

## MODELOS DE DATOS

**U**n **modelo de datos** es una colección de herramientas conceptuales para la descripción de datos, relaciones entre datos, semántica de los datos y restricciones de consistencia. En esta parte se estudiarán dos modelos de datos —el modelo entidad-relación y el modelo relacional.

El modelo entidad-relación (E-R) es un modelo de datos de alto nivel. Está basado en una percepción de un mundo real que consiste en una colección de objetos básicos, denominados *entidades*, y de *relaciones* entre estos objetos.

El modelo relaciona es un modelo de menor nivel. Usa una colección de tablas para representar tanto los datos como las relaciones entre los datos. Su simplicidad conceptual ha conducido a su adopción general; actualmente, una vasta mayoría de productos de bases de datos se basan en el modelo relacional. Los diseñadores formulan generalmente el diseño del esquema de la base de datos modelando primero los datos en alto nivel, usando el modelo E-R, y después traduciéndolo al modelo relacional.

Se estudiarán otros modelos de datos más tarde en este libro. El modelo de datos orientado a objetos, por ejemplo, extiende la representación de entidades añadiendo nociones de encapsulación, métodos (funciones) e identidad de objeto. El modelo de datos relacional orientado a objetos combina características del modelo de datos orientado a objetos y del modelo de datos relacional. Los Capítulos 8 y 9 tratan respectivamente estos dos modelos de datos.

## MODELO ENTIDAD-RELACIÓN

EL modelo de datos **entidad-relación (E-R)** está basado en una percepción del mundo real consistente en objetos básicos llamados *entidades* y de *relaciones* entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un *esquema de la empresa* que representa la estructura lógica completa de una base de datos. El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo yace en la representación del significado de los datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R.

### 2.1. CONCEPTOS BÁSICOS

Hay tres nociones básicas que emplea el modelo de datos E-R: conjuntos de entidades, conjuntos de relaciones y atributos.

#### 2.1.1. Conjuntos de entidades

Una **entidad** es una «cosa» u «objeto» en el mundo real que es distinguible de todos los demás objetos. Por ejemplo, cada persona en un desarrollo es una entidad. Una entidad tiene un conjunto de propiedades, y los valores para algún conjunto de propiedades pueden identificar una entidad de forma unívoca. Por ejemplo, el D.N.I. 67.789.901 identifica unívocamente una persona particular en la empresa. Análogamente, se puede pensar en los préstamos bancarios como entidades, y un número de préstamo P-15 en la sucursal de Castellana identifica unívocamente una entidad de préstamo. Una entidad puede ser concreta, como una persona o un libro, o puede ser abstracta, como un préstamo, unas vacaciones o un concepto.

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 pueden definir como el conjunto de entidades *cliente*. Análogamente, el conjunto de entidades *préstamo* podría representar el conjunto de todos los préstamos concedidos por un banco particular. Las entidades individuales que constituyen un conjunto se llaman la *extensión* del conjunto de entidades. Así, todos los clientes de un banco son la extensión del conjunto de entidades *cliente*.

Los conjuntos de entidades no son necesariamente disjuntos. Por ejemplo, es posible definir el conjunto de entidades de todos los empleados de un banco (*empleado*) y el conjunto de entidades de todos los clientes del banco (*cliente*). Una entidad *persona* puede ser una entidad *empleado*, una entidad *cliente*, ambas cosas, o ninguna.

Una entidad se representa mediante un conjunto de **atributos**. Los atributos describen propiedades 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 similar concerniente a cada entidad del conjunto de entidades; sin embargo, cada entidad puede tener su propio valor para cada atributo. Posibles atributos del conjunto de entidades *cliente* son *id-cliente*, *nombre-cliente*, *calle-cliente* y *ciudad-cliente*. En la vida real, habría más atributos, tales como el número de la calle, el número del portal, la provincia, el código postal, y la comunidad autónoma, pero no se incluyen en el ejemplo simple. Posibles atributos del conjunto de entidades *préstamo* son *número-préstamo* e *importe*.

Cada entidad tiene un **valor** para cada uno de sus atributos. Por ejemplo, una entidad *cliente* en concreto puede tener el valor 32.112.312 para *id-cliente*, el valor Santos para *nombre-cliente*, el valor Mayor para *calle-cliente* y el valor Peguerinos para *ciudad-cliente*.

El atributo *id-cliente* se usa para identificar unívocamente a los clientes, dado que no hay más de un cliente con el mismo nombre, calle y ciudad. En los Estados Unidos, muchas empresas encuentran conveniente usar el número *seguridad-social* de una persona<sup>1</sup> como un

<sup>1</sup> En España se asigna a cada persona del país un número único, denominado número del documento nacional de identidad (D.N.I.) para identificarla unívocamente. Se supone que cada persona tiene un único D.N.I., y no hay dos personas con el mismo D.N.I.

atributo cuyo valor identifica unívocamente a la persona. En general la empresa tendría que crear y asignar un identificador a cada cliente.

Para cada atributo hay un conjunto de valores permitidos, llamados el **dominio**, o el **conjunto de valores**, de ese atributo. El dominio del atributo *nombre-cliente* podría ser el conjunto de todas las cadenas de texto de una cierta longitud. Análogamente, el dominio del atributo *número-préstamo* podría ser el conjunto de todas las cadenas de la forma «P-*n*», donde *n* es un entero positivo.

Una base de datos incluye así una colección de conjuntos de entidades, cada una de las cuales contiene un número de entidades del mismo tipo. En la Figura 2.1 se muestra parte de una base de datos de un banco que consta de dos conjuntos de entidades, *cliente* y *préstamo*.

Formalmente, un atributo de un conjunto de entidades es una función que asigna al conjunto de entidades un dominio. Como un conjunto de entidades puede tener diferentes atributos, cada entidad se puede describir como un conjunto de pares (atributo,valor), un par para cada atributo del conjunto de entidades. Por ejemplo, una entidad concreta *cliente* se puede describir mediante el conjunto {(id-cliente, 67.789.901), (nombre-cliente, López), (calle-cliente, Mayor), (ciudad-cliente, Peguerinos)}, queriendo decir que la entidad describe una persona llamada López que tiene D.N.I. número 67.789.901, y reside en la calle Mayor en Peguerinos. Se puede ver, en este punto, que existe una integración del esquema abstracto con el desarrollo real de la empresa que se está modelando. Los valores de los atributos que describen una entidad constituirán una porción significativa de los datos almacenados en la base de datos.

Un atributo, 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 están divididos en subpartes. Los

atributos compuestos, en cambio, se pueden dividir en subpartes (es decir, en otros atributos). Por ejemplo, *nombre-cliente* podría 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, a algún componente del atributo. Se podrían haber sustituido los atributos del conjunto de entidades *cliente*, *calle-cliente* y *ciudad-cliente*, por el atributo compuesto *dirección-cliente*, con los atributos *calle*, *ciudad*, *provincia*, y *código-postal*<sup>2</sup>. Los atributos compuestos ayudan a agrupar los atributos relacionados, haciendo los modelos más claros.

Nótese también que un atributo compuesto puede aparecer como una jerarquía. Volviendo al ejemplo del atributo compuesto *dirección-cliente*, su componente *calle* puede ser a su vez dividido en *número-calle*, *nombre-calle* y *piso*. Estos ejemplos de atributos compuestos para el conjunto de entidades *cliente* se representa en la Figura 2.2.

- Atributos **monovalorados** y **multivalorados**. Los atributos que se han especificado en los ejemplos tienen todos un valor sólo para una entidad concreta. Por ejemplo, el atributo *número-préstamo* para una entidad préstamo específico, referencia a un único número de préstamo. Tales atributos se llaman **monovalorados**. Puede haber ocasiones en las que un atributo tiene un conjunto de valores para una entidad específica. Considérese un conjunto de entidades *empleado* con el atributo *número-teléfono*. Cualquier empleado particular puede tener cero, uno o más números de teléfono. Este tipo de atributo se llama **multivalorado**. En ellos, se pueden colocar apropiadamente límites inferior y superior en el número de valores en el atributo **multivalorado**. Como otro ejemplo, un atributo *nombre-subordinado* del conjunto de entidades *empleado*

<sup>2</sup> Se asume el formato de *calle-cliente* y *dirección* usado en España, que incluye un código postal numérico llamado «código postal».

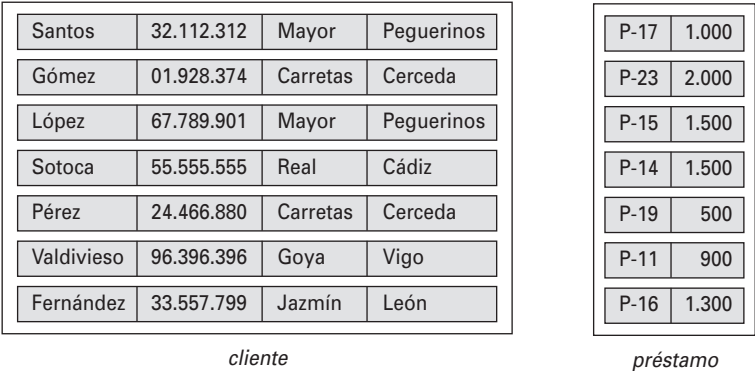


FIGURA 2.1. Conjunto de entidades *cliente* y *préstamo*.

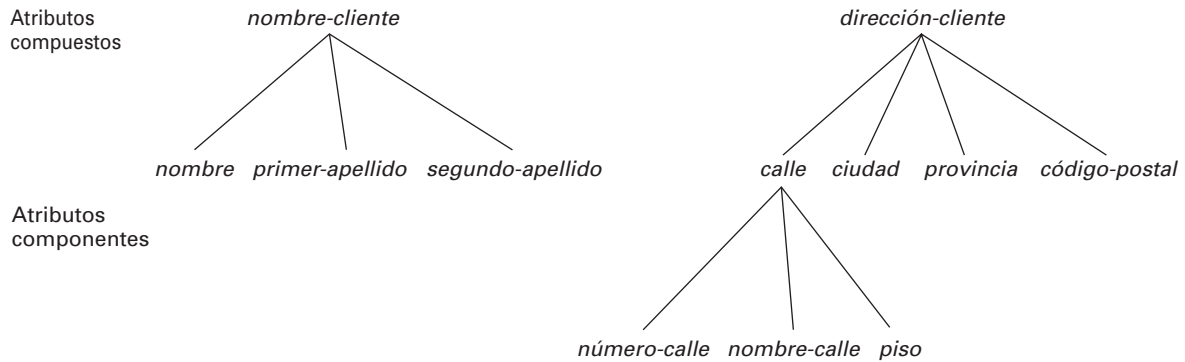


FIGURA 2.2. Atributos compuestos *nombre-cliente* y *dirección-cliente*.

sería multivalorado, ya que un empleado en concreto podría tener cero, uno o más subordinados.

Cuando sea apropiado se pueden establecer límites superior e inferior en el número de valores de un atributo multivalorado. Por ejemplo, un banco puede limitar el número de números de teléfono almacenados para un único cliente a dos. Colocando límites en este caso, se expresa que el atributo *número-teléfono* del conjunto de entidades *cliente* puede tener entre cero y dos valores.

- **Atributos derivados.** El valor para este tipo de atributo se puede derivar de los valores de otros atributos o entidades relacionados. Por ejemplo, sea el conjunto de entidades *cliente* que tiene un atributo *préstamos* que representa cuántos préstamos tiene un cliente en el banco. Ese atributo se puede derivar contando el número de entidades *préstamo* asociadas con ese *cliente*.

Como otro ejemplo, considérese que el conjunto de entidades *empleado* tiene un atributo *edad*, que indica la edad del cliente. Si el conjunto de entidades *cliente* tiene también un atributo *fecha-de-nacimiento*, se puede calcular *edad* a partir de *fecha-de-nacimiento* y de la fecha actual. Así, *edad* es un atributo derivado. En este caso, *fecha-de-nacimiento* y *antigüedad* pueden serlo, ya que representan el primer día en que el empleado comenzó a trabajar para el banco y el tiempo total que el empleado lleva trabajando para el banco, respectivamente. El valor de *antigüedad* se puede derivar del valor de *fecha-comienzo* y de la fecha actual. En este caso, *fecha-comienzo* se puede conocer como atributo *base* o atributo *almacenado*. El valor de un atributo derivado no se almacena, sino que se calcula cuando sea necesario.

Un atributo toma un valor **nulo** cuando una entidad no tiene un valor para un atributo. El valor *nulo* también puede indicar «no aplicable», es decir, que el valor no existe para la entidad. Por ejemplo, una persona puede no tener segundo nombre de pila. *Nulo* puede también designar que el valor de un atributo es desconocido. Un valor desconocido puede ser, bien *perdido* (el

valor existe pero no se tiene esa información) o *desconocido* (no se conoce si el valor existe realmente o no).

Por ejemplo, si el valor *nombre* para un *cliente* particular es *nulo*, se asume que el valor es perdido, ya que cada cliente debe tener un nombre. Un valor *nulo* para el atributo *piso* podría significar que la dirección no incluye un piso (no aplicable), que existe piso pero no se conoce cuál es (perdido), o que no se sabe si el piso forma parte o no de la dirección del cliente (desconocido).

Una base de datos para una empresa bancaria puede incluir diferentes conjuntos de entidades. Por ejemplo, además del mantenimiento de clientes y préstamos, el banco también proporciona cuentas, que se representan mediante el conjunto de entidades *cuenta* con atributos *número-cuenta* y *saldo*. También, si el banco tiene un número de sucursales diferentes, se puede mantener información acerca de todas las sucursales del banco. Cada conjunto de entidades *sucursal* se describe mediante los atributos *nombre-sucursal*, *ciudad-sucursal* y *activo*.

### 2.1.2. Conjuntos de relaciones

Una **relación** es una asociación entre diferentes entidades. Por ejemplo, se puede definir una relación que asocie al cliente López con el préstamo P-15. Esta relación especifica que López es un cliente con el préstamo número P-15.

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.

Considérense las dos entidades *cliente* y *préstamo* de la Figura 2.1. Se define el conjunto de relaciones *prestatario* para denotar la asociación entre clientes y préstamos bancarios que los clientes tengan. Esta asociación se describe en la Figura 2.3.

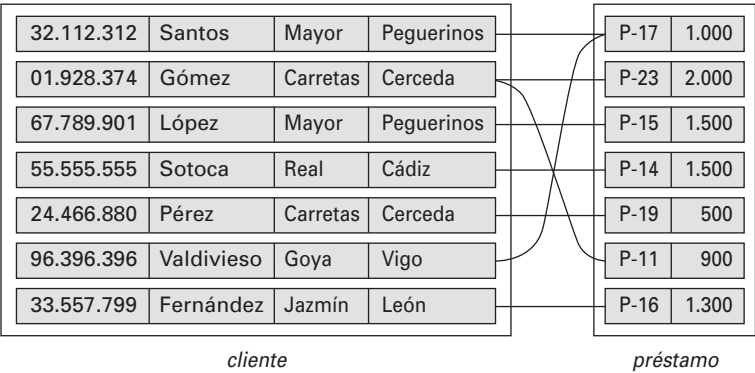


FIGURA 2.3. Conjunto de relaciones *prestatario*.

Como otro ejemplo, considérense los dos conjuntos de entidades *préstamo* y *sucursal*. Se puede definir el conjunto de relaciones *sucursal-préstamo* para denotar la asociación entre un préstamo y la sucursal en que se mantiene ese préstamo.

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 relación** en un esquema E-R representa que existe una asociación entre las entidades denominadas en la empresa del mundo real que se modela. Como ilustración, el *cliente* individual López, que tiene D.N.I. 67.789.901, y la entidad *préstamo* P-15 participan en un ejemplar de relación de *prestatario*. Este ejemplar de relación representa que, en la empresa del mundo real, la persona llamada López cuyo número de D.N.I. es 67.789.901 ha tomado un préstamo que está numerado como P-15.

La función que desempeña una entidad en una relación se llama **papel** de la entidad.

Debido a que los conjuntos de entidades que participan en un conjunto de relaciones son generalmente distintos, los papeles están implícitos y no se especifican normalmente. Sin embargo, son 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 una relación más de una vez con diferentes papeles. En este tipo de conjunto de relaciones, que se llama algunas veces conjunto de relaciones **recursivo**, es necesario hacer explícitos los papeles para especificar cómo participa una entidad en un ejemplar de relación. Por ejemplo, considérese una conjunto de entidades *empleado* que almacena información acerca de todos los empleados del banco. Se puede tener un conjunto de relaciones *trabaja-para* que se modela mediante pares ordenados de entidades *empleado*. El primer empleado de un par toma el papel de *trabajador*, mientras el segundo toma el papel de *jefe*. De esta manera, todas las relaciones *trabaja-para* son pares (trabajador, jefe); los pares (jefe, trabajador) están excluidos.

Una relación puede también tener **atributos descriptivos**. Considérese un conjunto de relaciones *impo-*

*sitor* con conjuntos de entidades *cliente* y *cuenta*. Se podría asociar el atributo *fecha-acceso* a esta relación para especificar la fecha más reciente en que un cliente accedió a una cuenta. La relación *impositor* entre las entidades correspondientes al cliente García y la cuenta C-217 se describen mediante  $\{(fecha-acceso, 23 \text{ mayo } 2002)\}$ , lo que significa que la última vez que García accedió a la cuenta C-217 fue el 23 de mayo de 2002.

Como otro ejemplo de atributos descriptivos para relaciones, supóngase que se tienen los conjuntos de entidades *estudiante* y *asignatura* que participan en una relación *matriculado*. Se podría desear almacenar un atributo descriptivo para *créditos* con la relación, para registrar si el estudiante se ha matriculado de la asignatura para obtener créditos o sólo como oyente.

