

## Capítulo 4: Ejemplo EER de un esquema, universidad, diseños y definiciones formales

---

Apartado realizado por el Sistema de Bibliotecas de la Universidad Andrés Bello con fines académicos. Autorizado según Ley N°20.435 artículo 71K.

Elmasri, R., Diaz Martín, J., Navathe, S. (2007). *Fundamentos de sistemas de bases de datos*. Madrid: Pearson Educación (pp.102-120)

## 4.5 Ejemplo EER de un esquema UNIVERSIDAD, diseños y definiciones formales

En esta sección empezaremos por dar un ejemplo de un esquema de base de datos en el modelo EER para ilustrar el uso de los distintos conceptos mostrados en este capítulo y en el anterior. A continuación, trataremos el tema del diseño para los esquemas conceptuales y terminaremos resumiendo los conceptos del modelo EER y los definiremos formalmente de la misma manera que hicimos con los del modelo ER en el Capítulo 3.

### 4.5.1 La base de datos UNIVERSIDAD

Para nuestro ejemplo, consideremos la base de datos UNIVERSIDAD que contiene información sobre los estudiantes y sus especialidades, traslados y registros, así como los cursos ofrecidos por la misma. Esta base de datos también almacena los proyectos de investigación patrocinados por la universidad y los estudiantes graduados. Este esquema se muestra en la Figura 4.9.

Para cada persona, la base de datos mantiene su [Nombre], [DNI], [Dirección], [Sexo] y [FechaNac]. Hay definidas dos subclases de la entidad PERSONA: PROFESOR y ESTUDIANTE. Los atributos específicos de PROFESOR son [Rango] (asistente, asociado, adjunto, investigador, etc.), [Oficina], [TlfOficina] y [Salario]. Todos los profesores están relacionados con el/los departamento/s a los que pertenecen: [PERTENECE] (un profesor puede estar asociado a varios de ellos, por lo que la relación es M:N). Un atributo específico de ESTUDIANTE es [Clase] (estudiante de primer año=1, estudiante de segundo año=2, . . . , graduado=5). Cada ESTUDIANTE está también relacionado con su especialidad y su formación secundaria, ([PRINCIPAL] y [SECUNDARIA]), de los niveles del curso en el que actualmente están matriculados [REGISTRADO], y de los cursos completados, [CERTIFICADO]. Cada instancia CERTIFICADO incluye la [Nota] que el estudiante recibe al completar ese nivel.

ESTUDIANTE\_GRADUADO es una subclase de ESTUDIANTE, con el predicado definido Clase = 5. Por cada estudiante graduado se mantiene una lista de las calificaciones anteriores en un atributo multivalor compuesto llamado [Licenciatura]. También relacionamos al estudiante graduado con un [TUTOR] y su [TRIBUNAL], en caso de existir.

Un departamento académico cuenta con los atributos nombre [NombreDpto], teléfono [TlfDpto] y número de oficina [NumOficina] y está relacionado con la persona que lo dirige [DIRECTOR] y la facultad a la que pertenece [PERTENECE\_A\_FACULTAD]. Los atributos de cada facultad son su nombre [NombreFacultad], su número de oficina [NumFacultad] y su [Decano].

Cada curso cuenta con los atributos número de curso [C#], el nombre [NombreCurso] y la descripción del mismo [DescripcionCurso]. Cada curso ofrece varios niveles, por lo que cada uno de estos niveles cuenta con un número de nivel [Niv#] y el año y el trimestre en el que se ofrece ([Año] y [Trimestre]).<sup>10</sup> Los números de nivel los identifican de forma única, y los que están siendo ofrecidos en el trimestre actual se encuentran en la subclase NIVEL\_ACTUAL de NIVEL, la cual cuenta con un predicado del tipo Trimestre = TrimActual y Año = AñoActual. Cada nivel está relacionado con el profesor que lo imparte ([INSTRUCTOR]) en caso de que se encuentre en la base de datos.

La categoría INSTRUCTOR\_INVESTIGADOR es un subconjunto de la unión de PROFESOR y ESTUDIANTE\_GRADUADO que incluye a todos los profesores, así como a los estudiantes graduados que están dedicados a la investigación o la enseñanza. Por último, la entidad SUBVENCIÓN contiene las subvenciones y los contratos adjudicados a la universidad. Cada subvención tiene los atributos [Título], número de subvención [No], la entidad adjudicataria [Agencia] y la fecha de inicio [FechaInicioSubvención], y está relacionado con un investigador principal [PI] y todos los [INVESTIGADORES\_A\_SU\_CARGO]. Cada instancia de esta última tiene como atributos la fecha en la que cada investigador empieza con su tarea [FechaInicioInvestigación], la

<sup>10</sup> Asumimos que en esta diversidad se está usando el sistema *trimestral* en lugar del *semestral*.



## 4.5.2 Consideraciones de diseño para la especialización/generalización

No siempre resulta sencillo elegir el diseño conceptual más apropiado para una base de datos. En la Sección 3.7.3, mostramos algunos de los problemas típicos a los que se suele enfrentar un diseñador de bases de datos cuando tiene que representar un tipo de entidad, una relación y los atributos en un esquema ER. En esta sección nos centraremos en las guías maestras para el diseño de la especialización/generalización y las categorías (tipos de unión) en un modelo EER.

Como ya se comentó en la Sección 3.7.3, el diseño conceptual de una base de datos debe considerarse como un proceso de refinamiento iterativo hasta llegar a lo que más se ajuste a nuestras necesidades. Las siguientes notas pueden ayudar a alcanzar este propósito:

- En general, se pueden definir muchas especializaciones y subclases para que el modelo conceptual sea fiel. Sin embargo, el inconveniente es que el diseño se vuelve algo confuso.
- Si una subclase tiene algunos atributos específicos (locales) y no cuenta con relaciones concretas, puede incluirse en la superclase. Los atributos específicos pueden contener valores NULL para aquellas entidades que no sean miembros de la subclase. Un atributo *tipo* podría especificar esta circunstancia.
- De forma análoga, si todas las subclases de una especialización/generalización cuentan con atributos específicos pero no con relaciones, pueden incluirse en la superclase y sustituirse con uno o más atributos *tipo* que especifiquen la subclase o subclases a la que cada entidad pertenece.
- Deben evitarse los tipos y las categorías a menos que la situación garantice este tipo de construcción, lo cual ocurre en ciertas situaciones prácticas. En caso de ser posible, intentaremos modelar usando especialización/generalización tal y como se ha comentado al final de la Sección 4.4.
- La elección de una restricción disyunción/solapamiento y total/parcial en una especialización/generalización está condicionada por las reglas en las que se está llevando a cabo el modelado. Si los requisitos no indican ningún tipo de restricción particular, la elección predeterminada debería ser el solapamiento parcial, ya que esto no especifica ninguna restricción en los miembros de la subclase.

Como ejemplo para la aplicación de esta indicación, considere el ejemplo mostrado en la Figura 4.6, donde no se muestran atributos específicos (locales). Podemos fundir todas las subclases en la entidad EMPLEADO y añadirle los siguientes atributos:

- TipoTrabajo, cuyo conjunto de valores {'Secretaria', 'Ingeniero', 'Técnico'} indicaría la subclase de la primera especialización a la que cada empleado pertenece.
- TipoContrato, cuyo conjunto de valores {'Fijo', 'Temporal'} indicaría la subclase de la segunda especialización a la que cada empleado pertenece.
- EsJefe, cuyo conjunto de valores {'Si', 'No'} indicaría si un empleado es jefe o no.

## 4.5.3 Definiciones formales para los conceptos del modelo EER\*

Vamos a resumir ahora los conceptos del modelo EER y dar definiciones formales. Una **clase**<sup>11</sup> es un conjunto o colección de entidades; aquí se incluye cualquier construcción del esquema EER que agrupe entidades, como tipos de entidad, subclases, superclases y categorías. Una **subclase** *S* es una clase cuyas entidades deben ser siempre un subconjunto de las entidades de otra clase llamada la **superclase** *C* de la **relación superclase/**

<sup>11</sup> Aquí, el uso de la palabra *clase* difiere del uso más habitual que se le da en los lenguajes de programación orientados a objetos como C++. En él, una clase es una definición de tipo estructurada junto con sus funciones (operaciones).

