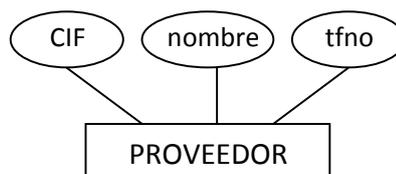
a) **Fuertes (o regulares)**, que son aquellas que tienen existencia por si mismas (Por ejemplo, EMPLEADO). Las entidades fuertes se representan como se ha dicho con un rectángulo con trazo simple.



b) **Débiles**, cuya existencia depende de otro tipo de entidad (Por ejemplo, FAMILIAR depende de EMPLEADO. La desaparición de un empleado de la base de datos hace que desaparezcan también todos los familiares del mismo). Estos tipos de entidades se representan normalmente con un rectángulo con líneas de doble trazo. Estas entidades normalmente no tienen suficientes atributos para formar una clave primaria.



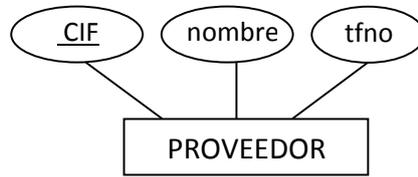
Cada entidad tiene propiedades específicas, llamadas **atributos**, que la describen. Por ejemplo, una entidad PROVEEDOR puede describirse por su C.I.F., su nombre, su teléfono, etc. Los atributos se representan por elipses que están conectadas a su entidad o relación mediante una línea recta.



Al conjunto de valores que puede tomar un atributo se le llama **dominio** del atributo.

Toda entidad debe tener al menos un atributo que permita diferenciar unas entidades particulares de otras, es decir que no toman nunca el mismo valor para dos entidades particulares diferentes. A estos atributos se les llaman **claves**. En el diagrama E-R los

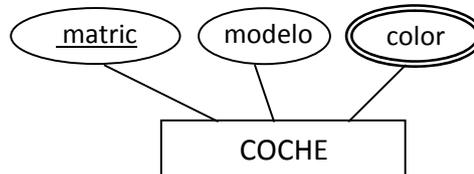
atributos clave deben aparecer destacados; por ejemplo, subrayando su nombre (por ejemplo, cif de la entidad PROVEEDOR).



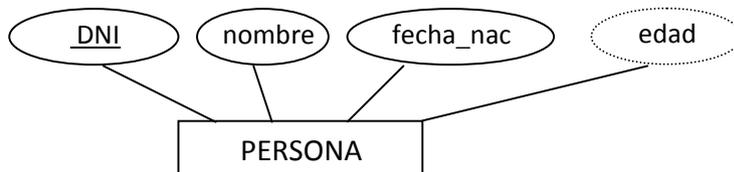
Tipos de atributos:

a) *Simples o compuestos*: Los compuestos están formados por un conjunto de atributos, mientras que los simples no se pueden dividir.

b) *Monovaluados o multivaluados*: Los monovaluados sólo pueden tener un valor para una entidad particular, mientras que los multivaluados pueden tener más de un valor. Los multivaluados se representan mediante una elipse con trazado doble. (Por ejemplo el atributo color de la entidad COCHE es un atributo multivaluado, pues un coche puede estar pintado de varios colores).



c) *Almacenados o derivados*: Los derivados son atributos cuyo valor para una entidad particular puede obtenerse en función de los valores almacenados en otros atributos. Se representan mediante una elipse con trazo discontinuo. (Por ejemplo el atributo edad de la entidad PERSONA es un atributo derivado porque se puede obtener en función del valor de la tributo fecha_nacimiento).



En 1979, Tardieu, propone tres reglas generales que debe cumplir una entidad:

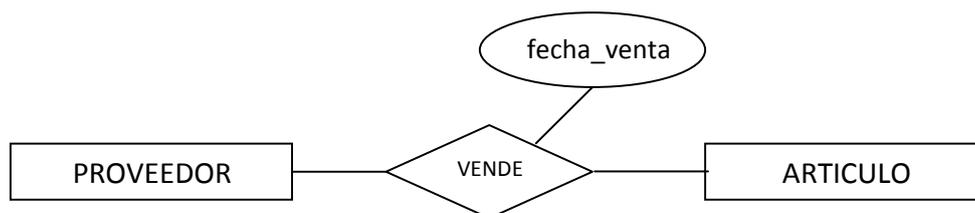
- Tiene que tener existencia propia.
- Cada ocurrencia de un tipo de entidad debe poder distinguirse de las demás.
- Todas las ocurrencias de un tipo de entidad deben tener los mismo tipos de propiedades (atributos).