Un ejemplar de relación en un conjunto de relaciones determinado debe ser identificado unívocamente a partir de sus entidades participantes, sin usar los atributos descriptivos. Para comprender este punto supóngase que deseamos modelar todas las fechas en las que un cliente ha accedido a una cuenta. El atributo monovalorado *fecha-acceso* puede almacenar sólo una única fecha de acceso. No se pueden representar varias fechas de acceso por varios ejemplares de relación entre el mismo cliente y cuenta, ya que los ejemplares de relación no estarían identificados unívocamente por las entidades participantes. La forma correcta de manejar este caso es crear un atributo multivalorado *fechas-acceso* que pueda almacenar todas las fechas de acceso.

Sin embargo, puede haber más de un conjunto de relaciones que involucren los mismos conjuntos de entidades. En nuestro ejemplo los conjuntos de entidades *cliente* y *préstamo* participan en el conjunto de relaciones *prestatario*. Además, supóngase que cada préstamo deba tener otro cliente que sirva como avalista para el préstamo. Entonces los conjuntos de entidades *cliente* y *préstamo* pueden participar en otro conjunto de relaciones: *avalista*.

Los conjuntos de relaciones *prestatario* y *sucursal-préstamo* proporcionan un ejemplo de un conjunto de relaciones **binario**, es decir, uno que implica dos conjuntos de entidades. La mayoría de los conjuntos de relaciones en un sistema de bases de datos son binarios.



Ocasionalmente, sin embargo, los conjuntos de relaciones implican más de dos conjuntos de entidades.

Por ejemplo, considérense los conjuntos de entidades *empleado*, *sucursal* y *trabajo*. Ejemplos de las entidades *trabajo* podrían ser director, cajero, auditor y otros. Las entidades *trabajo* pueden tener los atributos *puesto* y *nivel*. El conjunto de relaciones *trabaja-en* entre *empleado*, *sucursal* y *trabajo* es un ejemplo de una relación ternaria. Una relación ternaria entre Santos, Navacerrada y director indica que Santos actúa de

director de la sucursal Navacerrada. Santos también podría actuar como auditor de la sucursal Centro, que estaría representado por otra relación. Podría haber otra relación entre Gómez, Centro y cajero, indicando que Gómez actúa como cajero en la sucursal Centro.

El número de conjuntos de entidades que participan en un conjunto de relaciones es también el **grado** del conjunto de relaciones. Un conjunto de relaciones binario tiene grado 2; un conjunto de relaciones ternario tiene grado 3.

## 2.2. RESTRICCIONES

Un esquema de desarrollo E-R puede definir ciertas restricciones a las que los contenidos de la base de datos se deben adaptar. En este apartado se examina la correspondencia de cardinalidades y las restricciones de participación, que son dos de los tipos más importantes de restricciones.

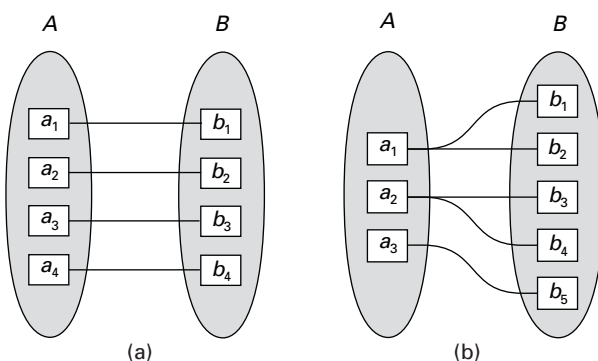
### 2.2.1. Correspondencia de cardinalidades

La **correspondencia de cardinalidades**, o razón de cardinalidad, expresa el número de entidades a las que otra entidad puede estar asociada vía un conjunto de relaciones.

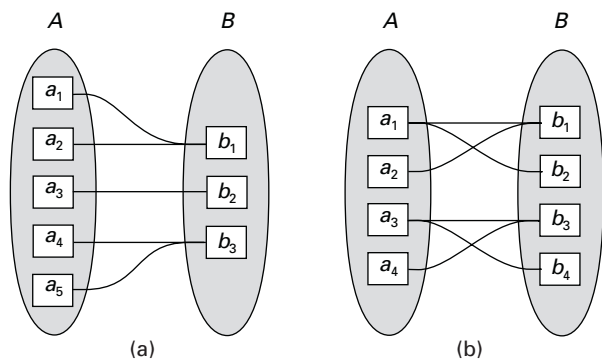
La correspondencia de cardinalidades es la más útil describiendo conjuntos de relaciones binarias, aunque ocasionalmente contribuye a la descripción de conjuntos de relaciones que implican más de dos conjuntos de entidades. Este apartado se centrará en conjuntos de relaciones binarias únicamente.

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:

- **Uno a uno.** Una entidad en  $A$  se asocia con *a lo sumo* una entidad en  $B$ , y una entidad en  $B$  se asocia con *a lo sumo* una entidad en  $A$  (véase la Figura 2.4a).



**FIGURA 2.4.** Correspondencia de cardinalidades. (a) Uno a uno. (b) Uno a varios.



**FIGURA 2.5.** Correspondencia de cardinalidades. (a) Varios a uno. (b) Varios a varios.

- **Uno a varios.** Una entidad en  $A$  se asocia con cualquier número de entidades en  $B$  (ninguna o varias). Una entidad en  $B$ , sin embargo, se puede asociar con *a lo sumo* una entidad en  $A$  (véase la Figura 2.4b).
- **Varios a uno.** Una entidad en  $A$  se asocia con *a lo sumo* una entidad en  $B$ . Una entidad en  $B$ , sin embargo, se puede asociar con cualquier número de entidades (ninguna o varias) en  $A$  (véase la Figura 2.5a).
- **Varios a varios.** Una entidad en  $A$  se asocia con cualquier número de entidades (ninguna o varias) en  $B$ , y una entidad en  $B$  se asocia con cualquier número de entidades (ninguna o varias) en  $A$  (véase la Figura 2.5b).

La correspondencia de cardinalidades apropiada para un conjunto de relaciones particular depende obviamente de la situación del mundo real que el conjunto de relaciones modela.

Como ilustración considérese el conjunto de relaciones *prestatario*. Si en un banco particular un préstamo puede pertenecer únicamente a un cliente y un cliente puede tener varios préstamos, entonces el conjunto de relaciones de *cliente* a *préstamo* es uno a varios. Si un préstamo puede pertenecer a varios clientes (como préstamos que se toman en conjunto por varios socios de un negocio) el conjunto de relaciones es varios a varios. Este tipo de relación se describe en la Figura 2.3.

### 2.2.2. Restricciones de participación

La participación de un conjunto de entidades  $E$  en un conjunto de relaciones  $R$  se dice que es **total** si cada entidad en  $E$  participa al menos en una relación en  $R$ . Si sólo algunas entidades en  $E$  participan en relaciones en  $R$ , la participación del conjunto de entidades  $E$  en la relación  $R$  se llama **parcial**. Por ejemplo, se puede esperar que cada entidad *préstamo* esté relacionada con al

menos un cliente mediante la relación *prestatario*. Por lo tanto, la participación de *préstamo* en el conjunto de relaciones *prestatario* es total. En cambio, un individuo puede ser cliente de un banco tenga o no tenga un préstamo en el banco. Así, es posible que sólo algunas de las entidades *cliente* estén relacionadas con el conjunto de entidades *préstamo* mediante la relación *prestatario*, y la participación de *cliente* en el conjunto de relaciones *prestatario* es por lo tanto parcial.

## 2.3. CLAVES

Es necesario tener una forma de especificar cómo las entidades dentro de un conjunto de entidades dado y las relaciones dentro de un conjunto de relaciones dado son distinguibles. Conceptualmente las entidades y relaciones individuales son distintas; desde una perspectiva de bases de datos, sin embargo, la diferencia entre ellas se debe expresar en término de sus atributos.

Por lo tanto, los valores de los atributos de una entidad deben ser tales que permitan *identificar unívocamente* a la entidad. En otras palabras, no se permite que ningún par de entidades tengan exactamente los mismos valores de sus atributos.

Una *clave* permite identificar un conjunto de atributos suficiente para distinguir las entidades entre sí. Las claves también ayudan a identificar unívocamente a las relaciones y así a distinguir las relaciones entre sí.

### 2.3.1. Conjuntos de entidades

Una **superclave** es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades. Por ejemplo, el atributo *id-cliente* del conjunto de entidades *cliente* es suficiente para distinguir una entidad *cliente* de las otras. Así, *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, porque varias personas podrían tener el mismo nombre.

El concepto de una superclave no es suficiente para lo que aquí se propone, ya que, como se ha visto, una superclave puede contener atributos innecesarios. Si  $K$  es una superclave, entonces también lo es cualquier superconjunto de  $K$ . A menudo interesan las superclaves tales que los subconjuntos propios de ellas no son superclave. Tales superclaves mínimas se llaman **claves candidatas**.

Es posible que conjuntos distintos de atributos pudieran servir como clave candidata. Supóngase que una combinación de *nombre-cliente* y *calle-cliente* es suficiente para distinguir entre los miembros del conjunto de entidades *cliente*. Entonces, los conjuntos  $\{id-cliente\}$  y  $\{nombre-cliente, calle-cliente\}$  son claves candi-

datas. Aunque los atributos *id-cliente* y *nombre-cliente* juntos puedan distinguir entidades *cliente*, su combinación no forma una clave candidata, ya que el atributo *id-cliente* por sí solo es una clave candidata.

Se usará el término **clave primaria** para denotar una clave candidata que es elegida por el diseñador de la base de datos como elemento principal para identificar las entidades dentro de un conjunto de entidades. Una clave (primaria, candidata y superclave) es una propiedad del conjunto de entidades, más que de las entidades individuales. Cualesquiera dos entidades individuales en el conjunto no pueden tener el mismo valor en sus atributos clave al mismo tiempo. La designación de una clave representa una restricción en el desarrollo del mundo real que se modela.

Las claves candidatas se deben designar con cuidado. Como se puede comprender, el nombre de una persona es obviamente insuficiente, ya que hay mucha gente con el mismo nombre. En España, el D.N.I. puede ser una clave candidata. Como los no residentes en España normalmente no tienen D.N.I., las empresas internacionales pueden generar sus propios identificadores únicos. Una alternativa es usar alguna combinación única de otros atributos como clave.

La clave primaria se debería elegir de manera que sus atributos nunca, o muy raramente, cambien. Por ejemplo, el campo dirección de una persona no debería formar parte de una clave primaria, porque probablemente cambiará. Los números de D.N.I., por otra parte, es seguro que no cambiarán. Los identificadores únicos generados por empresas generalmente no cambian, excepto si se fusionan dos empresas; en tal caso el mismo identificador puede haber sido emitido por ambas empresas y es necesario la reasignación de identificadores para asegurarse de que sean únicos.

### 2.3.2. Conjuntos de relaciones

La clave primaria de un conjunto de entidades permite distinguir entre las diferentes entidades del conjunto. Se necesita un mecanismo similar para distinguir entre las diferentes relaciones de un 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*

$ria(E_i)$  el conjunto de atributos que forma la clave primaria para el conjunto de entidades  $E_i$ .

Asúmase por el momento 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 para un conjunto de relaciones depende de la estructura de los atributos asociados al conjunto de relaciones  $R$ .

Si el conjunto de relaciones  $R$  no tiene atributos asociados, entonces el conjunto de atributos:

$$\text{clave-primaria}(E_1) \cup \text{clave-primaria}(E_2) \cup \dots \\ \cup \text{clave-primaria}(E_n)$$

describe una relación individual en el conjunto  $R$ .

Si el conjunto de relaciones  $R$  tiene atributos  $a_1, a_2, \dots, a_m$  asociados a él, entonces el conjunto de atributos

$$\text{clave-primaria}(E_1) \cup \text{clave-primaria}(E_2) \cup \dots \\ \cup \text{clave-primaria}(E_n) \cup \{a_1, a_2, \dots, a_m\}$$

describe una relación individual en el conjunto  $R$ .

En ambos casos, el conjunto de atributos

$$\text{clave-primaria}(E_1) \cup \text{clave-primaria}(E_2) \cup \dots \\ \cup \text{clave-primaria}(E_n)$$

forma una superclave para el conjunto de relaciones.

En el caso de que los nombres de atributos de las claves primarias no sean únicos en todos los conjuntos de entidades, los atributos se renombran para distinguirlos; el nombre del conjunto de entidades combinado con el atributo formaría un nombre único. En el caso de que

un conjunto de entidades participe más de una vez en un conjunto de relaciones (como en la relación *trabaja-para* del Apartado 2.1.2) el nombre del papel se usa en lugar del nombre del conjunto de entidades para formar un nombre único de atributo.

La estructura de la clave primaria para el conjunto de relaciones depende de la correspondencia de cardinalidades asociada al conjunto de relaciones. Como ilustración, considérese el conjunto de entidades *cliente* y *cuenta*, y un conjunto de relaciones *impositor*, con el atributo *fecha-acceso* del Apartado 2.1.2. Supóngase que el conjunto de relaciones es varios a varios. Entonces la clave primaria de *impositor* consiste en la unión de las claves primarias de *cliente* y *cuenta*. Sin embargo, si un cliente puede tener sólo una cuenta —es decir, si la relación *impositor* es varios a uno de *cliente* a *cuenta*— entonces la clave primaria de *impositor* es simplemente la clave primaria de *cliente*. Análogamente, si la relación es varios a uno de *cuenta* a *cliente* —es decir, cada cuenta pertenece a lo sumo a un cliente— entonces la clave primaria de *impositor* es simplemente la clave primaria de *cuenta*. Para relaciones uno a uno se puede usar cualquier clave primaria.

Para las relaciones no binarias, si no hay restricciones de cardinalidad, entonces la superclave formada como se describió anteriormente en este apartado es la única clave candidata, y se elige como clave primaria. La elección de la clave primaria es más complicada si aparecen restricciones de cardinalidad. Ya que no se ha discutido cómo especificar restricciones de cardinalidad en relaciones no binarias, no se discutirá este aspecto en este capítulo. Se considerará este aspecto con más detalle en el apartado 7.3.

## 2.4. CUESTIONES DE DISEÑO

Las nociones de conjunto de entidades y conjunto de relaciones no son precisas, y es posible definir un conjunto de entidades y las relaciones entre ellas de diferentes formas. En este apartado se examinan cuestiones básicas de diseño de un esquema de bases de datos E-R. El proceso de diseño se trata con más detalle en el Apartado 2.7.4.

### 2.4.1. Uso de conjuntos de entidades o atributos

Considérese el conjunto de entidades *empleado* con los atributos *nombre-empleado* y *número-teléfono*. Se puede argumentar fácilmente que un *teléfono* es una entidad por sí misma con atributos *número-teléfono* y *ubicación* (la oficina donde está ubicado el teléfono). Si se toma este punto de vista, el conjunto de entidades *empleado* debe ser redefinido como sigue:

- El conjunto de entidades *empleado* con el atributo *nombre-empleado*
- El conjunto de entidades *teléfono* con atributos *número-teléfono* y *ubicación*
- La relación *empleado-teléfono*, que denota la asociación entre empleados y los teléfonos que tienen.

¿Cuál es, entonces, la diferencia principal entre esas dos definiciones de un empleado? Al tratar un teléfono como un atributo *número-teléfono* implica que cada empleado tiene precisamente un número de teléfono. Al tratar un teléfono como una entidad *teléfono* permite que los empleados puedan tener varios números de teléfono (incluido ninguno) asociados a ellos. Sin embargo, se podría definir fácilmente *número-teléfono* como un atributo multivalorado para permitir varios teléfonos por empleado.



La diferencia principal es que al tratar un teléfono como una entidad se modela mejor una situación en la que se puede querer almacenar información extra sobre un teléfono, como su ubicación, su tipo (móvil, video-teléfono o fijo) o quiénes comparten un teléfono. Así, al tratar un teléfono como una entidad es más general que tratarlo como un atributo y es apropiado cuando la generalidad pueda ser de utilidad.

En cambio, no sería adecuado tratar el atributo *nombre-empleado* como una entidad; es difícil argumentar que *nombre-empleado* sea una entidad por sí mismo (a diferencia del teléfono). Así, es apropiado tener *nombre-empleado* como un atributo del conjunto de entidades *empleado*.

Por tanto, aparecen dos cuestiones naturales: ¿qué constituye un atributo? y ¿qué constituye un conjunto de entidades? Por desgracia no hay respuestas simples. Las distinciones dependen principalmente de la estructura de la empresa del mundo 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 un atributo de otro conjunto de entidades, en lugar de usar una relación. Por ejemplo, es incorrecto modelar *id-cliente* como un atributo de *préstamo* incluso si cada préstamo tiene sólo un cliente. La relación *prestatario* es la forma correcta de representar la conexión entre préstamos y clientes, ya que hace su conexión explícita en lugar de implícita mediante un atributo.

Otro error relacionado que se comete es designar a los atributos de la clave primaria de los conjuntos de entidades relacionados como atributos del conjunto de relaciones. Esto no se debería hacer, ya que los atributos de la clave primaria son ya implícitos en la relación.

#### 2.4.2. Uso de conjuntos de entidades o conjuntos de relaciones

No siempre está claro si es mejor expresar un objeto mediante un conjunto de entidades o mediante un conjunto de relaciones. En el Apartado 2.1.1 se asumió que un préstamo se modelaba como una entidad. Una alternativa es modelar un préstamo no como una entidad, sino como una relación entre clientes y 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 está asociado exactamente con un cliente y con una sucursal, se puede encontrar satisfactorio el diseño en el que un préstamo se representa como una relación. Sin embargo, con este diseño no se puede representar convenientemente una situación en que varios clientes comparten un préstamo. Habría que definir una relación separada para cada prestatario de ese préstamo común. Entonces habría que replicar los valores para los atributos descriptivos *número-préstamo* e *importe* en cada una de estas relaciones. Cada una de

estas relaciones debe, por supuesto, tener el mismo valor para los atributos descriptivos *número-préstamo* e *importe*.

Surgen dos problemas como resultado de esta réplica: 1) los datos se almacenan varias veces, desperdiçando espacio de almacenamiento; y 2) las actualizaciones dejan potencialmente los datos en un estado inconsistente, en el que los valores difieren en dos relaciones para atributos que se supone tienen el mismo valor. El asunto de cómo evitar esta réplica se trata formalmente mediante la *teoría de la normalización*, discutida en el Capítulo 7.

