

Modelagem de Dados

Parte 3

Os capítulos da Parte 3 abordam a modelagem de dados usando o Modelo Entidade–Relacionamento de modo a proporcionar as habilidades necessárias para o projeto conceitual de banco de dados. O Capítulo 5 apresenta a notação Pé-de-Galinha do Modelo Entidade–Relacionamento e explica as regras do diagrama para prevenir erros comuns no diagrama. O Capítulo 6 enfatiza a prática da modelagem de dados em problemas narrativos e apresenta as regras para converter Diagramas Entidade–Relacionamento (DERs) em tabelas relacionais. O Capítulo 6 explica as transformações de projeto e os erros comuns de projeto para aperfeiçoar as habilidades de modelagem de dados.

Capítulo 5. Entendendo os Diagramas Entidade–Relacionamento

Capítulo 6. Desenvolvendo Modelos de Dados para Bancos de Dados de Negócio

Entendendo os Diagramas Entidade–Relacionamento

Objetivos de Aprendizagem

Este capítulo explica a notação de diagramas entidade–relacionamento como um pré-requisito para o uso de diagramas entidade–relacionamento no processo de desenvolvimento de banco de dados. Ao final deste capítulo, o aluno deverá ter adquirido os seguintes conhecimentos e habilidades:

- Conhecer os símbolos e o vocabulário da notação Pé-de-Galinha para diagramas entidade–relacionamento.
- Usar os símbolos de cardinalidade para representar relacionamentos 1-1, 1-M e M-N.
- Comparar a notação Pé-de-Galinha com a representação de tabelas relacionais.
- Entender padrões de relacionamento importantes.
- Usar as hierarquias de generalização para representar tipos semelhantes de entidade.
- Detectar erros de notação em um diagrama entidade–relacionamento.
- Entender a representação de regras de negócio em um diagrama entidade–relacionamento.
- Ter uma noção da diversidade de notações para diagramas entidade–relacionamento.

Visão Geral

O Capítulo 2 trouxe uma ampla apresentação sobre o processo de desenvolvimento de banco de dados. Você aprendeu sobre o relacionamento entre o desenvolvimento de banco de dados e o desenvolvimento de sistemas de informação, as fases de desenvolvimento de banco de dados e os tipos de habilidades que necessita dominar. Este capítulo apresenta a notação de diagramas entidade–relacionamento a fim de fornecer uma base para o uso dos diagramas entidade–relacionamento no processo de desenvolvimento de banco de dados. Para ampliar suas habilidades para projetar um banco de dados, o Capítulo 6 descreve o processo de usar os diagramas entidade–relacionamento para desenvolver modelos de dados para bases de dados de negócio.

Para se tornar um bom modelador de dados, você necessita entender a notação nos diagramas entidade–relacionamento e aplicar a notação em problemas de maior complexidade.

Para ajudá-lo a dominar a notação, este capítulo apresenta os símbolos usados nos diagramas entidade–relacionamento e compara os diagramas entidade–relacionamento com os diagramas de banco de dados relacionais dos capítulos anteriores. Depois, o capítulo investiga mais a fundo os relacionamentos, a parte mais notável dos diagramas entidade–relacionamento. Você aprenderá sobre dependência de identificador, padrões de relacionamento e equivalência entre dois tipos de relacionamentos. Por fim, aprenderá como representar semelhanças entre entidades usando hierarquias de generalização.

Com o intuito de possibilitar um entendimento mais profundo da notação Pé-de-Galinha, serão apresentadas as regras de diagrama e a representação de regras de negócio. Para dar um foco organizacional nos diagramas entidade–relacionamento, este capítulo apresenta a representação informal e formal de regras de negócio em um diagrama entidade–relacionamento. A fim de ajudá-lo a usar a notação Pé-de-Galinha corretamente, este capítulo fala sobre consistência e regras de completude e explica seu uso no ER Assistant.

Devido à abundância de notações entidade–relacionamento, você pode não ter a oportunidade de usar a notação Pé-de-Galinha exatamente como mostramos nos Capítulos 5 e 6. Com a finalidade de prepará-lo para entender outras notações, o capítulo conclui com uma apresentação das variações de diagrama, incluindo a notação de Diagrama de Classe da Linguagem de Modelagem Unificada (UML), uma das alternativas mais populares ao Modelo Entidade–Relacionamento.