Vínculo o relación:

Se puede definir como una correspondencia, asociación o conexión entre dos o más entidades. En los diagramas E-R se representa gráficamente como un *rombo* y sus nombres son verbos. Por ejemplo: VENDE, PERTENECE, etc.



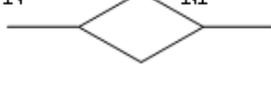
Una relación puede tener *atributos descriptivos*. Por ejemplo, en la relación anterior, podría tener como atributo descriptivo *fecha_venta* (la fecha en que se hace la venta).



Grado de una relación es el número de entidades que participan en la relación. Se puede restringir el modelo E-R para incluir solo conjuntos de relaciones binarias, es decir de grado 2 (es aconsejable).

Correspondencia de cardinalidad, expresa el número máximo de entidades que están relacionadas con una única entidad del otro conjunto de entidades que interviene en la relación. Aunque normalmente nos interesa sólo la cardinalidad máxima, a veces es útil

especificar la cardinalidad mínima. Según su cardinalidad, podemos clasificar las relaciones de los siguientes tipos:

TIPO	RELACIÓN	REPRESENTACIÓN
1:1	Una a una : La cardinalidad máxima en ambas direcciones es 1.	
1:N	Una a muchas: La cardinalidad máxima en una dirección es 1 y en la otra muchos.	
N:M	Muchas a muchas: La cardinalidad máxima en ambas direcciones es muchos.	

Tipos de participación de las entidades en una relación:

- *Opcional (parcial)*: No todas las ocurrencias de una entidad tienen que estar relacionadas con alguna de la otra entidad. Se representa mediante una línea con trazo sencillo. (Por ejemplo, no toda persona posee animales, y no todo animal es posesión de alguna persona. En este caso ambas entidades participan parcialmente en la relación).



- *Obligatoria (total)*: Todas las ocurrencias de una entidad deben estar relacionadas con alguna de la entidad con la que esta relacionada. Se dice también, que existen una *participación total* de ese conjunto de entidades en el conjunto de relaciones, y se representa mediante una línea con trazo doble. (Por ejemplo, todo proveedor tiene que vender algún artículo para serlo, y todo artículo es vendido por algún proveedor. En este caso ambas entidades participan de forma total en la relación).



MODELO E-R EXTENDIDO

El modelo E-R extendido pretende aportar soluciones a requerimientos un tanto más complejos no contemplados en el modelo E-R propuesto por Codd. Así se incorporan al modelo E-R dos nuevos elementos:

Superclase y subclases:

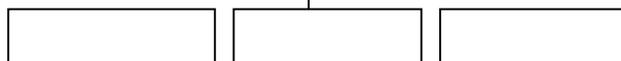
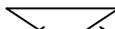
- Una **superclase** es todo tipo entidad sobre el que se definen subclases. Como se trata de entidades, se representa al igual que ellas con un rectángulo.
- Una **subclase** es un subconjunto del tipo entidad que tiene sentido en el minimundo ya que tiene atributos particulares. Como se trata de entidades, se representa al igual que ellas con un rectángulo.

El tipo relación entre una superclase y sus subclases, se dice que es un **tipo ES_UN (IS_A)**. Este tipo relación se representa a diferencia del resto de relaciones con un triángulo. El tipo ES_UN puede ser disjunto (disjunct) o solapado (overlap), es decir, que las entidades pertenecientes a las subclases pueden ser conjuntos disjuntos (es decir, una entidad no puede estar en dos subclases distintas) o conjuntos solapados (es decir, una entidad puede estar en dos o más superclases distintas). Cuando es tipo ES_UN es disjunto se denota con una **d** dentro del triángulo, por el contrario si es solapado se denota con una **o** dentro del triángulo.

