

## **Capítulo 06 Diseño de bases de datos y el modelo E-R**

El modelo de datos entidad-relación (E-R), que ofrece una manera de identificar las entidades que se van a representar en la base de datos y el modo en que se relacionan entre sí.

### **6.1 Visión general del proceso de diseño**

La tarea de creación de aplicaciones de bases de datos es una labor compleja, que implica varias fases, como el diseño del esquema de la base de datos, el diseño de los programas que tienen acceso a los datos y los actualizan y el diseño del esquema de seguridad para controlar el acceso a los datos.

#### **6.1.1 Fases del diseño**

Para aplicaciones pequeñas puede resultar factible para un diseñador de bases de datos que comprenda los requisitos de la aplicación decidir directamente sobre las relaciones que hay que crear, sus atributos y las restricciones sobre las relaciones. Frecuentemente no existe una sola persona que comprenda todas las necesidades de datos de la aplicación. El diseñador de la base de datos debe interactuar con los usuarios para comprender las necesidades de la aplicación, realizar una representación de alto nivel de esas necesidades que pueda ser comprendida por los usuarios y luego traducir esos requisitos a niveles inferiores del diseño. Los modelos de datos de alto nivel sirven a los diseñadores de bases de datos ofreciéndoles un marco conceptual en el que especificar de forma sistemática los requisitos de datos de los usuarios de la base de datos y una estructura para la base de datos que satisfaga esos requisitos.

- La fase inicial del diseño de las bases de datos es la caracterización completa de las necesidades de datos de los posibles usuarios de la base de datos. El diseñador de la base de datos debe interactuar intensamente con los expertos y los usuarios del dominio para realizar esta tarea. El resultado de esta fase es una especificación de requisitos del usuario.

- El diseñador elige el modelo de datos y, aplicando los conceptos del modelo de datos elegido, traduce estos requisitos en un esquema conceptual de la base de datos. El esquema desarrollado en esta fase de diseño conceptual proporciona una visión detallada de la empresa. Se suele emplear el modelo entidad-relación. En términos del modelo entidad-relación, el esquema conceptual especifica las entidades que se representan en la base de datos, sus atributos, las relaciones entre ellas y las restricciones que las afectan. Generalmente, la fase de diseño conceptual da lugar a la creación de un diagrama entidad-relación que ofrece una representación gráfica del esquema.

El diseñador revisa el esquema para confirmar que realmente se satisfacen todos los requisitos y que no entran en conflicto entre sí. También puede examinar el diseño para eliminar características redundantes.

- Un esquema conceptual completamente desarrollado indica también los requisitos funcionales de la empresa. En la especificación de requisitos funcionales los usuarios describen los tipos de operaciones (o transacciones) que se llevarán a cabo sobre los datos.

- El proceso de paso desde el modelo abstracto de datos a la implementación de la base de datos se divide en de dos fases de diseño finales:

1. En la fase de diseño lógico el diseñador traduce el esquema conceptual de alto nivel al

modelo de datos de la implementación del sistema de bases de datos que se va a usar. El modelo de implementación de los datos suele ser el modelo relacional, y este paso suele consistir en la traducción del esquema conceptual definido mediante el modelo entidad-relación en un esquema de relación.

2. Finalmente, el diseñador usa el esquema de base de datos resultante propio del sistema en la siguiente fase de diseño físico, en la que se especifican las características físicas de la base de datos. Entre estas características están la forma de organización de los archivos y las estructuras de almacenamiento interno.

El esquema físico de la base de datos puede modificarse con relativa facilidad una vez creada la aplicación. Es importante llevar a cabo con cuidado la fase de diseño de la base de datos antes de crear el resto de la aplicación de bases de datos.

### **6.1.2 Alternativas de diseño**

Una parte importante del proceso de diseño de las bases de datos consiste en decidir la manera de representar en el diseño los diferentes tipos de “cosas”, como personas, lugares, productos y similares. Se usa el término entidad para hacer referencia a cualquiera de esos elementos claramente identificables. Las diferentes entidades tienen algunos aspectos en común y algunas diferencias. Se desea aprovechar los aspectos en común para tener un diseño conciso, fácilmente comprensible, pero hay que conservar la suficiente flexibilidad como para representar las diferencias entre las diferentes entidades existentes en el momento del diseño o que se puedan materializar en el futuro. Las diferentes entidades se relacionan entre sí de diversas maneras, todas las cuales deben incluirse en el diseño de la base de datos. Al diseñar el esquema de una base de datos hay que asegurarse de que se evitan dos peligros importantes:

1. Redundancia. Un mal diseño puede repetir información. Lo ideal sería que la información apareciera exactamente en un solo lugar.

2. Incompletitud. Un mal diseño puede hacer que determinados aspectos de la empresa resulten difíciles o imposibles de modelar, en ocasiones lo que se podría intentar salir del paso con este diseño problemático almacenando valores nulos para la información, este parche no sólo resulta poco atractivo, sino que puede evitarse mediante restricciones de clave primaria.

Evitar los malos diseños no es suficiente. Puede haber gran número de buenos diseños entre los que haya que escoger. Como ejemplo sencillo, considérese un cliente que compra un producto. ¿La venta de este producto es una relación entre el cliente y el producto? Dicho de otra manera, ¿es la propia venta una entidad que está relacionada con el cliente y con el producto? Esta elección, aunque simple, puede suponer una importante diferencia en cuanto a los aspectos de la empresa que se pueden modelar bien. Considerando la necesidad de tomar decisiones como ésta para el gran número de entidades y de relaciones que hay en las empresas reales, no es difícil ver que el diseño de bases de datos puede constituir un problema arduo. En realidad, se verá que exige una combinación de conocimientos y de “buen gusto”.

## **6.2 El modelo entidad-relación**

El modelo de datos entidad-relación (E-R) se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica global de la base de datos. El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo radica en la

representación del significado de los datos. El modelo de datos E-R emplea tres conceptos básicos: los conjuntos de entidades, los conjuntos de relaciones y los atributos.

### 6.2.1 Conjuntos de entidades

Una entidad es una “cosa” u “objeto” del mundo real que es distinguible de todos los demás objetos. Por ejemplo, cada persona de una empresa es una entidad. Una entidad tiene un conjunto de propiedades, y los valores de algún conjunto de propiedades pueden identificar cada entidad de forma unívoca. Las entidades pueden ser concretas, como las personas o los libros, o abstractas, como los préstamos, las vacaciones o los conceptos. Un conjunto de entidades es un conjunto de entidades del mismo tipo que comparten las mismas propiedades, o atributos. El conjunto de todas las personas que son clientes en un banco dado, por ejemplo, se puede definir como el conjunto de entidades cliente. Análogamente, el conjunto de entidades préstamo puede representar el conjunto de todos los préstamos concedidos por un banco concreto. Cada una de las entidades que constituyen un conjunto se denomina extensión de ese conjunto de entidades. Por tanto, todos los clientes de un banco son la extensión del conjunto de entidades cliente. Los conjuntos de entidades no son necesariamente disjuntos.

Una entidad persona puede ser una entidad empleado, una entidad cliente, ambas cosas, o ninguna. Cada entidad se representa mediante un conjunto de atributos. Los atributos son propiedades descriptivas que posee cada miembro de un conjunto de entidades. La designación de un atributo para un conjunto de entidades expresa que la base de datos almacena información parecida relativa a cada entidad del conjunto de entidades; sin embargo, cada entidad puede tener su propio valor para cada atributo. Cada entidad tiene un valor para cada uno de sus atributos. En Estados Unidos, muchas empresas consideran adecuado usar el número seguridad\_social de cada persona<sup>1</sup> como atributo cuyo valor identifica unívocamente a esa persona. En general la empresa tendría que crear y asignar un identificador unívoco a cada cliente. Por tanto, las bases de datos incluyen una serie de conjuntos de entidades, cada una de las cuales contiene cierto número de entidades del mismo tipo.

### 6.2.2 Conjuntos de relaciones

Una relación es una asociación entre varias entidades. Un conjunto de relaciones es un conjunto de relaciones del mismo tipo. Formalmente es una relación matemática con  $n \geq 2$  de conjuntos de entidades (posiblemente no distintos). Si  $E_1, E_2, \dots, E_n$  son conjuntos de entidades, entonces un conjunto de relaciones  $R$  es un subconjunto de:  $\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$  donde  $(e_1, e_2, \dots, e_n)$  es una relación. La asociación entre conjuntos de entidades se conoce como participación; es decir, los conjuntos de entidades  $E_1, E_2, \dots, E_n$  participan en el conjunto de relaciones  $R$ . Un ejemplar de la relación de un esquema E-R representa una asociación entre las entidades citadas en la empresa real que se está modelando. La función que desempeña una entidad en una relación se denomina rol de esa entidad. Como los conjuntos de entidades que participan en un conjunto de relaciones, generalmente, son distintos, los roles están implícitos y no se suelen especificar. Sin embargo, resultan útiles cuando el significado de una relación necesita aclaración. Tal es el caso cuando los conjuntos de entidades de una relación no son distintos; es decir, el mismo conjunto de entidades participa en un conjunto de relaciones más de una vez, con diferentes roles. En este tipo de conjunto de relaciones, que se denomina a veces conjunto de relaciones recursivo, son necesarios los nombres explícitos para los roles para especificar la manera en que cada entidad participa en cada ejemplar de la relación. Por ejemplo, considérese un conjunto de entidades empleado que almacena información sobre todos los empleados del banco. Una relación puede también tener atributos denominados atributos descriptivos. Considérese el

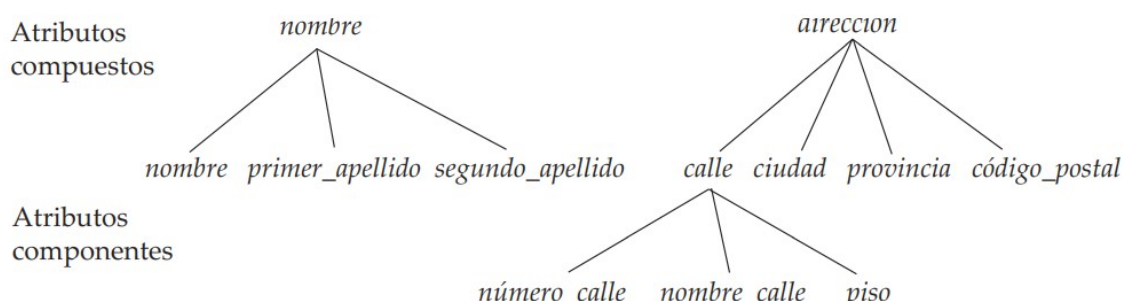
conjunto de relaciones impositor con los conjuntos de entidades cliente y cuenta. Se puede asociar el atributo *fecha\_acceso* con esta relación para especificar la fecha más reciente de acceso del cliente a la cuenta.

Como ejemplo adicional de los atributos descriptivos de las relaciones, supóngase que se tienen los conjuntos de entidades estudiante y asignatura que participan en el conjunto de relaciones matriculado. Puede que se desee almacenar el atributo descriptivo créditos con la relación, para registrar si el estudiante se ha matriculado en la asignatura para obtener créditos o sólo como oyente. Cada ejemplar de una relación de un conjunto de relaciones determinado debe identificarse unívocamente a partir de sus entidades participantes, sin usar los atributos descriptivos.

### 6.2.3 Atributos

Para cada atributo hay un conjunto de valores permitidos, denominados dominio o conjunto de valores de ese atributo. El dominio del atributo *nombre\_cliente* puede ser el conjunto de todas las cadenas de texto de una cierta longitud. Análogamente, el dominio del atributo *número\_préstamo* puede ser el conjunto de todas las cadenas de caracteres de la forma "P-n", donde n es un entero positivo. Formalmente, cada atributo de un conjunto de entidades es una función que asigna el conjunto de entidades a un dominio. Dado que el conjunto de entidades puede tener varios atributos, cada entidad se puede describir mediante un conjunto de pares (atributo,valor), un par por cada atributo del conjunto de entidades. Cada atributo, tal y como se usa en el modelo E-R, se puede caracterizar por los siguientes tipos de atributo:

- **Atributos simples y compuestos:** En los ejemplos considerados hasta ahora los atributos han sido simples; es decir, no estaban divididos en subpartes. Los atributos compuestos, en cambio, se pueden dividir en subpartes (es decir, en otros atributos). Por ejemplo, el atributo *nombre* puede estar estructurado como un atributo compuesto consistente en *nombre*, *primer\_apellido* y *segundo\_apellido*. Usar atributos compuestos en un esquema de diseño es una buena elección si el usuario desea referirse a un atributo completo en algunas ocasiones y, en otras, solamente a algún componente del atributo. Obsérvese también que los atributos compuestos pueden aparecer como una jerarquía. En el atributo compuesto *dirección*, el componente *calle* puede dividirse a su vez en *número\_calle*, *nombre\_calle* y *piso*.



**Figura 6.4** Atributos compuestos *nombre* y *dirección*.

- **Atributos monovalorados y multivalorados:** Todos los atributos que se han especificado en los ejemplos anteriores tienen un único valor para cada entidad concreta. Se dice que estos atributos son monovalorados. Puede haber ocasiones en las que un atributo tenga un conjunto de valores para una entidad concreta. Considérese un conjunto de entidades empleado con el atributo *número\_teléfono*.

Cada empleado puede tener cero, uno o varios números de teléfono, y empleados diferentes pueden tener diferente cantidad de teléfonos. Se dice que este tipo de atributo es multivalorado. Como ejemplo adicional, el atributo nombre \_subordinado del conjunto de entidades empleado es multivalorado, ya que cada empleado podría tener cero, uno o más subordinados. Si resulta necesario, se pueden establecer apropiadamente límites inferior y superior al número de valores en el atributo multivalorado.

- **Atributos derivados:** El valor de este tipo de atributo se puede obtener a partir del valor de otros atributos o entidades relacionados. Por ejemplo, supóngase que el conjunto de entidades cliente que tiene un atributo préstamos que representa el número de préstamos que cada cliente tiene concedidos en el banco. Ese atributo se puede obtener contando el número de entidades préstamo asociadas con cada cliente. El valor de los atributos derivados no se almacena, sino que se calcula cada vez que hace falta.

Los atributos toman valores nulos cuando las entidades no tienen ningún valor para ese atributo. El valor nulo también puede indicar “no aplicable”—es decir, que el valor no existe para esa entidad. Por ejemplo, una persona puede no tener un segundo nombre de pila. Nulo puede también designar que el valor del atributo es desconocido. Un valor desconocido puede ser falta (el valor existe pero no se tiene esa información) o desconocido (no se sabe si ese valor existe realmente o no). Un valor nulo para el atributo piso puede significar que la dirección no incluye un piso (no aplicable), que existe el valor piso pero no se conoce cuál es (falta), o que no se sabe si el valor piso forma parte o no de la dirección del cliente (desconocido).

### 6.3 Restricciones

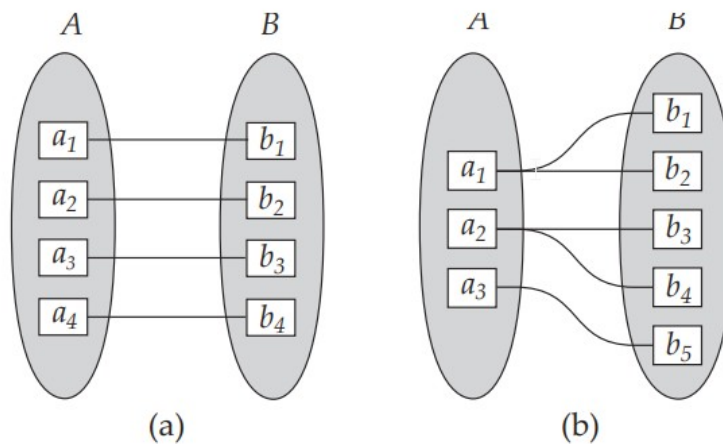
Un esquema de desarrollo E-R puede definir ciertas restricciones a las que el contenido de la base de datos se debe adaptar. La correspondencia de cardinalidades, las restricciones de claves y las restricciones de participación.

#### 6.3.1 Correspondencia de cardinalidades

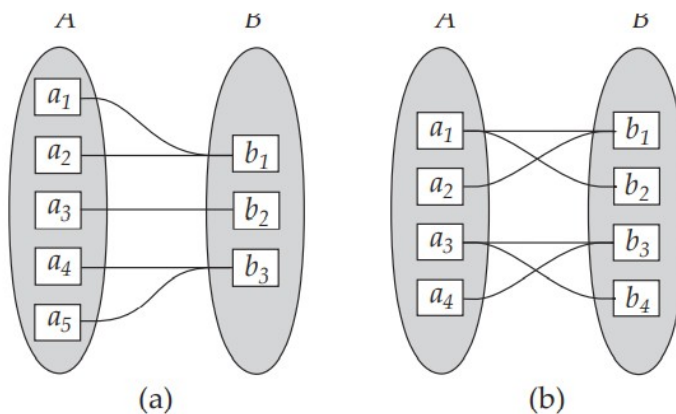
La correspondencia de cardinalidades, o razón de cardinalidad, expresa el número de entidades a las que otra entidad se puede asociar mediante un conjunto de relaciones. La correspondencia de cardinalidades resulta muy útil para describir conjuntos de relaciones binarias, aunque pueda contribuir a la descripción de conjuntos de relaciones que impliquen más de dos conjuntos de entidades. En este apartado se centrará la atención únicamente en los conjuntos de relaciones binarias. Para un conjunto de relaciones binarias R entre los conjuntos de entidades A y B, la correspondencia de cardinalidades debe ser una de las siguientes:

1. **Uno a uno** Cada entidad de A se asocia, a lo sumo, con una entidad de B, y cada entidad en B se asocia, a lo sumo, con una entidad de A.
2. **Uno a varios** Cada entidad de A se asocia con cualquier número (cero o más) de entidades de B. Cada entidad de B, sin embargo, se puede asociar, a lo sumo, con una entidad de A.
3. **Varios a uno** Cada entidad de A se asocia, a lo sumo, con una entidad de B. Cada entidad de B, sin embargo, se puede asociar con cualquier número (cero o más) de entidades de A.

