

Elmasri • Navathe

# Sistemas de banco de dados

6ª edição





Elmasri • Navathe

# Sistemas de banco de dados

6ª edição

**Ramez Elmasri**

*Departamento de Ciência da Computação e Engenharia  
Universidade do Texas em Arlington*

**Shamkant B. Navathe**

*Faculdade de Computação  
Georgia Institute of Technology*

**Tradução:**

*Daniel Vieira*

**Revisão técnica:**

*Enzo Seraphim*

*Thatyana de Faria Piola Seraphim*

Professores Doutores do Instituto de Engenharia de Sistemas e  
Tecnologias da Informação — Universidade Federal de Itajubá

**PEARSON**

**abdr**   
ASSOCIAÇÃO  
BRASILEIRA  
DE DIREITOS  
REPROGRÁFICOS  
*Respeite o direito autoral*

© 2011 by Pearson Education do Brasil.  
© 2011, 2007, 2004, 2000, 1994 e 1989 Pearson Education, Inc.

Tradução autorizada a partir da edição original, em inglês, *Fundamentals of Database Systems*, 6<sup>th</sup> edition, publicada pela Pearson Education, Inc., sob o selo Addison-Wesley.

Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Pearson Education do Brasil.

*Diretor editorial:* Roger Trimer  
*Gerente editorial:* Sabrina Cairo  
*Supervisor de produção editorial:* Marcelo França  
*Editora plena:* Thelma Babaoka  
*Editora de texto:* Sabrina Levensteinas  
*Preparação:* Paula Brandão Perez Mendes  
*Revisão:* Elisa Andrade Buzzo  
*Capa:* Thyago Santos sobre o projeto original de Lou Gibbs/Getty Images  
*Projeto gráfico e diagramação:* Globaltec Artes Gráficas Ltda.

**Dados Internacionais de Catalogação na Publicação (CIP)**  
**(Câmara Brasileira do Livro, SP, Brasil)**

---

Elmasri, Ramez

Sistemas de banco de dados / Ramez Elmasri e Shamkant B. Navathe ; tradução Daniel Vieira ; revisão técnica Enzo Seraphim e Thatyana de Faria Piola Seraphim. -- 6. ed. -- São Paulo : Pearson Addison Wesley, 2011.

Título original: Fundamentals of database systems.

ISBN 978-85-4301-381-7

1. Banco de dados I. Navathe, Shamkant B..II. Título.

10-11462

CDD-005.75

---

**Índices para catálogo sistemático:**

1. Banco de dados : Sistemas : Processamento de dados 005.75
2. Banco de dados : Fundamentos : Processamento de dados 005.75

3ª reimpressão – Julho 2014  
Direitos exclusivos para a língua portuguesa cedidos à  
Pearson Education do Brasil Ltda.,  
uma empresa do grupo Pearson Education  
Rua Nelson Francisco, 26  
CEP 02712-100 – São Paulo – SP – Brasil  
Fone: (11) 2178-8686 – Fax: (11) 2178-8688  
vendas@pearson.com

# O modelo Entidade-Relacionamento Estendido (EER)

Os conceitos de modelagem ER discutidos no Capítulo 7 são suficientes para representar muitos esquemas de banco de dados para aplicações *tradicionais*, que incluem diversas aplicações de processamento de dados no comércio e na indústria. Desde o final da década de 1970, porém, os projetistas de aplicações de banco de dados têm tentado projetar esquemas de banco de dados mais precisos, que refletem as propriedades de dados e restrições com mais precisão. Isso foi particularmente importante para aplicações mais novas da tecnologia de banco de dados, como aqueles para projeto de engenharia e manufatura (CAD/CAM),<sup>1</sup> telecomunicações, sistemas de software complexos e sistemas de informações geográficas (GIS — **Geographic Information Systems**), entre muitas outras aplicações. Esses tipos de bancos de dados possuem requisitos mais complexos do que as aplicações mais tradicionais. Isso levou ao desenvolvimento de conceitos adicionais de *modelagem semântica de dados*, que foram incorporados em modelos de dados conceituais, como o modelo ER. Vários modelos de dados semânticos têm sido propostos na literatura. Muitos desses conceitos também foram desenvolvidos independentemente nas áreas relacionadas de ciência da computação, como a área de **representação do conhecimento** da inteligência artificial e a área de **modelagem de objeto** na engenharia de software.

Neste capítulo, descrevemos recursos que foram propostos para modelos de dados semânticos, e mostramos como o modelo ER pode ser melhorado para

incluir esses conceitos, levando ao modelo **ER Estendido (EER)**.<sup>2</sup> Começamos na Seção 8.1 incorporando os conceitos de *relacionamentos de classe/subclasse* e *herança de tipo* ao modelo ER. Depois, na Seção 8.2, acrescentamos os conceitos de *especialização* e *generalização*. A Seção 8.3 discute os diversos tipos de *restrições* sobre especialização/generalização, e a Seção 8.4 mostra como a construção UNION pode ser modelada ao incluir o conceito de *categoria* no modelo EER. A Seção 8.5 oferece um esquema de banco de dados de exemplo UNIVERSIDADE no modelo EER e resume os conceitos do modelo EER oferecendo definições formais. Usaremos os termos *objeto* e *entidade* com o mesmo significado neste capítulo, pois muitos desses conceitos são comumente usados nos modelos orientados a objeto.

Apresentamos a notação do diagrama de classes UML para representar a especialização e a generalização na Seção 8.6, e as comparamos resumidamente com a notação e os conceitos de EER. Isso serve como um exemplo de notação alternativa, e é uma continuação da Seção 7.8, a qual apresentou a notação básica do diagrama de classes UML que corresponde ao modelo ER básico. Na Seção 8.7, discutimos as abstrações fundamentais que são usadas como base de muitos modelos de dados semânticos. No final do capítulo há um resumo.

Para uma introdução detalhada à modelagem conceitual, o Capítulo 8 deve ser considerado uma continuação do Capítulo 7. Contudo, se apenas uma introdução básica à modelagem ER for desejada, este

<sup>1</sup> CAD/CAM significa Computer-Aided Design/Computer-Aided Manufacturing (projeto auxiliado por computador/fabricação auxiliada por computador).

<sup>2</sup> EER também tem sido usado para indicar o modelo ER Aprimorado.

capítulo poderá ser omitido. Como alternativa, o leitor pode decidir pular algumas ou todas as seções posteriores deste capítulo (seções 8.4 a 8.8).

## 8.1 Subclasses, superclasses e herança

O modelo EER inclui *todos os conceitos de modelagem do modelo ER* que foram apresentados no Capítulo 7. Além disso, inclui os conceitos de **subclasse** e **superclasse** e os conceitos relacionados de **especialização** e **generalização** (ver seções 8.2 e 8.3). Outro conceito incluído no modelo EER é o de uma **categoria** ou **tipo de união** (ver Seção 8.4) usado para representar uma coleção de objetos (entidades), que é a *união* de objetos de diferentes tipos de entidade. Associado a esses conceitos está o importante mecanismo de **herança de atributo e relacionamento**. Infelizmente, não existe uma terminologia-padrão para esses conceitos, de modo que usamos a terminologia mais comum. A terminologia alternativa é dada nas notas de rodapé. Também descrevemos uma técnica diagramática para exibir esses conceitos quando eles surgem em um esquema EER. Chamamos os diagramas de esquema resultantes de **diagramas ER estendidos**, ou **diagramas EER**.

O primeiro conceito do modelo ER Estendidos (EER) ao qual nos dedicamos é o de um **subtipo** ou **subclasse** de um tipo de entidade. Conforme discu-

timos no Capítulo 7, um tipo de entidade é usado para representar um *tipo de entidade* e o *conjunto de entidades* ou *coleção de entidades desse tipo* que existem no banco de dados. Por exemplo, o tipo de entidade FUNCIONARIO descreve o tipo (ou seja, os atributos e relacionamentos) de cada entidade de funcionário, e também se refere ao conjunto atual de entidades FUNCIONARIO no banco de dados EMPRESA. Em muitos casos, um tipo de entidade tem diversos subagrupamentos ou subtipos de suas entidades que são significativos e precisam ser representados explicitamente, por causa de seu significado para a aplicação de banco de dados. Por exemplo, as entidades que são membros do tipo de entidade FUNCIONARIO podem ser distinguidas ainda mais em SECRETARIA, ENGENHEIRO, GERENTE, TECNICO, FUNCIONARIO\_MENSAL, FUNCIONARIO\_HORISTA, e assim por diante. O conjunto de entidades em cada um desses agrupamentos é um subconjunto das entidades que pertencem ao conjunto de entidades FUNCIONARIO, significando que cada entidade que é membro de um desses subagrupamentos também é um funcionário. Chamamos cada um desses subagrupamentos de **subclasse** ou **subtipo** do tipo de entidade FUNCIONARIO, e o tipo de entidade FUNCIONARIO é chamado de **superclasse** ou **supertipo** para cada uma dessas subclasses. A Figura 8.1 mostra como representar esses conceitos nos diagramas EER. (A notação de círculo na Figura 8.1 será explicada na Seção 8.2.)

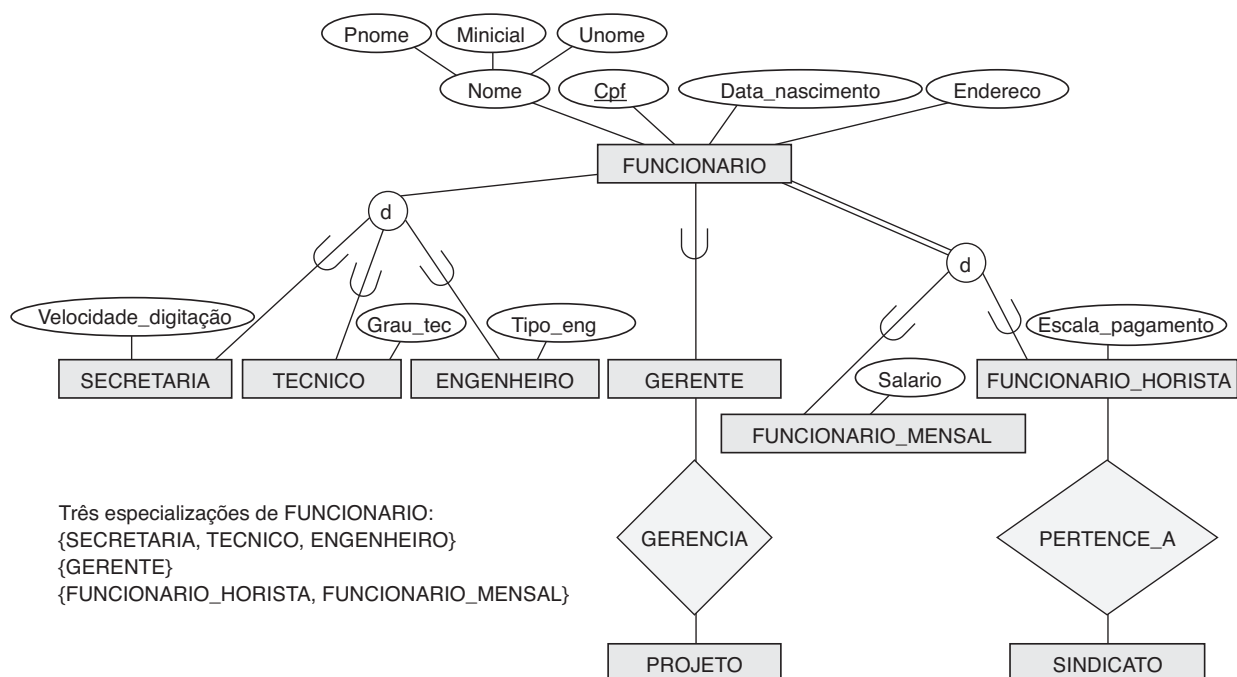


Figura 8.1

Notação do diagrama EER para representar subclasses e especialização.

Chamamos o relacionamento entre uma superclasse e qualquer uma de suas subclasses de **relacionamento superclasse/subclasse**, ou **supertipo/subtipo**, ou simplesmente **classe/subclasse**.<sup>3</sup> Em nosso exemplo anterior, FUNCIONARIO/SECRETARIA e FUNCIONARIO/TECNICO são dois relacionamentos de classe/subclasse. Observe que uma entidade-membro da subclasse representa a *mesma entidade do mundo real* de algum membro da superclasse. Por exemplo, uma entidade SECRETARIA ‘Joana Logano’ também é a entidade FUNCIONARIO ‘Joana Logano’. Logo, o membro da subclasse é o mesmo que a entidade na superclasse, mas em um *papel específico* distinto. Porém, quando implementamos um relacionamento de superclasse/subclasse no sistema de banco de dados, podemos representar um membro da subclasse como um objeto de banco de dados distinto — digamos, um registro distinto que é relacionado por meio do atributo-chave a sua entidade de superclasse. Na Seção 9.2, discutiremos diversas opções para representar os relacionamentos de superclasse/subclasse nos bancos de dados relacionais.

Uma entidade não pode existir no banco de dados simplesmente por ser um membro de uma subclasse; ela também precisa ser um membro da superclasse. Essa entidade pode ser incluída opcionalmente como um membro de qualquer número de subclasses. Por exemplo, um funcionário assalariado que também é um engenheiro pertence às duas subclasses ENGENHEIRO e FUNCIONARIO\_MENSAL do tipo de entidade FUNCIONARIO. Contudo, não é necessário que toda entidade em uma superclasse seja um membro de alguma subclasse.

Um conceito importante associado às subclasses (subtipos) é o de **herança de tipo**. Lembre-se de que o *tipo* de uma entidade é definido pelos atributos que ela possui e os tipos de relacionamento de que participa. Como uma entidade na subclasse representa a mesma entidade do mundo real da superclasse, ela deve possuir valores para seus atributos específicos *bem como* valores de seus atributos como um membro da superclasse. Dizemos que uma entidade que é um membro de uma subclasse **herda** todos os atributos da entidade como um membro da superclasse. A entidade também herda todos os relacionamentos de que a superclasse participa. Observe que uma subclasse, com os próprios atributos específicos (ou locais) e relacionamentos junto com todos os atributos e relacionamentos que herda da superclasse, pode ser considerada um *tipo de entidade* por si própria.<sup>4</sup>

## 8.2 Especialização e generalização

### 8.2.1 Especialização

**Especialização** é o processo de definir um *conjunto de subclasses* de um tipo de entidade. Esse tipo de entidade é chamado de **superclasse** da especialização. O conjunto de subclasses que forma uma especialização é definido com base em alguma característica distinta das entidades na superclasse. Por exemplo, o conjunto de subclasses {SECRETARIA, ENGENHEIRO, TECNICO} é uma especialização da superclasse FUNCIONARIO, que distingue as entidades do funcionário com base no *tipo de cargo* de cada uma. Podemos ter várias especializações do mesmo tipo de entidade com base em características distintas. Por exemplo, outra especialização do tipo de entidade FUNCIONARIO pode gerar o conjunto de subclasses {FUNCIONARIO\_MENSAL, FUNCIONARIO\_HORISTA}. Tal especialização distingue os funcionários baseando-se no *método de pagamento*.

A Figura 8.1 mostra como representamos uma especialização na forma de um diagrama EER. As subclasses que definem uma especialização são conectadas por linhas a um círculo que representa a especialização, o qual está conectado, por sua vez, à superclasse. O *símbolo de subconjunto* em cada linha que conecta uma subclasse ao círculo indica a direção do relacionamento superclasse/subclasse.<sup>5</sup> Os atributos que se aplicam apenas a entidades de uma subclasse em particular — como *Velocidade\_digitação* de SECRETARIA — são conectados ao retângulo que representa essa subclasse. Estes são chamados de **atributos específicos** (ou **atributos locais**) da subclasse. De modo semelhante, uma subclasse pode participar de **tipos de relacionamento específicos**, como a subclasse FUNCIONARIO\_HORISTA que participa do relacionamento PERTENCE\_A da Figura 8.1. Explicaremos o símbolo **d** nos círculos da Figura 8.1 e a notação adicional do diagrama EER em breve.

