

Introducción al diseño de bases de datos

Dolors Costal Costa

P06/M2109/02150

Índice


Introducción	5
Objetivos	5
1. Introducción al diseño de bases de datos	7
1.1. Etapas del diseño de bases de datos.....	7
2. Diseño conceptual: el modelo ER	10
2.1. Construcciones básicas.....	11
2.1.1. Entidades, atributos e interrelaciones.....	11
2.1.2. Grado de las interrelaciones.....	13
2.1.3. Interrelaciones binarias.....	16
2.1.4. Ejemplo: base de datos de casas de colonias.....	18
2.1.5. Interrelaciones <i>n</i> -arias	21
2.1.6. Interrelaciones recursivas.....	23
2.1.7. Entidades débiles.....	25
2.2. Extensiones del modelo ER	26
2.2.1. Generalización/especialización.....	26
2.2.2. Entidades asociativas.....	28
2.3. Ejemplo: base de datos del personal de una entidad bancaria	30
3. Diseño lógico: la transformación del modelo ER al modelo relacional	35
3.1. Introducción a la transformación de entidades e interrelaciones.....	35
3.2. Transformación de entidades	35
3.3. Transformación de interrelaciones binarias.....	36
3.3.1. Conectividad 1:1	36
3.3.2. Conectividad 1:N	37
3.3.3. Conectividad M:N.....	38
3.3.4. Influencia de la dependencia de existencia en la transformación de las interrelaciones binarias	39
3.4. Transformación de interrelaciones ternarias.....	40
3.4.1. Conectividad M:N:P.....	40
3.4.2. Conectividad M:N:1	41
3.4.3. Conectividad N:1:1	42
3.4.4. Conectividad 1:1:1	43
3.5. Transformación de interrelaciones <i>n</i> -arias	44
3.6. Transformación de interrelaciones recursivas.....	44
3.7. Transformación de entidades débiles.....	46
3.8. Transformación de la generalización/especialización.....	47

3.9. Transformación de entidades asociativas.....	48
3.10. Resumen de la transformación del modelo ER al modelo relacional	49
3.11. Ejemplo: base de datos del personal de una entidad bancaria	49
Resumen.....	51
Ejercicios de autoevaluación	53
Solucionario.....	55
Glosario.....	59
Bibliografía.....	60

Introducción al diseño de bases de datos

Introducción

En otras unidades didácticas se estudian las bases de datos relacionales y un lenguaje relacional, SQL, que nos proporciona mecanismos para crear, actualizar y consultar estas bases de datos.

Es necesario complementar estos conocimientos con un aspecto que es fundamental para poder utilizar adecuadamente la tecnología de las bases de datos relacionales: el diseño. Éste será el objeto de estudio de esta unidad, que tratará el **diseño de bases de datos** para el caso específico del modelo relacional. 

Concretamente, en esta unidad explicaremos en qué consiste el diseño de una base de datos, analizaremos las etapas en las que se puede descomponer y describiremos con detalle las etapas del diseño conceptual y lógico de una base de datos relacional.

Objetivos

En los materiales didácticos de esta unidad encontraréis las herramientas indispensables para alcanzar los siguientes objetivos:

1. Conocer las etapas que integran el proceso del diseño de una base de datos.
2. Conocer las estructuras del modelo ER.
3. Saber hacer el diseño conceptual de los datos de un sistema de información mediante el modelo ER.
4. Saber hacer el diseño lógico de una base de datos relacional partiendo de un diseño conceptual expresado con el modelo ER.

1. Introducción al diseño de bases de datos

En otras unidades hemos aprendido cómo es una base de datos relacional y hemos estudiado un lenguaje, el SQL, que nos proporciona mecanismos para crear estas bases de datos, así como para actualizarlas y consultarlas.

Recordad que las bases de datos relacionales y los lenguajes SQL se han estudiado en las unidades "El modelo relacional y el álgebra relacional" y "El lenguaje SQL", respectivamente.

Sin embargo, todavía debemos resolver algunas cuestiones fundamentales para poder emplear la tecnología de las bases de datos relacionales; por ejemplo, cómo se puede decidir qué relaciones debe tener una base de datos determinada o qué atributos deben presentar las relaciones, qué claves primarias y qué claves foráneas se deben declarar, etc. La tarea de tomar este conjunto de decisiones recibe el nombre de *diseñar la base de datos*.

Una base de datos sirve para almacenar la información que se utiliza en un sistema de información determinado. Las necesidades y los requisitos de los futuros usuarios del sistema de información se deben tener en cuenta para poder tomar adecuadamente las decisiones anteriores.

En resumen, el **diseño de una base de datos** consiste en definir la estructura de los datos que debe tener la base de datos de un sistema de información determinado. En el caso relacional, esta estructura será un conjunto de esquemas de relación con sus atributos, dominios de atributos, claves primarias, claves foráneas, etc.

Si recordáis los tres mundos presentados –el real, el conceptual y el de las representaciones–, observaréis que el diseño de una base de datos consiste en la obtención de una representación informática concreta a partir del estudio del mundo real de interés. !

1.1. Etapas del diseño de bases de datos

El diseño de una base de datos no es un proceso sencillo. Habitualmente, la complejidad de la información y la cantidad de requisitos de los sistemas de información hacen que sea complicado. Por este motivo, cuando se diseñan bases de datos es interesante aplicar la vieja estrategia de dividir para vencer.

Por lo tanto, conviene descomponer el proceso del diseño en varias etapas; en cada una se obtiene un resultado intermedio que sirve de punto de partida de la etapa siguiente, y en la última etapa se obtiene el resultado deseado. De este modo no hace falta resolver de golpe toda la problemática que plantea el diseño, sino que en cada etapa se afronta un solo tipo de subproblema. Así se divide el problema y, al mismo tiempo, se simplifica el proceso.

Descompondremos el diseño de bases de datos en tres etapas: !

1) **Etapla del diseño conceptual:** en esta etapa se obtiene una estructura de la información de la futura BD independiente de la tecnología que hay que emplear. No se tiene en cuenta todavía qué tipo de base de datos se utilizará –relacional, orientada a objetos, jerárquica, etc.–; en consecuencia, tampoco se tiene en cuenta con qué SGBD ni con qué lenguaje concreto se implementará la base de datos. Así pues, la etapa del diseño conceptual nos permite concentrarnos únicamente en la problemática de la estructuración de la información, sin tener que preocuparnos al mismo tiempo de resolver cuestiones tecnológicas.

El resultado del diseño conceptual

Si retomamos la idea de los tres mundos, podemos afirmar que la etapa del diseño conceptual obtiene un resultado que se sitúa en el mundo de las concepciones, y no en el mundo de las representaciones.

El resultado de la etapa del diseño conceptual se expresa mediante algún modelo de datos de alto nivel. Uno de los más empleados es el **modelo entidad-interrelación** (*entity-relationship*), que abreviaremos con la sigla ER.

La forma de elaborar un diseño conceptual expresado con el modelo ER se explica en el apartado 2 de esta unidad. !

2) **Etapla del diseño lógico:** en esta etapa se parte del resultado del diseño conceptual, que se transforma de forma que se adapte a la tecnología que se debe emplear. Más concretamente, es preciso que se ajuste al modelo del SGBD con el que se desea implementar la base de datos. Por ejemplo, si se trata de un SGBD relacional, esta etapa obtendrá un conjunto de relaciones con sus atributos, claves primarias y claves foráneas.

El resultado del diseño lógico

El resultado del diseño lógico se sitúa ya en el mundo de las representaciones.

Esta etapa parte del hecho de que ya se ha resuelto la problemática de la estructuración de la información en un ámbito conceptual, y permite concentrarnos en las cuestiones tecnológicas relacionadas con el modelo de base de datos.

Más adelante explicaremos cómo se hace el diseño lógico de una base de datos relacional, tomando como punto de partida un diseño conceptual expresado con el modelo ER; es decir, veremos cómo se puede transformar un modelo ER en un modelo relacional.

El diseño lógico de una base de datos relacional se explica en el apartado 3 de esta unidad didáctica. !

3) **Etapla del diseño físico:** en esta etapa se transforma la estructura obtenida en la etapa del diseño lógico, con el objetivo de conseguir una mayor eficiencia; además, se completa con aspectos de implementación física que dependerán del SGBD.


El resultado del diseño físico

El resultado de la etapa del diseño físico se sitúa en el mundo de las representaciones, al igual que el resultado de la etapa del diseño lógico. La diferencia con respecto a la etapa anterior es que ahora se tienen en cuenta aspectos de carácter más físico del mundo de las representaciones.

Por ejemplo, si se trata de una base de datos relacional, la transformación de la estructura puede consistir en lo siguiente: tener almacenada alguna relación que sea la combinación de varias relaciones que se han obtenido en la etapa del diseño lógico, partir una relación en varias, añadir algún atributo calculable a una relación, etc. Los aspectos de implementación física que hay que completar consisten normalmente en la elección de estructuras físicas de implementación de las relaciones, la selección del tamaño de las memorias intermedias (*buffers*) o de las páginas, etc.

En la etapa del diseño físico –con el objetivo de conseguir un buen rendimiento de la base de datos–, se deben tener en cuenta las características de los procesos que consultan y actualizan la base de datos, como por ejemplo los caminos de acceso que utilizan y las frecuencias de ejecución. También es necesario considerar los volúmenes que se espera tener de los diferentes datos que se quieren almacenar.

2. Diseño conceptual: el modelo ER

En este apartado trataremos el diseño conceptual de una base de datos mediante el modelo ER. Lo que explicaremos es aplicable al diseño de cualquier tipo de bases de datos –relacional, jerárquica, etc.–, porque, como ya hemos dicho, en la etapa del diseño conceptual todavía no se tiene en cuenta la tecnología concreta que se utilizará para implementar la base de datos. 

El **modelo ER** es uno de los enfoques de modelización de datos que más se utiliza actualmente por su simplicidad y legibilidad. Su legibilidad se ve favorecida porque proporciona una notación diagramática muy comprensiva. Es una herramienta útil tanto para ayudar al diseñador a reflejar en un modelo conceptual los requisitos del mundo real de interés como para comunicarse con el usuario final sobre el modelo conceptual obtenido y, de este modo, poder verificar si satisface sus requisitos.

El modelo ER resulta fácil de aprender y de utilizar en la mayoría de las aplicaciones. Además, existen herramientas informáticas de ayuda al diseño (herramientas CASE*) que utilizan alguna variante del modelo ER para hacer el diseño de los datos.

El nombre completo del modelo ER es *entity-relationship*, y proviene del hecho de que los principales elementos que incluye son las entidades y las interrelaciones (*entities* y *relationships*). Traduiremos este nombre por ‘entidad-interrelación’.

El origen del modelo ER se encuentra en trabajos efectuados por Peter Chen en 1976. Posteriormente, muchos otros autores han descrito variantes y/o extensiones de este modelo. Así pues, en la literatura se encuentran muchas formas diferentes del modelo ER que pueden variar simplemente en la notación diagramática o en algunos de los conceptos en que se basan para modelizar los datos.

Cuando se quiere utilizar el modelo ER para comunicarse con el usuario, es recomendable emplear una variante del modelo que incluya sólo sus elementos más simples –entidades, atributos e interrelaciones– y, tal vez, algunas construcciones adicionales, como por ejemplo entidades débiles y dependencias de existencia. Éstos eran los elementos incluidos en el modelo original propuesto por Chen. En cambio, para llevar a cabo la tarea de modelizar propiamente dicha, suele ser útil usar un modelo ER más completo que incluya construcciones más avanzadas que extienden el modelo original.

Según la noción de *modelo de datos* que hemos utilizado en los otros módulos, un modelo de datos tiene en cuenta tres aspectos de los datos: la estructura, la manipulación y la integridad. Sin embargo, el modelo ER habitualmente se

* La sigla CASE corresponde al término inglés *Computer Aided Software Engineering*.

El modelo entidad-interrelación

Algunos autores denominan *entidad-relación* al modelo ER, pero en nuestro caso hemos preferido traducir *relationship* por ‘interrelación’ y no por ‘relación’, con el objetivo de evitar confusiones entre este concepto y el de *relación* que se utiliza en el modelo relacional.

Recordad el modelo relacional, que se ha estudiado en la unidad “El modelo relacional y el álgebra relacional”.



utiliza para reflejar aspectos de la estructura de los datos y de su integridad, pero no de su manipulación. !

2.1. Construcciones básicas

2.1.1. Entidades, atributos e interrelaciones

Por **entidad** entendemos un objeto del mundo real que podemos distinguir del resto de objetos y del que nos interesan algunas propiedades.

Ejemplos de entidad

Algunos ejemplos de entidad son un empleado, un producto o un despacho. También son entidades otros elementos del mundo real de interés, menos tangibles pero igualmente diferenciables del resto de objetos; por ejemplo, una asignatura impartida en una universidad, un préstamo bancario, un pedido de un cliente, etc.

Las propiedades de los objetos que nos interesan se denominan **atributos**.

Ejemplos de atributo

Sobre una entidad *empleado* nos puede interesar, por ejemplo, tener registrados su DNI, su NSS, su nombre, su apellido y su sueldo como atributos.

El término *entidad* se utiliza tanto para denominar objetos individuales como para hacer referencia a conjuntos de objetos similares de los que nos interesan los mismos atributos; es decir, que, por ejemplo, se utiliza para designar tanto a un empleado concreto de una empresa como al conjunto de todos los empleados de la empresa. Más concretamente, el término *entidad* se puede referir a **instancias u ocurrencias concretas** (empleados concretos) o a **tipos o clases de entidades** (el conjunto de todos los empleados).

El modelo ER proporciona una **notación diagramática** para representar gráficamente las entidades y sus atributos: !

- Las **entidades** se representan con un rectángulo. El nombre de la entidad se escribe en mayúsculas dentro del rectángulo.
- Los **atributos** se representan mediante su nombre en minúsculas unido con un guión al rectángulo de la entidad a la que pertenecen. Muchas veces, dado que hay muchos atributos para cada entidad, se listan todos aparte del diagrama para no complicarlo.

Cada uno de los atributos de una entidad toma valores de un cierto dominio o conjunto de valores. Los valores de los dominios deben ser atómicos; es decir,



Notación diagramática de entidades y atributos

La figura muestra la notación diagramática para el caso de una entidad *empleado* con los atributos *dni*, *nss*, *nombre*, *apellido* y *sueldo*.

no deben poder ser descompuestos. Además, todos los atributos tienen que ser univaluados. Un atributo es univaluado si tiene un único valor para cada ocurrencia de una entidad.

Recordad que los valores de los atributos de las relaciones también deben ser atómicos, tal y como se ha explicado en la unidad "El modelo relacional y el álgebra relacional".

Ejemplo de atributo univaluado

El atributo *suelto* de la entidad *empleado*, por ejemplo, toma valores del dominio de los reales y únicamente toma un valor para cada empleado concreto; por lo tanto, ningún empleado puede tener más de un valor para el sueldo.

Como ya hemos comentado anteriormente, una entidad debe ser distinguible del resto de objetos del mundo real. Esto hace que para toda entidad sea posible encontrar un conjunto de atributos que permitan identificarla. Este conjunto de atributos forma una **clave de la entidad**.

Ejemplo de clave

La entidad *empleado* tiene una clave que consta del atributo *dni* porque todos los empleados tienen números de DNI diferentes.

Una determinada entidad puede tener más de una clave; es decir, puede tener varias **claves candidatas**.

Ejemplo de clave candidata

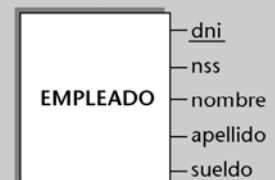
La entidad *empleado* tiene dos claves candidatas, la que está formada por el atributo *dni* y la que está constituida por el atributo *nss*, teniendo en cuenta que el NSS también será diferente para cada uno de los empleados.

Los conceptos de *clave candidata* y *clave primaria* de una entidad son similares a los conceptos de *clave candidata* y *clave primaria* de una relación, que hemos estudiado en la unidad "El modelo relacional y el álgebra relacional".

El diseñador elige una **clave primaria** entre todas las claves candidatas. En la notación diagramática, la clave primaria se subraya para distinguirla del resto de las claves. !

Ejemplo de clave primaria

En el caso de la entidad *empleado*, podemos elegir *dni* como clave primaria. En la figura del margen vemos que la clave primaria se subraya para distinguirla del resto.



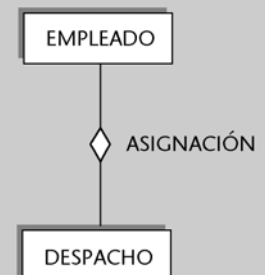
Se define **interrelación** como una asociación entre entidades.

Las interrelaciones se representan en los diagramas del modelo ER mediante un rombo. Junto al rombo se indica el nombre de la interrelación con letras mayúsculas. !

Ejemplo de interrelación

Consideremos una entidad *empleado* y una entidad *despacho* y supongamos que a los empleados se les asignan despachos donde trabajar. Entonces hay una interrelación entre la entidad *empleado* y la entidad *despacho*.

Esta interrelación, que podríamos denominar *asignación*, asocia a los empleados con los despachos donde trabajan. La figura del margen muestra la interrelación *asignación* entre las entidades *empleado* y *despacho*.




El término *interrelación* se puede utilizar tanto para denominar asociaciones concretas u ocurrencias de asociaciones como para designar conjuntos o clases de asociaciones similares.