4. **Varios a varios** Cada entidad de A se asocia con cualquier número (cero o más) de entidades de B, y cada entidad de B se asocia con cualquier número (cero o más) de entidades de A.



Correspondencia de cardinalidades. (a) Uno a uno. (b) Uno a varios.



Correspondencia de cardinalidades. (a) Varios a uno. (b) Varios a varios.

### 6.3.2 Claves

Es necesario tener una forma de especificar la manera de distinguir las entidades pertenecientes a un conjunto de entidades dado. Conceptualmente cada entidad es distinta; desde el punto de vista de las bases de datos, sin embargo, la diferencia entre ellas se debe expresar en términos de sus atributos. Por lo tanto, los valores de los atributos de cada entidad deben ser tales que permitan identificar unívocamente a esa entidad. En otras palabras, no se permite que ningún par de entidades de un conjunto de entidades tenga exactamente el mismo valor en todos sus atributos. Las claves permiten identificar un conjunto de atributos que resulta suficiente para distinguir las entidades entre sí. Las claves también ayudan a identificar unívocamente las relaciones y, por tanto, a distinguir las relaciones entre sí.

#### 6.3.2.1 Conjuntos de entidades

Una superclave es un conjunto de uno o más atributos que, tomados conjuntamente, permiten identificar de forma unívoca una entidad del conjunto de entidades. Por ejemplo, el atributo `id_cliente` del conjunto de entidades cliente es suficiente para distinguir una entidad cliente de las demás. Por tanto, `id_cliente` es una superclave. Análogamente, la combinación de `nombre_cliente` e `id_cliente` es una superclave del conjunto de entidades cliente. El atributo `nombre_cliente` de cliente no es una superclave, ya que varias

personas pueden tener el mismo nombre. A menudo interesan las superclaves para las que ningún subconjunto propio es superclave. Esas superclaves mínimas se denominan **claves candidatas**.

Es posible que conjuntos distintos de atributos puedan servir como clave candidata. Supóngase que una combinación de nombre\_cliente y calle\_cliente sea suficiente para distinguir entre los miembros del conjunto de entidades cliente. Entonces, tanto id\_cliente como nombre\_cliente, calle\_cliente son claves candidatas. Aunque los atributos id\_cliente y nombre\_cliente juntos puedan diferenciar las entidades cliente, su combinación no forma una clave candidata, ya que el atributo id\_cliente por sí solo es una clave candidata. Se usa el término clave primaria para denotar la clave candidata elegida por el diseñador de la base de datos como elemento principal de identificación de las entidades pertenecientes a un conjunto de entidades. Las claves (primarias, candidatas y superclaves) son propiedades del conjunto de entidades, más que de cada una de las entidades. Dos entidades cualesquiera del conjunto no pueden tener el mismo valor de los atributos de su clave al mismo tiempo. La designación de una clave representa una restricción de la empresa real que se está modelando. La clave primaria se debe escoger de manera que sus atributos nunca, o muy raramente, cambien. Los identificadores únicos generados por las empresas generalmente no cambian, excepto si se fusionan dos empresas; en tal caso, las dos empresas pueden haber emitido el mismo identificador y será necesario reasignar los identificadores para asegurarse de que sean únicos.

### 6.3.2.2 Conjuntos de relaciones

La clave primaria de cada conjunto de entidades permite distinguir entre las diferentes entidades del conjunto. Se necesita un mecanismo parecido para distinguir entre las diferentes relaciones de cada conjunto de relaciones. Sea  $R$  un conjunto de relaciones que involucra los conjuntos de entidades  $E_1, E_2, \dots, E_n$ . Sea clave\_primaria( $E_i$ ) el conjunto de atributos que forman la clave primaria del conjunto de entidades  $E_i$ . Por ahora se dará por supuesto que los nombres de los atributos de todas las claves primarias son únicos y que cada conjunto de entidades participa sólo una vez en la relación. La composición de la clave primaria de un conjunto de relaciones depende del conjunto de atributos asociado con el conjunto de relaciones  $R$ . Si el conjunto de relaciones  $R$  no tiene atributos asociados, entonces el conjunto de atributos **clave\_primaria( $E_1$ )  $\cup$  clave\_primaria( $E_2$ )  $\cup \dots \cup$  clave\_primaria( $E_n$ )** describe una relación concreta del conjunto  $R$ . Si el conjunto de relaciones  $R$  tiene asociados los atributos  $a_1, a_2, \dots, a_m$ , entonces el conjunto de atributos **clave\_primaria( $E_1$ )  $\cup$  clave\_primaria( $E_2$ )  $\cup \dots \cup$  clave\_primaria( $E_n$ )  $\cup \{a_1, a_2, \dots, a_m\}$**  describe una relación concreta del conjunto  $R$ . En ambos casos, el conjunto de atributos **clave\_primaria( $E_1$ )  $\cup$  clave\_primaria( $E_2$ )  $\cup \dots \cup$  clave\_primaria( $E_n$ )** forma una superclave del conjunto de relaciones. En el caso de que los nombres de los atributos de las claves primarias no sean únicos en todos los conjuntos de entidades, hay que renombrar los atributos para distinguirlos; el nombre del conjunto de entidades combinado con el del atributo formará un nombre único. En el caso de que un conjunto de entidades participe más de una vez en el conjunto de relaciones, se usa el nombre del rol en lugar del nombre del conjunto de entidades para formar un nombre de atributo único. Para las relaciones no binarias, si no hay restricciones de cardinalidad, la superclave formada como se ha descrito anteriormente en este apartado es la única clave candidata, y se elige como clave primaria. La elección de la clave primaria resulta más complicada si hay restricciones de cardinalidad. Dado que no se ha estudiado la manera de especificar las restricciones de cardinalidad en relaciones no binarias, no se estudiará más este aspecto en este capítulo.

### 6.3.3 Restricciones de participación

Se dice que la participación de un conjunto de entidades  $E$  en un conjunto de relaciones  $R$

es total si cada entidad de E participa, al menos, en una relación de R. Si sólo algunas entidades de E participan en relaciones de R, se dice que la participación del conjunto de entidades E en la relación R es parcial.

#### **6.4 Diagramas entidad-relación**

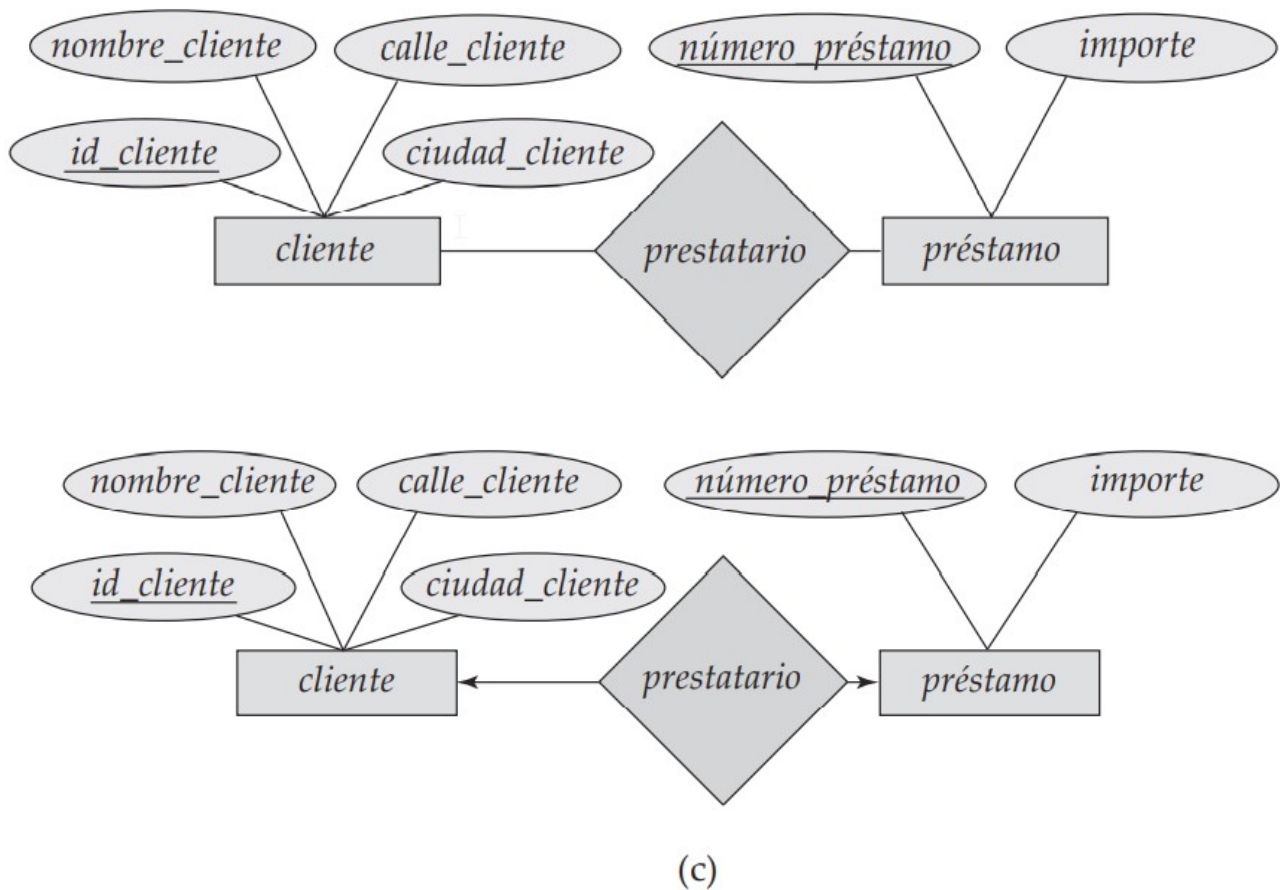
Los diagramas E-R pueden expresar gráficamente la estructura lógica general de las bases de datos. Los diagramas E-R son sencillos y claros—cualidades que pueden ser responsables en gran parte de la popularidad del modelo E-R. Estos diagramas constan de los siguientes componentes principales:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombos, que representan conjuntos de relaciones.
- Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con los conjuntos de relaciones.
- Elipses dobles, que representan atributos multivalorados.
- Elipses discontinuas, que denotan atributos derivados.
- Líneas dobles, que indican participación total de una entidad en un conjunto de relaciones.
- Rectángulos dobles, que representan conjuntos de entidades débiles.

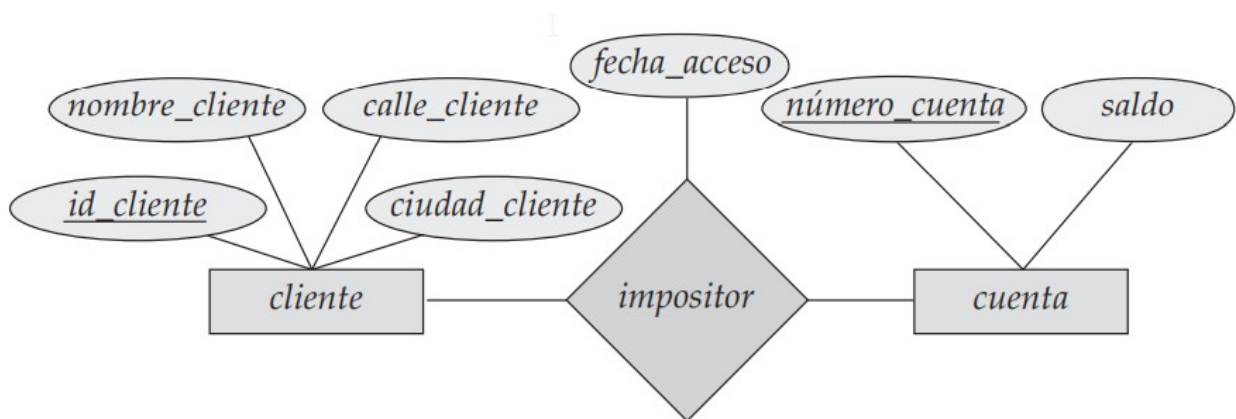
Una línea dirigida desde el conjunto de relaciones prestatario al conjunto de entidades préstamo especifica que prestatario es un conjunto de relaciones uno a uno o varios a uno desde cliente a préstamo; prestatario no puede ser un conjunto de relaciones varios a varios ni uno a varios desde cliente a préstamo.

Una línea no dirigida desde el conjunto de relaciones prestatario al conjunto de relaciones préstamo especifica que prestatario es un conjunto de relaciones varios a varios o uno a varios desde cliente a préstamo.





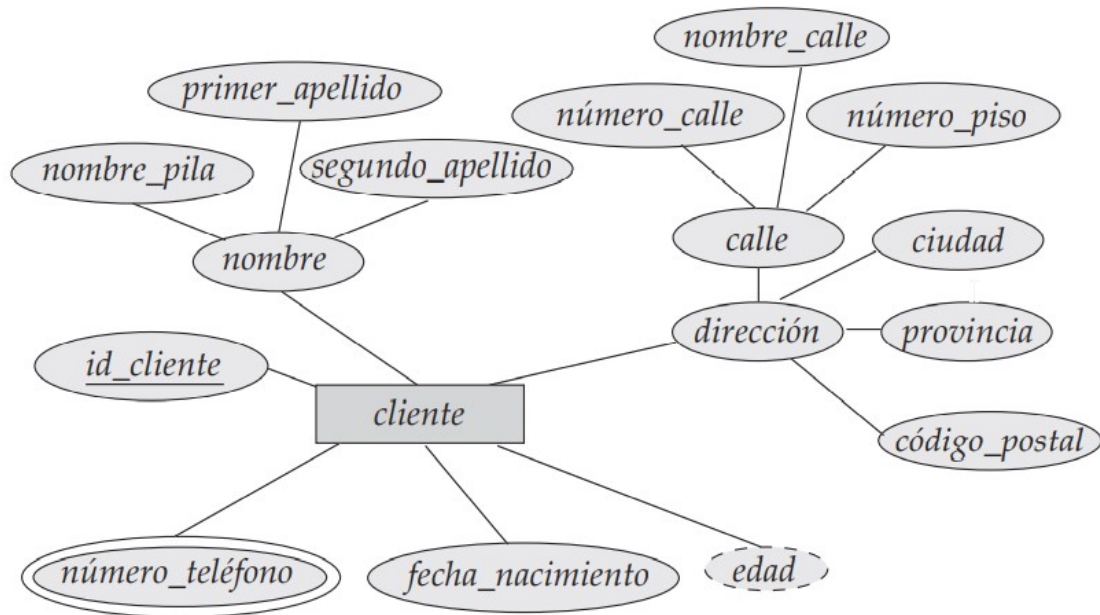
Si un conjunto de relaciones también tiene asociados algunos atributos, entonces esos atributos se unen con el conjunto de relaciones. se tiene el atributo descriptivo fecha\_acceso unido al conjunto de relaciones impositor para especificar la fecha del último acceso del cliente a esa cuenta.



**Figura 6.9** Diagrama E-R con un atributo unido a un conjunto de relaciones.

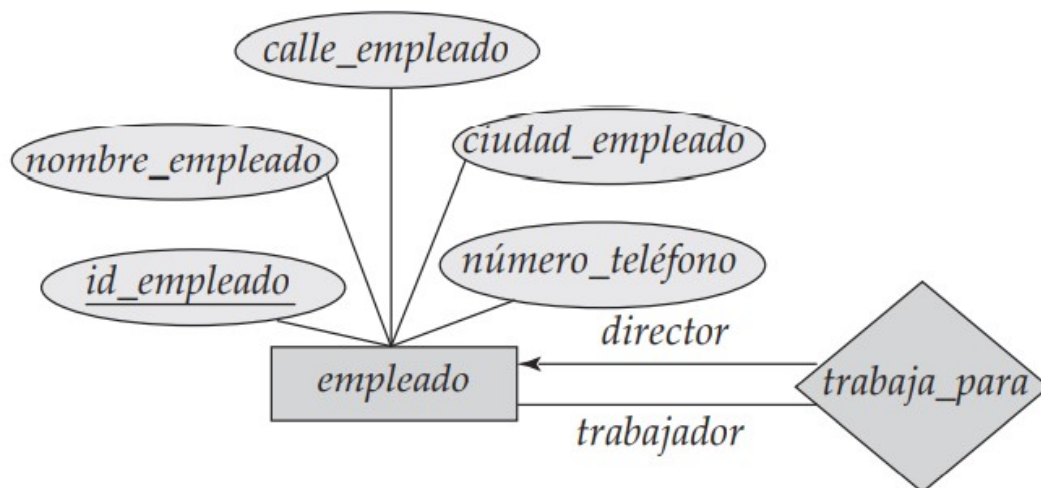
Se muestra cómo se pueden representar los atributos compuestos en la notación E-R. En este caso, el atributo compuesto nombre, con los atributos componentes nombre\_pila, primer\_apellido y segundo\_apellido sustituye al atributo simple nombre\_cliente de cliente. Además, el atributo compuesto dirección, cuyos atributos componentes son calle, ciudad, provincia y código\_postal, sustituye a los atributos calle\_cliente y ciudad\_cliente de cliente. El atributo calle es por sí mismo un atributo compuesto cuyos atributos

componentes son número\_calle, nombre\_calle y número\_piso, también muestra un atributo multivalorado, número\_téfono, indicado por una elipse doble, y un atributo derivado, edad, indicado por una elipse discontinua. En los diagramas E-R los roles se indican mediante etiquetas en las líneas que unen los rombos con los rectángulos.