**subclase** (o **IS-A**, o **ES-UN**). Indicamos una relación de este tipo como  $C/S$ . Para una relación superclase/subclase, siempre debemos tener:

Una **especialización**  $Z = \{S_1, S_2, \dots, S_n\}$  es un conjunto de subclases que tienen la misma superclase  $G$ , es decir,  $G/S_i$  es una relación superclase/subclase para  $i = 1, 2, \dots, n$ .  $G$  recibe el nombre de **tipo entidad-generalizada** (o la **superclase** de la especialización, o una **generalización** de las subclases  $\{S_1, S_2, \dots, S_n\}$ ).  $Z$  se dice que es **total** si siempre (y en cualquier momento) tenemos:

$$\bigcup_{i=1}^n S_i = G$$

En cualquier otro caso,  $Z$  se dice que es **parcial**.  $Z$  es una disyunción si siempre tenemos:

$$S_i \cap S_j = \emptyset \text{ (conjunto vacío) para } i \neq j$$

En cualquier otro caso,  $Z$  se dice que es **solapada**.

Se dice que una subclase  $S$  de  $C$  es de **predicado definido** si un predicado  $p$  de los atributos de  $C$  se utiliza para especificar las entidades de  $C$  que son miembros de  $S$ , esto es,  $S = C[p]$ , donde  $C[p]$  es el conjunto de entidades de  $C$  que satisfacen  $p$ . Una subclase que no está definida por un predicado se dice que es de **usuario definido**.

Una especialización  $Z$  (o generalización  $G$ ) se dice que es de **atributo definido** si un predicado  $(A = c_i)$ , donde  $A$  es un atributo de  $G$  y  $c_i$  es una constante del dominio de  $A$ , se utiliza para especificar los miembros de cada subclase  $S_i$  en  $Z$ . Observe que si  $c_i \neq c_j$  para  $i \neq j$ , y  $A$  es un atributo de un solo valor, entonces la especialización será una disyunción.

Una **categoría**  $T$  es una clase que es un subconjunto de la unión de  $n$  superclases  $D_1, D_2, \dots, D_n$ ,  $n > 1$ , y está formalmente especificada como:

$$T \subseteq (D_1 \cup D_2 \dots \cup D_n)$$

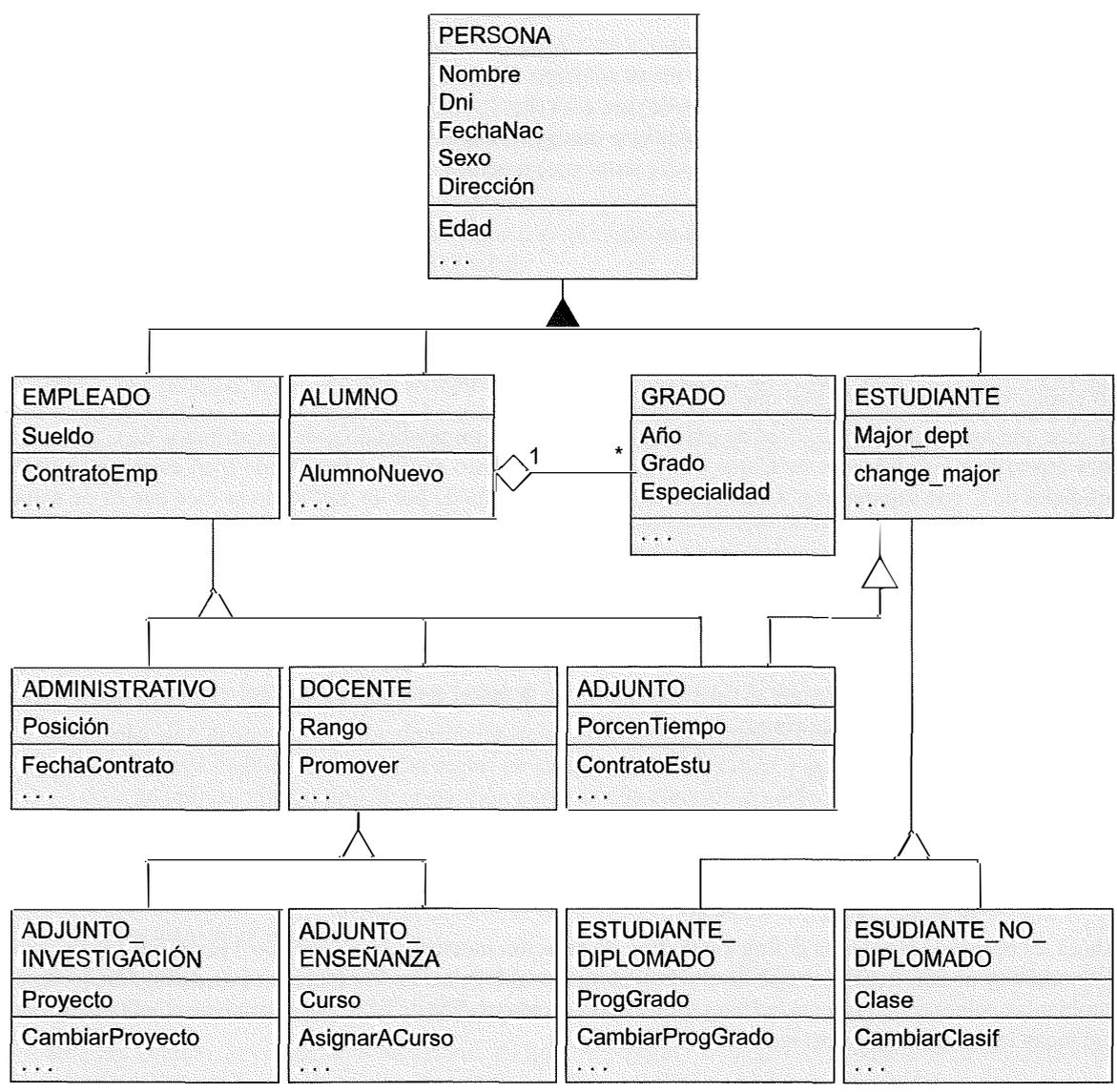
Puede usarse un predicado  $p_i$  en los atributos de  $D_i$  para especificar los miembros de cada  $D_i$  que lo son también de  $T$ . Si se especifica un predicado en cada  $D_i$  tenemos

Ahora, es necesario ampliar la definición de **tipo de relación** mostrada en el Capítulo 3 permitiendo que cualquier clase (y no sólo cualquier tipo de entidad) participe en la relación. Por tanto, en esta definición debemos cambiar las palabras *tipo de entidad* por *clase*. La indicación grafica de EER es consecuente con la de ER porque todas las clases están representadas por rectángulos.

## 4.6 Ejemplo de otra notación: representación de la especialización y la generalización en diagramas de clase UML

Vamos a tratar ahora la notación UML para la generalización/especialización y la herencia. En la Sección 3.8 ya se presentó la terminología y la notación básica del diagrama de clase UML. La Figura 4.10 muestra una posible diagramación UML que coincide con el diagrama EER mostrado en la Figura 4.7. La notación básica para la especialización/generalización (véase la Figura 4.10) es conectar las subclases por líneas verticales a otra horizontal, la cual cuenta con un triángulo que conecta la línea horizontal a la superclase a través de otra línea vertical. Un triángulo en blanco indica una especialización/generalización con la restricción de

Figura 4.10. Diagrama de clase UML correspondiente al diagrama EER de la Figura 4.7, el cual ilustra la notación UML para una especialización/generalización.



disyunción, mientras que otro relleno especifica otra de tipo *solapamiento*. La superclase raíz recibe el nombre de **clase base**, mientras que los nodos hoja se conocen como **clases hoja**. En ambos casos está permitida la herencia simple o múltiple.

El comentario y el ejemplo anteriores (junto con la Sección 3.8) ofrecen una breve panorámica de los diagramas de clase UML y su terminología. En UML existen muchos otros detalles sobre los que no hemos hablado ya que están fuera del ámbito de este libro, y son poco relevantes para la ingeniería de software. Por ejemplo, las clases pueden ser de varios tipos:

- Las clases abstractas definen atributos y operaciones, pero no cuentan con objetos que se correspondan con esas clases. Se utilizan principalmente para especificar un conjunto de atributos y operaciones que pueden ser heredados.

- Las clases concretas pueden tener objetos (entidades) instanciados que pertenezcan a la clase.
- Las clases plantilla especifican un patrón que puede usarse más adelante para definir otras clases.

En el diseño de bases de datos, nos centramos principalmente en la especificación de clases concretas cuyas colecciones de objetos están permanentemente (o persistentemente) almacenadas en la base de datos. Las notas bibliográficas del final del capítulo muestran algunas referencias a libros que describen con detalle la operativa UML. El Capítulo 12 contiene material adicional sobre UML, y el modelado de objetos en general se trata en el Capítulo 20.