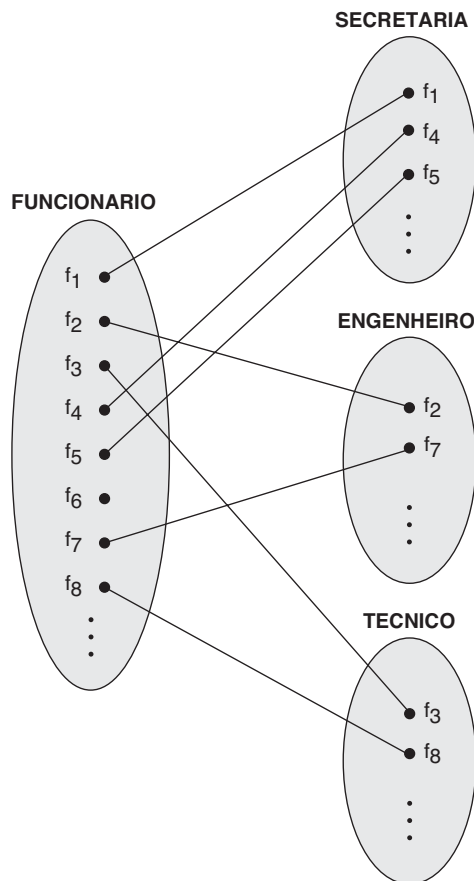
A Figura 8.2 mostra algumas instâncias de entidade que pertencem às subclasses da especialização {SECRETARIA, ENGENHEIRO, TECNICO}. Novamente, observe que uma entidade que pertence a uma subclasse representa a *mesma entidade do mundo real* que a entidade conectada a ela na superclasse FUNCIONARIO, embora a mesma entidade apareça duas vezes; por exemplo,  $f_1$  aparece em FUNCIONARIO e SECRETARIA na Figura 8.2. Como a figura sugere,

<sup>3</sup> Um relacionamento de classe/subclasse com frequência é chamado de **relacionamento É UM (IS-A)** — ou seja, é um(a), devido ao modo como nos referimos ao conceito. Dizemos que uma SECRETARIA *é uma* FUNCIONARIA, um TECNICO *é um* FUNCIONARIO, e assim por diante.

<sup>4</sup> Em algumas linguagens de programação orientadas a objeto, uma restrição comum é que uma entidade (ou objeto) tem *apenas um tipo*. Isso geralmente é muito restritivo para a modelagem conceitual do banco de dados.

<sup>5</sup> Existem muitas notações alternativas para a especialização; apresentamos a notação UML na Seção 8.6 e outras notações propostas no Apêndice A.





**Figura 8.2**

Instâncias de uma especialização.

um relacionamento de superclasse/subclasse, como FUNCIONARIO/SECRETARIA, assemelha-se a um relacionamento 1:1 *no nível de instância* (ver Figura 7.12). A principal diferença é que, em um relacionamento 1:1, duas *entidades distintas* estão relacionadas, enquanto em um relacionamento superclasse/subclasse a entidade na subclasse é a mesma entidade do mundo real que a entidade na superclasse, mas está desempenhando um *papel especializado* — por exemplo, um FUNCIONARIO especializado no papel de SECRETARIA, ou um FUNCIONARIO especializado no papel de TECNICO.

Existem dois motivos principais para incluir relacionamentos de classe/subclasse e especializações em um modelo de dados. O primeiro é que certos atributos podem se aplicar a algumas, mas não a todas as entidades da superclasse. Uma subclasse é definida a fim de agrupar as entidades às quais esses atributos se aplicam. Os membros da subclasse ainda podem compartilhar a maioria de seus atributos com os outros membros da superclasse. Por exemplo, na Figura 8.1, a subclasse SECRETARIA tem o atributo específico *Velocidade\_digitação*, enquanto

a subclasse ENGENHEIRO tem o atributo específico *Tipo\_eng*, mas SECRETARIA e ENGENHEIRO compartilham seus outros atributos herdados do tipo de entidade FUNCIONARIO.

O segundo motivo para usar subclasses é que alguns tipos de relacionamento podem participar apenas de entidades que são membros da subclasse. Por exemplo, se apenas FUNCIONARIOS\_HORISTAS puderem pertencer a um sindicato, podemos representar esse fato criando a subclasse FUNCIONARIO\_HORISTA de FUNCIONARIO e relacionando a subclasse a um tipo de entidade SINDICATO por meio do tipo de relacionamento PERTENCE\_A, conforme ilustrado na Figura 8.1.

Resumindo, o processo de especialização nos permite fazer o seguinte:

- Definir um conjunto de subclasses de um tipo de entidade.
- Estabelecer atributos específicos adicionais com cada subclasse.
- Estabelecer tipos de relacionamento específico adicionais entre cada subclasse e outros tipos de entidade ou outras subclasses.

## 8.2.2 Generalização

Podemos pensar em um *processo reverso* da abstração em que suprimimos as diferenças entre vários tipos de entidade, identificamos suas características comuns e as **generalizamos** em uma única **superclasse** da qual os tipos de entidade originais são **subclasses** especiais. Por exemplo, considere os tipos de entidade CARRO e CAMINHAO mostrados na Figura 8.3(a). Como eles têm vários atributos comuns, podem ser generalizados no tipo de entidade VEICULO, como mostra a Figura 8.3(b). Tanto CARRO quanto CAMINHAO agora são subclasses da **superclasse generalizada** VEICULO. Usamos o termo **generalização** para nos referir ao processo de definição de um tipo de entidade generalizado com base nos tipos de entidade dados.

Observe que o processo de generalização pode ser visto como sendo funcionalmente o inverso do processo de especialização. Assim, na Figura 8.3, podemos ver {CARRO, CAMINHAO} como uma especialização de VEICULO, em vez de VEICULO como uma generalização de CARRO e CAMINHAO. De modo semelhante, na Figura 8.1, podemos ver FUNCIONARIO como uma generalização de SECRETARIA, TECNICO e ENGENHEIRO. Uma notação diagramática para distinguir generalização de especialização é usada em algumas metodologias de projeto. Uma seta apontando para a superclasse ge-

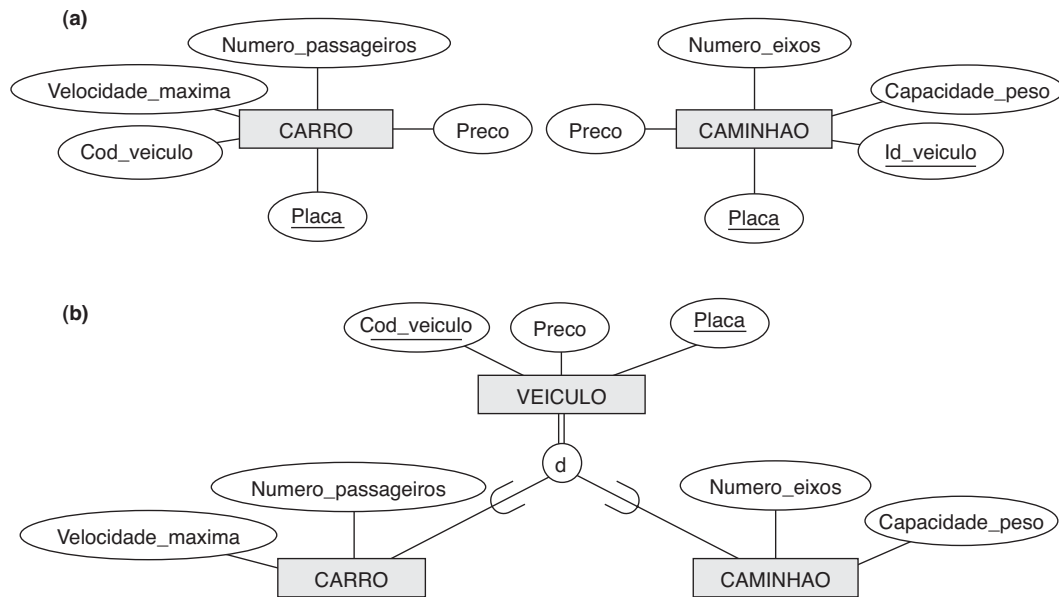


Figura 8.3

Generalização. (a) Dois tipos de entidade, CARRO e CAMINHAO. (b) Generalizando CARRO e CAMINHAO na superclasse VEICULO.

neralizada representa uma generalização, ao passo que setas apontando para subclasses especializadas representam uma especialização. Não usaremos essa notação porque a decisão sobre qual processo é seguido em determinada situação costuma ser subjetiva. O Apêndice A oferece algumas notações diagramáticas alternativas sugeridas para diagramas de esquema e diagramas de classe.

Até aqui, apresentamos os conceitos de subclasses e relacionamentos de superclasse/subclasse, bem como os processos de especialização e generalização. Em geral, uma superclasse ou subclasse representa uma coleção de entidades do mesmo tipo e, portanto, também descreve um *tipo de entidade*; é por isso que as superclasses e subclasses são todas mostradas em retângulos nos diagramas EER, como os tipos de entidade. Em seguida, discutimos as propriedades das especializações e generalizações com mais detalhes.

### 8.3 Restrições e características das hierarquias de especialização e generalização

Primeiro, vamos discutir as restrições que se aplicam a uma única especialização ou a uma única generalização. Para abreviar, nossa discussão refere-se apenas à *especialização*, embora se aplique também à generalização. Depois, vamos discutir as diferenças entre *reticulado* (*herança múltipla*) e *hierarquias* (*herança simples*) de especialização/generalização, e detalhar as diferenças entre os processos de especiali-

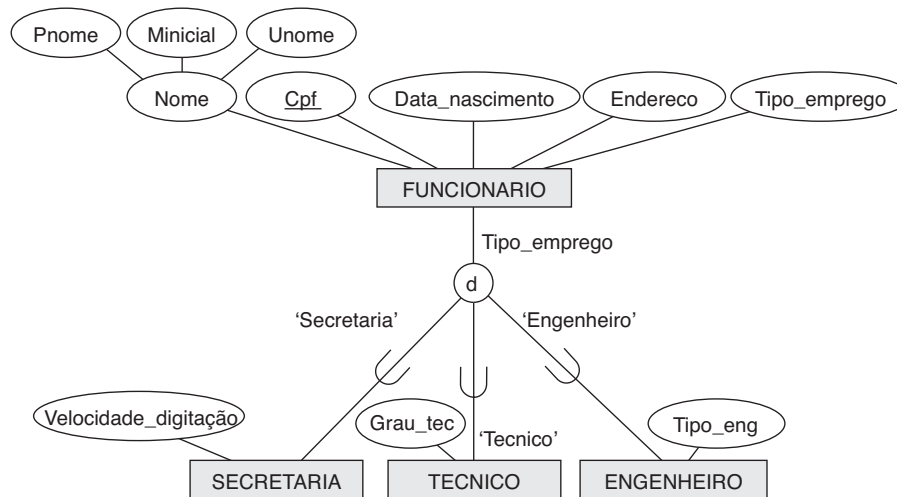
zação e generalização durante o projeto de esquema de banco de dados conceitual.

#### 8.3.1 Restrições sobre especialização e generalização

Em geral, podemos ter várias especializações definidas no mesmo tipo de entidade (ou superclasse), como mostra a Figura 8.1. Nesse caso, as entidades podem pertencer a subclasses em cada uma das especializações. Contudo, uma especialização também pode consistir em uma *única* subclasse apenas, como a especialização {GERENTE} na Figura 8.1; em tal caso, não usamos a notação de círculo.

Em algumas especializações, podemos determinar exatamente as entidades que se tornarão membros de cada subclasse ao colocar uma condição sobre o valor de algum atributo da superclasse. Essas subclasses são chamadas **subclasses definidas por predicado** (ou **definidas por condição**). Por exemplo, se o tipo de entidade FUNCIONARIO tiver um atributo Tipo\_emprego, como mostra a Figura 8.4, podemos especificar a condição de membro na subclasse SECRETARIA pela condição (Tipo\_emprego = 'Secretaria'), que chamamos de **predicado de definição** da subclasse. Essa condição é uma *restrição* que especifica exatamente que aquelas entidades do tipo de entidade FUNCIONARIO, cujo valor de atributo para Tipo\_emprego é 'Secretaria', pertencem à subclasse. Apresentamos uma subclasse definida pelo predicado escrevendo a condição de predicado ao lado da linha que conecta a subclasse ao círculo de especialização.



**Figura 8.4**

Notação do diagrama EER para uma especialização definida por atributo sobre Tipo\_emprego.

Se *todas* as subclasses em uma especialização tiverem sua condição de membro no *mesmo* atributo da superclasse, a própria especialização é chamada de **especialização definida por atributo**, e o atributo é chamado de **atributo de definição** da especialização.<sup>6</sup> Nesse caso, todas as entidades com o mesmo valor para o atributo pertencem à mesma subclasse. Apresentamos uma especialização definida por atributo colocando o nome do atributo de definição ao lado do arco que vai do círculo à superclasse, como mostra a Figura 8.4.

Quando não temos uma condição para determinar os membros em uma subclasse, diz-se que esta é **definida pelo usuário**. A condição de membro nessa subclasse é determinada pelos usuários do banco de dados quando eles aplicam a operação para incluir uma entidade à subclasse; logo, a condição de membro é *especificada individualmente para cada entidade pelo usuário*, e não por qualquer condição que possa ser avaliada automaticamente.

Duas outras restrições podem se aplicar a uma especialização. A primeira é a **restrição de disjunção** (ou **desconexão**), que especifica que as subclasses da especialização devem ser disjuntas. Isso significa que uma entidade pode ser um membro de *no máximo* uma das subclasses da especialização. Uma especialização que é definida por atributo implica a restrição de disjunção (se o atributo usado para definir o predicado de membro for de valor único). A Figura 8.4 ilustra esse caso, onde o **d** no círculo significa *disjunção*. A notação **d** também se aplica a subclasses definidas pelo usuário de uma especialização que precisa

ser disjunta, conforme ilustrado pela especialização {FUNCIONARIO\_HORISTA, FUNCIONARIO\_MENSAL} na Figura 8.1. Se as subclasses não forem restringidas a serem disjuntas, seus conjuntos de entidades podem ser **sobrepostos**; ou seja, a mesma entidade (mundo real) pode ser um membro de mais de uma subclasse da especialização. Esse caso, que é o padrão, é exibido colocando-se um **o** (de **overlapping**) no círculo, como mostra a Figura 8.5.

A segunda restrição sobre a especialização é chamada de **restrição de completude** (ou **totalidade**), que pode ser total ou parcial. Uma restrição de **especialização total** especifica que *toda* entidade na superclasse precisa ser um membro de pelo menos uma subclasse na especialização. Por exemplo, se todo FUNCIONARIO tiver de ser um FUNCIONARIO\_HORISTA ou um FUNCIONARIO\_MENSAL, então a especialização {FUNCIONARIO\_HORISTA, FUNCIONARIO\_MENSAL} da Figura 8.1 é uma especialização total de FUNCIONARIO. Isso é mostrado nos diagramas EER usando uma linha dupla para conectar a superclasse ao círculo. Uma linha simples é utilizada para exibir uma **especialização parcial**, que permite que uma entidade não pertença a qualquer uma das subclasses. Por exemplo, se algumas entidades FUNCIONARIO não pertencerem a nenhuma das subclasses {SECRETARIA, ENGENHEIRO, TECNICO} nas figuras 8.1 e 8.4, então essa especialização será parcial.<sup>7</sup>

Observe que as restrições de disjunção e completude são *independentes*. Logo, temos quatro restrições possíveis na especialização:

<sup>6</sup> Tal atributo é chamado de *discriminador* em terminologia UML.

<sup>7</sup> A notação de usar linhas simples ou duplas é semelhante à da participação parcial ou total de um tipo de entidade em um tipo de relacionamento, conforme descrito no Capítulo 7.

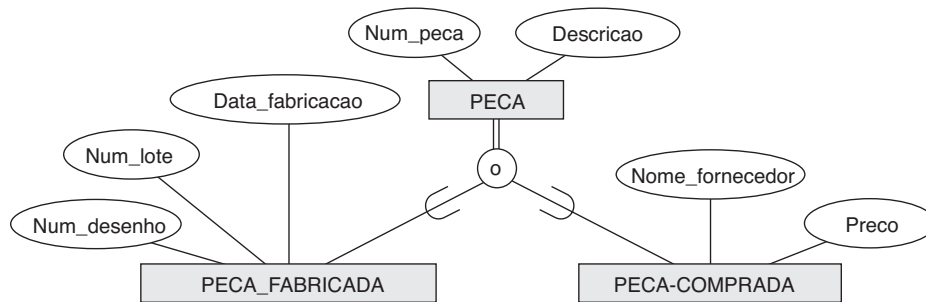


Figura 8.5

Notação de diagrama EER para uma especialização sobreposta (não disjunta).

- Disjunção, total.
- Disjunção, parcial.
- Sobreposição, total.
- Sobreposição, parcial.

Naturalmente, a restrição correta é determinada com base no significado do mundo real que se aplica a cada especialização. Em geral, uma superclasse que foi identificada por meio do processo de *generalização* costuma ser *total*, pois a superclasse é *derivada das* subclasses e, portanto, contém apenas as entidades que estão nas subclasses.

Certas regras de inserção e exclusão se aplicam à especialização (e generalização) como uma consequência das restrições especificadas anteriormente. Algumas dessas regras são as seguintes:

- Excluir uma entidade de uma superclasse implica que ela seja automaticamente excluída de todas as subclasses às quais pertence.
- Inserir uma entidade em uma superclasse implica que a entidade seja obrigatoriamente inserida em todas as subclasses *definidas por predicado* (ou *definidas por atributo*) para as quais a entidade satisfaz o predicado de definição.
- Inserir uma entidade em uma superclasse de uma *especialização total* implica que a entidade é obrigatoriamente inserida em, pelo menos, uma das subclasses da especialização.