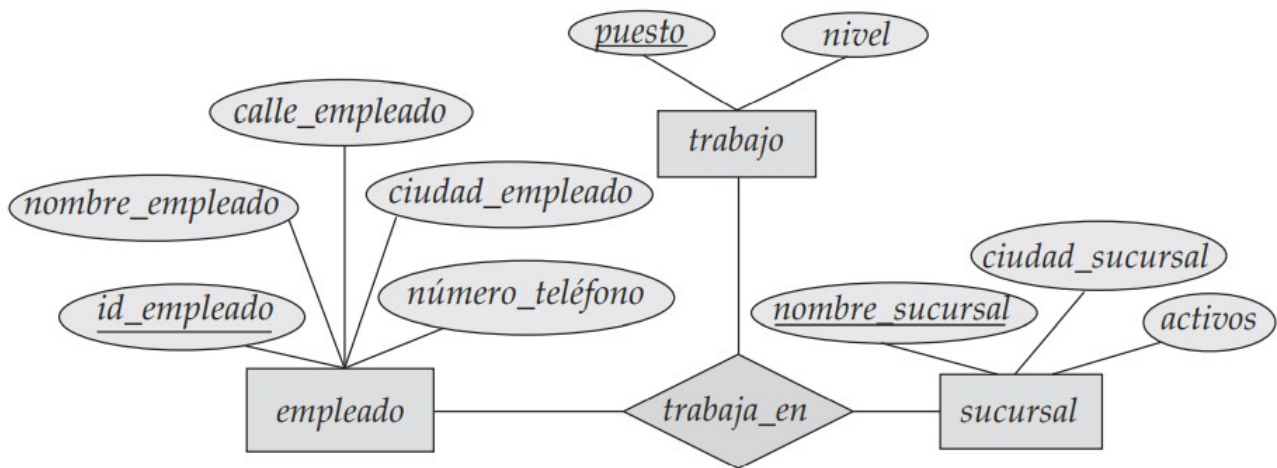
**Figura 6.10** Diagrama E-R con atributos compuestos, multivalorados y derivados.

Se muestra los indicadores de roles director y trabajador entre el conjunto de entidades empleado y el conjunto de relaciones trabaja\_para. Los conjuntos de relaciones no binarias se pueden especificar fácilmente en los diagramas E-R.



**Figura 6.11** Diagrama E-R con indicadores de roles.

Se muestran tres conjuntos de entidades empleado, trabajo y sucursal, relacionados mediante el conjunto de relaciones trabaja\_en.

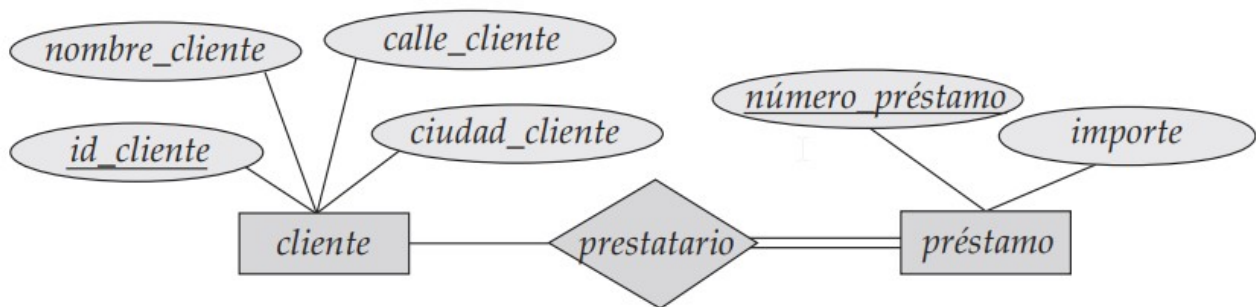


**Figura 6.12** Diagrama E-R con una relación ternaria.

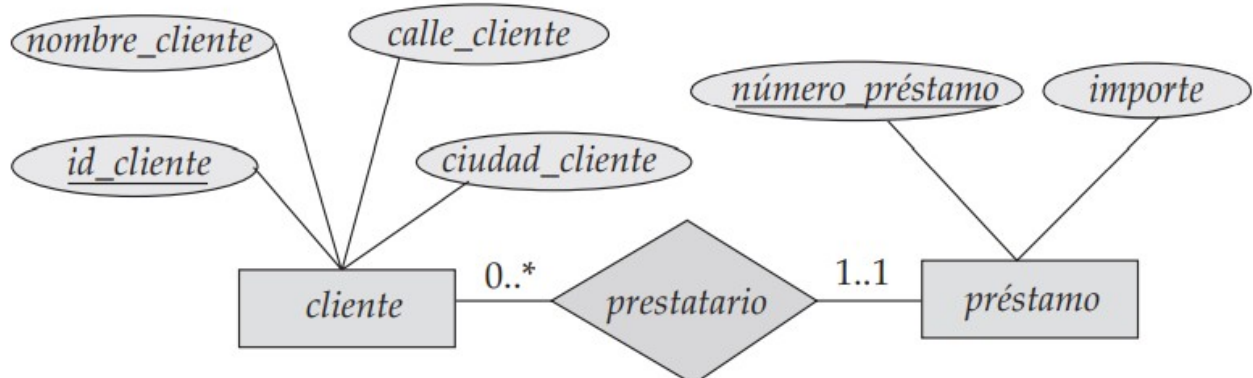
En el caso de conjuntos de relaciones no binarias se pueden especificar algunos tipos de relaciones varios a uno. Como máximo se permite una flecha desde cada conjunto de relaciones, ya que los diagramas E-R con dos o más flechas salientes de cada conjunto de relaciones no binarias se pueden interpretar de dos formas. Supónganse que hay un conjunto de relaciones  $R$  entre los conjuntos de entidades  $A_1, A_2, \dots, A_n$  y que las únicas flechas están en los arcos de los conjuntos de entidades  $A_{i+1}, A_{i+2}, \dots, A_n$ . Entonces, las dos interpretaciones posibles son:

- Una combinación concreta de entidades de  $A_1, A_2, \dots, A_i$  se puede asociar, a lo sumo, con una combinación de entidades de  $A_{i+1}, A_{i+2}, \dots, A_n$ . Por tanto, la clave primaria de la relación  $R$  se puede crear mediante la unión de las claves primarias de  $A_1, A_2, \dots, A_i$ .
- Para cada conjunto de entidades  $A_k$ ,  $i < k \leq n$ , cada combinación de las entidades de los otros conjuntos de entidades se puede asociar, a lo sumo, con una entidad de  $A_k$ . Cada conjunto  $\{A_1, A_2, \dots, A_{k-1}, A_{k+1}, \dots, A_n\}$ , para  $i < k \leq n$ , forma, entonces, una clave candidata.

Para evitar confusiones, sólo se permite una flecha saliente de cada conjunto de relaciones, en cuyo caso las dos interpretaciones son equivalentes. En los diagramas E-R se usan líneas dobles para indicar que la participación de un conjunto de entidades en un conjunto de relaciones es total; es decir, cada entidad del conjunto de entidades aparece, al menos, en una relación de ese conjunto de relaciones, considérese la relación prestatario entre clientes y préstamos. Una línea doble de préstamo a prestatario, indica que cada préstamo debe tener, al menos, un cliente asociado. Los diagramas E-R también ofrecen una manera de indicar restricciones más complejas sobre el número de veces que cada entidad participa en las relaciones de un conjunto de relaciones. Un segmento entre un conjunto de entidades y un conjunto de relaciones binarias puede tener unas cardinalidades mínima y máxima asociadas, que se muestran de la forma min..max, donde min es la cardinalidad mínima y max es la máxima. Un valor mínimo de 1 indica una participación total del conjunto de entidades en el conjunto de relaciones. Un valor máximo de 1 indica que la entidad participa, a lo sumo, en una relación, mientras que un valor máximo de \* indica que no hay límite. Téngase en cuenta que una etiqueta 1..\* en un segmento es equivalente a una línea doble.



El segmento entre préstamo y prestatario tiene una restricción de cardinalidad de 1..1, que significa que tanto la cardinalidad mínima como la máxima son 1. Es decir, cada préstamo debe tener exactamente un cliente asociado. El límite 0..\* en el segmento de cliente a prestatario indica que cada cliente puede no tener ningún préstamo o tener varios. Por tanto, la relación prestatario es uno a varios de cliente a préstamo y, además, la participación de préstamo en prestatario es total. Es fácil malinterpretar 0..\* en el segmento entre cliente y prestatario y pensar que la relación prestatario es varios a uno de cliente a préstamo—esto es exactamente lo contrario a la interpretación correcta. Si los dos segmentos de una relación binaria tienen un valor máximo de 1, la relación es uno a uno. Si se hubiese especificado un límite de cardinalidad de 1..\* en el segmento entre cliente y prestatario, se estaría diciendo que cada cliente debe tener, al menos, un préstamo.



## 6.5 Aspectos del diseño entidad-relación

Los conceptos de conjunto de entidades y de conjunto de relaciones no son precisos, y es posible definir el conjunto de entidades y las relaciones entre ellas de diferentes formas.

### 6.5.1 Uso de conjuntos de entidades y de atributos

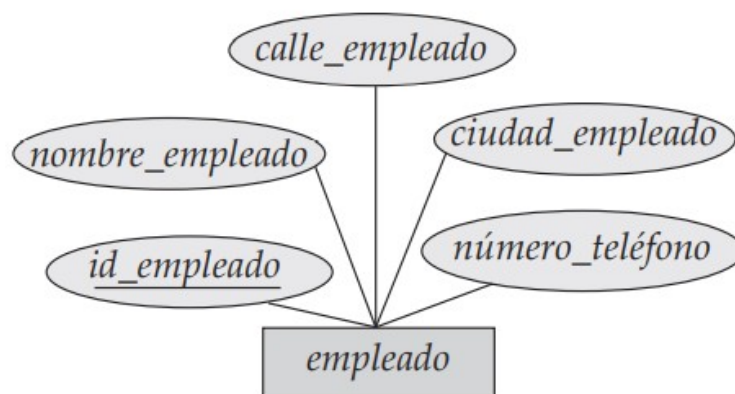
Considérese el conjunto de entidades empleado con los atributos nombre\_empleado y número\_teléfono. Se puede argumentar que un teléfono es una entidad en sí misma con los atributos número\_teléfono y ubicación; la ubicación puede ser la sucursal o el domicilio donde el teléfono está instalado, y se pueden representar los teléfonos móviles (celulares) mediante el valor “móvil”. Si se adopta este punto de vista, hay que redefinir el conjunto de entidades empleado como:

- El conjunto de entidades empleado con los atributos id\_empleado y nombre\_empleado.
- El conjunto de entidades teléfono con los atributos número\_teléfono y ubicación.

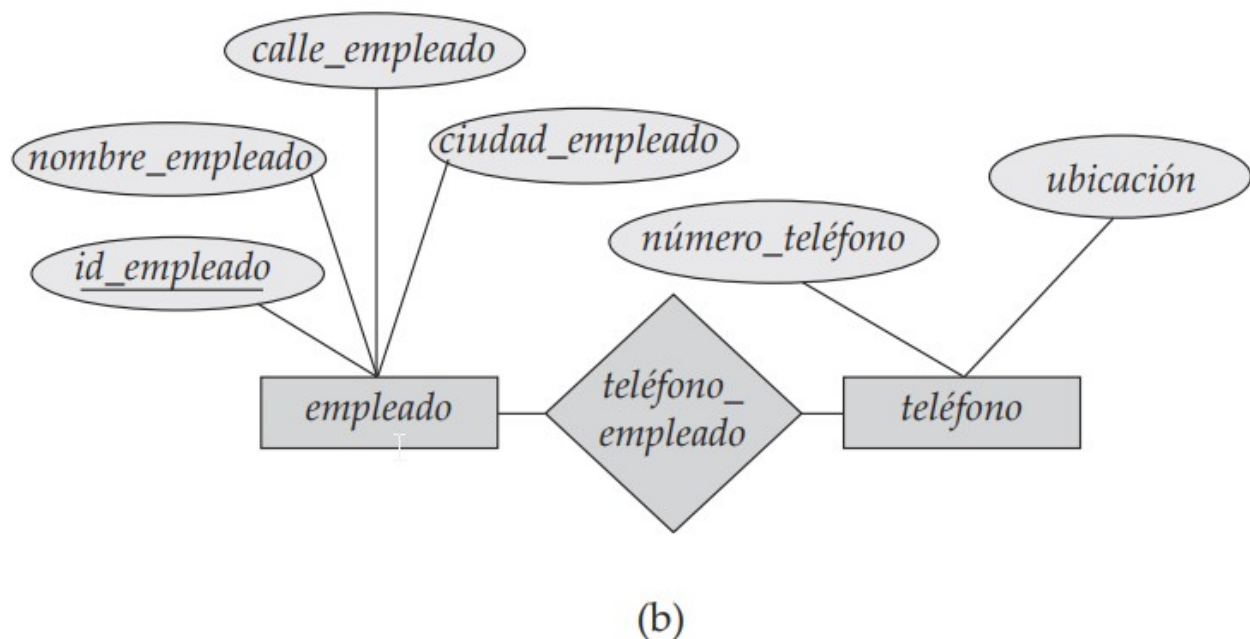


- La relación teléfono\_Empleado, que denota la asociación entre los empleados y sus teléfonos.

¿Cuál es, entonces, la diferencia principal entre estas dos definiciones de empleado? Tratar el teléfono como el atributo número\_teléfono implica que cada empleado tiene exactamente un número de teléfono. Tratar el teléfono como la entidad teléfono permite que los empleados tengan varios números de teléfonos (o ninguno) asociados. Sin embargo, se puede definir en su lugar número\_teléfono como atributo multivalorado para permitir varios teléfonos por empleado. La diferencia principal es que tratar el teléfono como entidad modela mejor una situación en la que se puede querer guardar información extra sobre el teléfono, como su ubicación, su tipo (móvil, videoteléfono o fijo) o las personas que lo comparten. Por tanto, tratar el teléfono como entidad es más general que tratarlo como atributo y resulta adecuado cuando la generalidad pueda ser útil. En cambio, no sería apropiado tratar el atributo nombre\_empleado como entidad; es difícil argumentar que nombre\_empleado sea una entidad en sí misma (a diferencia de lo que ocurre con teléfono). Así pues, resulta adecuado tener nombre\_empleado como atributo del conjunto de entidades empleado. Por tanto, se suscitan naturalmente dos preguntas: ¿qué constituye un atributo? y ¿qué constituye un conjunto de entidades? Desafortunadamente no hay respuestas sencillas. Las distinciones dependen principalmente de la estructura de la empresa real que se esté modelando y de la semántica asociada con el atributo en cuestión. Un error común es usar la clave primaria de un conjunto de entidades como atributo de otro conjunto de entidades en lugar de usar una relación. Por ejemplo, es incorrecto modelar id\_cliente como atributo de préstamo, aunque cada préstamo tenga sólo un cliente. La relación prestatario es la forma correcta de representar la conexión entre los préstamos y los clientes, ya que hace explícita su conexión, en lugar de dejarla implícita mediante un atributo. Otro error relacionado con éste que se comete a veces es escoger los atributos de clave primaria de los conjuntos de entidades relacionados como atributos del conjunto de relaciones.



(a)



### 6.5.2 Uso de los conjuntos de entidades y de los conjuntos de relaciones

No siempre está claro si es mejor expresar un objeto mediante un conjunto de entidades o mediante un conjunto de relaciones. Una alternativa es modelar los préstamos no como entidades, sino como relaciones entre los clientes y las sucursales, con número\_préstamo e importe como atributos descriptivos. Cada préstamo se representa mediante una relación entre un cliente y una sucursal. Si cada préstamo se concede exactamente a un cliente y se asocia exactamente con una sucursal, se puede encontrar satisfactorio el diseño en el que cada préstamo se representa como una relación. Sin embargo, con este diseño, no se puede representar convenientemente una situación en la que un préstamo se conceda a varios clientes conjuntamente. Para tratar esta situación se debe definir otra relación para cada prestatario de ese préstamo conjunto. Entonces habrá que replicar los valores de los atributos descriptivos número\_préstamo e importe en cada una de esas relaciones. Cada una de esas relaciones debe, por supuesto, tener el mismo valor para los atributos descriptivos número\_préstamo e importe.

Se plantean dos problemas como resultado de esta réplica: (1) los datos se almacenan varias veces, lo que hace que se desperdicie espacio de almacenamiento, y (2) las actualizaciones pueden dejar los datos en un estado inconsistente, en el que los valores de atributos que se supone que tienen el mismo valor difieran. El problema de cómo evitar esta réplica se trata formalmente mediante la teoría de la normalización. Un criterio para determinar si se debe usar un conjunto de entidades o un conjunto de relaciones puede ser escoger un conjunto de relaciones para describir las acciones que ocurran entre entidades. Este enfoque también puede ser útil para decidir si ciertos atributos se pueden expresar mejor como relaciones.

Un criterio para determinar si se debe usar un conjunto de entidades o un conjunto de relaciones puede ser escoger un conjunto de relaciones para describir las acciones que ocurran entre entidades. Este enfoque también puede ser útil para decidir si ciertos atributos se pueden expresar mejor como relaciones.

### 6.5.3 Conjuntos de relaciones binarias y n-arias

Las relaciones en las bases de datos suelen ser binarias. Puede que algunas relaciones que no parecen ser binarias se puedan representar mejor mediante varias relaciones binarias.

De hecho, siempre es posible sustituir los conjuntos de relaciones no binarias (n-aria, para  $n > 2$ ) por varios conjuntos de relaciones binarias. Por simplificar, considérense el conjunto de relaciones abstracto ternario ( $n = 3$ )  $R$  y los conjuntos de entidades  $A$ ,  $B$ , y  $C$ . Se sustituye el conjunto de relaciones  $R$  por un conjunto de entidades  $E$  y se crean tres conjuntos de relaciones:

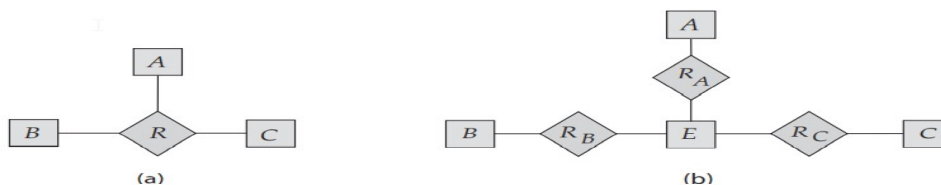
- $PA$ , que relaciona  $E$  y  $A$
- $PB$ , que relaciona  $E$  y  $B$
- $PC$ , que relaciona  $E$  y  $C$

Si el conjunto de relaciones  $R$  tiene atributos, éstos se asignan al conjunto de entidades  $E$ ; además, se crea un atributo de identificación especial para  $E$  (ya que se debe poder distinguir las diferentes entidades de cada conjunto de entidades con base en los valores de sus atributos). Para cada relación  $(a_i, b_i, c_i)$  del conjunto de relaciones  $R$  se crea una nueva entidad  $e_i$  del conjunto de entidades  $E$ . Luego, en cada uno de los tres nuevos conjuntos de relaciones, se inserta una relación del modo siguiente:

- $(e_i, a_i)$  en  $RA$
- $(e_i, b_i)$  en  $RB$
- $(e_i, c_i)$  en  $RC$

Este proceso se puede generalizar de forma directa a los conjuntos de relaciones n-arias. Por tanto, conceptualmente, se puede restringir el modelo E-R para que sólo incluya conjuntos de relaciones binarias. Sin embargo, esta restricción no siempre es deseable.