SUPERCLASE



RELACIÓN ES_UN

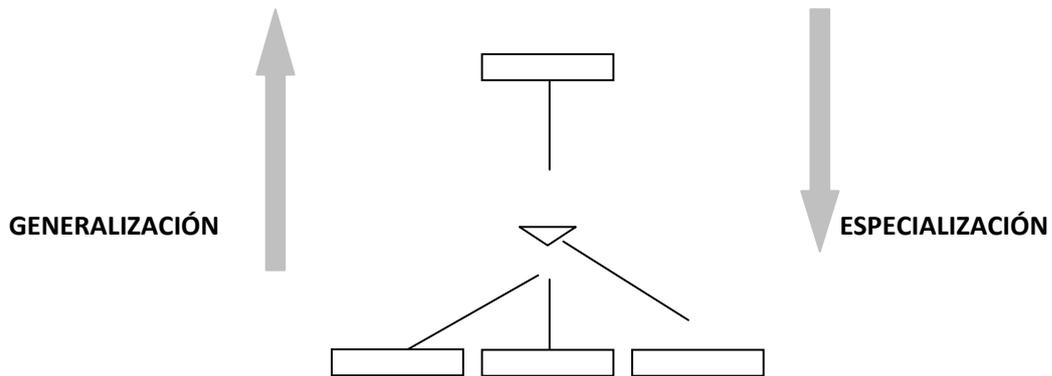


SUBCLASES



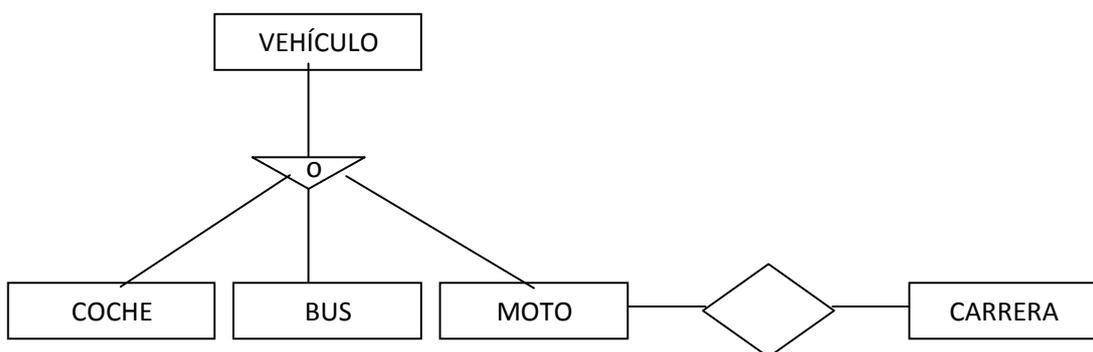
El proceso para determinar la necesidad de usar estos elementos puede producirse en dos sentidos:

- Especialización: es el proceso de definir un conjunto de subclases a partir de un tipo entidad.
- Generalización: es el proceso de suprimir las diferencias entre varios tipo entidad, identificando sus cualidades comunes.



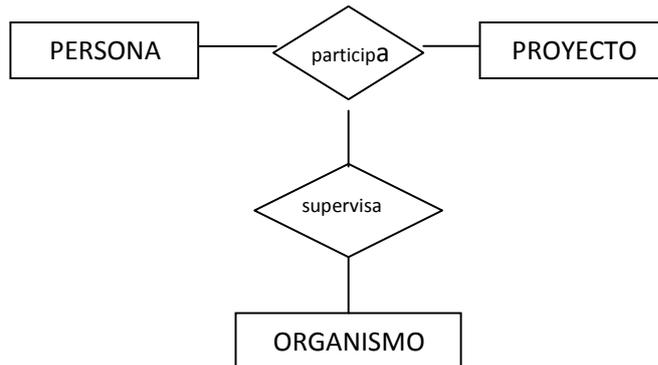
El uso de este elemento está justificado (y recomendado) en dos casos:

- Cuando las subclases tienen atributos particulares que no tiene la superclase.
- Cuando existen tipos relación en los que participan solo algunas subclases.



Agregado:

Es un elemento que nos permite relacionar una relación con otra entidad. Hasta el momento solo se podían relacionar entidades. Este nuevo elemento permite relacionar relaciones con entidades, o relaciones entre sí siempre que el caso lo requiera.