Ejemplo

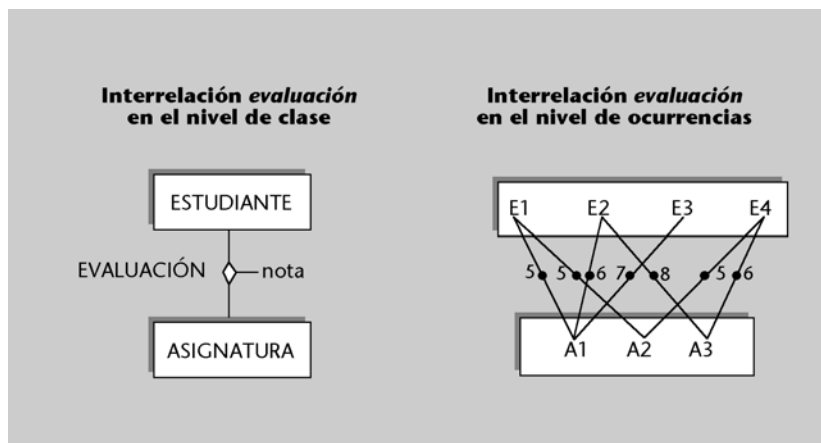
Una interrelación se aplica tanto a una asociación concreta entre el empleado de DNI '50.455.234' y el despacho 'Diagonal, 20' como a la asociación genérica entre la entidad *empleado* y la entidad *despacho*.

En ocasiones interesa reflejar algunas propiedades de las interrelaciones. Por este motivo, las interrelaciones pueden tener también atributos. Los **atributos de las interrelaciones**, igual que los de las entidades, tienen un cierto dominio, deben tomar valores atómicos y deben ser univaluados.

Los atributos de las interrelaciones se representan mediante su nombre en minúsculas unido con un guión al rombo de la interrelación a la que pertenecen. 

Ejemplo de atributo de una interrelación

Observemos la entidad *estudiante* y la entidad *asignatura* que se muestran en la figura siguiente:



Entre estas dos entidades se establece la interrelación *evaluación* para indicar de qué asignaturas han sido evaluados los estudiantes. Esta interrelación tiene el atributo *nota*, que sirve para especificar qué nota han obtenido los estudiantes de las asignaturas evaluadas.

Conviene observar que el atributo *nota* deber ser forzosamente un atributo de la interrelación *evaluación*, y que no sería correcto considerarlo un atributo de la entidad *estudiante* o un atributo de la entidad *asignatura*. Lo explicaremos analizando las ocurrencias de la interrelación *evaluación* que se muestran en la figura anterior.

Si *nota* se considerase un atributo de *estudiante*, entonces para el estudiante 'E1' de la figura necesitaríamos dos valores del atributo, uno para cada asignatura que tiene el estudiante; por lo tanto, no sería univaluado. De forma similar, si *nota* fuese atributo de *asignatura* tampoco podría ser univaluado porque, por ejemplo, la asignatura 'A1' requeriría tres valores de nota, una para cada estudiante que se ha matriculado en ella. Podemos concluir que el atributo *nota* está relacionado al mismo tiempo con una asignatura y con un estudiante que la cursa y que, por ello, debe ser un atributo de la interrelación que asocia las dos entidades.

2.1.2. Grado de las interrelaciones

Una interrelación puede asociar dos o más entidades. El número de entidades que asocia una interrelación es el **grado de la interrelación**.

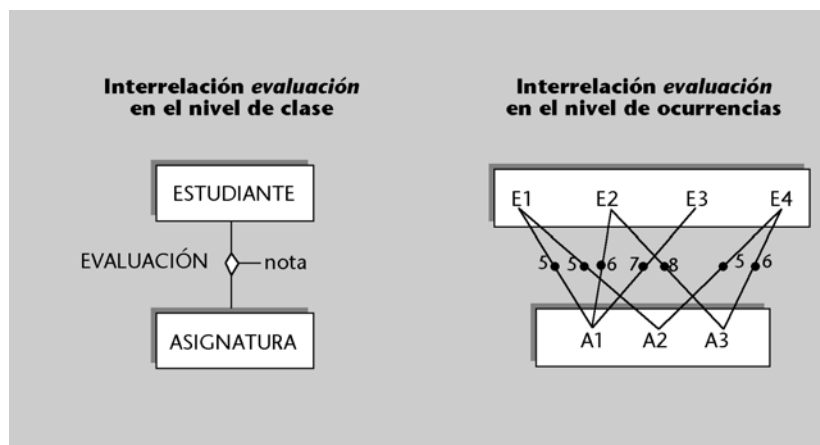
Interrelaciones de grado dos

Las interrelaciones *evaluación* y *asignación* de los ejemplos anteriores tienen grado dos:

- La interrelación *evaluación* asocia la entidad *estudiante* y la entidad *asignatura*; es decir, asocia dos entidades.
- De forma análoga, la interrelación *asignación* asocia *empleado* y *despacho*.

Las interrelaciones de grado dos se denominan también **interrelaciones binarias**. Todas las interrelaciones de grado mayor que dos se denominan, en conjunto, **interrelaciones *n*-arias**. Así pues, una interrelación *n*-aria puede tener grado tres y ser una interrelación ternaria, puede tener grado cuatro y ser una interrelación cuaternaria, etc.

A continuación presentaremos un ejemplo que nos ilustrará el hecho de que, en ocasiones, las interrelaciones binarias no nos permiten modelizar correctamente la realidad y es necesario utilizar interrelaciones de mayor grado.

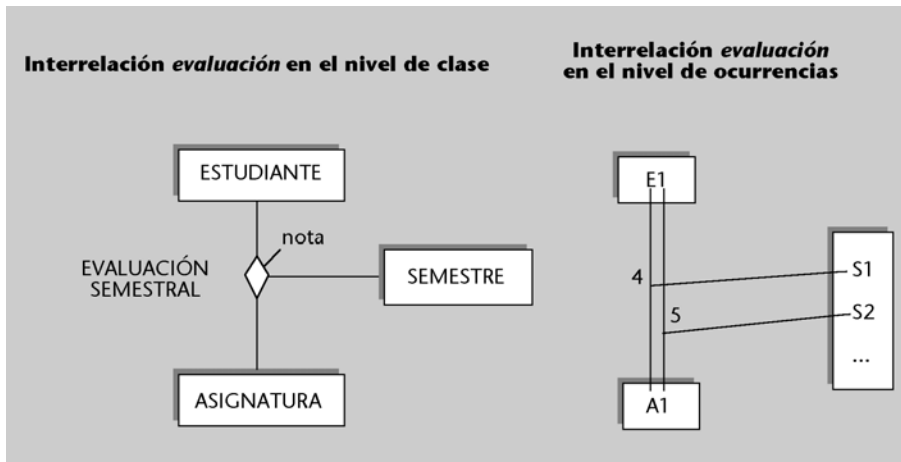


Consideremos la interrelación *evaluación* de la figura anterior, que tiene un atributo *nota*. Este atributo permite registrar la nota obtenida por cada estudiante en cada asignatura de la que ha sido evaluado. Una interrelación permite establecer una sola asociación entre unas entidades individuales determinadas. En otras palabras, sólo se puede interrelacionar una vez al estudiante 'E1' con la asignatura 'A1' vía la interrelación *evaluación*.

Observad que, si pudiese haber más de una interrelación entre el estudiante 'E1' y la asignatura 'A1', no podríamos distinguir estas diferentes ocurrencias de la interrelación. Esta restricción hace que se registre una sola nota por estudiante y asignatura.

Supongamos que deseamos registrar varias notas por cada asignatura y estudiante correspondientes a varios semestres en los que un mismo estudiante ha cursado una asignatura determinada (desgraciadamente, algunos estudiantes tienen que cursar una asignatura varias veces antes de aprobarla). La interrelación

anterior no nos permitiría reflejar este caso. Sería necesario aumentar el grado de la interrelación, tal y como se muestra en la figura siguiente:



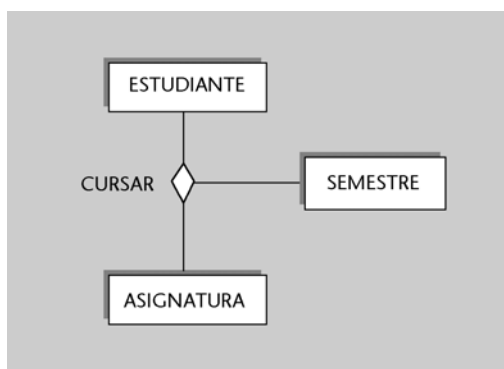
La interrelación ternaria *evaluación-semesteral* asocia estudiantes, asignaturas y una tercera entidad que denominamos *semester*. Su atributo *nota* nos permite reflejar todas las notas de una asignatura que tiene un estudiante correspondientes a diferentes semestres.

De hecho, lo que sucede en este caso es que, según los requisitos de los usuarios de esta BD, una nota pertenece al mismo tiempo a un estudiante, a una asignatura y a un semestre y, lógicamente, debe ser un atributo de una interrelación ternaria entre estas tres entidades.

Este ejemplo demuestra que una interrelación binaria puede no ser suficiente para satisfacer los requisitos de los usuarios, y puede ser necesario aplicar una interrelación de mayor grado. Conviene observar que esto también puede ocurrir en interrelaciones que no tienen atributos. **!**

Ejemplo de interrelación ternaria sin atributos

Consideremos un caso en el que deseamos saber para cada estudiante qué asignaturas ha cursado cada semestre, a pesar de que no queremos registrar la nota que ha obtenido. Entonces aplicaríamos también una interrelación ternaria entre las entidades *estudiante*, *asignatura* y *semester* que no tendría atributos, tal y como se muestra en la figura siguiente:



Hemos analizado casos en los que era necesario utilizar interrelaciones ternarias para poder modelizar correctamente ciertas situaciones de interés del mundo

real. Es preciso remarcar que, de forma similar, a veces puede ser necesario utilizar interrelaciones de grado todavía mayor: cuaternarias, etc. !

En el subapartado siguiente analizaremos con detalle las interrelaciones binarias, y más adelante, las interrelaciones n -arias.

Las relaciones n -arias se analizan en el subapartado 2.1.4 de esta unidad didáctica.

2.1.3. Interrelaciones binarias

Conectividad de las interrelaciones binarias

La **conectividad de una interrelación** expresa el tipo de correspondencia que se establece entre las ocurrencias de entidades asociadas con la interrelación. En el caso de las interrelaciones binarias, expresa el número de ocurrencias de una de las entidades con las que una ocurrencia de la otra entidad puede estar asociada según la interrelación.

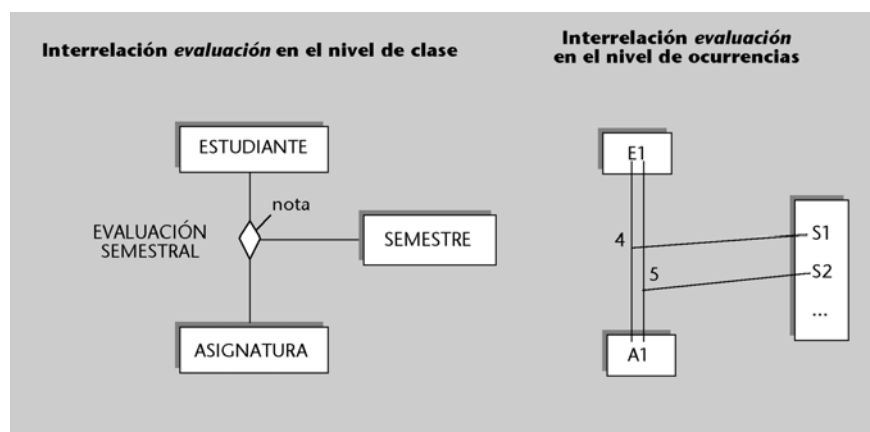
Una interrelación binaria entre dos entidades puede tener tres tipos de conectividad: !

- **Conectividad uno a uno (1:1).** La conectividad 1:1 se denota poniendo un 1 a lado y lado de la interrelación.
- **Conectividad uno a muchos (1:N).** La conectividad 1:N se denota poniendo un 1 en un lado de la interrelación y una N en el otro.
- **Conectividad muchos a muchos: (M:N).** La conectividad M:N se denota poniendo una M en uno de los lados de la interrelación, y una N en el otro.

Ejemplos de conectividad en una interrelación binaria

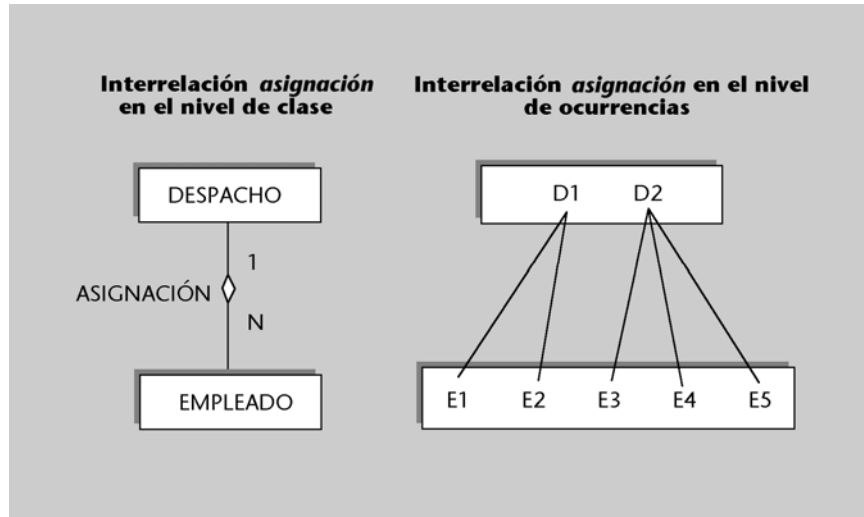
A continuación analizaremos un ejemplo de cada una de las conectividades posibles para una interrelación binaria:

a) Conectividad 1:1



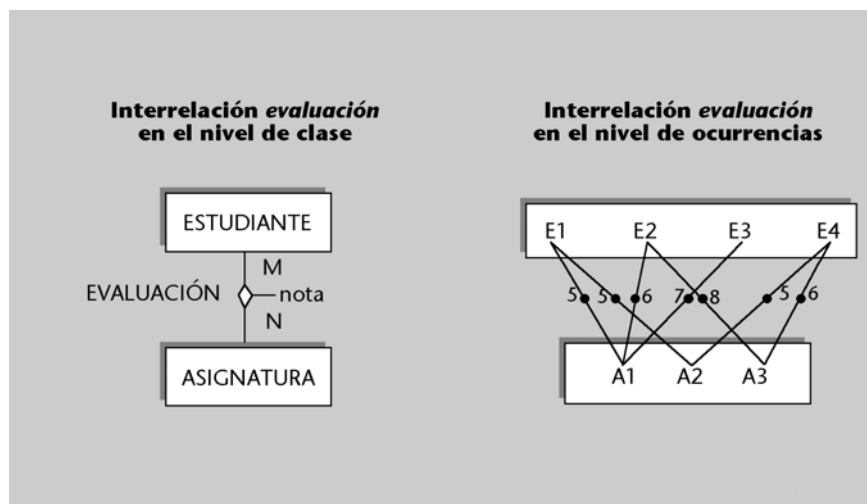
La interrelación anterior tiene conectividad 1:1. Esta interrelación asocia las delegaciones de una empresa con las ciudades donde están situadas. El hecho de que sea 1:1 indica que una ciudad tiene sólo una delegación, y que una delegación está situada en una única ciudad.

b) Conectividad 1:N




La interrelación *asignación* entre la entidad *empleado* y la entidad *despacho* tiene conectividad 1:N, y la N está en el lado de la entidad *empleado*. Esto significa que un empleado tiene un solo despacho asignado, pero que, en cambio, un despacho puede tener uno o más empleados asignados.

c) Conectividad M:N




Para analizar la conectividad M:N, consideramos la interrelación *evaluación* de la figura anterior. Nos indica que un estudiante puede ser evaluado de varias asignaturas y, al mismo tiempo, que una asignatura puede tener varios estudiantes por evaluar.

Es muy habitual que las interrelaciones binarias M:N y todas las n -arias tengan atributos. En cambio, las interrelaciones binarias 1:1 y 1:N no tienen por qué tenerlos. Siempre se pueden asignar estos atributos a la entidad del lado N, en el caso de las 1:N, y a cualquiera de las dos entidades interrelacionadas en el caso de las 1:1. Este cambio de situación del atributo se puede hacer porque no origina un atributo multivaluado. 

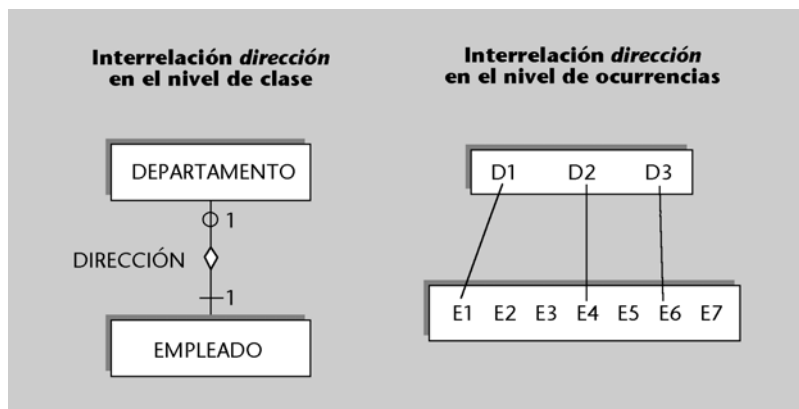
Dependencias de existencia en las interrelaciones binarias

En algunos casos, una entidad individual sólo puede existir si hay como mínimo otra entidad individual asociada con ella mediante una interrelación binaria determinada. En estos casos, se dice que esta última entidad es una **entidad obligatoria en la interrelación**. Cuando esto no sucede, se dice que es una **entidad opcional en la interrelación**.

En el modelo ER, un círculo en la línea de conexión entre una entidad y una interrelación indica que la entidad es opcional en la interrelación. La obligatoriedad de una entidad a una interrelación se indica con una línea perpendicular. Si no se consigna ni un círculo ni una línea perpendicular, se considera que la dependencia de existencia es desconocida. 