## 4.7 Abstracción de datos, representación del conocimiento y conceptos de ontología

En esta sección comentaremos, en términos abstractos, algunos de los conceptos de modelado que describimos algo más concretamente en nuestra representación de los modelos ER y EER del Capítulo 3 y, anteriormente, en este mismo capítulo. Esta terminología no sólo se emplea en el modelado de datos conceptual, sino también en la literatura sobre inteligencia artificial cuando se trata de la **KR (Representación del conocimiento, Knowledge Representation)**. Esta sección trata sobre las similitudes y las diferencias existentes entre el modelado conceptual y la representación del conocimiento, e introduce algo de terminología alternativa y algunos conceptos adicionales.

El objetivo de las técnicas KR es desarrollar conceptos para el modelado acertado de ciertos **dominios de conocimiento** creando una **ontología**<sup>12</sup> que describe los conceptos del área. Esto se utiliza para almacenar y manipular conocimiento para el dibujo de inferencias, la toma de decisiones o la respuesta a preguntas. Los objetivos de la KR son similares a los de los modelos de datos semánticos, aunque existen importantes similitudes y diferencias entre ambas:

- Las dos disciplinas usan un proceso de abstracción para identificar propiedades comunes y aspectos importantes de los objetos del minimundo (conocido también como dominio de discurso en el KR), a la vez que elimina diferencias insignificantes y detalles sin importancia.
- Ambas disciplinas ofrecen conceptos, restricciones, operaciones y lenguajes para la definición de datos y la representación del conocimiento.
- La KR es, generalmente, más extensa en el ámbito que en los modelos de datos semánticos. Diferentes formas de conocimiento, como reglas (usadas en la inferencia, la deducción y la búsqueda), conocimiento predeterminado e incompleto, y conocimiento espacial y temporal, son representados en esquemas KR. Los modelos de bases de datos están empezando a expandirse para incluir algunos de estos conceptos (consulte el Capítulo 24).
- Los esquemas KR incluyen **mecanismos de razonamiento** que deducen hechos adicionales a partir de otros almacenados en la base de datos. Así pues, mientras la mayor parte de los sistemas de bases de datos están limitados a realizar consultas directas, los sistemas basados en conocimiento que utilizan esquemas KR pueden efectuar preguntas que impliquen **inferencias** sobre los datos almacenados. Las bases de datos actuales están empezando a incluir mecanismos de inferencia (consulte la Sección 24.4).
- Mientras la mayor parte de los modelos de bases de datos se concentran en la representación de sus esquemas, o meta-conocimiento, los esquemas KR suelen mezclar los esquemas con las propias instancias para proporcionar mayor flexibilidad a la hora de representar excepciones. Esto, con frecuencia, suele desembocar en deficiencias cuando estos esquemas KR son implementados, especialmente

<sup>12</sup> Una *ontología* es algo similar a un esquema conceptual, aunque con más conocimiento, reglas y excepciones.

cuando se comparan con bases de datos y cuando es preciso almacenar una gran cantidad de datos (hechos).

En esta sección, trataremos cuatro **conceptos de abstracción** utilizados en modelos de datos semánticos, como el EER y en esquemas KR: (1) clasificación e instanciación, (2) identificación, (3) especialización y generalización y (4) agregación y asociación. Los conceptos pareados de clasificación e instanciación son inversos a los de generalización y especialización. La agregación y la asociación también están relacionadas. Para clarificar el proceso de abstracción de datos y mejorar nuestra comprensión de los procesos relacionados del diseño de esquema conceptual, hablaremos de estos conceptos abstractos y de su relación con las representaciones concretas usadas en el modelo EER. Terminaremos la sección con un breve comentario acerca de la *ontología*, la cual está siendo usada ampliamente en recientes investigaciones de la representación del conocimiento.

### 4.7.1 Clasificación e instanciación

El proceso de **clasificación** supone la asignación sistemática de objetos/entidades similares a objetos de tipo clase/entidad. Ahora podemos describir (hablando de bases de datos) o razonar (en KR) las clases en lugar de los objetos individuales. Las colecciones de objetos comparten los mismos tipos de atributos, relaciones y restricciones, y al clasificar los objetos simplificamos el descubrimiento de sus propiedades. La **instanciación** es la operación inversa a la clasificación y se refiere al proceso de generación y examen de los distintos objetos de una clase. Por tanto, una instancia de un objeto está relacionada con su clase objeto por la relación **ES-UNA-INSTANCIA-DE** o **ES-UN-MIEMBRO-DE**. Aunque los diagramas EER no muestran las instancias, los UML disponen de una forma de instanciación que permite la visualización de objetos individuales. En nuestra introducción a UML no describimos esta característica.

En general, los objetos de una clase deben tener una estructura similar. Sin embargo, algunos objetos pueden mostrar propiedades que difieran en parte de las de otros objetos de la clase; estos **objetos excepción** también tienen que modelarse, y los esquemas KR permiten más excepciones que los de base de datos. Además, ciertas propiedades se aplican a toda la clase y no a objetos individuales; los esquemas KR permiten este tipo de **propiedades de clase**, al igual que los UML.

En el modelo EER, las entidades están clasificadas en tipos de entidad según sus relaciones y atributos básicos. Las entidades, a su vez, están divididas en subclases y categorías en base a las similitudes y diferencias (excepciones) existentes entre ellas. Las instancias relación están clasificadas en tipos de relación. Por tanto, los tipos de entidad y de relación, las subclases y las categorías son diferentes tipos de clases en el modelo EER. Además, no ofrece explícitamente las propiedades de clase, aunque puede desarrollarse para hacerlo. En UML, los objetos están clasificados en clases, y es posible mostrar tanto las propiedades de la clase como la de los objetos individuales.

Los modelos de representación de conocimiento permiten múltiples esquemas de clasificación en los que una clase es una *instancia* de otra clase (llamada **meta-clase**). Tenga en cuenta que esto no puede mostrarse directamente en un modelo EER, ya que sólo disponemos de dos niveles: clases e instancias. Por tanto, la única relación posible entre las clases es la de tipo superclase/subclase, mientras que en algunos esquemas KR es posible representar directamente una relación adicional clase/instancia en una jerarquía de clase. Una instancia puede ser, por sí misma, otra clase, permitiendo esquemas de clasificación de múltiples niveles.

### 4.7.2 Identificación

La **identificación** es el proceso de abstracción por el que las clases y los objetos son identificables de forma única por medio de algún **identificador**. Por ejemplo, un nombre de clase identifica inequívocamente a toda esa clase. Es necesario un mecanismo adicional para mantener separadas distintas instancias de objetos mediante identificadores de objeto. Además, es preciso identificar múltiples manifestaciones del mismo obje-

to del mundo real en la base de datos. Por ejemplo, podemos tener una tupla <'Matías Flis', '610618', '376-9821'> en una relación PERSONA y otra tupla <'301-54-0836', 'CC', 3.8> en ESTUDIANTE que pareciera que representasen a la misma persona. No existe manera de identificar que estos dos objetos de base de datos (tuplas) representan a la misma persona a menos que, durante la *fase de diseño*, preparemos los mecanismos adecuados para establecer esta relación cruzada. Por consiguiente, la identificación es necesaria a dos niveles:

- Para distinguir entre objetos y clases de la base de datos.
- Para identificar objetos de base de datos y relacionarlos con sus homólogos del mundo real.

En el modelo EER, la construcción de un esquema de identificación se basa en un sistema de nombres únicos para los constructores. Por ejemplo, cada clase (ya sea un tipo de entidad, una subclase, una categoría o un tipo de relación) debe contar con un nombre distinto. Los nombres de atributo de una clase específica también deben ser diferentes. También son necesarias reglas para identificar con claridad las referencias a los nombres de atributo en una jerarquía o entramado de especialización o generalización.

A nivel de objeto, se emplean los valores de los atributos clave para distinguir entre entidades de un tipo de entidad particular. Para los tipos débiles, las entidades se identifican por una combinación de sus propios valores clave parciales y las entidades que están relacionadas en el tipo (o tipos) de entidad propietaria. Las instancias de relación están identificadas por alguna combinación de las entidades que relacionan, en base al índice de cardinalidad especificado.

### 4.7.3 Especialización y generalización

La especialización es el proceso para clasificar una clase en subclases más especializadas. La generalización es el proceso inverso de generalizar varias clases en una clase abstracta de nivel superior que incluya los objetos de todas esas clases. La especialización es un refinamiento conceptual, mientras que la generalización es una síntesis conceptual. Las subclases se emplean en el modelo EER para representar la especialización y la generalización. La relación entre una subclase y su superclase recibe el nombre de relación **ES-UNA-SUBCLASE-DE** o, abreviando, una relación **ES-UN/ES-UNA**.