Este capítulo proporciona as habilidades básicas da modelagem de dados de modo a capacitá-lo a entender a notação de diagramas entidade–relacionamento. Para aplicar a modelagem de dados como parte do processo de desenvolvimento de banco de dados, você deve estudar o Capítulo 6 para desenvolver modelos de dados para bancos de dados de negócio. O Capítulo 6 enfatiza as habilidades em solução de problemas para gerar alternativas de projeto, mapear uma descrição de problema na forma de um diagrama entidade–relacionamento e justificar decisões de projeto. Com a base obtida em ambos os capítulos, você estará preparado para executar modelagem de dados em estudos de caso e bancos de dados para organizações de tamanho médio.

5.1 Introdução aos Diagramas Entidade–Relacionamento

Obter um entendimento inicial dos diagramas entidade–relacionamento (DERs) exige estudo cuidadoso. Esta seção introduz a notação Pé-de-Galinha para DERs, uma notação muito utilizada e apoiada por muitas ferramentas CASE. Para dar partida, esta seção começa com os símbolos básicos de tipos de entidade, relacionamentos e atributos. Depois, esta seção explica as cardinalidades e seu formato na notação Pé-de-Galinha. Esta seção termina comparando a notação Pé-de-Galinha aos diagramas de banco de dados relacionais. Se você está estudando modelagem de dados antes dos bancos de dados relacionais, pode optar por pular a última parte desta seção.

tipo de entidade

um conjunto de entidades (pessoas, lugares, eventos ou coisas) de interesse representado por um retângulo em um diagrama entidade–relacionamento.

atributo

uma propriedade de um tipo de entidade ou relacionamento. Cada atributo tem um tipo de dado definindo o tipo de valores e operações permitidas no atributo.

relacionamento

uma associação nomeada entre tipos de entidade. Um relacionamento representa uma associação de mão dupla ou bidirecional entre entidades. A maioria dos relacionamentos envolve dois tipos distintos de entidade.

5.1.1 Símbolos Básicos

Os DERs têm três elementos básicos: tipos de entidade, relacionamentos e atributos. Os tipos de entidade são conjuntos de coisas de interesse (entidades) em uma aplicação. Os tipos de entidade representam grupos de coisas físicas, tais como livros, pessoas e lugares, e também eventos, como, por exemplo, pagamentos. Uma entidade é membro ou instância de um tipo de entidade. As entidades são identificadas de forma única para permitir seu rastreamento por todos os processos de negócio. Por exemplo, os clientes têm uma identificação única para possibilitar o processamento dos procedimentos de pedidos, remessas e garantia dos produtos. Na notação Pé-de-Galinha assim como na maioria das outras notações, retângulos denotam tipos de entidade. Na Figura 5.1, o tipo de entidade *Curso* representa o conjunto de cursos no banco de dados.

Os atributos são propriedades dos tipos de entidade ou relacionamentos. Um tipo de entidade deve ter uma chave primária assim como outros atributos descritivos. Os atributos são mostrados dentro de um retângulo de tipo de entidade. Se há muitos atributos, os atributos podem ser omitidos e listados em uma página separada. Algumas ferramentas de projeto de DER mostram os atributos em uma exibição em zoom, separadamente do resto do diagrama. Sublinhados indicam que o(s) atributo(s) funciona(m) como a chave primária do tipo de entidade.

Os relacionamentos são associações nomeadas entre tipos de entidade. Na notação Pé-de-Galinha, os nomes dos relacionamentos aparecem em linhas ligando os tipos de entidade envolvidos no relacionamento. Na Figura 5.1, o relacionamento *Tem* mostra que os tipos de

FIGURA 5.1
Símbolos Básicos de
Ilustração em um
Diagrama Entidade-
Relacionamento