O leitor é encorajado a fazer uma lista completa das regras para inserções e exclusões para os vários tipos de especializações.

### 8.3.2 Hierarquias e reticulado da especialização e generalização

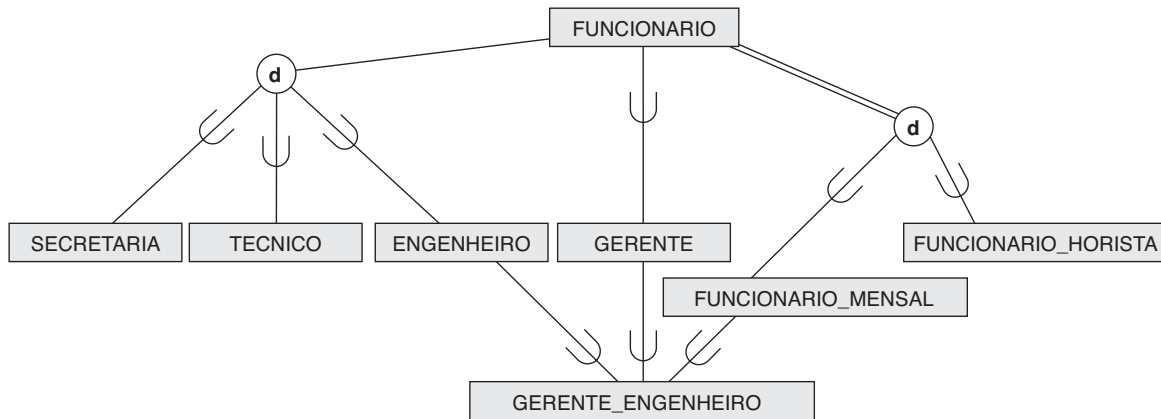
A própria subclasse pode ter mais subclasses especificadas nela, formando uma hierarquia ou um reticulado de especializações. Por exemplo, na Figura 8.6, ENGENHEIRO é uma subclasse de FUNCIONARIO

e também uma superclasse de GERENTE\_ENGENHEIRO. Isso representa a restrição do mundo real de que cada gerente engenheiro precisa ser um engenheiro. Uma **hierarquia de especialização** tem a restrição de que cada subclasse participa *como uma subclasse* em *apenas um* relacionamento de classe/subclasse; ou seja, cada subclasse tem apenas um pai, que resulta em uma **estrutura de árvore** ou **hierarquia estrita**. Ao contrário, para um **reticulado de especialização**, uma subclasse pode ser uma subclasse em *mais de um* relacionamento de classe/subclasse. Assim, a Figura 8.6 é um reticulado.

A Figura 8.7 mostra outro reticulado de especialização de mais de um nível. Isso pode ser parte de um esquema conceitual para um banco de dados UNIVERSIDADE. Observe que essa organização teria sido uma hierarquia, exceto para a subclasse ALUNO\_COLABORADOR, que é uma subclasse em dois relacionamentos de classe/subclasse distintos.

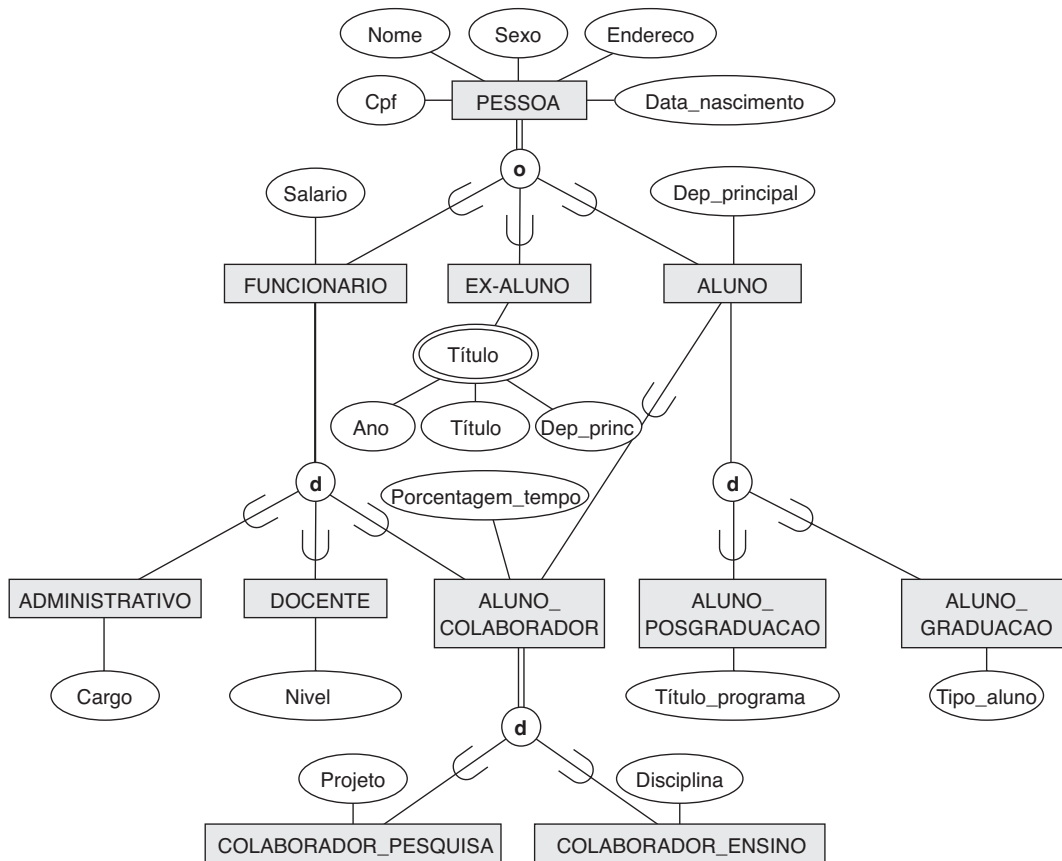
Os requisitos para a parte do banco de dados UNIVERSIDADE mostrados na Figura 8.7 são os seguintes:

1. O banco de dados registra três tipos de pessoas: funcionários, ex-alunos e alunos. Uma pessoa pode pertencer a um, dois ou a todos esses três tipos. Cada pessoa tem um nome, CPF, sexo, endereço e data de nascimento.
2. Cada funcionário tem um salário, e existem três tipos de funcionários: docente, administrativo e aluno colaborador. Cada funcionário pertence a exatamente um desses tipos. Para cada ex-aluno, é mantido um registro do título ou dos títulos que ele obteve na universidade, incluindo o nome do título, o ano em que o título foi concedido e o curso em que se formou. Cada aluno tem um departamento principal.
3. Cada membro do corpo docente tem um nível, enquanto cada membro administrativo tem um cargo administrativo. Os alunos colaboradores são classificados ainda como colabora-



**Figura 8.6**

Um reticulado de especialização com a subclasse compartilhada GERENTE\_ENGENHEIRO.



**Figura 8.7**

Um reticulado de especialização com herança múltipla para um banco de dados UNIVERSIDADE.

dores de pesquisa ou colaboradores de ensino, e a porcentagem de tempo em que eles trabalham é registrada no banco de dados. Os colaboradores de pesquisa têm seu projeto de pesquisa armazenado, ao passo que os colaboradores de ensino têm a disciplina atual em que trabalham.

4. Os alunos são classificados ainda como pós-graduação ou graduação, com os atributos espe-

cíficos de título do programa (mestrado, doutorado, M.B.A. etc.) para alunos de pós-graduação formados e tipo de aluno (novato, segundo ano etc.) para os alunos de graduação.

Na Figura 8.7, todas as entidades de pessoa representadas no banco de dados são membros do tipo de entidade PESSOA, que é especializado nas subclasses



{FUNCIONARIO, EX-ALUNO, ALUNO}. Essa especialização é sobreposta; por exemplo, um ex-aluno também pode ser um funcionário e ainda pode ser um aluno buscando um título avançado. A subclasse ALUNO é a superclasse para a especialização {ALUNO\_POSGRADUACAO, ALUNO\_GRADUACAO}, enquanto FUNCIONARIO é a superclasse para a especialização {ALUNO\_COLABORADOR, DOCENTE, ADMINISTRATIVO}. Observe que ALUNO\_COLABORADOR também é uma subclasse de ALUNO. Finalmente, ALUNO\_COLABORADOR é a superclasse para a especialização em {COLABORADOR\_PESQUISA, COLABORADOR\_ENSINO}.

Em tal reticulado ou hierarquia de especialização, uma subclasse herda os atributos não apenas de sua superclasse direta, mas também de todas as suas superclasses predecessoras, *até chegar à raiz* da hierarquia ou reticulado, se necessário. Por exemplo, uma entidade em ALUNO\_POSGRADUACAO herda todos os atributos dessa entidade como um ALUNO e como uma PESSOA. Observe que uma entidade pode existir em vários *nós folha* da hierarquia, em que um *nó folha* é uma classe que *não tem subclasses próprias*. Por exemplo, um membro de ALUNO\_GRADUACAO também pode ser um membro de COLABORADOR\_PESQUISA.

Uma subclasse com *mais de uma* superclasse é chamada de **subclasse compartilhada**, como GERENTE\_ENGENHEIRO na Figura 8.6. Isso leva ao conceito conhecido como **herança múltipla**, na qual a subclasse compartilhada GERENTE\_ENGENHEIRO herda diretamente atributos e relacionamentos de múltiplas classes. Observe que a existência de pelo menos uma subclasse compartilhada leva a um reticulado (e, portanto, à *herança múltipla*). Se não existisse qualquer subclasse compartilhada, teríamos uma hierarquia em vez de um reticulado, e somente a **herança simples** existiria. Uma regra importante relacionada à herança múltipla pode ser ilustrada pelo exemplo da subclasse compartilhada ALUNO\_COLABORADOR na Figura 8.7, que herda atributos de FUNCIONARIO e ALUNO. Aqui, tanto FUNCIONARIO quanto ALUNO herdam *os mesmos atributos* de PESSOA. A regra declara que, se um atributo (ou relacionamento) originado na *mesma superclasse* (PESSOA) é herdado mais de uma vez por caminhos diferentes (FUNCIONARIO e ALUNO) no reticulado, então ele deverá ser incluído apenas uma vez na subclasse compartilhada (ALUNO\_COLABORADOR). Logo, os atributos de PESSOA são herdados *apenas uma vez* na subclasse ALUNO\_COLABORADOR da Figura 8.7.

É importante observar aqui que alguns modelos e linguagens são limitados à **herança simples** e *não permitem* a herança múltipla (subclasses comparti-

lhadas). Também é importante observar que alguns modelos não permitem que uma entidade tenha tipos múltiplos, e, portanto, uma entidade pode ser membro de *apenas uma classe folha*.<sup>8</sup> Em tal modelo, é necessário criar subclasses adicionais como nós folha para cobrir todas as combinações possíveis de classes que podem ter alguma entidade que pertença a todas essas classes simultaneamente. Por exemplo, em uma especialização de sobreposição de PESSOA para {FUNCIONARIO, EX-ALUNO, ALUNO} (ou {F, E, A} para abreviar), seria necessário criar sete subclasses de PESSOA a fim de cobrir todos os tipos de entidades possíveis: F, E, A, F\_E, F\_A, E\_A e F\_E\_A. Obviamente, isso pode gerar uma complexidade extra.

Embora tenhamos usado a especialização para ilustrar nossa discussão, conceitos semelhantes se *aplicam igualmente* à generalização, conforme mencionamos no início desta seção. Logo, também podemos falar de **hierarquias de generalização** e **reticulados de generalização**.

### 8.3.3 Utilizando especialização e generalização no refinamento de esquemas conceituais

Agora, vamos detalhar as diferenças entre os processos de especialização e generalização, e como eles são usados para refinar os esquemas conceituais durante o projeto conceitual do banco de dados. No processo de especialização, normalmente começamos com um tipo de entidade e depois definimos subclasses do tipo de entidade pela especialização sucessiva; ou seja, definimos repetidamente agrupamentos mais específicos do tipo de entidade. Por exemplo, ao projetar o reticulado de especialização da Figura 8.7, podemos primeiro especificar um tipo de entidade PESSOA para um banco de dados de universidade. Depois, descobriremos que três tipos de pessoas serão representados no banco de dados: funcionários da universidade, ex-alunos e alunos. Criamos a especialização {FUNCIONARIO, EX-ALUNO, ALUNO} para essa finalidade e escolhemos a restrição de sobreposição, pois uma pessoa pode pertencer a mais de uma das subclasses. Especializamos FUNCIONARIO ainda mais em {DOCENTE, ADMINISTRATIVO, ALUNO\_COLABORADOR} e especializamos ALUNO em {ALUNO\_GRADUACAO, ALUNO\_POSGRADUACAO}. Por fim, especializamos ALUNO\_COLABORADOR em {COLABORADOR\_PESQUISA, COLABORADOR\_ENSINO}. Essa especialização sucessiva corresponde a um **processo de refinamento conceitual de cima para baixo (top-down)** durante o projeto do esquema conceitual. Até aqui, temos uma hierarquia; depois, obser-

<sup>8</sup> Em alguns modelos, a classe é restringida ainda mais para ser um *nó folha* na hierarquia ou no reticulado.

vamos que ALUNO\_COLABORADOR é uma subclasse compartilhada, pois ela também é uma subclasse de ALUNO, o que leva ao reticulado.

É possível chegar à mesma hierarquia ou reticulado vindo de outra direção. Nesse caso, o processo envolve a generalização, em vez da especialização, e corresponde a uma **síntese conceitual de baixo para cima (bottom-up)**. Por exemplo, os projetistas de banco de dados podem primeiro descobrir tipos de entidade como ADMINISTRATIVO, DOCENTE, EX-ALUNO, ALUNO\_GRADUACAO, ALUNO\_POSGRADUACAO, COLABORADOR\_PESQUISA, COLABORADOR\_ENSINO, e assim por diante; depois, eles podem generalizar {ALUNO\_GRADUACAO, ALUNO\_POSGRADUACAO} para ALUNO; daí generalizar {COLABORADOR\_PESQUISA, COLABORADOR\_ENSINO} para ALUNO\_COLABORADOR; então generalizar {ADMINISTRATIVO, DOCENTE, ALUNO\_COLABORADOR} para FUNCIONARIO; e, finalmente, generalizar {FUNCIONARIO, EX-ALUNO, ALUNO} para PESSOA.

Em termos estruturais, hierarquias ou reticulados resultantes de qualquer processo podem ser idênticas. A única diferença relaciona-se à maneira ou ordem em que as superclasses e subclasses do esquema foram criadas durante o processo de projeto. Na prática, é provável que nem o processo de generalização nem de especialização seja seguido estritamente, mas que seja empregada uma combinação dos dois processos. Novas classes são incorporadas de maneira contínua em uma hierarquia ou reticulado à medida que elas se tornam aparentes aos usuários e projetistas. Observe que a noção de representar dados e conhecimento usando hierarquias e reticulados de superclasse/subclasse é muito comum em sistemas baseados em conhecimento e sistemas especialistas, que combinam a tecnologia de banco de dados com técnicas de inteligência artificial. Por exemplo, esquemas de representação do conhecimento baseada em quadro são muito semelhantes às hierarquias de classes. A especialização também é comum nas metodologias de projeto de engenharia de software que são baseadas no paradigma orientado a objeto.