- Es posible que sea necesario crear un atributo de identificación para que el conjunto de entidades represente el conjunto de relaciones. Este atributo, junto con los conjuntos de relaciones adicionales necesarios, incrementa la complejidad del diseño y los requisitos globales de almacenamiento.
- Un conjunto de relaciones n-arias muestra más claramente que varias entidades participan en una sola relación.
- Puede que no haya forma de traducir las restricciones a la relación ternaria en restricciones a las relaciones binarias. Por ejemplo, considérese una restricción que dice que  $R$  es varios a uno de  $A$ ,  $B$  a  $C$ ; es decir, cada par de entidades de  $A$  y de  $B$  se asocia, a lo sumo, con una entidad de  $C$ . Esta restricción no se puede expresar mediante restricciones de cardinalidad sobre los conjuntos de relaciones  $RA$ ,  $RB$  y  $RC$ .



**Figura 6.17** Una relación ternaria y tres relaciones binarias.

#### 6.5.4 Ubicación de los atributos de las relaciones

La razón de cardinalidad de una relación puede afectar a la ubicación de sus atributos. Por tanto, los atributos de los conjuntos de relaciones uno a uno o uno a varios pueden

estar asociados con uno de los conjuntos de entidades participantes, en lugar de con el conjunto de relaciones. Por ejemplo, especifiquemos que impositor es un conjunto de relaciones uno a varios tal que cada cliente puede tener varias cuentas, pero cada cuenta sólo puede tener un cliente como titular. En este caso, el atributo `fecha_acceso`, que especifica la fecha en que el cliente tuvo acceso a la cuenta por última vez, podría estar asociado con el conjunto de entidades cuenta, se muestran algunos de los atributos de los dos conjuntos de entidades. Dado que cada entidad cuenta participa en una relación con un ejemplar de cliente, como máximo, hacer esta designación de atributos tendría el mismo significado que colocar `fecha_acceso` en el conjunto de relaciones impositor. Los atributos de un conjunto de relaciones uno a varios sólo se pueden recolocar en el conjunto de entidades de la parte “varios” de la relación. Por otra parte, para los conjuntos de entidades uno a uno, los atributos de la relación se pueden asociar con cualquiera de las entidades participantes. La decisión de diseño sobre la ubicación de los atributos descriptivos en estos casos—como atributo de la relación o de la entidad—debe reflejar las características de la empresa que se modela. El diseñador puede elegir mantener `fecha_acceso` como atributo de impositor para expresar explícitamente que se produce un acceso en el punto de interacción entre los conjuntos de entidades cliente y cuenta. La elección de la ubicación del atributo es más sencilla para los conjuntos de relaciones varios a varios.

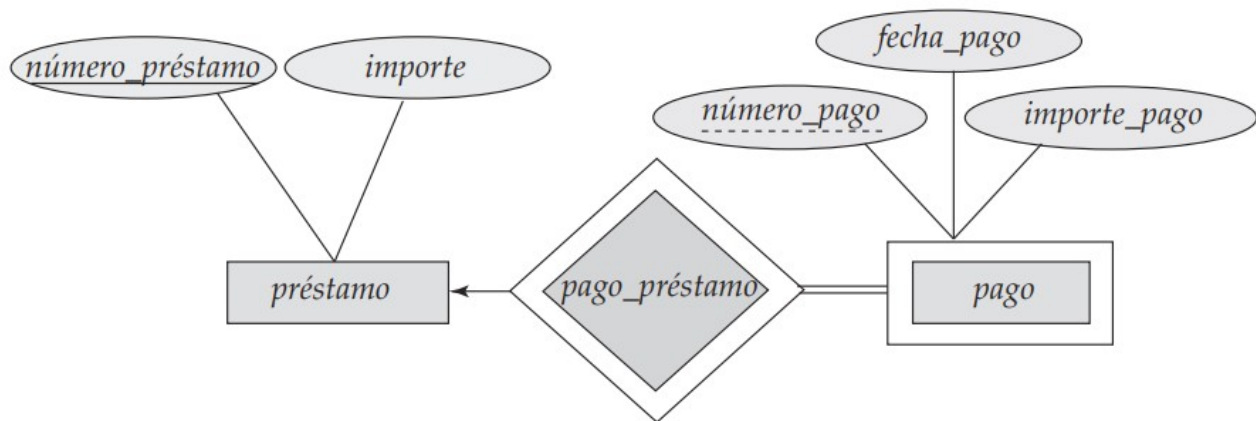
## **6.6 Conjuntos de entidades débiles**

Puede que un conjunto de entidades no tenga suficientes atributos para formar una clave primaria. Ese conjunto de entidades se denomina conjunto de entidades débiles. Los conjuntos de entidades que tienen una clave primaria se denominan conjuntos de entidades fuertes. Como ilustración, considérese el conjunto de entidades pago, que tiene tres atributos: `número_pago`, `fecha_pago` e `importe_pago`. Los números de pago suelen ser números secuenciales, a partir de 1, generados independientemente para cada préstamo. Por tanto, aunque cada entidad pago es distinta, los pagos de diferentes préstamos pueden compartir el mismo número de pago. Así, este conjunto de entidades no tiene clave primaria; es un conjunto de entidades débiles. Para que un conjunto de entidades débiles tenga sentido, debe estar asociado con otro conjunto de entidades, denominado conjunto de entidades identificadoras o propietarias. Cada entidad débil debe estar asociada con una entidad identificadora; es decir, se dice que el conjunto de entidades débiles depende existencialmente del conjunto de entidades identificadoras. Se dice que el conjunto de entidades identificadoras posee el conjunto de entidades débiles al que identifica. La relación que asocia el conjunto de entidades débiles con el conjunto de entidades identificadoras se denomina relación identificadora. La relación identificadora es varios a uno del conjunto de entidades débiles al conjunto de entidades identificadoras y la participación del conjunto de entidades débiles en la relación es total. En nuestro ejemplo, el conjunto de entidades identificador de pago es préstamo, y la relación `pago_préstamo` que asocia las entidades pago con sus correspondientes entidades préstamo es la relación identificadora.

Aunque los conjuntos de entidades débiles no tienen clave primaria, hace falta un medio para distinguir entre todas las entidades del conjunto de entidades débiles que dependen de una entidad fuerte concreta. El discriminante de un conjunto de entidades débiles es un conjunto de atributos que permite que se haga esta distinción. Por ejemplo, el discriminante del conjunto de entidades débiles pago es el atributo `número_pago` ya que, para cada préstamo, el número de pago identifica de forma única cada pago de ese préstamo. El discriminante del conjunto de entidades débiles se denomina clave parcial del conjunto de entidades. La clave primaria de un conjunto de entidades débiles se forma con la clave primaria del conjunto de entidades identificadoras y el discriminante del



conjunto de entidades débiles. En el caso del conjunto de entidades pago, su clave primaria es {número\_préstamo, número\_pago}, donde número\_préstamo es la clave primaria del conjunto de entidades identificadoras, es decir, préstamo, y número\_pago distingue las entidades pago de un mismo préstamo. El conjunto de entidades identificadoras no debe tener atributos descriptivos, ya que cualquier atributo requerido puede estar asociado con el conjunto de entidades débiles. Los conjuntos de entidades débiles pueden participar en otras relaciones aparte de la relación identificadora. Por ejemplo, la entidad pago puede participar en una relación con el conjunto de entidades cuenta, identificando la cuenta desde la que se ha realizado el pago. Los conjuntos de entidades débiles pueden participar como propietario de una relación identificadora con otro conjunto de entidades débiles. También es posible tener conjuntos de entidades débiles con más de un conjunto de entidades identificadoras. Cada entidad débil se identificaría mediante una combinación de entidades, una de cada conjunto de entidades identificadoras. La clave primaria de la entidad débil consistiría de la unión de las claves primarias de los conjuntos de entidades identificadoras y el discriminante del conjunto de entidades débiles. En los diagramas E-R los rectángulos con líneas dobles indican conjuntos de entidades débiles, mientras que un rombo con líneas dobles indica la correspondiente relación de identificación. el conjunto de entidades débiles pago depende del conjunto de entidades fuertes préstamo mediante el conjunto de relaciones pago\_préstamo. La figura también ilustra el uso de líneas dobles para indicar participación total—la participación del conjunto de entidades (débiles) pago en la relación pago\_préstamo es total, lo que significa que cada pago debe estar relacionando mediante pago\_préstamo con alguna cuenta. Finalmente, la flecha de pago\_préstamo a préstamo indica que cada pago es para un solo préstamo. Los discriminantes de los conjuntos de entidades débiles también se subrayan, pero con una línea discontinua, en lugar de con una continua. En algunos casos, puede que el diseñador de la base de datos decida expresar un conjunto de entidades débiles como atributo compuesto multivalorado del conjunto de entidades propietarias. En el ejemplo, esta alternativa exigiría que el conjunto de entidades préstamo tuviera el atributo compuesto y multivalorado pago, que constara de número\_pago, fecha\_pago e importe\_pago. Los conjuntos de entidades débiles se pueden modelar mejor como atributos si sólo participan en la relación identificadora y tienen pocos atributos. A la inversa, las representaciones de los conjuntos de entidades débiles modelarán mejor las situaciones en las que esos conjuntos participen en otras relaciones aparte de la relación identificadora y tengan muchos atributos. Como ejemplo adicional de conjunto de entidades que se puede modelar como conjunto de entidades débiles, considérese las ofertas de asignaturas en una universidad. La misma asignatura se puede ofrecer en diferentes cursos y, dentro de cada curso, puede haber varios grupos para la misma asignatura. Por tanto, se puede crear el conjunto de entidades débiles oferta\_asignatura, que depende existencialmente de asignatura; las diferentes ofertas de la misma asignatura se identifican mediante curso y número\_grupo, que forman un discriminante pero no una clave primaria.



## 6.7 Características del modelo E-R extendido

Aunque los conceptos básicos del modelo E-R pueden modelar la mayor parte de las características de las bases de datos, algunos aspectos de las bases de datos se pueden expresar mejor mediante ciertas extensiones del modelo E-R básico.

### 6.7.1 Especialización

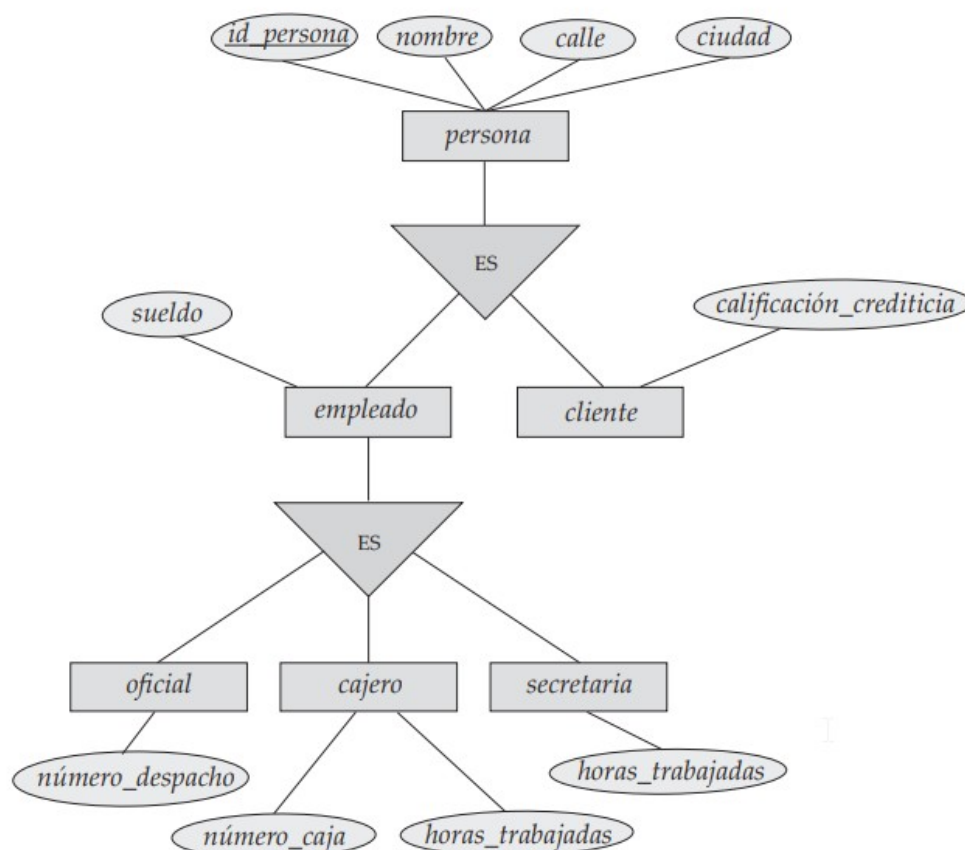
Los conjuntos de entidades pueden incluir subgrupos de entidades que se diferencian de alguna forma de las demás entidades del conjunto. Por ejemplo, un subconjunto de entidades de un conjunto de entidades puede tener atributos que no sean compartidos por todas las entidades del conjunto de entidades. El modelo E-R ofrece un medio de representar estos grupos de entidades diferentes. Como ejemplo, considérese el conjunto de entidades persona con los atributos `id_persona`, `nombre`, `calle` y `ciudad`. Cada persona puede clasificarse además en una de las categorías siguientes:

- Cliente
- Empleado

Cada uno de estos tipos de persona se describen mediante un conjunto de atributos que incluye todos los atributos del conjunto de entidades persona más otros posibles atributos adicionales. Por ejemplo, las entidades cliente se pueden describir además mediante el atributo `calificación_crediticia`, mientras que las entidades empleado se pueden describir además mediante el atributo `sueldo`. El proceso de establecimiento de subgrupos dentro del conjunto de entidades se denomina especialización. La especialización de persona permite distinguir entre las personas basándonos en si son empleados o clientes: en general, cada persona puede ser empleado, cliente, las dos cosas o ninguna de ellas. Como ejemplo adicional, supóngase que el banco desea dividir las cuentas en dos categorías: cuentas corrientes y cuentas de ahorro. Las cuentas de ahorro necesitan un saldo mínimo, pero el banco puede establecer diferentes tasas de interés para cada cliente y ofrecer mejores tasas a los clientes preferentes. Las cuentas corrientes tienen una tasa de interés fija, pero permiten los descubiertos; hay que registrar el importe de los descubiertos de las cuentas corrientes. Cada uno de estos tipos de cuenta se describe mediante un conjunto de atributos que incluye todos los atributos del conjunto de entidades cuenta más otros atributos adicionales. El banco puede crear dos especializaciones de cuenta, por ejemplo, `cuenta_ahorro` y `cuenta_corriente`. Como ya se ha visto, las entidades cuenta se describen mediante los atributos `número_cuenta` y `saldo`. El conjunto de entidades `cuenta_ahorro` tendría todos los atributos de cuenta y el atributo adicional `tasa_interés`. El conjunto de entidades `cuenta_corriente` tendría todos los atributos de cuenta y el atributo adicional `importe_descubierto`. La especialización se puede aplicar repetidamente para refinar el esquema de diseño. Por ejemplo, los empleados del banco se pueden clasificar también en alguna de las categorías siguientes:

- Oficial
- Cajero
- Secretaria

Cada uno de estos tipos de empleado se describe mediante un conjunto de atributos que incluye todos los atributos del conjunto de entidades empleado y otros adicionales. Por ejemplo, las entidades oficial se pueden describir además por el atributo número\_despacho, las entidades cajero por los atributos número\_caja y horas\_trabajadas, y las entidades secretaria por el atributo horas\_trabajadas. Además, las entidades secretaria pueden participar en la relación secretaria\_de, que identifica a los empleados a los que ayuda cada secretaria. Cada conjunto de entidades se puede especializar en más de una característica distintiva. En este ejemplo, la característica distintiva entre las entidades empleado es el trabajo que desempeña cada empleado. Otra especialización coexistente se puede basar en si cada persona es un trabajador temporal o fijo, lo que da lugar a los conjuntos de entidades empleado\_temporal y empleado\_fijo. Cuando se forma más de una especialización en un conjunto de entidades, cada entidad concreta puede pertenecer a varias especializaciones. Por ejemplo, un empleado dado puede ser un empleado temporal y una secretaria. En términos de los diagramas E-R, la especialización se representa mediante un componente triangular etiquetado ES. La etiqueta ES representa, por ejemplo, que cada cliente “es” una persona. La relación ES también se puede denominar relación superclase-subclase. Los conjuntos de entidades de nivel superior e inferior se representan como conjuntos de entidades regulares—es decir, como rectángulos que contienen el nombre del conjunto de entidades.