El problema de la réplica de los atributos *número-préstamo* e *importe* no aparece en el diseño original del Apartado 2.1.1. porque *préstamo* es un conjunto de entidades.

Una posible guía para determinar si usar un conjunto de entidades o un conjunto de relaciones es designar un conjunto de relaciones para describir una acción que ocurre entre entidades. Este enfoque puede también ser útil para decidir si ciertos atributos se pueden expresar más apropiadamente como relaciones.

#### 2.4.3. Conjuntos de relaciones binarias o *n*-arias

Las relaciones en las bases de datos son generalmente binarias. Algunas relaciones que parecen no ser binarias podrían ser representadas mejor con varias relaciones binarias. Por ejemplo, uno podría crear una relación ternaria *padres*, que relaciona un hijo con su padre y su madre. Sin embargo, tal relación se podría representar por dos relaciones binarias *padre* y *madre*, relacionando un hijo con su padre y su madre por separado. Al usar las dos relaciones *padre* y *madre* se permite registrar la madre de un niño incluso si no se conoce la identidad del padre; en la relación ternaria *padres* se necesitaría usar un valor nulo. En este caso es preferible usar conjuntos de relaciones binarias.

De hecho, siempre es posible reemplazar un conjunto de relaciones no binarias (*n*-aria, para  $n > 2$ ) por un número de diferentes conjuntos de relaciones binarias. Por simplicidad, considérese el conjunto de relaciones abstracto *R*, ternario ( $n = 3$ ), 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:

- $R_A$ , relacionando *E* y *A*
- $R_B$ , relacionando *E* y *B*
- $R_C$ , relacionando *E* y *C*

Si el conjunto de relaciones *R* tiene atributos, éstos se asignan al conjunto de entidades *E*; por otra parte se crea un atributo de identificación especial para *E* (debido a que cada conjunto de entidades debe tener al menos un atributo para distinguir los miembros del conjunto).

Para cada relación  $(a_i, b_i, c_i)$  del conjunto de relaciones  $R$ , se crea una nueva entidad  $e_i$  en el conjunto de entidades  $E$ . Entonces, en cada uno de los tres nuevos conjuntos de relaciones, se inserta un nuevo miembro como sigue:

- $(e_i, a_i)$  en  $R_A$
- $(e_i, b_i)$  en  $R_B$
- $(e_i, c_i)$  en  $R_C$

Se puede generalizar este proceso de una forma semejante a conjuntos de relaciones  $n$ -arias. Así, conceptualmente, se puede restringir el modelo E-R para incluir sólo conjuntos de relaciones binarias. Sin embargo, esta restricción no siempre es deseable.

- Un atributo de identificación puede haber sido creado para el conjunto de entidades para representar el conjunto de relaciones. Este atributo, con los conjuntos de relaciones extra necesarios, incrementa la complejidad del diseño y (como se verá en el Apartado 2.9) los requisitos de almacenamiento.
- Un conjunto de relaciones  $n$ -arias muestra más claramente que varias entidades participan en una relación simple.
- Podría no haber una forma de traducir restricciones en la relación ternaria en restricciones sobre 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  $B$  se asocia con a lo sumo una entidad  $C$ . Esta restricción no se puede expresar usando restricciones de cardinalidad sobre los conjuntos de relaciones  $R_A, R_B$  y  $R_C$ .

Considérese el conjunto de relaciones *trabaja-en* del Apartado 2.1.2 que relaciona *empleado*, *sucursal* y *trabajo*. No se puede dividir directamente *trabaja-en* en relaciones binarias entre *empleado* y *sucursal* y entre *empleado* y *trabajo*. Si se hiciese habría que registrar que Santos es director y auditor y que Santos trabaja en Navacerrada y Centro; sin embargo, no se podría registrar que Santos es director de Navacerrada y auditor de Centro, pero que no es auditor de Navacerrada y director de Centro.

El conjunto de relaciones *trabaja-en* se puede dividir en relaciones binarias creando nuevos conjuntos de entidades como se describió anteriormente. Sin embargo, no sería muy natural.

#### 2.4.4. Ubicación de los atributos de las relaciones

La razón de cardinalidad de una relación puede afectar a la situación de los atributos de la relación. 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, especificamos que *impositor* es un conjunto de relaciones uno a varios tal que un cliente puede tener varias cuentas, pero cada cuenta está asociada únicamente con un cliente. En este caso, el atributo *fecha-acceso*, que especifica cuándo accedió por última vez el cliente a la cuenta, podría estar asociado con el conjunto de entidades *cuenta*, como se describe en la Figura 2.6; para mantener la simplicidad de la figura sólo se muestran algunos de los atributos de los dos conjuntos de entidades. Como cada entidad *cuenta* participa en una relación con a lo sumo un ejemplar de *cliente*, hacer esta designación de atributos tendría el mismo significado que si se colocase *fecha-acceso* en el conjunto de relaciones *impositor*. Los atributos de un conjunto de relaciones uno a varios se pueden colocar sólo 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 de dónde colocar los atributos descriptivos en tales casos — como un atributo de la relación o de la entidad — podría reflejar las características de la empresa que se modela. El diseñador puede elegir mantener *fecha-acceso* como un atributo de *impositor* para expresar explícitamente que ocurre un acceso en el punto de interacción entre los conjuntos de entidades *cliente* y *cuenta*.

La elección de la colocación del atributo es más clara para los conjuntos de relaciones varios a varios. Volviendo al ejemplo, especificamos el caso quizá más realista de *impositor* que es un conjunto de relaciones varios a varios, expresando que un cliente puede tener una o más cuentas, y que una cuenta puede ser mantenida por uno o más clientes. Si se expresa la fecha en que un cliente específico accedió por última vez a una cuenta específica, *fecha-acceso* debe ser un atributo del conjunto de relaciones *impositor*, en lugar de una de las entidades participantes. Si *fecha-acceso* fuese un atributo de *cuenta*, por ejemplo, no se podría determinar

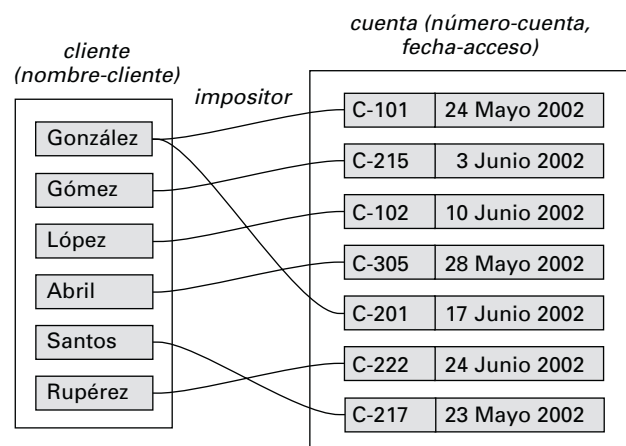


FIGURA 2.6. *Fecha-acceso* como atributo del conjunto de entidades *cuenta*.

qué cliente hizo el acceso más reciente a una cuenta conjunta. Cuando un atributo se determina mediante la combinación de los conjuntos de entidades participantes, en lugar de por cada entidad por separado, ese atributo debe estar asociado con el conjunto de relaciones varios a

varios. La colocación de *fecha-acceso* como un atributo de la relación se describe en la Figura 2.7; de nuevo, para mantener la simplicidad de la figura, sólo se muestran algunos de los atributos de los dos conjuntos de entidades.

## 2.5. DIAGRAMA ENTIDAD-RELACIÓN

Como se vio brevemente en el Apartado 1.4, la estructura lógica general de una base de datos se puede expresar gráficamente mediante un **diagrama E-R**. Los diagramas son simples y claros, cualidades que pueden ser responsables del amplio uso del modelo E-R. Tal diagrama consta de los siguientes componentes principales:

- **Rectángulos**, que representan conjuntos de entidades.
- **Elipses**, que representan atributos.
- **Rombos**, que representan relaciones.
- **Líneas**, que unen atributos a conjuntos de entidades y conjuntos de entidades a 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 (se describirán posteriormente en el Apartado 2.6).

Considérese el diagrama entidad-relación de la Figura 2.8, que consta de dos conjuntos de entidades, *cliente* y *préstamo*, relacionadas a través de un conjunto de relaciones binarias *prestatario*. Los atributos asociados con *cliente* son *id-cliente*, *nombre-cliente*, *calle-cliente*, y *ciudad-cliente*. Los atributos asociados con *préstamo* son *número-préstamo* e *importe*. Como se muestra en la Figura 2.8, los atributos de un conjunto de entidades que son miembros de la clave primaria están subrayados.

El conjunto de relaciones *prestatario* puede ser varios a varios, uno a varios, varios a uno o uno a uno. Para distinguir entre estos tipos, se dibuja o una línea dirigida ( $\rightarrow$ ) o una línea no dirigida ( $—$ ) entre el conjunto de relaciones y el conjunto de entidades en cuestión.

- Una línea dirigida desde el conjunto de relaciones *prestatario* al conjunto de entidades *préstamo* espe-

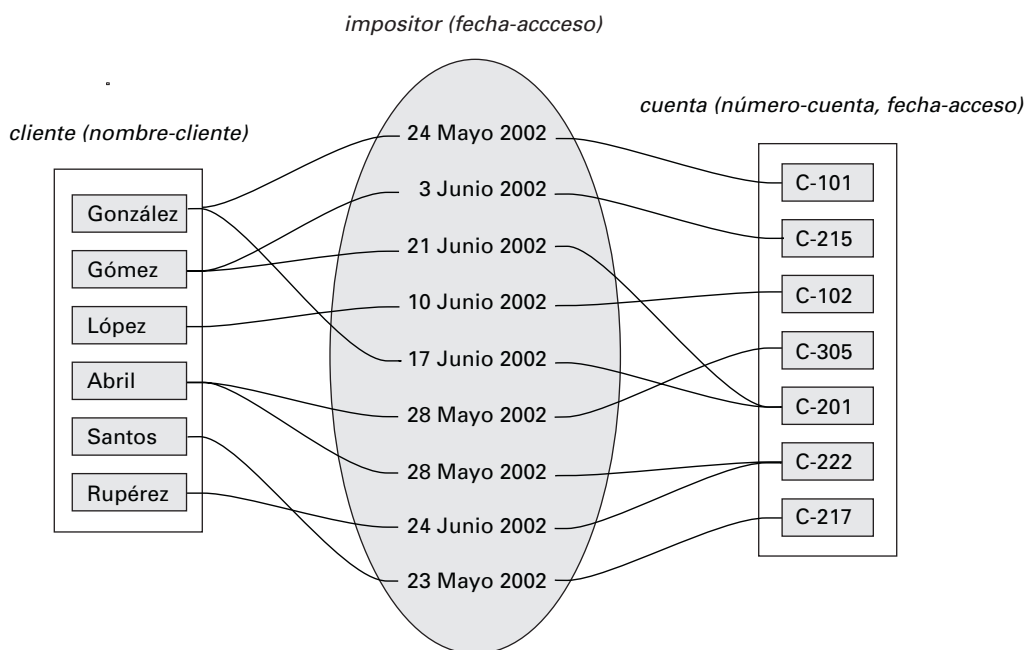
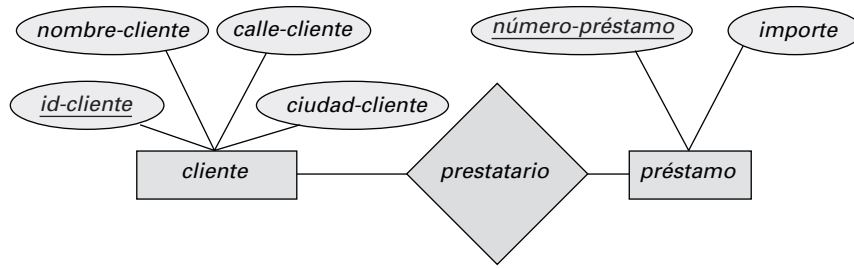


FIGURA 2.7. *Fecha-acceso* como atributo del conjunto de relaciones *impositor*.



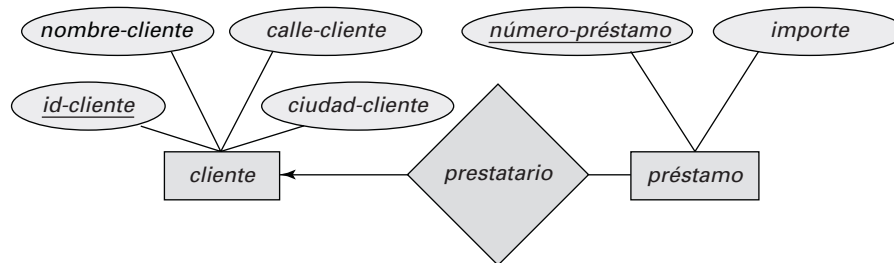
**FIGURA 2.8.** Diagrama E-R correspondiente a clientes y préstamos.

cifica que *prestatario* es un conjunto de relaciones uno a uno, o bien 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*.

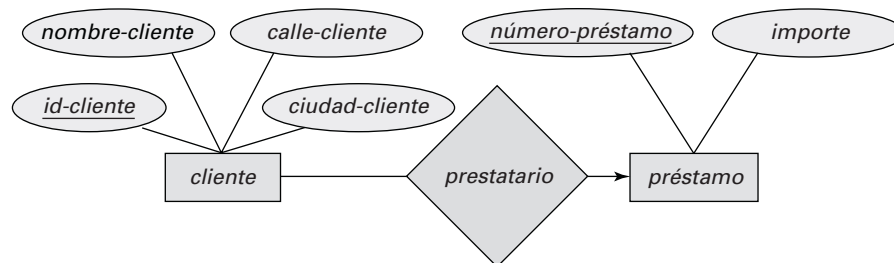
- Una línea no dirigida desde el conjunto de relaciones *prestatario* al conjunto de relaciones *préstamo* especifica que *prestatario* es o bien un con-

junto de relaciones varios a varios, o bien uno a varios, desde *cliente* a *préstamo*.

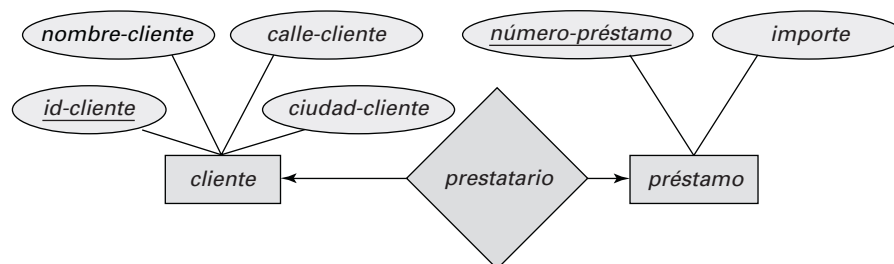
Volviendo al diagrama E-R de la Figura 2.8, se ve que el conjunto de relaciones *prestatario* es varios a varios. Si el conjunto de relaciones *prestatario* fuera uno a varios, desde *cliente* a *préstamo*, entonces la línea desde *prestatario* a *cliente* sería dirigida, con una flecha apuntando al conjunto de entidades *cliente* (Figura 2.9a). Análogamente, si el conjunto de relaciones *pres-*



(a)

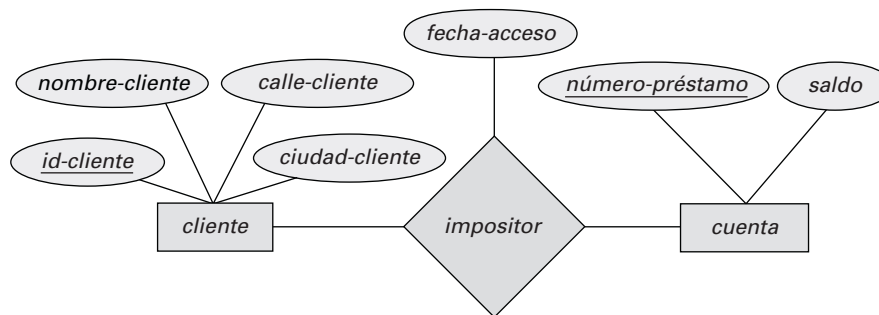


(b)



(c)

**FIGURA 2.9.** Relaciones. (a) Uno a varios. (b) Varios a uno. (c) Uno a uno.



**FIGURA 2.10.** Diagrama E-R con un atributo unido a un conjunto de relaciones.

*tatario* fuera varios a uno desde *cliente* a *préstamo*, entonces la línea desde *prestatario* a *préstamo* tendría una flecha apuntando al conjunto de entidades *préstamo* (Figura 2.9b). Finalmente, si el conjunto de relaciones *prestatario* fuera uno a uno, entonces ambas líneas desde *prestatario* tendrían flechas: una apuntando al conjunto de entidades *préstamo* y otra apuntando al conjunto de entidades *cliente* (Figura 2.9c).

Si un conjunto de relaciones tiene también algunos atributos asociados a él, entonces se unen esos atributos a ese conjunto de relaciones. Por ejemplo, en la Figura 2.10, se tiene el atributo descriptivo *fecha-acceso* unido al conjunto de relaciones *impositor* para especificar la fecha más reciente en la que un cliente accedió a esa cuenta.