### 4.7.4 Agregación y asociación

La agregación es un concepto abstracto para la construcción de objetos complejos a partir de sus objetos componente. Existen tres situaciones en las que este concepto puede estar relacionado con el modelo EER. El primero se produce cuando añadimos atributos de un objeto para formar el objeto completo. El segundo se da cuando representamos una relación de agregación como una relación común. El tercer caso, para el cual no se proporciona explícitamente el modelo EER, supone la posibilidad de combinar objetos que están relacionados con una instancia relación particular en un *objeto agregado de nivel superior*. Esto suele ser útil a veces cuando este objeto, por sí mismo, está relacionado con otro. Llamamos a la relación existente entre los objetos primitivos y sus objetos agregados **ES-UNA-PARTE-DE**; la situación inversa recibe el nombre de **ES-UN-COMPONENTE-DE**. UML proporciona soporte para los tres tipos de agregación.

La abstracción de **asociación** se utiliza para asociar objetos procedentes de varias *clases independientes*. Por tanto, es muy parecido al segundo uso de la agregación. En el modelo EER está representado por los tipos de relación, mientras que en UML lo está por las asociaciones. Esta relación abstracta se llama **ESTÁ-ASOCIADA-CON**.

Para entender mejor los distintos usos de la agregación, considere el esquema ER de la Figura 4.11(a), el cual almacena información sobre las entrevistas realizadas por los aspirantes a obtener un empleo en distintas empresas. La clase EMPRESA es una agregación de los atributos (u objetos componente) NombreEmpresa (nombre de la empresa) y DirEmpresa (dirección de la empresa), mientras que ASPIRANTE\_TRABAJO es una agregación de Dni, Nombre, Dirección y Teléfono. Los atributos de relación NombreContacto y TlfContacto

representan el nombre y el teléfono de la persona de la empresa responsable de la entrevista. Supongamos que alguna de las entrevistas tiene como consecuencia una oferta de trabajo, mientras que otras no. Podríamos querer tratar ENTREVISTA como una clase para asociarla con OFERTA\_TRABAJO. El esquema de la Figura 4.11(b) es *incorrecto* porque supone que cada instancia de la relación entrevista tiene una oferta de trabajo. El esquema de la Figura 4.11(c) *no está permitido*, ya que el modelo ER no permite establecer relaciones entre relaciones.

Una forma de representar esta situación es crear una clase agregada de nivel superior compuesta por EMPRESA, ASPIRANTE\_TRABAJO y ENTREVISTA, y relacionarla con OFERTA\_TRABAJO (véase la Figura 4.11[d]). Aunque el modelo EER, como ya hemos mencionado, no permite esta situación, algunos modelos de datos semánticos sí que lo hacen y nombran al objeto resultante un **compuesto u objeto molecular**. Otros modelos tratan a los objetos entidad y relación uniformemente y, por ello, permiten las relaciones entre relaciones (véase la Figura 4.11[c]).

Para representar correctamente esta situación en el modelo ER, necesitamos crear un nuevo tipo de entidad débil ENTREVISTA (véase la Figura 4.11[e]), y relacionarlo con OFERTA\_TRABAJO. De este modo, siempre es posible representar correctamente estas situaciones en el modelo ER creando tipos de entidad adicionales aunque, conceptualmente, puede ser más deseable permitir la representación directa de la agregación, como puede verse en la Figura 4.11(d), o consentir las relaciones entre relaciones (véase la Figura 4.11[c]).

La principal distinción estructural entre la agregación y la asociación es que cuando una instancia de asociación se borra, los objetos que participan de ella pueden seguir existiendo. Sin embargo, si abogamos por la noción de objeto agregado (por ejemplo, un COCHE compuesto de objetos MOTOR, CHASIS y RUEDAS), el borrado del objeto COCHE implica la eliminación de todos los demás.

### 4.7.5 Ontologías y la semántica Web

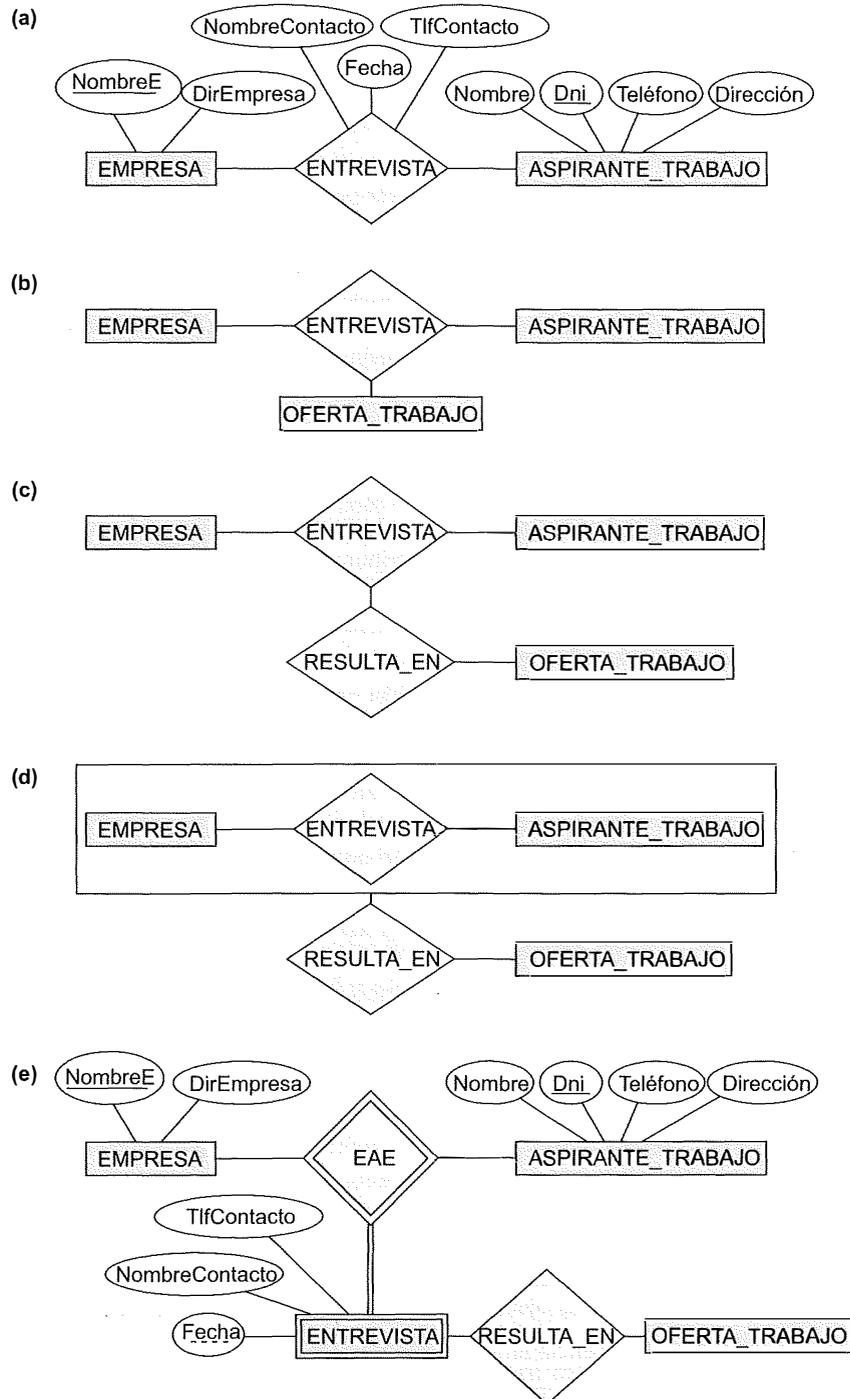
Últimamente, la cantidad de datos informatizados y de información disponible en la Web está fuera de control. Para ello, se utilizan muchos modelos y formatos diferentes. Además de los modelos de bases de datos mostrados en este libro, una gran cantidad de información se almacena en forma de **documentos**, los cuales precisan de una estructura mucho menor de la necesaria en la información de una base de datos. **Semantic Web** es un proyecto de investigación que está intentando permitir el intercambio de información entre computadores de la Web, además de intentar crear modelos de representación de conocimiento que sean lo más generales posible para permitir el intercambio y la búsqueda de información significativa entre máquinas. Se está intentando que la *ontología* sea la piedra angular sobre la que se asiente Semantic Web, y está íntimamente relacionado con la representación del conocimiento. En esta sección, se ofrecerá una breve introducción sobre qué es la ontología y cómo puede usarse para automatizar la comprensión, búsqueda e intercambio de información.