### 6.7.2 Generalización

El refinamiento a partir del conjunto de entidades inicial en sucesivos niveles de subgrupos de entidades representa un proceso de diseño descendente en el que las distinciones se hacen explícitas. El proceso de diseño también puede proceder de forma ascendente, en la que varios conjuntos de entidades se sintetizan en un conjunto de entidades de nivel superior basado en características comunes. El diseñador de la base

de datos puede haber identificado primero el conjunto de entidades cliente con los atributos `id_cliente`, `nombre_cliente`, `calle_cliente`, `ciudad_cliente` y `calificación_creditticia`, y el conjunto de entidades empleado con los atributos `id_empleado`, `nombre_empleado`, `calle_empleado`, `ciudad_empleado` y `suelo_empleado`. Existen analogías entre el conjunto de entidades cliente y el conjunto de entidades empleado en el sentido de que tienen varios atributos que, conceptualmente, son iguales en los dos conjuntos de entidades: los atributos para el identificador, el nombre, la calle y la ciudad. Esta similitud se puede expresar mediante la generalización, que es una relación de contención que existe entre el conjunto de entidades de nivel superior y uno o varios conjuntos de entidades de nivel inferior. En este ejemplo, persona es el conjunto de entidades de nivel superior y cliente y empleado son conjuntos de entidades de nivel inferior. En este caso, los atributos que son conceptualmente iguales tienen nombres diferentes en los dos conjuntos de entidades de nivel inferior. Para crear generalizaciones los atributos deben tener un nombre común y representarse mediante la entidad de nivel superior persona. Se pueden usar los nombres de atributos `id_persona`, `nombre`, `calle` y `ciudad`. Los conjuntos de entidades de nivel superior e inferior también se pueden denominar con los términos superclase y subclase, respectivamente. El conjunto de entidades persona es la superclase de las subclases cliente y empleado. A todos los efectos prácticos, la generalización es una inversión simple de la especialización. Se aplicarán ambos procesos, combinados, en el transcurso del diseño del esquema E-R de una empresa. En términos del propio diagrama E-R no se distingue entre especialización y generalización. Los niveles nuevos de representación de las entidades se distinguen (especialización) o sintetizan (generalización) cuando el esquema de diseño llega a expresar completamente la aplicación de la base de datos y los requisitos del usuario de la base de datos. Las diferencias entre los dos enfoques se pueden caracterizar mediante su punto de partida y su objetivo global. La especialización parte de un único conjunto de entidades; destaca las diferencias entre las entidades del conjunto mediante la creación de diferentes conjuntos de entidades de nivel inferior. Esos conjuntos de entidades de nivel inferior pueden tener atributos o participar en relaciones que no se aplican a todas las entidades del conjunto de entidades de nivel inferior. Realmente, la razón de que el diseñador aplique la especialización es poder representar esas características distintivas. Si cliente y empleado no tuvieran atributos que no tuvieran las entidades persona ni participaran en relaciones diferentes de las relaciones en las que participan las entidades persona, no habría necesidad de especializar el conjunto de entidades persona. La generalización parte del reconocimiento de que varios conjuntos de entidades comparten algunas características comunes (es decir, se describen mediante los mismos atributos y participan en los mismos conjuntos de relaciones). Con base en esas similitudes, la generalización sintetiza esos conjuntos de entidades en un solo conjunto de nivel superior. La generalización se usa para destacar las similitudes entre los conjuntos de entidades de nivel inferior y para ocultar las diferencias; también permite una economía de representación, ya que no se repiten los atributos compartidos.

### **6.7.3 Herencia de los atributos**

Una propiedad crucial de las entidades de nivel superior e inferior creadas mediante la especialización y la generalización es la herencia de los atributos. Se dice que los atributos de los conjuntos de entidades de nivel superior son heredados por los conjuntos de entidades de nivel inferior.

Los conjuntos de entidades de nivel inferior (o subclases) también heredan la participación en los conjuntos de relaciones en los que participa su entidad de nivel superior (o superclase). Los conjuntos de entidades oficial, cajero y secretaria pueden participar en el conjunto de relaciones `trabaja_para`, ya que la superclase empleado participa en la relación `trabaja_para`. La herencia de los atributos se aplica a todas las capas de conjuntos de entidades de nivel inferior. Los conjuntos de entidades anteriores

pueden participar en cualquier relación en la que participe el conjunto de entidades persona. Tanto si se llega a una porción dada del modelo E-R mediante la especialización como si se hace mediante la generalización, el resultado es básicamente el mismo:

- Un conjunto de entidades de nivel superior con los atributos y las relaciones que se aplican a todos sus conjuntos de entidades de nivel inferior.
- Conjuntos de entidades de nivel inferior con características distintivas que sólo se aplican en un conjunto dado de entidades de nivel inferior.

En las jerarquías un conjunto de entidades dado sólo puede estar implicado como conjunto de entidades de nivel inferior en una relación ES; es decir, los conjuntos de entidades de este diagrama sólo tienen herencia única. Si un conjunto de entidades es un conjunto de entidades de nivel inferior en más de una relación ES, el conjunto de entidades tiene herencia múltiple, y la estructura resultante se denomina retículo.

#### 6.7.4 Restricciones a las generalizaciones

Para modelar una empresa con más precisión, el diseñador de la base de datos puede decidir imponer ciertas restricciones sobre una generalización concreta. Un tipo de restricción implica la determinación de las entidades que pueden formar parte de un conjunto de entidades de nivel inferior dado. Esa pertenencia puede ser una de las siguientes:

- **Definida por la condición.** En los conjuntos de entidades de nivel inferior definidos por la condición, la pertenencia se evalúa en función del cumplimiento de una condición o predicado explícito por la entidad. Por ejemplo, supóngase que el conjunto de entidades de nivel superior cuenta tiene el atributo tipo\_cuenta. Todas las entidades cuenta se evalúan según el atributo tipo\_cuenta que las define. Sólo las entidades que satisfacen la condición tipo\_cuenta = "cuenta de ahorro" pueden pertenecer al conjunto de entidades de nivel inferior cuenta\_ahorro. Todas las entidades que satisfacen la condición tipo\_cuenta = "cuenta corriente" se incluyen en cuenta\_corriente. Dado que todas las entidades de nivel inferior se evalúan en función del mismo atributo (en este caso, tipo\_cuenta), se dice que este tipo de generalización está definida por el atributo.
- **Definida por el usuario.** Los conjuntos de entidades de nivel inferior definidos por el usuario no están restringidos por una condición de pertenencia; más bien, el usuario de la base de datos asigna las entidades a un conjunto de entidades dado. Por ejemplo, supóngase que, después de tres meses de trabajo, los empleados del banco se asignan a uno de los cuatro grupos de trabajo. En consecuencia, los grupos se representan como cuatro conjuntos de entidades de nivel inferior del conjunto de entidades de nivel superior empleado. No se asigna cada empleado a una entidad grupo concreta automáticamente de acuerdo con una condición explícita que lo defina. En vez de eso, la asignación al grupo la lleva a cabo el usuario que toma persona a persona. La asignación se implementa mediante una operación que añade cada entidad a un conjunto de entidades.

Un segundo tipo de restricciones tiene relación con la pertenencia de las entidades a más de un conjunto de entidades de nivel inferior de la generalización. Los conjuntos de entidades de nivel inferior pueden ser de uno de los tipos siguientes:

- **Disjuntos.** La restricción sobre la condición de disjunción exige que cada entidad no pertenezca a más de un conjunto de entidades de nivel inferior. En el ejemplo, cada

entidad cuenta sólo puede cumplir una condición del atributo tipo\_cuenta; cada entidad puede ser una cuenta de ahorro o una cuenta corriente, pero no ambas cosas a la vez.

- Solapados. En las generalizaciones solapadas la misma entidad puede pertenecer a más de un conjunto de entidades de nivel inferior de la generalización. Como ilustración, considérese el ejemplo del grupo de trabajo de empleados y supóngase que algunos directores participan en más de un grupo de trabajo. Cada empleado, por tanto, puede aparecer en más de uno de los conjuntos de entidades grupo que son conjuntos de entidades de nivel inferior de empleado. Por tanto, la generalización es solapada. Como ejemplo adicional, supóngase que la generalización aplicada a los conjuntos de entidades cliente y empleado conduce a un conjunto de entidades de nivel superior persona. La generalización es solapada si los empleados también pueden ser clientes.

El solapamiento de las entidades de nivel inferior es el caso predeterminado; la restricción sobre la condición de disjunción se debe imponer explícitamente a la generalización (o especialización). La restricción sobre condición de disjunción se puede denotar en los diagramas E-R añadiendo la palabra disjunta junto al símbolo del triángulo. Una última restricción, la restricción de completitud sobre una generalización o especialización, especifica si una entidad del conjunto de entidades de nivel superior debe pertenecer, al menos, a uno de los conjuntos de entidades de nivel inferior de la generalización o especialización. Esta restricción puede ser de uno de los tipos siguientes:

- Generalización o especialización total. Cada entidad de nivel superior debe pertenecer a un conjunto de entidades de nivel inferior.
- Generalización o especialización parcial. Puede que alguna entidad de nivel superior no pertenezca a ningún conjunto de entidades de nivel inferior

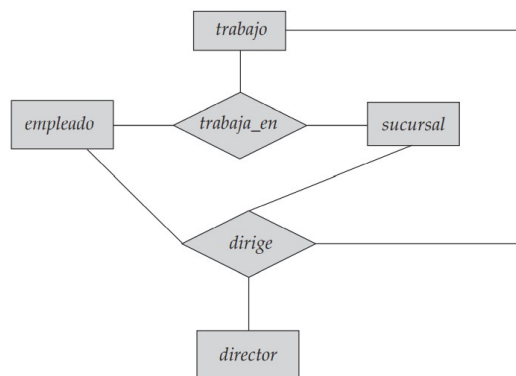
La generalización parcial es la predeterminada. Se puede especificar la generalización total en los diagrama E-R usando una línea doble para conectar el rectángulo que representa el conjunto de entidades de nivel superior con el símbolo del triángulo. (Esta notación es parecida a la usada para la participación total en una relación). La generalización de cuenta es total: todas las entidades cuenta deben ser cuentas de ahorro o cuentas corrientes. Como el conjunto de entidades de nivel superior al que se llega mediante la generalización suele estar compuesto únicamente de entidades de los conjuntos de entidades de nivel inferior, la restricción de completitud para los conjuntos de entidades de nivel superior generalizados suele ser total. Cuando la restricción es parcial, las entidades de nivel superior no están limitadas a aparecer en los conjuntos de entidades de nivel inferior. Los conjuntos de entidades grupo de trabajo ilustran una especialización parcial. Como los empleados sólo se asignan a cada grupo después de llevar tres meses en el trabajo, puede que algunas entidades empleado no pertenezcan a ninguno de los conjuntos de entidades grupo de nivel inferior.

Los conjuntos de entidades equipo se pueden caracterizar mejor como especialización de empleado parcial y solapada. La generalización de cuenta\_corriente y cuenta\_ahorro en cuenta es una generalización total y disjunta. Las restricciones de completitud y sobre la condición de disjunción, sin embargo, no dependen una de la otra. Las características de las restricciones también pueden ser parcial—disjunta y total—solapada. Es evidente que algunos requisitos de inserción y de borrado son consecuencia de las restricciones que se aplican a una generalización o especialización dada. Por ejemplo, cuando se impone una restricción de completitud total, las entidades insertadas en un conjunto de entidades de

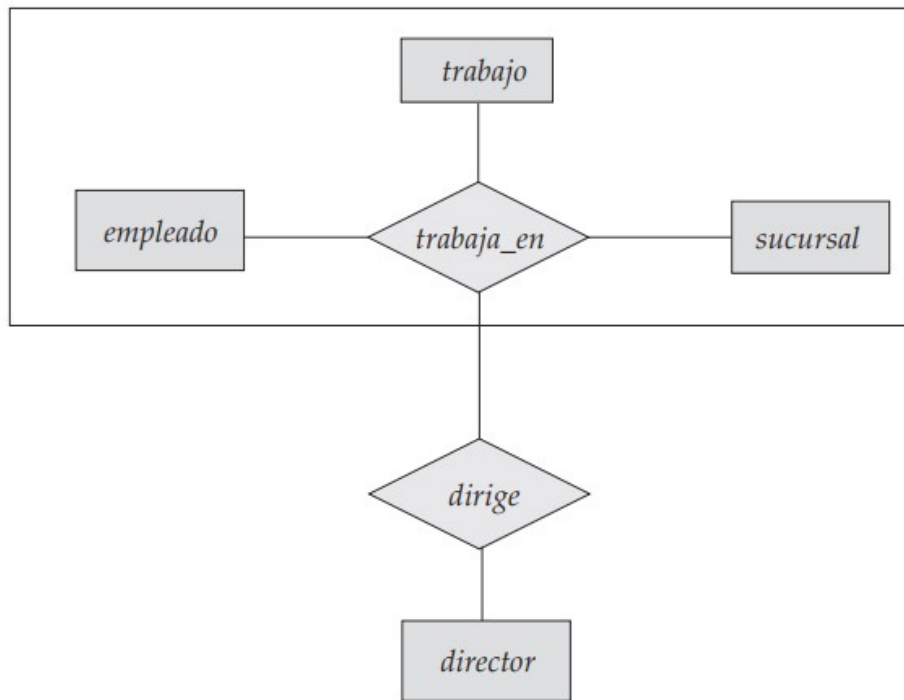
nivel superior se deben insertar, al menos, en uno de los conjuntos de entidades de nivel inferior. Con una restricción de definición por condición, todas las entidades de nivel superior que cumplen la condición se deben insertar en ese conjunto de entidades de nivel inferior. Finalmente, las entidades que se borren de los conjuntos de entidades de nivel superior, se deben borrar también de todos los conjuntos de entidades de nivel inferior asociados a los que pertenezcan.

### 6.7.5 Agregación

Una limitación del modelo E-R es que no es posible expresar relaciones entre las relaciones. Para ilustrar la necesidad de estos constructores, considérese la relación ternaria trabaja\_en, que se ya se ha visto anteriormente, entre empleado, sucursal y trabajo. Supóngase ahora que se desea registrar el director responsable de las tareas realizadas por cada empleado de cada sucursal; es decir, se desea registrar al director responsable de las combinaciones (empleado, sucursal, trabajo). Supóngase que existe un conjunto de entidades director. Una alternativa para representar esta relación es crear una relación cuaternaria dirige entre empleado, sucursal, trabajo y director (se necesita una relación cuaternaria—una relación binaria entre director y empleado no permitiría representar las combinaciones (sucursal, trabajo) de cada empleado que son responsabilidad de cada director). Mediante los constructores de modelado básicos del modelo E-R se obtiene el diagrama E-R (por simplificar se han omitido los atributos de los conjuntos de entidades).



Parece que los conjuntos de relaciones trabaja\_en y dirige se pueden combinar en un solo conjunto de relaciones. No obstante, no se deben combinar en una sola relación, ya que puede que algunas combinaciones empleado, sucursal, trabajo no tengan director. No obstante, hay información redundante en la figura obtenida, ya que cada combinación empleado, sucursal, trabajo de dirige también está en trabaja\_en. Si el director fuese un valor en lugar de una entidad director, se podría hacer que director fuese un atributo multivalorado de la relación trabaja\_en. Pero eso dificulta (tanto lógicamente como en coste de ejecución) encontrar, por ejemplo, las tripletas empleadosucursal-trabajo de las que es responsable cada director. Como el director es una entidad director, esta alternativa queda descartada en cualquier caso. La mejor forma de modelar una situación como la descrita es usar la agregación. La agregación es una abstracción a través de la cual las relaciones se tratan como entidades de nivel superior. Así, para este ejemplo, se considera el conjunto de relaciones trabaja\_en (que relaciona los conjuntos de entidades empleado, sucursal y trabajo) como el conjunto de entidades de nivel superior denominado trabaja\_en. Ese conjunto de entidades se trata de la misma forma que cualquier otro conjunto de entidades. Se puede crear entonces la relación binaria dirige entre trabaja\_en y director para representar al responsable de cada tarea.



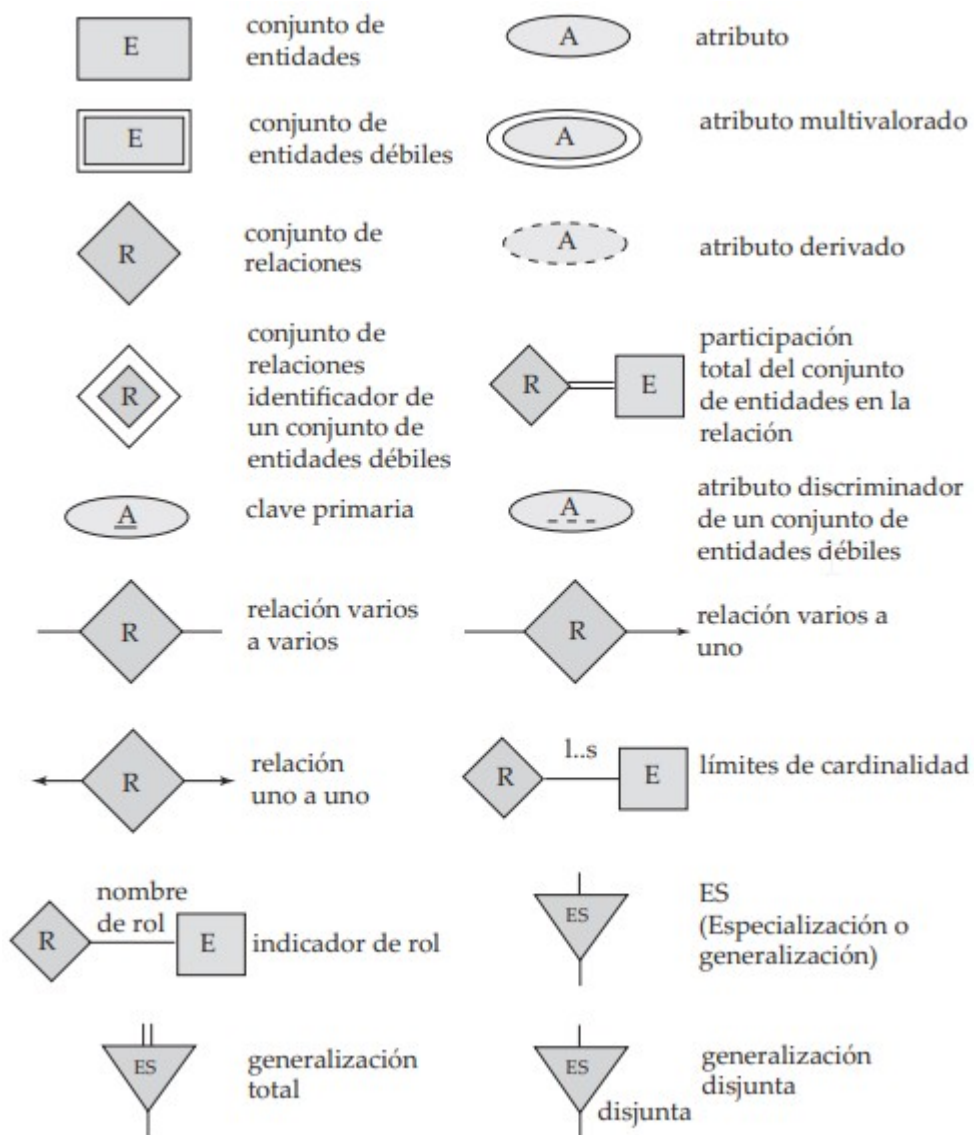
#### 6.7.6 Notaciones E-R alternativas

No hay una norma universal para la notación de los diagramas E-R, y cada libro y cada programa informático de diagramas E-R usa notaciones diferentes. Los conjuntos de entidades se pueden representar como rectángulos con el nombre por fuera, y los atributos relacionados unos debajo de los otros dentro del rectángulo. Los atributos clave primaria se indican relacionándolos en la parte superior, con una línea que los separa de los demás atributos. Las restricciones de cardinalidad se pueden indicar de varias formas, Las etiquetas \* y 1 en los segmentos que salen de las relaciones se usan a menudo para denotar relaciones varios a varios, uno a uno y varios a uno, El caso de uno a varios es simétrico con el de varios a uno y no se muestra. En otra notación alternativa de la figura los conjuntos de relaciones se representan mediante líneas entre los conjuntos de entidades, sin rombos; por tanto, sólo se podrán modelar relaciones binarias. Desafortunadamente no existe una notación E-R normalizada. La notación que se usa en este libro, con rectángulos, rombos y elipses se denomina notación de Chen, y la usó Chen en el artículo que introdujo el concepto de modelado E-R. El Instituto Nacional de EEUU para Normalización y Tecnología (U.S. National Institute for Standards and Technology) definió una norma denominada IDEF1X en 1993, que usa la notación de pata de gallo. IDEF1X también incluye gran variedad de notaciones diferentes que no se han mostrado, incluidas las barras verticales en los segmentos de las relaciones para indicar participación total y los círculos vacíos para denotar participación parcial.