La Figura 2.11 muestra cómo se pueden representar atributos compuestos en la notación E-R. Aquí, el atributo compuesto *nombre*, con atributos componentes *nombre-pila*, *primer-apellido* y *segundo-apellido* reemplaza al atributo simple *nombre-cliente* de *cliente*. También se muestra el atributo compuesto *dirección*, cuyos atributos componentes son *calle*, *ciudad*, *provincia* y *código-postal*, que reemplaza a los atributos *calle-cliente* y *ciudad-cliente* de *cliente*. El atributo *calle* es por si

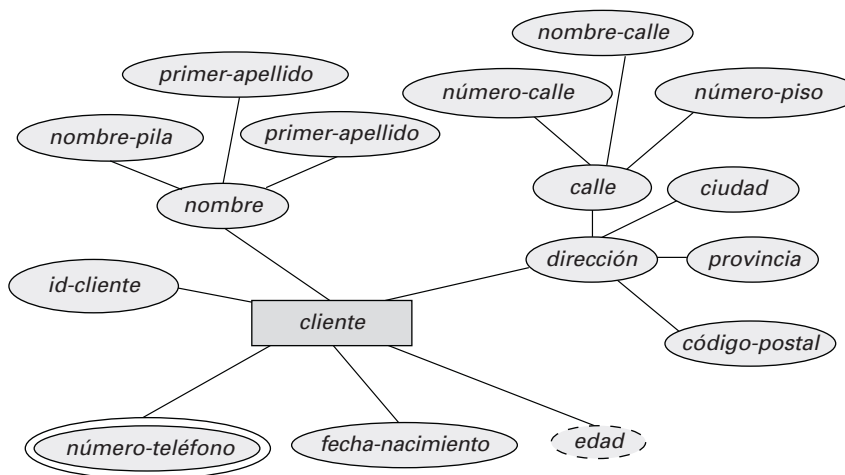
mismo un atributo compuesto cuyos atributos componentes son *número-calle*, *nombre-calle* y *número-piso*.

La Figura 2.11 también muestra un atributo multivalorado, *número-teléfono*, indicado por una elipse doble, y un atributo derivado *edad*, indicado por una elipse discontinua.

En los diagramas E-R se indican papeles mediante etiquetas en las líneas que unen rombos con rectángulos. En la Figura 2.12 se muestran los indicadores de papeles *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 un diagrama E-R. La Figura 2.13 consta de tres conjuntos de entidades *cliente*, *trabajo* y *sucursal*, relacionados a través del conjunto de relaciones *trabaja-en*.

Se pueden especificar algunos tipos de relaciones varios a uno en el caso de conjuntos de relaciones no binarias. Supóngase un empleado que tenga a lo sumo un trabajo en cada sucursal (por ejemplo, Santos no puede ser director y auditor en la misma sucursal). Esta restricción se puede especificar con una flecha apuntando a *trabajo* en el borde de *trabaja-en*.



**FIGURA 2.11.** Diagrama E-R con atributos compuestos, multivalorados y derivados.



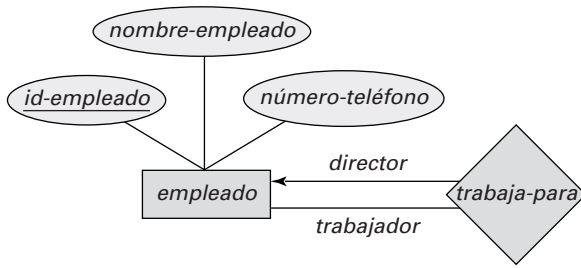


FIGURA 2.12. Diagrama E-R con indicadores de papeles.

Se permite a lo sumo una flecha desde un conjunto de relaciones, ya que un diagrama E-R con dos o más flechas salientes de un conjunto de relaciones no binarias se puede interpretar de dos formas. Supónganse que hay un conjunto de relaciones  $R$  entre conjuntos de entidades  $A_1, A_2, \dots, A_n$ , y las únicas flechas están en los bordes de los conjuntos de entidades  $A_{i+1}, A_{i+2}, \dots, A_n$ . Entonces, las dos posibles interpretaciones son:

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

Cada una de estas interpretaciones se han usado en diferentes libros y sistemas. Para evitar confusión se permite sólo una flecha que salga de un conjunto de relaciones, y así las representaciones son equivalentes. En el Capítulo 7 (Apartado 7.3) se estudia la noción de *dependencias funcionales*, que permiten especificar cualquiera de estas dos interpretaciones sin ambigüedad.

En el diagrama E-R se usan las líneas dobles para indicar que la participación de un conjunto de entidades en un conjunto de relaciones es total; es decir, cada entidad en el conjunto de entidades aparece al menos en una relación en ese conjunto de relaciones. Por ejemplo, considérese la relación *prestamista* entre clientes y préstamos. Una línea doble de *préstamo* a *prestamista*, como en la Figura 2.14, indica que cada préstamo debe tener al menos un cliente asociado.

Los diagramas E-R también proporcionan una forma de indicar restricciones más complejas sobre el número de veces en 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 una cardinalidad mínima y máxima, mostrada de la forma *mín..máx*, donde *mín* es la mínima cardinalidad y *máx* 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 de a lo sumo una relación, mientras que un valor máximo de \* indica que no hay límite. Nótese que una etiqueta 1..\* en un segmento es equivalente a una línea doble.

Por ejemplo, considérese la Figura 2.15. El segmento entre *préstamo* y *prestamista* tiene una restricción de cardinalidad de 1..1, significando que la cardinalidad mínima y máxima son ambas 1. Es decir, cada préstamo debe tener exactamente un cliente asociado. El límite 0..\* en el segmento de *cliente* a *prestamista* indica que un cliente puede tener ninguno o varios préstamos. Así, la relación *prestamista* es uno a varios de *cliente* a *préstamo*, y además la participación de *préstamo* en *prestamista* es total.

Es fácil malinterpretar 0..\* en el segmento entre *cliente* y *prestamista*, y pensar que la relación *prestamista* es de varios a uno de *cliente* a *préstamo* —esto es exactamente lo contrario a la interpretación correcta.

Si ambos 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 *prestamista*, se estaría diciendo que cada cliente debe tener al menos un préstamo.

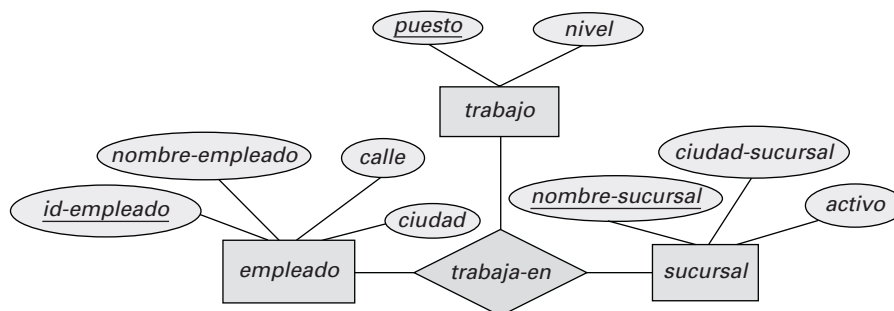


FIGURA 2.13. Diagrama E-R con una relación ternaria.



FIGURA 2.14. Participación total de un conjunto de entidades en un conjunto de relaciones.

## 2.6. CONJUNTOS DE ENTIDADES DÉBILES

Un conjunto de entidades puede no tener suficientes atributos para formar una clave primaria. Tal conjunto de entidades se denomina **conjunto de entidades débiles**. Un conjunto de entidades que tiene una clave primaria se denomina **conjunto de entidades fuertes**.

Como ilustración, considérese el conjunto de entidades *pago*, que tiene los tres atributos: *número-pago*, *fecha-pago* e *importe-pago*. Los números de pago son generalmente números secuenciales, empezando por 1, generados por separado por cada préstamo. Así, aunque cada entidad *pago* es distinta, los pagos para diferentes préstamos pueden compartir el mismo número de pago. Así, este conjunto de entidades no tiene una clave primaria; es un conjunto de entidades débiles.

Para que un conjunto de entidades débiles tenga sentido, debe estar asociada con otro conjunto de entidades, denominado el **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 es **propietaria** del conjunto de entidades débiles 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 para *pago* es *préstamo*, y la relación *préstamo-pago* que asocia las entidades *pago* con sus correspondientes entidades *préstamo* es la relación identificadora.

Aunque un conjunto de entidades débiles no tiene clave primaria, no obstante se necesita conocer un medio para distinguir todas aquellas entidades del conjunto de entidades que dependen de una entidad fuerte particular. El **discriminante** de un conjunto de entidades débiles es un conjunto de atributos que permite que esta distinción se haga. Por ejemplo, el discriminante del conjunto de entidades débiles *pago* es el atributo *número-pago*, ya que, para cada préstamo, un número de pago identifica de forma única cada pago para ese préstamo. El discriminante de un conjunto de entidades débiles se denomina la *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, más 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* dentro del mismo préstamo.

El conjunto de entidades identificadoras no debería tener atributos descriptivos, ya que cualquier atributo requerido puede estar asociado con el conjunto de enti-

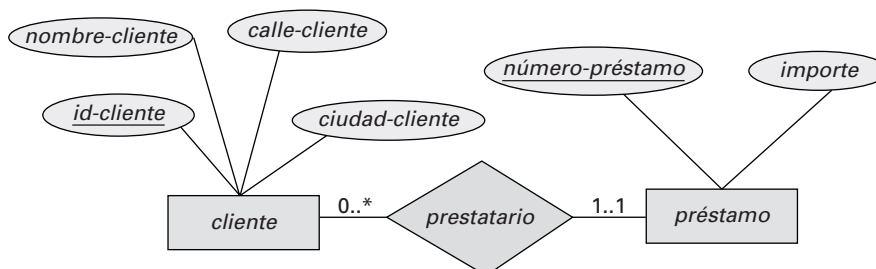


FIGURA 2.15. Límites de cardinalidad en conjuntos de relaciones.

dades débiles (véase la discusión de trasladar los atributos del conjunto de relaciones a los conjuntos de entidades participantes en el Apartado 2.2.1).

Un conjunto de entidades débiles puede participar en relaciones distintas de relaciones identificadoras. Por ejemplo, la entidad *pago* podría participar en una relación con el conjunto de entidades con el conjunto de entidades *cuenta*, identificando la cuenta desde la que se realizó el pago. Un conjunto de entidades débiles puede participar como propietario en una relación identificadora con otro conjunto de entidades débiles. También es posible tener un conjunto de entidades débiles con más de un conjunto de entidades identificadoras. Una entidad débil en concreto podría ser identificada por 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 más el discriminante del conjunto de entidades débiles.

Un conjunto de entidades débiles se indica en los diagramas E-R mediante un rectángulo dibujado con una línea doble y la correspondiente relación de identificación mediante un rombo dibujado con línea doble. En la Figura 2.16, el conjunto de entidades débiles *pago* es dependiente del conjunto de entidades fuertes *préstamo* a través del conjunto de relaciones *pago-préstamo*.

La figura ilustra también el uso de líneas dobles para indicar *participación total*; la participación del conjunto de entidades (débiles) *pago* en la relación *pago-préstamo* es total, significando que cada pago debe estar relacionando a través de *pago-préstamo* con alguna

cuenta. Finalmente, la flecha desde *pago-préstamo* a *préstamo* indica que cada pago es para un único préstamo. El discriminante del conjunto de entidades débiles también está subrayado, pero con un línea discontinua, en lugar de una continua.

En algunos casos, el diseñador de la base de datos puede elegir expresar un conjunto de entidades débiles como un atributo compuesto multivalorado del conjunto de entidades propietarias. En el ejemplo, esta alternativa requeriría que el conjunto de entidades *préstamo* tuviera un atributo compuesto y multivalorado *pago*, que constara de *número-pago*, *fecha-pago* e *importe-pago*. Un conjunto de entidades débiles se puede modelar más adecuadamente como un atributo si sólo participa en la relación identificadora y si tiene pocos atributos. Alternativamente, una representación de conjunto de entidades débiles será más adecuada para modelar una situación en la que el conjunto participe en otras relaciones además de la relación identificadora y donde el conjunto de entidades débiles tenga muchos atributos.

Como otro de un conjunto de entidades que se puede modelar como un conjunto de entidades débiles considérense las ofertas de asignaturas en una universidad. La misma asignatura se puede ofrecer en diferentes cursos y dentro de un curso puede haber varios grupos para la misma asignatura. Así, se crea un conjunto de entidades débiles *oferta-asignatura*, que depende existencialmente de *asignatura*; las diferentes ofertas de la misma asignatura se identifican por un *curso* y un *número-grupo*, que forma un discriminante pero no una clave primaria.

## 2.7. CARACTERÍSTICAS DEL MODELO E-R EXTENDIDO

Aunque los conceptos básicos de E-R pueden modelar la mayoría de las características de las bases de datos, algunos aspectos de una base de datos pueden ser más adecuadamente expresados mediante ciertas extensio-

nes del modelo E-R básico. En este apartado se discuten las características E-R extendidas de especialización, generalización, conjuntos de entidades de nivel más alto y más bajo, herencia de atributos y agregación.

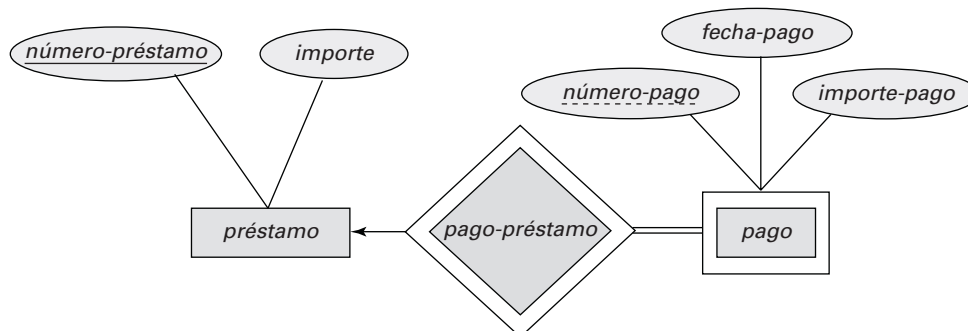


FIGURA 2.16. Diagrama E-R con un conjunto de entidades débiles.

### 2.7.1. Especialización

Un conjunto de entidades puede incluir subgrupos de entidades que se diferencian de alguna forma de las otras entidades del conjunto. Por ejemplo, un subconjunto de entidades en un conjunto de entidades puede tener atributos que no son compartidos por todas las entidades del conjunto de entidades. El modelo E-R proporciona una forma de representación de estos grupos de entidades distintos.

Considérese el conjunto de entidades *persona* con atributos *nombre*, *calle* y *ciudad*. Una persona puede clasificarse además como:

- *cliente*
- *empleado*

Cada uno de estos tipos de persona se describen mediante un conjunto de atributos que incluyen los atributos del conjunto de entidades *persona* más otros posibles atributos adicionales. Por ejemplo, las entidades *cliente* se pueden describir además mediante el atributo *id-cliente*, mientras que las entidades *empleado* se pueden describir además mediante los atributos *id-empleado* y *sueldo*. El proceso de designación de subgrupos dentro de un conjunto de entidades se denomina **especialización**. La especialización de *persona* permite distinguir entre las personas basándose en si son empleados o clientes.

Como otro ejemplo 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 establece diferentes tasas de interés a cada cliente, ofreciendo mejores tasas a los clientes favorecidos. Las cuentas corrientes tienen una tasa fija de interés, pero permiten los descubiertos; el importe de descubierto de una cuenta corriente se debe registrar.

El banco podría crear dos especializaciones de *cuenta*, denominadas *cuenta-ahorro* y *cuenta-corriente*. Como se vio anteriormente, las entidades *cuenta* se describen por los atributos *número-cuenta* y *saldo*. El conjunto de entidades *cuenta-ahorro* tendría todos los atributos de *cuenta* y un atributo adicional denominado *tasa-interés*. El conjunto de entidades *cuenta-corriente* tendría todos los atributos de *cuenta* y un atributo adicional *importe-descubierto*.

Se puede aplicar repetidamente la especialización para refinar el esquema de diseño. Por ejemplo, los empleados del banco se pueden clasificar en uno de los siguientes:

- *oficial*
- *cajero*
- *secretaria*

Cada uno de estos tipos de empleado se describe por un conjunto de atributos que incluye todos los atributos

del conjunto de entidades *empleado* más otros adicionales. Por ejemplo, las entidades *oficial* se puede describir por el atributo *número-despacho*, las entidades *cajero* por los atributos *número-sección* y *horas-semana*, y las entidades *secretaria* por el atributo *horas-semana*. Además, las entidades *secretaria* pueden participar en una relación *secretaria-de*, que identifica al empleado ayudado por una secretaria.

Un conjunto de entidades se puede especializar por más de una característica distintiva. En el ejemplo, la característica distintiva entre entidades *empleado* es el trabajo que realiza el empleado. Otra especialización coexistente podría estar basada en si la persona es un trabajador temporal o fijo, resultado en los conjuntos de entidades *empleado-temporal* y *empleado-fijo*. Cuando se forma más de una especialización de un conjunto de entidades, una entidad en particular puede pertenecer a varias especializaciones. Por ejemplo, una empleada dada puede ser una empleada temporal y secretaria.

En términos de un diagrama E-R, la especialización se representa mediante un componente *triangular* etiquetado ES, como se muestra en la Figura 2.17. La etiqueta ES representa, por ejemplo, que un cliente «es» una persona. La relación ES se puede llamar también relación **superclase-subclase**. Los conjuntos de entidades de nivel más alto y más bajo se representan como conjuntos de entidades regulares, es decir, como rectángulos que contienen el nombre del conjunto de entidades.

### 2.7.2. Generalización

El refinamiento a partir de un 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 puede ser también de una forma **ascendente**, en el que varios conjuntos de entidades se sintetizan en un conjunto de entidades de nivel más alto basado en características comunes. El diseñador de la base de datos puede haber identificado primero el conjunto de entidades *cliente* con los atributos *nombre*, *calle*, *ciudad* e *id-cliente*, y el conjunto de entidades *empleado* con los atributos *nombre*, *calle*, *ciudad*, *id-empleado* y *sueldo*.

Hay similitudes entre el conjunto de entidades *cliente* y el conjunto de entidades *empleado* en el sentido de que tienen varios atributos en común. Esta similitud se puede expresar mediante la **generalización**, que es una relación contenedora que existe entre el conjunto de entidades de *nivel más alto* y uno o más conjuntos de entidades de *nivel más bajo*. En el ejemplo, *persona* es el conjunto de entidades de nivel más alto y los conjuntos de entidades *cliente* y *empleado* son de nivel más bajo. Los conjuntos de entidades de nivel más alto y nivel más bajo también se pueden llamar **superclase** y **subclase**, respectivamente. El conjunto de entidades *persona* es la superclase de las subclases *cliente* y *empleado*.