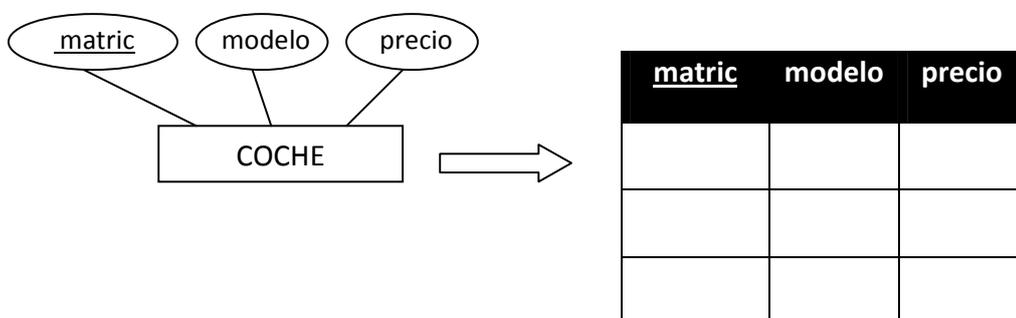


REDUCCIÓN DE UN DIAGRAMA E-R A TABLAS

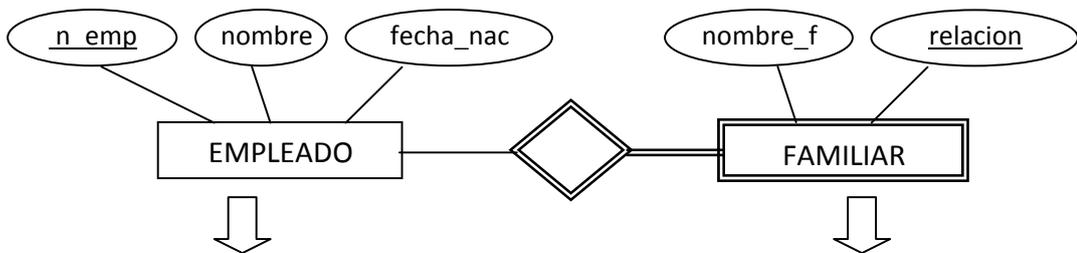
Tanto el modelo E-R, como el modelo de BD relacional son representaciones abstractas y lógicas del desarrollo del mundo real. Debido a que los dos modelos emplean principios de diseño similares, se puede convertir un diseño E-R en un diseño relacional, siguiendo una serie de normas que podemos resumir de la siguiente forma:

a) Para las **ENTIDADES**:

- Se genera una tabla con los atributos de una entidad. La clave primaria de la tabla es la misma que la de la entidad del modelo E-R.



- En el caso de entidades débiles, se genera una tabla con los atributos de la entidad débil, mas la clave primaria de la entidad fuerte. La clave primaria de la tabla generada por la entida débil estara formada por los atributos clave de la entidad débil en el modelo E-R más los atributos clave de la entidad fuerte en el modelo E-R.



EMPLEADO

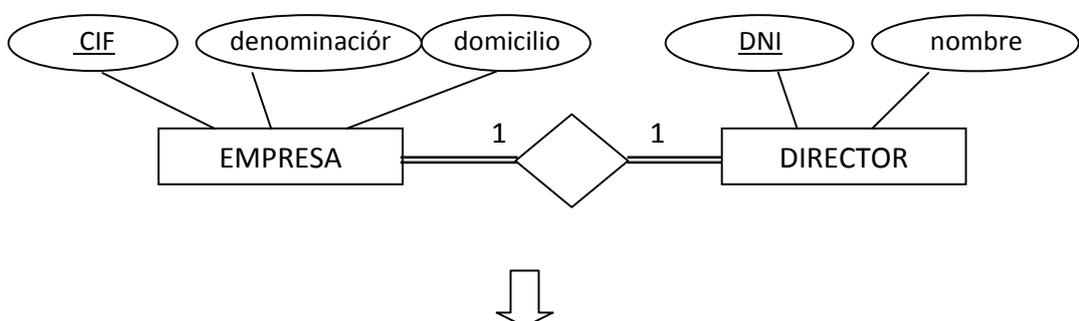
<u>n_emp</u>	nombre	fecha_nac

FAMILIAR

<u>n_emp</u>	<u>relacion</u>	nombre_f

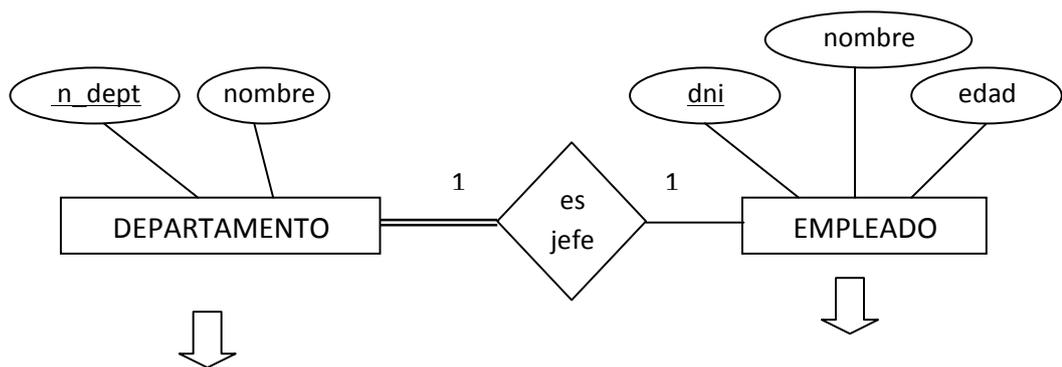
b) Para las **RELACIONES**:

- Si la relación es del tipo 1:1 y es obligatorio (total) tipo de participación de ambas entidades, solo es necesario una tabla con los atributos de las entidades que participan en la relación. Como clave primaria se puede tornar cualquiera de las claves de las entidades.