Ejemplo de dependencias de existencia

La figura siguiente nos servirá para entender el significado práctico de la dependencia de existencia. La entidad *empleado* es obligatoria en la interrelación *dirección*. Esto indica que no puede existir un departamento que no tenga un empleado que actúa de director del departamento. La entidad *departamento*, en cambio, es opcional en la interrelación *dirección*. Es posible que haya un empleado que no está interrelacionado con ningún departamento: puede haber –y es el caso más frecuente– empleados que no son directores de departamento.



Aplicaremos la dependencia de existencia en las interrelaciones binarias, pero no en las *n*-arias.

2.1.4. Ejemplo: base de datos de casas de colonias

En este punto, y antes de continuar explicando construcciones más complejas del modelo ER, puede resultar muy ilustrativo ver la aplicación práctica de las construcciones que hemos estudiado hasta ahora. Por este motivo, analizaremos un caso práctico de diseño con el modelo ER que corresponde a una base de datos destinada a la gestión de las inscripciones en un conjunto de casas de colonias. El modelo ER de esta base de datos será bastante sencillo e incluirá sólo entidades, atributos e interrelaciones binarias (no incluirá interrelaciones *n*-arias ni otros tipos de estructuras).

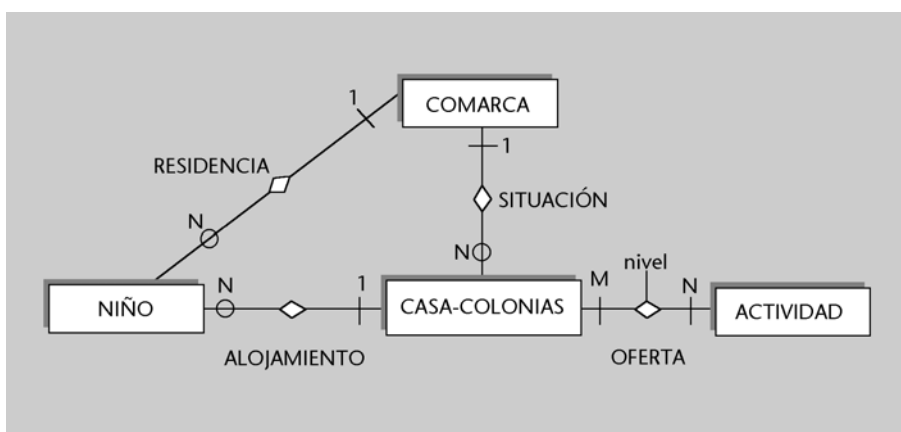
La descripción siguiente explica con detalle los requisitos de los usuarios que hay que tener en cuenta al hacer el diseño conceptual de la futura base de datos:

- a) Cada casa de colonias tiene un nombre que la identifica. Se desea saber de cada una, aparte del nombre, la capacidad (el número de niños que se pueden alojar en cada una como máximo), la comarca donde está situada y las ofertas de actividades que proporciona. Una casa puede ofrecer actividades como por ejemplo natación, esquí, remo, pintura, fotografía, música, etc.
- b) Es necesario tener en cuenta que en una casa de colonias se pueden practicar varias actividades (de hecho, cada casa debe ofrecer como mínimo una), y también puede ocurrir que una misma actividad se pueda llevar a cabo en varias casas. Sin embargo, toda actividad que se registre en la base de datos debe ser ofertada como mínimo en una de las casas.
- c) Interesa tener una evaluación de las ofertas de actividades que proporcionan las casas. Se asigna una calificación numérica que indica el nivel de calidad que tiene cada una de las actividades ofertadas.
- d) Las casas de colonias alojan niños que se han inscrito para pasar en ellas unas pequeñas vacaciones. Se quiere tener constancia de los niños que se alojan en cada una de las casas en el momento actual. Se debe suponer que hay casas que están vacías (en las que no se aloja ningún niño) durante algunas temporadas.
- e) De los niños que se alojan actualmente en alguna de las casas, interesa conocer un código que se les asigna para identificarlos, su nombre, su apellido, el número de teléfono de sus padres y su comarca de residencia.
- f) De las comarcas donde hay casas o bien donde residen niños, se quiere tener registrados la superficie y el número de habitantes. Se debe considerar que puede haber comarcas donde no reside ninguno de los niños que se alojan en un momento determinado en las casas de colonias, y comarcas que no disponen de ninguna casa.

Es posible, ...

... por ejemplo, que una actividad como por ejemplo el esquí tenga una calificación de 10 en la oferta de la casa Grévol, y que la misma actividad tenga una calificación de 8 en la casa Ardilla.

La figura siguiente muestra un diagrama ER que satisface los requisitos anteriores. Los atributos de las entidades no figuran en el diagrama y se listan aparte.



Los atributos de las entidades que figuran en el diagrama son los siguientes (las claves primarias están subrayadas):

CASA-COLONIAS

nombre-casa, capacidad

ACTIVIDAD

nombre-actividad

NIÑO

código-niño, nombre, apellido, teléfono

COMARCA

nombre-comarca, superficie, número-habitantes

A continuación comentamos los aspectos más relevantes de este modelo ER: 

1) Una de las dificultades que en ocasiones se presenta durante la modelización conceptual es decidir si una información determinada debe ser una entidad o un atributo. En nuestro ejemplo, puede resultar difícil decidir si *comarca* se debe modelizar como una entidad o como un atributo.

A primera vista, podría parecer que *comarca* debe ser un atributo de la entidad *casa-colonias* para indicar dónde está situada una casa de colonias, y también un atributo de la entidad *niño* para indicar la residencia del niño. Sin embargo, esta solución no sería adecuada, porque se quieren tener informaciones adicionales asociadas a la comarca: la superficie y el número de habitantes. Es preciso que *comarca* sea una entidad para poder reflejar estas informaciones adicionales como atributos de la entidad.

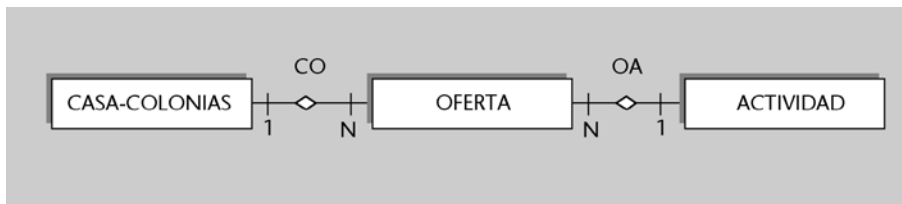
La entidad *comarca* tendrá que estar, evidentemente, interrelacionada con las entidades *niño* y *casa-colonias*. Observad que de este modo, además, se hace patente que las comarcas de residencia de los niños y las comarcas de situación de las casas son informaciones de un mismo tipo.

2) Otra decisión que hay que tomar es si el concepto *actividad* se debe modelizar como una entidad o como un atributo. *Actividad* no tiene informaciones adicionales asociadas; no tiene, por lo tanto, más atributos que los que forman la clave. Aun así, es necesario que *actividad* sea una entidad para que, mediante la interrelación *oferta*, se pueda indicar que una casa de colonias ofrece actividades.

Observad que las actividades ofertadas no se pueden expresar como un atributo de *casa-colonias*, porque una casa puede ofrecer muchas actividades y, en este caso, el atributo no podría tomar un valor único.

3) Otra elección difícil, que con frecuencia se presenta al diseñar un modelo ER, consiste en modelizar una información determinada como una entidad o

como una interrelación. Por ejemplo, podríamos haber establecido que *oferta*, en lugar de ser una interrelación, fuese una entidad; lo habríamos hecho así:



La entidad *oferta* representada en la figura anterior tiene los atributos que presentamos a continuación:

OFERTA

nombre-casa, nombre-actividad, nivel

Esta solución no acaba de reflejar adecuadamente la realidad. Si analizamos la clave de *oferta*, podemos ver que se identifica con *nombre-casa*, que es la clave de la entidad *casa-colonias*, y con *nombre-actividad*, que es la clave de la entidad *actividad*. Esto nos debe hacer sospechar que *oferta*, de hecho, corresponde a una asociación o interrelación entre *casas* y *actividades*. En consecuencia, reflejaremos la realidad con más exactitud si modelizamos *oferta* como una interrelación entre estas entidades.

4) Finalmente, un aspecto que hay que cuidar durante el diseño conceptual es el de evitar las redundancias. Por ejemplo, si hubiésemos interrelacionado *comarca* con *actividad* para saber qué actividades se realizan en las casas de cada una de las comarcas, habríamos tenido información redundante. La interrelación *oferta* junto con la interrelación *situación* ya permiten saber, de forma indirecta, qué actividades se hacen en las comarcas.

2.1.5. Interrelaciones *n*-arias

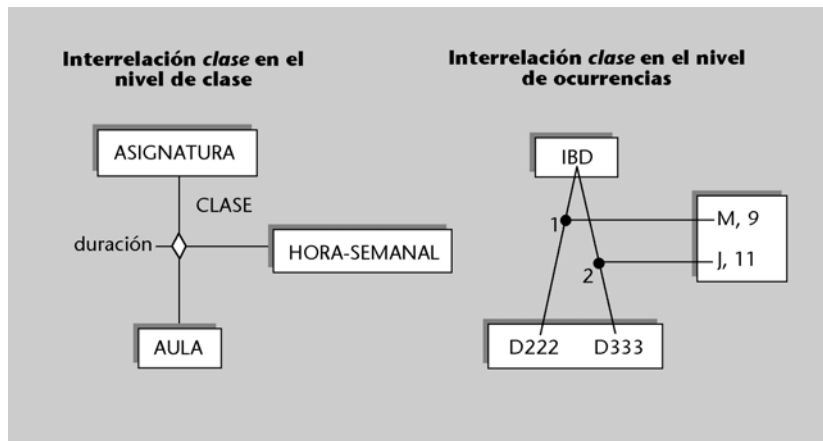
Las interrelaciones *n*-arias, igual que las binarias, pueden tener diferentes tipos de conectividad. En este subapartado analizaremos primero el caso particular de las interrelaciones ternarias y, a continuación, trataremos las conectividades de las interrelaciones *n*-arias en general. !

Conectividad de las interrelaciones ternarias

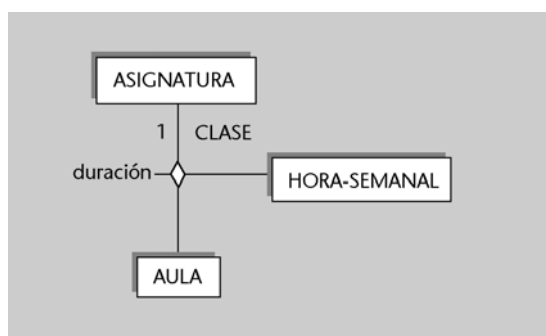
Cada una de las tres entidades asociadas con una interrelación ternaria puede estar conectada con conectividad “uno” o bien con conectividad “muchos”. En consecuencia, las interrelaciones ternarias pueden tener cuatro tipos de conectividad: M:N:P, M:M:1, N:1:1 y 1:1:1.

Observad que usamos M, N y P para representar “muchos”, y 1 para representar “uno”.

Analizaremos cómo se decide cuál es la conectividad adecuada de una interrelación ternaria mediante el siguiente ejemplo. Consideremos una interrelación que denominamos *clase* y que asocia las entidades *asignatura*, *aula* y *hora-semanal*. Esta interrelación permite registrar clases presenciales. Una clase corresponde a una asignatura determinada, se imparte en un aula determinada y a una hora de la semana determinada. Por ejemplo, podemos registrar que se hace clase de la asignatura IBD en el aula D222 el martes a las 9, tal y como se muestra en la figura de la página siguiente. El atributo *duración* nos permite saber cuántas horas dura la clase.



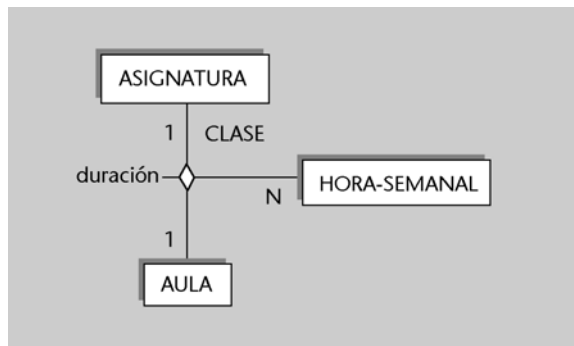
Para decidir si el lado de la entidad *asignatura* se conecta con “uno” o con “muchos”, es necesario preguntarse si, dadas un *aula* y una *hora-semanal*, se puede hacer clase de sólo una o bien de muchas asignaturas en aquellas aula y hora. La respuesta sería que sólo se puede hacer clase de una asignatura en una misma aula y hora. Esto nos indica que *asignatura* se conecta con “uno”, tal y como reflejamos en la figura siguiente:



Como nos indica este ejemplo, para decidir cómo se debe conectar una de las entidades, es necesario preguntarse si, ya fijadas ocurrencias concretas de las otras dos, es posible conectar sólo “una” o bien “muchas” ocurrencias de la primera entidad. 🗨️

Utilizaremos el mismo procedimiento para determinar cómo se conectan las otras dos entidades del ejemplo. Una vez fijadas una asignatura y un aula, es posible que se haga clase de aquella asignatura en aquella aula, en varias horas de la semana; entonces, *hora-semana* se conecta con “muchos”. Finalmente, la

entidad *aula* se conecta con “uno”, teniendo en cuenta que, fijadas una asignatura y una hora de la semana, sólo se puede hacer una clase de aquella asignatura a aquella hora. La conectividad resultante, de este modo, es N:1:1.



Caso general: conectividad de las interrelaciones n -arias

Lo que hemos explicado sobre la conectividad para las interrelaciones ternarias es fácilmente generalizable a interrelaciones n -arias.

Una interrelación n -aria puede tener $n + 1$ tipos de conectividad, teniendo en cuenta que cada una de las n entidades puede estar conectada con “uno” o con “muchos” en la interrelación*.

* Recordad que para las interrelaciones ternarias hay cuatro tipos posibles de conectividad.

Para decidir si una de las entidades se conecta con “uno” o con “muchos”, es necesario preguntarse si, fijadas ocurrencias concretas de las otras $n - 1$ entidades, es posible conectar sólo una o bien muchas ocurrencias de la primera entidad: !

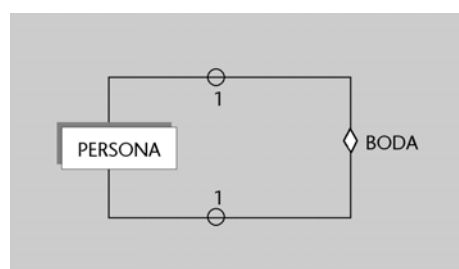
- Si la respuesta es que sólo una, entonces se conecta con “uno”.
- Si la respuesta es que muchas, la entidad se conecta con “muchos”.

2.1.6. Interrelaciones recursivas

Una **interrelación recursiva** es una interrelación en la que alguna entidad está asociada más de una vez.

Ejemplo de interrelación recursiva

Si, para una entidad *persona*, queremos tener constancia de qué personas están actualmente casadas entre ellas, será necesario definir la siguiente interrelación, que asocia dos veces la entidad *persona*:



Una interrelación recursiva puede ser tanto binaria como n -aria:

1) **Interrelación recursiva binaria:** interrelación en la que las ocurrencias asocian dos instancias de la misma entidad*. Las interrelaciones binarias recursivas pueden tener conectividad 1:1, 1:N o M:N, como todas las binarias. En esta interrelación también es posible expresar la dependencia de existencia igual que en el resto de las interrelaciones binarias.

* Éste es el caso de la interrelación *boda* anterior.

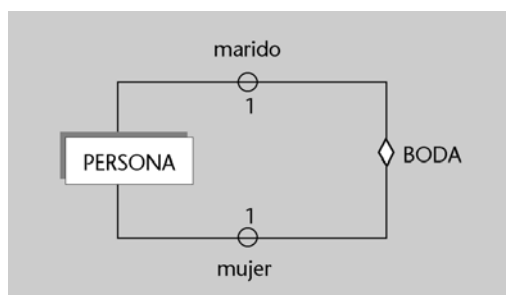
Ejemplo de interrelación recursiva binaria

La interrelación *boda* tiene conectividad 1:1 porque un marido está casado con una sola mujer y una mujer está casada con un solo marido. También tiene un círculo en los dos lados (según la dependencia de existencia), porque puede haber personas que no estén casadas.

En una interrelación recursiva, puede interesar distinguir los diferentes papeles que una misma entidad tiene en la interrelación. Con este objetivo, se puede etiquetar cada línea de la interrelación con un rol. En las interrelaciones no recursivas normalmente no se especifica el rol; puesto que todas las entidades interrelacionadas son de clases diferentes, sus diferencias de rol se sobreentienden.

Roles diferentes

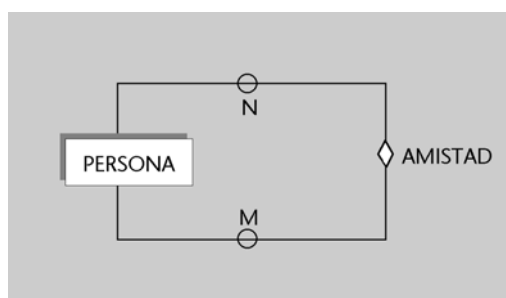
Una ocurrencia de la interrelación *boda* asocia a dos personas concretas. Para reflejar el papel diferente que tiene cada una de ellas en la interrelación, una de las personas tendrá el rol de marido y la otra tendrá el rol de mujer.



Algunas interrelaciones recursivas no presentan diferenciación de roles; entonces, las líneas de la interrelación no se etiquetan.