El estudio de las ontologías intenta describir las estructuras y las relaciones que son posibles en la realidad a través de vocabulario común; por consiguiente, puede considerarse como una forma de describir el conocimiento de la realidad de una cierta comunidad. La ontología tuvo su origen en la filosofía y la metafísica. Una definición de **ontología** comúnmente usada es la de la *especificación* de una *conceptualización*.<sup>13</sup>

En esta definición, una **conceptualización** es el conjunto de conceptos usados para representar la parte de realidad o conocimiento que son de interés a una comunidad de usuarios. La **especificación** se refiere al lenguaje y el vocabulario empleados para especificar la conceptualización. La ontología incluye tanto la *especificación* como la *conceptualización*. Por ejemplo, a través de dos ontologías diferentes puede especificarse la misma conceptualización. Aun basándonos en esta definición general, no existe consenso acerca de lo que es exactamente la ontología. Éstas son algunas formas de describirla:

<sup>13</sup> Esta definición la propuso Gruber (1995).

**Figura 4.11.** Agregación. (a) Tipo de relación ENTREVISTA. (b) Incluyendo OFERTA\_TRABAJO en un tipo de relación ternario (incorrecto). (c) Con la relación RESULTA\_EN participando en otras relaciones (no está permitido en ER). (d) Usando una agregación y un objeto compuesto (molecular) (normalmente, no está permitido en ER, aunque sí en algunas herramientas de modelado). (e) Representación correcta en ER.



- Un **diccionario de sinónimos** (e incluso un **diccionario** o un **glosario** de términos) describen las relaciones existentes entre las palabras (vocabulario) que representan diferentes conceptos.
- Una **taxonomía** describe el modo en que están relacionados los conceptos de un dominio particular de conocimiento usando estructuras similares a las empleadas en una especialización o una generalización.
- Un **esquema de base de datos** detallado está considerado por algunos como una ontología que describe los conceptos (entidades y atributos) y las relaciones de un minimundo real.
- Una **teoría lógica** usa los conceptos de la lógica matemática para intentar definir los conceptos y sus interrelaciones.

Habitualmente, los conceptos utilizados para describir ontologías son muy similares a los que empleamos en el modelado conceptual, como entidades, atributos, relaciones, especializaciones, etc. La diferencia principal entre una ontología y, digamos, un esquema de base de datos es que el esquema suele limitarse a describir un pequeño subconjunto de un minimundo real con el objetivo de almacenar y administrar datos. Una ontología suele considerarse algo más general, ya que intenta describir una parte de la realidad o un área de interés (por ejemplo, términos médicos, aplicaciones de comercio electrónico) de la forma más completa posible.

## 4.8 Resumen

En este capítulo hemos estudiado las extensiones del modelo ER que mejoran sus capacidades de representación. Llamamos al modelo resultante ER mejorado o modelo EER. Presentamos el concepto de una subclase y su superclase y el mecanismo relacionado de herencia atributo/relación. Mostramos cómo, a veces, es necesario crear clases de entidades adicionales, ya fuera debido a atributos específicos adicionales o debido a tipos de relación concretos. Abordamos los dos procesos principales para la definición de jerarquías y entramados superclase/subclase: la especialización y la generalización.

A continuación, mostramos la forma de representar estas nuevas construcciones en un diagrama EER. También debatimos los diferentes tipos de restricciones que pueden aplicarse a la especialización o la generalización: total/parcial y disyunción/solapamiento. Además, puede definirse un predicado para una subclase o un atributo para una especialización. Explicamos las diferencias existentes entre subclases definidas por usuario y de predicado definido y entre especializaciones del mismo tipo. Para terminar, planteamos el concepto de una categoría o tipo unión, la cual se define como un subconjunto de la unión de dos o más clases, y ofrecemos definiciones formales de todos los conceptos presentados.

Mostramos parte de la notación y la terminología UML para representar la especialización y la generalización. En la Sección 4.7 abordamos brevemente la disciplina de la representación del conocimiento y el modo que está relacionado con el modelado de datos semántico. También ofrecemos una panorámica y un resumen de los tipos de conceptos de la representación abstracta de datos: la clasificación y la instanciación, la identificación, la especialización y la generalización, y la agregación y la asociación, sin olvidarnos de la forma en que los conceptos EER y UML están relacionados con todos ellos.

### Preguntas de repaso

- 4.1. ¿Qué es una subclase? ¿Cuándo es necesaria una subclase en el modelado de datos?
- 4.2. Defina los siguientes términos: superclase de una subclase, relación superclase/subclase, relación es-una, especialización, generalización, categoría, atributos específicos (locales) y relaciones específicas.
- 4.3. Aborde el mecanismo de herencia atributo/relación. ¿Por qué es útil?
- 4.4. Comente las subclases definidas por usuario y de predicado definido, e identifique las diferencias existentes entre ellas.

- 4.5. Plantee las especializaciones definidas por usuario y de predicado definido, e identifique las diferencias existentes entre ellas.
- 4.6. Explique los dos tipos principales de restricciones en las especializaciones y las generalizaciones.
- 4.7. ¿Cuál es la diferencia entre una jerarquía y un entramado de especialización?
- 4.8. ¿Cuál es la diferencia entre una especialización y una generalización? ¿Por qué no podemos mostrar esta diferencia en los diagramas de esquema?
- 4.9. ¿En qué difiere una categoría de una subclase compartida corriente? ¿Para qué se usa una categoría? Argumente su respuesta con ejemplos.
- 4.10. Por cada uno de los siguiente términos UML (consulte las Secciones 3.8 y 4.6), indique el correspondiente en el modelo EER, en caso de existir: objeto, clase, asociación, agregación, generalización, multiplicidad, atributos, discriminador, enlace, atributo de enlace, asociación reflexiva y asociación cualificada.
- 4.11. Comente las diferencias principales existentes entre la notación en los diagramas de esquema EER y los de clase UML comparando el modo en que se representan los conceptos comunes.
- 4.12. Enumere los distintos conceptos de abstracción de datos y los conceptos de modelado correspondientes en el modelo EER.
- 4.13. ¿Qué característica de agregación no existe en el modelo EER? ¿Cómo podría mejorarse para soportarla?
- 4.14. ¿Cuáles son las principales similitudes y diferencias existentes entre las técnicas de modelado de bases de datos conceptuales y las de representación del conocimiento?
- 4.15. Comente las similitudes y diferencias existentes entre una ontología y un esquema de base de datos.

## Ejercicios

- 4.16. Diseñe un esquema EER para una aplicación de bases de datos en la que esté interesado. Especifique todas las restricciones que necesite. Asegúrese de que el esquema dispone de, al menos, cinco tipos de entidad, cuatro tipos de relación, un tipo de entidad débil, una relación superclase/subclase, una categoría y un tipo de relación n-cualquiera ( $n > 2$ ).
- 4.17. Considere el esquema ER BANCO de la Figura 3.21, y que es necesario controlar los diferentes tipos de CUENTA (AHORRO, ARQUEO, etc.) y PRÉSTAMO (COCHE, HIPOTECARIO, etc.). Suponga también que es aconsejable gestionar cada TRANSACCIÓN de una CUENTA (depósitos, retiradas, cheques, etc.) y cada PAGO del PRÉSTAMO; ambas situaciones incluyen la cantidad, la fecha y la hora. Modifique el esquema BANCO usando los conceptos ER y EER de especialización y generalización. Haga constar cualquier supuesto que haga sobre requerimientos adicionales.
- 4.18. La siguiente historia narra una versión simplificada de la organización de las instalaciones para unas Olimpiadas de verano. Dibuje un diagrama EER que muestre los tipos de entidad, los atributos, las relaciones y las especializaciones para esta aplicación. Haga constar cualquier supuesto que haga. Las instalaciones olímpicas están divididas en complejos deportivos, los cuales, a su vez, son de tipo *monodeportivo* y *multideportivo*. Los complejos multideportivos tienen áreas diseñadas para cada una de las especialidades y cuentan con un indicador (por ejemplo, centro, esquina NE, etc.). Un complejo dispone de una localización, su jefe de organización, el área ocupada, etc. Cada complejo alberga una serie de eventos (por ejemplo, en la pista de carreras se pueden disputar varios tipos de ellas). Cada evento está planificado para una fecha, tendrá una duración, un número de participantes y jueces, etc. Será preciso mantener una lista de todos los jueces junto con las pruebas en las que estarán presentes. Para cada prueba será necesario un equipamiento distinto (por ejemplo, las porterías, las pértigas, las barras paralelas) y su mantenimiento. Ambos tipos de complejos (monodeportivo y multideportivo) contarán con distintos tipos de información. Para cada uno, es preciso mantener el número de instalaciones necesarias, junto con un presupuesto aproximado.