<u>CIF</u>	denominación	domicilio	DNI	nombre

- Si la relación es del tipo 1:1 y el tipo de participación de una entidad es obligatoria (total) y el de la otra es opcional (parcial), son necesarias dos tablas. Cada una contendrá los atributos de las entidades que participan en la relación. En la tabla correspondiente a la entidad con participación obligatoria se añade una columna que contendrá la clave primaria de la otra entidad (clave ajena). La clave primaria de cada tabla del modelo relacional serán las mismas que las de las entidades asociadas del modelo E-R.



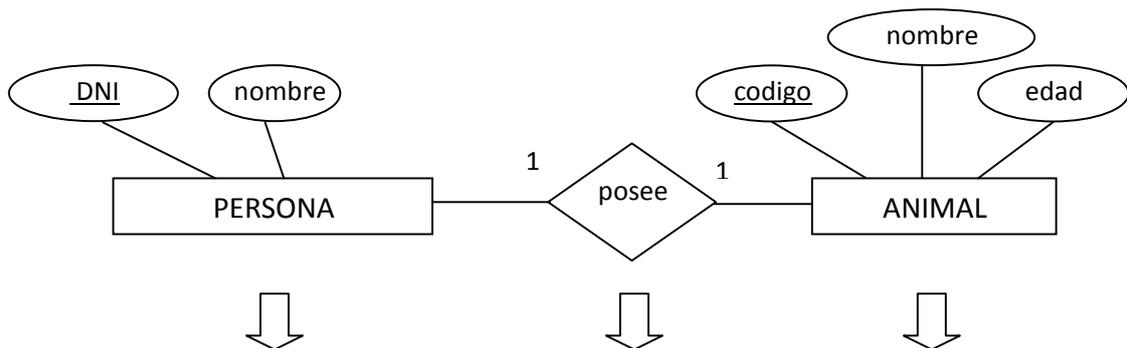
DEPARTAMENTO

<u>n_dept</u>	nombre	dni

EMPLEADO

<u>dni</u>	nombre	edad

Si la relación es del tipo 1:1 y el tipo de participación es opcional (parcial) para las dos entidades, entonces es necesario generar tres tablas, una para cada entidad y otra para la relación que deberá contener como atributos las claves primarias de las entidades que participan en la relación.



PERSONA

<u>DNI</u>	nombre

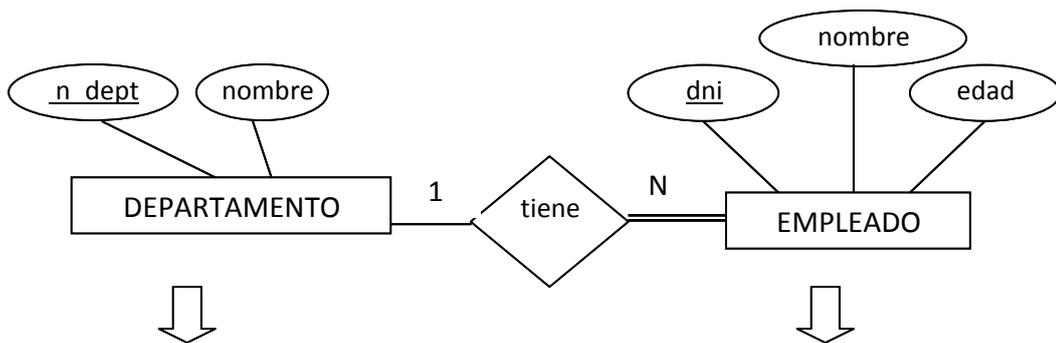
PERSONA_ANIMAL

<u>DNI</u>	<u>codigo</u>

ANIMAL

<u>codigo</u>	nombre	edad

Cuando la relación es del tipo 1:N, y la entidad del lado N es de participación obligatoria (total) se necesita una tabla para cada entidad. A la tabla que representa la entidad N se le añade un atributo que contenga la clave primaria de la entidad con la que se relaciona (clave ajena).