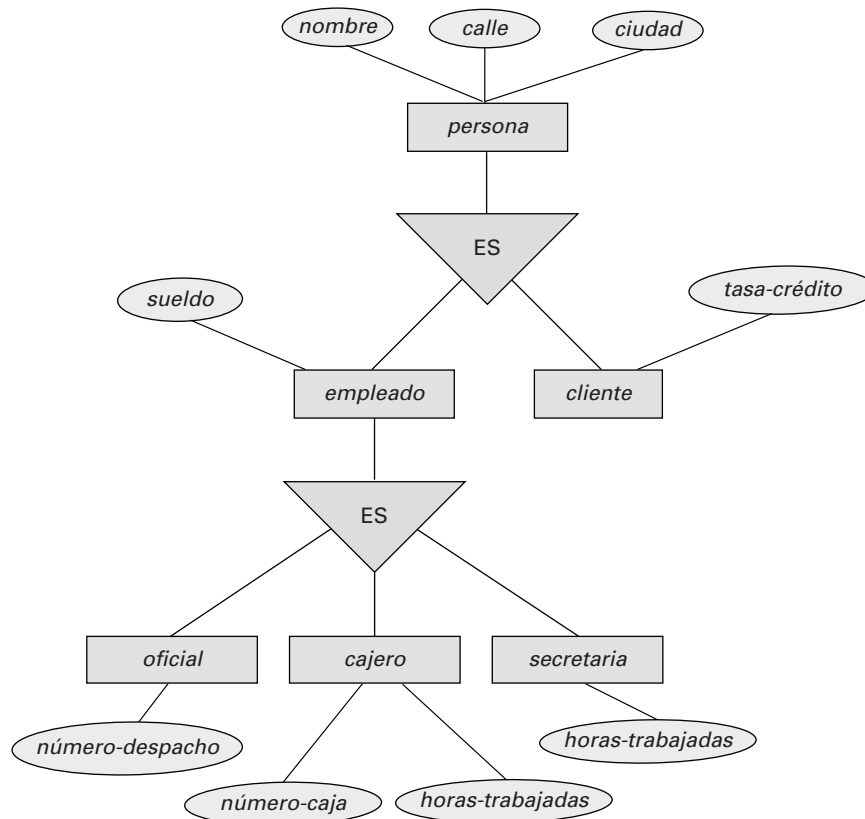


FIGURA 2.17. Especialización y generalización.

Para todos los propósitos prácticos, la generalización es una inversión simple de la especialización. Se aplicarán ambos procesos en combinación en el curso del diseño del esquema E-R para una empresa. En términos del propio diagrama E-R no se distingue entre especialización y generalización. Los niveles nuevos de representación de entidades serán distinguidos (especialización) o sintetizados (generalización) cuando el esquema de diseño llegue a expresar completamente la aplicación de base de datos y los requisitos de uso de la base de datos. Las diferencias entre los dos enfoques se pueden caracterizar mediante su punto de partida y el objetivo global.

La especialización parte de un conjunto de entidades simple; enfatiza las diferencias entre las entidades dentro del conjunto mediante la creación de distintos conjuntos de entidades de nivel más bajo. Estos conjuntos de entidades de nivel más bajo pueden tener atributos, o pueden participar en relaciones que no se aplican a todas las entidades del conjunto de entidades de nivel más alto. Realmente, la razón de que el diseñador aplique la especialización es representar tales características diferentes. Si *cliente* y *empleado* no tuvieran cada una atributos únicos que no tuvieran las entidades *persona* en la que participan, no habría necesidad de especializar el conjunto de entidades *persona*.

La generalización procede de observar que varios conjuntos de entidades que comparten algunas características comunes (se describen mediante los mismos atributos y participan en los mismos conjuntos de relaciones). Basada en sus similitudes, la generalización sintetiza estos conjuntos de entidades en uno solo, el conjunto de entidades de nivel más alto. La generalización se usa para resaltar las similitudes entre los conjuntos de entidades de nivel más bajo y para ocultar las diferencias; también permite economizar la representación para que los atributos compartidos no estén repetidos.

### 2.7.3. Herencia de atributos

Una propiedad crucial de las entidades de nivel más alto y más bajo creadas mediante especialización y generalización es la **herencia de atributos**. Los atributos de los conjuntos de entidades de nivel más alto se dice que son **heredados** por los conjuntos de entidades de nivel más bajo. Por ejemplo, *cliente* y *empleado* heredan los atributos de *persona*. Así, *cliente* se describe mediante sus atributos *nombre*, *calle* y *ciudad* y adicionalmente por el atributo *id-cliente*; *empleado* se describe mediante sus atributos *nombre*, *calle* y *ciudad* y adicionalmente por los atributos *id-empleado* y *sueldo*.

Un conjunto de entidades de nivel más bajo (o sub-clase) también hereda la participación en los conjuntos



de relaciones en los que su entidad de nivel más alto (o superclase) participa. Ambos conjuntos de entidades *oficial*, *cajero* y *secretaria* participan en el conjunto de relaciones *trabaja-para*. La herencia de atributos se aplica en todas las capas de los conjuntos de entidades de nivel más bajo. Los conjuntos de entidades anteriores pueden participar cualquier relación en que participe el conjunto de entidades *persona*.

Si se llega a una porción dada de un modelo E-R mediante especialización o generalización, el resultado es básicamente el mismo:

- Un conjunto de entidades de nivel más alto con atributos y relaciones que se aplican a todos los conjuntos de entidades de nivel más bajo.
- Conjuntos de entidades de nivel más bajo con características distintivas que se aplican sólo en un conjunto de entidades particular.

En lo que sigue, aunque a menudo se hará referencia sólo a la generalización, las propiedades que se discuten pertenecen a ambos procesos.

En la Figura 2.17 se describe una **jerarquía** de conjuntos de entidades. En la figura, *empleado* es un conjunto de entidades de nivel más bajo de *persona* y un conjunto de entidades de nivel más alto de los conjuntos de entidades *oficial*, *cajero* y *secretaria*. En una jerarquía, un conjunto de entidades dado puede estar implicado como un conjunto de entidades de nivel más bajo sólo en una única relación ES. Si un conjunto de entidades es un conjunto de entidades de nivel más bajo en más de una relación ES, entonces el conjunto de entidades tiene **herencia múltiple**, y la estructura resultante se denomina *retículo*.

#### 2.7.4. Restricciones sobre las generalizaciones

Para modelar una empresa más exactamente, el diseñador de la base de datos puede elegir colocar ciertas restricciones en una generalización particular. Un tipo de restricción implica determinar qué entidades pueden ser miembros de un conjunto de entidades de nivel más bajo dado. Tales relaciones de miembros pueden ser algunas de los siguientes:

- **Definido por condición.** En los conjuntos de entidades de nivel más bajo, la relación miembro se evalúa en función de si una entidad satisface o no una condición explícita o predicado. Por ejemplo, asúmase que el conjunto de entidades de nivel más alto *cuenta* tiene el atributo *tipo-cuenta*. Todas las entidades *cuenta* se evalúan según la definición del atributo *tipo-cuenta*. Sólo aquellas entidades que satisfagan la condición *tipo-cuenta* = «cuenta de ahorro» podrán pertenecer al conjunto de entidades de nivel más bajo *cuenta-ahorro*. Todas las entidades que satisfagan la condición *tipo-cuenta* = «cuenta corriente» estarán incluidas en *cuenta-*

*corriente*. Como todas las entidades de nivel más bajo se evalúan en función del mismo atributo (en este caso, *tipo-cuenta*), este tipo de generalización se denomina **definido por atributo**.

- **Definido por el usuario.** Los conjuntos de entidades de nivel más bajo definidos por el usuario no están restringidos mediante una condición de miembro; en cambio, las entidades se asignan a un conjunto de entidades dado por el usuario de la base de datos. Por ejemplo, asúmase que, después de tres meses de empleo, se asignan los empleados del banco a uno de los cuatro grupos de trabajo. Los grupos se representan, por tanto, como cuatro conjuntos de entidades de nivel más bajo del conjunto de entidades de nivel más alto *empleado*. Un empleado dado no se asigna a una entidad grupo automáticamente en términos de una condición que lo defina explícitamente. En su lugar, la asignación al grupo se hace de forma individual por el usuario a cargo de la decisión. La asignación se implementa mediante una operación que añade una entidad a un conjunto de entidades.

Un segundo tipo de restricciones se define según si las entidades pueden pertenecer a más de un conjunto de entidades de nivel más bajo en una generalización simple. Los conjuntos de entidades de nivel más bajo pueden ser uno de los siguientes:

- **Disjunto.** Una *restricción sobre el carácter disjunto* requiere que una entidad no pertenezca a más de un conjunto de entidades de nivel más bajo. En el ejemplo, una entidad *cuenta* puede satisfacer sólo una condición para el atributo *tipo-cuenta*; una entidad puede ser bien una cuenta de ahorro o bien una cuenta corriente, pero no ambas cosas a la vez.
- **Solapado.** En las *generalizaciones solapadas*, la misma entidad puede pertenecer a más de un conjunto de entidades de nivel más bajo en una generalización simple. Como ilustración, tomando el ejemplo del grupo de trabajo del empleado, asúmase que ciertos directores participan en más de un grupo de trabajo. Un empleado dado puede, por lo tanto, aparecer en más de uno de los conjuntos de entidades grupo que son conjuntos de entidades de nivel más bajo de *empleado*. Así, la generalización es solapada.

Como otro ejemplo, supóngase la generalización aplicada a los conjuntos de entidades *cliente* y *empleado* conduce a un conjunto de entidades de nivel más alto *persona*. La generalización está solapada si un empleado también puede ser un cliente.

La entidad de nivel más bajo solapada es el caso predeterminado; la restricción sobre el carácter disjunto se debe colocar explícitamente en una generali-

zación (o especialización). Se puede identificar una restricción sobre el carácter disjunto en un diagrama E-R añadiendo la palabra *disjunto* en el símbolo del triángulo.

Una restricción final, la **restricción de completitud** en una generalización o especialización, especifica si un conjunto de entidades de nivel más alto debe pertenecer o no a al menos a uno de los conjuntos de entidades de nivel más bajo en una generalización/especialización. Esta restricción puede ser una de las siguientes:

- **Generalización o especialización total.** Cada entidad de nivel más alto debe pertenecer a un conjunto de entidades de nivel más bajo.
- **Generalización o especialización parcial.** Algunas entidades de nivel más alto pueden no pertenecer a algún conjunto de entidades de nivel más bajo.

La generalización parcial es la predeterminada. Se puede especificar una generalización total en un diagrama E-R usando una línea doble para conectar el rectángulo que representa el conjunto de entidades de nivel más alto con el símbolo del triángulo (esta notación es similar a la notación de participación total en una relación).

La generalización de *cuenta* es total: todas las entidades *cuenta* deben ser o bien cuentas de ahorro o bien cuentas corrientes. Debido a que el conjunto de entidades de nivel más alto alcanzado a través de la generalización está generalmente compuesta únicamente por aquellas entidades del conjunto de entidades de nivel más bajo, la restricción de completitud para un conjunto de entidades de nivel más alto generalizado es habitualmente total. Cuando la restricción es parcial, la entidad de nivel más alto no aparece necesariamente en el conjunto de entidades de nivel más bajo. Los conjuntos de entidades grupo de trabajo ilustran una especialización parcial. Como los empleados se asignan a grupos sólo después de llevar tres meses en el trabajo, algunas entidades *empleado* pueden no ser miembros de ningún conjunto de entidades grupo de nivel más bajo.

Los conjuntos de entidades equipo se pueden caracterizar más completamente como una 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 el carácter disjunto, sin embargo, no dependen una de la otra. Los patrones de restricciones pueden ser también parcial-disjunta y total-solapada.

Se puede ver que ciertos requisitos de inserción y borrado son consecuencia de las restricciones que se aplican a una generalización o especialización dada. Por ejemplo, cuando se coloca una restricción de completitud total, una entidad insertada en un conjunto de enti-

dades de nivel más alto se debe insertar en al menos uno de los conjuntos de entidades de nivel más bajo. Con una restricción de definición por condición, todas las entidades de nivel más alto que satisfacen la condición se deben insertar en el conjunto de entidades de nivel más bajo. Finalmente, una entidad que se borra de un conjunto de entidades de nivel más alto, también se debe borrar de todos los conjuntos de entidades de nivel más bajo asociados a los que pertenezca.

### 2.7.5. Agregación

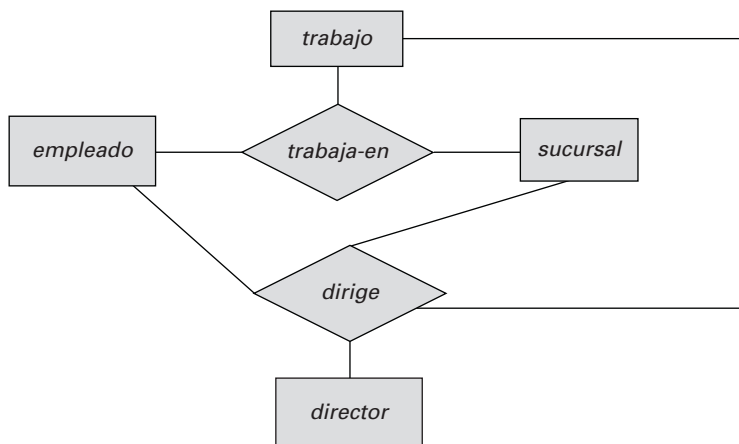
Una limitación del modelo E-R es que no resulta posible expresar relaciones entre relaciones. Para ilustrar la necesidad de tales construcciones considérese la relación ternaria *trabaja-en*, que se vio anteriormente, entre *empleado*, *sucursal* y *trabajo* (véase la Figura 2.13). Supóngase ahora que se desean registrar los directores para las tareas realizadas por un empleado en una sucursal; es decir, se desean registrar directores por combinaciones (*empleado*, *sucursal*, *trabajo*). Asíase que existe una entidad *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 un empleado que están dirigidas por un director). Al usar los constructores básicos del modelado E-R se obtiene el diagrama E-R de la Figura 2.18 (por simplicidad se han omitido los atributos).

Parece que los conjuntos de relaciones *trabaja-en* y *dirige* se pueden combinar en un único conjunto de relaciones. No obstante, no se deberían combinar, dado que algunas combinaciones *empleado*, *sucursal*, *trabajo* puede que no tengan director.

Hay información redundante en la figura resultante, ya que cada combinación *empleado*, *sucursal*, *trabajo* en *dirige* también lo 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 esto implica que es más difícil (tanto lógicamente como en coste de ejecución) encontrar, por ejemplo, los triples empleado-sucursal-trabajo de los que un director es responsable. Como el director es una entidad *director*, se descarta esta alternativa en cualquier caso.

La mejor forma de modelar una situación como ésta es usar la agregación. La **agregación** es una abstracción a través de la cual las relaciones se tratan como entidades de nivel más alto. Así, para este ejemplo, se considera el conjunto de relaciones *trabaja-en* (que relaciona los conjuntos de entidades *empleado*, *sucursal* y *trabajo*) como un conjunto de entidades de nivel más alto denominado *trabaja-en*. Tal conjunto de entidades se trata de la misma forma que cualquier otro conjunto de entidades. Se puede crear entonces una relación binaria *dirige* entre *trabaja-en* y *director* para representar



**FIGURA 2.18.** Diagrama E-R con relaciones redundantes.

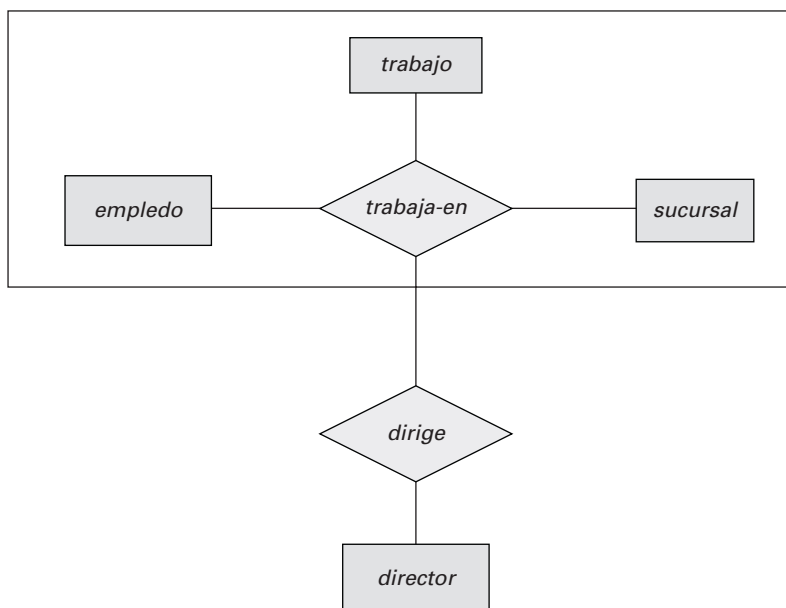
quién dirige las tareas. En la Figura 2.19 se muestra una notación para la agregación que se usa habitualmente para esta situación.

#### 2.7.6. Notaciones E-R alternativas

La Figura 2.20 resume el conjunto de símbolos que hemos usado en los diagramas E-R. No hay ningún estándar universal para la notación de los diagramas E-R y diferentes libros y diferente software de diagramas E-R usan notaciones diferentes; la Figura 2.21 indica alguna de las notaciones alternativas que se usan ampliamente. Un conjunto de entidades se puede representar como un cuadro con el nombre fuera, y los atributos listados unos

debajo de otros dentro del cuadro. Los atributos clave primaria se indican listándolos en la parte superior, con una línea separándolos de los otros atributos.

Las restricciones de cardinalidad se pueden indicar de varias formas como se muestra en la Figura 2.21. Las etiquetas \* y 1 en los arcos que salen de las relaciones se usan a menudo para denotar relaciones varios a varios, uno a uno y varios a uno como se muestra en la figura. En otra notación alternativa de la figura los conjuntos de relaciones se representan por líneas entre conjuntos de entidades sin rombos; sólo se pueden modelar de esta forma las relaciones binarias. Las restricciones de cardinalidad en esta notación se muestran por la notación «pata de gallo», como en la figura.



