

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*, 6th 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

Projeto de banco de dados relacional por mapeamento ER e EER para relacional

capítulo

9

Este capítulo discute como projetar um esquema de banco de dados relacional com base em um projeto de esquema conceitual. A Figura 7.1 apresentou uma visão de alto nível do processo de projeto de banco de dados, e neste capítulo focamos a etapa de projeto lógico de banco de dados lógico ou mapeamento de modelo de dados do projeto de banco de dados. Apresentamos os procedimentos para criar um esquema relacional com base em um esquema Entidade-Relacionamento (ER) ou ER estendido (EER). Nossa discussão relaciona as construções dos modelos ER e EER, apresentadas nos capítulos 7 e 8, às construções do modelo relacional, apresentadas nos capítulos 3 a 6. Muitas ferramentas de engenharia de software auxiliada por computador (CASE) são baseadas nos modelos ER e EER, ou outros modelos semelhantes, conforme discutimos nos capítulos 7 e 8. Muitas ferramentas utilizam diagramas ER ou EER ou variações para desenvolver um esquema graficamente, e depois o convertem de maneira automática em um esquema de banco de dados relacional na DDL de um SGBD relacional específico, empregando algoritmos semelhantes aos apresentados neste capítulo.

Esboçamos um algoritmo de sete etapas na Seção 9.1 para converter as construções básicas do modelo ER — tipos de entidade (forte e fraca), relacionamentos binários (com várias restrições estruturais), relacionamentos n -ários e atributos (simples, compostos e multivalorados) — em relações. Depois, na Seção 9.2, continuamos o algoritmo de mapeamento ao descrever como mapear as construções do modelo EER — especialização/generalização e tipos de união (categorias) — em relações. No final do capítulo há um resumo.

9.1 Projeto de banco de dados relacional usando o mapeamento ER para relacional

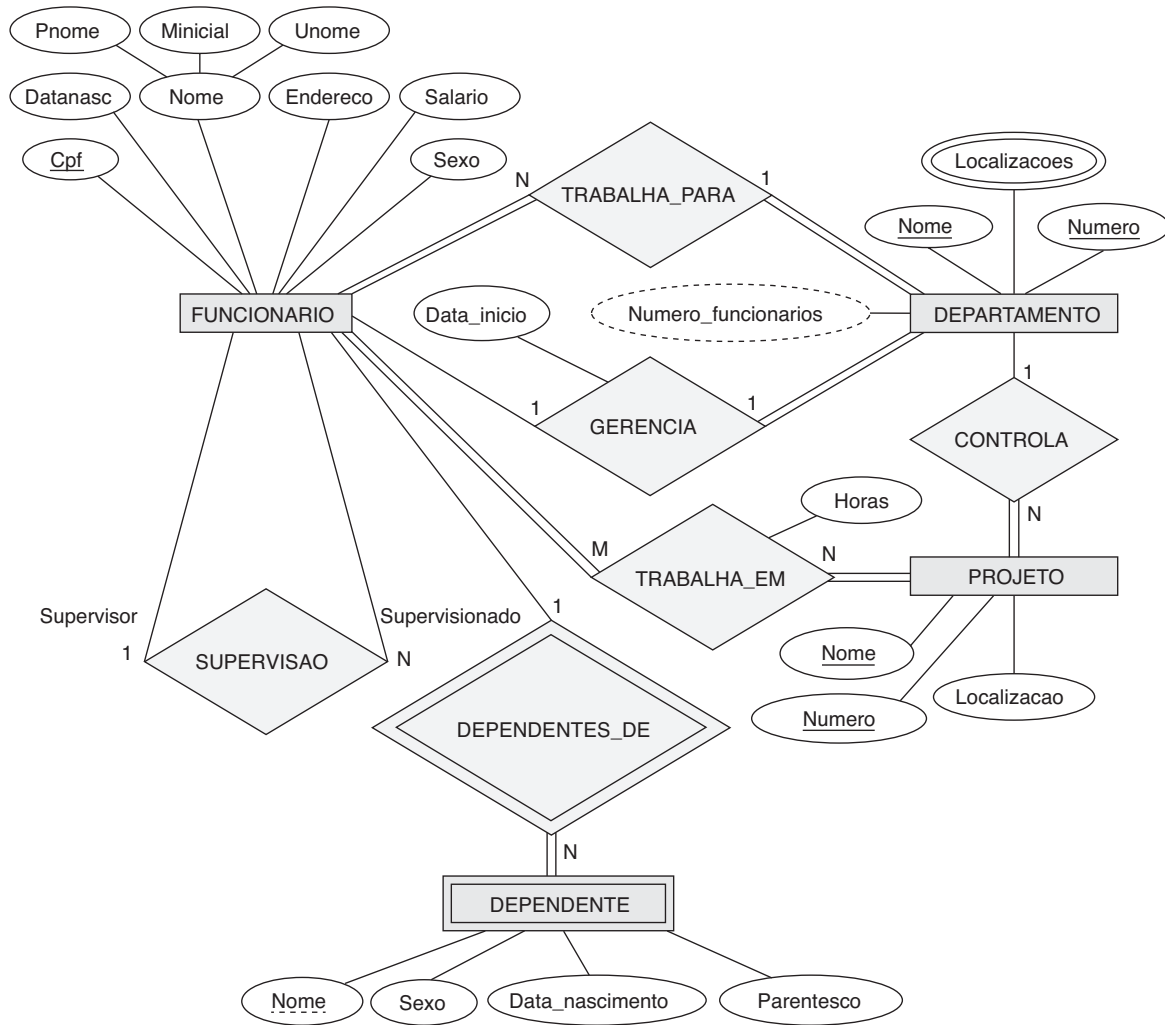
9.1.1 Algoritmo de mapeamento ER para relacional

Nesta seção, vamos descrever as etapas de um algoritmo para mapeamento ER para relacional. Usamos o exemplo de banco de dados EMPRESA para ilustrar o procedimento de mapeamento. O esquema ER EMPRESA aparece novamente na Figura 9.1, e o esquema de banco de dados relacional EMPRESA correspondente aparece na Figura 9.2 para ilustrar as etapas de mapeamento. Assumimos que o mapeamento criará tabelas com atributos simples de único valor. As restrições do modelo relacional definidas no Capítulo 3, que incluem chaves primárias, chaves únicas (se houver) e restrições de integridade referencial sobre as relações, também serão especificadas nos resultados do mapeamento.

Etapas 1: Mapeamento de tipos de entidade regular.

Para cada tipo de entidade regular (forte) E no esquema ER, crie uma relação R que inclua todos os atributos simples de E . Inclua apenas os atributos de componente simples de um atributo composto. Escolha um dos atributos-chave de E como chave primária para R . Se a chave escolhida de E for composta, então o conjunto de atributos simples que a compõem juntos formarão a chave primária de R .

Se várias chaves fossem identificadas para E durante o projeto conceitual, a informação que descreve os atributos que formam cada chave adicional é

**Figura 9.1**

O diagrama do esquema conceitual ER para o banco de dados EMPRESA.

mantida a fim de especificar chaves secundárias (únicas) da relação R . O conhecimento sobre as chaves também é mantido para fins de indexação e outros tipos de análises.