## 8.4 Modelagem dos tipos UNIAO usando categorias

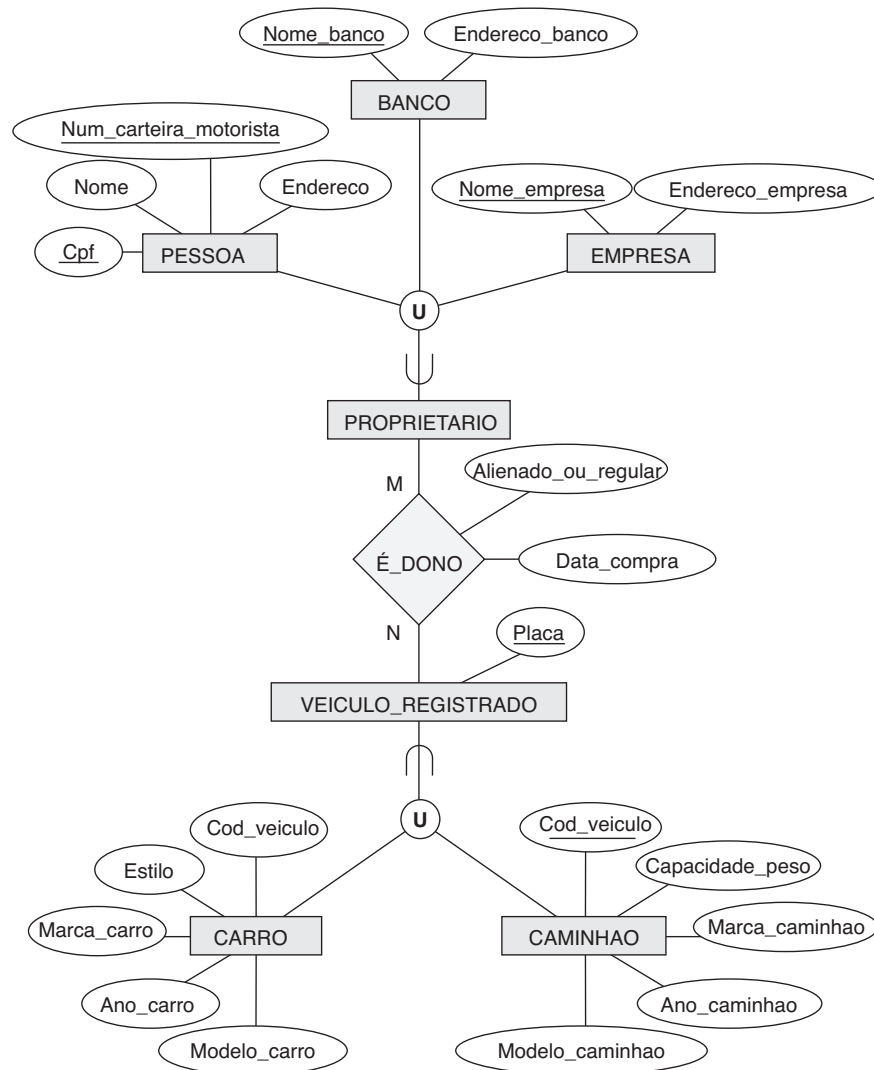
Todos os relacionamentos de superclasse/subclasse que vimos até aqui têm uma *única superclasse*. Uma subclasse compartilhada como GERENTE\_ENGENHEIRO no reticulado da Figura 8.6 é a subclasse em três relacionamentos de superclasse/subclasse *dis-*

*tintos*, em que cada um dos três relacionamentos tem uma *única* superclasse. Porém, às vezes é necessário representar um único relacionamento de superclasse/subclasse com *mais de uma* superclasse, no qual as superclasses representam diferentes tipos de entidade. Nesse caso, a subclasse representará uma coleção de objetos que é um subconjunto da UNIAO de tipos de entidade distintos. Chamamos essa *subclasse de tipo de união* ou *categoria*.<sup>9</sup>

Por exemplo, suponha que tenhamos três tipos de entidade: PESSOA, BANCO e EMPRESA. Em um banco de dados para registro de veículo a motor, o proprietário de um veículo pode ser uma pessoa, um banco (mantendo uma alienação sobre um veículo) ou uma empresa. Precisamos criar uma classe (coleção de entidades) que inclui entidades de todos os três tipos para desempenhar o papel de *proprietário de veículo*. Uma categoria (tipo de união) PROPRIETARIO, que é uma *subclasse da UNIÃO* dos três conjuntos de entidades de EMPRESA, BANCO e PESSOA, pode ser criada para essa finalidade. Exibimos categorias no diagrama EER como mostra a Figura 8.8. As superclasses EMPRESA, BANCO e PESSOA são conectadas ao círculo com o símbolo  $\cup$ , que significa *operação de união de conjuntos*. Um arco com o símbolo de subconjunto conecta o círculo à categoria (subclasse) PROPRIETARIO. Se um predicado de definição for necessário, ele é exibido ao lado da linha da superclasse à qual o predicado se aplica. Na Figura 8.8, temos duas categorias: PROPRIETARIO, que é uma subclasse da união de PESSOA, BANCO e EMPRESA; e VEICULO\_REGISTRADO, que é uma subclasse da união de CARRO e CAMINHAO.

Uma categoria tem duas ou mais superclasses que podem representar *tipos de entidade distintos*, enquanto outros relacionamentos de superclasse/subclasse sempre têm uma única superclasse. Para entender melhor a diferença, podemos comparar uma categoria, como PROPRIETARIO da Figura 8.8, com a subclasse compartilhada GERENTE\_ENGENHEIRO da Figura 8.6. Esta última é uma subclasse de *cada uma* das três superclasses ENGENHEIRO, GERENTE e FUNCIONARIO\_MENSAL, de modo que uma entidade que é um membro de GERENTE\_ENGENHEIRO deva existir em *todas as três*. Isso representa a restrição de que um gerente de engenharia precisa ser um ENGENHEIRO, um GERENTE e um FUNCIONARIO\_MENSAL; ou seja, GERENTE\_ENGENHEIRO é um subconjunto da *interseção* das três classes (conjuntos de entidades). Por sua vez, uma categoria é um subconjunto da *união* de suas su-

<sup>9</sup> Nosso uso do termo *categoria* é baseado no modelo ECR (Entity-Category-Relationship, ou entidade-categoria-relacionamento) (Elmasri et al., 1985).

**Figura 8.8**

Duas categorias (tipos de união): PROPRIETARIO e VEICULO\_REGISTRADO.

perclasses. Logo, uma entidade que é um membro de PROPRIETARIO deve existir em *apenas uma* das superclasses. Isso representa a restrição de que um PROPRIETARIO pode ser uma EMPRESA, um BANCO *ou* uma PESSOA na Figura 8.8.

A herança de atributo funciona de maneira mais seletiva no caso de categorias. Por exemplo, na Figura 8.8, cada entidade PROPRIETARIO herda os atributos de uma EMPRESA, uma PESSOA ou um BANCO, dependendo da superclasse à qual a entidade pertence. Por sua vez, uma subclasse compartilhada como GERENTE\_ENGENHEIRO (Figura 8.6) herda *todos* os atributos de suas superclasses FUNCIONARIO\_MENSAL, ENGENHEIRO e GERENTE.

É interessante observar a diferença entre a categoria VEICULO\_REGISTRADO (Figura 8.8) e a superclas-

se generalizada VEICULO (Figura 8.3(b)). Na Figura 8.3(b), todo carro e todo caminhão é um VEICULO; mas, na Figura 8.8, a categoria VEICULO\_REGISTRADO inclui alguns carros e alguns caminhões, mas não necessariamente todos eles (por exemplo, alguns carros ou caminhões podem não ser registrados). Em geral, uma especialização ou generalização como a da Figura 8.3(b), se fosse *parcial*, não impediria VEICULO de conter outros tipos de entidades, como motocicletas. Porém, uma categoria como VEICULO\_REGISTRADO na Figura 8.8 implica que somente carros e caminhões, mas não outros tipos de entidades, possam ser membros de VEICULO\_REGISTRADO.

Uma categoria pode ser **total** ou **parcial**. A total mantém a *união* de todas as entidades em suas superclasses, enquanto a parcial pode manter um *subconjunto da união*. Uma categoria total é represen-



tada em diagrama por uma linha dupla que conecta a categoria e o círculo, ao passo que uma categoria parcial é indicada por uma linha simples.

As superclasses de uma categoria podem ter diferentes atributos de chave, conforme demonstrado pela categoria PROPRIETARIO da Figura 8.8, ou podem ter o mesmo atributo de chave, conforme demonstrado pela categoria VEICULO\_REGISTRADO. Observe que, se uma categoria é total (não parcial), ela pode ser representada alternativamente como uma especialização total (ou uma generalização total). Nesse caso, a escolha de qual representação usar é subjetiva. Se as duas classes representam o mesmo tipo de entidades e compartilham diversos atributos, incluindo os mesmos atributos-chave, a especialização/generalização é preferida; caso contrário, a categorização (tipo de união) é mais apropriada.

É importante observar que algumas metodologias de modelagem não possuem tipos de união. Nesses modelos, um tipo de união precisa ser representado de uma maneira indireta (ver Seção 9.2).

## 8.5 Um exemplo de esquema

### UNIVERSIDADE de EER, opções de projeto e definições formais

Nesta seção, primeiro vamos dar um exemplo de esquema de banco de dados no modelo EER para ilustrar o uso dos diversos conceitos discutidos aqui e no Capítulo 7. Depois, vamos discutir as escolhas de projeto para esquemas conceituais e, por fim, resumir os conceitos do modelo EER e defini-los formalmente da mesma maneira que fizemos com os conceitos do modelo ER básico, no Capítulo 7.

#### 8.5.1 O exemplo do banco de dados

##### UNIVERSIDADE

Para nosso exemplo de aplicação de banco de dados, considere um banco de dados UNIVERSIDADE que registra alunos e suas disciplinas, históricos e registro, bem como as ofertas de curso da universidade. O banco de dados também registra os projetos de pesquisa patrocinados do corpo docente e dos alunos de pós-graduação. Esse esquema aparece na Figura 8.9. Uma discussão dos requisitos que levaram a esse esquema pode ser vista em seguida.

Para cada pessoa, o banco de dados mantém informações sobre o nome dela [Nome], número do Cadastro de Pessoa Física [Cpf], endereço [Endereco], sexo [Sexo] e data de nascimento [Data\_nasc].

Duas subclasses do tipo de entidade PESSOA são identificadas: DOCENTE e ALUNO. Atributos específicos de DOCENTE são a classificação [Classificacao] (assistente, associado, adjunto, pesquisador, visitante etc.), escritório [Doc\_escritorio], telefone do escritório [Doc\_telefone] e salário [Salario]. Todos os membros do corpo docente estão relacionados a departamento(s) acadêmico(s) ao(s) qual(is) eles estão afiliados [PERTENCE] (um membro do corpo docente pode ser associado a vários departamentos, de modo que o relacionamento é M:N). Um atributo específico de ALUNO é [Tipo\_aluno] (novato = 1, segundo ano = 2, ..., aluno formado = 5). Cada ALUNO também está relacionado a seus departamentos principal e secundário (se forem conhecidos) [PRINCIPAL] e [SECUNDARIO], às turmas da disciplina que está frequentando atualmente e às disciplinas completadas [HISTORICO\_ESCOLAR]. Cada instância de HISTORICO\_ESCOLAR inclui a nota que o aluno recebeu [Nota] em uma turma de um curso.

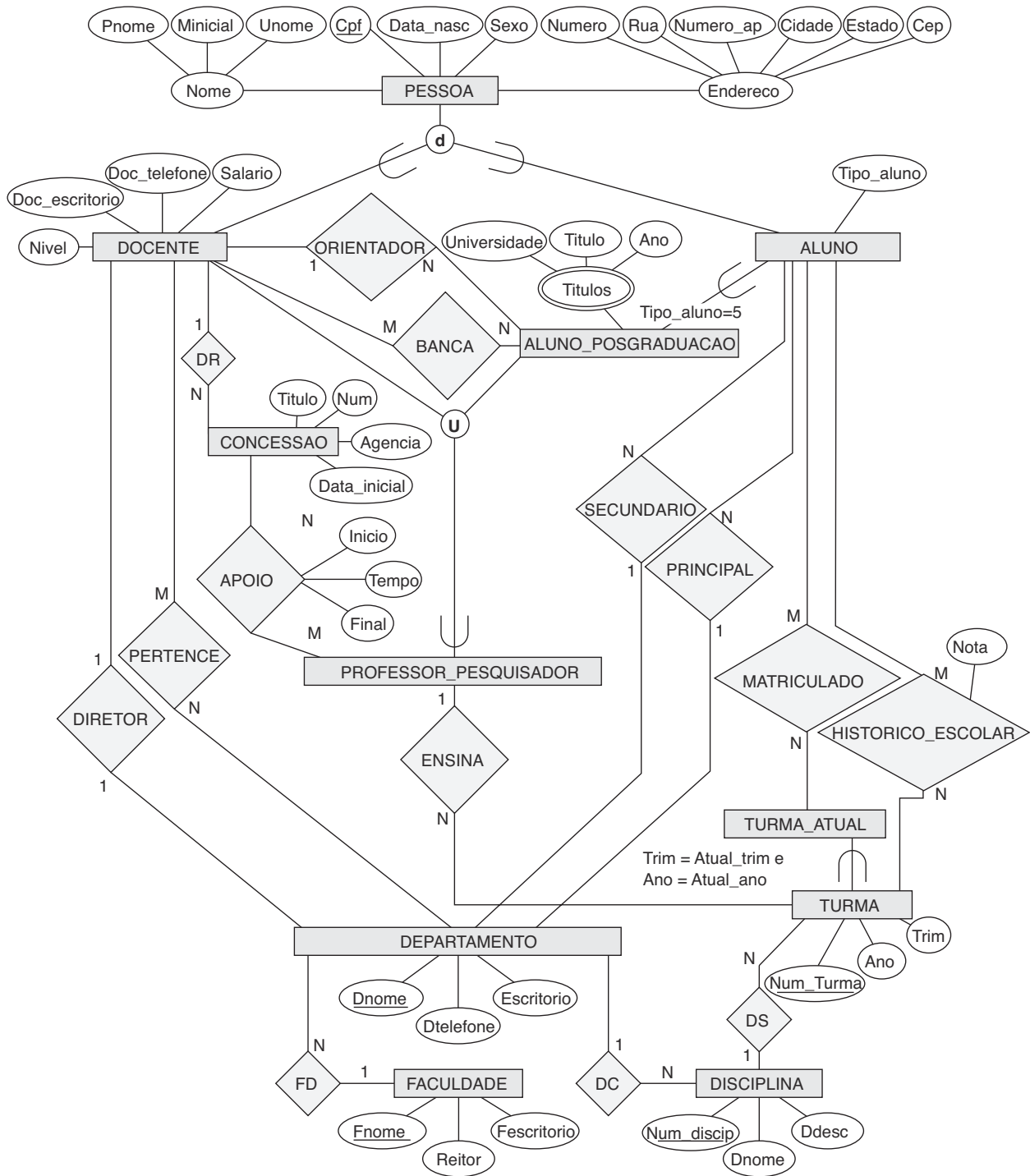
ALUNO\_POSGRADUACAO é uma subclasse de ALUNO, com o predicado de definição Tipo\_aluno = 5. Para cada aluno de pós-graduação, mantemos uma lista dos títulos anteriores em um atributo composto, multivalorado [Títulos]. Também relacionamos o aluno formado a um orientador acadêmico [ORIENTADOR] e a um comitê de tese [BANCA], se existir.

Um departamento acadêmico tem os atributos nome [Dnome], telefone [Dtelefone] e número de escritório [Escritorio], e está relacionado ao membro acadêmico que é seu DIRETOR e à faculdade à qual pertence [DF]. Cada faculdade tem como atributos o nome da faculdade [Fnome], número do escritório [Fescritorio] e o nome de seu reitor [Reitor].

Uma disciplina tem os atributos número da disciplina [Num\_discip], nome da disciplina [Dnome] e descrição da disciplina [Ddesc]. São oferecidas várias turmas de cada disciplina, com cada uma tendo os atributos número da turma [Num\_turma], o ano e trimestre em que foi oferecida ([Ano] e [Trim]).<sup>10</sup> Os números de turma identificam cada uma de maneira exclusiva. As turmas oferecidas durante o trimestre atual estão em uma subclasse TURMA\_ATUAL de TURMA, com o predicado de definição Trim = Atual\_trim e Ano = Atual\_ano. Cada turma está relacionada ao professor que lecionou ou está lecionando ([ENSINA]), se ele estiver no banco de dados.

A categoria PROFESSOR\_PESQUISADOR é um subconjunto da união de DOCENTE e ALUNO\_POSGRADUACAO e inclui todos os docentes, bem como

<sup>10</sup> Consideramos que o sistema de *trimestre*, em vez de *semestre*, seja utilizado na universidade do exemplo.



**Figura 8.9**

O esquema conceitual EER para um banco de dados UNIVERSIDADE.

alunos formados que recebem apoio por ensino ou pesquisa. Finalmente, o tipo de entidade CONCESSAO registra concessões e contratos de pesquisa outorgados à universidade. Cada concessão tem os atributos de título da concessão [Titulo], número da concessão [Num], agência de fomento [Agencia] e a data inicial [Data\_inicial]. Uma concessão está rela-

cionada a um docente responsável [DR] e a todos os pesquisadores a que ele dá apoio [APOIO]. Cada instância de apoio tem como atributos a data inicial do apoio [Inicio], a data final do apoio (se for conhecida) [Final] e a porcentagem do tempo gasto no projeto [Tempo] pelo pesquisador que recebe o apoio.

### 8.5.2 Escolhas de projeto para especialização/generalização

Nem sempre é fácil escolher o projeto conceitual mais apropriado para uma aplicação de banco de dados. Na Seção 7.7.3, apresentamos algumas das questões típicas enfrentadas por um projetista de banco de dados ao escolher entre os conceitos de tipos de entidade, tipos de relacionamento e atributos para representar uma situação em particular do minimundo como um esquema ER. Nesta seção, vamos discutir as diretrizes e escolhas de projeto para os conceitos EER de especialização/generalização e categorias (tipos de união).