No-diferencia de roles

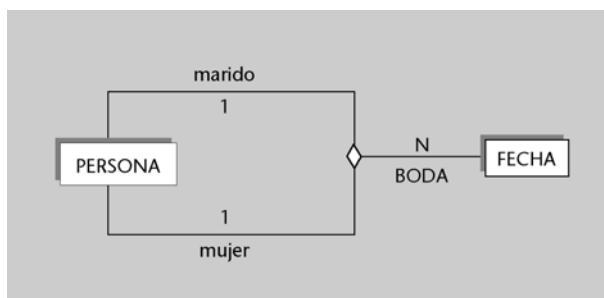
Consideremos una interrelación *amistad* que asocia a personas concretas que son amigas. A diferencia de lo que sucedía en la interrelación *boda*, donde una de las personas es el marido y la otra la mujer, en este caso no hay diferenciación de roles entre las dos personas interrelacionadas. A continuación se muestra esta interrelación. Observad que su conectividad es M:N, teniendo en cuenta que una persona puede tener muchos amigos y, al mismo tiempo, puede haber muchas personas que la consideran amiga.



2) **Interrelación recursiva n -aria:** interrelación recursiva en la que las ocurrencias asocian más de dos instancias.

Ejemplo de interrelación recursiva ternaria

Consideremos una interrelación que registra todas las bodas que se han producido a lo largo del tiempo entre un conjunto de personas determinado. Esta interrelación permite tener constancia no sólo de las bodas vigentes, sino de todas las bodas realizadas en un cierto periodo de tiempo.



Esta interrelación es recursiva y ternaria. Una ocurrencia de la interrelación asocia a una persona que es el marido, a otra que es la mujer y la fecha de su boda. La conectividad es N:1:1. A los lados del marido y de la mujer les corresponde un 1, porque un marido o una mujer, en una fecha determinada, se casa con una sola persona. Al lado de la entidad fecha le corresponde una N, porque se podría dar el caso de que hubiese, en fechas diferentes, más de una boda entre las mismas personas.

2.1.7. Entidades débiles

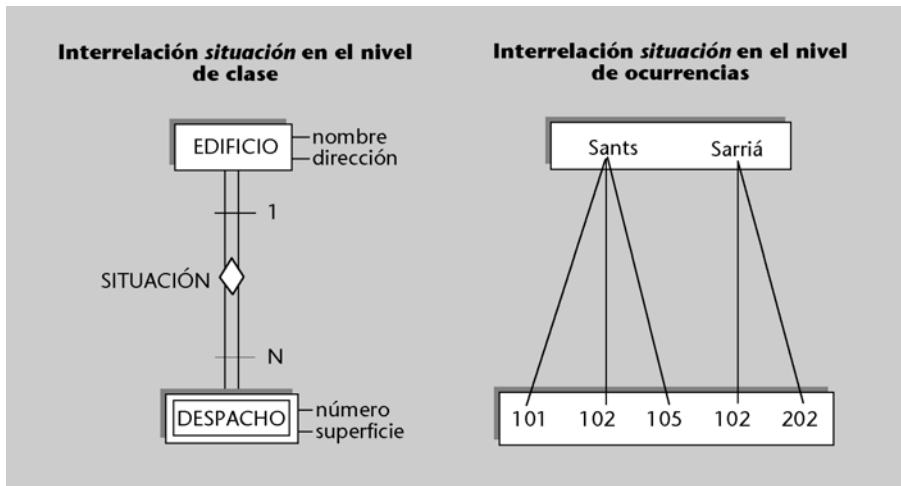
Las entidades que hemos considerado hasta ahora tienen un conjunto de atributos que forman su claves primarias y que permiten identificarlas completamente. Estas entidades se denominan, de forma más específica, **entidades fuertes**. En este subapartado consideraremos otro tipo de entidades que denominaremos *entidades débiles*.


Una **entidad débil** es una entidad cuyos atributos no la identifican completamente, sino que sólo la identifican de forma parcial. Esta entidad debe participar en una interrelación que ayuda a identificarla.

Una entidad débil se representa con un rectángulo doble, y la interrelación que ayuda a identificarla se representa con una doble línea.

Ejemplo de entidad débil

Consideremos las entidades *edificio* y *despacho* de la figura siguiente. Supongamos que puede haber despachos con el mismo número en edificios diferentes. Entonces, su número no identifica completamente un despacho. Para identificar completamente un despacho, es necesario tener en cuenta en qué edificio está situado. De hecho, podemos identificar un despacho mediante la interrelación *situación*, que lo asocia a un único edificio. El nombre del edificio donde está situado junto con el número de despacho lo identifican completamente.



En el ejemplo anterior, la interrelación *situación* nos ha permitido completar la identificación de los despachos. Para toda entidad débil, siempre debe haber una única interrelación que permita completar su identificación. Esta interrelación debe ser binaria con conectividad 1:N, y la entidad débil debe estar en el lado N. De este modo, una ocurrencia de la entidad débil está asociada con una sola ocurrencia de la entidad del lado 1, y será posible completar su identificación de forma única. Además, la entidad del lado 1 debe ser obligatoria en la interrelación porque, si no fuese así, alguna ocurrencia de la entidad débil podría no estar interrelacionada con ninguna de sus ocurrencias y no se podría identificar completamente. 

2.2. Extensiones del modelo ER

En este subapartado estudiaremos algunas construcciones avanzadas que extienden el modelo ER estudiado hasta ahora.

2.2.1. Generalización/especialización


En algunos casos, hay ocurrencias de una entidad que tienen características propias específicas que nos interesa modelizar. Por ejemplo, puede ocurrir que se quiera tener constancia de qué coche de la empresa tienen asignado los empleados que son directivos; también que, de los empleados técnicos, interese tener una interrelación con una entidad *proyecto* que indique en qué proyectos trabajan y se desee registrar su titulación. Finalmente, que convenga conocer la antigüedad de los empleados administrativos. Asimismo, habrá algunas características comunes a todos los empleados: todos se identifican por un DNI, tienen un nombre, un apellido, una dirección y un número de teléfono.

La **generalización/especialización** permite reflejar el hecho de que hay una entidad general, que denominamos *entidad superclase*, que se puede especializar en entidades subclase:

a) La **entidad superclase** nos permite modelizar las características comunes de la entidad vista de una forma genérica.

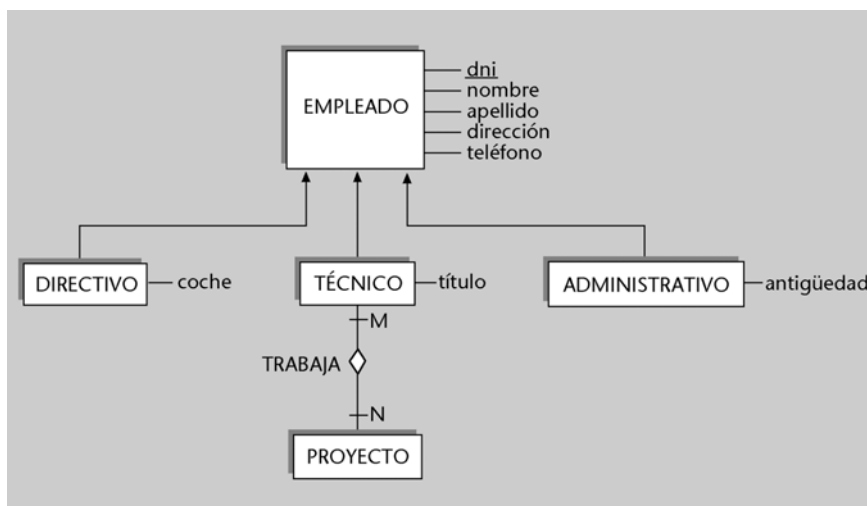
b) Las **entidades subclase** nos permiten modelizar las características propias de sus especializaciones.

Es necesario que se cumpla que toda ocurrencia de una entidad subclase sea también una ocurrencia de su entidad superclase.

Denotamos la generalización/especialización con una flecha que parte de las entidades subclase y que se dirige a la entidad superclase. 

Ejemplo de entidades superclase y subclase

En la figura siguiente están representadas la entidad superclase, que corresponde al empleado del ejemplo anterior, y las entidades subclase, que corresponden al directivo, al técnico y al administrativo del mismo ejemplo.



En la generalización/especialización, las características (atributos o interrelaciones) de la entidad superclase se propagan hacia las entidades subclase. Es lo que se denomina **herencia de propiedades**.

En el diseño de una generalización/especialización, se puede seguir uno de los dos procesos siguientes:

1) Puede ocurrir que el diseñador primero identifique la necesidad de la entidad superclase y, posteriormente, reconozca las características específicas que hacen necesarias las entidades subclase. En estos casos se dice que ha seguido un **proceso de especialización**.

2) La alternativa es que el diseñador modelice en primer lugar las entidades subclase y, después, se dé cuenta de sus características comunes e identifique la entidad superclase. Entonces se dice que ha seguido un **proceso de generalización**.

La generalización/especialización puede ser de dos tipos: !

a) **Disjunta**. En este caso no puede suceder que una misma ocurrencia aparezca en dos entidades subclase diferentes. Se denota gráficamente con la etiqueta D.

b) **Solapada**. En este caso no tiene lugar la restricción anterior. Se denota gráficamente con la etiqueta S.

Nuestro ejemplo de los empleados...

... corresponde a una generalización/especialización disjunta porque ningún empleado puede ser de más de un tipo. Se denota con la etiqueta D.

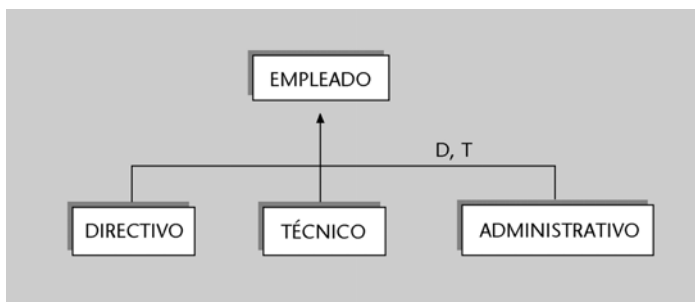
Además, una generalización/especialización también puede ser: !

1) **Total**. En este caso, toda ocurrencia de la entidad superclase debe pertenecer a alguna de las entidades subclase. Esto se denota con la etiqueta T.

2) **Parcial**. En este caso no es necesario que se cumpla la condición anterior. Se denota con la etiqueta P.

La generalización/especialización de los empleados

La generalización/especialización de los empleados es total porque suponemos que todo empleado debe ser directivo, técnico o administrativo. Se denota con la etiqueta T.



2.2.2. Entidades asociativas

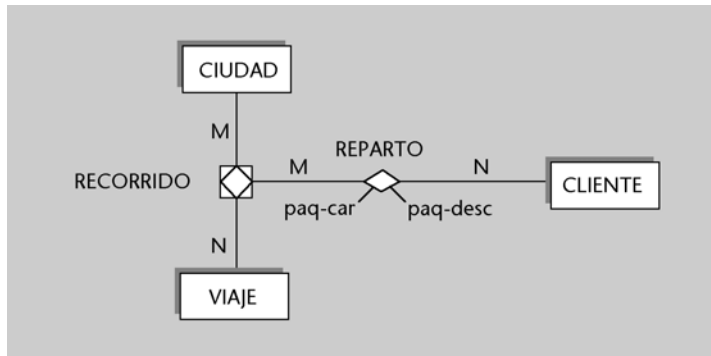
En este subapartado veremos un mecanismo que nos permite considerar una interrelación entre entidades como si fuese una entidad.

La entidad que resulta de considerar una interrelación entre entidades como si fuese una entidad es una **entidad asociativa**, y tendrá el mismo nombre que la interrelación sobre la que se define.

La utilidad de una entidad asociativa consiste en que se puede interrelacionar con otras entidades y, de forma indirecta, nos permite tener interrelaciones en las que intervienen interrelaciones. Una entidad asociativa se denota recuadrando el rombo de la interrelación de la que proviene. !

Ejemplo de entidad asociativa

La figura siguiente muestra un ejemplo de entidad asociativa:

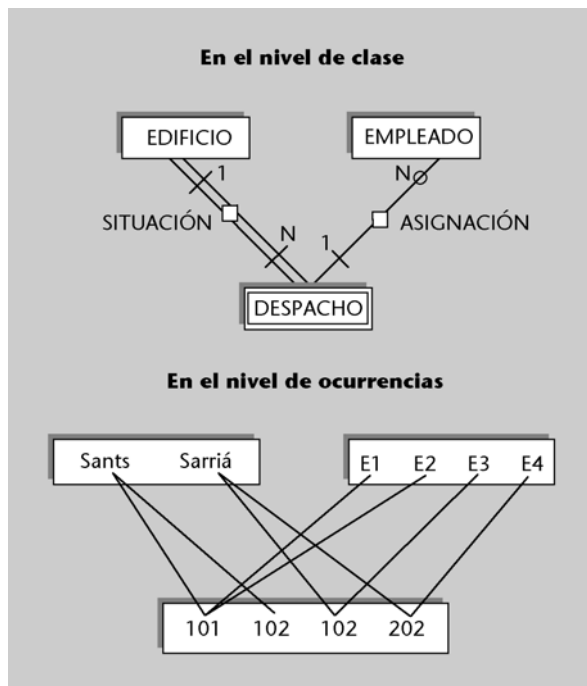


Recorrido es una interrelación de conectividad M:N que registra las ciudades por donde han pasado los diferentes viajes organizados por una empresa de reparto de paquetes. Consideramos *recorrido* una entidad asociativa con el fin de interrelacionarla con la entidad *cliente*; de este modo nos será posible reflejar por orden de qué clientes se han hecho repartos en una ciudad del recorrido de un viaje, así como el número de paquetes cargados y descargados siguiendo sus indicaciones.

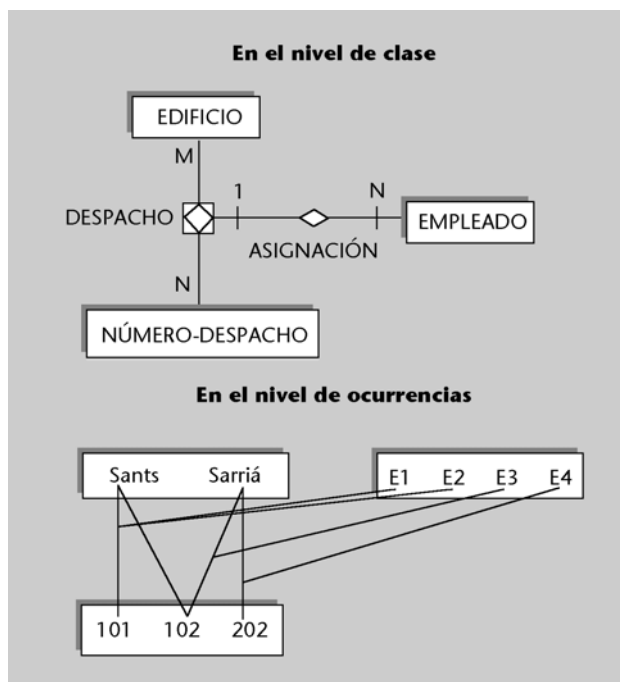
El mecanismo de las entidades asociativas subsume el de las entidades débiles y resulta todavía más potente. Es decir, siempre que utilicemos una entidad débil podremos sustituirla por una entidad asociativa, pero no al revés.

Ejemplo de sustitución de una entidad débil por una asociativa

A continuación se muestra la entidad débil *despacho*, que tiene la interrelación *asignación* con la entidad *empleado*.



Podríamos modelizar este caso haciendo que *despacho* fuese una entidad asociativa si consideramos una nueva entidad *número-despacho* que contiene simplemente números de despachos. Entonces, la entidad asociativa *despacho* se obtiene de la interrelación entre *edificio* y *número-despacho*.



Aunque las entidades débiles se puedan sustituir por el mecanismo de las entidades asociativas, es adecuado mantenerlas en el modelo ER porque resultan menos complejas y son suficientes para modelizar muchas de las situaciones que se producen en el mundo real.

2.3. Ejemplo: base de datos del personal de una entidad bancaria

En este subapartado veremos un ejemplo de diseño conceptual de una base de datos mediante el modelo ER.

Se trata de diseñar una base de datos para la gestión del personal de una entidad bancaria determinada que dispone de muchos empleados y de una amplia red de agencias. La siguiente descripción resume los requisitos de los usuarios de la futura base de datos:

- a) Los empleados se identifican por un código de empleado, y también deseamos conocer su DNI, su NSS, su nombre y su apellido. Será importante registrar su ciudad de residencia, considerando que hay ciudades donde no reside ningún empleado.
- b) Interesa saber en qué ciudades están ubicadas las diversas agencias de la entidad bancaria. Estas agencias bancarias se identifican por la ciudad donde están y por un nombre que permite distinguir las agencias de una misma ciudad. Se quiere tener constancia del número de habitantes de las ciudades, así como de la dirección y el número de teléfono de las agencias. Se debe consi-

derar que la base de datos también incluye ciudades donde no hay ninguna agencia.

c) Un empleado, en un momento determinado, trabaja en una sola agencia, lo cual no impide que pueda ser trasladado a otra o, incluso, que vuelva a trabajar en una agencia donde ya había trabajado anteriormente. Se quiere tener constancia del historial del paso de los empleados por las agencias.

d) Los empleados pueden tener títulos académicos (aunque no todos los tienen). Se quiere saber qué títulos tienen los empleados.

e) Cada empleado tiene una categoría laboral determinada (auxiliar, oficial de segunda, oficial de primera, etc.). A cada categoría le corresponde un sueldo base determinado y un precio por hora extra también determinado. Se quiere tener constancia de la categoría actual de cada empleado, y del sueldo base y el precio de la hora extra de cada categoría.

f) Algunos empleados (no todos) están afiliados a alguna central sindical. Se ha llegado al pacto de descontar de la nómina mensual la cuota sindical a los afiliados a cada central. Esta cuota es única para todos los afiliados a una central determinada. Es necesario almacenar las afiliaciones a una central de los empleados y las cuotas correspondientes a las diferentes centrales sindicales.

g) Hay dos tipos de empleados diferentes:

- Los que tienen contrato fijo, cuya antigüedad queremos conocer.
- Los que tienen contrato temporal, de los cuales nos interesa saber las fechas de inicio y finalización de su último contrato.