Em nosso exemplo, criamos as relações FUNCIONARIO, DEPARTAMENTO e PROJETO na Figura 9.2 para corresponder aos tipos de entidade regular FUNCIONARIO, DEPARTAMENTO e PROJETO da Figura 9.1. Os atributos de chave estrangeira e relacionamento, se houver, ainda não estão incluídos; eles serão acrescentados durante as etapas seguintes. Estes incluem os atributos Cpf_supervisor e Dnr de FUNCIONARIO, Cpf_gerente e Data_inicio_gerente de DEPARTAMENTO, e Dnum de PROJETO. Em nosso exemplo, escolhemos Cpf, Dnumero e Projnumero como chaves primárias para as relações FUNCIONARIO, DEPARTAMENTO e PROJETO, respectivamente. O conhecimento de que o Dnome de DEPARTAMENTO e o Projnome

de PROJETO são chaves secundárias é mantido para possível uso posterior no projeto.

As relações que são criadas com base no mapeamento dos tipos de entidade às vezes são chamadas **relações de entidade**, pois cada tupla representa uma instância de entidade. O resultado após essa etapa de mapeamento aparece na Figura 9.3(a).

Etapla 2: Mapeamento de tipos de entidade fraca.

Para cada tipo de entidade fraca F no esquema ER com tipo de entidade proprietária E , crie uma relação R e inclua todos os atributos simples (ou componentes simples dos atributos compostos) de F como atributos de R . Além disso, inclua como atributos de chave estrangeira de R os atributos de chave primária da(s) relação(ões) que corresponde(m) aos tipos de entidade proprietária. Isso consegue mapear o tipo de relacionamento de identificação de F . A chave pri-

FUNCIONARIO

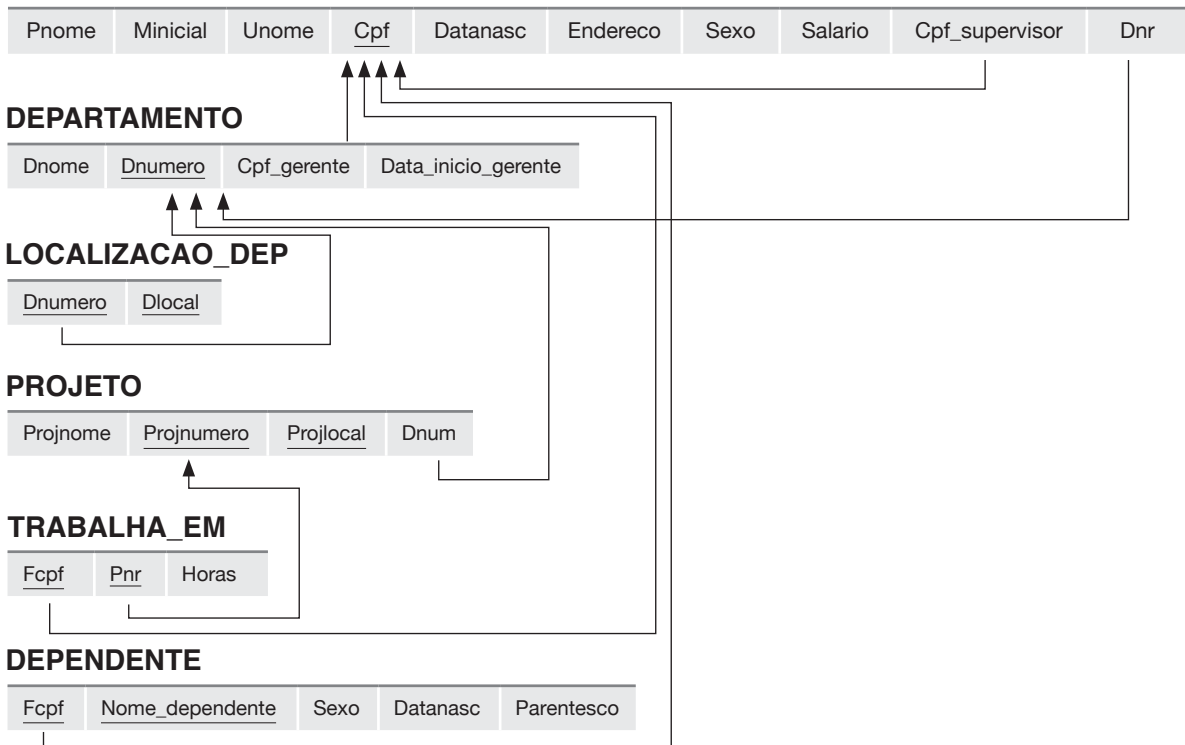


Figura 9.2

Resultado do mapeamento do esquema ER EMPRESA para um esquema de banco de dados relacional.

(a) FUNCIONARIO

Pnome	Minicial	Unome	<u>Cpf</u>	Datanasc	Endereco	Sexo	Salario
-------	----------	-------	------------	----------	----------	------	---------

DEPARTAMENTO

Dnome	<u>Dnumero</u>
-------	----------------

PROJETO

Projnome	<u>Projnumero</u>	<u>Projlocal</u>
----------	-------------------	------------------

(b) DEPENDENTE

<u>Fcpf</u>	<u>Nome_dependente</u>	Sexo	Datanasc	Parentesco
-------------	------------------------	------	----------	------------

(c) TRABALHA_EM

<u>Fcpf</u>	<u>Pnr</u>	Horas
-------------	------------	-------

(d) LOCALIZACAO_DEP

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------

Figura 9.3

Ilustração de algumas etapas de mapeamento. (a) Relações de entidade após a etapa 1. (b) Relação de entidade fraca após a etapa 2. (c) Relação de relacionamento após a etapa 5. (d) Relação representando atributo multivalorado após a etapa 6.

mária de R é a combinação das chaves primárias dos proprietários e a chave parcial do tipo de entidade fraca F , se houver.

Se houver um tipo de entidade fraca E_2 , cujo proprietário também é um tipo de entidade E_1 , então E_1 deve ser mapeado antes de E_2 para determinar primeiro sua chave primária.

Em nosso exemplo, criamos a relação DEPENDENTE nesta etapa para corresponder ao tipo de entidade fraca DEPENDENTE (ver Figura 9.3(b)). Incluímos a chave primária Cpf da relação FUNCIONARIO — que corresponde ao tipo de entidade proprietária — como um atributo de chave estrangeira de DEPENDENTE; renomeamos para Fcpf, embora isso não seja necessário. A chave primária da relação DEPENDENTE é a combinação {Fcpf, Nome_dependente}, pois Nome_dependente (também renomeado de Nome na Figura 9.1) é a chave parcial de DEPENDENTE.

É comum escolher a opção de propagação (CASCADE) para a ação de disparo referencial (ver Seção 4.2) na chave estrangeira na relação correspondente ao tipo de entidade fraca, pois uma entidade fraca tem uma dependência de existência em sua entidade proprietária. Isso pode ser usado para ON UPDATE e ON DELETE.

Etapa 3: Mapeamento dos tipos de relacionamento binários 1:1. Para cada tipo de relacionamento binário 1:1 R no esquema ER, identifique as relações S e T que correspondem aos tipos de entidade participantes em R . Existem três técnicas possíveis: (1) a técnica de chave estrangeira, (2) a técnica de relacionamento mesclado e (3) a técnica de relação de referência cruzada ou relacionamento. A primeira técnica é a mais útil e deve ser seguida a menos que haja condições especiais, conforme discutimos a seguir.