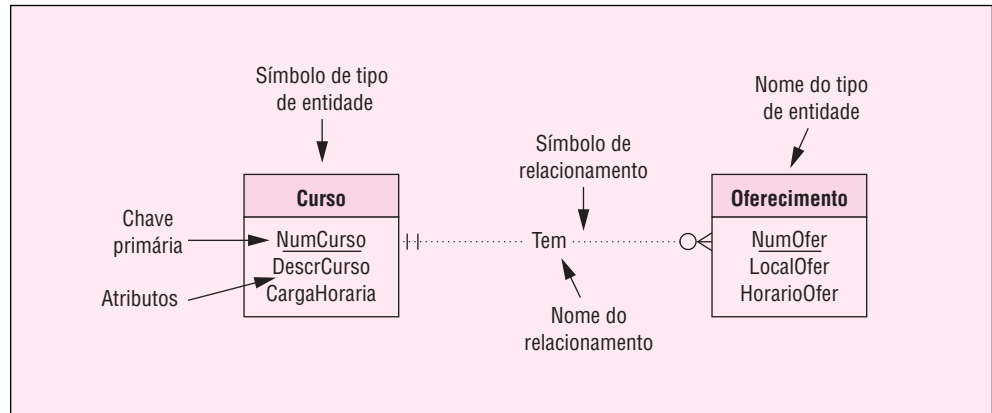
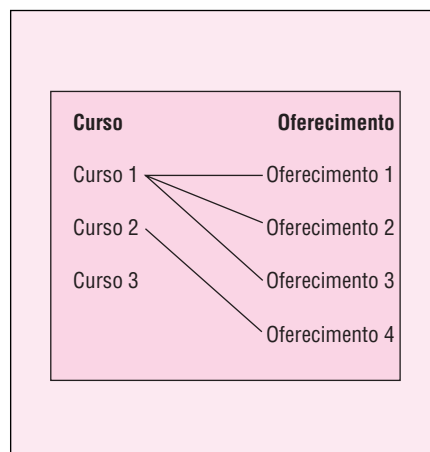


FIGURA 5.2
Diagrama de
Instâncias para o
Relacionamento *Tem*



entidade *Curso* e *Oferecimento* estão diretamente relacionados. Os relacionamentos armazenam as associações em ambas as direções. Por exemplo, o relacionamento *Tem* mostra os oferecimentos para um dado curso e o curso associado a um determinado oferecimento. O relacionamento *Tem* é binário porque envolve dois tipos de entidade. A Seção 5.2 traz exemplos de relacionamentos mais complexos envolvendo só um tipo distinto de entidade (auto-relacionamentos) e mais de dois tipos de entidade (relacionamentos N-ários).

Em um sentido amplo, os DERs têm uma correspondência natural com a linguagem. Os tipos de entidade podem corresponder a substantivos e os relacionamentos a verbos ou frases prepositivas ligando substantivos. Neste sentido, pode-se ler um diagrama entidade-relacionamento como um conjunto de frases. Por exemplo, o DER da Figura 5.1 pode ser lido como “o curso tem oferecimentos”. Note que há uma direção subentendida em cada relacionamento. Na outra direção, pode-se escrever “um oferecimento foi criado para um curso”. Por praticidade, é uma boa idéia usar verbos na voz ativa em vez da passiva para descrever relacionamentos. Portanto, *Tem* é mais indicado como nome de relacionamento. Você deve usar a correspondência natural com a linguagem mais como um guia do que como uma regra estrita. Para grandes DERs, você nem sempre encontrará uma boa correspondência natural com a linguagem para todas as partes dos diagramas.

5.1.2 Cardinalidade do Relacionamento

As cardinalidades restringem o número de objetos que participam em um relacionamento. Para demonstrar o significado das cardinalidades, um diagrama de instâncias é útil. A Figura 5.2 mostra um conjunto de cursos ({Curso1, Curso2, Curso3}), um conjunto de oferecimentos ({Oferecimento1, Oferecimento2, Oferecimento3, Oferecimento4}) e conexões entre os dois conjuntos. Na Figura 5.2, o Curso1 está relacionado a Oferecimento1, Oferecimento2 e

cardinalidade

uma restrição sobre o número de entidades participantes em um relacionamento. Em um DER, as cardinalidades mínima e máxima são especificadas para ambas as direções de um relacionamento.

Oferecimento3, o Curso2 está relacionado a Oferecimento4 e o Curso3 não está relacionado a nenhuma entidade de *Oferecimento*. Da mesma forma, Oferecimento1 está relacionado a Curso1, Oferecimento2 está relacionado a Curso1, Oferecimento3 está relacionado a Curso1 e Oferecimento4 está relacionado a Curso2. Desse diagrama de instâncias, podemos concluir que cada oferecimento está relacionado exatamente a um curso. Na outra direção, cada curso está relacionado a zero ou mais oferecimentos.