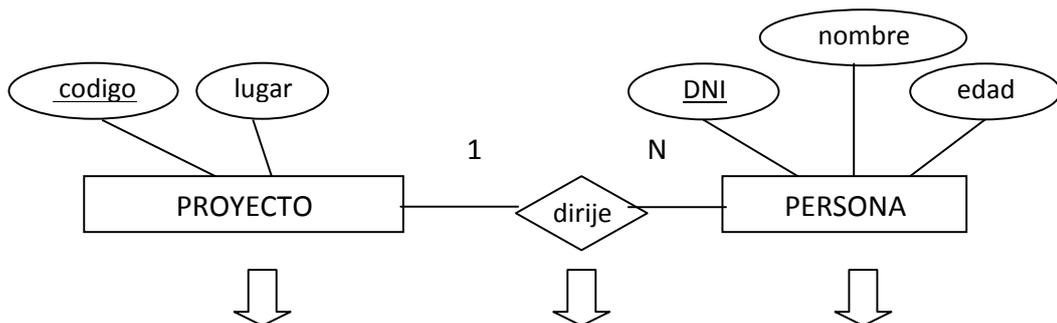
DEPARTAMENTO

<u>n_dept</u>	nombre

EMPLEADO

<u>dni</u>	nombre	edad	N_dept

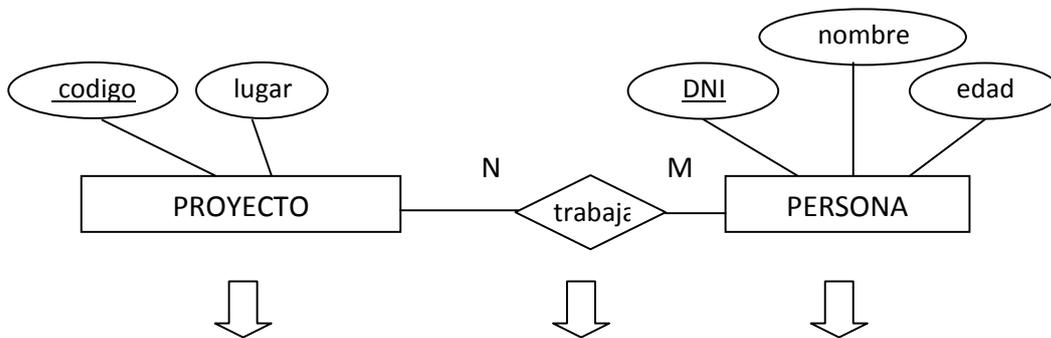
- Cuando la relación es del tipo 1:N, y la entidad del lado N es de participación optativa (parcial) se necesitan tres tablas: una para representar cada entidad y una para representar la relación.



PROYECTO

<u>Codigo</u>	lugar

- Si la relación es del tipo N:M, se generan tres tablas, una para cada entidad y otra que contiene los atributos propios de la relación más la claves primarias de las entidades que participan en la relación.



PROYECTO

<u>Codigo</u>	lugar

TRABAJA

<u>Codigo</u>	<u>DNI</u>

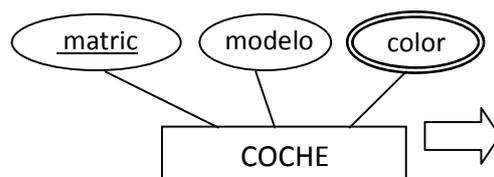
PERSONA

<u>DNI</u>	Nombre	edad

- En general, cuando la relación es entre una entidad fuerte y una entidad débil, no necesita ser representada en forma de tabla.

c) Para **atributos multivaluados**:

- Para. estos atributos se generan tablas separadas, con la clave primaria del conjunto de entidades o relaciones al que pertenecen.