1. **Técnica de chave estrangeira:** escolha uma das relações — digamos, S — e inclua como chave estrangeira em S a chave primária de T . É melhor escolher um tipo de entidade com *participação total* em R no papel de S . Inclua todos os atributos simples (ou componentes simples dos atributos compostos) do tipo de relacionamento 1:1 R como atributos de S . Em nosso exemplo, mapeamos o tipo de relacionamento 1:1 GERENCIA da Figura 9.1 ao escolher o tipo de entidade de participação DEPARTAMENTO para servir ao papel de S , pois sua participação no tipo de relacionamento GERENCIA é total (cada departamento tem um gerente). Incluímos a chave primária da relação FUNCIONARIO como chave estrangeira na relação DEPARTAMENTO e a renomeamos como Cpf_gerente. Também incluímos

o atributo simples Data_inicio do tipo de relacionamento GERENCIA na relação DEPARTAMENTO e o renomeamos como Data_inicio_gerente (ver Figura 9.2).

Observe que é possível incluir a chave primária de S como uma chave estrangeira em T em vez disso. Em nosso exemplo, isso significa ter um atributo de chave estrangeira, digamos, Depart_gerenciado na relação FUNCIONARIO, mas terá um valor NULL para as tuplas de funcionários que não gerenciam um departamento. Se apenas 2 por cento dos funcionários gerenciam um departamento, então 98 por cento das chaves estrangeiras seriam NULL nesse caso. Outra possibilidade é ter chaves estrangeiras nas relações S e T de maneira redundante, mas isso cria redundância e agrega uma penalidade para a manutenção da consistência.

2. **Técnica de relação mesclada:** um mapeamento alternativo de um tipo de relacionamento 1:1 é mesclar os dois tipos de entidade e o relacionamento em uma única relação. Isso é possível quando *ambas as participações são totais*, pois indicaria que as duas tabelas terão exatamente o mesmo número de tuplas o tempo inteiro.
3. **Técnica de relação de referência cruzada ou relacionamento:** a terceira opção é configurar uma terceira relação R para a finalidade de referência cruzada das chaves primárias das duas relações S e T representando os tipos de entidade. Conforme veremos, essa técnica é exigida para relacionamentos M:N binários. A relação R é chamada de **relação de relacionamento** (ou, às vezes, de **tabela de pesquisa**), porque cada tupla em R representa uma instância de relacionamento que relaciona uma tupla de S a uma tupla de T . A relação R incluirá os atributos de chave primária de S e T como chaves estrangeiras para S e T . A chave primária de R será uma das duas chaves estrangeiras, e a outra chave estrangeira será uma chave única de R . A desvantagem é ter uma relação extra e exigir uma operação de junção extra ao combinar tuplas relacionadas das tabelas.

Etapa 4: Mapeamento de tipos de relacionamento binário 1:N. Para cada tipo de relacionamento R binário regular 1:N, identifique a relação S que representa o tipo de entidade participante no *lado N* do tipo de relacionamento. Inclua como chave estrangeira em S a chave primária da relação T que representa

o outro tipo de entidade participante em R ; fazemos isso porque cada instância de entidade no lado N está relacionada a, no máximo, uma instância de entidade no lado 1 do tipo de relacionamento. Inclua quaisquer atributos simples (ou componentes simples dos atributos compostos) do tipo de relacionamento $1:N$ como atributos de S .

Em nosso exemplo, agora mapeamos os tipos de relacionamento $1:N$ TRABALHA_PARA, CONTROLA e SUPERVISAO da Figura 9.1. Para TRABALHA_PARA, incluímos a chave primária Dnumero da relação DEPARTAMENTO como chave estrangeira na relação FUNCIONARIO e a chamamos de Dnr. Para SUPERVISAO, incluímos a chave primária da relação FUNCIONARIO como chave estrangeira na própria relação FUNCIONARIO — pois o relacionamento é recursivo — e a chamamos de Cpf_supervisor. O relacionamento CONTROLA é mapeado para o atributo de chave estrangeira Dnum de PROJETO, que referencia a chave primária Dnumero da relação DEPARTAMENTO. Essas chaves estrangeiras são mostradas na Figura 9.2.

Uma técnica alternativa é usar a opção de **relação de relacionamento** (referência cruzada) como na terceira opção para os relacionamentos binários $1:1$. Criamos uma relação separada R cujos atributos são chaves primárias de S e T , que também serão chaves estrangeiras para S e T . A chave primária de R é igual à chave primária de S . Essa opção pode ser usada se algumas tuplas em S participarem do relacionamento para evitar valores NULL excessivos na chave estrangeira.

Etapas 5: Mapeamento de tipos de relacionamento binário $M:N$. Para cada tipo de relacionamento R binário $M:N$, crie uma nova relação S para representar R . Inclua como atributos de chave estrangeira em S as chaves primárias das relações que representam os tipos de entidade participantes; sua *combinação* formará a chave primária de S . Inclua também quaisquer atributos simples do tipo de relacionamento $M:N$ (ou componentes simples dos atributos compostos) como atributos de S . Observe que não podemos representar um tipo de relacionamento $M:N$ por um único atributo de chave estrangeira em uma das relações participantes (como fizemos para os tipos de relacionamento $1:1$ ou $1:N$) devido à razão de cardinalidade $M:N$; temos de criar uma *relação de relacionamento* S separada.

Em nosso exemplo, mapeamos o tipo de relacionamento $M:N$ TRABALHA_EM da Figura 9.1 criando a relação TRABALHA_EM na Figura 9.2. Incluímos as chaves primárias das relações PROJETO e FUNCIONARIO como chaves estrangeiras em TRABALHA_EM e as renomeamos como Pnr e Fcpf, respectivamente.

Também incluímos um atributo Horas em TRABALHA_EM para representar o atributo Horas do tipo de relacionamento. A chave primária da relação TRABALHA_EM é a combinação dos atributos de chave estrangeira {Fcpf, Pnr}. Essa **relação de relacionamento** aparece na Figura 9.3(c).

A opção de propagação (CASCADE) para a ação de disparo referencial (ver Seção 4.2) deve ser especificada sobre as chaves estrangeiras na relação correspondente ao relacionamento R , pois cada instância de relacionamento tem uma dependência de existência sobre cada uma das entidades a que ela se relaciona. Isso pode ser usado tanto para ON UPDATE quanto para ON DELETE.

Observe que sempre podemos mapear relacionamentos $1:1$ ou $1:N$ de uma maneira semelhante aos relacionamentos $M:N$ usando a técnica de referência cruzada (relação de relacionamento), conforme discutimos anteriormente. Essa alternativa é particularmente útil quando existem poucas instâncias de relacionamentos, a fim de evitar valores NULL em chaves estrangeiras. Nesse caso, a chave primária da relação de relacionamento será *apenas uma* das chaves estrangeiras que referenciam as relações da entidade participante. Para um relacionamento $1:N$, a chave primária da relação de relacionamento será a chave estrangeira que referencia a relação de entidade no lado N . Para um relacionamento $1:1$, qualquer chave estrangeira pode ser usada como chave primária da relação de relacionamento.

Etapas 6: Mapeamento de atributos multivalorados.

Para cada atributo multivalorado A , crie uma relação R . Essa relação R incluirá um atributo correspondente a A , mais o atributo da chave primária Ch — como uma chave estrangeira em R — da relação que representa o tipo de entidade ou tipo de relacionamento que tem A como atributo multivalorado. A chave primária de R é a combinação de A e Ch . Se o atributo multivalorado for composto, incluímos seus componentes simples.