- 4.19.** Identifique los conceptos más importantes representados en el estudio de la base de datos de una biblioteca mostrado más adelante. En particular, preste atención a las abstracciones de la clasificación (tipos de entidad y de relación), la agregación, la identificación y la especialización/generalización. Especifique las restricciones de cardinalidad (mínima, máxima) siempre que sea posible. Enumere los detalles que afectarán al diseño eventual, pero que no tengan interrelación con el conceptual. Identifique de forma separada las restricciones semánticas. Dibuje un diagrama EER de esta base de datos.

**Caso a estudiar.** La Georgia Tech Library (GTL) cuenta aproximadamente con unos 16.000 miembros, 100.000 títulos y 250.000 volúmenes (una media de 2,5 copias por libro). Alrededor del 10% se encuentra permanentemente fuera en modo de préstamo. Los bibliotecarios aseguran que los libros que se deseen pedir prestados lo estarán en el momento en que los miembros así lo deseen. Además, es preciso que conozcan cuántas copias están prestadas en cada momento. Existe un catálogo de libros *online* disponible que enumera las obras por autor, título y área. Para cada una de ellas, el catálogo mantiene un descriptor de libro que oscila desde una frase a varias páginas. Se quiere que estas referencias estén accesibles para los bibliotecarios cuando un miembro solicita información acerca de un libro. La plantilla de la biblioteca incluye un bibliotecario jefe, los bibliotecarios departamentales asociados, los de referencia, el plantel de verificación y los bibliotecarios asistentes.

Los libros pueden retenerse durante 21 días, y los miembros sólo pueden tener 5 ejemplares a la vez. Por lo general, los usuarios devuelven los libros a las tres o cuatro semanas, y la mayoría sabe que disponen de una semana de gracia antes de que se les notifique esta situación, por lo que todos intentan hacerlo antes de que expire dicho periodo. Es necesario hacer un recordatorio de la devolución a alrededor del 5% de los miembros, y la mayor parte de los libros se devuelven un mes después de vencer la fecha tope. Incluso, existe un 5% de obras que se pierden definitivamente. Los miembros más activos de la biblioteca son aquéllos que solicitan libros, al menos, diez veces al año. El 1% de los miembros más activos realiza el 15% de las peticiones, y el 10% de los miembros más activos el 40%. Cerca del 20% nunca realiza una petición.

Para pertenecer a la biblioteca, los solicitantes rellenan un formulario en el que se incluye su DNI, su dirección postal personal y la de su centro de estudios y los números de teléfono. Los bibliotecarios expiden entonces una tarjeta magnética numerada con su foto y válida para cuatro años. Un mes antes de que expire, se le envía un mensaje de renovación. Los profesores de los centros de enseñanza son considerados miembros de forma automática. Cuando un nuevo profesor entra en un colegio, se obtiene la información necesaria de su registro de empleado y se envía por correo una tarjeta a su dirección profesional. Los profesores pueden revisar libros en intervalos de tres meses, y su periodo de gracia es de dos semanas. Las renovaciones se remiten a su dirección profesional.

La biblioteca no presta ciertos libros, como obras de referencia, volúmenes raros y mapas, por lo que los bibliotecarios deben ser capaces de distinguir qué obras pueden prestar y cuáles no. Además, cuentan con una lista de algunos libros que resultaría interesante adquirir pero que no pueden, como obras raras o descatalogadas y otras que se perdieron o se destruyeron y que no han sido reemplazadas. Algunos libros pueden tener el mismo título; por consiguiente, este dato no puede usarse como campo clave. Cada uno de ellos está identificado por su ISBN (*International Standard Book Number*), un código internacional único asignado a todos los libros. Dos obras con el mismo título pueden tener ISBN diferentes si están escritos en idiomas diferentes o tienen distintas encuadernaciones (tapa dura, tapa blanda). Las ediciones de la misma obra tienen distintos ISBN.

La base de datos propuesta debe diseñarse de forma que controle los miembros, los libros, el catálogo y los préstamos.

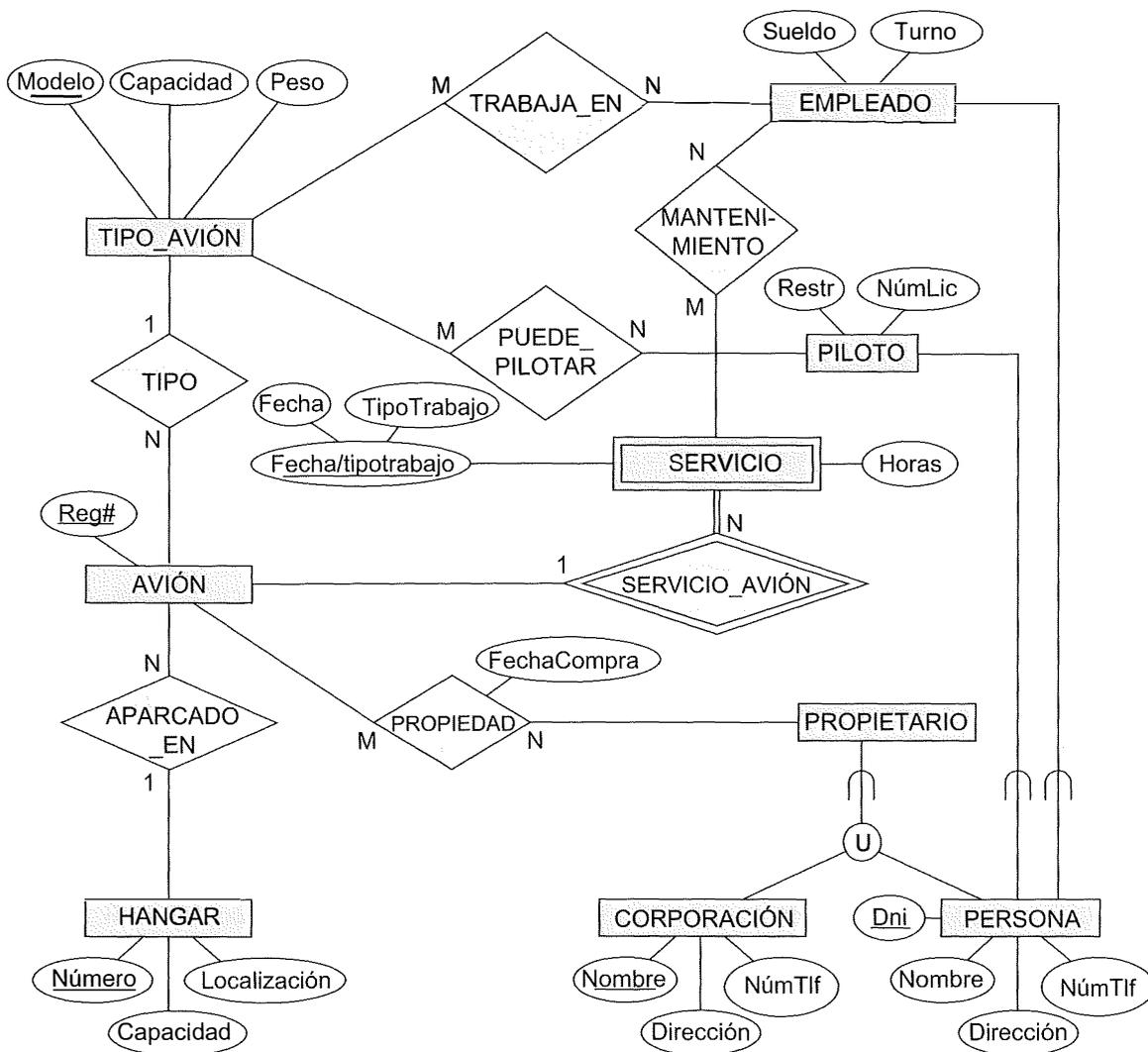
- 4.20.** Diseñe una base de datos para gestionar la información de un museo de arte. Asumimos que los siguientes datos fueron recopilados:

- El museo dispone de una colección de `OBJETOS_DE_ARTE`. Cada uno de ellos cuenta con un identificador único (`Id`), un `Artista` (en caso de conocerse), el `Año` de creación (si se conoce), un `Título` y una `Descripción`. Los objetos están clasificados de varias formas, tal y como se comentará más adelante.
- `OBJETOS_DE_ARTE` está categorizada en función a sus tipos, de los cuales hay tres principales: `PINTURA`, `ESCULTURA` y `MONUMENTO` más un cuarto llamado `OTRO` para acomodar a aquéllos que no se ajustan a ninguno de los otros tres.
- Una `PINTURA` tiene un `TipoPintura` (aceite, al agua, etc.), el material sobre el que está Dibujado (papel, lienzo, madera, etc.) y un `EstiloPintura` (moderno, abstracto, etc.).
- Una `ESCULTURA` o un `MONUMENTO` tiene el `Material` sobre el que fue creado (madera, piedra, etc.), una `Altura`, una `Anchura` y un `EstiloEscultura`.
- Un objeto de arte encuadrado en la categoría `OTRO` tiene un `TipoObra` (impresión, fotografía, etc.) y un `EstiloOtro`.
- Los `OBJETOS_DE_ARTE` están clasificados como una `COLECCIÓN_PERMANENTE` (aquéllos que son propiedad del museo) y como `PRESTADO`. Los datos con los que contamos del primer tipo son la `FechaAdquisición`, su `Estado` (en exhibición, en préstamo o almacenada) y su `Coste`. La información tomada sobre los objetos de tipo `PRESTADO` incluye la `Colección` propietaria de la misma, su `FechaPréstamo` y la `FechaDevolución`.
- La información acerca del país o la cultura de `Origen` (Italia, Egipto, América, India, etc.) y su `Época` (Renacimiento, Moderna, Antigua, etc.) se almacena en cada `OBJETO_DE_ARTE`.
- El museo conserva información sobre el `ARTISTA`, en caso de conocerse: `Nombre`, `FechaNac` (si se sabe), `FechaFallecimiento` (si corresponde), `PaísOrigen`, `Época`, `EstiloPrincipal` y `Descripción`. Se asume que el `Nombre` es un dato único.
- Se celebran distintas `EXHIBICIONES`, cada una con su `Nombre`, su `FechaInicio` y su `FechaFinalización`. Las `EXHIBICIONES` están relacionadas con los objetos de arte que están en estado de exhibición durante la misma.
- También se mantiene información sobre otras `COLECCIONES` con las que el museo interactúa, incluyendo su `Nombre` (único), el `Tipo` (museo, personal, etc.), `Descripción`, `Dirección`, `Teléfono` y `PersonaContacto` actual.

Diseñe un diagrama EER para esta aplicación. Haga constar cualquier supuesto que haga, y justifíquelo.

- 4.21. La Figura 4.12 muestra un diagrama EER para la base de datos de un pequeño aeropuerto que se utiliza para mantener la información de las aeronaves, sus propietarios, los empleados del aeropuerto y los pilotos. Éstos son los datos recopilados. Cada `AVIÓN` dispone de un número de registro [`Reg#`], es de un tipo particular [`TIPO`] y está aparcado en un hangar concreto [`APARCADO_EN`]. Cada `TIPO_AVIÓN` tiene un número de [`Modelo`], una [`Capacidad`] y un [`Peso`]. Cada `HANGAR` cuenta con un [`Número`], una [`Capacidad`] y una [`Localización`]. La base de datos también controla los propietarios de cada aeronave [`PROPIETARIO`] y los empleados encargados del [`MANTENIMIENTO`]. Cada `PROPIETARIO` está relacionado con un `AVIÓN` e incluye la fecha de adquisición [`FechaCompra`]. Cada relación en `MANTENIMIENTO` asocia a un empleado con un [`SERVICIO`]. Cada avión se somete a revisión cada cierto tiempo; por tanto, está relacionado con [`SERVICIO_AVIÓN`] por un número de registro `SERVICIO`, cada uno de los cuales incluye como atributos la fecha de mantenimiento [`Fecha`], el número de horas empleadas en el trabajo [`Horas`] y el tipo de servicio efectuado [`TipoTrabajo`]. Utilizamos una entidad débil [`SERVICIO`] para representar el servicio del avión, ya que su número de registro se utiliza para identificar un registro de servicio. Un `PROPIETARIO` puede ser tanto una persona como una corporación. Por tanto, utilizamos

Figura 4.12. Esquema EER para la base de datos PEQUEÑO\_AEROPUERTO.



un tipo de unión (categoría) [PROPIETARIO] que es un subconjunto de los tipos de entidad [CORPORACIÓN] y [PERSONA]. Tanto los pilotos [PILOTO] como los empleados [EMPLEADO] son subclasses de PERSONA. Cada PILOTO tiene, como atributos específicos, su número de licencia de vuelo [NúmeroLicencia] y sus restricciones [Restricción], mientras que los de cada EMPLEADO son el [Salario] y el [Turno]. Toda entidad PERSONA de la base de datos tiene su número del Documento nacional de identidad [Dni], un [Nombre], una [Dirección] y un [NúmeroTeléfono]. Para cada CORPORACIÓN existe un [Nombre], una [Dirección] y un [NúmeroTeléfono]. La base de datos también mantiene los tipos de aviones que cada piloto está autorizado a pilotar [PUEDE\_PILOTAR] y aquéllos en los que cada empleado puede realizar tareas de mantenimiento [TRABAJA\_EN]. Observe cómo el esquema EER de PEQUEÑO\_AEROPUERTO de la Figura 4.12 puede representarse en notación UML. (Nota. No hemos comentado la forma de representar las categorías [tipos de unión] en UML, por lo que no tiene que indicarlo ni en ésta ni en la siguiente pregunta.)

**4.22.** Desarrolle el modo de representar en notación UML el esquema EER UNIVERSIDAD de la Figura 4.9.

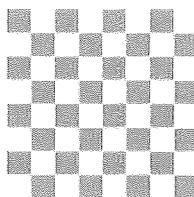
**4.23.** Considere los conjuntos de entidades y atributos mostrados en la siguiente tabla. Coloque una marca en una de las columnas de cada fila para indicar la relación entre las columnas derecha y la situada más a la izquierda.

- El lado izquierdo tiene una relación con el derecho.
- El lado derecho es un atributo del lado izquierdo.
- El lado izquierdo es una especialización del derecho.
- El lado izquierdo es una generalización del derecho.

Entidad	(a) Tiene una relación con	(b) Tiene un atributo que es	(c) Es una especialización de	(d) Es una generalización de	Entidad o atributo
1. MADRE					PERSONA
2. HIJA					MADRE
3. ESTUDIANTE					PERSONA
4. ESTUDIANTE					IdEstudiante
5. COLEGIO					ESTUDIANTE
6. COLEGIO					AULA
7. ANIMAL					CABALLO
8. CABALLO					Raza
9. CABALLO					Edad
10. EMPLEADO					DNI
11. MUEBLE					SILLA
12. SILLA					Peso
13. HUMANO					MUJER
14. SOLDADO					PERSONA
15. SOLDADO_ENEMIGO					PERSONA

**4.24.** Dibuje un diagrama UML para almacenar en una base de datos una partida de ajedrez. Puede consultar la dirección <http://www.chessgames.com> para obtener información acerca de una aplicación similar a la que tiene que desarrollar. Documente claramente cualquier decisión que tome en su diagrama UML. Las siguientes pueden ser algunas de esas decisiones:

- La partida se realiza entre dos jugadores.
- Se juega en un tablero de 8×8 similar al mostrado a continuación:

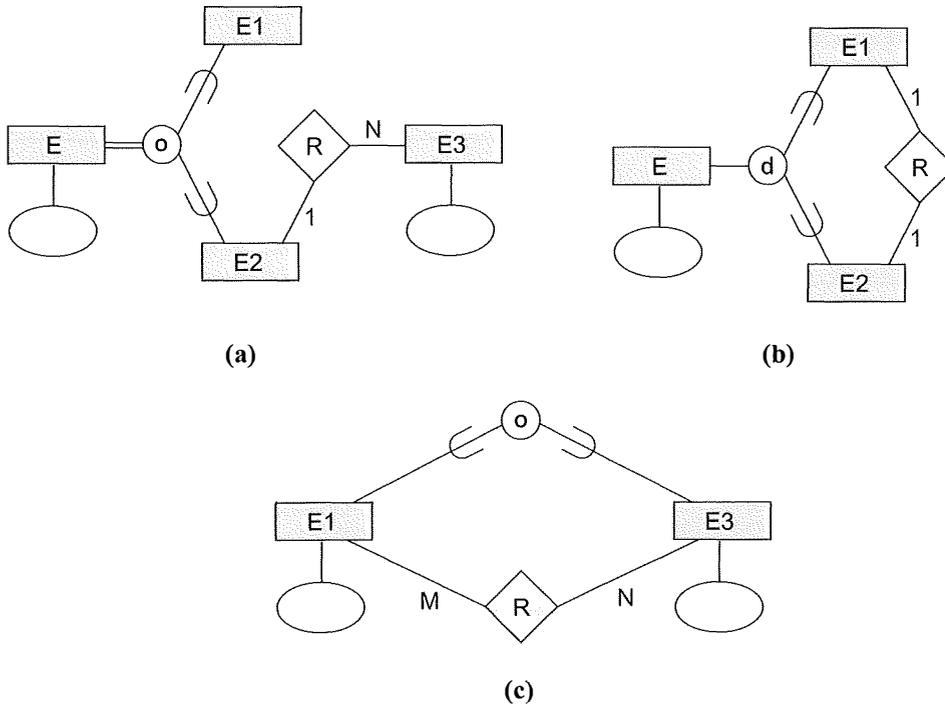


3. Los jugadores asumen un color (negro o blanco) al inicio de la partida.
4. Cada jugador empieza con las siguientes piezas:
 