Representação Pé-de-Galinha das Cardinalidades

A notação Pé-de-Galinha usa três símbolos para representar cardinalidades. O símbolo Pé-de-Galinha (ou seja, duas linhas anguladas e uma linha reta) denota muitas (zero ou mais) entidades relacionadas. Na Figura 5.3, o símbolo Pé-de-Galinha perto do tipo de entidade *Oferecimento* significa que um curso pode estar relacionado a muitos oferecimentos. O círculo significa uma cardinalidade zero, enquanto uma linha perpendicular à linha de relacionamento quer dizer cardinalidade um.

Para representar cardinalidades mínima e máxima, os símbolos de cardinalidade são colocados junto a cada tipo de entidade em um relacionamento. O símbolo mínimo de cardinalidade aparece do lado do nome do relacionamento enquanto o símbolo máximo de cardinalidade aparece do lado do tipo de entidade. Na Figura 5.3, um curso está relacionado a um mínimo de zero oferecimento (círculo na posição interna) e a um máximo de muitos oferecimentos (Pé-de-Galinha na posição externa). De forma semelhante, um oferecimento está relacionado a exatamente um (um e único) curso como mostrado pelas únicas linhas na vertical em ambas as posições interna e externa.

Classificação de Cardinalidades

As cardinalidades são classificadas por valores comuns para as cardinalidades mínima e máxima. A Tabela 5.1 traz duas classificações para cardinalidades mínimas. Uma cardinalidade mínima de um ou mais indica um relacionamento obrigatório. Por exemplo, a participação no relacionamento *Tem* é obrigatória para cada entidade *Oferecimento* devido à cardinalidade mínima um. Um relacionamento obrigatório torna o tipo de entidade dependente da existência

dependência de existência
uma entidade que não pode existir a menos que outra entidade relacionada exista.
Um relacionamento obrigatório cria uma dependência de existência.

FIGURA 5.3
Diagrama Entidade-Relacionamento com as Notações de Cardinalidades

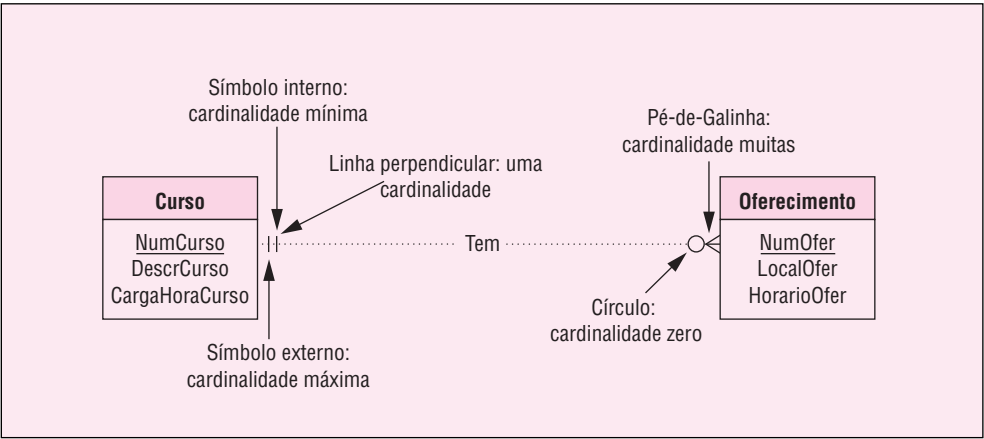


TABELA 5.1
Resumo das Classificações de Cardinalidade

Classificação	Restrições de Cardinalidade
Obrigatória	Cardinalidade mínima ≥ 1
Opcional	Cardinalidade mínima = 0
Funcional ou de valor único	Cardinalidade máxima = 1
1-M	Cardinalidade máxima = 1 em uma direção e cardinalidade máxima > 1 na outra direção
M-N	Cardinalidade máxima ≥ 1 nas duas direções
1-1	Cardinalidade máxima = 1 nas duas direções