Si un empleado temporal pasa a ser fijo, se le asigna un nuevo código de empleado; consideraremos que un empleado fijo nunca pasa a ser temporal. Todo lo que se ha indicado hasta ahora (traslados, categorías, afiliación sindical, etc.) es aplicable tanto a empleados fijos como a temporales.

h) Los empleados fijos tienen la posibilidad de pedir diferentes tipos preestablecidos de préstamos (por matrimonio, por adquisición de vivienda, por estudios, etc.), que pueden ser concedidos o no. En principio, no hay ninguna limitación a la hora de pedir varios préstamos a la vez, siempre que no se pida más de uno del mismo tipo al mismo tiempo. Se quiere registrar los préstamos pedidos por los empleados, y hacer constar si han sido concedidos o no. Cada tipo de préstamo tiene establecidas diferentes condiciones; de estas condiciones, en particular, nos interesará saber el tipo de interés y el periodo de vigencia del préstamo.

AGENCIA (entidad débil: nombre-agencia la identifica parcialmente, se identifica completamente con la ciudad de situación)

nombre-agencia, dirección, teléfono

TÍTULO

nombre-título

CATEGORÍA

nombre-categ, sueldo-base, hora-extra

CENTRAL-SINDICAL

central, cuota

TIPO-PRÉSTAMO

código-préstamo, tipo-interés, período-vigencia

FECHA

fecha

A continuación, comentaremos los aspectos que pueden resultar más complejos de este modelo ER:

- 1) La entidad *agencia* se ha considerado una entidad débil porque su atributo *nombre-agencia* sólo permite distinguir las agencias situadas en una misma ciudad, pero para identificar de forma total una agencia, es necesario saber en qué ciudad está situada. De este modo, la interrelación *situación* es la que nos permite completar la identificación de la entidad *agencia*.
- 2) La interrelación *petición* es ternaria y asocia a empleados fijos que hacen peticiones de préstamos, tipos de préstamos pedidos por los empleados y fechas en las que se hacen estas peticiones.
- 3) El lado de la entidad *fecha* se conecta con “muchos” porque un mismo empleado puede pedir un mismo tipo de préstamo varias veces en fechas distintas. La entidad *fijo* se conecta con “muchos” porque un tipo de préstamo determinado puede ser pedido en una misma fecha por varios empleados. También la entidad *tipo-préstamo* se conecta con “muchos” porque es posible que un empleado en una fecha determinada pida más de un préstamo de tipo diferente.
- 4) El atributo *concedido/no* indica si el préstamo se ha concedido o no. Es un atributo de la interrelación porque su valor depende al mismo tiempo del empleado fijo que hace la petición, del tipo de préstamo pedido y de la fecha de petición.
- 5) La interrelación *traslado* también es una interrelación ternaria que permite registrar el paso de los empleados por las distintas agencias. Un traslado concreto asocia a un empleado, una agencia donde él trabajará y una fecha inicial en la que empieza a trabajar en la agencia. El atributo de la interrelación *fecha-fin* indica en qué fecha finaliza su asignación a la agencia (*fecha-fin* tendrá el valor nulo cuando


un empleado trabaja en una agencia en el momento actual y no se sabe cuándo se le trasladará). Observad que *fecha-fin* debe ser un atributo de la interrelación. Si se colocase en una de las tres entidades interrelacionadas, no podría ser un atributo univaluado.

Conviene observar que esta interrelación no registra todas y cada una de las fechas en las que un empleado está asignado a una agencia, sino sólo la fecha inicial y la fecha final de la asignación. Es muy habitual que, para informaciones que son ciertas durante todo un periodo de tiempo, se registre en la base de datos únicamente el inicio y el final del periodo.

Notad que la entidad *agencia* se ha conectado con “uno” en la interrelación *traslado*, porque no puede ocurrir que, en una fecha, un empleado determinado sea trasladado a más de una agencia.

6) Finalmente, comentaremos la generalización/especialización de la entidad *empleado*. Los empleados pueden ser de dos tipos; se quieren registrar propiedades diferentes para cada uno de los tipos y también se requieren algunas propiedades comunes a todos los empleados. Por este motivo, es adecuado utilizar una generalización/especialización.

3. Diseño lógico: la transformación del modelo ER al modelo relacional

En este apartado trataremos el diseño lógico de una base de datos relacional. Partiremos del resultado de la etapa del diseño conceptual expresado mediante el modelo ER y veremos cómo se puede transformar en una estructura de datos del modelo relacional. 


3.1. Introducción a la transformación de entidades e interrelaciones


Los elementos básicos del modelo ER son las entidades y las interrelaciones:

- a) Las entidades, cuando se traducen al modelo relacional, originan **relaciones**.
- b) Las interrelaciones, en cambio, cuando se transforman, pueden dar lugar a **claves foráneas** de alguna relación ya obtenida o pueden dar lugar a una **nueva relación**.

En el caso de las interrelaciones, es necesario tener en cuenta su grado y su conectividad para poder decidir cuál es la transformación adecuada:

- 1) Las interrelaciones binarias 1:1 y 1:N dan lugar a claves foráneas.
- 2) Las interrelaciones binarias M:N y todas las n -arias se traducen en nuevas relaciones.

En los subapartados siguientes explicaremos de forma más concreta las transformaciones necesarias para obtener un esquema relacional a partir de un modelo ER. Más adelante proporcionamos una tabla que resume los aspectos más importantes de cada una de las transformaciones para dar una visión global sobre ello. Finalmente, describimos su aplicación en un ejemplo. 

 Encontraréis la tabla de las transformaciones en el subapartado 3.10. de esta unidad; en el subapartado 3.11. veremos el ejemplo de aplicación.

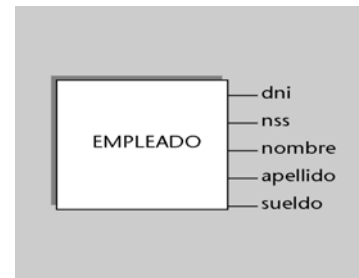
3.2. Transformación de entidades

Empezaremos el proceso transformando todas las entidades de un modelo ER adecuadamente. Cada entidad del modelo ER se transforma en una relación del modelo relacional. Los atributos de la entidad serán atributos de la relación y, de forma análoga, la clave primaria de la entidad será la clave primaria de la relación.

Ejemplo de transformación de una entidad

Según esto, la entidad de la figura del margen se transforma en la relación que tenemos a continuación:

EMPLEADO(DNI, NSS, nombre, apellido, sueldo)



Una vez transformadas todas las entidades en relaciones, es preciso transformar todas las interrelaciones en las que intervienen estas entidades.

Si una entidad interviene en alguna interrelación binaria 1:1 o 1:N, puede ser necesario añadir nuevos atributos a la relación obtenida a partir de la entidad. Estos atributos formarán claves foráneas de la relación.

Veremos las transformaciones de las interrelaciones binarias en el siguiente subapartado.

3.3. Transformación de interrelaciones binarias

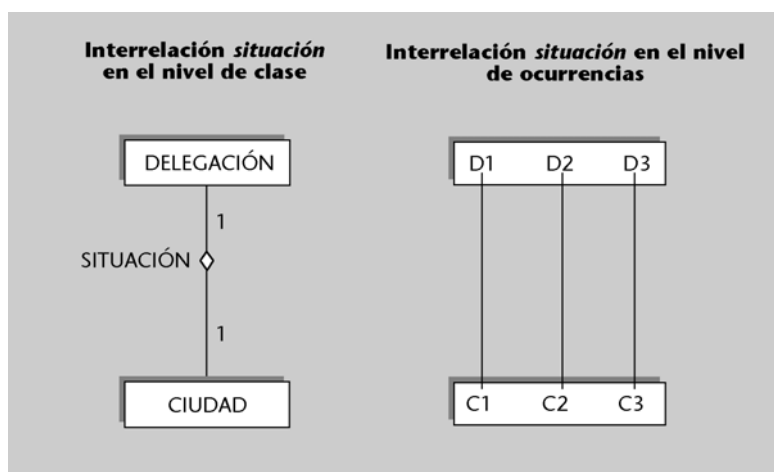
Para transformar una interrelación binaria es necesario tener en cuenta su conectividad, y si las entidades son obligatorias u opcionales en la interrelación.

3.3.1. Conectividad 1:1

Nuestro punto de partida es que las entidades que intervienen en la interrelación 1:1 ya se han transformado en relaciones con sus correspondientes atributos.

Entonces sólo será necesario añadir a cualquiera de estas dos relaciones una clave foránea que referencie a la otra relación.

Ejemplo de transformación de una interrelación binaria 1:1



Para la interrelación de la figura anterior, tenemos dos opciones de transformación:

- Primera opción:

```
DELEGACIÓN(nombre-del, ..., nombre-ciudad)
donde {nombre-ciudad} referencia CIUDAD
CIUDAD(nombre-ciudad, ...)
```

- Segunda opción:

```
DELEGACIÓN(nombre-del, ...)
CIUDAD(nombre-ciudad, ..., nombre-del)
donde {nombre-del} referencia DELEGACIÓN
```

Ambas transformaciones nos permiten saber en qué ciudad hay una delegación, y qué delegación tiene una ciudad. De este modo, reflejan correctamente el significado de la interrelación *situación* del modelo ER.

En la primera transformación, dado que una delegación está situada en una sola ciudad, el atributo *nombre-ciudad* tiene un único valor para cada valor de la clave primaria {*nombre-del*}. Observad que, si pudiese tener varios valores, la solución no sería correcta según la teoría relacional.

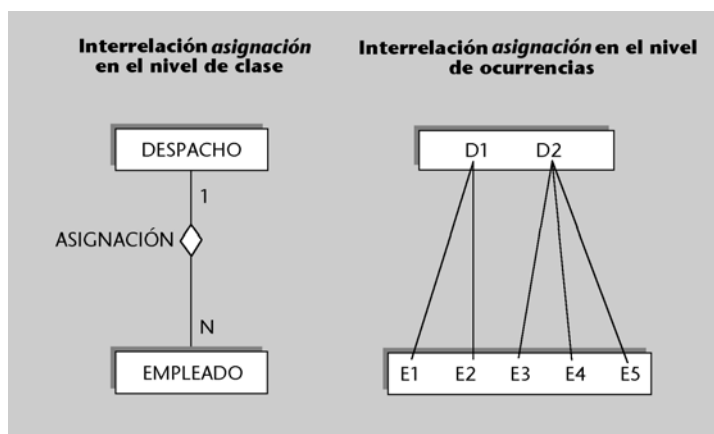
En la segunda transformación, teniendo en cuenta que una ciudad tiene una sola delegación, el atributo *nombre-del* también toma un solo valor para cada valor de la clave primaria {*nombre-ciudad*}.

También es necesario tener en cuenta que, en las dos transformaciones, la clave foránea que se les añade se convierte en una clave alternativa de la relación porque no admite valores repetidos. Por ejemplo, en la segunda transformación no puede haber más de una ciudad con la misma delegación; de este modo, *nombre-del* debe ser diferente para todas las tuplas de *CIUDAD*.

3.3.2. Conectividad 1:N

Partimos del hecho de que las entidades que intervienen en la interrelación 1:N ya se han transformado en relaciones con sus correspondientes atributos. En este caso sólo es necesario añadir en la relación correspondiente a la entidad del lado N, una clave foránea que referencie la otra relación.

Ejemplo de transformación de una interrelación binaria 1:N



La interrelación de la figura anterior se transforma en:

```
DESPACHO(desp, ...)
EMPLEADO(emp, ..., desp)
donde {desp} referencia DESPACHO
```

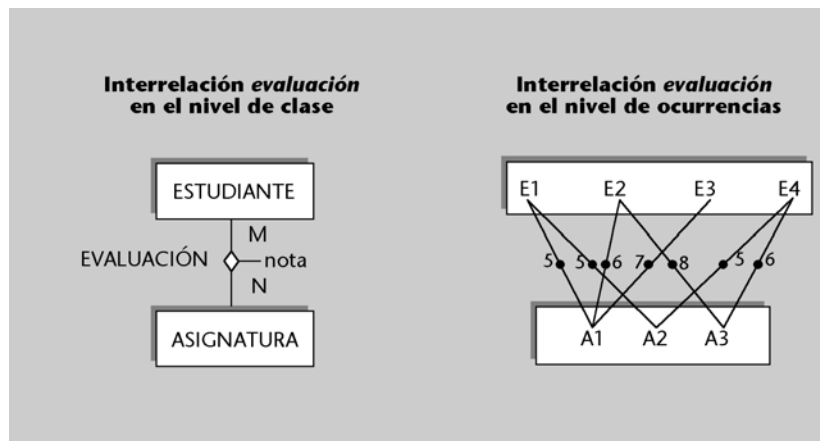
Esta solución nos permite saber en qué despacho está asignado cada empleado, y también nos permite consultar, para cada despacho, qué empleados hay. Es decir, refleja correctamente el significado de la interrelación *asignación*.

Teniendo en cuenta que un empleado está asignado a un único despacho, el atributo *desp* tiene un valor único para cada valor de la clave primaria {*emp*}. Si hubiésemos puesto la clave foránea {*emp*} en la relación *DESPACHO*, la solución habría sido incorrecta, porque *emp* habría tomado varios valores, uno para cada uno de los distintos empleados que pueden estar asignados a un despacho.

3.3.3. Conectividad M:N

Una interrelación M:N se transforma en una relación. Su clave primaria estará formada por los atributos de la clave primaria de las dos entidades interrelacionadas. Los atributos de la interrelación serán atributos de la nueva relación.

Ejemplo de transformación de una interrelación binaria M:N



La interrelación de la figura anterior se transforma en:

```
ESTUDIANTE(est, ...)
ASIGNATURA(asig, ...)
EVALUACIÓN(est, asig, nota)
donde {est} referencia ESTUDIANTE
y {asig} referencia ASIGNATURA
```

Observad que la clave de *evaluación* debe constar tanto de la clave de *estudiante* como de la clave de *asignatura* para identificar completamente la relación.

La solución que hemos presentado refleja correctamente la interrelación *evaluación* y su atributo *nota*. Permite saber, para cada estudiante, qué notas obtiene de las varias asignaturas y, para cada asignatura, qué notas tienen los diferentes estudiantes de aquella asignatura.

En el caso M:N no podemos utilizar claves foráneas para transformar la interrelación, porque obtendríamos atributos que necesitarían tomar varios valores, y esto no se permite en el modelo relacional. !

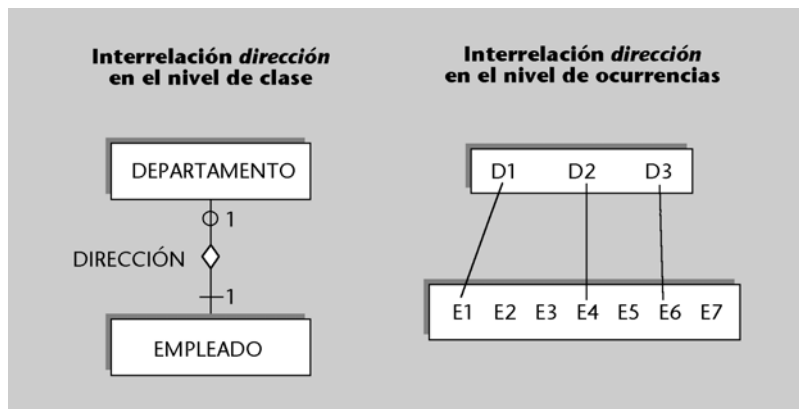
3.3.4. Influencia de la dependencia de existencia en la transformación de las interrelaciones binarias

La dependencia de existencia, o más concretamente, el hecho de que alguna de las entidades sea opcional en una interrelación se debe tener en cuenta al hacer la transformación de algunas relaciones binarias 1:1 y 1:N.

Si una de las entidades es opcional en la interrelación, y la transformación ha consistido en poner una clave foránea en la relación que corresponde a la otra entidad, entonces esta clave foránea puede tomar valores nulos.

Ejemplo de transformación de una entidad opcional en la interrelación

En el ejemplo siguiente, la entidad *departamento* es opcional en *dirección* y, por lo tanto, puede haber empleados que no sean directores de ningún departamento.



En principio, hay dos opciones de transformación:

- Primera opción:

```
DEPARTAMENTO(dep, ..., emp-dir)
  donde {emp-dir} referencia EMPLEADO
EMPLEADO(emp, ...)
```

- Segunda opción:

```
DEPARTAMENTO(dep, ...)
EMPLEADO(emp, ..., dep)
  donde {dep} referencia DEPARTAMENTO
  y dep puede tomar valores nulos
```

La segunda transformación da lugar a una clave foránea que puede tomar valores nulos (porque puede haber empleados que no son directores de ningún departamento). Entonces será preferible la primera transformación, porque no provoca la aparición de valores nulos en la clave foránea y, de este modo, nos ahorra espacio de almacenamiento.

En las interrelaciones 1:N, el hecho de que la entidad del lado 1 sea opcional también provoca que la clave foránea de la transformación pueda tener valores nulos. En este caso, sin embargo, no se pueden evitar estos valores nulos porque hay una única transformación posible.

3.4. Transformación de interrelaciones ternarias

La transformación de las interrelaciones ternarias presenta similitudes importantes con la transformación de las binarias M:N. No es posible representar la interrelación mediante claves foráneas, sino que es necesario usar una nueva relación. Para que la nueva relación refleje toda la información que modeliza la interrelación, es necesario que contenga las claves primarias de las tres entidades interrelacionadas y los atributos de la interrelación.

Así pues, la transformación de una interrelación ternaria siempre da lugar a una nueva relación, que tendrá como atributos las claves primarias de las tres entidades interrelacionadas y todos los atributos que tenga la interrelación. La clave primaria de la nueva relación depende de la conectividad de la interrelación.