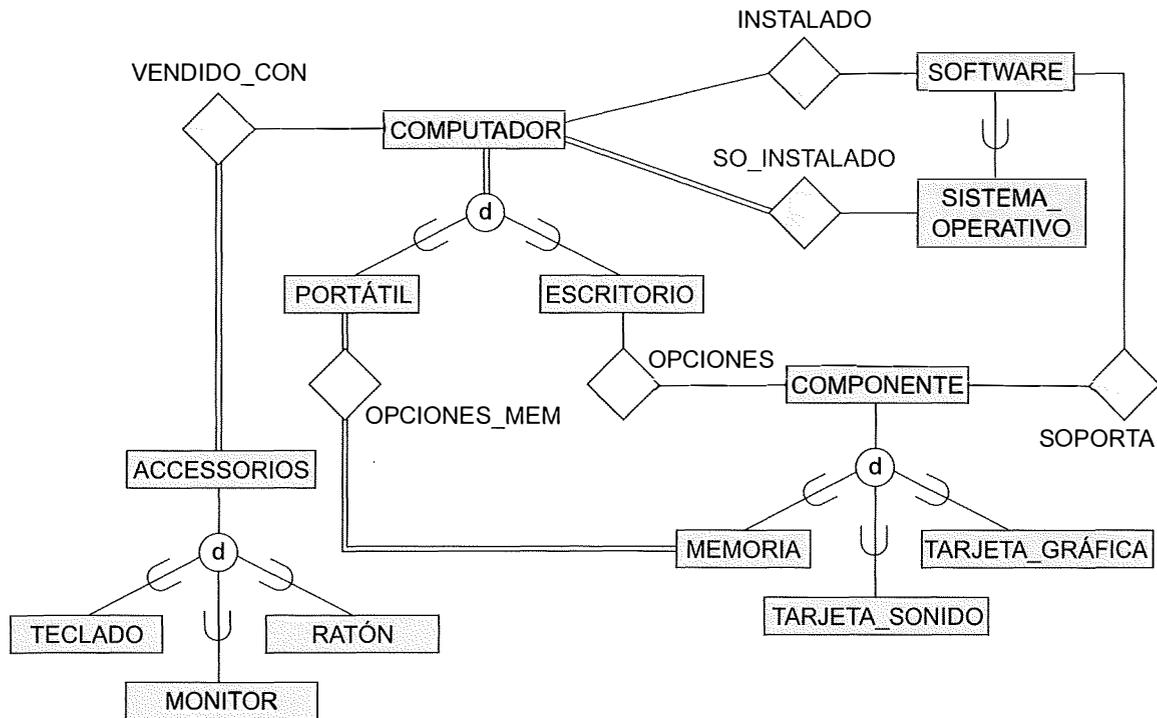
a. 1 rey	d. 2 alfiles
b. 1 reina	e. 2 caballos
c. 2 torres	f. 8 peones
5. Cada pieza se encuentra en su posición inicial.
6. Cada pieza cuenta con su propio conjunto de movimientos permitidos en función al estado de la partida. No es necesario preocuparse de qué movimientos son legales y cuáles no, excepto en los siguientes casos:
  - a. Una pieza puede moverse a un cuadro vacío o capturar una pieza del contrario.
  - b. Si una pieza es capturada, se elimina del tablero.
  - c. Si un peón alcanza la última fila es “promocionado”, convirtiéndose en otra pieza (reina, torre, alfil o rey).

*Nota:* Algunas de estas funciones pueden aplicarse a múltiples clases.

- 4.25. Dibuje un diagrama EER para la partida de ajedrez descrita en el Ejercicio 4.24. Concéntrese en los aspectos de almacenamiento persistente del sistema. Por ejemplo, puede que sea necesario recuperar todos los movimientos de cada partida en orden secuencial.
- 4.26. ¿Cuáles de los siguientes diagramas EER son incorrectos y por qué? Documente claramente cualquier decisión que tome.



- 4.27. Considere un diagrama EER en el que se describe los sistemas informáticos de una empresa. Proporcione sus propios atributos y claves para cada tipo de entidad. Facilite las restricciones de cardinalidad máximas justificando su elección. Escriba una descripción completa de lo que representa este diagrama EER.



## Ejercicios de práctica

- 4.28. Considere el diagrama EER de la base de datos UNIVERSIDAD mostrada en la Figura 4.9. Realice este diseño usando alguna herramienta de modelado de datos como ERWin o Rational Rose. Elabore una lista de las diferencias de notación existente entre el diagrama en modo texto y el equivalente construido con la herramienta.
- 4.29. Considere el diagrama EER de la base de datos del PEQUEÑO\_AEROPUERTO mostrada en la Figura 4.12. Realice este diseño usando alguna herramienta de modelado de datos como ERWin o Rational Rose. Preste especial atención a la forma de modelar la categoría PROPIETARIO en este diagrama. (*Consejo:* Considere usar PROPIEDAD\_DE\_EMPRESA y una PROPIEDAD\_DE\_PERSONA como dos tipos de relación diferentes).
- 4.30. Considere la base de datos UNIVERSIDAD descrita en el Ejercicio 3.16. En la Práctica 3.31 realizó un esquema ER para la misma mediante herramientas de modelado como ERWin o Rational Rose. Modifique este diagrama clasificando un CURSO como CURSO\_BÁSICO o CURSO\_SUPERIOR y un INSTRUCTOR como PROFESOR\_AGREGADO o PROFESOR\_TITULAR. Incluya los atributos apropiados para estos nuevos tipos de entidades. A continuación, establezca las relaciones necesarias para que el profesor agregado sea el encargado de impartir los cursos básicos, mientras que los titulares se encarguen de los superiores.

## Bibliografía seleccionada

Son muchos los artículos que han propuesto modelos de datos semánticos o conceptuales. A continuación le ofrecemos una lista representativa de los mismos. Un grupo de ellos, entre los que se incluyen Abrial (1974), el modelo DIAM de Senko (1975), el método NIAM (Verheijen y VanBekum 1982), y Bracchi y otros (1976), presentan los modelos semánticos que están basados en el concepto de relaciones binarias. Un segundo grupo más moderno aborda los métodos para extender el modelo relacional y mejorar sus posibilidades.

Entre ellos se incluyen los de Schmid y Swenson (1975), Navathe y Schkolnick (1978), el modelo RM/T de Codd (1979), Furtado (1978) y el modelo estructural de Wiederhold y Elmasri (1979).

El modelo ER fue propuesto originalmente por Chen (1976) y formalizado en Ng (1981). Desde entonces se han propuesto numerosas extensiones, como la de Scheuermann y otros (1979), Dos Santos y otros (1979), Teorey y otros (1986), Gogolla y Hohenstein (1991) y el modelo ECR (Entidad-categoría-relación, *Entity-Category-Relationship*) de Elmasri y otros (1985). Smith y Smith (1977) presentan los conceptos de generalización y agregación. El modelo de datos semántico de Hammer y McLeod (1981) presentó los conceptos de entramados clase/subclase, así como conceptos de modelado avanzados.

En Hull y King (1987) aparece un sondeo sobre el modelado de datos semántico. Eick (1991) plantea el diseño y las transformaciones de los esquemas conceptuales. Los análisis de las restricciones para  $n$  relaciones aparece en Soutou (1998). El UML se muestra con detalle en Booch, Rumbaugh y Jacobson (1999). Fowler y Scott (2000), junto con Stevens y Pooley (2000), ofrecen una introducción concisa a los conceptos UML.

Fensel (2000, 2003) habla sobre la Semantic Web y la aplicación de ontologías. Uschold y Gruninger (1996), junto con Gruber (1995), abordan las ontologías. El número de junio de 2002 de *Communications of the ACM* está dedicado a los conceptos y las aplicaciones de la ontología. Fensel (2003) es un libro que trata las ontologías y el comercio electrónico.