Em nosso exemplo, criamos uma relação LOCALIZACAO_DEP (ver Figura 9.3(d)). O atributo Dlocalizacao representa o atributo multivalorado LOCALIZACOES de DEPARTAMENTO, enquanto Dnumero — como chave estrangeira — representa a chave primária da relação DEPARTAMENTO. A chave primária de LOCALIZACAO_DEP é a combinação de {Dnumero, Dlocalizacao}. Uma tupla separada existirá em LOCALIZACAO_DEP para cada local que tenha um departamento.

A opção de propagação (CASCADE) para a ação de disparo referencial (ver Seção 4.2) deve ser especificada na chave estrangeira da relação R correspondente ao atributo multivalorado para ON UPDATE e

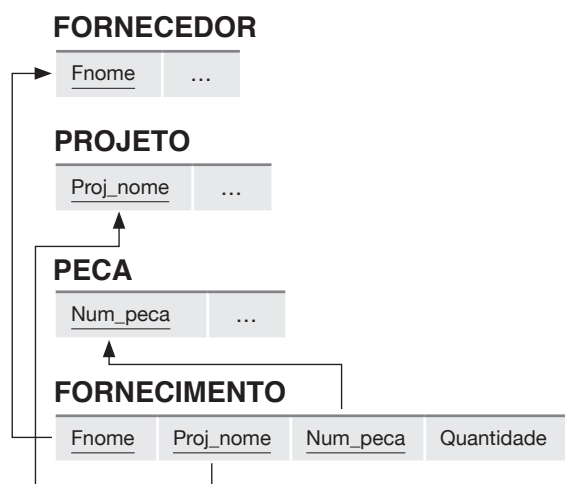


Figura 9.4
Mapeando o tipo de relacionamento *n*-ário FORNECIMENTO da Figura 7.17(a).

ON DELETE. Também devemos observar que a chave de *R*, ao mapear um atributo composto, multivalorado, requer alguma análise do significado dos atributos componentes. Em alguns casos, quando um atributo multivalorado é composto, somente alguns dos atributos componentes são exigidos para fazer parte da chave de *R*. Esses atributos são semelhantes à chave parcial de um tipo de entidade fraca que corresponde ao atributo multivalorado (ver Seção 7.5).

A Figura 9.2 mostra o esquema de banco de dados relacional EMPRESA obtido com as etapas 1 a 6, e a Figura 3.6 mostra um exemplo de estado de banco de dados. Observe que ainda não discutimos o mapeamento de tipos de relacionamento *n*-ário ($n > 2$), pois ele não existe na Figura 9.1. Estes são mapeados de um modo semelhante aos tipos de relacionamento M:N, incluindo a etapa adicional a seguir no algoritmo de mapeamento.

Etapla 7: Mapeamento de tipos de relacionamen- to *n*-ário. Para cada tipo de relacionamento *n*-ário *R*, onde $n > 2$, crie uma relação *S* para representar *R*. Inclua como atributos de chave estrangeira em *S* as chaves primárias das relações que representam os tipos de entidade participantes. Inclua também quaisquer atributos simples do tipo de relacionamen- to *n*-ário (ou componentes simples de atributos compo- stos) como atributos de *S*. A chave primária de *S* normalmente é uma combinação de todas as chaves estrangeiras que referenciam as relações represen- tando os tipos de entidade participantes. Porém, se as restrições de cardinalidade sobre qualquer um dos tipos de entidade *E* participantes em *R* for 1, então a chave primária de *S* não deve incluir o atributo de chave estrangeira que referencia a relação *E* corres-

Tabela 9.1
Correspondência entre os modelos ER e relacional.

MODELO ER	MODELO RELACIONAL
Tipo de entidade	Relação de <i>entidade</i>
Tipo de relacionamento 1:1 ou 1:N	Chave estrangeira (ou relação de <i>relacionamento</i>)
Tipo de relacionamento M:N	Relação de <i>relacionamento</i> e <i>duas</i> chaves estrangeiras
Tipo de relacionamento <i>n</i> -ário	Relação de <i>relacionamento</i> e <i>n</i> chaves estrangeiras
Atributo simples	Atributo
Atributo composto	Conjunto de atributos componentes simples
Atributo multivalorado	Relação e chave estrangeira
Conjunto de valores	Domínio
Atributo-chave	Chave primária (ou secundária)

pondente a *E* (ver discussão na Seção 7.9.2, referente a restrições sobre relacionamentos *n*-ários).

Por exemplo, considere o tipo de relacionamento FORNECIMENTO da Figura 7.17. Este pode ser mapeado para a relação FORNECIMENTO mostrada na Figura 9.4, cuja chave primária é a combinação das três chaves estrangeiras {Fnome, Num_peca, Proj_nome}.

9.1.2 Discussão e resumo do mapeamento para construções no modelo ER

A Tabela 9.1 resume as correspondências entre as construções e restrições do modelo ER e relacional.

Um dos principais pontos a observar em um esquema relacional, ao contrário de um esquema ER, é que os tipos de relacionamento não são representados explicitamente. Em vez disso, eles são representados com dois atributos *A* e *B*, um é uma chave primária e o outro é uma chave estrangeira (no mesmo domínio) incluída em duas relações *S* e *T*. Duas tuplas em *S* e *T* são relacionadas quando têm o mesmo valor para *A* e *B*. Usando a operação EQUIJUNÇÃO (ou JUNÇÃO NATURAL, se os dois atributos de junção tiverem o mesmo nome) em *S.A* e *T.B*, podemos combinar todos os pares de tuplas relacionadas de *S* e *T* e materializar o relacionamento. Quando um tipo de relacionamento binário 1:1 ou 1:N é envolvido, uma única operação de junção costuma ser necessária. Para um tipo de relacionamento binário M:N, duas operações de junção são necessárias, enquanto para tipos de relacionamento *n*-ários, *n* junções são necessárias para materializar totalmente as instâncias de relacionamento.

Por exemplo, para formar uma relação que inclui o nome do funcionário, nome do projeto e horas

que o funcionário trabalha em cada projeto, precisamos conectar cada tupla FUNCIONARIO às tuplas PROJETO relacionadas por meio da relação TRABALHA_EM na Figura 9.2. Logo, precisamos aplicar a operação EQUIJUNÇÃO às relações FUNCIONARIO e TRABALHA_EM com a condição de junção $Cpf = Fcpf$, e depois aplicar outra operação EQUIJUNÇÃO à relação resultante e a relação PROJETO com a condição de junção $Pnr = Projnumero$. Em geral, quando vários relacionamentos precisam ser examinados, diversas operações de junção precisam ser especificadas. Um usuário de banco de dados relacional sempre precisa estar ciente dos atributos de chave estrangeira para poder usá-los corretamente na combinação de tuplas relacionadas de duas ou mais relações. Isso às vezes é considerado uma desvantagem do modelo de dados relacional, porque as correspondências de chave estrangeira/chave primária nem sempre são óbvias pela inspeção dos esquemas relacionais. Se uma EQUIJUNÇÃO for realizada entre atributos de duas relações que não representam um relacionamento de chave estrangeira/chave primária, o resultado pode com frequência ser sem sentido e levar a dados falsos (espúrios). Por exemplo, o leitor pode tentar juntar as relações PROJETO e LOCALIZACAO_DEP na condição $Dlocal = Projlocal$ e examinar o resultado (veja a discussão sobre tuplas espúrias na Seção 15.1.4).