Conforme mencionamos na Seção 7.7.3, o projeto conceitual do banco de dados deve ser considerado um processo de refinamento iterativo, até que o projeto mais adequado seja alcançado. As orientações a seguir ajudam a guiar o processo de projeto para conceitos de EER:

- Em geral, muitas especializações e subclasses podem ser definidas para tornar o modelo conceitual preciso. No entanto, a desvantagem é que o projeto se torna muito confuso. É importante representar apenas as subclasses que se julgue necessárias para evitar uma aglomeração extrema do esquema conceitual.
- Se uma subclasse possui poucos atributos específicos (locais) e nenhum relacionamento específico, ela pode ser mesclada à superclasse. Os atributos específicos manteriam valores NULL para entidades que não são membros da subclasse. Um atributo de *tipo* poderia especificar se uma entidade é um membro da subclasse.
- De modo semelhante, se todas as subclasses da especialização/generalização tiverem alguns atributos específicos e nenhum relacionamento específico, elas podem ser mescladas à superclasse e substituídas por um ou mais atributos de *tipo* que especificam a subclasse ou subclasses a que cada entidade pertence (ver Seção 9.2 para saber como esse critério se aplica aos bancos de dados relacionais).
- Os tipos de união e categorias geralmente devem ser evitados, a menos que a situação definitivamente justifique esse tipo de construção, o que ocorre em algumas situações

práticas. Se possível, tentamos modelar usando a especialização/generalização conforme discutimos no final da Seção 8.4.

- A escolha de restrições disjuntas/sobrepostas e totais/parciais sobre a especialização/generalização é controlada pelas regras no minimundo que está sendo modelado. Se os requisitos não indicarem quaisquer restrições em particular, o padrão geralmente seria sobreposição e parcial, pois isso não especifica quaisquer restrições sobre a condição de membro da subclasse.

Como um exemplo da aplicação dessas orientações, considere a Figura 8.6, na qual nenhum atributo específico (local) aparece. Poderíamos mesclar todas as subclasses no tipo de entidade FUNCIONARIO e acrescentar os seguintes atributos a ele:

- Um atributo Tipo\_emprego cujo conjunto de valores {'Secretária', 'Engenheiro', 'Técnico'} indicaria a qual subclasse cada funcionário pertence na primeira especialização.
- Um atributo Forma\_pagto cujo conjunto de valores {'Mensal', 'Horista'} indicaria a qual subclasse cada funcionário pertence na segunda especialização.
- Um atributo É\_gerente cujo conjunto de valores {'Sim', 'Não'} indicaria se uma entidade de funcionário individual é um gerente ou não.

### 8.5.3 Definições formais para os conceitos do modelo EER

Agora, vamos resumir os conceitos do modelo EER e mostrar as definições formais. Uma **classe**<sup>11</sup> é um conjunto ou coleção de entidades, incluindo qualquer uma das construções de esquema EER das entidades de grupo, como tipos de entidade, subclasses, superclasses e categorias. Uma **subclasse**  $S$  é uma classe cujas entidades sempre precisam ser um subconjunto das entidades em outra classe, chamada **superclasse**  $C$  do **relacionamento superclasse/subclasse** (ou **É\_UM**). Indicamos tal relacionamento com  $C/S$ . Para tal relacionamento superclasse/subclasse, sempre devemos ter

$$S \subseteq C$$

Uma **especialização**  $Z = \{S_1, S_2, \dots, S_n\}$  é um conjunto de subclasses que têm a mesma superclasse  $G$ ;

<sup>11</sup> O uso da palavra *classe* aqui difere de sua utilização mais comum nas linguagens de programação orientadas a objeto, como C++. Em C++, uma classe é uma definição de tipo estruturada com suas funções (operações) aplicáveis.



ou seja,  $G/S_i$  é um relacionamento superclasse/subclasse para  $i = 1, 2, \dots, n$ .  $G$  é chamado de **tipo de entidade generalizada** (ou a superclasse da especialização, ou uma **generalização** das subclasses  $\{S_1, S_2, \dots, S_n\}$ ).  $Z$  é considerada **total** se sempre (em qualquer ponto no tempo) tivermos

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

Caso contrário,  $Z$  é considerada **parcial**.  $Z$  é considerada **disjunta** se sempre tivermos

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

Caso contrário,  $Z$  é considerada **sobreposta**.

Uma subclasse  $S$  de  $C$  é considerada **definida por predicado** se um predicado  $p$  sobre os atributos de  $C$  for usado para especificar quais entidades em  $C$  são membros de  $S$ ; ou seja,  $S = C[p]$ , que é o conjunto de entidades em  $C$  que satisfazem a  $p$ . Uma subclasse que não é definida por um predicado é chamada de **definida pelo usuário**.

Uma especialização  $Z$  (ou generalização  $G$ ) é considerada **definida por atributo** se um predicado ( $A = c_i$ ), onde  $A$  é um atributo de  $G$  e  $c_i$  é um valor constante do domínio de  $A$ , for usado para especificar a condição de membro em cada subclasse  $S_i$  em  $Z$ . Observe que, se  $c_i \neq c_j$  para  $i \neq j$ , e  $A$  for um atributo de único valor, então a especialização será disjunta.

A **categoria**  $T$  é uma classe que é um subconjunto da união de  $n$  que define as superclasses  $D_1, D_2, \dots, D_n$ ,  $n > 1$ , podendo ser especificada formalmente da seguinte maneira:

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

Um predicado  $p_i$  sobre os atributos de  $D_i$  pode ser usado para especificar os membros de cada  $D_i$  que são membros de  $T$ . Se um predicado for especificado sobre cada  $D_i$ , obtemos

$$T = (D_1[p_1] \cup D_2[p_2] \dots \cup D_n[p_n])$$

Agora, devemos estender a definição de **tipo de relacionamento** dada no Capítulo 7, permitindo que qualquer classe — não apenas um tipo de entidade — participe de um relacionamento. Logo, devemos substituir as palavras *tipo de entidade* por *classe* naquela definição. A notação gráfica de EER é coerente com ER porque todas as classes são representadas por retângulos.

## 8.6 Exemplo de outra notação: representando especialização e generalização em diagramas de classes em UML

Agora, vamos discutir as notações UML para especialização/generalização e herança. Já apresentamos a notação e terminologia básica do diagrama de classes UML na Seção 7.8. A Figura 8.10 ilustra um possível diagrama de classes UML correspondente ao diagrama EER da Figura 8.7. A notação básica para especialização/generalização (ver Figura 8.10) é conectar as subclasses por linhas verticais a uma linha horizontal, que tem um triângulo conectando a linha horizontal por meio de outra linha vertical até a superclasse. Um triângulo preto indica uma especialização/generalização com a restrição *disjunta*, e um triângulo contornado indica uma restrição de *sobreposição*. A superclasse raiz é chamada de **classe base**, e as subclasses (nós folhas) são chamadas de **classes de folha**.

A discussão citada e o exemplo da Figura 8.10, junto com a apresentação da Seção 7.8, oferecem uma rápida introdução aos diagramas de classes UML e sua terminologia. Focalizamos os conceitos que são relevantes à modelagem de banco de dados ER e EER, em vez dos conceitos que são mais relevantes à engenharia de software. Em UML, existem muitos detalhes que não discutimos, pois estão fora do escopo deste livro e são relevantes principalmente para a engenharia de software. Por exemplo, as classes podem ser de vários tipos:

- Classes abstratas definem atributos e operações, mas não têm objetos correspondentes a essas classes. Elas são usadas principalmente para especificar um conjunto de atributos e operações que podem ser herdados.
- Classes concretas podem ter objetos (entidades) instanciados para pertencer à classe.
- Classes genéricas (ou *template*) especificam um modelo que pode ser usado também para definir outras classes.

No projeto de banco de dados, estamos preocupados principalmente com a especificação de classes concretas, cujas coleções de objetos são permanentemente (ou persistentemente) armazenadas no banco de dados. A bibliografia selecionada ao final deste capítulo oferece algumas referências a livros que descrevem os detalhes completos de UML. Há um material adicional relacionado à UML incluído no Capítulo 10.

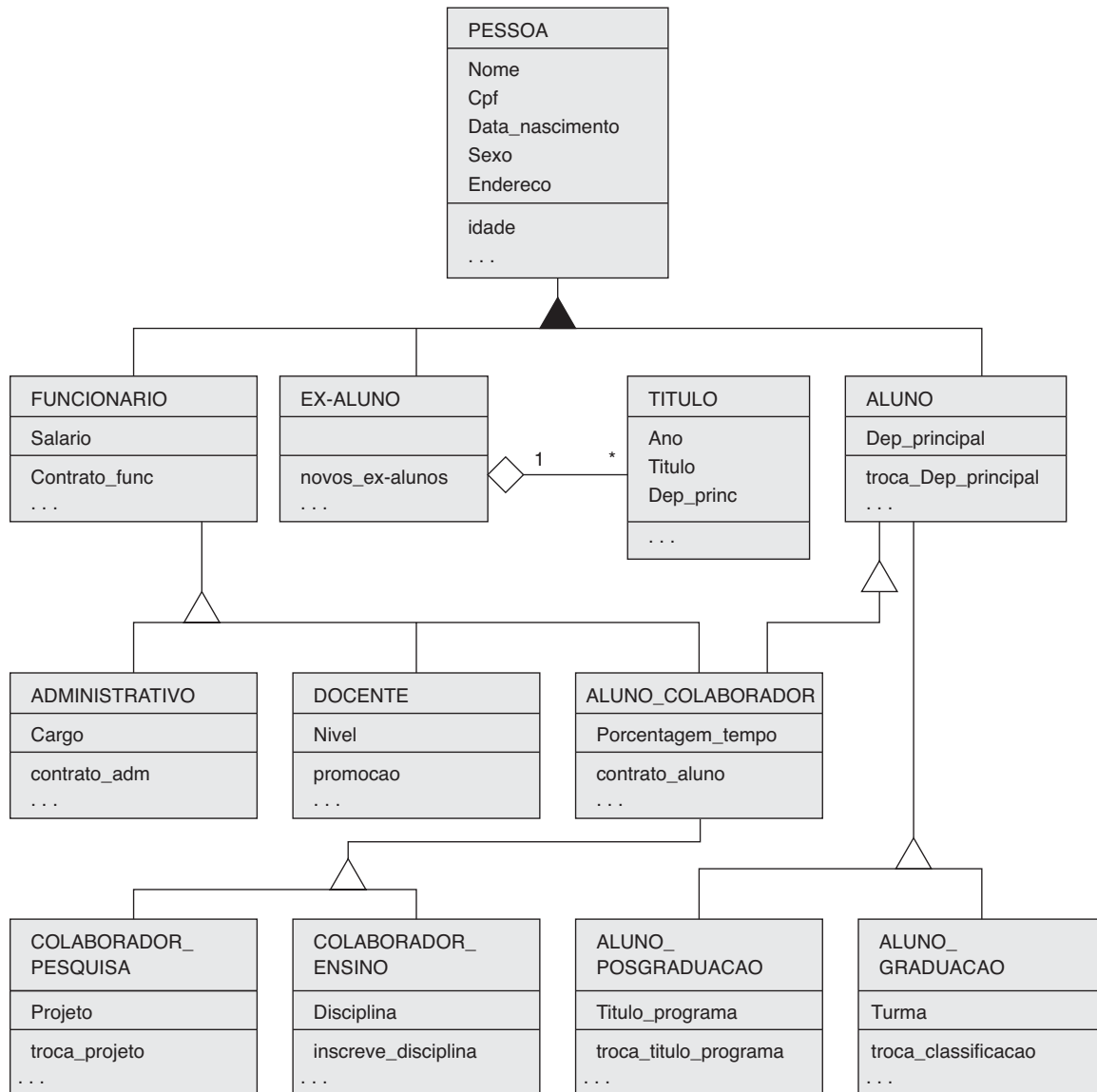


Figura 8.10

Um diagrama de classes UML correspondente ao diagrama EER da Figura 8.7, ilustrando a notação UML para especialização/generalização.

## 8.7 Conceitos de abstração de dados, representação do conhecimento e ontologia

Nesta seção, discutimos em termos gerais alguns dos conceitos de modelagem que descrevemos bem especificamente em nossa apresentação dos modelos ER e EER no Capítulo 7 e anteriormente neste capítulo. Essa terminologia não é usada apenas na modelagem de dados conceituais, mas também na literatura de inteligência artificial quando se

discute a **representação do conhecimento** (ou RC, *knowledge representation*). Esta seção discute as semelhanças e diferenças entre a modelagem conceitual e a representação do conhecimento, além de introduzir um pouco da terminologia alternativa e de alguns conceitos adicionais.

O objetivo das técnicas de RC é desenvolver conceitos para modelar com precisão algum **domínio de conhecimento**, criando uma **ontologia**<sup>12</sup> que descreve os conceitos do domínio e como esses conceitos estão inter-relacionados. Tal ontologia é usada para arma-

<sup>12</sup> Uma *ontologia* é algo semelhante a um esquema conceitual, mas com mais conhecimento, regras e exceções.

zenar e manipular o conhecimento para tirar conclusões, tomar decisões ou responder a perguntas. Os objetivos da RC são semelhantes aos dos modelos de dados semânticos, mas existem algumas semelhanças e diferenças importantes entre as duas disciplinas:

- Ambas as disciplinas usam um processo de abstração para identificar propriedades comuns e aspectos importantes de objetos no minimundo (também conhecido como *domínio de discurso* em RC), enquanto suprimem diferenças insignificantes e detalhes sem importância.
- As duas disciplinas oferecem conceitos, relacionamentos, restrições, operações e linguagens para definir dados e representar conhecimento.
- A RC geralmente é mais ampla em escopo do que os modelos de dados semânticos. Diferentes formas de conhecimento, como regras (usadas na inferência, dedução e pesquisa), conhecimento incompleto e padrão, e conhecimento temporal e espacial, são representados em esquemas RC. Os modelos de banco de dados estão sendo expandidos para incluir alguns desses conceitos (ver Capítulo 26).
- Esquemas RC incluem **mecanismos de raciocínio** que deduzem fatos adicionais dos fatos armazenados em um banco de dados. Logo, embora a maioria dos sistemas de banco de dados atuais seja limitada a responder a consultas diretas, os sistemas baseados em conhecimento que usam esquemas RC podem responder a consultas que envolvem **inferências** sobre os dados armazenados. A tecnologia de banco de dados está sendo estendida com mecanismos de inferência (ver Seção 26.5).
- Embora a maioria dos modelos de dados se concentre na representação dos esquemas de banco de dados, ou metaconhecimento, os esquemas RC costumam misturar os esquemas com as próprias instâncias, a fim de oferecer flexibilidade na representação de exceções. Isso normalmente resulta em ineficiências quando esses esquemas RC são implementados, especialmente quando comparados com bancos de dados e quando uma grande quantidade de dados (fatos) precisa ser armazenada.

Agora, vamos discutir sobre quatro **conceitos de abstração** que são usados em modelos de dados semânticos, como o modelo EER, bem como em esquemas RC: (1) classificação e instanciação, (2) identificação, (3) especialização e generalização e (4) agregação e associação. Os conceitos emparelhados

de classificação e instanciação são inversos um do outro, assim como a generalização e a especialização. Os conceitos de agregação e associação também são relacionados. Discutimos esses conceitos abstratos e sua relação com as representações concretas usadas no modelo EER para esclarecer o processo de abstração de dados e melhorar nosso conhecimento do processo relacionado de projeto de esquema conceitual. Fechamos a seção com uma rápida discussão sobre *ontologia*, que está sendo bastante usada na pesquisa recente sobre representação do conhecimento.

### 8.7.1 Classificação e instanciação

O processo de **classificação** envolve atribuir sistematicamente objetos/entidades semelhantes aos tipos classe/entidade do objeto. Agora, podemos descrever (em BD) ou raciocinar sobre (em RC) as classes em vez dos objetos individuais. Coleções de objetos que compartilham os mesmos tipos de atributos, relacionamentos e restrições são classificadas em classes, a fim de simplificar o processo de descoberta de suas propriedades. A **instanciação** é o inverso da classificação, e refere-se à geração e exame específico de objetos distintos de uma classe. Uma instância de objeto está relacionada à sua classe de objeto por um relacionamento **É\_UMA\_INSTÂNCIA\_DE** ou **É\_UM\_MEMBRO\_DE**. Embora os diagramas EER não apresentem instâncias, os diagramas UML permitem uma forma de instanciação ao possibilitar a exibição de objetos individuais. Não vamos descrever esse recurso em nossa introdução aos diagramas de classes UML.

Em geral, os objetos de uma classe devem ter uma estrutura de tipo semelhante. Contudo, alguns objetos podem exibir propriedades que diferem em alguns aspectos dos outros objetos da classe. Esses **objetos de exceção** também precisam ser modelados, e os esquemas RC permitem exceções mais variadas do que os modelos de banco de dados. Além disso, certas propriedades se aplicam à classe como um todo, e não aos objetos individuais; os esquemas RC permitem tais **propriedades de classe**. Os diagramas UML também permitem a especificação de propriedades de classe.