COCHE

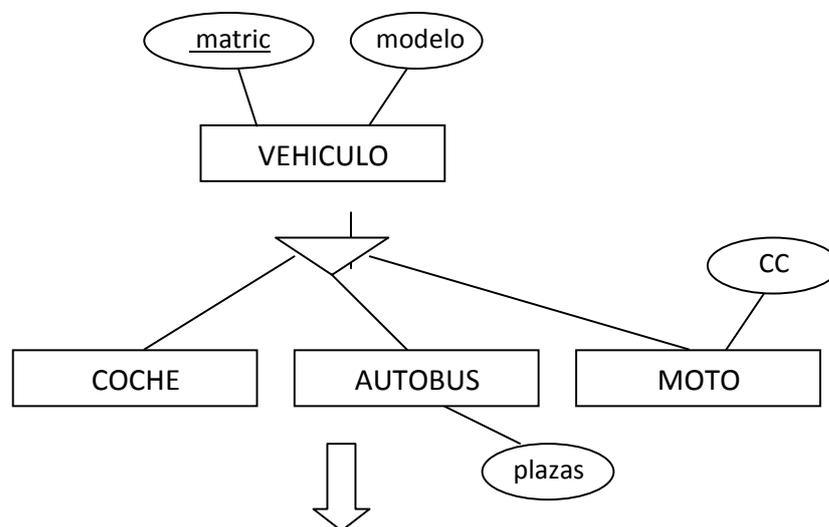
<u>matric</u>	modelo

COCHE_COLOR

<u>matric</u>	<u>color</u>

d) Para la *especialización y generalización*:

- Se crea una tabla para el conjunto de entidades del nivel mas alto.
- Para conjunto de entidades de nivel. mas bajo, se crea una tabla que incluya una columna para cada uno de los atributos de ese conjunto de entidades, mas una columna que contendría la clave primaria del conjunto de entidades de nivel superior.



VEHICULO		COCHE	AUTOBUS	MOTO
<u>matric</u>	modelo	<u>matric</u>	<u>matric</u> plazas	<u>matric</u> CC

APROXIMACIÓN POR DESCOMPOSICIÓN

La aproximación por descomposición para concebir esquemas relacionales parte de una relación compuesta de todos los atributos obtenidos en la fase de recolección y análisis de requerimientos llamada relación *universal* para descomponer después esta relación en un conjunto de relaciones normalizadas. El proceso es un proceso de depuración sucesiva que debe lograr aislar unas entidades y asociaciones del mundo real, evitando que se produzca redundancia de datos

"La *descomposición* consiste, en la sustitución de la relación $R (A_1, A_2, \dots, A_n)$ por una serie de relaciones R_1, R_2, \dots, R_n obtenidas mediante proyecciones de R y tales que la relación resultante de las reuniones de R_1, R_2, \dots, R_n tiene el mismo esquema que R ". La *descomposición se debe hacer sin pérdidas*, es decir, que la extensión de las relaciones normalizadas nos permitan obtener todos los valores de R .

Dependencias Funcionales

Codd introdujo el concepto de *dependencia funcional* para caracterizar aquellas relaciones que pueden descomponerse sin pérdida de informaciones. Se puede definir la dependencia funcional (D.F.) de la siguiente forma:

"Dados dos atributos A y B de una relación R, se dice que B es funcionalmente dependiente de A, si para cada valor de A existe un valor de B, y sólo uno, asociado con él".

En otros términos, se puede decir que si dos tuplas de una relación R tienen el mismo valor en el atributo A deben tener el mismo valor en el atributo B . O dicho de otro modo, si conocemos el valor de A podemos conocer el valor de B . Esto se representa como:

DF: A → B

La notación -> se lee "**determina funcionalmente**".

Por ejemplo, en una relación CLIENTES (Número_cliente, Nombre, Teléfono), existen las siguientes dependencias funcionales:

DF: Número_cliente → Nombre

DF: Número_cliente → Teléfono

Así pues para comenzar el proceso de normalización tenemos que estudiar las propiedades de todos los atributos de la relación y analizar cómo están relacionados entre sí, buscando las posibles dependencias funcionales que existan. Otro de los pasos previos al proceso de normalización es decidir cuál es la clave primaria de la relación.

Claves de una relación

La **clave** de una relación es el conjunto mínimo de atributos que nos permite diferenciar cada fila de una relación de todas las demás. Si la clave está formada por más de un atributo se le llama **clave compuesta**.

En una relación puede que más de un conjunto de atributos puedan ser elegidos como clave. A estos atributos se les llama **claves candidatas** y a la clave candidata elegida como clave de la relación se le llama **clave primaria** y al resto de claves candidatas se les llama **claves secundarias** o alternativas.

Una notación común que se usa para representar el esquema de una relación, como la del ejemplo, es:

Cliente (numero_cliente, nombre, dirección, teléfono)

- *Cliente* es el nombre de la tabla.
- *numero_cliente*, *nombre*, *dirección* y *teléfono* son los atributos (columnas)
- *numero_cliente*, es el nombre de la clave.

También se puede decir que un atributo o conjunto de atributos X es clave, si se cumple:

1. $X \rightarrow A_1, A_2, \dots, A_n$
2. No existe ningún atributo o conjunto de atributos Y, contenido en X, tal que
 $Y \rightarrow A_1, A_2, \dots, A_n$

En otras palabras, una clave es un “conjunto mínimo de atributos que determina a todos los demás”.