No esquema relacional, criamos uma relação separada para *cada* atributo multivalorado. Para uma entidade em particular com um conjunto de valores para o atributo multivalorado, o valor do atributo-chave da entidade é repetido uma vez para cada valor do atributo multivalorado em uma tupla separada, pois o modelo relacional básico *não* permite valores múltiplos (uma lista, ou um conjunto de valores) para um atributo em uma única tupla. Por exemplo, como o departamento 5 tem três locais, existem três tuplas na relação LOCALIZACAO_DEP da Figura 3.6; cada tupla especifica um dos locais. Em nosso exemplo, aplicamos EQUIJUNÇÃO a LOCALIZACAO_DEP e DEPARTAMENTO no atributo Dnumero para obter os valores de todas as localizações junto com outros atributos de DEPARTAMENTO. Na relação resultante, os valores dos outros atributos de DEPARTAMENTO são repetidos em tuplas separadas para cada local que tenha um departamento.

A álgebra relacional básica não tem uma operação ANINHAR ou COMPRIMIR que produziria um conjunto de tuplas na forma $\{ \langle '1', 'São Paulo' \rangle, \langle '4', 'Mauá' \rangle, \langle '5', 'Santo André' \rangle, \langle 'Itu', 'São Paulo' \rangle \}$ com base na relação LOCALIZACAO_DEP da Figura 3.6. Essa é uma desvantagem séria da versão básica normalizada ou do modelo relacional.

O modelo de dados de objeto e os sistemas objeto-relacional (ver Capítulo 11) permitem atributos multivalorados.

9.2 Mapeando construções do modelo EER para relações

A seguir, vamos discutir o mapeamento das construções do modelo EER para relações, estendendo o algoritmo de mapeamento ER para relacional que foi apresentado na Seção 9.1.1.

9.2.1 Mapeamento da especialização ou generalização

Existem várias opções para mapear uma série de subclasses que juntas formam uma especialização (ou, como alternativa, que são generalizadas para uma superclasse), como as subclasses {SECRETARIA, TECNICO, ENGENHEIRO} de FUNCIONARIO na Figura 8.4. Podemos acrescentar outro passo ao nosso algoritmo de mapeamento da Seção 9.1.1, que tem sete etapas, para lidar com o mapeamento da especialização. A etapa 8, que vem a seguir, oferece as opções mais comuns; outros mapeamentos também são possíveis. Discutimos as condições sob as quais cada opção deve ser usada. Usamos $Atrs(R)$ para indicar os atributos da relação R e $ChP(R)$ para indicar a *chave primária* de R . Primeiro, vamos descrever o mapeamento formalmente e depois ilustrar com exemplos.

Etapla 8: Opções para mapeamento da especialização ou generalização. Converta cada especialização com m subclasses $\{S_1, S_2, \dots, S_m\}$ e superclasse (generalizada) C , em que os atributos de C são $\{ch, a_1, \dots, a_n\}$ e ch é a chave (primária) para os esquemas da relação usando uma das seguintes opções:

- **Opção 8A: Múltiplas relações — superclasse e subclasses.** Crie uma relação L para C com atributos $Atrs(L) = \{ch, a_1, \dots, a_n\}$ e $ChP(L_i) = ch$. Crie uma relação L_i para cada subclasse S_i , $1 \leq i \leq m$, com os atributos $Atrs(L_i) = \{ch\} \cup \{\text{atributos de } S_i\}$ e $ChP(L_i) = ch$. Essa opção funciona para qualquer especialização (total ou parcial, disjunta ou sobreposta).
- **Opção 8B: Múltiplas relações — apenas relações de subclasse.** Crie uma relação L_i para cada subclasse S_i , $1 \leq i \leq m$, com os atributos $Atrs(L_i) = \{\text{atributos de } S_i\} \cup \{ch, a_1, \dots, a_n\}$ e $ChP(L_i) = ch$. Essa opção só funciona para uma especialização cujas subclasses são *totais* (cada entidade na superclasse deve pertencer a (pelo menos) uma das subclas-

ses). Além disso, isso só é recomendado se a especialização tiver a *restrição de disjunção* (ver Seção 8.3.1). Se a especialização for *sobreposta*, a mesma entidade pode ser duplicada em várias relações.

- **Opção 8C: Relação única com um atributo de tipo.** Crie uma única relação L com atributos $\text{Atrs}(L) = \{ch, a_1, \dots, a_n\} \cup \{\text{atributos de } S_1\} \cup \dots \cup \{\text{atributos de } S_m\} \cup \{t\}$ e $\text{ChP}(L) = ch$. O atributo t é chamado de atributo de tipo (ou **discriminador**), cujo valor indica a subclasse à qual cada tupla pertence, se houver alguma. Essa opção funciona somente para uma especialização cujas subclasses são *disjuntas*, e tem o potencial para gerar muitos valores NULL se diversos atributos específicos existirem nas subclasses.
- **Opção 8D: Relação isolada com atributos de múltiplos tipos.** Crie um único esquema de relação L com atributos $\text{Atrs}(L) = \{ch, a_1, \dots, a_n\} \cup \{\text{atributos de } S_1\} \cup \dots \cup \{\text{atributos de } S_m\} \cup \{t_1, t_2, \dots, t_m\}$ e $\text{ChP}(L) = ch$. Cada t_i , $1 \leq i \leq m$, é um **atributo de tipo booleano** indicando se uma tupla pertence à subclasse S_i . Essa opção é usada para uma especialização cujas subclasses são *sobrepostas* (mas também funcionará para uma especialização disjunta).

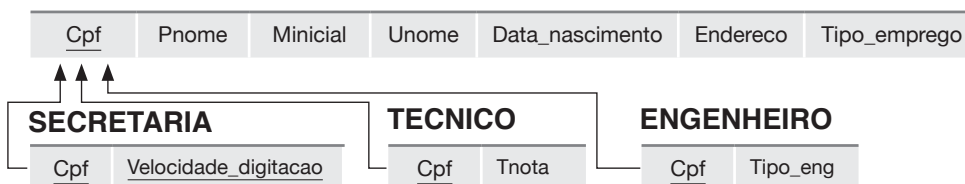
As opções 8A e 8B podem ser chamadas de **opções de relação múltipla**, enquanto as opções 8C e 8D podem ser chamadas de **opções de relação única**. A opção 8A cria uma relação L para a superclasse C e seus atributos, mais uma relação L_i para cada subclasse S_i ; cada L_i inclui os atributos específicos (ou locais) de S_i , mais a chave primária da superclasse C , que é propagada para L_i e torna-se sua chave primária. Ela também se torna uma chave estrangeira para a relação da superclasse. Uma operação EQUIJUNÇÃO na chave primária entre qualquer L_i e L produz todos os atributos específicos e herdados das entidades em S_i . Essa opção é ilustrada na Figura 9.5(a) para o esquema EER da Figura 8.4. A opção 8A funciona para quaisquer restrições sobre a especialização: disjuntas ou sobrepostas, totais ou parciais. Observe que a restrição

$$\pi_{\langle ch \rangle}(L_i) \subseteq \pi_{\langle ch \rangle}(L)$$

precisa ser mantida para cada L_i . Esta especifica uma chave estrangeira de cada L_i para L , bem como uma *dependência de inclusão* $L_i.ch < L.ch$ (ver Seção 16.5).