No modelo EER, as entidades são classificadas em tipos de entidade de acordo com seus atributos e relacionamentos básicos. As entidades são classificadas ainda em subclasses e categorias, com base nas semelhanças e diferenças adicionais (exceções) entre elas. As instâncias de relacionamento são classificadas em tipos de relacionamento. Logo, os tipos de entidade, subclasses, categorias e tipos de relacionamento são os diferentes conceitos usados para



classificação no modelo EER. O modelo EER não provê explicitamente propriedades de classificação, mas pode ser estendido para fazer isso. Em UML, os objetos são classificados, e é possível exibir tanto propriedades de classe quanto objetos individuais.

Os modelos de representação do conhecimento permitem múltiplos esquemas de classificação, em que uma classe é uma *instância* de outra classe (chamada **metaclass**). Observe que isso *não pode* ser representado diretamente no modelo EER, pois temos apenas dois níveis — classes e instâncias. O único relacionamento entre classes no modelo EER é um relacionamento de superclasse/subclasse, ao passo que em alguns esquemas RC um relacionamento adicional de classe/instância pode ser representado diretamente em uma hierarquia de classes. Uma instância pode, por si só, ser outra classe, permitindo esquemas de classificação multiníveis.

## 8.7.2 Identificação

**Identificação** é o processo de abstração pelo qual classes e objetos se tornam exclusivamente identificáveis por meio de algum **identificador**. Por exemplo, um nome de classe identifica de maneira exclusiva uma classe inteira dentro de um esquema. É necessário que haja um mecanismo adicional para distinguir instâncias de objeto distintas por meio de identificadores de objeto. Além disso, é necessário identificar múltiplas manifestações no banco de dados do mesmo objeto no mundo real. Por exemplo, podemos ter uma tupla <‘Mauro Campos’, ‘610618’, ‘3376-9821’> em uma relação PESSOA e outra tupla <‘301-540-836-51’, ‘CC’, 3,8> em uma relação ALUNO que representem a mesma entidade do mundo real. Não há como identificar o fato de que esses dois objetos de banco de dados (tuplas) representam a mesma entidade do mundo real, a menos que tomemos uma providência *em tempo de projeto* para a referência cruzada apropriada, que fornece essa identificação. Logo, a identificação é necessária em dois níveis:

- Para distinguir objetos de classes de banco de dados.
- Para identificar objetos de banco de dados e relacioná-los a seus equivalentes no mundo real.

No modelo EER, a identificação das construções de esquema é baseada em um sistema de nomes exclusivos para tais construções. Por exemplo, cada classe em um esquema EER — seja um tipo de entidade, uma subclasse, uma categoria ou um tipo de relacionamento — precisa ter um nome distinto. Os nomes de atributos de determinada classe também precisam ser distintos. As regras para identificar refe-

rências de nome de atributo sem ambiguidade em um reticulado ou hierarquia de especialização ou generalização também são necessárias.

No nível de objeto, os valores dos atributos-chave são usados para distinguir entre entidades de um tipo em particular. Para tipos de entidade fraca, as entidades são identificadas por uma combinação de valores próprios de chave parcial e aquelas às quais estão relacionadas, dependendo da razão de cardinalidade especificada.

## 8.7.3 Especialização e generalização

**Especialização** é o processo de classificar uma classe de objetos em subclasses mais especializadas. **Generalização** é o processo inverso de generalizar várias classes em uma classe abstrata de mais alto nível, que inclua os objetos em todas essas classes. A especialização é o refinamento conceitual, enquanto a generalização é a síntese conceitual. Subclasses são usadas no modelo EER para representar a especialização e a generalização. Chamamos o relacionamento entre uma subclasse e suas superclasses de relacionamento **É\_UMA\_SUBCLASSE\_DE**, ou simplesmente um relacionamento **É\_UM**. Trata-se do mesmo relacionamento **É\_UM** discutido anteriormente, na Seção 8.5.3.

## 8.7.4 Agregação e associação

**Agregação** é um conceito de abstração para a criação de objetos compostos com base em seus objetos componentes. Existem três casos em que esse conceito pode estar relacionado ao modelo EER. O primeiro caso é a situação em que agregamos valores de atributo de um objeto para formar o objeto total. O segundo caso é quando representamos um relacionamento de agregação como um relacionamento comum. O terceiro caso, que o modelo EER não determina explicitamente, envolve a possibilidade de combinar objetos que são relacionados por uma instância de relacionamento em particular a um *objeto agregado de nível superior*. Isso às vezes é útil quando o próprio objeto de agregação de nível mais alto tem de estar relacionado a outro objeto. Chamamos o relacionamento entre os objetos primitivos e seu objeto de agregação **É\_UMA\_PARTE\_DE**; o inverso é chamado de **É\_UM\_COMPONENTE\_DE**. A UML possibilita todos os três tipos de agregação.

A abstração da **associação** é usada para associar objetos de várias *classes independentes*. Assim, às vezes ela é semelhante ao segundo uso da agregação, representada no modelo EER por tipos de relacionamento, e em UML, por associações. Esse relacionamento abstrato é chamado de **É\_ASSOCIADO\_A**.

Para entender melhor os diferentes usos da agregação, considere o esquema ER mostrado na Figura 8.11(a), que armazena informações sobre entrevistas por candidatos a emprego para várias empresas. A classe EMPRESA é uma agregação dos atributos (ou objetos componentes) Enome (nome de empresa) e Eendereco (endereço da empresa), enquanto CANDIDATO é uma agregação de Cpf, Nome, Eendereço e Telefone. Os atributos de relacionamento Nome\_resp e Telefone\_resp representam o nome e o número de telefone da pessoa na empresa que é responsável pela entrevista. Suponha que algumas entrevistas resultem em ofertas de emprego, ao passo que outras não. Gostaríamos de tratar ENTREVISTA como uma classe para associá-la a OFERTA\_EMPREGO. O esquema mostrado na Figura 8.11(b) está *incorreto* porque requer que cada instância de relacionamento de entrevista tenha uma oferta de emprego. O esquema mostrado na Figura 8.11(c) *não é permitido* porque o modelo ER não permite relacionamentos entre relacionamentos.

Uma forma de representar essa situação é criar uma classe agregada de nível mais alto, composta de EMPRESA, CANDIDATO e ENTREVISTA, e relacioná-la a OFERTA\_EMPREGO, como mostra a Figura 8.11(d). Embora o modelo EER, conforme descrito neste livro, não tenha essa facilidade, alguns modelos de dados semânticos o permitem, e chamam o objeto resultante de **objeto composto** ou **molecular**. Outros modelos tratam tipos de entidade e tipos de relacionamento de maneira uniforme e, portanto, permitem relacionamentos entre relacionamentos, conforme ilustrado pela Figura 8.11(c).

Para representar essa situação corretamente no modelo ER descrito aqui, precisamos criar um novo tipo de entidade fraca ENTREVISTA, como mostra a Figura 8.11(e), e relacioná-lo a OFERTA\_EMPREGO. Logo, sempre podemos representar essas situações de modo correto no modelo ER criando tipos de entidade adicionais, embora possa ser conceitualmente mais desejável permitir a representação direta da agregação, como na Figura 8.11(d), ou permitir relacionamentos entre relacionamentos, como na Figura 8.11(c).

A distinção estrutural principal entre agregação e associação é que, quando uma instância de associação é excluída, os objetos participantes podem continuar a existir. Porém, se dermos suporte à noção de um objeto de agregação — por exemplo, um CARRO que é composto dos objetos MOTOR, CHASSI e PNEUS —, então a exclusão do objeto de agregação CARRO corresponde a excluir todos os seus objetos componentes.

## 8.7.5 Ontologias e a Web semântica

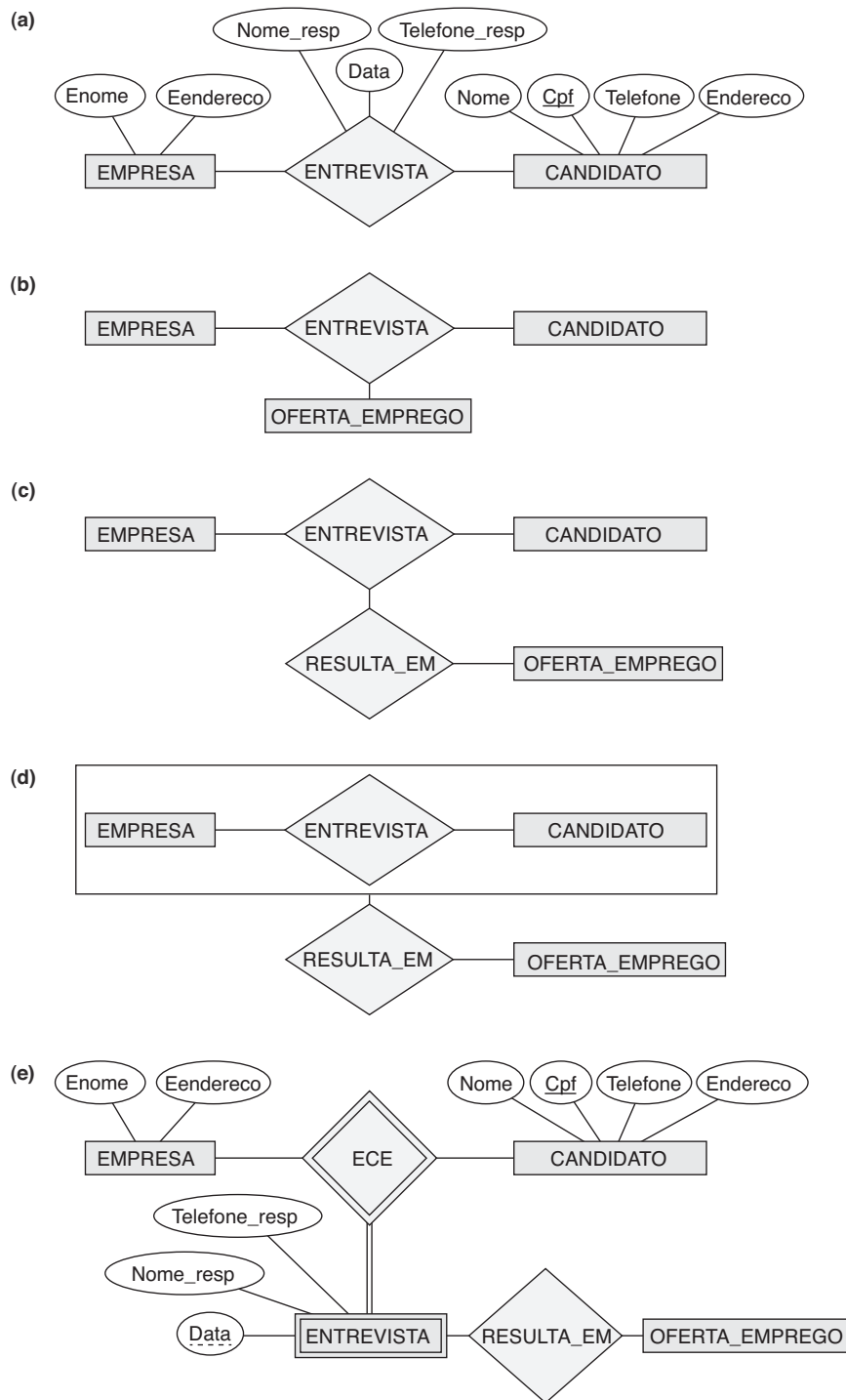
Nos últimos anos, a quantidade de dados e informações computadorizadas disponíveis na Web se tornou algo fora de controle. Muitos modelos e formatos diferentes são utilizados. Além dos modelos de banco de dados que apresentamos neste livro, muita informação é armazenada na forma de **documentos**, que possuem consideravelmente muito menos estrutura do que a informação do banco de dados. Um projeto em andamento, que está tentando permitir a troca de informações entre computadores na Web, é chamado de **Web semântica**, que tenta criar modelos de representação do conhecimento que sejam bastante genéricos, a fim de permitir a troca e a pesquisa de informações significativas entre máquinas. O conceito de *ontologia* é considerado a base mais promissora para alcançar os objetivos da Web semântica e está bastante relacionado à representação do conhecimento. Nesta seção, oferecemos uma rápida introdução ao que é a ontologia e como ela pode ser usada como uma base para automatizar o conhecimento, a busca e troca de informações.

O estudo das ontologias tenta descrever as estruturas e os relacionamentos possíveis na realidade por meio de um vocabulário comum. Portanto, ele pode ser considerado um meio para descrever o conhecimento de certa comunidade sobre a realidade. A ontologia originou-se nas áreas de filosofia e metafísica. Uma definição muito usada de **ontologia** é *uma especificação de uma conceitualização*.<sup>13</sup>

Nessa definição, uma **conceitualização** é o conjunto de conceitos usados para representar a parte da realidade ou conhecimento que é de interesse de uma comunidade de usuários. **Especificação** refere-se à linguagem e termos do vocabulário usados para especificar a conceitualização. A ontologia inclui tanto a *especificação* quanto a *conceitualização*. Por exemplo, a mesma conceitualização pode ser especificada em duas linguagens diferentes, gerando duas ontologias separadas. Com base nessa definição bastante geral, não existe consenso sobre o que é exatamente uma ontologia. Algumas maneiras possíveis de descrever as ontologias são as seguintes:

- Um **thesaurus** (ou ainda um **dicionário** ou um **glossário** de termos) descreve os relacionamentos entre palavras (vocabulário) que representam diversos conceitos.
- Uma **taxonomia** descreve como os conceitos de determinada área do conhecimento são relacionados usando estruturas semelhantes àsquelas utilizadas em uma especialização ou generalização.

<sup>13</sup> Essa definição é dada em Gruber (1995).

**Figura 8.11**

Agregação. (a) O tipo de relacionamento **ENTREVISTA**. (b) Incluindo **OFERTA\_EMPREGO** em um tipo de relacionamento ternário (incorreto). (c) Fazendo o relacionamento **RESULTA\_EM** participar de outros relacionamentos (não permitido em ER). (d) Usando agregação e um objeto composto (molecular) (geralmente, não permitido em ER, mas permitido por algumas ferramentas de modelagem). (e) Representação correta em ER.

■ Um **esquema de banco de dados** detalhado é considerado por alguns uma ontologia que descreve os conceitos (entidades e atributos) e relacionamentos de um minimundo da realidade.

■ Uma **teoria lógica** usa conceitos da lógica matemática para tentar definir conceitos e seus inter-relacionamentos.



Normalmente, os conceitos usados para descrever ontologias são muito semelhantes aos conceitos que discutimos na modelagem conceitual, como entidades, atributos, relacionamentos, especializações, e assim por diante. A principal diferença entre uma ontologia e, digamos, um esquema de banco de dados é que o esquema em geral está limitado a descrever um pequeno subconjunto de um minimundo da realidade a fim de armazenar e gerenciar dados. Uma ontologia costuma ser considerada mais geral no sentido de tentar descrever uma parte da realidade ou de um domínio de interesse (por exemplo, termos médicos, aplicações de comércio eletrônico, esportes etc.) o mais completamente possível.

## Resumo

Neste capítulo, discutimos as extensões ao modelo ER que melhoram suas capacidades de representação. Chamamos o modelo resultante de modelo ER estendido ou EER. Apresentamos o conceito de uma subclasse e sua superclasse e o mecanismo relacionado de herança de atributo/relacionamento. Vimos como às vezes é necessário criar classes de entidades adicionais, seja por causa dos atributos específicos adicionais ou tipos de relacionamento específicos. Discutimos dois processos principais para definir hierarquias e reticulados de superclasse/subclasse: especialização e generalização.

Em seguida, mostramos como apresentar essas novas construções em um diagrama EER. Também discutimos os diversos tipos de restrições que podem se aplicar à especialização ou generalização. As duas principais restrições são total/parcial e disjunta/sobreposta. Além disso, um predicado de definição para uma subclasse ou um atributo de definição para uma especialização podem ser especificados. Abordamos as diferenças entre subclasses definidas pelo usuário, por predicado e entre especializações definidas pelo usuário e por atributo. Por fim, discutimos o conceito de uma categoria ou tipo de união, que é um subconjunto da união de duas ou mais classes, e mostramos as definições formais de todos os conceitos apresentados.

Apresentamos parte da notação e terminologia da UML para representar a especialização e a generalização. Na Seção 8.7, discutimos rapidamente a disciplina de representação do conhecimento (RC) e como ela está relacionada à modelagem de dados semântica. Também demos uma visão geral e um resumo dos tipos de conceitos abstratos da representação de dados: classificação e instanciação, identificação, especialização e generalização, e agregação e associação. Vimos como os conceitos de EER e UML estão relacionados a cada um deles.

## Perguntas de revisão

- 8.1. O que é uma subclasse? Quando uma subclasse é necessária na modelagem de dados?