PROCESO DE NORMALIZACIÓN DE UNA RELACIÓN

En el proceso de normalización, según la propuesta original de Codd (1972), se somete un 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, segunda y tercera formas normales (1FN, 2FN, 3FN). Posteriormente, Boyce y Codd propusieron una definición más estricta de 3FN, a la que se conoce como forma normal de Boyce-Codd (FNBC). Todas estas formas normales se basan en las dependencias funcionales entre los atributos de una relación. Más adelante se propusieron una cuarta forma normal (4FN) y una quinta (5FN), con fundamento en los conceptos de dependencias multivaluadas y dependencias de reunión, respectivamente.

La *normalización de los datos* puede considerarse como un proceso durante el cual los esquemas de relación que no cumplen las condiciones se descomponen repartiendo sus atributos entre esquemas de relación más pequeños que cumplen las condiciones establecidas. Un objetivo del proceso de normalización es garantizar que no ocurran anomalías de actualización.

Las formas normales, consideradas aparte de otros factores, no garantizan un buen diseño de BD. En general no basta con comprobar por separado que cada esquema de relación de la BD esté en, digamos, FNBC o 3FN. Más bien, el proceso de normalización por descomposición debe confirmar la existencia de propiedades adicionales que los esquemas relacionales, en conjunto, deben poseer. Dos de estas propiedades son:

- La propiedad de reunión sin pérdida, que garantiza que no se presentará el problema de las tuplas erróneas.
- La propiedad de conservación de las dependencias, que asegura que todas las dependencias funcionales estén representadas en alguna de las relaciones individuales resultantes.

La utilidad práctica de las formas normales queda en entredicho cuando las restricciones en las que se basan son difíciles de entender o de detectar por parte de los diseñadores de BD y usuarios que deben descubrir estas restricciones.

Otro punto que merece la pena destacar es que los diseñadores de BD *no tienen que normalizar* hasta la forma normal más alta posible. Las relaciones pueden dejarse en formas normales inferiores por razones de rendimiento (Ej.: la relación EMP-DEPTO (NOMBRE, NSS, FECHAN, DIRECCION, NUMEROD, NOMBRED, NSSJEFED) la dejaríamos así, sin dividir, si por ejemplo una consulta importante obtiene información relativa -al departamento de un empleado, junto con los atributos del empleado. Pero hay que tomar nota de sus anomalías y entenderlas perfectamente de modo que, al actualizar la relación, no se produzcan inconsistencias).

Primera forma normal (1FN):

“Una relación está en primera forma normal (1FN) si los valores para cada atributo de la relación son atómicos”.

Esto quiere decir simplemente que cada atributo sólo puede pertenecer a un dominio (es indivisible) y que tiene un valor único para cada fila.

La primera forma normal se definió para prohibir los atributos multivaluados, compuestos y sus combinaciones.

Cuando una relación no está en primera forma normal, se divide en otras relaciones, repartiendo sus atributos entre las resultantes. Normalmente la idea es eliminar el atributo que viola la 1ª FN de la relación original y colocarlo en una relación aparte junto con la clave primaria de la relación de partida.

Segunda forma normal (2FN):

"Una relación está en segunda a normal si está en la 1ª FN y todos los atributos no clave dependen de la clave completa y no sólo de una parte de esta".

Este paso sólo se aplica a relaciones que tienen claves compuestas, es decir, que están formadas por más de un atributo. Si un esquema de relación no está en 2ªFN, se le puede normalizar a varias relaciones en 2ªFN en las que los atributos que dependen de una parte de la clave formarán una nueva relación que tendrá esa parte de la clave como clave primaria.

Tercera forma normal (3FN):

"Una relación está en tercera forma normal si todos los atributos de la relación dependen funcionalmente sólo de la clave, y no de ningún otro atributo".

Esto significa que en una relación en 3FN, para toda DF: $X \rightarrow Y$, X es una clave.

Podemos observar que si una relación está en tercera forma normal, está también en segunda forma normal, sin embargo lo inverso no siempre es cierto.

Nota: Los conceptos del modelo ER (entidades, relaciones, atributos, claves y restricciones estructurales) que hemos visto son suficientes para modelar algunas aplicaciones sencillas de bases de datos, sin embargo para algunas aplicaciones es necesario utilizar conceptos adicionales si lo que se quiere es un modelado más exacto