Na opção 8B, a operação EQUIJUNÇÃO entre cada subclasse e a superclasse é *embutida* no esquema e a relação L é abolida, conforme ilustra a Figura 9.5(b) para a especialização EER na Figura 8.3(b). Essa opção funciona bem somente quando *ambas* as restrições de disjunção e total são mantidas. Se a especialização não

(a) FUNCIONARIO



(b) CARRO

<u>Id_veiculo</u>	Placa	Preco	Velocidade_max	Numero_passageiros
-------------------	-------	-------	----------------	--------------------

CAMINHAO

<u>Id_veiculo</u>	Placa	Preco	Numero_eixos	Capacidade_peso
-------------------	-------	-------	--------------	-----------------

(c) FUNCIONARIO

<u>Cpf</u>	Pnome	Minicial	Unome	Data_nascimento	Endereco	Tipo_emplogo	Velocidade_digitacao	Grau_tec	Tipo_eng
------------	-------	----------	-------	-----------------	----------	--------------	----------------------	----------	----------

(d) PECA

<u>Peca_nr</u>	Descricao	Tipo_fabr	Num_desenho	Data_fabricacao	Num_lote	Tipo_compr	Nome_fornecedor	Preco
----------------	-----------	-----------	-------------	-----------------	----------	------------	-----------------	-------

Figura 9.5

Opções para mapeamento de especialização ou generalização. (a) Mapeando o esquema EER na Figura 8.4 ao usar a opção 8A. (b) Mapeando o esquema EER na Figura 8.3(b) ao usar a opção 8B. (c) Mapeando o esquema EER na Figura 8.4 ao usar a opção 8C. (d) Mapeando a Figura 8.5 ao usar a opção 8D com campos de tipo booleano Tipo_fabr e Tipo_compr.

for total, uma entidade que não pertence a qualquer uma das subclasses S_i é perdida. Se a especialização não for disjunta, uma entidade pertencente a mais de uma subclasse terá seus atributos herdados da superclasse C armazenada de maneira redundante em mais de um L_i . Com a opção 8B, nenhuma relação mantém todas as entidades na superclasse C ; consequentemente, temos de aplicar uma operação UNIÃO EXTERNA (ou JUNÇÃO EXTERNA COMPLETA) (ver Seção 6.4) às relações L_i para recuperar todas as entidades em C . O resultado da união externa será semelhante às relações sob as opções 8C e 8D, exceto que os campos de tipo estarão faltando. Sempre que procurarmos uma entidade arbitrária em C , devemos procurar todas as m relações L_i .

As opções 8C e 8D criam uma única relação para representar a superclasse C e todas as suas subclasses. Uma entidade que não pertence a nenhuma das subclasses terá valores NULL para os atributos específicos dessas subclasses. Essas opções não são recomendadas se muitos atributos específicos forem definidos para as subclasses. Contudo, se houver poucos atributos de subclasse, esses mapeamentos são preferíveis às opções 8A e 8B porque dispensam a necessidade de especificar operações EQUIJUNÇÃO e UNIÃO EXTERNA. Portanto, eles podem produzir uma implementação mais eficiente.

A opção 8C é utilizada para lidar com subclasses disjuntas, incluindo um único **atributo de tipo** (ou de **imagem** ou **discriminador**) t para indicar a qual das m subclasses cada tupla pertence; logo, o domínio de t poderia ser $\{1, 2, \dots, m\}$. Se a especialização for parcial, t pode ter valores NULL em tuplas que não pertencem a nenhuma subclasse. Se a especialização for definida por atributo, esse atributo que serve à finalidade de t e t não é necessário. Tal opção é ilustrada na Figura 9.5(c) para a especialização EER da Figura 8.4.

A opção 8D é projetada para lidar com subclasses sobrepostas incluindo m campos de **tipo** (ou **flag**) *booleano*, um para *cada* subclasse. Ela também pode ser usada para subclasses disjuntas. Cada campo de tipo t_i pode ter um domínio $\{\text{sim}, \text{não}\}$, em que um valor *sim* indica que a tupla é um membro da subclasse S_i . Se usarmos essa opção para a especialização EER na Figura 8.4, incluiríamos três atributos de tipo — *tipo_secretaria*, *tipo_engenheiro* e *tipo_tecnico* — no lugar do atributo *Tipo_emprego* da Figura 9.5(c). Observe que também é possível criar um único atributo de tipo de m bits em vez dos m campos de tipo. A Figura 9.5(d) mostra o mapeamento da especialização da Figura 8.5 usando a opção 8D.

Quando temos uma hierarquia ou reticulado de especialização (ou generalização) multinível, não precisamos seguir a mesma opção de mapeamento para todas as especializações. Em vez disso, podemos utilizar uma opção de mapeamento para parte da hierarquia ou reticulado e outras opções para outras partes. A Figura 9.6 mostra um mapeamento possível para as relações do reticulado EER da Figura 8.7. Aqui, usamos a opção 8A para PESSOA/{FUNCIONARIO, EX_ALUNO, ALUNO}, a opção 8C para FUNCIONARIO/{ADMINISTRATIVO, DOCENTE, ALUNO_COLABORADOR}, incluindo o atributo de tipo *Tipo_funcionario*, e a opção 8D para ALUNO_COLABORADOR/{COLABORADOR_PESQUISA, COLABORADOR_ENSINO} ao incluir os atributos de tipo *Tipo_col_ensino* e *Tipo_col_pesq* em FUNCIONARIO, ALUNO/ALUNO_COLABORADOR incluindo os atributos de tipo *Tipo_aluno_col* em ALUNO, e ALUNO/{ALUNO_POSGRADUACAO, ALUNO_GRADUACAO} incluindo os atributos de tipo *tipo_pos* e *tipo_grad* em

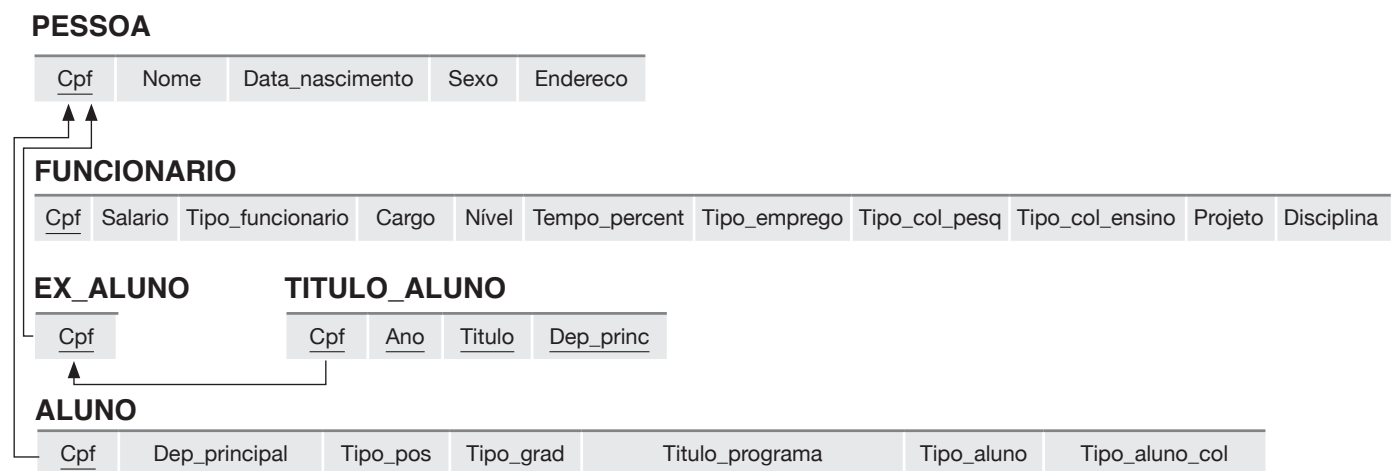


Figura 9.6

Mapeando o reticulado de especialização EER da Figura 8.7 usando opções múltiplas.