- 8.2. Defina os seguintes termos: *superclasse de uma subclasse*, *relacionamento de superclasse/subclasse*, *relacionamento É-UM*, *especialização*, *generalização*, *categoria*, *atributos específicos (locais)* e *relacionamentos específicos*.
- 8.3. Discuta o mecanismo de herança de atributo/relacionamento. De que forma ele é útil?
- 8.4. Discuta as subclasses definidas pelo usuário e definidas por predicado, e identifique as diferenças entre as duas.
- 8.5. Discuta as especializações definidas pelo usuário e definidas por atributo, e identifique as diferenças entre as duas.
- 8.6. Discuta os dois tipos principais de restrições sobre especializações e generalizações.
- 8.7. Qual é a diferença entre uma hierarquia de especialização e um reticulado de especialização?
- 8.8. Qual é a diferença entre especialização e generalização? Por que não exibimos essa diferença nos diagramas de esquema?
- 8.9. Como uma categoria difere de uma subclasse compartilhada regular? Para que uma categoria é usada? Ilustre sua resposta com exemplos.
- 8.10. Para cada um dos seguintes termos da UML (ver seções 7.8 e 8.6), discuta o termo correspondente no modelo EER, se houver: *objeto*, *classe*, *associação*, *agregação*, *generalização*, *multiplicidade*, *atributos*, *discriminador*, *ligação*, *atributo de ligação*, *associação reflexiva* e *associação qualificada*.
- 8.11. Discuta as principais diferenças entre a notação para diagramas de esquema EER e diagramas de classe UML comparando como os conceitos comuns são representados em cada um.
- 8.12. Liste os diversos conceitos de abstração de dados e os conceitos de modelagem correspondentes no modelo EER.
- 8.13. Que recurso de agregação está faltando do modelo EER? Como o modelo EER pode ser melhorado para dar suporte a esse recurso?
- 8.14. Quais são as principais semelhanças e diferenças entre as técnicas conceituais de modelagem de banco de dados e as técnicas de representação do conhecimento?
- 8.15. Discuta as semelhanças e diferenças entre uma ontologia e um esquema de banco de dados.

## Exercícios

- 8.16. Projete um esquema EER para uma aplicação de banco de dados em que você está interessado. Especifique todas as restrições que devem ser mantidas no banco de dados. Cuide para que o esquema tenha pelo menos cinco tipos de entidade, quatro tipos de relacionamento, um tipo de entidade fraca, um relacionamento de superclas-

se/subclasse, uma categoria e um tipo de relacionamento  $n$ -ário ( $n > 2$ ).

- 8.17. Considere o esquema ER BANCO da Figura 7.21 e suponha que seja necessário registrar diferentes tipos de CONTAS (CONTA\_POU-PANCA, CONTA\_CORRENTE, ...) e EMPRES-TIMOS (EMPRESTIMO\_CARRO, EMPRESTI-MO\_HABITACAO, ...). Suponha que também se deseje registrar cada uma das TRANSACOES de CONTA (depósitos, saques, cheques, ...) e os PAGAMENTOS de EMPRESTIMO; ambos incluem o valor, data e hora. Modifique o esquema BANCO, usando os conceitos de ER e EER de especialização e generalização. Indique quaisquer suposições que você fizer sobre os requisitos adicionais.
- 8.18. A narrativa a seguir descreve uma versão simplificada da organização das instalações olímpicas planejadas para os Jogos Olímpicos de verão. Desenhe um diagrama EER que mostre os tipos de entidade, atributos, relacionamentos e especializações para essa aplicação. Indique quaisquer suposições que você fizer. As instalações olímpicas são divididas em complexos esportivos. Os complexos esportivos são divididos em tipos de *um esporte* e *poliesportivo*. Os complexos poliesportivos possuem áreas designadas para cada esporte com um indicador de localização (por exemplo, centro, canto NE, e assim por diante). Um complexo tem um local, organizador chefe, área total ocupada, e assim por diante. Cada complexo mantém uma série de eventos (por exemplo, o estádio com raias pode englobar muitas corridas diferentes). Para cada evento, existe uma data planejada, duração, número de participantes, número de oficiais e assim por diante. Uma relação de todos os oficiais será mantida junto com a lista dos eventos em que cada oficial estará envolvido. Diferentes equipamentos são necessários para os eventos (por exemplo, balizas, postes, barras paralelas), bem como para a manutenção. Os dois tipos de instalações (um esporte e poliesportivo) terão diferentes tipos de informação. Para cada tipo, o número de instalações necessárias é mantido, junto com um orçamento aproximado.
- 8.19. Identifique todos os conceitos importantes representados no estudo de caso do banco de dados de biblioteca descrito a seguir. Em particular, identifique as abstrações de classificação (tipos de entidade e tipos de relacionamento), agregação, identificação e especialização/generalização. Especifique as restrições de cardinalidade (min, max) sempre que possível. Liste detalhes que afetarão o eventual projeto, mas que não têm significado sobre o projeto conceitual. Liste as restrições semânticas separadamente.

Desenhe um diagrama EER do banco de dados de biblioteca.

**Estudo de caso:** A Georgia Tech Library (GTL) tem aproximadamente 16 mil usuários, 100 mil títulos e 250 mil volumes (uma média de 2,5 cópias por livro). Cerca de 10 por cento dos volumes estão emprestados a qualquer momento. Os bibliotecários garantem que os livros estejam disponíveis quando os usuários querem apanhá-los emprestado. Além disso, os bibliotecários precisam saber quantas cópias de cada livro existem na biblioteca ou emprestadas a qualquer momento. Um catálogo de livros está disponível on-line, listando livros por autor, título e assunto. Para cada título da biblioteca, uma descrição do livro é mantida no catálogo, que varia de parágrafos a várias páginas. Os bibliotecários de referência desejam poder acessar essa descrição quando os usuários solicitarem informações sobre um livro. O pessoal da biblioteca inclui o bibliotecário chefe, bibliotecários associados ao departamento, bibliotecários de referência, pessoal de despacho e assistentes de bibliotecário.

Os livros podem ser emprestados por 21 dias. Os usuários têm permissão para pegar apenas cinco livros de uma só vez. Os usuários normalmente retornam os livros dentro de três a quatro semanas. A maioria dos usuários sabe que têm uma semana de tolerância antes que uma nota seja enviada para eles, e por isso tentam retornar os livros antes que o período de tolerância termine. Cerca de 5 por cento dos usuários precisa receber lembretes para devolver os livros. A maioria dos livros atrasados é retornada dentro de um mês da data de vencimento. Aproximadamente 5 por cento dos livros atrasados são mantidos ou nunca são retornados. Os membros mais ativos da biblioteca são definidos como aqueles que pegam livros emprestados pelo menos dez vezes durante o ano. Um por cento dos usuários que mais utilizam empréstimos realizam 15 por cento dos empréstimos, e os maiores 10 por cento dos usuários realizam 40 por cento dos empréstimos. Cerca de 20 por cento dos usuários são totalmente inativos por nunca terem emprestado livros.

Para tornar-se um usuário da biblioteca, os candidatos preenchem um formulário incluindo seu CPF, endereço de correspondência do campus e da residência, e números de telefone. Os bibliotecários emitem um cartão numerado, legível à máquina, com a foto do usuário. Esse cartão tem validade de quatro anos. Um mês antes de um cartão expirar, uma nota é enviada ao usuário para fazer a renovação. Os professores do instituto são considerados usuários automáticos. Quando um novo usuário do corpo docente entra para o instituto, sua informação é puxada

dos registros de funcionário e um cartão da biblioteca é remetido ao seu endereço no campus. Os professores têm permissão para retirar livros por intervalos de três meses, e possuem um período de tolerância de duas semanas. As notas de renovação para os professores são enviadas para seu endereço no campus.

A biblioteca não empresta alguns livros, como livros de referência, livros raros e mapas. Os bibliotecários precisam diferenciar livros que podem ser emprestados daqueles que não podem. Além disso, eles possuem uma lista de alguns livros em que estão interessados em adquirir, mas não conseguem obter, como livros raros ou que estão esgotados, e livros que foram perdidos ou destruídos, mas não substituídos. Os bibliotecários precisam ter um sistema que registre os livros que não podem ser emprestados bem como os livros que eles estão interessados em adquirir. Alguns livros podem ter o mesmo título; portanto, o título não pode ser usado como um meio de identificação. Cada livro é identificado por seu International Standard Book Number (ISBN), um código internacional exclusivo atribuído a todos os livros. Dois livros com o mesmo título podem ter diferentes ISBNs se estiverem em diferentes idiomas ou tiverem diferentes encadernações (capa dura ou brochura). As edições de um mesmo livro possuem ISBNs diferentes.

O sistema de banco de dados proposto precisa ser projetado para registrar os usuários, os livros, o catálogo e a atividade de empréstimo.

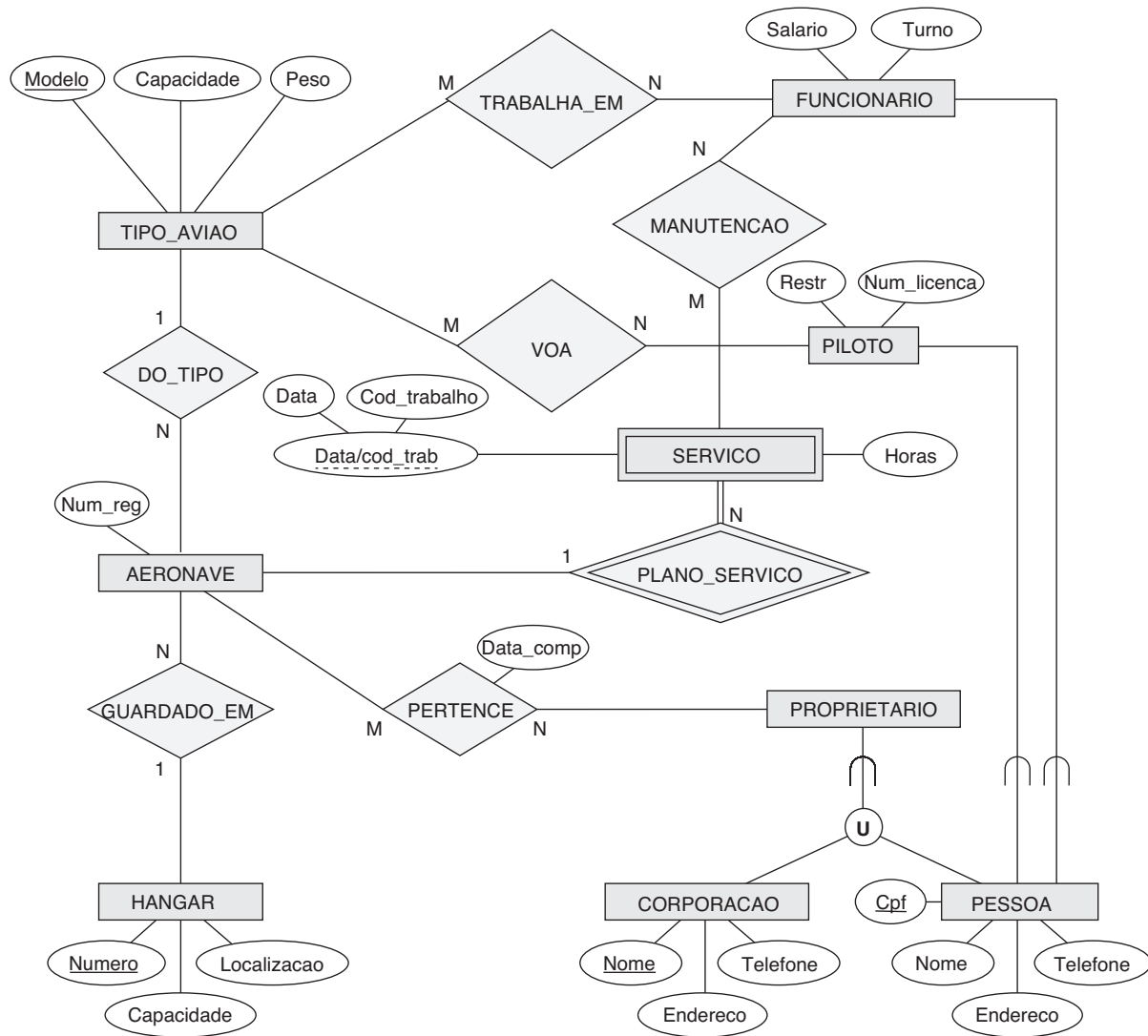
**8.20.** Projete um banco de dados para registrar informações para um museu de arte. Suponha que os seguintes requisitos foram coletados:

- O museu tem uma coleção de OBJETOS\_ARTE. Cada OBJETO\_ARTE tem um número\_id exclusivo, um Artista (se conhecido), um Ano (quando foi criado, se conhecido), um Título e uma Descrição. Os objetos de arte são categorizados de várias maneiras, conforme discutido a seguir.
- OBJETOS\_ARTE são categorizados com base em seu tipo. Existem três tipos principais: PINTURA, ESCULTURA e ESTATUA, mais um tipo chamado OUTRO para acomodar objetos que não se encaixam em nenhum dos três tipos principais.
- Uma PINTURA tem um Tipo\_pintura (óleo, aquarela etc.), material em que é desenhada Desenhado\_em (papel, tela, madeira etc.) e Estilo (moderno, abstrato etc.).
- Uma ESCULTURA ou uma estátua tem um Material com a qual foi criada (madeira, pedra etc.), Altura, Peso e Estilo.
- Um objeto de arte na categoria OUTRO tem um Tipo (impressão, foto etc.) e Estilo.

- OBJETOS\_ARTE são categorizados como COLECAO\_PERMANENTE (objetos que pertencem ao museu) e EMPRESTADOS. As informações capturadas sobre os objetos na COLECAO\_PERMANENTE incluem Data\_aquisicao, Status (em exibição, emprestado ou guardado) e Custo. A informação capturada sobre objetos EMPRESTADOS inclui a Colecao da qual foi emprestado, Data\_emprestimo e Data\_retorno.
- A informação descrevendo o país ou cultura da Origem (italiano, egípcio, norte-americano, indiano etc.) e Epoca (Renascença, Moderno, Antiguidade, e assim por diante) é capturada para cada OBJETO\_ARTE.
- O museu registra a informação de ARTISTA, se for conhecida: Nome, Data\_nascimento (se conhecida), Data\_morte (se não estiver vivo), Pais\_de\_origem, Epoca, Estilo\_principal e Descricao. O Nome é considerado exclusivo.
- Ocorrem diferentes EXPOSICOES, cada uma com um Nome, Data\_inicio e Data\_final. As EXPOSICOES são relacionadas a todos os objetos de arte que estavam em amostra durante a exposição.
- A informação é mantida em outras COLECOES com as quais o museu interage, incluindo Nome (exclusivo), Tipo (museu, pessoal etc.), Descricao, Endereco, Telefone e Pessoa\_contato\_atual.

Desenhe um diagrama de esquema EER para essa aplicação. Discuta quaisquer suposições que você fizer e que justifiquem suas escolhas de projeto EER.

**8.21.** A Figura 8.12 mostra um exemplo de diagrama EER para o banco de dados de um pequeno aeroporto particular, que é usado para registrar aeronaves, seus proprietários, funcionários do aeroporto e pilotos. Com base nos requisitos para esse banco de dados, a informação a seguir foi coletada: cada AERONAVE tem um número de registro [Num\_reg], é de um tipo de avião em particular [DO\_TIPO] e é mantido em um hangar em particular [GUARDADO\_EM]. Cada TIPO\_AVIAO tem um número de modelo [Modelo], uma capacidade [Capacidade] e um peso [Peso]. Cada HANGAR tem um número [Numero], uma capacidade [Capacidade] e um local [Localizacao]. O banco de dados também registra os PROPRIETARIOS de cada avião [PERTENCE] e os FUNCIONARIOS que fazem a manutenção do avião [MANUTENCAO]. Cada instância de relacionamento em PERTENCE relaciona uma AERONAVE a um PROPRIETARIO e inclui a data de compra [Data\_comp]. Cada instância de relacionamento em MANUTENCAO relaciona um FUNCIONARIO a um registro de serviço [SERVICO].

**Figura 8.12**

Esquema EER para um banco de dados PEQUENO\_AEROPORTO.

Cada avião passa por manutenção muitas vezes; logo, ela é relacionada por [PLANO\_SERVICO] a uma série de registros de SERVICO. Um registro de SERVICO inclui como atributos a data da manutenção [Data], o número de horas gastas no trabalho [Horas] e o tipo de trabalho realizado [Cod\_trab]. Usamos um tipo de entidade fraca [SERVICO] para representar o serviço na aeronave, pois o número de registro da aeronave é usado para identificar um registro de manutenção. Um PROPRIETARIO é uma pessoa ou uma corporação. Assim, usamos um tipo de união (categoria) [PROPRIETARIO] que é um subconjunto da união dos tipos de entidade corporação [CORPORACAO] e pessoa [PESSOA]. Tanto pilotos [PILOTO] quanto funcionários [FUNCIONARIO] são subclasses de PESSOA. Cada PILOTO tem atributos específicos de número de licença [Num\_licenca] e restrições [Restricoes];

cada FUNCIONARIO tem atributos específicos de salário [Salario] e turno trabalhado [Turno]. Todas as entidades PESSOA no banco de dados possuem dados mantidos sobre seu número de Cadastro de Pessoa Física [Cpf], nome [Nome], endereço [Endereco] e número de telefone [Telefone]. Para entidades CORPORACAO, os dados mantidos incluem nome [Nome], endereço [Endereco] e número de telefone [Telefone]. O banco de dados também registra os tipos de aviões que cada piloto é autorizado a voar [VOA] e os tipos de aviões em que cada funcionário pode realizar o trabalho de manutenção [TRABALHA\_EM]. Mostre como o esquema EER PEQUENO\_AEROPORTO da Figura 8.12 pode ser representado em notação UML. (Nota: não discutimos como representar categorias (tipos de união) em UML, de modo que você não precisa mapear as categorias nesta e na próxima questão.)