**FIGURA 2.19.** Diagrama E-R con agregación.

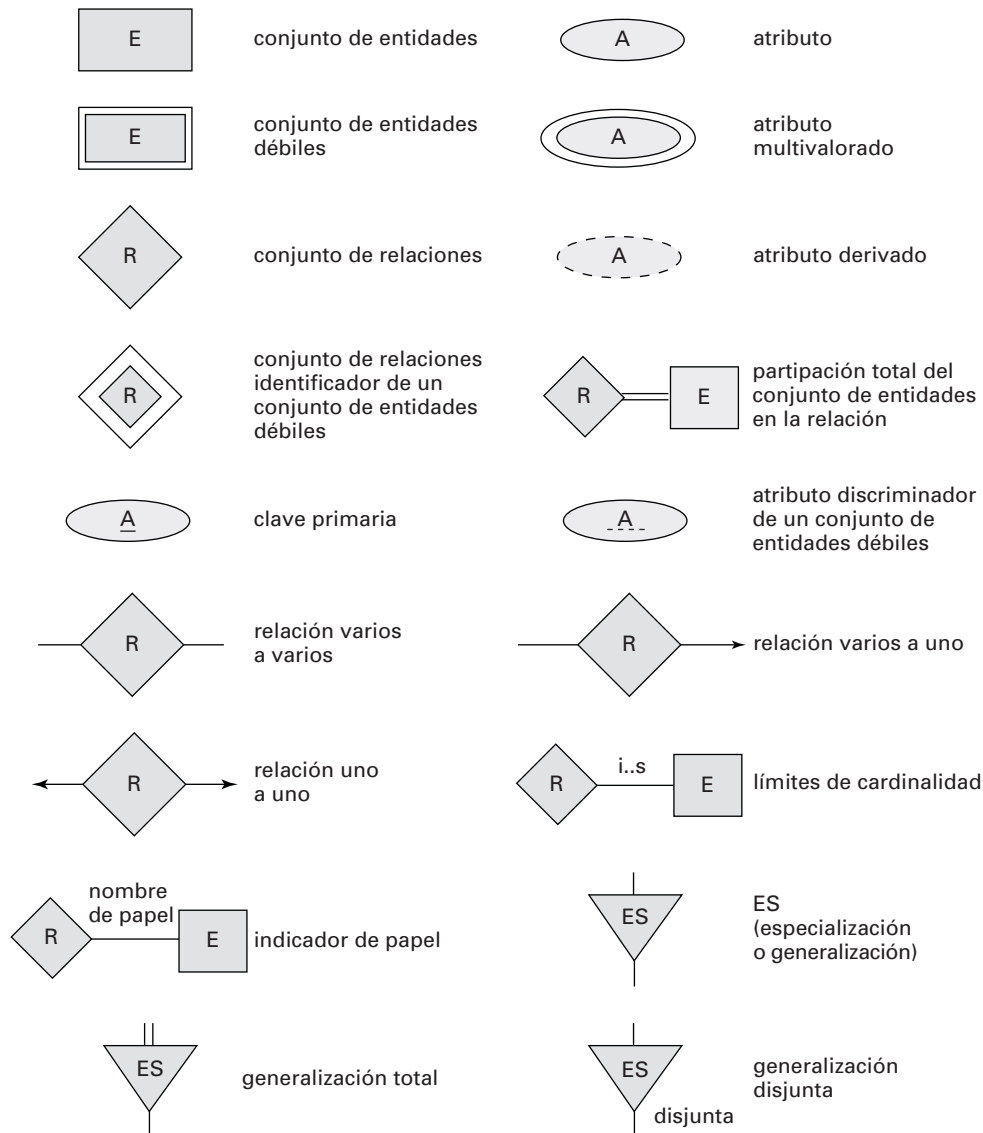


FIGURA 2.20. Símbolos usados en la notación E-R.

## 2.8. DISEÑO DE UN ESQUEMA DE BASE DE DATOS E-R

El modelo de datos E-R da una flexibilidad sustancial en el diseño de un esquema de bases de datos para modelar una empresa dada. En este apartado se considera cómo un diseñador de bases de datos puede seleccionar entre el amplio rango de alternativas. Entre las decisiones que se toman están las siguientes:

- Si se usa un atributo o un conjunto de entidades para representar un objeto (discutido anteriormente en el Apartado 2.2.1)
- Si un concepto del mundo real se expresa más exactamente mediante un conjunto de entidades o mediante un conjunto de relaciones (Apartado 2.2.2)
- Si se usa una relación ternaria o un par de relaciones binarias (Apartado 2.2.3)
- Si se usa un conjunto de entidades fuertes o débiles (Apartado 2.6); un conjunto de entidades fuertes y sus conjuntos de entidades débiles dependientes se pueden considerar como un «objeto» en la base de datos, debido a que la existencia de las entidades débiles depende de la entidad fuerte
- Si el uso de la generalización (Apartado 2.7.2) es apropiado; la generalización, o una jerarquía de relaciones ES, contribuye a la modularidad por permitir que los atributos comunes de conjuntos de entidades similares se representen en un único lugar en un diagrama E-R
- Si el uso de la agregación (Apartado 2.7.5) es apropiado; la agregación agrupa una parte de un dia-

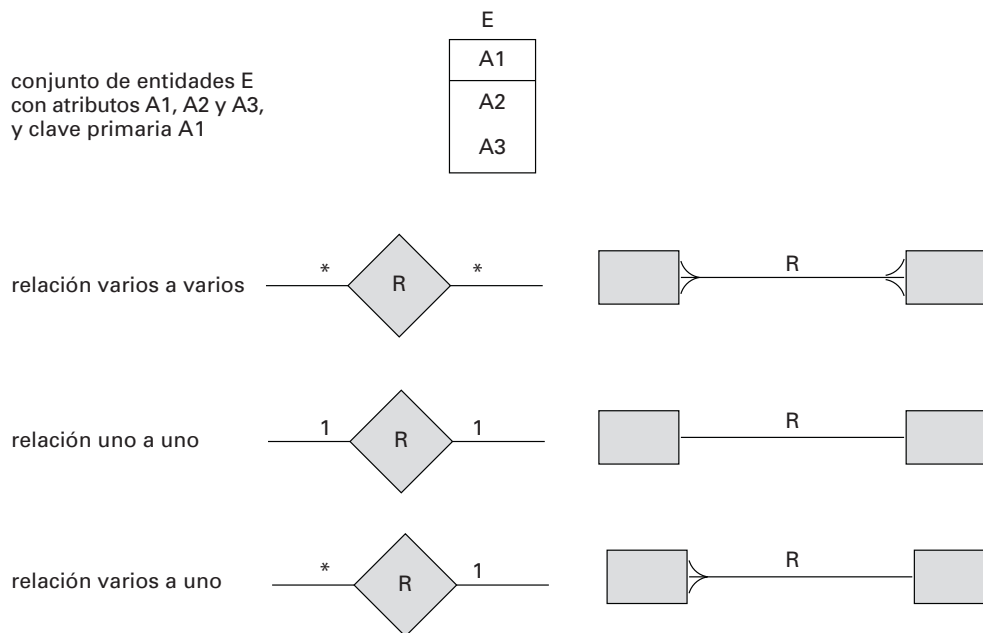


FIGURA 2.21. Notaciones E-R alternativas.

grama E-R en un único conjunto de entidades, permitiendo tratar el conjunto de entidades de la agregación como una unidad única sin importar los detalles de su estructura interna.

Se verá que el diseñador de bases de datos necesita un buen entendimiento de la empresa que se modela para tomar estas decisiones.

### 2.8.1. Fases de diseño

Un modelo de datos de alto nivel sirve al diseñador de la base de datos para proporcionar un marco conceptual en el que especificar de forma sistemática los requisitos de datos de los usuarios de la base de datos que existen, y cómo se estructurará la base de datos para completar estos requisitos. La fase inicial del diseño de bases de datos, por tanto, es caracterizar completamente las necesidades de datos esperadas por los usuarios de la base de datos. El resultado de esta fase es una *especificación de requisitos del usuario*.

A continuación, el diseñador elige un modelo de datos y, aplicando los conceptos del modelo de datos elegido, traduce estos requisitos a un esquema conceptual de la base de datos. El esquema desarrollado en esta fase de **diseño conceptual** proporciona una visión detallada del desarrollo. Debido a que sólo se ha estudiado el modelo E-R hasta ahora, se usará éste para desarrollar el esquema conceptual. En términos del modelo E-R, el esquema especifica todos los conjuntos de entidades, conjuntos de relaciones, atributos y restricciones de correspondencia. El diseñador revisa el esquema para confirmar que todos los requisitos de datos se satisfacen realmente y no hay conflictos entre sí. También se examina el diseño para eli-

minar características redundantes. Lo importante en este punto es describir los datos y las relaciones, más que especificar detalles del almacenamiento físico.

Un esquema conceptual completamente desarrollado indicará también los requisitos funcionales de la empresa. En una **especificación de requisitos funcionales** los usuarios describen los tipos de operaciones (o transacciones) que se realizarán sobre los datos. Algunos ejemplos de operaciones son la modificación o actualización de datos, la búsqueda y recuperación de datos específicos y el borrado de datos. En esta fase de diseño conceptual se puede hacer una revisión del esquema para encontrar los requisitos funcionales.

El proceso de trasladar un modelo abstracto de datos a la implementación de la base de datos consta de dos fases de diseño finales. 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 base de datos que se usará. El diseñador usa el esquema resultante específico a la base de datos en la siguiente **fase de diseño físico**, en la que se especifican las características físicas de la base de datos. Estas características incluyen la forma de organización de los archivos y las estructuras de almacenamiento interno, que se discutirán en el Capítulo 11.

En este capítulo se tratan sólo los conceptos del modelo E-R usados en la fase de diseño del esquema conceptual. Se ha presentado una breve visión del proceso de diseño de bases de datos para proporcionar un contexto para la discusión del modelo de datos E-R. El diseño de bases de datos recibe un tratamiento completo en el Capítulo 7.

En el Apartado 2.8.2 se aplican las dos fases iniciales de diseño de bases de datos al ejemplo del banco.



Se emplea el modelo de datos E-R para traducir los requisitos de usuario al esquema de diseño conceptual que se describe como un diagrama E-R.

## 2.8.2. Diseño de base de datos para el banco

Nos centramos ahora en los requisitos de diseño de la base de datos para el banco en más detalle y desarrollamos un diseño más realista, aunque también más complicado, de lo que se ha visto en los ejemplos anteriores. Sin embargo, no se intentará modelar cada aspecto del diseño de la base de datos para un banco; se considerarán sólo unos cuantos aspectos para ilustrar el proceso de diseño de bases de datos.

### 2.8.2.1. Requisitos de datos

La especificación inicial de los requisitos de usuario se puede basar en entrevistas con los usuarios de la base de datos y en el análisis propio del diseñador del desarrollo. La descripción que surge de esta fase de diseño sirve como base para especificar la estructura conceptual de la base de datos. La siguiente lista describe los principales requisitos del banco:

- El banco está organizado en sucursales. Cada sucursal está ubicada en una ciudad particular y se identifica por un nombre único. El banco supervisa los activos de cada sucursal.
- Los clientes del banco se identifican mediante sus valores de *id-cliente*. El banco almacena cada nombre de cliente, y la calle y ciudad donde viven los clientes. Los clientes pueden tener cuentas y pueden pedir préstamos. Un cliente puede estar asociado con un banquero particular, que puede actuar como responsable de préstamos o banquero personal para un cliente.
- Los empleados del banco se identifican mediante sus valores de *id-empleado*. La administración del banco almacena el nombre y número de teléfono de cada empleado, los nombres de los subordinados del empleado, y el número *id-empleado* del jefe del empleado. El banco también mantiene registro de la fecha de comienzo del contrato del empleado, así como su antigüedad.
- El banco ofrece dos tipos de cuentas: cuentas de ahorro y cuentas corrientes. Las cuentas pueden asociarse a más de un cliente y un cliente puede tener más de una cuenta. Cada cuenta está asignada a un único número de cuenta. El banco mantiene un registro del saldo de cada cuenta y la fecha más reciente en que la cuenta fue accedida por cada cliente que mantiene la cuenta. Además, cada cuenta de ahorro tiene un tipo de interés y para cada cuenta corriente se almacena el descubierto.
- Un préstamo tiene lugar en una sucursal particular y puede estar asociado a uno o más clientes. Un préstamo se identifica mediante un único número

de préstamo. Para cada préstamo el banco mantiene registro del importe del préstamo y de los pagos del préstamo. Aunque un número de pago del préstamo no identifica de forma única un pago entre todos los préstamos del banco, un número de pago identifica un pago particular para un préstamo específico. Para cada pago se almacenan la fecha y el importe.

En un desarrollo de un banco real, el banco mantendría información de los abonos y cargos en las cuentas de ahorros y en las cuentas corrientes, igual que se mantiene registro de los pagos para los préstamos. Debido a que los requisitos del modelo para este seguimiento son similares, y para mantener nuestro ejemplo reducido, en este modelo no se mantiene un seguimiento de tales abonos y cargos.

### 2.8.2.2. Designación de los conjuntos de entidades

La especificación de los requisitos de datos sirve como punto de partida para la construcción de un esquema conceptual para la base de datos. Desde la especificación listada en el Apartado 2.8.2.1 se comienzan a identificar los conjuntos de entidades y sus atributos.

- El conjunto de entidades *sucursal*, con los atributos *nombre-sucursal*, *ciudad-sucursal* y *activo*.
- El conjunto de entidades *cliente*, con los atributos *id-cliente*, *nombre-cliente*, *calle-cliente* y *ciudad-cliente*. Un posible atributo adicional es *nombre-banquero*.
- El conjunto de entidades *empleado*, con los atributos *id-empleado*, *nombre-empleado*, *número-teléfono*, *suelo* y *jefe*. Algunas características descriptivas adicionales son el atributo multivalorado *nombre-subordinado*, el atributo base *fecha-comienzo* 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*.

### 2.8.2.3. Designación de los conjuntos de relaciones

Volviendo ahora al esquema de diseño rudimentario del Apartado 2.8.2.2 se especifican los siguientes conjuntos de relaciones y correspondencia de cardinalidades:

- *prestatario*, un conjunto de relaciones varios a varios entre *cliente* y *préstamo*.

- *préstamo-sucursal*, un conjunto de relaciones varios a uno que indica la sucursal en que se ha originado un préstamo. Nótese que este conjunto de relaciones reemplaza 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*, indicando que un cliente posee una cuenta.
- *banquero-consejero*, con el atributo de relación *tipo*, un conjunto de relaciones varios a uno que expresa que un cliente puede ser aconsejado por un empleado del banco, y que un empleado del

banco puede aconsejar a uno o más clientes. Nótese que este conjunto de relaciones ha reemplazado al atributo *nombre-banquero* del conjunto de entidades *cliente*.

- *trabaja-para*, un conjunto de relaciones entre entidades *empleado* con papeles que indican *jefe* y *trabajador*; la correspondencia de cardinalidades expresa que un empleado trabaja para un único jefe, y que un jefe supervisa uno o más empleados. Nótese que este conjunto de relaciones reemplaza el atributo *jefe* de *empleado*.

#### 2.8.2.4. Diagrama E-R

Conforme a lo discutido en el Apartado 2.8.2.3 se presenta ahora el diagrama E-R completo para el ejemplo del banco. En la Figura 2.22 se muestra la representación

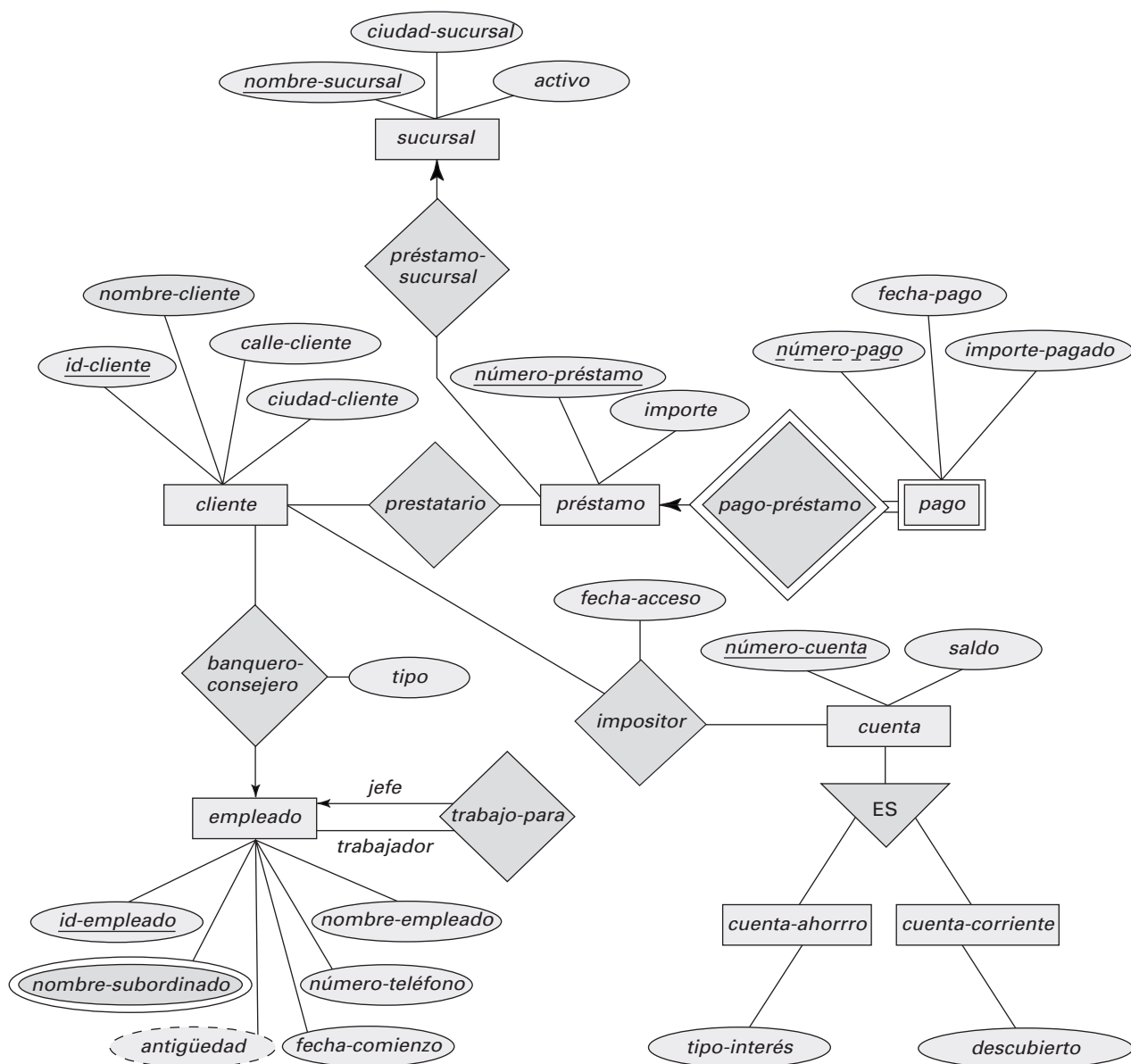


FIGURA 2.22. Diagrama E-R para un banco.

completa de un modelo conceptual de un banco, expresada en términos de los conceptos E-R. El diagrama incluye los conjuntos de entidades, atributos, conjuntos de

relaciones, y correspondencia de cardinalidades alcanzados a través del proceso de diseño de los Apartados 2.8.2.1 y 2.8.2.2, y refinados en el Apartado 2.8.2.3.