ALUNO. Na Figura 9.6, todos os atributos cujos nomes iniciam com *tipo* são campos de tipo.

9.2.2 Mapeamento de subclasses compartilhadas (herança múltipla)

Uma subclasse compartilhada, como GERENTE_ENGENHEIRO da Figura 8.6, é uma subclasse de várias superclasses, indicando a herança múltipla. Todas essas classes precisam ter o mesmo atributo-chave; caso contrário, a subclasse compartilhada seria modelada como uma categoria (tipo de união), conforme discutimos na Seção 8.4. Podemos aplicar qualquer uma das opções discutidas na etapa 8 a uma subclasse compartilhada, sujeita às restrições discutidas na etapa 8 do algoritmo de mapeamento. Na Figura 9.6, as opções 8C e 8D são utilizadas para a subclasse compartilhada ALUNO_COLABORADOR. A opção 8C é usada na relação FUNCIONARIO (atributo Tipo_funcionario) e a opção 8D, na relação ALUNO (atributo Tipo_aluno_col).

9.2.3 Mapeamento de categorias (tipos de união)

Acrescentamos outra etapa ao procedimento de mapeamento — etapa 9 — para lidar com categorias. Uma categoria (ou tipo de união) é uma subclasse da *união* de duas ou mais superclasses que podem ter diferentes chaves porque podem ser de diferentes tipos de entidade (ver Seção 8.4). Um exemplo é a categoria PROPRIETARIO mostrada na Figura 8.8, que é um subconjunto da união de três tipos de entidade PESSOA, BANCO e EMPRESA. A outra categoria nessa figura, VEICULO_REGISTRADO, tem duas superclasses que possuem o mesmo atributo de chave.

Etapla 9: Mapeamento de tipos de união (categorias). Para o mapeamento de uma categoria cuja definição de superclasses tem chaves diferentes, é comum especificar um novo atributo-chave, chamado **chave substituta**, ao criar uma relação para corresponder à categoria. As chaves das classes de definição são diferentes e, portanto, não podemos usar nenhuma delas exclusivamente para identificar todas as entidades na categoria. Em nosso exemplo da Figura 8.8, criamos uma relação PROPRIETARIO para corresponder à categoria PROPRIETARIO, conforme ilustrado na Figura 9.7, e incluímos alguns atributos da categoria nessa relação. A chave primária da relação PROPRIETARIO é a chave substituta, que chamamos de *Id_proprietario*. Também incluímos o atributo de chave substituta *Id_proprietario* como uma chave estrangeira em cada relação correspondente a uma superclasse da categoria,

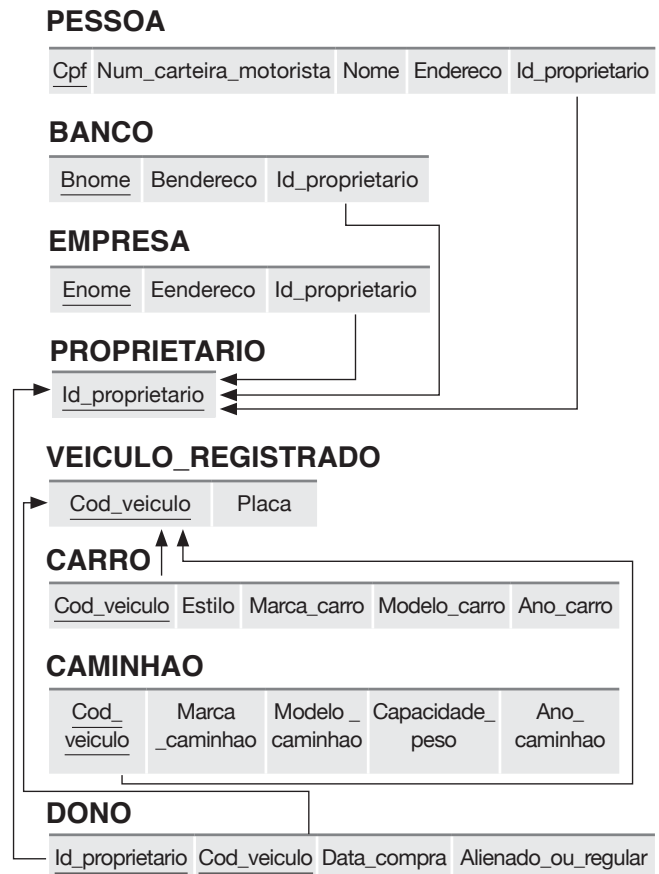


Figura 9.7

Mapeamento das categorias EER (tipos de união) na Figura 8.8 para relações.

para especificar a correspondência nos valores entre a chave substituta e a chave de cada superclasse. Observe que, se determinada entidade PESSOA (ou BANCO ou EMPRESA) não for um membro de PROPRIETARIO, ela teria um valor NULL para seu atributo *Id_proprietario* em sua tupla correspondente na relação PESSOA (ou BANCO ou EMPRESA), e não teria uma tupla na relação PROPRIETARIO. Também é recomendado acrescentar um atributo de tipo (não mostrado na Figura 9.7) à relação PROPRIETARIO para indicar o tipo de entidade em particular ao qual cada tupla pertence (PESSOA, BANCO ou EMPRESA).

Para uma categoria cujas superclasses têm a mesma chave, como VEICULO na Figura 8.8, não há necessidade de uma chave substituta. O mapeamento da categoria VEICULO_REGISTRADO, que ilustra esse caso, também aparece na Figura 9.7.

Resumo

Na Seção 9.1, mostramos como um projeto de esquema conceitual no modelo ER pode ser mapeado para um esquema de banco de dados relacional. Um algorit-

mo para mapeamento ER para relacional foi dado e ilustrado por meio de exemplos do banco de dados EMPRESA. A Tabela 9.1 resumiu as correspondências entre as construções e restrições do modelo ER e relacional. Em seguida, acrescentamos as etapas adicionais ao algoritmo da Seção 9.2 para mapear as construções do modelo EER para o modelo relacional. Algoritmos semelhantes são incorporados nas ferramentas gráficas de banco de dados para criar um esquema relacional com base em um projeto de esquema conceitual automaticamente.

Perguntas de revisão

- 9.1. Discuta as correspondências entre as construções do modelo ER e as construções do mo-

delo relacional. Mostre como cada construção do modelo ER pode ser mapeada para o modelo relacional e discuta quaisquer mapeamentos alternativos.

- 9.2. Discuta as opções para mapear as construções do modelo EER para relações.

Exercícios

- 9.3. Tente mapear o esquema relacional da Figura 6.14 em um esquema ER. Isso faz parte de um processo conhecido como *engenharia reversa*, em que um esquema conceitual é criado para um banco de dados implementado existente. Indique quaisquer suposições que você fizer.