#### 6.8 Diseño de una base de datos para un banco

Ahora se centrará la atención en los requisitos de diseño de la base de datos de una entidad bancaria con más detalle y se desarrollará un diseño más realista, aunque también más complicado, de lo que se ha visto en los ejemplos anteriores.

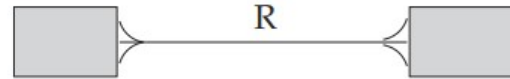
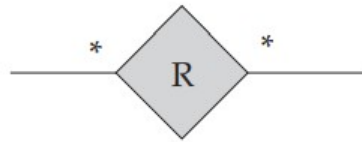




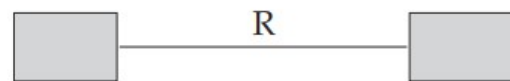
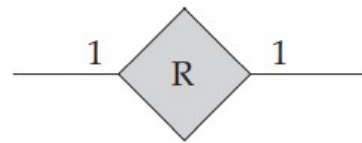
conjunto de entidades E  
con atributos A1, A2 y A3,  
y clave primaria A1

E
A1
A2
A3

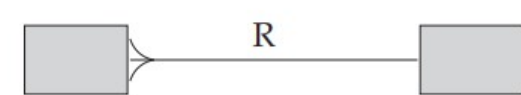
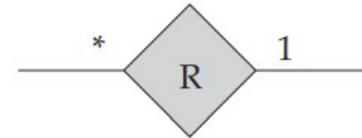
relación varios  
a varios



relación uno  
a uno



relación varios  
a uno



### 6.8.1 Alternativas de diseño E-R

El modelo de datos E-R permite gran flexibilidad en el diseño de los esquemas de las bases de datos para modelar una empresa dada. Entre las decisiones que debe tomar el diseñador están:

- Si usar atributos o conjuntos de entidades para representar los objetos
- Si los conceptos del mundo real se expresan mejor mediante conjuntos de entidades o mediante conjuntos de relaciones
- Si usar relaciones ternarias o pares de relaciones binarias
- Si usar conjuntos de entidades fuertes o débiles ; cada conjunto de entidades fuertes y sus conjuntos de entidades débiles se puede considerar como un solo "objeto" de la base de datos, ya que las entidades débiles dependen existencialmente de la entidad fuerte
- Si es adecuado usar la generalización ; la generalización, o la jerarquía de relaciones ES, contribuye a la modularidad al permitir que los atributos comunes de entidades parecidas se representen en un solo sitio del diagrama E-R.
- Si es adecuado usar la agregación (Apartado 6.7.5); la agregación agrupa parte del diagrama E-R en un solo conjunto de entidades, lo que permite tratar el conjunto de entidades agregadas como una sola unidad sin necesidad de preocuparse por los detalles de su estructura interna.

### 6.8.2 Requisitos de datos de la base de datos bancaria

La especificación inicial de los requisitos de los usuarios se puede basar en entrevistas con los usuarios de la base de datos y en el análisis propio del diseñador de la empresa. La descripción que surge de esta fase de diseño sirve de base para concretar la estructura conceptual de la base de datos. La siguiente lista describe las principales características de la entidad bancaria.

- El banco está organizado en sucursales. Cada sucursal está ubicada en una ciudad concreta y se identifica con un nombre único. El banco supervisa los activos de cada sucursal.
- Los clientes del banco se identifican mediante su valor de `id_cliente`. El banco almacena cada nombre de cliente, y la calle y la ciudad donde vive cada cliente. Los clientes pueden tener cuentas y pueden solicitar préstamos. Cada cliente puede estar asociado con un empleado del banco concreto, que puede actuar como responsable de préstamos o como asesor personal de ese cliente.
- Los empleados del banco se identifican mediante su valor de `id_empleado`. La administración del banco almacena el nombre y el número de teléfono de cada empleado, el nombre de los subordinados de cada empleado, y el número `id_empleado` del jefe de cada empleado. El banco también mantiene un registro de la fecha de incorporación a la empresa del empleado y, por tanto, de su antigüedad.
- El banco ofrece dos tipos de cuentas: cuentas de ahorro y cuentas corrientes. Las cuentas pueden tener como titular a más un cliente, y cada cliente puede tener más de una cuenta. Cada cuenta tiene asignado un número de cuenta único. El banco mantiene un registro del saldo de cada cuenta y de la fecha más reciente en que cada titular de la cuenta tuvo acceso a esa cuenta. Además, cada cuenta de ahorro tiene un tipo de interés y para cada cuenta corriente se registran los descubiertos generados.
- Cada préstamo se genera en una sucursal concreta y pueden solicitarlo uno o más clientes. Cada préstamo se identifica mediante un número de préstamo único. Para cada préstamo el banco mantiene un registro del importe del préstamo y de los pagos realizados y pendientes. Aunque los números de los pagos del préstamo no identifican de forma unívoca cada pago entre los de todos los préstamos del banco, el número de pago sí que identifica cada pago de un préstamo concreto. De cada pago se registran la fecha y el importe.

### 6.8.3 Conjuntos de entidades de la base de datos bancaria

La especificación de los requisitos de datos sirve como punto de partida para la construcción del esquema conceptual de la base de datos. Los conjuntos de entidades y sus atributos:

- El conjunto de entidades sucursal, con los atributos `nombre_sucursal`, `ciudad_sucursal` y `activos`.
- El conjunto de entidades cliente, con los atributos `id_cliente`, `nombre_cliente`, `calle_cliente` y `ciudad_cliente`. Un posible atributo adicional es `nombre_asesor`.
- El conjunto de entidades empleado, con los atributos `id_empleado`, `nombre_empleado`, `número_teléfono`, `sueldo` y `jefe`. Algunas características descriptivas adicionales son el atributo multivalorado `nombre_subordinado`, el atributo básico `fecha_contratación` y el atributo derivado `antigüedad`.
- Dos conjuntos de entidades cuenta—`cuenta_ahorro` y `cuenta_corriente`—con los atributos comunes `número_cuenta` y `saldo`; además, `cuenta_ahorro` tiene el atributo `tipo_interés` y `cuenta_corriente` tiene el atributo `descubierto`.
- El conjunto de entidades préstamo, con los atributos `número_préstamo`, `importe` y `sucursal_origen`.
- El conjunto de entidades débiles pago\_préstamo, con los atributos `número_pago`, `fecha_pago` e `importe_pago`.

### 6.8.4 Conjuntos de relaciones de la base de datos bancaria

Se pueden especificar los conjuntos de relaciones y correspondencias de cardinalidades siguientes. En el proceso también se perfeccionan algunas de las decisiones tomadas anteriormente en relación con los atributos de los conjuntos de entidades.

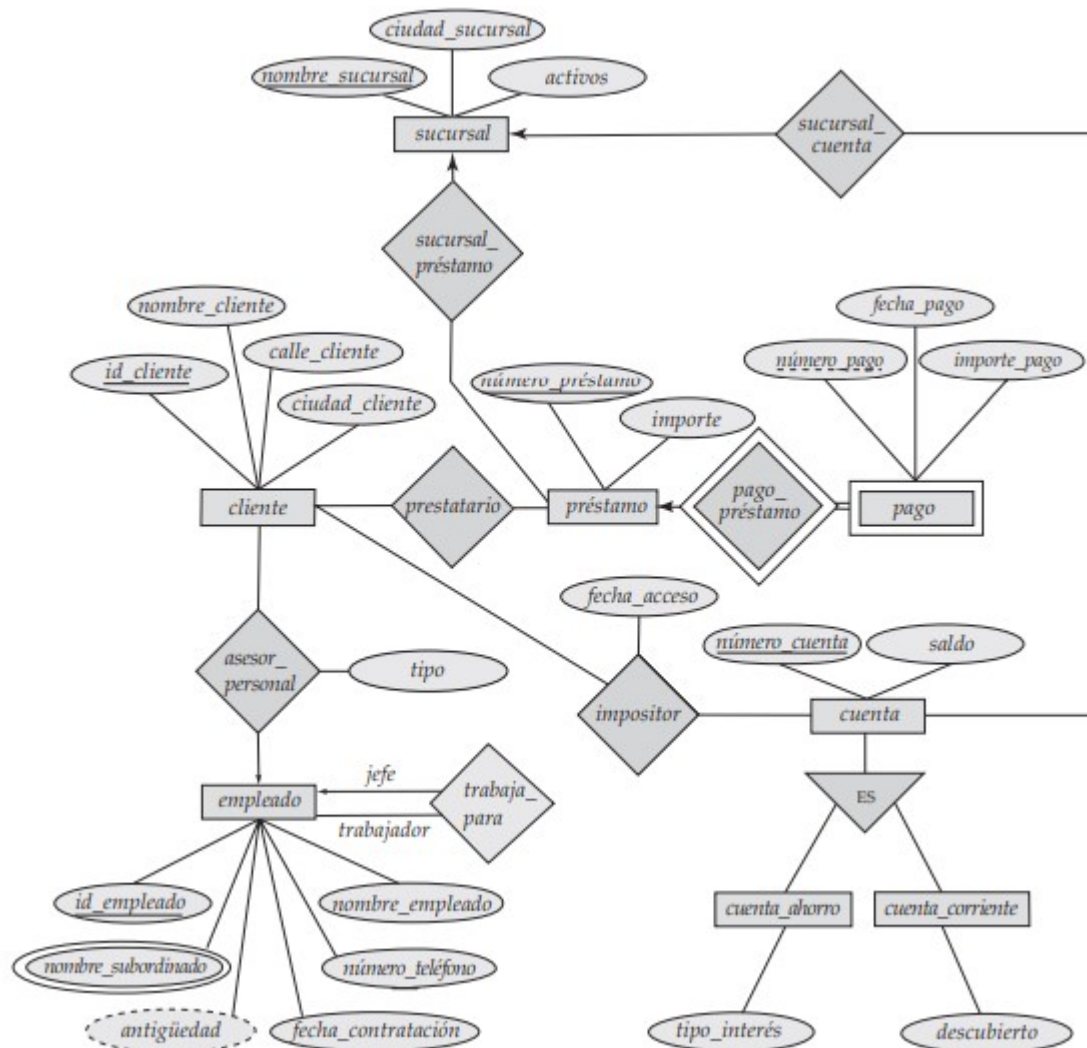
- prestatario, un conjunto de relaciones varios a varios entre cliente y préstamo.
- sucursal\_préstamo, un conjunto de relaciones varios a uno que indica la sucursal en que se ha originado un préstamo. Obsérvese que este conjunto de relaciones sustituye al atributo sucursal \_origen del conjunto de entidades préstamo.
- pago\_préstamo, un conjunto de relaciones uno a varios de préstamo a pago, que documenta que se ha realizado un pago de un préstamo.
- impositor, con el atributo de relación fecha\_acceso, un conjunto de relaciones varios a varios entre cliente y cuenta, que indica que un cliente posee una cuenta.
- asesor\_personal, con el atributo de relación tipo, un conjunto de relaciones varios a uno que expresa que un cliente puede ser asesorado por un empleado del banco, y que un empleado del banco puede asesorar a uno o más clientes. Obsérvese que este conjunto de relaciones ha sustituido al atributo nombre\_asesor del conjunto de entidades cliente.
- trabaja\_para, un conjunto de relaciones entre entidades empleado con los indicadores de rol jefe y trabajador; la correspondencia de cardinalidades expresa que cada empleado trabaja para un único jefe, y que cada jefe supervisa a uno o más empleados. Obsérvese que este conjunto de relaciones ha sustituido al atributo jefe de empleado.

#### **6.8.5 Diagrama E-R de la base de datos bancaria**

El diagrama E-R de esta visión simplificada de una entidad bancaria ya es bastante complejo. Los diagramas E-R de las empresas reales no se pueden dibujar en una sola página y hay que dividirlos en varias partes. Puede hacer falta que las entidades aparezcan varias veces en diferentes partes del diagrama. Los atributos de cada entidad se muestran en una sola aparición de la entidad (preferiblemente la primera) y todas las demás apariciones de la entidad se muestran sin atributos.

#### **6.9 Reducción a esquemas relacionales**

Las bases de datos que se ajustan a un esquema de bases de datos E-R se pueden representar mediante conjuntos de esquemas de relación. Para cada conjunto de entidades y para cada conjunto de relaciones de la base de datos hay un solo esquema de relación a la que se asigna el nombre del conjunto de entidades o del conjunto de relaciones correspondiente. Tanto el modelo E-R de bases de datos como el relacional son representaciones abstractas y lógicas de empresas del mundo real. Como los dos modelos usan principios de diseño parecidos, los diseños E-R se pueden convertir en diseños relacionales. En este apartado se describe la manera de representar los esquemas E-R mediante esquemas de relación y el modo de asignar las restricciones que surgen del modelo E-R a restricciones de los esquemas de relación



### 6.9.1 Representación de los conjuntos de entidades fuertes

Sea  $E$  un conjunto de entidades fuertes con los atributos descriptivos  $a_1, a_2, \dots, a_n$ . Esta entidad se representa mediante un esquema denominado  $E$  con  $n$  atributos distintos. Cada tupla de las relaciones de este esquema corresponde a una entidad del conjunto de entidades  $E$ . Para los esquemas derivados de los conjuntos de entidades fuertes la clave primaria del conjunto de entidades sirve de clave primaria de los esquemas resultantes. Esto se deduce directamente del hecho de que cada tupla corresponde a una entidad concreta del conjunto de entidades. Este conjunto de entidades tiene dos atributos: número\_préstamo e importe. Como ilustración, considérese el conjunto de entidades préstamo. Este conjunto de entidades se representa mediante un esquema denominado préstamo, con dos atributos: **préstamo = (número\_préstamo, importe)**

### 6.9.2 Representación de los conjuntos de entidades débiles

Sea  $A$  un conjunto de entidades débiles con los atributos  $a_1, a_2, \dots, a_m$ . Sea  $B$  el conjunto de entidades fuertes del que  $A$  depende. La clave primaria de  $B$  consiste en los atributos  $b_1, b_2, \dots, b_n$ . El conjunto de entidades  $A$  se representa mediante el esquema de relación denominado  $A$  con una columna por cada miembro del conjunto:  $\{a_1, a_2, \dots, a_m\} \cup \{b_1, b_2, \dots, b_n\}$ . Para los esquemas derivados de conjuntos de entidades débiles la combinación de la clave primaria del conjunto de entidades fuertes y del discriminador del conjunto de entidades débiles sirve de clave primaria del esquema. Además de crear una clave primaria, también se crea una restricción de clave externa para la relación  $A$ , que especifica que los atributos  $b_1, b_2, \dots, b_n$  hacen referencia a la clave primaria de la relación  $B$ . La restricción de clave externa garantiza que por cada tupla que represente a

una entidad débil exista la tupla correspondiente que representa a la entidad fuerte correspondiente. Como ilustración, considérese el conjunto de entidades pago. Este conjunto de entidades tiene tres atributos: **número\_pago**, **fecha\_pago** e **importe\_pago**. La clave primaria del conjunto de entidades préstamo, de la que pago depende, es **número\_préstamo**. Por tanto, pago se representa mediante un esquema con cuatro atributos: **pago = (número\_préstamo, número\_pago, fecha\_pago, importe)**. La clave primaria consiste en la clave primaria de préstamo y el discriminador de pago (**número\_pago**). También se crea una restricción de clave externa para el esquema pago, con el atributo **número\_pago** que hace referencia a la clave primaria del esquema préstamo.

### 6.9.3 Representación de los conjuntos de relaciones

Sea  $R$  un conjunto de relaciones, sea  $a_1, a_2, \dots, a_m$  el conjunto de atributos formado por la unión de las claves primarias de cada uno de los conjuntos de entidades que participan en  $R$ , y sean  $b_1, b_2, \dots, b_n$  los atributos descriptivos de  $R$  (si los hay). El conjunto de relaciones se representa mediante el esquema de relación  $R$ , con un atributo por cada uno de los miembros del conjunto:  $\{a_1, a_2, \dots, a_m\} \cup \{b_1, b_2, \dots, b_n\}$ . Tomar todos los atributos de las claves primarias de todos los conjuntos de entidades primarias sirve para identificar una tupla concreta pero, para los conjuntos de relaciones uno a uno, varios a uno y uno a varios, esto resulta un conjunto de atributos mayor del que hace falta en la clave primaria. En vez de eso, la clave primaria se escoge de la manera siguiente:

- Para las relaciones binarias varios a varios la unión de los atributos de clave primaria de los conjuntos de entidades participantes pasa a ser la clave primaria.
- Para los conjuntos de relaciones binarias uno a uno la clave primaria de cualquiera de los conjuntos de entidades puede escogerse como clave primaria de la relación. La elección del conjunto de entidades de entre los relacionados por el conjunto de relaciones puede realizarse de manera arbitraria.
- Para los conjuntos de relaciones binarias varios a uno o uno a varios la clave primaria del conjunto de entidades de la parte “varios” de la relación sirve de clave primaria.
- Para los conjuntos de relaciones  $n$ -arias sin flechas en los segmentos la unión de los atributos de clave primaria de los conjuntos de entidades participantes pasa a ser la clave primaria.
- Para los conjuntos de relaciones  $n$ -arias con una flecha en uno de los segmentos las claves primarias de los conjuntos de entidades que no están en el lado “flecha” del conjunto de relaciones sirven de clave primaria del esquema. Recuérdese que sólo se permite una flecha saliente por conjunto de relaciones.