## 2.9. REDUCCIÓN DE UN ESQUEMA E-R A TABLAS

Una base de datos que se ajusta a un esquema de bases de datos E-R se puede representar por una colección de tablas. Para cada conjunto de entidades de la base de datos y para cada conjunto de relaciones de la base de datos hay una única tabla a la que se asigna el nombre del conjunto de entidades o del conjunto de relaciones correspondiente. Cada tabla tiene varias columnas, cada una de las cuales tiene un nombre único.

Los modelos E-R y el de bases de datos relacionales son representaciones abstractas y lógicas de empresas del mundo real. Debido a que los dos modelos emplean principios de diseño similares, se puede convertir un diseño E-R en un diseño relacional. Convertir una representación de bases de datos de un diagrama E-R a un formato de tablas es la base para la derivación de un diseño de bases de datos relacional desde un diagrama E-R. Aunque existen diferencias importantes entre una relación y una tabla, una relación se puede considerar informalmente como una tabla de valores.

En este apartado se describe cómo se puede representar un esquema E-R mediante tablas; y en el Capítulo 3 se muestra cómo generar un esquema de bases de datos relacional a partir de un esquema E-R.

Las restricciones especificadas en un diagrama E-R, tales como las claves primarias y las restricciones de cardinalidad, se corresponden con restricciones sobre las tablas generadas a partir del diagrama E-R. Se proporcionan más detalles sobre esta correspondencia en el Capítulo 6 después de describir cómo especificar restricciones sobre tablas.

### 2.9.1. Representación tabular 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 una tabla llamada  $E$  con  $n$  columnas distintas, cada una de las cuales corresponde a uno de los atributos de  $E$ . Cada fila de la tabla corresponde a una entidad del conjunto de entidades  $E$ . (En los apartados 2.9.4 y 2.9.5 se describe cómo manejar los atributos compuestos y multivalorados.)

Como ilustración considérese el conjunto de entidades *préstamo* del diagrama E-R mostrado en la Figura 2.8. Este conjunto de entidades tiene dos atributos: *número-préstamo* e *importe*. Se representa este conjunto de entidades mediante una tabla llamada *préstamo*, con dos columnas, como se muestra en la Figura 2.23. La fila

(P-17,1.000)

número-préstamo	importe
P-11	900
P-14	1.500
P-15	1.500
P-16	1.300
P-17	1.000
P-23	2.000
P-93	500

FIGURA 2.23. La tabla *préstamo*.

de la tabla *préstamo* significa que el número de préstamo P-17 tiene un importe de préstamo de 1.000 €. Se puede añadir una nueva entidad a la base de datos insertando una fila en una tabla. También se pueden borrar o modificar las filas.

$D_1$  denota el conjunto de todos los números de préstamo y  $D_2$  denota el conjunto de todos los saldos. Cualquier fila de la tabla *préstamo* debe consistir en una tupla  $(v_1, v_2)$ , donde  $v_1$  es un número de préstamo (es decir,  $v_1$  está en el conjunto  $D_1$ ) y  $v_2$  es un importe (es decir,  $v_2$  está en el conjunto  $D_2$ ). En general, la tabla *préstamo* contendrá sólo un subconjunto del conjunto de todas las filas posibles. El conjunto de todas las filas posibles de *préstamo* es el *producto cartesiano* de  $D_1$  y  $D_2$ , denotado por

$$D_1 \times D_2$$

En general, si se tiene una tabla de  $n$  columnas, se denota el producto cartesiano de  $D_1, D_2, \dots, D_n$  por

$$D_1 \times D_2 \times \dots \times D_{n-1} \times D_n$$

Como otro ejemplo considérese el conjunto de entidades *cliente* del diagrama E-R mostrado en la Figura 2.8. Este conjunto de entidades tiene los atributos *id-cliente*, *nombre-cliente*, *calle-cliente* y *ciudad-cliente*. La tabla correspondiente a *cliente* tiene cuatro columnas, como se muestra en la Figura 2.24.

### 2.9.2. Representación tabular 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. Sea la clave primaria de  $B$  el conjunto de atributos  $b_1, b_2, \dots, b_n$ . Se representa el conjunto de entidades  $A$  mediante una tabla llamada  $A$  con una columna por cada uno de los atributos del conjunto:

<i>id-cliente</i>	<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
01.928.374	Gómez	Carretas	Cerceda
18.273.609	Abril	Preciados	Valsain
19.283.746	González	Arenal	La Granja
24.466.880	Pérez	Carretas	Cerceda
32.112.312	Santos	Mayor	Peguerinos
33.557.799	Fernández	Jazmín	León
33.666.999	Rupérez	Ramblas	León
67.789.901	López	Mayor	Peguerinos
96.396.396	Valdivieso	Goya	Vigo

FIGURA 2.24. La tabla *cliente*.

$$\{a_1, a_2, \dots, a_m\} \cup \{b_1, b_2, \dots, b_n\}$$

Como ilustración considérese el conjunto de entidades *pago* mostrado en el diagrama E-R de la Figura 2.16. 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*. Así, *pago* se representa mediante una tabla con cuatro columnas etiquetadas con *número-préstamo*, *número-pago*, *fecha-pago* e *importe-pago*, como se describe en la Figura 2.25.

### 2.9.3. Representación tabular de los conjuntos de relaciones

Sea  $R$  un conjunto de relaciones, sean  $a_1, a_2, \dots, a_m$  el conjunto de atributos formados 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 una tabla llamada  $R$  con una columna por cada uno de los atributos del conjunto:

$$\{a_1, a_2, \dots, a_m\} \cup \{b_1, b_2, \dots, b_n\}$$

Como ilustración considérese el conjunto de relaciones *prestatario* del diagrama E-R de la Figura 2.8. Este conjunto de relaciones involucra los dos siguientes conjuntos de entidades:

- *cliente*, con la clave primaria *id-cliente*.
- *préstamo*, con la clave primaria *número-préstamo*.

Debido a que el conjunto de relaciones no tiene atributos, la tabla *prestatario* tiene dos columnas etiquetadas *id-cliente* y *número-préstamo*, como se muestra en la Figura 2.22.

#### 2.9.3.1. Redundancia de tablas

Un conjunto de relaciones uniendo un conjunto de entidades débiles con el correspondiente conjunto de entidades fuertes es un caso especial. Como se hizo notar en el Apartado 2.6, estas relaciones son varios a uno y no tienen atributos descriptivos. Además, la clave primaria de un conjunto de entidades débiles incluye la clave primaria del conjunto de entidades fuertes. En el diagrama E-R de la Figura 2.16, el conjunto de entidades débiles *pago* depende del conjunto de entidades fuertes *préstamo* a través del conjunto de relaciones *pago-préstamo*. La clave primaria de *pago* es  $\{\text{número-préstamo}, \text{número-pago}\}$  y la clave primaria de *préstamo* es  $\{\text{número-préstamo}\}$ . Como *pago-préstamo* no tiene atributos descriptivos, la tabla para *pago-préstamo* tendría dos columnas, *número-préstamo* y *número-pago*. La tabla para el conjunto de entidades *pago* tiene cuatro columnas, *número-préstamo*, *número-pago*, *fecha-pago* e *importe-pago*. Cada combinación (*número-préstamo*, *número-pago*) en *pago-préstamo* también se encontraría en la tabla *pago*, y viceversa. Por tanto, la tabla *pago-préstamo* es redundante. En general, la tabla para el conjunto de relaciones que une un conjunto de entidades débiles con su correspondiente conjunto de entidades fuertes es redundante y no necesita estar presente en una representación tabular de un diagrama E-R.

<i>número-préstamo</i>	<i>número-pago</i>	<i>fecha-pago</i>	<i>importe-pago</i>
P-11	53	7 junio 2001	125
P-14	69	28 mayo 2001	500
P-15	22	23 mayo 2001	300
P-16	58	18 junio 2001	135
P-17	5	10 mayo 2001	50
P-17	6	7 junio 2001	50
P-17	7	17 junio 2001	100
P-23	11	17 mayo 2001	75
P-93	103	3 junio 2001	900
P-93	104	13 junio 2001	200

FIGURA 2.25. La tabla *pago*.

<i>id-cliente</i>	<i>número-préstamo</i>
01.928.374	P-11
01.928.374	P-23
24.466.880	P-93
32.112.312	P-17
33.557.799	P-16
55.555.555	P-14
67.789.901	P-15
96.396.396	P-17

FIGURA 2.26. La tabla *prestatario*.

### 2.9.3.2. Combinación de tablas

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 tablas descrito previamente se consiguen tres tablas: *A*, *B* y *AB*. Supóngase además que la participación de *A* en la relación es total; es decir, cada entidad *a* en el conjunto de entidades *A* debe participar en la relación *AB*. Entonces se pueden combinar las tablas *A* y *AB* para formar una única tabla consistente en la unión de las columnas de ambas tablas.

Como ilustración considérese el diagrama E-R de la Figura 2.27. La doble línea del diagrama E-R indica que la participación de *cuenta* en *cuenta-sucursal* es total. Así, una cuenta no puede existir sin estar asociada con una sucursal particular. Además, el conjunto de relaciones *cuenta-sucursal* es varios a uno desde *cuenta* a *sucursal*. Por lo tanto, se puede combinar la tabla para *cuenta-sucursal* con la tabla para *cuenta* y se necesitan sólo las dos tablas siguientes:

- *cuenta*, con los atributos *número-cuenta*, *saldo* y *nombre-cuenta*
- *sucursal*, con los atributos *nombre-sucursal*, *ciudad-sucursal* y *activo*

En el caso de relaciones uno a uno, la tabla del conjunto de relaciones se puede combinar con las tablas de cualquiera de los conjuntos de entidades. Las tablas se pueden combinar incluso si la participación es parcial usando valores nulos; en el ejemplo anterior se usarían valores nulos para el atributo *nombre-sucursal* para las cuentas que no tengan una sucursal asociada.

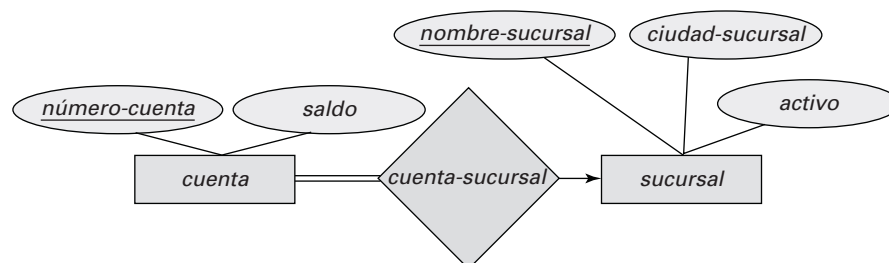


FIGURA 2.27. Diagrama E-R.

### 2.9.4. Atributos compuestos

Los atributos compuestos se manejan creando un atributo separado para cada uno de los atributos componentes; no se crea una columna separada para el propio atributo compuesto. Supóngase que *dirección* es un atributo compuesto del conjunto de entidades *cliente* y que los componentes de *dirección* son *ciudad* y *calle*. La tabla generada de cliente contendría las columnas *calle-dirección* y *ciudad-dirección*; no hay una columna separada para *dirección*.

### 2.9.5. Atributos multivalorados

Se ha visto que los atributos en un diagrama E-R generalmente se asocian directamente en columnas para las tablas apropiadas. Los atributos multivalorados, sin embargo, son una excepción; para estos atributos se crean tablas nuevas.

Para un atributo multivalorado *M* se crea una tabla *T* con una columna *C* que corresponde a la clave primaria del conjunto de entidades o conjunto de relaciones del que *M* es atributo. Como ilustración considérese el diagrama E-R de la Figura 2.22. El diagrama incluye el atributo multivalorado *nombre-subordinado*. Para este atributo multivalorado se crea una tabla *nombre-subordinado* con columnas *nombres*, referenciando al atributo *nombre-subordinado* de *empleado*, e *id-empleado*, representado la clave primaria del conjunto de entidades *empleado*. Cada subordinado de un empleado se representa como una única fila en la tabla.

### 2.9.6. Representación tabular de la generalización

Hay dos métodos diferentes para transformar a forma tabular un diagrama E-R que incluya generalización. Aunque la generalización a la que se va a hacer referencia es la de la Figura 2.17, para simplificar esta discusión se incluye sólo la primera capa de los conjuntos de entidades de nivel más bajo —es decir, *empleado* y *cliente*. Se asume que *nombre* es la clave primaria de *persona*.

1. Crear una tabla para el conjunto de entidades de nivel más alto. Para cada conjunto de entidades



de nivel más bajo, crear una tabla que incluya una columna para cada uno de los atributos de ese conjunto de entidades más una columna por cada atributo de la clave primaria del conjunto de entidades de nivel más alto. Así, para el diagrama E-R de la Figura 2.15, se tienen tres tablas:

- *persona*, con atributos *nombre*, *calle* y *ciudad*
- *empleado*, con atributos *nombre* y *salario*
- *cliente*, con atributos *nombre*, *límite-crédito*

2. Es posible una representación alternativa si la generalización es disjunta y completa —es decir, si no hay ninguna entidad que sea miembro de dos conjuntos de entidades de menor nivel directamente debajo de un conjunto de entidades de nivel más alto, y si cada entidad del conjunto de entidades de nivel más alto también pertenece a uno de los conjuntos de entidades de nivel más bajo. Aquí no se crea una tabla para el conjunto de entidades de nivel más alto. En su lugar, para cada conjunto de entidades de nivel más bajo se crea una tabla que incluya una columna por cada atributo del conjunto de entidades más una columna por *cada* atributo del conjunto de entidades de nivel más alto. Entonces, para el diagrama E-R de la Figura 2.15 se tienen dos tablas.

- *empleado*, con atributos *nombre*, *calle*, *ciudad* y *sueldo*

- *cliente*, con atributos *nombre*, *calle*, *ciudad* y *límite-crédito*

Las relaciones *cuenta-ahorro* y *cuenta-corriente* correspondientes a esas tablas tienen *número-cuenta* como clave primaria.

Si se usara el segundo método para una generalización solapada, algunos valores se almacenarían varias veces innecesariamente. Por ejemplo, si una persona es tanto empleado como cliente, los valores de *calle* y *ciudad* se almacenarían dos veces. Si la generalización no fuera completa (es decir, si alguna persona no fuera ni empleado ni cliente) entonces se necesitaría una tabla extra *persona* para representarlos.

### 2.9.7. Representación tabular de la agregación

Transformar a forma tabular un diagrama E-R que incluya agregación es sencillo. Considérese el diagrama de la Figura 2.19. La tabla para el conjunto de relaciones *dirige* entre la agregación de *trabaja-en* y el conjunto de entidades *director* incluye una columna para cada atributo de la clave primaria del conjunto de entidades *director* y del conjunto de relaciones *trabaja-en*. También incluiría una columna para los atributos descriptivos, si los hubiera, del conjunto de relaciones *dirige*. Por tanto, se transforman los conjuntos de relaciones y los conjuntos de entidades dentro de la entidad agregada.