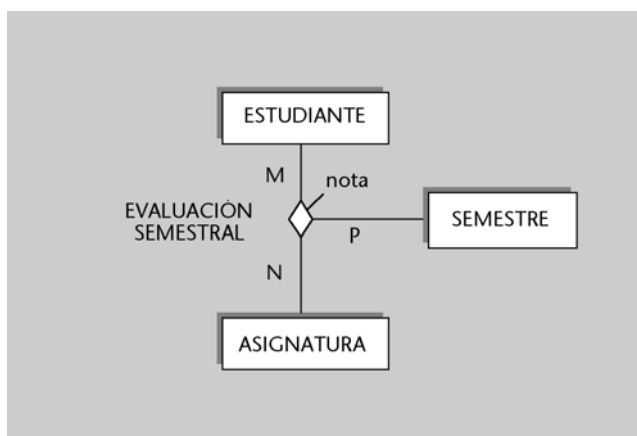
A continuación analizaremos cuál debe ser la clave primaria de la nueva relación según la conectividad. Empezaremos por el caso M:N:P y acabaremos con el caso 1:1:1. !

3.4.1. Conectividad M:N:P

Cuando la conectividad de la interrelación es M:N:P, la relación que se obtiene de su transformación tiene como clave primaria todos los atributos que forman las claves primarias de las tres entidades interrelacionadas.

Ejemplo de transformación de una interrelación ternaria M:N:P

Analizaremos la transformación con un ejemplo:



La interrelación anterior se transforma en:

```
ESTUDIANTE(est, ...)
ASIGNATURA(asig, ...)
SEMESTRE(sem, ...)
EVALUACIÓN-SEMESTRAL(est, asig, sem, nota)
    donde {est} referencia ESTUDIANTE,
    {asig} referencia ASIGNATURA
    y {sem} referencia SEMESTRE
```

Para identificar completamente la relación, la clave debe constar de la clave de *estudiante*, de la clave de *asignatura* y de la clave de *semestre*. Si nos faltase una de las tres, la clave de la relación podría tener valores repetidos. Consideremos, por ejemplo, que no tuviésemos la clave de *semestre*. Dado que *semestre* está conectada con “muchos” en la interrelación, puede haber estudiantes que han sido evaluados de una misma asignatura en más de un semestre. Entonces, para estos casos habría valores repetidos en la clave de la relación *EVALUACION-SEMESTRAL*.

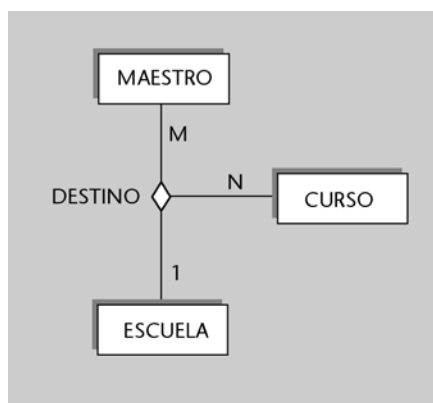
Observad que, del mismo modo que es necesaria la clave de semestre, también lo son la de *estudiante* y la de *asignatura*.

3.4.2. Conectividad M:N:1

Cuando la conectividad de la interrelación es M:N:1, la relación que se obtiene de su transformación tiene como clave primaria todos los atributos que forman las claves primarias de las dos entidades de los lados de la interrelación etiquetados con M y con N.

Ejemplo de transformación de una interrelación ternaria M:N:1

Lo ilustraremos con un ejemplo:



Esta interrelación refleja los destinos que se dan a los maestros de escuela en los diferentes cursos. El 1 que figura en el lado de *escuela* significa que un maestro no puede ser destinado a más de una escuela en un mismo curso.

El ejemplo de la figura se transforma en:

```
MAESTRO(código-maestro, ...)
CURSO(código-curso, ...)
ESCUELA(código-esc, ...)
DESTINO(código-maestro, código-curso, código-esc)
    donde {código-maestro} referencia MAESTRO
    {código-curso} referencia CURSO
    y {código-esc} referencia ESCUELA
```

No es necesario que la clave incluya *código-esc* para identificar completamente la relación. Si se fijan un maestro y un curso, no puede haber más de una escuela de destino y, por lo tanto, no habrá claves repetidas.

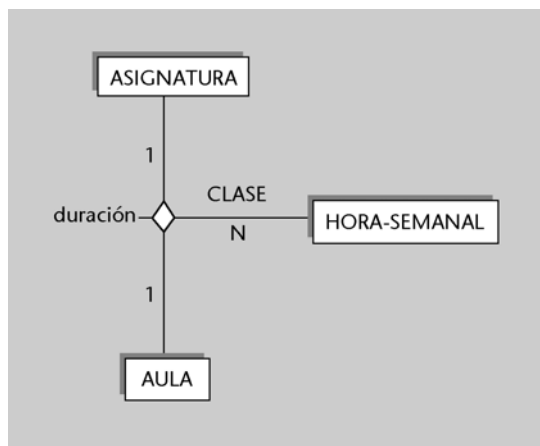
3.4.3. Conectividad N:1:1

Cuando la conectividad de la interrelación es N:1:1, la relación que se consigue de su transformación tiene como clave primaria los atributos que forman la clave primaria de la entidad del lado N y los atributos que forman la clave primaria de cualquiera de las dos entidades que están conectadas con 1.

Así pues, hay dos posibles claves para la relación que se obtiene. Son dos claves candidatas entre las cuales el diseñador deberá escoger la primaria.

Ejemplo de transformación de una interrelación ternaria N:1:1

Veamos un ejemplo de ello:



1) Una posible transformación es la siguiente:

```
HORA-SEMANAL(código-hora, ...)
AULA(código-aula, ...)
ASIGNATURA(asig, ...)
CLASE (código-hora, código-aula, asig, duración)
    donde {código-hora} referencia HORA-SEMANAL,
    {código-aula} referencia AULA
    y {asig} referencia ASIGNATURA
```

En este caso, la clave, a pesar de no incluir el atributo *asig*, identifica completamente la relación porque para una hora-semanal y un aula determinadas hay una única asignatura de la que se hace clase a esa hora y en esa aula.

2) La segunda transformación posible es esta otra:

```
HORA-SEMANAL(código-hora, ...)
AULA(código-aula, ...)
ASIGNATURA(asig, ...)
CLASE (código-hora, código-aula, asig, duración)
    donde {código-hora} referencia HORA-SEMANAL,
    {código-aula} referencia AULA
    y {asig} referencia ASIGNATURA
```

Ahora la clave incluye el atributo *asig* y, en cambio, no incluye el atributo *código-aula*. La relación también queda completamente identificada porque, para una asignatura y hora-semanal determinadas, de aquella asignatura se da clase en una sola aula a aquella hora.

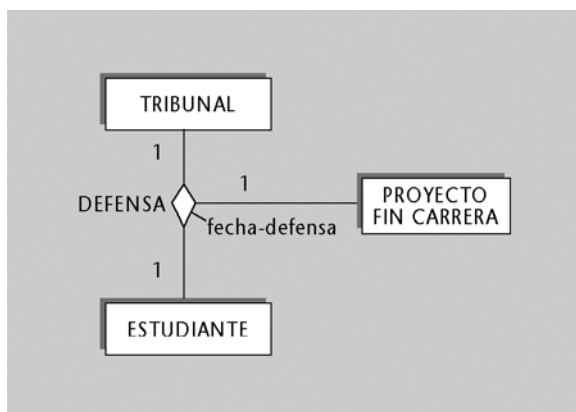
3.4.4. Conectividad 1:1:1

Cuando la conectividad de la interrelación es 1:1:1, la relación que se obtiene de su transformación tiene como clave primaria los atributos que forman la clave primaria de dos entidades cualesquiera de las tres interrelacionadas.

Así pues, hay tres claves candidatas para la relación.

Ejemplo de transformación de una interrelación ternaria 1:1:1

Veamos un ejemplo de ello:



Esta interrelación registra información de defensas de proyectos de fin de carrera. Intervienen en ella el estudiante que presenta el proyecto, el proyecto presentado y el tribunal evaluador.

La transformación del ejemplo anterior se muestra a continuación:

```

TRIBUNAL(trib, ...)
ESTUDIANTE(est, ...)
PROYECTO-FIN-CARRERA(pro, ...)

```

Hemos considerado que, ...

... si dos estudiantes presentan un mismo proyecto de fin de carrera, el tribunal será necesariamente diferente.

Para la nueva relación DEFENSA, tenemos las tres posibilidades siguientes:

- Primera opción:

```

DEFENSA(trib, est, pro, fecha-defensa)
donde {trib} referencia TRIBUNAL,
{est} referencia ESTUDIANTE
y {pro} referencia PROYECTO-FIN-CARRERA

```

- Segunda opción:

```

DEFENSA(trib, pro, est, fecha-defensa)
donde {trib} referencia TRIBUNAL,
{est} referencia ESTUDIANTE
y {pro} referencia PROYECTO-FIN-CARRERA

```

- Tercera opción:

DEFENSA(est, pro, trib, fecha-defensa)
donde {trib} referencia TRIBUNAL,
{est} referencia ESTUDIANTE
y {pro} referencia PROYECTO-FIN-CARRERA

En los tres casos, es posible comprobar que la clave identifica completamente la relación si se tiene en cuenta la conectividad de la interrelación *defensa*.

3.5. Transformación de interrelaciones n -arias

La transformación de las interrelaciones n -arias se puede ver como una generalización de lo que hemos explicado para las ternarias. !

En todos los casos, la transformación de una interrelación n -aria consistirá en la obtención de una nueva relación que contiene todos los atributos que forman las claves primarias de las n entidades interrelacionadas y todos los atributos de la interrelación.


Podemos distinguir los casos siguientes:

- a) Si todas las entidades están conectadas con “muchos”, la clave primaria de la nueva relación estará formada por todos los atributos que forman las claves de las n entidades interrelacionadas.
- b) Si una o más entidades están conectadas con “uno”, la clave primaria de la nueva relación estará formada por las claves de $n - 1$ de las entidades interrelacionadas, con la condición de que la entidad, cuya clave no se ha incluido, debe ser una de las que está conectada con “uno”.

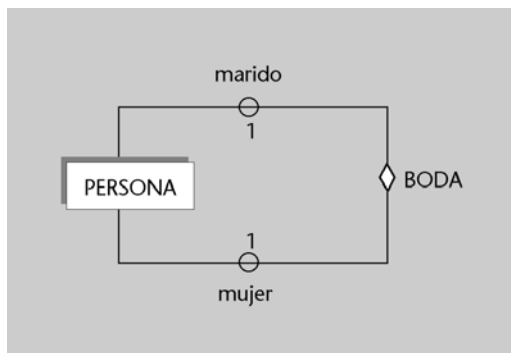
3.6. Transformación de interrelaciones recursivas

Las transformaciones de las interrelaciones recursivas son similares a las que hemos visto para el resto de las interrelaciones.

De este modo, si una interrelación recursiva tiene conectividad 1:1 o 1:N, da lugar a una clave foránea, y si tiene conectividad M:N o es n -aria, origina una nueva relación.

Mostraremos la transformación de algunos ejemplos concretos de interrelaciones recursivas para ilustrar los detalles de la afirmación anterior. 

Ejemplo de transformación de una interrelación recursiva binaria 1:1



La interrelación de la figura anterior es recursiva, binaria y tiene conectividad 1:1. Las interrelaciones 1:1 originan una clave foránea que se pone en la relación correspondiente a una de las entidades interrelacionadas. En nuestro ejemplo, sólo hay una entidad interrelacionada, la entidad *persona*. Entonces, la clave foránea deberá estar en la relación *PERSONA*. Esta clave foránea deberá referenciar a la misma relación para que refleje una interrelación entre una ocurrencia de persona y otra ocurrencia de persona. Así, obtenemos:

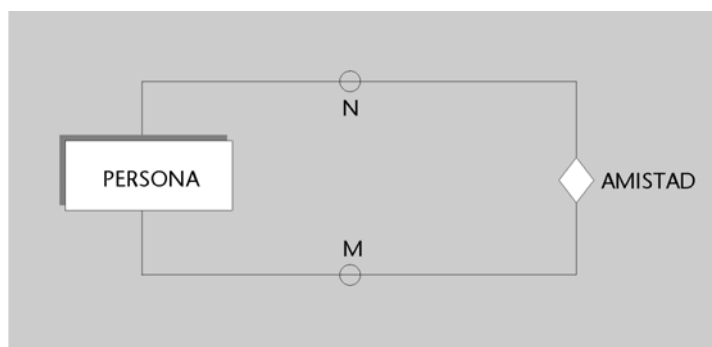
PERSONA (código-per, ..., código-conyuge)
 donde {código-conyuge} referencia *PERSONA*
 y código-conyuge admite valores nulos

La clave foránea {código-conyuge} referencia la relación *PERSONA* a la que pertenece.

Conviene tener en cuenta que, en casos como éste, los atributos de la clave foránea no pueden tener los mismos nombres que los atributos de la clave primaria que referencian porque, según la teoría relacional, una relación no puede tener nombres de atributos repetidos.

Ejemplo de transformación de una interrelación recursiva M:N

Veamos a continuación un ejemplo en el que interviene una interrelación recursiva y con conectividad M:N.



Las interrelaciones M:N se traducen en nuevas relaciones que tienen como clave primaria las claves de las entidades interrelacionadas.

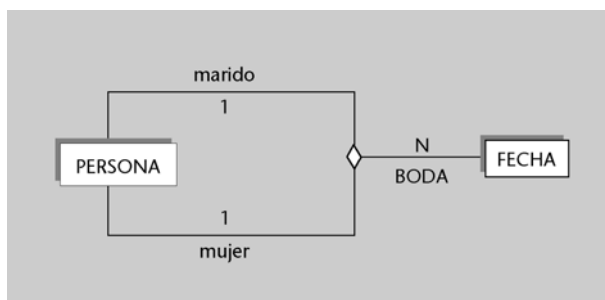
En nuestro ejemplo, la interrelación vincula ocurrencias de persona con otras ocurrencias de *persona*. En este caso, la clave primaria de la nueva relación estará formada por la clave de la

entidad *persona* dos veces. Convendrá dar nombres diferentes a todos los atributos de la nueva relación. De este modo, la traducción del ejemplo anterior será:

```
PERSONA (código-per, ...)
AMISTAD (código-per, código-per-amiga)
donde {código-per} referencia PERSONA
y {código-per-amiga} referencia PERSONA
```

Ejemplo de transformación de una interrelación recursiva *n*-aria N:1:1

Finalmente, analizaremos un ejemplo en el que la interrelación recursiva es *n*-aria:



La anterior interrelación *boda* es recursiva, ternaria y tiene conectividad N:1:1. Las interrelaciones N:1:1 originan siempre una nueva relación que contiene, además de los atributos de la interrelación, todos los atributos que forman la clave primaria de las tres entidades interrelacionadas.

En nuestro ejemplo, la interrelación asocia ocurrencias de *persona* con otras ocurrencias de *persona* y con ocurrencias de *fecha*. Entonces, la clave de *persona* tendrá que figurar dos veces en la nueva relación, y la clave de *fecha*, solo una.

La clave primaria de la relación que se obtiene para interrelaciones N:1:1 está formada por la clave de la entidad del lado N y por la clave de una de las entidades de los lados 1.

En nuestro ejemplo, en los dos lados 1 de la interrelación tenemos la misma entidad: *persona*. La clave primaria estará formada por la clave de la entidad *fecha* y por la clave de la entidad *persona*.

Según todo esto, la transformación será la siguiente:

```
PERSONA(código-per, ...)
FECHA(fecha-bod, ...)
BODA (fecha-bod, código-per, código-conyuge)
donde {fecha-bod} referencia FECHA,
{código-per} referencia PERSONA
y {código-conyuge} referencia PERSONA
```

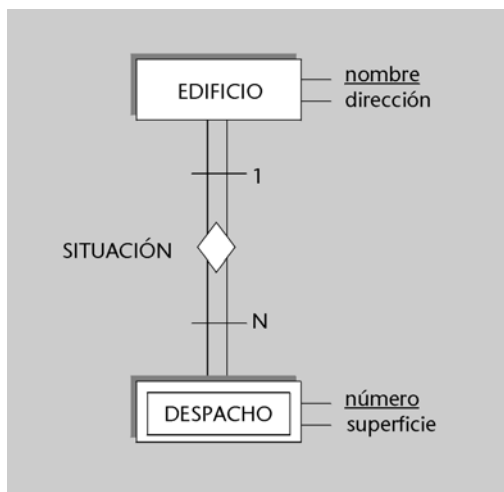
3.7. Transformación de entidades débiles

Las entidades débiles se traducen al modelo relacional igual que el resto de entidades, con una pequeña diferencia. Estas entidades siempre están en el lado N de una interrelación 1:N que completa su identificación.

Así pues, la clave foránea originada por esta interrelación 1:N debe formar parte de la clave primaria de la relación correspondiente a la entidad débil.

Ejemplo de transformación de entidad débil

Lo explicaremos con un ejemplo:



Este ejemplo se transforma tal y como se muestra a continuación:

```
EDIFICIO(nombre, dirección)
DESPACHO(nombre, número, superficie)
donde {nombre} referencia EDIFICIO
```

Observad que la clave foránea {nombre} forma parte también de la clave primaria de *DESPACHO*. Si no fuese así, y la clave primaria contuviese sólo el atributo *número*, los despachos no quedarían totalmente identificados, teniendo en cuenta que puede haber despachos situados en edificios diferentes que tengan el mismo número.