FIGURA 5.4
Relacionamento Opcional
para Ambos os Tipos de
Entidade

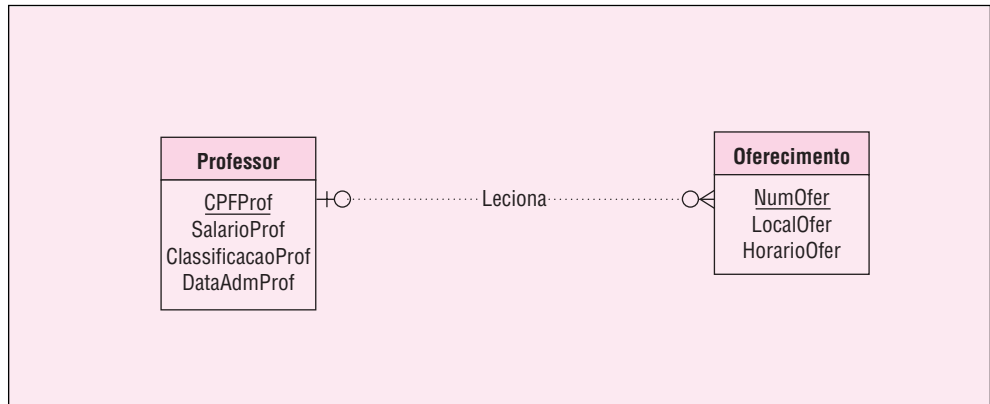
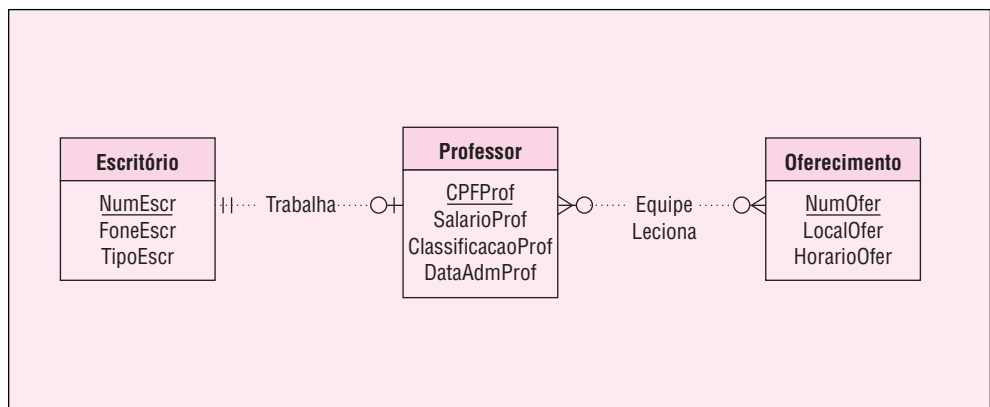


FIGURA 5.5
Exemplos de
Relacionamentos M-N e 1-1



do relacionamento. O tipo de entidade *Oferecimento* depende do relacionamento *Tem* porque uma entidade de *Oferecimento* não pode ser armazenada sem uma entidade *Curso* relacionada. Em contraste, uma cardinalidade mínima zero indica um relacionamento opcional. Por exemplo, o relacionamento *Tem* é opcional ao tipo de entidade *Curso* porque uma entidade *Curso* pode ser armazenada sem estar relacionada a uma entidade *Oferecimento*. A Figura 5.4 mostra que o relacionamento *Leciona* é opcional para ambos os tipos de entidade.

A Tabela 5.1 também mostra várias classificações para cardinalidades máximas. Uma cardinalidade máxima um significa que o relacionamento é de valor único (monovalorado) ou funcional. Por exemplo, os relacionamentos *Tem* e *Leciona* são funcionais para *Oferecimento* porque uma entidade *Oferecimento* pode estar relacionada a no máximo um *Curso* e uma entidade *Professor*. A palavra *função* vem da matemática onde uma função dá um valor. Um relacionamento que tem uma cardinalidade máxima um em uma direção e mais de um (muitos) na outra direção é chamado de relacionamento 1-M (lê-se um-para-muitos ou um-para-vários). Ambos os relacionamentos *Tem* e *Leciona* são 1-M.

Da mesma forma, um relacionamento que tem uma cardinalidade máxima de mais de um em ambas as direções é conhecido como um relacionamento M-N (muitos-para-muitos). Na Figura 5.5, o relacionamento *Equipe Leciona* permite que múltiplos professores em conjunto ensinem o mesmo oferecimento de curso, como mostramos no diagrama de instâncias da Figura 5.6. Os relacionamentos M-N são comuns em bases de dados de negócio para representar a conexão entre peças e fornecedores, autores e livros e habilidades e funcionários. Por exemplo, uma peça pode ser fornecida por muitos fornecedores e um fornecedor pode fornecer muitas peças.