## 2.10. EL LENGUAJE DE MODELADO UNIFICADO UML (*Unified Modeling Language*)\*\*

Los diagramas entidad-relación ayudan a modelar el componente de representación de datos de un sistema software. La representación de datos, sin embargo, sólo forma parte de un diseño completo de un sistema. Otros componentes son modelos de interacción del usuario con el sistema, especificación de módulos funcionales del sistema y su interacción, etc. El **lenguaje de modelado unificado** (UML, *Unified Modeling Language*) es un estándar propuesto para la creación de especificaciones de varios componentes de un sistema software. Algunas de las partes de UML son:

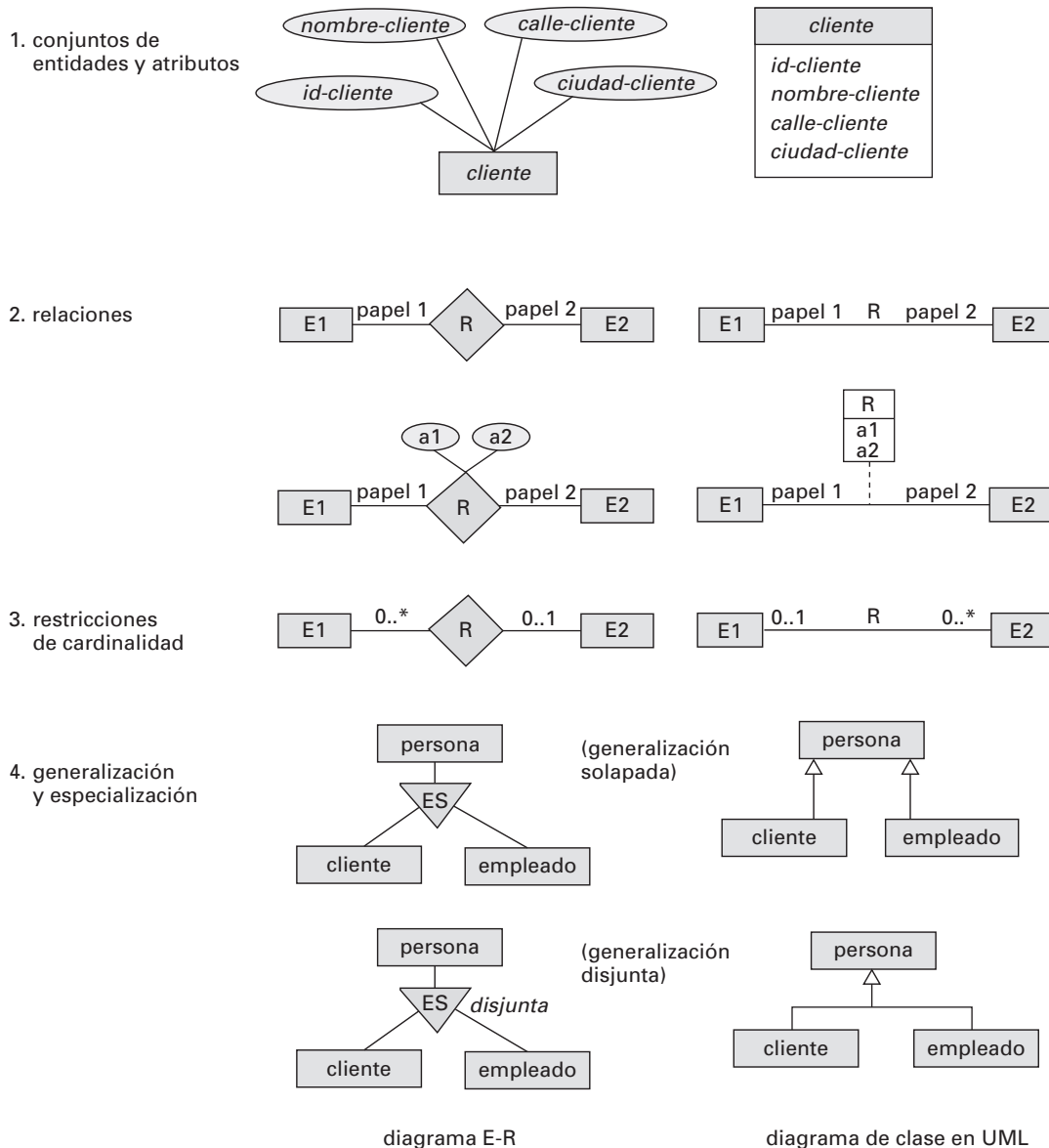
- **Diagrama de clase.** Un diagrama de clase es similar a un diagrama E-R. Más adelante en este apartado se mostrarán algunas características de los diagramas de clase y cómo se corresponden con los diagramas E-R.
- **Diagrama de caso de uso.** Los diagramas de caso de uso muestran la interacción entre los usuarios y el sistema, en particular los pasos de las tareas

que realiza el usuario (tales como prestar dinero o matricularse de una asignatura).

- **Diagrama de actividad.** Los diagramas de actividad describen el flujo de tareas entre varios componentes de un sistema.
- **Diagrama de implementación.** Los diagramas de implementación muestran los componentes del sistema y sus interconexiones tanto en el nivel del componente software como el hardware.

Aquí no se intentará proporcionar un tratamiento detallado de las diferentes partes de UML. Véanse las notas bibliográficas para encontrar referencias de UML. En su lugar se ilustrarán algunas características de UML mediante ejemplos.

La Figura 2.28 muestra varios constructores de diagramas E-R y sus constructores equivalentes de los diagramas de clase UML. Más abajo se describen estos constructores. UML muestra los conjuntos de entidades como cuadros y, a diferencia de E-R, muestra los atributos dentro del cuadro en lugar de como elipses sepa-



**FIGURA 2.28.** Símbolos usados en la notación de diagramas de clase UML.

radas. UML modela realmente objetos, mientras que E-R modela entidades. Los objetos son como entidades y tienen atributos, pero además proporcionan un conjunto de funciones (denominadas métodos) que se pueden invocar para calcular valores en términos de los atributos de los objetos, o para modificar el propio objeto. Los diagramas de clase pueden describir métodos además de atributos. Los objetos se tratan en el Capítulo 8.

Los conjuntos de relaciones binarias se representan en UML dibujando simplemente una línea que conecte los conjuntos de entidades. Se escribe el nombre del conjunto de relaciones adyacente a la línea. También se puede especificar el papel que juega un conjunto de entidades en un conjunto de relaciones escribiendo el nombre del papel en un cuadro, junto con los atributos del conjunto de relaciones, y conectar el cuadro con una línea discontinua a la línea que describe el conjunto de

relaciones. Este cuadro se puede tratar entonces como un conjunto de entidades, de la misma forma que una agregación en los diagramas E-R puede participar en relaciones con otros conjuntos de entidades.

La relaciones no binarias no se pueden representar directamente en UML —se deben convertir en relaciones binarias por la técnica que se describió en el Apartado 2.4.3.

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 mínimo y  $s$  el máximo número de relaciones en que puede participar una entidad. Sin embargo, se debería ser consciente que la ubicación de las restricciones es exactamente el inverso de la ubicación de las restricciones en los diagramas E-R, como muestra la Figura 2.28. 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 como 1 o \* se pueden escribir en los arcos; el valor 1 sobre un arco se trata equivalentemente como 1..1, mientras que \* es equivalente a 0..\*.

La generalización y especialización se representan en el diagrama E-R conectando conjuntos de entidades por una línea con un triángulo al final correspondiente al conjunto de entidades más general. Por ejem-

plo, el conjunto de entidades *persona* es una generalización de *cliente* y *empleado*. Los diagramas UML también pueden representar explícitamente las restricciones de generalizaciones disjuntas y solapadas. La Figura 2.28 muestra generalizaciones disjuntas y solapadas de *cliente* y *empleado* a *persona*. Recuérdese que se la generalización de *cliente* / *empleado* a *persona* es disjunta, y significa que ninguna entidad puede ser a la vez un *cliente* y un *empleado*. Una generalización solapada permite que una persona sea tanto *cliente* como *empleado*.

## 2.11. RESUMEN

- El modelo de datos **entidad-relación (E-R)** se basa en una percepción del mundo real consistente en un conjunto de objetos básicos llamados **entidades** y en **relaciones** entre esos objetos.
- El modelo está pensado principalmente para el proceso de diseño de la base de datos. Fue desarrollado para facilitar el diseño permitiendo la especificación de un **esquema de la empresa**. Tal 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 y es distinguible de otros objetos. Se expresa la distinción asociando con 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 relaciones del mismo tipo y un **conjunto de entidades** es una colección de entidades del mismo tipo.
- La **correspondencia de cardinalidades** expresa el número de entidades a las que otra entidad se puede asociar a través de un conjunto de relaciones.
- Una **superclave** de un conjunto de entidades es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar unívocamente una entidad en un conjunto de entidades. Se elige una superclave mínima para cada conjunto de entidades de entre sus superclaves; la superclave mínima se denomina la **clave primaria** del conjunto de entidades. Análogamente, un conjunto de relaciones es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar unívocamente una relación en un 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.
- 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 contenido entre un conjunto de entidades de nivel más alto y uno o más conjuntos de entidades de nivel más bajo. La especialización es el resultado de tomar un subconjunto de un conjunto de entidades de nivel más alto para formar un conjunto de entidades de nivel más bajo. La generalización es el resultado de tomar la unión de dos o más conjuntos disjuntos de entidades (de nivel más bajo) para producir un conjunto de entidades de nivel más alto. Los atributos de los conjuntos de entidades de nivel más alto los heredan los conjuntos de entidades de nivel más bajo.
- 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 más alto, y pueden participar en las relaciones.
- Las diferentes características del modelo E-R ofrecen al diseñador de bases de datos numerosas decisiones de cómo representar mejor la empresa que se modela. Los conceptos y 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 conjuntos de entidades débiles, generalización, especialización o agregación. A menudo el diseñador debe sopesar las ventajas de un modelo simple y compacto frente a otros más precisos pero más completos.
- Una base de datos que se representa en un diagrama E-R se puede representar mediante una colección de tablas. Para cada conjunto de entidades y para cada conjunto de relaciones de la base de datos hay una única tabla a la que se le asigna el nombre del conjunto de entidades o del conjunto de relaciones correspondiente. Cada tabla tiene un número de columnas, cada una de las cuales tiene un nombre único. La conversión de una representación de base de datos en un diagrama E-R a un formato de tabla se basa en la deri-

vacación de un diseño de bases de datos relacional desde un diagrama E-R.

- El **lenguaje de modelado unificado (UML)** proporciona un medio gráfico de modelar varios compo-

nentes de un sistema software. El componente diagrama de clase de UML se basa en diagramas E-R. Sin embargo, hay algunas diferencias entre ambos que se deben tener presentes.

## TÉRMINOS DE REPASO

- Agregación
- Atributo derivado
- Atributos
- Atributos descriptivos
- Atributos monovalorados y multivalorados
- Atributos simples y compuestos
- Conjunto de entidades
- Conjunto de relaciones
- Conjunto de relaciones binario
- Conjunto de relaciones recursivo
- Conjuntos de entidades débiles y fuertes
  - Atributos discriminantes
  - Relaciones identificadoras
- Correspondencia de cardinalidad:
  - Relación uno a uno
  - Relación uno a varios
  - Relación varios a uno
  - Relación varios a varios
- Diagrama E-R
- Dominio
- Entidad
- Especialización y generalización
  - Superclase y subclase
  - Herencia de atributos
  - Herencia simple y múltiple
  - Pertenencia definida por condición y definida por el usuario
  - Generalización disjunta y solapada
- Grado de un conjunto de relaciones
- Lenguaje de modelado unificado (UML)
- Modelo de datos entidad-relación
- Papel
- Participación
  - Participación total
  - Participación parcial
- Relación
- Restricción de completitud
  - Generalización total y parcial
- Superclave, clave candidata y clave primaria
- Valor nulo

## EJERCICIOS

- 2.1. Explíquense las diferencias entre los términos clave primaria, clave candidata y superclave.
  - 2.2. Constrúyase un diagrama E-R para una compañía de seguros de coches cuyos clientes poseen uno o más coches. Cada coche tiene asociado un número de cero a cualquier valor que almacena el número de accidentes.
  - 2.3. Constrúyase un diagrama E-R para un hospital con un conjunto de pacientes y un conjunto de médicos. Asíciase con cada paciente un registro de las diferentes pruebas y exámenes realizados.
  - 2.4. Una oficina de registro de una universidad mantiene datos acerca de las siguientes entidades: (a) asignaturas, incluyendo el número, título, programa, y prerequisites; (b) ofertas de asignaturas, incluyendo número de asignatura, año, semestre, número de sección, profesor(es), horarios y aulas; (c) estudiantes, incluyendo id-estudiante, nombre y programa; y (d) profesores, incluyendo número de identificación, nombre, departamento y título. Además, la matrícula de los estudiantes en asignaturas y las notas concedidas a estudiantes en cada asignatura en la que están matriculados se deben modelar adecuadamente.
- Constrúyase un diagrama E-R para la oficina de registro. Documentense todas las decisiones que se hagan acerca de restricciones de correspondencia.
- 2.5. Considérese una base de datos usada para registrar las notas que obtienen los estudiantes en diferentes exámenes de diferentes ofertas de asignaturas.
    - a. Constrúyase un diagrama E-R que modele exámenes como entidades y use una relación ternaria para esta base de datos.
    - b. Constrúyase un diagrama E-R alternativo que use sólo una relación binaria entre *estudiantes* y *ofertas-asignaturas*. Asegúrese de que sólo existe una relación entre un par determinado estudiante y oferta-asignatura y de que aún se pueden representar las notas que obtiene un estudiante en diferentes exámenes de una oferta de una asignatura.
  - 2.6. Constrúyanse tablas apropiadas para cada uno de los diagramas E-R de los Ejercicios 2.2 al 2.4.

- 2.7. Diseñese un diagrama E-R para almacenar los logros de su equipo deportivo favorito. Se deberían almacenar los partidos jugados, los resultados de cada partido, los jugadores de cada partido y las estadísticas individuales de cada jugador para cada partido. Las estadísticas de resumen se deberían modelar como atributos derivados.
- 2.8. Extiéndase el diagrama E-R del ejercicio anterior para almacenar la misma información para todos los equipos de una liga.
- 2.9. Explíquense las diferencias entre conjunto de entidades débiles y fuertes.
- 2.10. Se puede convertir cualquier conjunto de entidades débiles en un conjunto de entidades fuertes simplemente añadiendo los atributos apropiados. ¿Por qué, entonces, se tienen conjuntos de entidades débiles?
- 2.11. Defínase el concepto de agregación. Propónganse ejemplos para los que este concepto es útil.
- 2.12. Considérese el diagrama de la Figura 2.29, que modela una librería en línea.
- Lístense los conjuntos de entidades y sus claves primarias.
  - Supóngase que la librería añade casetes de música y discos compactos a su colección. El mismo elemento musical puede estar presente en formato de casete o de disco compacto con diferentes precios. Extiéndase el diagrama E-R para modelar esta adición, ignorando el efecto sobre las cestas de la compra.

ción, ignorando el efecto sobre las cestas de la compra.

- Extiéndase ahora el diagrama E-R usando generalización para modelar el caso en que una cesta de la compra pueda contener cualquier combinación de libros, casetes de música o discos compactos.

- 2.13. Considérese un diagrama E-R en el que el mismo conjunto de entidades aparece varias veces. ¿Por qué está permitida esta redundancia, una mala práctica que se debería evitar siempre que sea posible?

- 2.14. Considérese una base de datos de una universidad para la planificación de las aulas para los exámenes finales. Esta base de datos se modelaría mediante un único conjunto de entidades *examen*, con atributos *nombre-asignatura*, *número-sección*, *número-aula* y *hora*. Alternativamente se podrían definir uno o más conjuntos de entidades, con conjuntos de relaciones para sustituir algunos de los atributos del conjunto de entidades *examen*, como

- asignatura* con atributos *nombre*, *departamento* y *número-a*
  - sección* con atributos *número-s* y *matriculados*, que es un conjunto de entidades débiles dependiente de *curso*.
  - aula* con atributos *número-a*, *capacidad* y *edificio*.
- Muéstrese en un diagrama E-R el uso de los tres conjuntos de entidades adicionales listados.

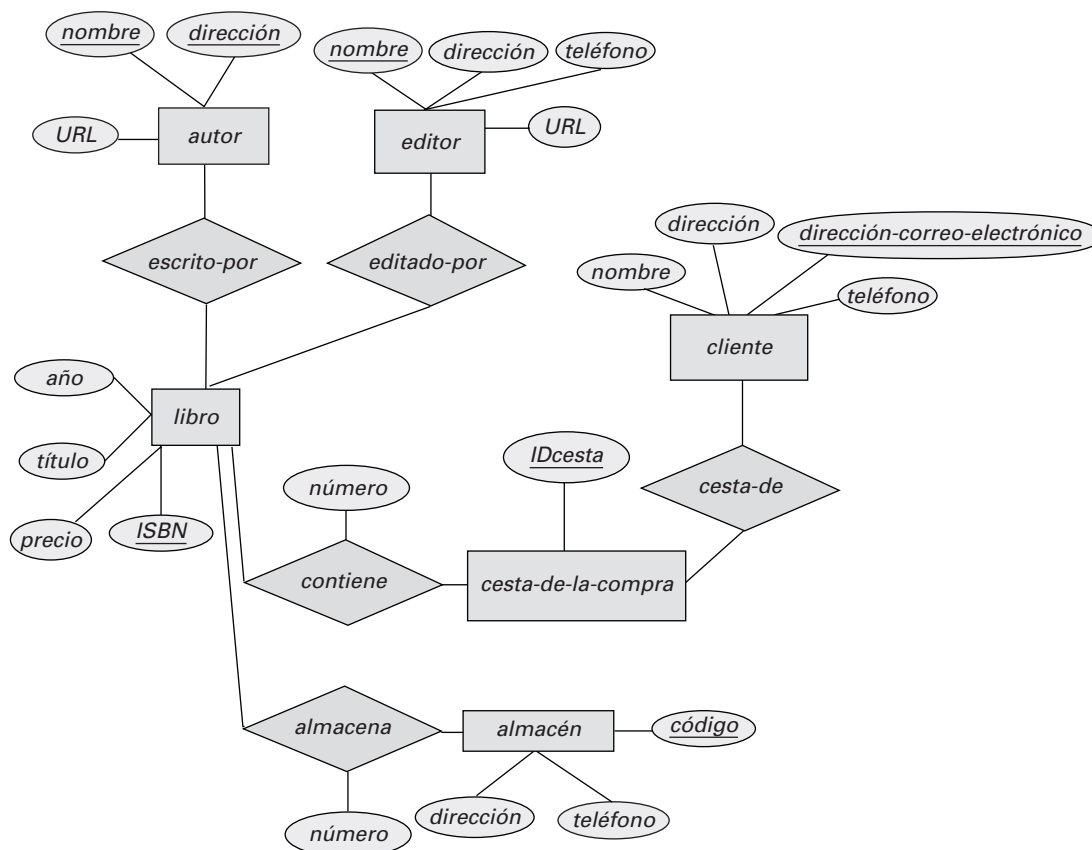


FIGURA 2.29. Diagrama E-R para el Ejercicio 2.12.