	Conjunto de entidades	(a) Tem relacionamento com	(b) Tem um atributo que é	(c) É uma especialização de	(d) É uma generalização de	Conjunto de entidades ou atributo
1.	MÃE					PESSOA
2.	FILHA					MÃE
3.	ALUNO					PESSOA
4.	ALUNO					Cod_aluno
5.	ESCOLA					ALUNO
6.	ESCOLA					SALA_AULA
7.	ANIMAL					CAVALO
8.	CAVALO					Raça
9.	CAVALO					Idade
10.	FUNCIONÁRIO					CPF
11.	MÓVEL					CADEIRA
12.	CADEIRA					Peso
13.	HUMANO					MULHER
14.	SOLDADO					PESSOA
15.	COMBATENTE_INIMIGO					PESSOA

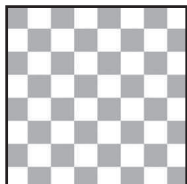
**8.22.** Mostre como o esquema EER UNIVERSIDADE da Figura 8.9 pode ser representado em notação UML.

**8.23.** Considere os conjuntos de entidades e atributos mostrados na tabela desta página. Coloque uma marcação em uma coluna de cada linha, para indicar o relacionamento entre as colunas mais à esquerda e à direita.

- a. O lado esquerdo tem um relacionamento com o lado direito.
- b. O lado direito é um atributo do lado esquerdo.
- c. O lado esquerdo é uma especialização do lado direito.
- d. O lado esquerdo é uma generalização do lado direito.

**8.24.** Desenhe um diagrama UML para armazenar um jogo de xadrez em um banco de dados. Você pode examinar em <<http://www.chessgames.com>> como fazer uma aplicação semelhante à que você está projetando. Indique claramente quaisquer suposições que você fizer em seu diagrama UML. Uma amostra das suposições que você pode fazer sobre o escopo é a seguinte:

- 1. O jogo de xadrez é realizado por dois jogadores.
- 2. O jogo é realizado em um tabuleiro de  $8 \times 8$ , como o que aparece a seguir:



3. Os jogadores recebem uma cor preta ou branca no início do jogo.

4. Cada jogador começa com as seguintes peças:

- a. rei
- b. rainha
- c. 2 torres
- d. 2 bispos
- e. 2 cavalos
- f. 8 peões

5. Cada peça tem a própria posição inicial.

6. Cada peça tem o próprio conjunto de jogadas válidas com base no estado do jogo. Você não precisa se preocupar com quais jogadas são válidas ou não, exceto pelas seguintes questões:

- a. Uma peça pode se mover para um quadrado vazio ou capturar uma peça do oponente.
- b. Se uma peça for capturada, ela é removida do tabuleiro.
- c. Se um peão se mover para a última fileira, ele é 'promovido', sendo convertido para outra peça (rainha, torre, bispo ou cavalo).

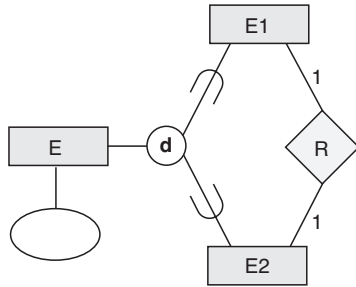
*Nota:* algumas dessas funções podem se espalhar por várias classes.

**8.25.** Desenhe um diagrama EER para um jogo de xadrez conforme descrito no Exercício 8.24. Focalize nos aspectos de armazenamento persistente do sistema. Por exemplo, o sistema precisaria recuperar todas as jogadas de cada jogo realizado em ordem sequencial.

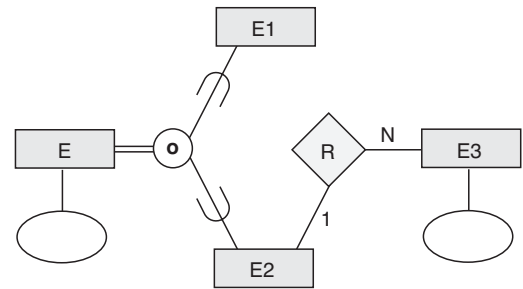
**8.26.** Quais dos seguintes diagramas EER são incorretos e por quê? Indique claramente quaisquer suposições que você fizer.

8.27. Considere o seguinte diagrama EER que descreve os sistemas de computador em uma empresa. Forneça os próprios atributos e chave para cada tipo de entidade. Forneça restrições de cardinalidade max justificando sua escolha. Escreva uma descrição narrativa completa do que esse diagrama EER representa.

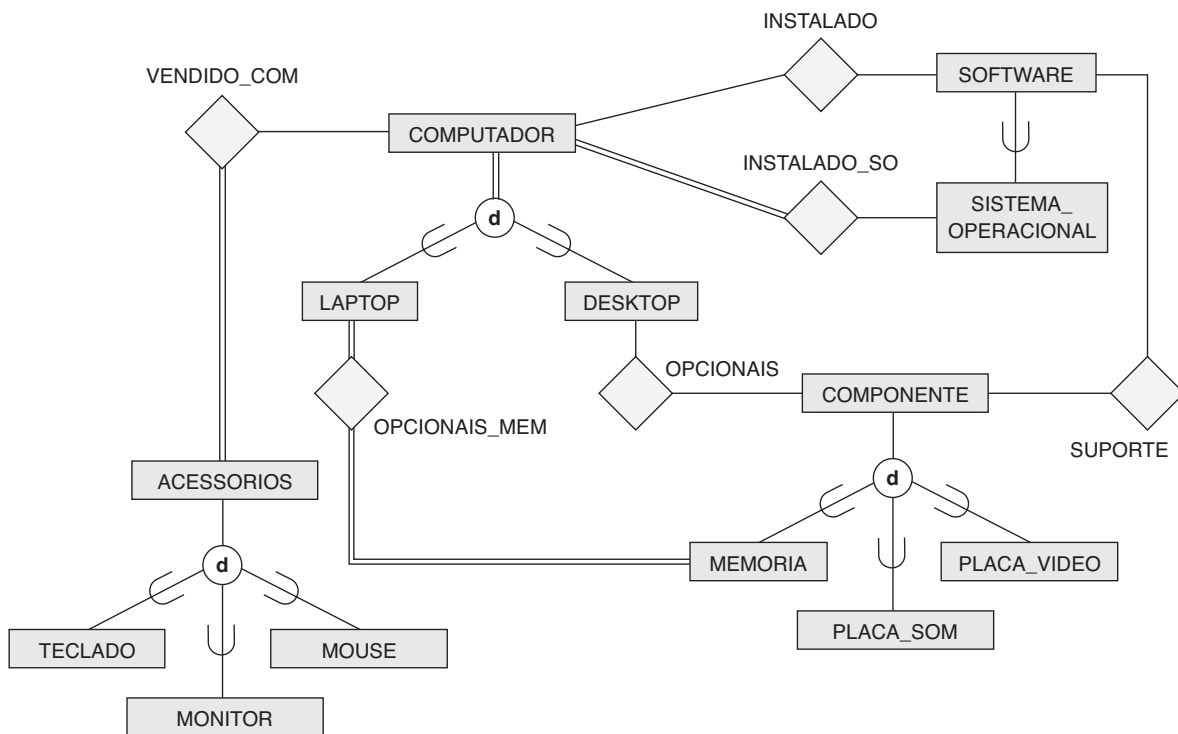
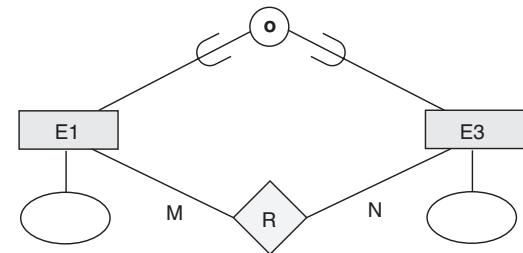
b.



a.



c.



## Exercícios de laboratório

8.28. Considere um banco de dados DIARIO\_NOTAS em que os professores de um departamento acadêmico registram pontos ganhos por alunos individuais em suas aulas. Os requisitos de dados são resumidos da seguinte forma:

- Cada aluno é identificado por um identificador exclusivo, nome e sobrenome, e por um endereço de e-mail.

- Cada professor leciona certas disciplinas a cada período. Cada disciplina é identificada por um número, um número de seção e o período em que ela é realizada. Para cada disciplina, o professor especifica o número mínimo de pontos necessários para ganhar notas A, B, C, D e F. Por exemplo, 90 pontos para um A, 80 pontos para um B, 70 pontos para um C, e assim por diante.

- Os alunos são matriculados em cada disciplina lecionada pelo professor.
- Cada disciplina tem uma série de componentes de avaliação (como exame do meio do período, exame final, projeto, e assim por diante). Cada componente de avaliação tem um número máximo de pontos (como 100 ou 50) e um peso (como 20 por cento ou 10 por cento). Os pesos de todos os componentes de avaliação de um curso em geral totalizam 100.
- Finalmente, o professor registra os pontos ganhos por aluno em cada um dos componentes de avaliação em cada uma das disciplinas. Por exemplo, o aluno 1234 ganha 84 pontos para o componente de avaliação do meio do período da disciplina CCc2310 da seção 2 no período do segundo semestre de 2009. O componente de avaliação de exame do meio do período pode ter sido definido para ter um máximo de 100 pontos e um peso de 20 por cento da nota da disciplina.

Crie um diagrama Entidade-Relacionamento estendido para o banco de dados do diário e monte o projeto usando uma ferramenta de modelagem como ERwin ou Rational Rose.

- 8.29.** Considere um sistema de banco de dados LEILAO\_ON-LINE em que os membros (compradores e vendedores) participam na venda de itens. Os requisitos de dados para esse sistema são resumidos a seguir:
- O site on-line tem membros, e cada um é identificado por um número de membro exclusivo e descrito por um endereço de e-mail, nome, senha, endereço residencial e número de telefone.
  - Um membro pode ser um comprador ou um vendedor. Um comprador tem um endereço de entrega registrado no banco de dados. Um vendedor tem um número de conta bancária e um número de encaminhamento registrados no banco de dados.
  - Os itens são colocados à venda por um vendedor e identificados por um número de item exclusivo, atribuído pelo sistema. Os itens também são descritos por um título de item, uma descrição, um preço de lance inicial, um incremento de lance, a data inicial do leilão e a data final do leilão.
  - Os itens também são classificados com base em uma hierarquia de classificação fixa (por exemplo, um modem pode ser classificado como COMPUTADOR→HARDWARE→MODEM).
  - Os compradores fazem lances para os itens em que estão interessados. O preço do lance e a hora do lance são registrados. O comprador ao final do leilão com o maior preço de

lance é declarado o vencedor e uma transação entre comprador e vendedor pode então prosseguir.

- O comprador e o vendedor podem registrar uma nota em relação às transações completadas. A nota contém uma pontuação da outra parte na transação (1-10) e um comentário.

Crie um diagrama Entidade-Relacionamento estendido para o banco de dados LEILAO\_ON-LINE e monte o projeto usando uma ferramenta de modelagem como ERwin ou Rational Rose.

- 8.30.** Considere um sistema de banco de dados para uma organização de beisebol como as principais ligas nacionais. Os requisitos de dados são resumidos a seguir:

- O pessoal envolvido na liga inclui jogadores, técnicos, dirigentes e árbitros. Cada um tem uma identificação pessoal exclusiva. Eles também são descritos por seu nome e sobrenome, junto com a data e local de nascimento.
- Os jogadores são descritos ainda por outros atributos, como sua orientação de batida (esquerda, direita ou ambas) e têm uma média de batidas (MB) por toda a vida.
- Dentro do grupo de jogadores existe um subgrupo de jogadores chamados lançadores. Os lançadores têm uma média de corrida ganha (MCG) por toda a vida associada a eles.
- As equipes são identificadas exclusivamente por seus nomes. As equipes também são descritas pela cidade em que estão localizadas e pela divisão e liga em que jogam (como a divisão Central da Liga Norte-americana).
- As equipes possuem um dirigente, uma série de técnicos e uma série de jogadores.
- Os jogos são realizados entre dois times com um designado como o time da casa e o outro como o time visitante em determinada data. A pontuação (corridas, batidas e erros) é registrado para cada time. O time com a maioria das corridas é declarado o vencedor do jogo.
- A cada jogo terminado, um lançador vencedor e um lançador perdedor são registrados. Caso seja concedido um salvamento, o lançador salvo também é registrado.
- A cada jogo terminado, o número de acertos (simples, duplos, triplos e *home runs*) obtidos por jogador também é registrado.

Crie um diagrama Entidade-Relacionamento estendido para o banco de dados BEISEBOL e monte o projeto usando uma ferramenta de modelagem como ERwin ou Rational Rose.

- 8.31.** Considere o diagrama EER para o banco de dados UNIVERSIDADE mostrado na Figura 8.9.

Entre com seu projeto usando uma ferramenta de modelagem de dados como ERwin ou Rational Rose. Faça uma lista das diferenças na notação entre o diagrama no texto e a notação diagramática equivalente que você acabou usando com a ferramenta.

- 8.32. Considere o diagrama EER para o pequeno banco de dados AEROPORTO mostrado na Figura 8.12. Monte esse projeto usando uma ferramenta de modelagem de dados como ERwin ou Rational Rose. Tenha cuidado ao modelar a categoria PROPRIETARIO nesse diagrama. (*Dica:* considere o uso de CORPORACAO\_É\_PROPRIETARIA e PESSOA\_É\_PROPRIETARIA como tipos de relacionamento distintos.)
- 8.33. Considere o banco de dados UNIVERSIDADE descrito no Exercício 7.16. Você já desenvolveu um esquema ER para esse banco de dados usando uma ferramenta de modelagem de dados como ERwin ou Rational Rose no Exercício de Laboratório 7.31. Modifique esse diagrama classificando DISCIPLINAS como DISCIPLINA\_GRADUACAO ou DISCIPLINA\_POSGRADUACAO e PROFESSORES como PROFESSORES\_JUNIOR ou PROFESSORES\_SENIOR. Inclua atributos apropriados para esses novos tipos de entidade. Depois, estabeleça relacionamentos indicando que os professores júnior lecionam disciplinas para alunos em graduação enquanto os professores sênior lecionam disciplinas para alunos de pós-graduação.

## Bibliografia selecionada

Muitos artigos propuseram modelos de dados conceituais ou semânticos. Aqui, oferecemos uma lista representativa. Um grupo de artigos, incluindo Abrial

(1974), modelo DIAM de Senko (1975), o método NIAM (Verheijen e VanBekkum, 1982) e Bracchi et al. (1976), apresenta modelos semânticos que são baseados no conceito de relacionamentos binários. Outro grupo de artigos antigos discute métodos para estender o modelo relacional para melhorar suas capacidades de modelagem. Isso inclui os artigos de Schmid e Swenson (1975), Navathe e Schkolnick (1978), modelo RM/T de Codd (1979), Furtado (1978) e o modelo estrutural de Wiederhold e Elmasri (1979).

O modelo ER foi proposto originalmente por Chen (1976) e é formalizado em Ng (1981). Desde então, diversas extensões de suas capacidades de modelagem foram propostas, como em Scheuermann et al. (1979), Dos Santos et al. (1979), Teorey et al. (1986), Gogolla e Hohenstein (1991) e o modelo Entidade-Categoria-Relacionamento (ECR) de Elmasri et al. (1985). Smith e Smith (1977) apresentam os conceitos de generalização e agregação. O modelo de dados semântico de Hammer e McLeod (1981) introduziu os conceitos de reticulados de classe/subclasse, bem como outros conceitos de modelagem avançados.

Um estudo da modelagem semântica de dados aparece em Hull e King (1987). Eick (1991) discute projeto e transformações dos esquemas conceituais. A análise de restrições para relacionamentos  $n$ -ários é dada em Soutou (1998). A UML é descrita detalhadamente em Booch, Rumbaugh e Jacobson (1999). Fowler e Scott (2000) e Stevens e Pooley (2000) oferecem introduções concisas aos conceitos de UML.

Fensel (2000, 2003) discute a Web semântica e a aplicação de ontologias. Uschold e Gruninger (1996) e Gruber (1995) discutem sobre ontologias. A edição de junho de 2002 de *Communications of the ACM* é dedicada a conceitos e aplicações da ontologia. Fensel (2003) é um livro que discute as ontologias e o comércio eletrônico.