3.8. Transformación de la generalización/especialización

Cada una de las entidades superclase y subclase que forman parte de una generalización/especialización se transforma en una relación:

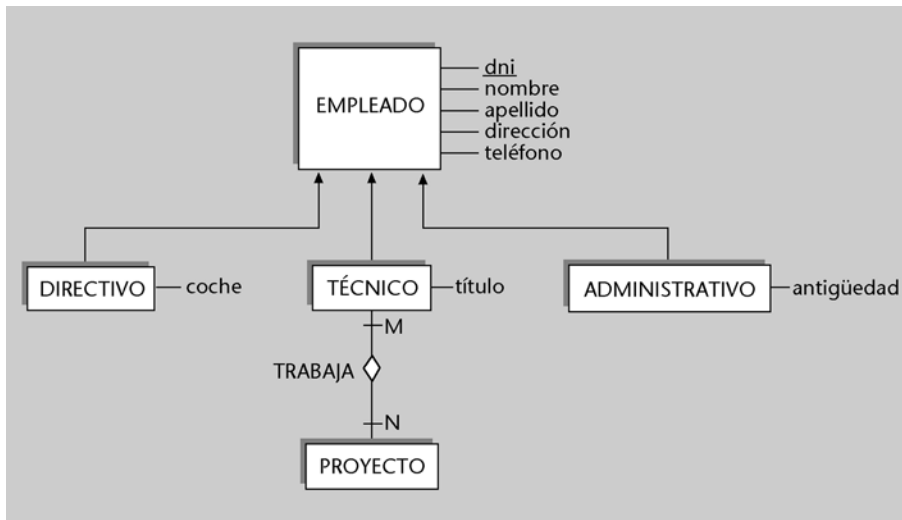
- a) La relación de la entidad superclase tiene como clave primaria la clave de la entidad superclase y contiene todos los atributos comunes.
- b) Las relaciones de las entidades subclase tienen como clave primaria la clave de la entidad superclase y contienen los atributos específicos de la subclase.

Ejemplo de transformación de la generalización/especialización

Veamos un ejemplo (consultad el gráfico en la página siguiente) que contiene una generalización/especialización y, también, una interrelación M:N en la que interviene una de las entidades subclase. Este ejemplo se traduce al modelo relacional como se indica a continuación:

```
EMPLEADO(DNI, nombre, dirección, teléfono)
DIRECTIVO(DNI, coche)
donde {DNI} referencia EMPLEADO
```

ADMINISTRATIVO (DNI, antigüedad)
 donde {DNI} referencia EMPLEADO
 TÉCNICO (DNI, título)
 donde {DNI} referencia EMPLEADO
 PROYECTO(pro, ...)
 TRABAJA(DNI, pro, superficie)
 donde {DNI} referencia TÉCNICO
 y {pro} referencia PROYECTO



Conviene observar que los atributos comunes se han situado en la relación *EMPLEADO* y que los atributos específicos se han situado en la relación de su entidad subclase. De este modo, *coche* está en *DIRECTIVO*, *título* en *TÉCNICO* y *antigüedad* en *ADMINISTRATIVO*.

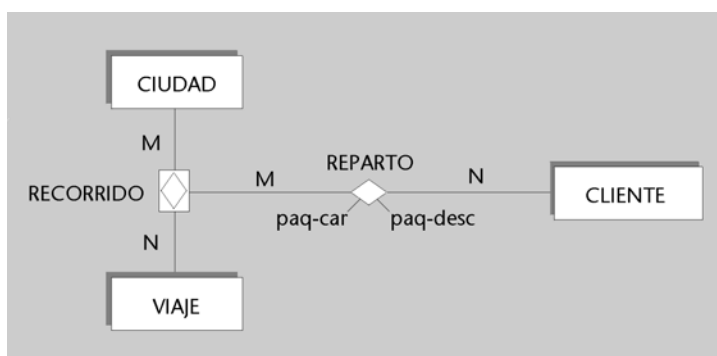
Por otro lado, la interrelación específica para los empleados técnicos denominada *trabaja* se transforma en la relación *TRABAJA*. Observad que esta relación tiene una clave foránea que referencia sólo a los empleados técnicos, y no a los empleados directivos o administrativos.

3.9. Transformación de entidades asociativas

Una entidad asociativa tiene su origen en una interrelación. En consecuencia, sucede que la transformación de la interrelación originaria es, al mismo tiempo, la transformación de la entidad asociativa.

Ejemplo de transformación de una entidad asociativa

Veamos un ejemplo, que incluye una entidad asociativa interrelacionada con otra entidad:



La transformación del ejemplo anterior será:


```

CIUDAD(nombre-ciudad, ...)
VIAJE(id-viaje, ...)
RECORRIDO (nombre-ciudad, id-viaje)
    donde {nombre-ciudad} referencia CIUDAD
    e {id-viaje} referencia VIAJE
CLIENTE {código-cliente, ...}
REPARTO(nombre-ciudad, id-viaje, código-cliente, paq-car, paq-desc)
    donde {nombre-ciudad, id-viaje} referencia RECORRIDO
    y {código-cliente} referencia CLIENTE
  
```

Tal y como se puede observar, la traducción de la interrelación *recorrido* es, al mismo tiempo, la traducción de su entidad asociativa.

La relación *REPARTO* nos ilustra la transformación de una interrelación en la que participa una entidad asociativa. Puesto que se trata de una interrelación M:N entre *recorrido* y *ciudad*, una parte de la clave primaria de *REPARTO* referencia la clave de *RECORRIDO*, y el resto, la clave de *CIUDAD*.

3.10. Resumen de la transformación del modelo ER al modelo relacional


La tabla que mostramos a continuación resume los aspectos más básicos de las transformaciones que hemos descrito en las secciones anteriores, con el objetivo de presentar una visión rápida de los mismos: 

Elemento del modelo ER	Transformación al modelo relacional
Entidad	Relación
Interrelación 1:1	Clave foránea
Interrelación 1:1	Clave foránea
Interrelación M:N	Relación
Interrelación <i>n</i> -aria	Relación
Interrelación recursiva	Como en las interrelaciones no recursivas: <ul style="list-style-type: none"> • Clave foránea para binarias 1:1 y 1:N • Relación para binarias M:N y <i>n</i>-arias
Entidad débil	La clave foránea de la interrelación identificadora forma parte de la clave primaria
Generalización/especialización	<ul style="list-style-type: none"> • Relación para la entidad superclase • Relación para cada una de las entidades subclase
Entidad asociativa	La transformación de la interrelación que la origina es a la vez su transformación

3.11. Ejemplo: base de datos del personal de una entidad bancaria

En este apartado aplicaremos las transformaciones que hemos explicado en el caso práctico de la base de datos del personal de una entidad bancaria. Antes hemos presentado el diseño conceptual de esta base de datos. A continuación, veremos su transformación al modelo relacional.

Empezaremos por transformar todas las entidades en relaciones y todas las interrelaciones 1:1 y 1:N en claves foráneas de estas relaciones.

 Hemos presentado el diseño conceptual de la base de datos del personal de la entidad bancaria en el subapartado 2.3 de esta unidad didáctica.

EMPLEADO(código-empleado, DNI, NSS, nombre, apellido, nombre-categ, central, ciudad-res)
 donde {nombre-categ} referencia CATEGORÍA,
 {central} referencia CENTRAL-SINDICAL,
 el atributo central admite valores nulos
 y {ciudad-res} referencia CIUDAD
 FIJO(código-empleado, antigüedad)
 donde {código-empleado} referencia EMPLEADO
 TEMPORAL(código-empleado, fecha-inicio-cont, fecha-final-cont)
 donde {código-empleado} referencia EMPLEADO
 CIUDAD(nombre-ciudad, número-hab)
 AGENCIA(nombre-ciudad, nombre-agencia, dirección, teléfono)
 donde {nombre-ciudad} referencia CIUDAD
 TÍTULO(nombre-título)
 CATEGORÍA(nombre-categoría, sueldo-base, hora-extra)
 CENTRAL-SINDICAL(central, cuota)
 TIPO-PRÉSTAMO(código-préstamo, tipo-interés, período-vigencia)
 FECHA(fecha)

Observad que, en la transformación de la generalización/especialización correspondiente a la entidad *empleado*, hemos situado los atributos comunes a la relación *EMPLEADO* y los atributos específicos se han situado en las relaciones *FIJO* y *TEMPORAL*.

En la relación *AGENCIA*, el atributo *nombre-ciudad* es una clave foránea y al mismo tiempo forma parte de la clave primaria porque *agencia* es una entidad débil que requiere la interrelación *situacion* para ser identificada.

Veamos ahora las relaciones que se obtienen a partir de la transformación de las interrelaciones binarias y *n*-arias:

TITULACIÓN(código-empleado, nombre-título)
 donde {código-empleado} referencia EMPLEADO
 y {nombre-título} referencia TÍTULO
 TRASLADO(código-empleado, fecha, nombre-ciudad, nombre-agencia, fecha-fin)
 donde {código-empleado} referencia EMPLEADO,
 {nombre-ciudad, nombre-agencia} referencia AGENCIA
 y {fecha} referencia FECHA
 PETICIÓN(código-empleado, código-préstamo, fecha, concedido/no)
 donde {código-empleado} referencia FIJO
 {código-préstamo} referencia TIPO-PRÉSTAMO
 y {fecha} referencia FECHA


Para elegir las claves primarias adecuadas, se ha tenido en cuenta la conectividad de las interrelaciones.

Resumen

Esta unidad es una introducción a un tema de gran interés: el **diseño de bases de datos**.

En primer lugar, hemos explicado qué se entiende por *diseñar una base de datos* y hemos analizado las etapas en las que se puede descomponer el proceso de diseño:

- la **etapa del diseño conceptual**,
- la **etapa del diseño lógico**,
- la **etapa del diseño físico**.

En el resto de la unidad hemos tratado el diseño conceptual y el diseño lógico de la base de datos. No hemos estudiado el diseño físico porque requiere unos conocimientos previos de estructuras de implementación física que hacen inadecuado explicarlo en este curso. 

Para el diseño conceptual hemos adoptado el enfoque del **modelo ER**, un modelo de datos muy utilizado y comprensible. Hemos descrito las diversas construcciones que proporciona y hemos dado ejemplos de aplicación a casos prácticos.

En lo que respecta al diseño lógico, lo hemos centrado en el caso de utilización de la **tecnología relacional**. De este modo, hemos explicado cómo se puede transformar un modelo conceptual expresado mediante el modelo ER en una estructura de datos del modelo relacional.

Ejercicios de autoevaluación

1. Haced un diseño conceptual de una base de datos mediante el modelo ER que satisfaga los requisitos que se resumen a continuación:

- a) Un directivo de un club de fútbol quiere disponer de una base de datos que le permita controlar datos que le interesan sobre competiciones, clubes, jugadores, entrenadores, etc. de ámbito estatal.
- b) Los clubes disputan cada temporada varias competiciones (liga, copa, etc.) entre sí. Nuestro directivo desea información histórica de las clasificaciones obtenidas por los clubes en las diferentes competiciones a lo largo de todas las temporadas. La clasificación se especificará mediante un número de posición: 1 significa campeón, 2 significa subcampeón, etc.
- c) Los distintos clubes están agrupados en las federaciones regionales correspondientes. Toda federación tiene como mínimo un club. Quiere saber el nombre y la fecha de creación de las federaciones así como el nombre y el número de socios de los clubes.
- d) Es muy importante la información sobre jugadores y entrenadores. Se identificarán por un código, y quiere saber el nombre, la dirección, el número de teléfono y la fecha de nacimiento de todos. Es necesario mencionar que algunos entrenadores pueden haber sido jugadores en su juventud. De los jugadores, además, quiere saber el peso, la altura, la especialidad o las especialidades y qué dominio tienen de ellas (grado de especialidad). Todo jugador debe tener como mínimo una especialidad, pero puede haber especialidades en las que no haya ningún jugador. De los entrenadores le interesa la fecha en que iniciaron su carrera como entrenadores de fútbol.
- e) De todas las personas que figuran en la base de datos (jugadores y entrenadores), quiere conocer el historial de contrataciones por parte de los diferentes clubes, incluyendo el importe y la fecha de baja de cada contratación. En un momento determinado, una persona puede estar contratada por un único club, pero puede cambiar de club posteriormente e, incluso, puede volver a un club en el que ya había trabajado.
- f) También quiere registrar las ofertas que las personas que figuran en la base de datos han recibido de los clubes durante su vida deportiva (y de las que se ha enterado). Considera básico tener constancia del importe de las ofertas. Se debe tener en cuenta que, en un momento determinado, una persona puede recibir muchas ofertas, siempre que provengan de clubes distintos.

2. Haced un diseño conceptual de una base de datos mediante el modelo ER que satisfaga los requisitos que se resumen a continuación:

- a) Se quiere diseñar una base de datos para facilitar la gestión de una empresa dedicada al transporte internacional de mercancías que opera en todo el ámbito europeo.
- b) La empresa dispone de varias delegaciones repartidas por toda la geografía europea. Las delegaciones se identifican por un nombre, y se quiere registrar también su número de teléfono. En una determinada ciudad no hay nunca más de una delegación. Se desea conocer la ciudad donde está situada cada delegación. Se debe suponer que no hay ciudades con el nombre repetido (por lo menos en el ámbito de esta base de datos).
- c) El personal de la empresa se puede separar en dos grandes grupos:
 - Administrativos, sobre los cuales interesa saber su nivel de estudios.
 - Conductores, sobre los que interesa saber el año en el que obtuvieron el carnet de conducir y el tipo de carnet que tienen.De todo el personal de la empresa, se quiere conocer el código de empleado (que lo identifica), su nombre, su número de teléfono y el año de nacimiento. Todos los empleados están asignados a una delegación determinada. Se quiere tener constancia histórica de este hecho teniendo en cuenta que pueden ir cambiando de delegación (incluso pueden volver a una delegación donde ya habían estado anteriormente).
- d) La actividad de la empresa consiste en efectuar los viajes pertinentes para transportar las mercancías según las peticiones de sus clientes. Todos los clientes se identifican por un código de cliente. Se quiere conocer, además, el nombre y el teléfono de contacto de cada uno de ellos.
- e) La empresa, para llevar a cabo su actividad, dispone de muchos camiones identificados por un código de camión. Se quiere tener constancia de la matrícula, la marca y la tara de los camiones.
- f) Los viajes los organiza siempre una delegación, y se identifican mediante un código de viaje, que es interno de cada delegación (y que se puede repetir en delegaciones diferentes). Para cada uno de los viajes que se han hecho, es necesario saber:
 - Qué camión se ha utilizado (ya que cada viaje se hace con un solo camión).
 - Qué conductor o conductores han ido (considerando que en viajes largos pueden ir varios conductores). Se quiere saber también el importe de las dietas pagadas a cada conductor (teniendo en cuenta que las dietas pueden ser diferentes para los diferentes conductores de un mismo viaje).
 - El recorrido del viaje; es decir, la fecha y la hora en que el camión llega a cada una de las ciudades donde deberá cargar o descargar. Supondremos que un viaje no pasa nunca dos veces por una misma ciudad.

- El número de paquetes cargados y de paquetes descargados en cada ciudad, y para cada uno de los clientes. En un mismo viaje se pueden dejar y/o recoger paquetes en diferentes ciudades por encargo de un mismo cliente. También, en un mismo viaje, se pueden dejar y/o recoger paquetes en una misma ciudad por encargo de diferentes clientes.

3. Haced un diseño conceptual de una base de datos mediante el modelo ER que satisfaga los requisitos que se resumen a continuación:

a) Es necesario diseñar una base de datos para una empresa inmobiliaria con el objetivo de gestionar la información relativa a su cartera de pisos en venta.

b) Cada uno de los pisos que tienen pendientes de vender tiene asignado un código de piso que lo identifica. Además de este código, se quiere conocer la dirección del piso, la superficie, el número de habitaciones y el precio. Tienen estos pisos clasificados por zonas (porque a sus clientes, en ocasiones, sólo les interesan los pisos de una zona determinada) y se quiere saber en qué zona está situado cada piso. Las zonas tienen un nombre de zona que es diferente para cada una de una misma población, pero que pueden coincidir en zonas de poblaciones diferentes. En ocasiones sucede que en algunas de las zonas no tienen ningún piso pendiente de vender.

c) Se quiere tener el número de habitantes de las poblaciones. Se quiere saber qué zonas son limítrofes, (porque, en caso de no disponer de pisos en una zona que desea un cliente, se le puedan ofrecer los que tengan en otras zonas limítrofes). Es necesario considerar que pueden existir zonas sin ninguna zona limítrofe en algunas poblaciones pequeñas que constan de una sola zona.

d) Se disponen de diferentes características codificadas de los pisos, como por ejemplo tener ascensor, ser exterior, tener terraza, etc. Cada característica se identifica mediante un código y tiene una descripción. Para cada característica y cada piso se quiere saber si el piso satisface la característica o no. Además, quieren tener constancia del propietario o los propietarios de cada piso.

e) También necesitan disponer de información relativa a sus clientes actuales que buscan piso (si dos o más personas buscan piso conjuntamente, sólo se guarda información de una de ellas como cliente de la empresa). En particular, interesa saber las zonas donde busca piso cada cliente (sólo en caso de que tenga alguna zona de preferencia).

f) A cada uno de estos clientes le asignan un vendedor de la empresa para que se ocupe de atenderlo. A veces, estas asignaciones varían con el tiempo y se cambia al vendedor asignado a un determinado cliente. También es posible que a un cliente se le vuelva a asignar un vendedor que ya había tenido con anterioridad. Se quiere tener constancia de las asignaciones de los clientes actuales de la empresa.

g) Los vendedores, clientes y propietarios se identifican por un código de persona. Se quiere registrar, de todos, su nombre, dirección y número de teléfono. Además, se quiere disponer del número de Seguridad Social y el sueldo de los vendedores, y del NIF de los propietarios. Puede haber personas que sean al mismo tiempo clientes y propietarios, o bien vendedores y propietarios, etc.