También se crean restricciones de clave externa para la relación  $R$  de la manera siguiente. Para cada conjunto de entidades  $E_i$  relacionado con el conjunto de relaciones  $R$  se crea una restricción de clave primaria de la relación  $R$ , con los atributos de  $R$  que eran atributos de clave primaria de  $E_i$  que hacen referencia a la clave primaria de la relación que representa  $E_i$ . Considérese el conjunto de relaciones prestatario. Este conjunto de relaciones implica a los dos conjuntos de entidades siguientes:

- cliente, con la clave primaria **id\_cliente**
- préstamo, con la clave primaria **número\_préstamo**

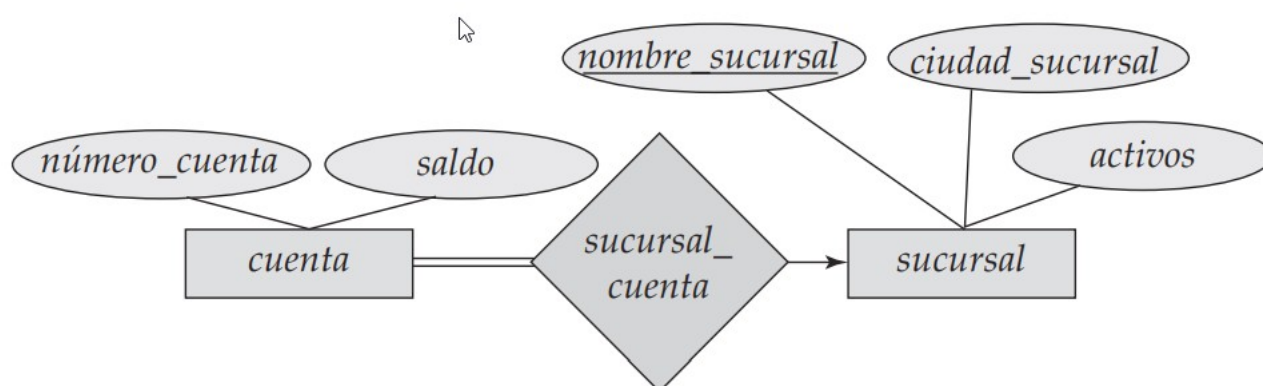
Como el conjunto de relaciones no tiene ningún atributo, el esquema prestatario tiene dos atributos: **prestatario = (id\_cliente, número\_préstamo)**. La clave primaria de la relación prestatario es la unión de los atributos de clave primaria de cliente y de préstamo.

También se crean dos restricciones de clave externa para la relación prestatario, con el

atributo `id_cliente` que hace referencia a la clave primaria de cliente y el atributo `número_préstamo` que hace referencia a la clave primaria de préstamo.

### 6.9.3.1 Redundancia de los esquemas

Los conjuntos de relaciones que enlazan los conjuntos de entidades débiles con el conjunto correspondiente de entidades fuertes se tratan de manera especial. Estas relaciones son varios a uno y no tienen atributos descriptivos. Además, la clave primaria de los conjuntos de entidades débiles incluye la clave primaria de los conjuntos de entidades fuertes. La clave primaria de pago es `número_préstamo`, `número_pago` y la clave primaria de préstamo es `número_préstamo`. Como `pago_préstamo` no tiene atributos descriptivos, el esquema `pago_préstamo` tiene dos atributos, `número_préstamo` y `número_pago`. El esquema del conjunto de entidades pago tiene cuatro atributos, `número_préstamo`, `número_pago`, `fecha_pago` e `importe_pago`. Cada combinación (`número_préstamo`, `número_pago`) de una relación de `pago_préstamo` también se halla presente en el esquema de relación pago, y viceversa. Por tanto, el esquema `pago_préstamo` es redundante. En general, el esquema de los conjuntos de relaciones que enlazan los conjuntos de entidades débiles con su conjunto correspondiente de entidades fuertes es redundante y no hace falta que esté presente en el diseño de la base de datos relacional basado en el diagrama E-R.



### 6.9.3.2 Combinación de esquemas

Considérese un conjunto AB de relaciones varios a uno del conjunto de entidades A al conjunto de entidades B. Usando el esquema de construcción de esquemas de relación descrito previamente se consiguen tres esquemas: A, B y AB. Supóngase, además, que la participación de A en la relación es total; es decir, todas las entidades A del conjunto de entidades A deben participar en la relación AB. Entonces se pueden combinar los esquemas A y AB para formar un solo esquema consistente en la unión de los atributos de los dos esquemas.

. La línea doble del diagrama E-R indica que la participación de cuenta en `cuenta_sucursal` es total. Por tanto, no puede haber ninguna cuenta que no esté asociada a alguna sucursal. Además, el conjunto de relaciones `cuenta_sucursal` es varios a uno de cuenta a sucursal. Por lo tanto, se puede combinar el esquema de `cuenta_sucursal` con el esquema de cuenta y sólo se necesitan los dos esquemas siguientes:

- `cuenta` = (`número_cuenta`, `saldo`, `nombre_sucursal`)
- `sucursal` = (`nombre_sucursal`, `ciudad_sucursal`, `activos`)

En el caso de las relaciones uno a uno, el esquema de relación del conjunto de relaciones puede combinarse con el esquema de cualquiera de los conjuntos de entidades. Se pueden combinar esquemas aunque la participación sea parcial, usando los valores nulos;



en el ejemplo anterior se almacenarían valores nulos para el atributo nombre\_sucursal de las cuentas que no tengan sucursal asociada. La clave primaria del esquema combinado es la clave primaria del conjunto de entidades en cuyo esquema se ha fusionado el conjunto de relaciones. En el ejemplo anterior, la clave primaria es número\_cuenta. En circunstancias normales el esquema que representa al conjunto de relaciones habría tenido restricciones de clave externa que harían referencia a cada uno de los conjuntos de entidades que participan en el conjunto de relaciones. En este caso se ha descartado la restricción que hace referencia al conjunto de entidades en cuyo esquema se ha fusionado el conjunto de relaciones y se ha añadido las demás restricciones de clave externa al esquema combinado. En el ejemplo anterior, se descarta la restricción de clave externa que hace referencia a cuenta, pero se conserva la restricción de clave externa en la que nombre\_sucursal hace referencia a sucursal como restricción del esquema combinado cuenta.

#### 6.9.4 Atributos compuestos y multivalorados

Los atributos compuestos se tratan mediante la creación de un atributo diferente para cada uno de los atributos componentes; no se crea ningún atributo para el atributo compuesto propiamente dicho. Supóngase que dirección es un atributo compuesto del conjunto de entidades cliente y que los componentes de dirección son calle y ciudad. El esquema generado a partir de cliente contiene las columnas calle\_dirección y ciudad\_dirección; no hay ningún atributo ni esquema para dirección.

Se ha visto que en los diagramas E-R los atributos se suelen asignar directamente a las columnas de los esquemas de relación correspondientes. Los atributos multivalorados, sin embargo, son una excepción; para estos atributos se crean esquemas de relación nuevos. Para cada atributo multivalorado M se crea un esquema de relación E con un atributo A que corresponde a M y a los atributos correspondientes a la clave primaria del conjunto de entidades o de relaciones del que M es atributo. El diagrama incluye el conjunto de entidades empleado con el atributo multivalorado nombre\_subordinado. La clave primaria de empleado es id\_empleado. Para este atributo multivalorado se crea el esquema de relación **nombre\_subordinado (id\_empleado, nombre\_subordinado)**. En este esquema cada subordinado de un empleado se representa mediante una sola tupla de la relación. Por tanto, si se tuviera un empleado con id\_empleado 12-234 y los subordinados Martín y María, la relación nombre\_subordinado tendría dos tuplas, (12-234, Martín) y (12-234, María). Se crea una clave primaria del esquema de la relación consistente en todos los atributos del esquema. En el ejemplo anterior, la clave primaria consiste en todos los atributos de la relación nombre\_subordinado. Además, se crea una clave externa para el esquema de la relación, con el atributo generado a partir de la clave primaria del conjunto de entidades que hace referencia a la relación generada a partir del conjunto de entidades. En el ejemplo anterior, la restricción sería que el atributo id\_empleado hace referencia a la relación empleado.

#### 6.9.5 Representación de la generalización

Existen dos métodos diferentes para designar los esquemas de relación de los diagramas E-R que incluyen generalización. Se da por supuesto que id\_persona es la clave primaria de persona.

1. Se crea un esquema para el conjunto de entidades de nivel superior. Para cada conjunto de entidades de nivel inferior se crea un esquema que incluye un atributo para cada uno de los atributos de ese conjunto de entidades más un atributo por cada atributo de la clave primaria del conjunto de entidades de nivel superior, se tienen tres esquemas:
  - persona = (id\_persona, nombre, calle, ciudad)
  - empleado = (id\_persona, sueldo)



- cliente = (id\_persona, calificación\_crediticia)

Los atributos de clave primaria del conjunto de entidades de nivel superior pasan a ser atributos de clave primaria del conjunto de entidades de nivel superior y de todos los conjuntos de entidades de nivel inferior. Además, se crean restricciones de clave externa para los conjuntos de entidades de nivel inferior, con sus atributos de clave primaria que hacen referencia a la clave primaria de la relación creada a partir del conjunto de entidades de nivel superior.

2. Es posible una representación alternativa si la generalización es disjunta y completa—es decir, si no hay ninguna entidad miembro de dos conjuntos de entidades de nivel inferior directamente por debajo de un conjunto de entidades de nivel superior, y si todas las entidades del conjunto de entidades de nivel superior también pertenece a uno de los conjuntos de entidades de nivel inferior. En este caso no se crea un esquema para el conjunto de entidades de nivel superior. En vez de eso, para cada conjunto de entidades de nivel inferior se crea un esquema que incluye un atributo por cada atributo de ese conjunto de entidades más un atributo por cada atributo del conjunto de entidades de nivel superior, se tienen dos esquemas:

- empleado = (id\_persona, nombre, calle, ciudad, sueldo)

- cliente = (id\_persona, nombre, calle, ciudad, calificación\_crediticia)

Estos dos esquemas tiene id\_persona, que es el atributo de clave primaria del conjunto de entidades de nivel superior persona, como clave primaria.

Un inconveniente del segundo método es la definición de las restricciones de clave externa. Para ilustrar el problema, supóngase que se tiene un conjunto de relaciones R que implica al conjunto de entidades persona. Con el primer método, al crear un esquema de relación R a partir del conjunto de relaciones, también se define una restricción de clave externa par R, que hace referencia al esquema persona. Desafortunadamente, con el segundo método no se tiene una única relación a la que pueda hacer referencia la restricción de clave externa de R. Para evitar este problema, hay que crear un esquema de relación persona que contenga, al menos, los atributos de clave primaria de la entidad persona. Si se usara el segundo método para una generalización solapada, algunos valores se almacenarían varias veces, de manera innecesaria. Por ejemplo, si una persona es a la vez empleado y cliente, los valores de calle y de ciudad se almacenarían dos veces. Si la generalización no fuera completa—es decir, si alguna persona no fuera ni empleado ni cliente—entonces haría falta un esquema para representar a esas personas.

#### **6.9.6 Representación de la agregación**

El esquema del conjunto de relaciones dirige entre la agregación de trabaja\_en y el conjunto de entidades director incluye un atributo para cada atributo de las claves primarias del conjunto de entidades director y del conjunto de relaciones trabaja\_en. También incluye un atributo para los atributos descriptivos, si los hay, del conjunto de relaciones dirige. Por tanto, se transforman los conjuntos de relaciones y de entidades de la entidad agregada siguiendo las reglas que se han definido anteriormente. Las reglas que se han visto anteriormente para la creación de restricciones de clave primaria y de clave externa para los conjuntos de relaciones se pueden aplicar también a los conjuntos de relaciones que incluyen agregación, tratando la agregación como cualquier otra entidad. La clave primaria de la agregación es la clave primaria del conjunto de relaciones que la define. No hace falta ninguna relación más para que represente la agregación; en vez de eso, se usa la relación creada a partir de la relación definidora.

#### **6.9.7 Esquemas relacionales para la entidad bancaria**

El conjunto correspondiente de esquemas de relación, generado mediante las técnicas ya descritas en este apartado, se muestra a continuación. La clave primaria de cada esquema de relación se denota mediante el subrayado.

- Esquemas derivados de entidades fuertes:
  - sucursal = (nombre\_sucursal, ciudad\_sucursal, activos)
  - cliente = (id\_cliente, nombre\_cliente, calle\_cliente, ciudad\_cliente)
  - préstamo = (número\_préstamo, importe)
  - cuenta = (número\_cuenta, saldo)
  - empleado = (id\_empleado, nombre\_empleado, número\_teléfono, fecha\_contratación)
- Esquemas derivados de atributos multivalorados (no se representan los atributos derivados). Se definen en una vista o en una función definida especialmente:
  - nombre\_subordinado = (id\_empleado, nombre\_subordinado)
- Esquemas derivados de conjuntos de relaciones que implican a conjuntos de entidades fuertes:
  - sucursal\_cuenta = (número\_cuenta, nombre\_sucursal)
  - sucursal\_préstamo = (número\_préstamo, nombre\_sucursal)
  - prestatario = (id\_cliente, número\_préstamo)
  - impositor = (id\_cliente, número\_cuenta)
  - asesor = (id\_cliente, id\_empleado, tipo)
  - trabaja\_para = (id\_empleado\_trabajador, id\_empleado\_jefe)
- Esquemas derivados de conjuntos de entidades débiles
  - pago = (número\_préstamo, número\_pago, fecha\_pago, importe)
- Esquemas derivados de relaciones ES
  - cuenta\_ahorro = (número\_cuenta, tasa\_interés)
  - cuenta\_corriente = (número\_cuenta, importe\_descubierto)

6.10 Otros aspectos del diseño de bases de datos La explicación sobre el diseño de esquemas dada en este capítulo puede crear la falsa impresión de que el diseño de esquemas es el único componente del diseño de bases de datos. En realidad, hay otras consideraciones que se tratarán con más profundidad en capítulos posteriores y que se describirán brevemente a continuación.

### 6.10.1 Restricciones de datos y diseño de bases de datos relacionales

Se ha visto gran variedad de restricciones de datos que pueden expresarse mediante SQL, como las restricciones de clave primaria, las de clave externa, las restricciones check, los asertos y los disparadores. Las restricciones tienen varios propósitos. El más evidente es la automatización de la conservación de la consistencia. Al expresar las restricciones en el lenguaje de definición de datos de SQL, el diseñador puede garantizar que el propio sistema de bases de datos haga que se cumplan las restricciones. Esto es más digno de confianza que dejar que cada programa haga cumplir las restricciones por su cuenta. También ofrece una ubicación central para la actualización de las restricciones y la adición de otras nuevas. Otra ventaja de definir explícitamente las restricciones es que algunas restricciones resultan especialmente útiles en el diseño de esquemas de bases de datos relacionales. Las restricciones de datos también resultan útiles para determinar la estructura física de los datos. Puede resultar útil almacenar físicamente próximos en el disco los datos que están estrechamente relacionados entre sí, de modo que se mejore la eficiencia del acceso a disco. Algunas estructuras de índices funcionan mejor cuando el índice se crea sobre una clave primaria. La aplicación de las restricciones se lleva a cabo a un precio potencialmente alto en rendimiento cada vez que se actualiza la base de datos. En cada actualización el sistema debe comprobar todas las restricciones y rechazar las actualizaciones que no las cumplen o ejecutar los disparadores

correspondientes. La importancia de la penalización en rendimiento no sólo depende de la frecuencia de actualización, sino también del modo en que se haya diseñado la base de datos.

### **6.10.2 Requisitos de uso: consultas y rendimiento**

El rendimiento de los sistemas de bases de datos es un aspecto crítico de la mayor parte de los sistemas informáticos empresariales. El rendimiento no sólo tiene que ver con el uso eficiente del hardware de cálculo y de almacenamiento que se usa, sino también con la eficiencia de las personas que interactúan con el sistema y de los procesos que dependen de los datos de las bases de datos. Existen dos métricas principales para el rendimiento.

- Productividad—el número de consultas o actualizaciones (a menudo denominadas transacciones) que pueden procesarse en promedio por unidad de tiempo.
- Tiempo de respuesta—el tiempo que tarda una sola transacción desde el comienzo hasta el final en promedio o en el peor de los casos.

Los sistemas que procesan gran número de transacciones agrupadas por lotes se centran en tener una productividad elevada. Los sistemas que interactúan con personas y los sistemas de tiempo crítico suelen centrarse en el tiempo de respuesta. Estas dos métricas no son equivalentes. La productividad elevada se consigue mediante un elevado uso de los componentes del sistema. Ello puede dar lugar a que algunas transacciones se pospongan hasta el momento en que puedan ejecutarse con mayor eficiencia. Las transacciones pospuestas sufren un bajo tiempo de respuesta. Históricamente, la mayor parte de los sistemas de bases de datos comerciales se han centrado en la productividad; no obstante, gran variedad de aplicaciones, incluidas las aplicaciones basadas en la Web y los sistemas informáticos para telecomunicaciones necesitan un buen tiempo de respuesta promedio y una cota razonable para el peor tiempo de respuesta que pueden ofrecer. La comprensión de los tipos de consultas que se espera que sean más frecuentes ayuda al proceso de diseño. Las consultas que implican reuniones necesitan evaluar más recursos que las que no las implican. A veces, cuando se necesita una reunión, puede que el administrador de la base de datos decida crear un índice que facilite la evaluación de la reunión. Para las consultas—tanto si está implicada una reunión como si no—se pueden crear índices para acelerar la evaluación de los predicados de selección (la cláusula where de SQL) que sea posible que aparezcan. Otro aspecto de las consultas que afecta a la elección de índices es la proporción relativa de operaciones de actualización y de lectura. Aunque los índices pueden acelerar las consultas, también ralentiza las actualizaciones, que se ven obligadas a realizar un trabajo adicional para mantener la exactitud de los índices.