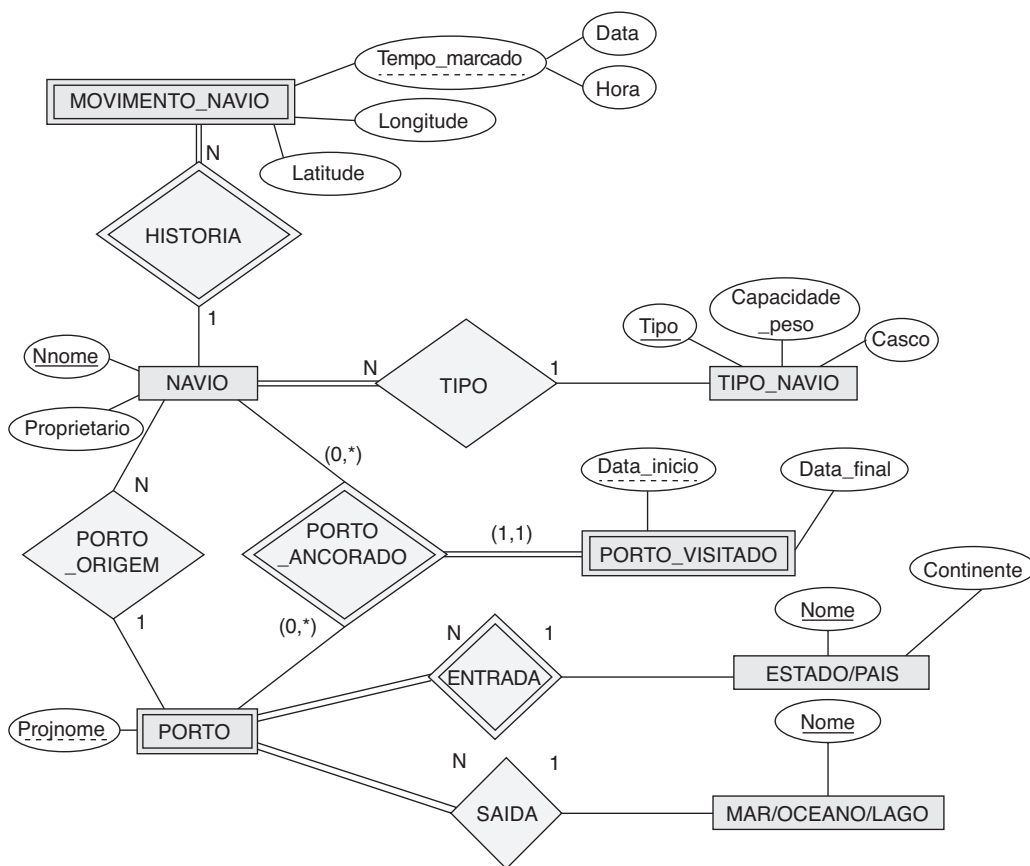


Figura 9.8

Um esquema ER para um banco de dados MOVIMENTO_NAVIO.

- 9.4. A Figura 9.8 mostra um esquema ER para um banco de dados que pode ser usado para registrar navios de transporte e seus locais para autoridades marítimas. Mapeie esse esquema para um esquema relacional e especifique todas as chaves primárias e estrangeiras.
- 9.5. Mapeie o esquema ER BANCO do Exercício 7.23 (mostrado na Figura 7.21) em um esquema relacional. Especifique todas as chaves primárias e estrangeiras. Repita para o esquema COMPANHIA

AEREA (Figura 7.20) do Exercício 7.19 e para os outros esquemas dos exercícios 7.16 a 7.24.

- 9.6. Mapeie os diagramas EER das figuras 8.9 e 8.12 para esquemas relacionais. Justifique sua escolha de opções de mapeamento.
- 9.7. É possível mapear com sucesso um tipo de relacionamento binário M:N sem exigir uma nova relação? Por quê?
- 9.8. Considere o diagrama EER da Figura 9.9 para um revendedor de automóveis.

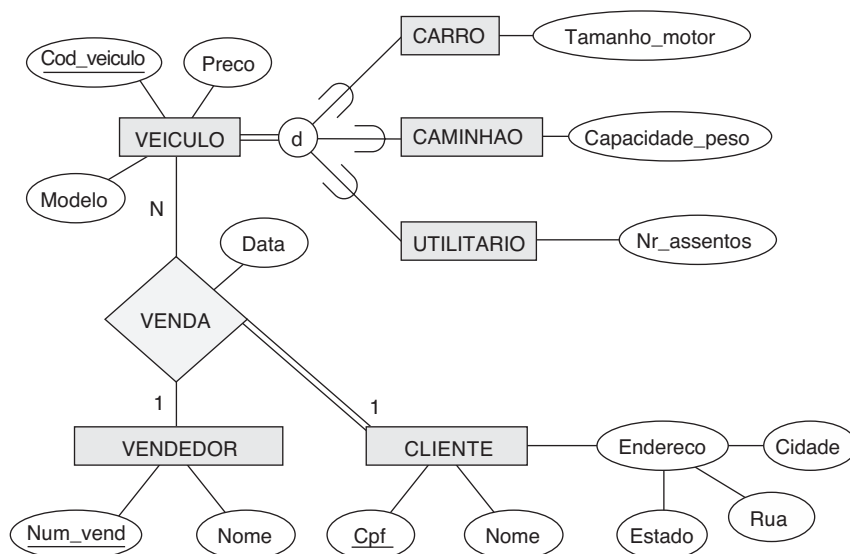
**Figura 9.9**

Diagrama EER para um revendedor de automóveis.

Mapeie o esquema EER para um conjunto de relações. Para a generalização de VEICULO para CARRO/CAMINHAO/UTILITARIO, considere as quatro opções apresentadas na Seção 9.2.1 e mostre o projeto de esquema relacional sob cada uma dessas opções.

- 9.9.** Usando os atributos que você forneceu para o diagrama EER do Exercício 8.27, mapeie o esquema completo para um conjunto de relações. Escolha uma opção apropriada de 8A até 8D, da Seção 9.2.1, fazendo o mapeamento de generalizações, e defenda sua escolha.

Exercícios de laboratório

- 9.10.** Considere o projeto ER para o banco de dados UNIVERSIDADE que foi modelado usando uma ferramenta como ERwin ou Rational Rose no Exercício de Laboratório 7.31. Utilizando o recurso de geração de esquema SQL da ferramenta de modelagem, gere o esquema SQL para um banco de dados Oracle.
- 9.11.** Considere o projeto ER para o banco de dados PEDIDO_CORREIO que foi modelado usando uma ferramenta como ERwin ou Rational Rose no Exercício de Laboratório 7.32. Utilizando o recurso de geração de esquema SQL da ferramenta de modelagem, gere o esquema SQL para um banco de dados Oracle.

- 9.12.** Considere o projeto ER para o banco de dados REVISAO_CONFERENCIA que foi modelado usando uma ferramenta como ERwin ou Rational Rose no Exercício de Laboratório 7.34. Utilizando o recurso de geração de esquema SQL da ferramenta de modelagem, gere o esquema SQL para um banco de dados Oracle.

- 9.13.** Considere o projeto EER para o banco de dados DIARIO_NOTAS que foi modelado usando uma ferramenta como ERwin ou Rational Rose no Exercício de Laboratório 8.28. Utilizando o recurso de geração de esquema SQL da ferramenta de modelagem, gere o esquema SQL para um banco de dados Oracle.

- 9.14.** Considere o projeto EER para o banco de dados LEILAO_ONLINE que foi modelado usando uma ferramenta como ERwin ou Rational Rose no Exercício de Laboratório 8.29. Utilizando o recurso de geração de esquema SQL da ferramenta de modelagem, gere o esquema SQL para um banco de dados Oracle.

Bibliografia selecionada

O algoritmo de mapeamento ER para relacional original foi descrito no artigo clássico de Chen (1976), que apresentou o modelo ER original. Batini et al. (1992) discutem uma série de algoritmos de mapeamento de modelos ER e EER para modelos legados, e vice-versa.