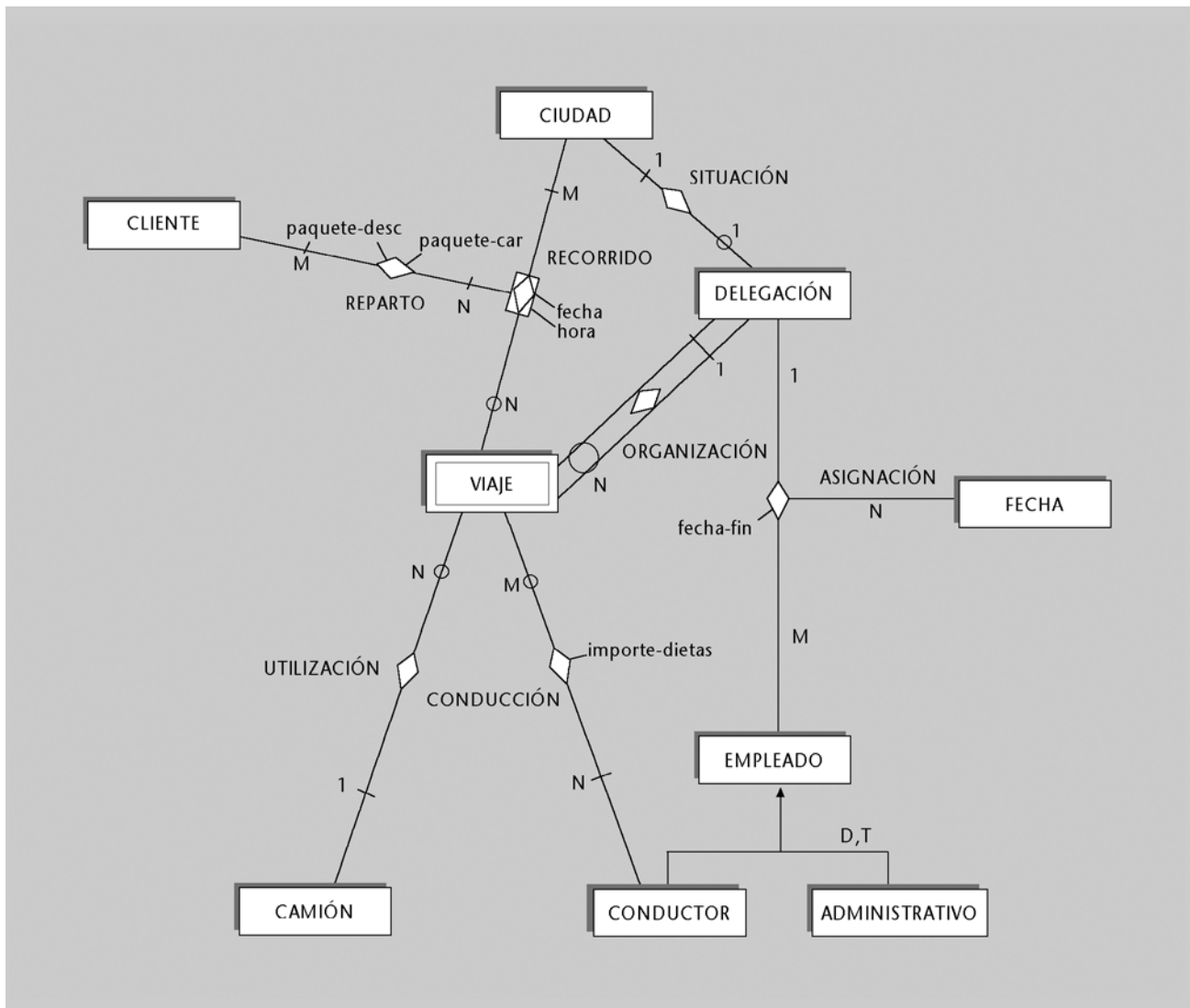
h) Finalmente, para ayudar a programar y consultar las visitas que los clientes hacen a los pisos en venta, se quiere guardar información de todas las visitas correspondientes a los clientes y a los pisos actuales de la empresa. De cada visita hay que saber el cliente que la hace, el piso que se va a ver y la hora concreta en que se inicia la visita. Entendemos que la hora de la visita está formada por la fecha, la hora del día y el minuto del día (por ejemplo, 25-FEB-98, 18:30). Hay que considerar que un mismo cliente puede visitar un mismo piso varias veces para asegurarse de si le gusta o no, y también que para evitar conflictos no se programan nunca visitas de clientes diferentes a un mismo piso y a la misma hora.

4. Transformad a relacional el diseño conceptual que habéis obtenido en el ejercicio 1.

5. Transformad a relacional el diseño conceptual que habéis obtenido en el ejercicio 2.

6. Transformad a relacional el diseño conceptual que habéis obtenido en el ejercicio 3.

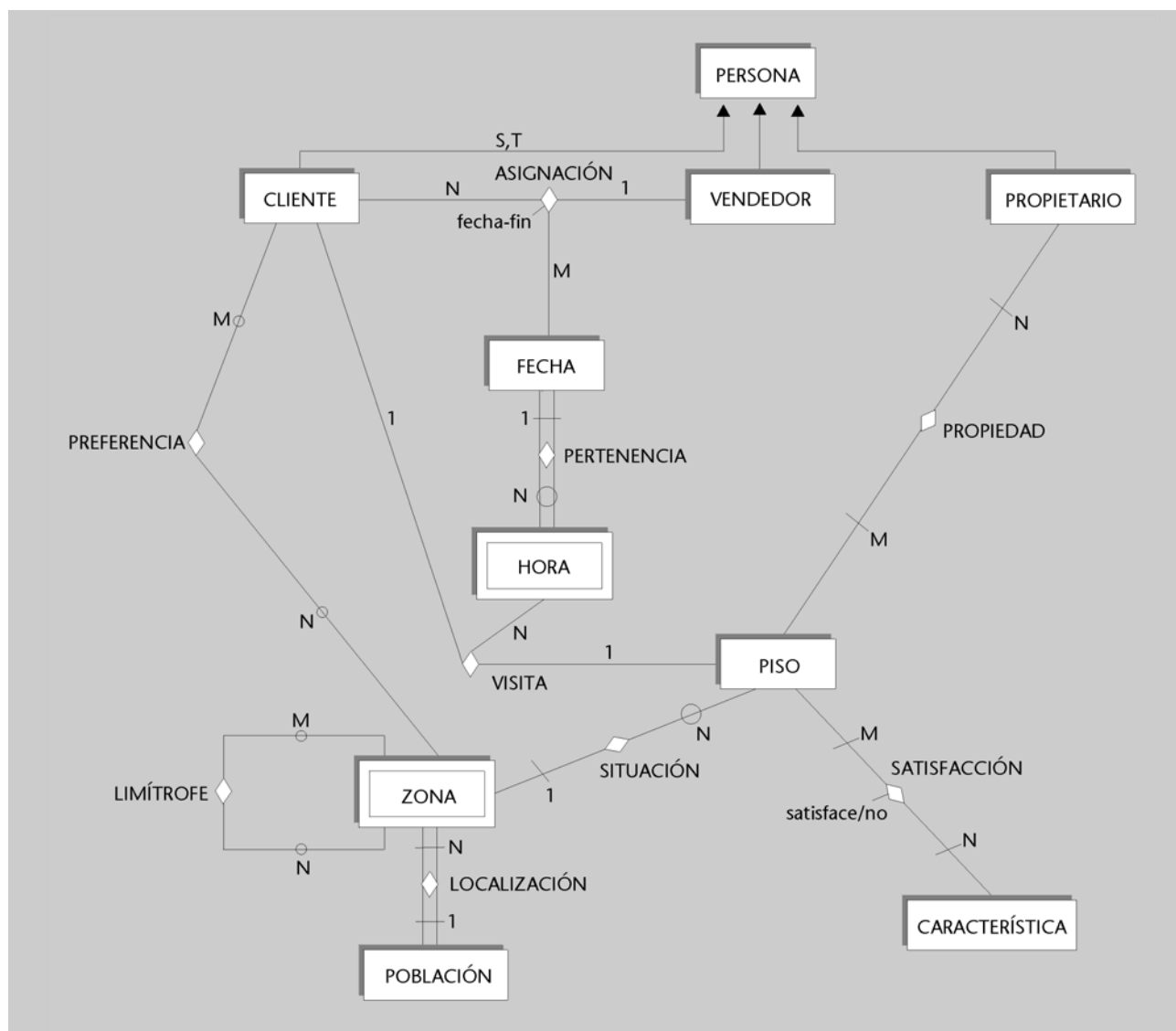
2. La siguiente figura muestra un diagrama ER que satisface los requisitos que se han descrito:



Los atributos de las entidades que figuran en el diagrama son los siguientes (las claves primarias se han subrayado):

CIUDAD
nombre-ciudad
 DELEGACIÓN
nombre-del, teléfono
 EMPLEADO
código-empleado, nombre, teléfono, año-nacimiento
 CONDUCTOR (subclase de empleado)
código-empleado, año-carnet, tipo-carnet
 ADMINISTRATIVO (subclase de empleado)
código-empleado, nivel-estudios
 FECHA
fecha
 VIAJE (entidad débil: código-viaje la identifica parcialmente, se identifica completamente con la delegación de organización).
código-viaje
 CAMIÓN
código-camion, matrícula, marca, tara
 CLIENTE
código-cliente, nombre, teléfono-contacto

3. La figura que podéis ver a continuación muestra un diagrama ER que satisface los requisitos que se han descrito:



Los atributos de las entidades que figuran en el diagrama son los siguientes (las claves primarias se han subrayado):

POBLACIÓN

nombre-pobl, número-hab

ZONA (entidad débil: nombre-zona la identifica parcialmente, se identifica completamente con la población de localización)

nombre-zona

PISO

código-piso, dirección, superficie, número-habitaciones, precio

CARACTERÍSTICA

código-car, descripción

PERSONA

código-persona, nombre, dirección, teléfono

VENDEDOR (entidad subclase de persona)

código-persona, nss, sueldo

CLIENTE (entidad subclase de persona)

código-persona

PROPIETARIO (entidad subclase de persona)

código-persona, nif

FECHA

fecha

HORA (entidad débil: hora-minuto la identifica parcialmente, se identifica completamente con la fecha de pertenencia)

hora-minuto

4. El resultado de la transformación a relacional del modelo ER propuesto como solución del ejercicio 1 consta de las siguientes relaciones:

COMPETICIÓN(nombre-comp)
 TEMPORADA(código-temp)
 FEDERACIÓN(nombre-fed, fecha-creación)
 CLUB(nombre-club, número-socios, nombre-fed)
 donde {nombre-fed} referencia FEDERACIÓN
 PERSONA(código-persona, nombre, dirección, teléfono, fecha-nacimiento)
 JUGADOR(código-persona, peso, altura)
 donde {código-persona} referencia PERSONA
 ENTRENADOR(código-persona, fecha-inicio-carrera)
 donde {código-persona} referencia PERSONA
 ESPECIALIDAD(nombre-esp)
 FECHA(fecha)
 HABILIDAD(código-persona, nombre-esp, grado)
 donde {código-persona} referencia JUGADOR
 y {nombre-esp} referencia ESPECIALIDAD
 CLASIFICACIÓN(nombre-comp, código-temp, nombre-club, num-posición)
 donde {nombre-comp} referencia COMPETICIÓN,
 {código-temp} referencia TEMPORADA
 y {nombre-club} referencia CLUB
 CONTRATO(código-persona, fecha, nombre-club, importe-cont, fecha-baja)
 donde {código-persona} referencia PERSONA,
 {fecha} referencia FECHA
 y {nombre-club} referencia CLUB
 OFERTA(código-persona, fecha, nombre-club, importe-oferta)
 donde {código-persona} referencia PERSONA,
 {fecha} referencia FECHA
 y {nombre-club} referencia CLUB

5. El resultado de la transformación a relacional del modelo ER propuesto como solución del ejercicio 2 consta de las siguientes relaciones:

CIUDAD(nombre-ciudad)
 DELEGACIÓN(nombre-del, teléfono, nombre-ciudad)
 donde {nombre-ciudad} referencia CIUDAD
 EMPLEADO(código-empleado, nombre, teléfono, año-nacimiento)
 CONDUCTOR(código-empleado, año-carnet, tipo-carnet)
 donde {código-empleado} referencia EMPLEADO
 ADMINISTRATIVO (código-empleado, nivel-estudios)
 donde {código-empleado} referencia EMPLEADO
 FECHA(fecha)
 VIAJE(nombre-del, código-viaje, código-camión)
 donde {nombre-del} referencia DELEGACIÓN
 y {código-camión} referencia CAMIÓN
 CAMIÓN(código-camión, matrícula, marca, tara)
 CLIENTE(código-cliente, nombre, teléfono-contacto)
 ASIGNACION(código-empleado, fecha, nombre-del, fecha-fin)
 donde {código-empleado} referencia EMPLEADO,
 {nombre-del} referencia DELEGACIÓN
 y {fecha} referencia FECHA
 CONDUCCIÓN(código-empleado, nombre-del, código-viaje, importe-dietas)
 donde {código-empleado} referencia CONDUCTOR
 y {nombre-del, código-viaje} referencia VIAJE
 RECORRIDO(nombre-ciudad, nombre-del, código-viaje, fecha, hora)
 donde {nombre-ciudad} referencia CIUDAD
 y {nombre-del, código-viaje} referencia VIAJE
 REPARTO (nombre-ciudad, nombre-del, código-viaje, código-cliente,
 paquetes-car, paquetes-desc)
 donde {nombre-ciudad, nombre-del, código-viaje} referencia RECORRIDO
 y {código-cliente} referencia CLIENTE

6. El resultado de la transformación a relacional del modelo ER propuesto como solución del ejercicio 3 consta de las siguientes relaciones:

POBLACIÓN(nombre-pobl, nombre-hab)
 ZONA(nombre-pobl, nom-zona)
 donde {nombre-pobl} referencia POBLACIÓN
 PISO(código-piso, dirección, superficie, número-habitaciones, precio,
 nombre-pobl, nombre-zona)
 donde {nombre-pobl, nombre-zona} referencia ZONA
 CARACTERÍSTICA(código-car, descripción)
 PERSONA(código-persona, nombre, dirección, teléfono)
 VENDEDOR(código-persona, nss, sueldo)
 donde {código-persona} referencia PERSONA
 CLIENTE(código-persona)
 donde {código-persona} referencia PERSONA
 PROPIETARIO(código-persona, nif)
 donde {código-persona} referencia PERSONA
 FECHA(fecha)
 HORA(fecha, hora-minuto)
 donde {fecha} referencia FECHA
 ASIGNACIÓN(código-cliente, fecha, código-vendedor, fecha-fin)
 donde {código-cliente} referencia CLIENTE,
 {fecha} referencia FECHA
 y {código-vendedor} referencia VENDEDOR
 PREFERENCIA(código-persona, nombre-pobl, nombre-zona)
 donde {código-persona} referencia CLIENTE
 y {nombre-pobl, nom-zona} referencia ZONA
 SATISFACCIÓN(código-piso, código-car, satisface/no)
 donde {código-piso} referencia PISO
 y {código-car} referencia CARACTERÍSTICA
 LÍMITROFE(nombre-pobl, nombre-zona, nombre-pobl-lim, nombre-zona-lim)
 donde {nombre-pobl, nombre-zona} referencia ZONA
 y {nombre-pobl-lim, nombre-zona-lim} referencia ZONA
 PROPIEDAD(código-piso, código-persona)
 donde {código-piso} referencia PISO
 y {código-persona} referencia PROPIETARIO

Para la interrelación *visita*, hay dos transformaciones posibles:

1)

VISITA(fecha, hora-minuto, código-piso, código-persona)
 donde {fecha, hora-minuto} referencia HORA,
 {código-persona} referencia CLIENTE
 y {código-piso} referencia PISO

2)

VISITA(fecha, hora-minuto, código-persona, código-piso)
 donde {fecha, hora-minuto} referencia HORA,
 {código-persona} referencia CLIENTE
 y {código-piso} referencia PISO

Glosario

atributo de una entidad

Propiedad que interesa de una entidad.

atributo de una interrelación

Propiedad que interesa de una interrelación.

conectividad de una interrelación

Expresión del tipo de correspondencia entre las ocurrencias de entidades asociadas con la interrelación.

diseño conceptual

Etapa del diseño de una base de datos que obtiene una estructura de la información de la futura BD independiente de la tecnología que se quiere utilizar.

diseño físico

Etapa del diseño de una base de datos que transforma la estructura obtenida en la etapa del diseño lógico con el objetivo de conseguir una mayor eficiencia y que, además, la completa con aspectos de implementación física que dependerán del SGBD que se debe utilizar.

diseño lógico

Etapa del diseño de una base de datos que parte del resultado del diseño conceptual y lo transforma de modo que se adapte al modelo del SGBD con el que se desea implementar la base de datos.

entidad

Objeto del mundo real que podemos distinguir del resto de los objetos y del cual nos interesan algunas propiedades.

entidad asociativa

Entidad resultante de considerar una interrelación entre entidades como una nueva entidad.

entidad débil

Entidad cuyos atributos no la identifican completamente, sino que sólo la identifican de forma parcial.

entidad obligatoria en una interrelación binaria

Entidad tal que una ocurrencia de la otra entidad que interviene en la interrelación sólo puede existir si se da como mínimo una ocurrencia de la entidad obligatoria a la que está asociada.

entidad opcional en una interrelación binaria

Entidad tal que una ocurrencia de la otra entidad que interviene en la interrelación puede existir aunque no haya ninguna ocurrencia de la entidad opcional a la que está asociada.

generalización/especialización

Construcción que permite reflejar que existe una entidad general, denominada *entidad superclase*, que se puede especializar en entidades subclase. La entidad superclase nos permite modelizar las características comunes de la entidad vista a un nivel genérico, y con las entidades subclase podemos modelizar las características propias de sus especializaciones.

grado de una interrelación

Número de entidades que asocia la interrelación.

interrelación

Asociación entre entidades.

interrelación recursiva

Interrelación en la que alguna entidad está asociada más de una vez.

Bibliografía

Batini, C.; Ceri, S.; Navathe, S.B. (1992). *Conceptual Database Design: An Entity-Relationship Approach*. Reading, Massachusetts: Addison Wesley.

Teorey, T.J. (1999). *Database Modeling & Design. The Fundamental Principles* (3ª ed.). San Francisco: Morgan Kaufmann Publishers, Inc.