### **6.10.3 Requisitos de autorización**

Las restricciones de autorización también afectan al diseño de las bases de datos, ya que SQL permite que se autorice el acceso a los usuarios en función de los componentes del diseño lógico de la base de datos. Puede que haga falta descomponer un esquema de relación en dos o más esquemas para facilitar la concesión de derechos de acceso en SQL. 6.10.4 Flujos de datos y de trabajo Las aplicaciones de bases de datos suelen formar parte de una aplicación empresarial de mayor tamaño que no sólo interactúa con el sistema de bases de datos, sino también con diferentes aplicaciones especializadas. Por ejemplo, en una compañía manufacturera, puede que un sistema de diseño asistido por computadora (computer-aided design, CAD) ayude al diseño de nuevos productos. Puede que el sistema CAD extraiga datos de la base de datos mediante instrucciones de SQL, procese internamente los datos, quizás interactuando con un diseñador de productos, y luego actualice la base de datos. Durante este proceso el control de los datos puede

pasar a manos de varios diseñadores de productos y de otras personas. Como ejemplo adicional, considérese un informe de gastos de viaje. Lo crea un empleado que vuelve de un viaje de negocios (posiblemente mediante un paquete de software especial) y luego se envía al jefe de ese empleado, quizás a otros jefes de niveles superiores y, finalmente, al departamento de contabilidad para su pago (momento en el que interactúa con los sistemas informáticos de contabilidad de la empresa). El término flujo de trabajo hace referencia a la combinación de datos y de tareas implicados en procesos como los de los ejemplos anteriores. Los flujos de trabajo interactúan con el sistema de bases de datos cuando se mueven entre los usuarios y los usuarios llevan a cabo sus tareas en el flujo de trabajo. Además de los datos sobre los que opera el flujo de trabajo, puede que la base de datos almacene datos sobre el propio flujo de trabajo, incluidas las tareas que lo conforman y la manera en que se han de hacer llegar a los usuarios. Por tanto, los flujos de trabajo especifican una serie de consultas y de actualizaciones de la base de datos que pueden tenerse en cuenta como parte del proceso de diseño de la base de datos. En otras palabras, el modelado de la empresa no sólo exige la comprensión de la semántica de los datos, sino también la de los procesos comerciales que los usan.

#### **6.10.5 Otros problemas del diseño de bases de datos**

El diseño de bases de datos no suele ser una actividad que se pueda dar por acabada. Las necesidades de las organizaciones evolucionan continuamente, y los datos que necesitan almacenar también evolucionan en consonancia. Durante las fases iniciales del diseño de la base de datos, o durante el desarrollo de las aplicaciones, puede que el diseñador de la base de datos se dé cuenta de que hacen falta cambios en el nivel del esquema conceptual, lógico o físico. Los cambios del esquema pueden afectar a todos los aspectos de la aplicación de bases de datos. Un buen diseño de bases de datos se anticipa a las necesidades futuras de la organización y el diseño se lleva a cabo de manera que se necesiten modificaciones mínimas a medida que evolucionen las necesidades de la organización. Es importante distinguir entre las restricciones fundamentales y las que se anticipa que puedan cambiar. Por ejemplo, la restricción de que cada identificador de cliente identifique a un solo cliente es fundamental. Por otro lado, el banco puede tener la norma de que cada cliente sólo pueda tener una cuenta, lo que puede cambiar más adelante. Un diseño de la base de datos que sólo permita una cuenta por cliente necesitaría cambios importantes si el banco cambiase su normativa. Esos cambios no deben exigir modificaciones importantes en el diseño de la base de datos. Además, es probable que la empresa a la que sirve la base de datos interactúe con otras empresas y, por tanto, puede que tengan que interactuar varias bases de datos. La conversión de los datos entre esquemas diferentes es un problema importante en las aplicaciones del mundo real. Se han propuesto diferentes soluciones para este problema. El modelo de datos XML, que se estudia en el Capítulo 10, se usa mucho para representar los datos cuando se intercambian entre diferentes aplicaciones. Finalmente, merece la pena destacar que el diseño de bases de datos es una actividad orientada a los seres humanos en dos sentidos: los usuarios finales son personas (aunque se sitúe alguna aplicación entre la base de datos y los usuarios finales) y el diseñador de la base de datos debe interactuar intensamente con los expertos en el dominio de la aplicación para comprender los requisitos de datos de la aplicación. Todas las personas involucradas con los datos tienen necesidades y preferencias que se deben tener en cuenta para que el diseño y la implantación de la base de datos tengan éxito en la empresa.

#### **6.11 El lenguaje de modelado unificado UML\*\***

Los diagramas entidad-relación ayudan a modelar el componente de representación de datos de los sistemas de software. La representación de datos, sin embargo, sólo forma

parte del diseño global del sistema. Otros componentes son los modelos de interacción del usuario con el sistema, la especificación de los módulos funcionales del sistema y su interacción, etc. El lenguaje de modelado unificado (Unified Modeling Language, UML) es una norma desarrollada bajo los auspicios del Grupo de Administración de Objetos (Object Management Group, OMG) para la creación de especificaciones de diferentes componentes de los sistemas de software. Algunas de las partes de UML son:

- **Diagramas de clase.** Los diagramas de clase son parecidos a los diagramas E-R. Más adelante en este apartado se mostrarán algunas características de los diagramas de clase y del modo en que se relacionan con los diagramas E-R
- **Diagramas de caso de uso.** Los diagramas de caso de uso muestran la interacción entre los usuarios y el sistema, en especial los pasos de las tareas que llevan a cabo los usuarios (como retirar dinero o matricularse en una asignatura).
- **Diagramas de actividad.** Los diagramas de actividad describen el flujo de tareas entre los diferentes componentes del sistema.
- **Diagramas de implementación.** Los diagramas de implementación muestran los componentes del sistema y sus interconexiones, tanto en el nivel de los componentes de software como en el de hardware.

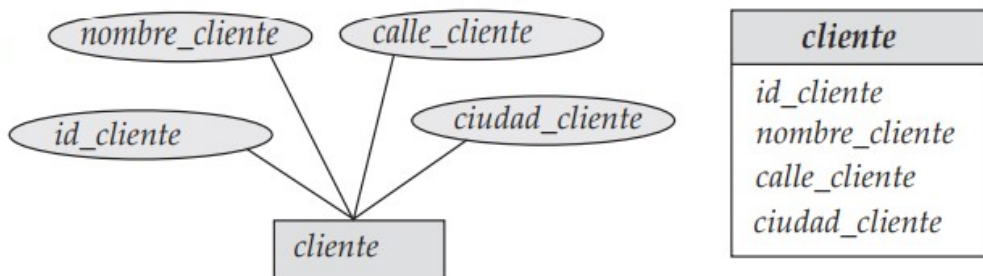
UML muestra los conjuntos de entidades como cuadros y, a diferencia de E-R, muestra los atributos dentro de los cuadros en lugar de como elipses separadas. UML modela realmente objetos, mientras que E-R modela entidades. Los objetos son como entidades y tienen atributos, pero también proporcionan un conjunto de funciones (denominadas métodos) que se pueden invocar para calcular valores con base en los atributos de los objetos, o para actualizar el propio objeto. Los diagramas de clases pueden describir métodos además de atributos.

Los conjuntos de relaciones binarias se representan en UML dibujando simplemente una línea que conecte los conjuntos de entidades. El nombre del conjunto de relaciones se escribe junto a la línea. También se puede especificar el rol que desempeña cada conjunto de entidades en un conjunto de relaciones escribiendo el nombre del rol sobre la línea, junto al conjunto de entidades. De manera alternativa, se puede escribir el nombre del conjunto de relaciones en un recuadro, junto con los atributos del conjunto de relaciones, y conectar el recuadro con una línea discontinua a la línea que describe el conjunto de relaciones. Este recuadro se puede tratar entonces como un conjunto de entidades, de la misma forma que la agregación en los diagramas E-R, y puede participar en relaciones con otros conjuntos de entidades. Desde la versión 1.3 de UML, UML soporta las relaciones no binarias, usando la misma notación de rombos usada en los diagramas E-R. Las relaciones no binarias no se podían representar directamente en versiones anteriores de UML—había que convertirlas en relaciones binarias. Las restricciones de cardinalidad se especifican en UML de la misma forma que en los diagramas E-R, de la forma *i..s*, donde *i* denota el número mínimo y *s* el máximo de relaciones en que puede participar cada entidad. Sin embargo, hay que ser consciente de que la ubicación de las restricciones es exactamente la contraria que en los diagramas E-R. La restricción *0..\** en el lado E2 y *0..1* en el lado E1 significa que cada entidad E2 puede participar, a lo sumo, en una relación, mientras que cada entidad E1 puede participar en varias relaciones; en otras palabras, la relación es varios a uno de E2 a E1. Los valores aislados como *1* o *\** se pueden escribir en los arcos; el valor *1* sobre un arco se trata como equivalente de *1..1*, mientras que *\** es equivalente a *0..\**. La generalización y la especialización se representan en UML conectando conjuntos de entidades mediante una línea con un triángulo al final correspondiente al conjunto de entidades más general. Por ejemplo, el conjunto de entidades persona es una generalización de cliente y de

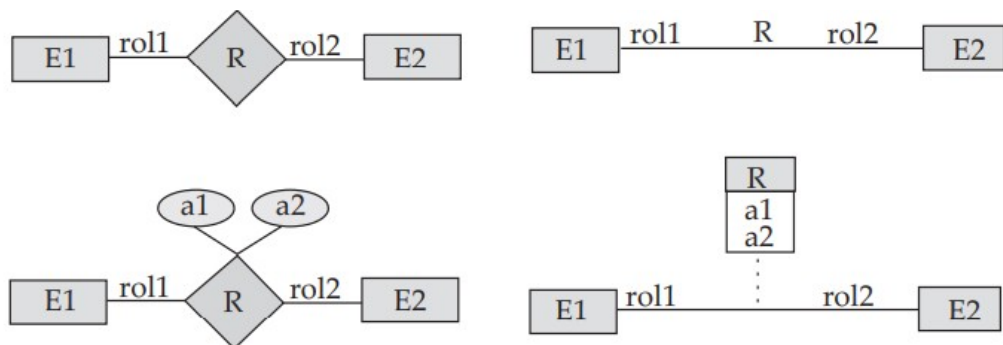
empleado. Los diagramas UML también pueden representar explícitamente las restricciones de la condición de disjunción y de solapamiento de las generalizaciones. Una generalización solapada permite que una persona sea tanto cliente como empleado. Los diagramas de clases de UML incluyen otras notaciones que no se corresponden con las notaciones E-R que se han visto. Por ejemplo, una línea entre dos conjuntos de entidades con un rombo en un extremo especifica que la entidad en el extremo del rombo contiene a la otra (la inclusión se denomina “agregación” en la terminología de UML). Por ejemplo, la entidad vehículo puede contener una entidad motor. Los diagramas de clases de UML también ofrecen notaciones para representar características del lenguaje orientadas a objetos, como las anotaciones públicas o privadas de los miembros de la clase e interfaces (esto debe resultarle familiar a cualquiera que conozca los lenguajes Java o C#). Véanse las referencias en las notas bibliográficas para obtener más información sobre los diagramas de clases de UML.

### Equivalencias de E-R a UML

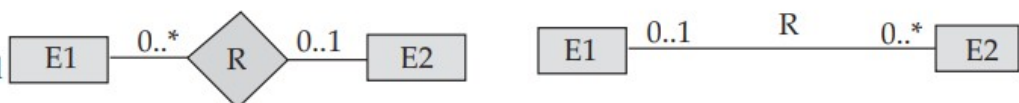
#### 1. Conjuntos de entidades y atributos



#### 2. Relaciones



#### 3. Restricciones de cardinalidad



#### 4. Generalización y especialización

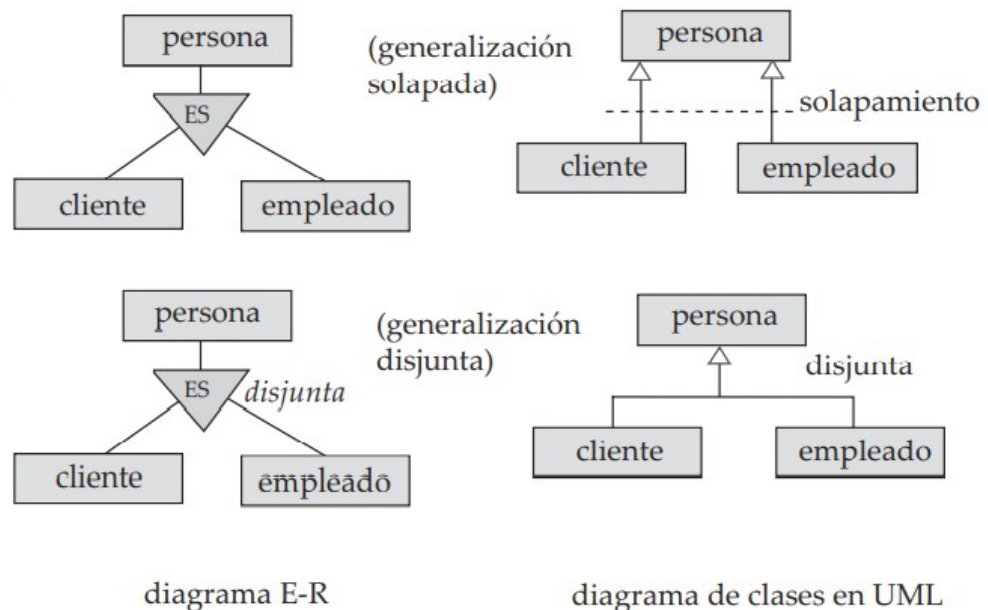


diagrama E-R

diagrama de clases en UML

### 6.12 Resumen

- El diseño de bases de datos supone principalmente el diseño del esquema de la base de datos. El modelo de datos entidad-relación (E-R) es un modelo de datos muy usado para el diseño de bases de datos. Ofrece una representación gráfica adecuada para ver los datos, las relaciones y las restricciones.
- El modelo está pensado principalmente para el proceso de diseño de la base de datos. Se desarrolló para facilitar el diseño de bases de datos al permitir la especificación de un esquema de la empresa. Este esquema representa la estructura lógica general de la base de datos. Esta estructura general se puede expresar gráficamente mediante un diagrama E-R.
- Una entidad es un objeto que existe en el mundo real y es distinguible de otros objetos. Esa distinción se expresa asociando a cada entidad un conjunto de atributos que describen el objeto
- Una relación es una asociación entre diferentes entidades. Un conjunto de relaciones es una colección de entidades del mismo tipo, y un conjunto de entidades es una colección de entidades del mismo tipo. Una superclave de un conjunto de entidades es un conjunto de uno o más atributos que, tomados en conjunto, permiten identificar unívocamente una entidad del conjunto de entidades. Se elige una superclave mínima para cada conjunto de entidades de entre sus superclaves; la superclave mínima se denomina clave primaria del conjunto de entidades. Análogamente, un conjunto de relaciones es un conjunto de uno o más atributos que, tomados en conjunto, permiten identificar unívocamente una relación del conjunto de relaciones. De igual forma se elige una superclave mínima para cada conjunto de relaciones de entre todas sus superclaves; ésta es la clave primaria del conjunto de relaciones.
- La correspondencia de cardinalidades expresa el número de entidades con las que otra entidad se puede asociar mediante un conjunto de relaciones.
- Un conjunto de entidades que no tiene suficientes atributos para formar una clave primaria se denomina conjunto de entidades débiles. Un conjunto de entidades que tiene una clave primaria se denomina conjunto de entidades fuertes
- La especialización y la generalización definen una relación de inclusión entre un conjunto de entidades de nivel superior y uno o más conjuntos de entidades de nivel inferior. La especialización es el resultado de tomar un subconjunto de un conjunto de entidades de nivel superior para formar un conjunto de entidades de inferior. La generalización es el resultado de tomar la unión de dos o más conjuntos

disjuntos de entidades (de nivel inferior) para producir un conjunto de entidades de nivel superior. Los atributos de los conjuntos de entidades de nivel superior los heredan los conjuntos de entidades de nivel inferior.

- La agregación es una abstracción en la que los conjuntos de relaciones (junto con sus conjuntos de entidades asociados) se tratan como conjuntos de entidades de nivel superior y puede participar en las relaciones.
- Las diferentes características del modelo E-R ofrecen al diseñador de bases de datos numerosas opciones a la hora de representar lo mejor posible la empresa que se modela. Los conceptos y los objetos pueden, en ciertos casos, representarse mediante entidades, relaciones o atributos. Ciertos aspectos de la estructura global de la empresa se pueden describir mejor usando los conjuntos de entidades débiles, la generalización, la especialización o la agregación. A menudo, el diseñador debe sopesar las ventajas de un modelo simple y compacto frente a las de otro más preciso pero más complejo
- El diseño de una base de datos especificado en un diagrama E-R se puede representar mediante un conjunto de esquemas de relación. Para cada conjunto de entidades y para cada conjunto de relaciones de la base de datos hay un solo esquema de relación al que se le asigna el nombre del conjunto de entidades o de relaciones correspondiente. Esto forma la base para la obtención del diseño de la base de datos relacional a partir del E-R.
- El lenguaje de modelado unificado (UML) ofrece un medio gráfico de modelar los diferentes componentes de los sistemas de software. El componente diagrama de clases de UML se basa en los diagramas E-R. Sin embargo, hay algunas diferencias entre los dos que se deben tener presentes.