Menos comuns são os relacionamentos 1-1 em que a cardinalidade máxima é igual a um em ambas as direções. Por exemplo, o relacionamento *Trabalha* na Figura 5.5 permite que um professor seja designado para um escritório e que um escritório seja ocupado por no máximo um professor.

FIGURA 5.6
Diagrama de Instâncias para
o Relacionamento M-N
Equipe Leciona

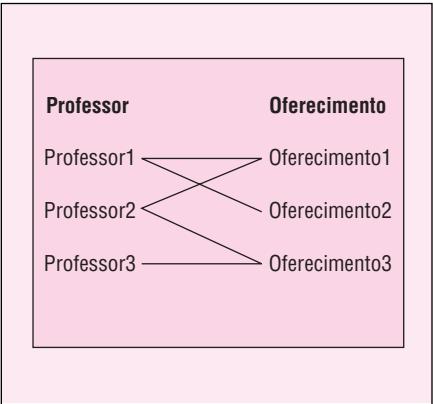
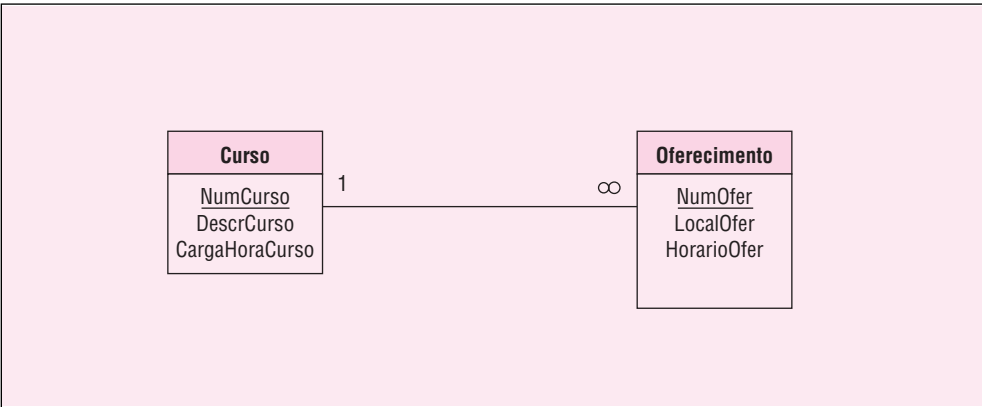


FIGURA 5.7
Banco de Dados Relacional
para o Exemplo
Curso-Oferecimento



5.1.3 Comparação com Diagramas de Banco de Dados Relacionais

Para encerrar esta seção, vamos comparar a notação na Figura 5.3 com os diagramas de banco de dados relacionais (do Microsoft Access) com o qual você está mais familiarizado. É fácil confundir-se entre as duas notações. Algumas diferenças importantes estão listadas a seguir¹. Para ajudá-lo a visualizar as diferenças, a Figura 5.7 mostra um diagrama de banco de dados relacional para o exemplo *Curso-Oferecimento*.

1. Os diagramas de banco de dados relacionais não usam nomes para relacionamentos. Em vez disso, chaves estrangeiras representam os relacionamentos. A notação DER não usa chaves estrangeiras. Por exemplo, *NumOferCurso* é uma coluna na Figura 5.7 mas não é um atributo na Figura 5.3.
2. Os diagramas de banco de dados relacionais mostram somente cardinalidades máximas.
3. Algumas notações DER (inclusive a notação Pé-de-Galinha) permitem que tanto os tipos de entidade quanto os relacionamentos possuam atributos. Os diagramas de banco de dados relacionais só permitem que as tabelas tenham colunas
4. Os diagramas de banco de dados relacionais permitem um relacionamento entre duas tabelas. Algumas notações DER (embora não a notação Pé-de-Galinha) permitem relacionamentos N-ários envolvendo mais de dois tipos de entidade. A próxima seção mostra como representar relacionamentos N-ários na notação de Pé-de-Galinha.
5. Em algumas notações DER (embora não na notação Pé-de-Galinha), a posição das cardinalidades é invertida.

¹ O Capítulo 6 apresenta as regras de conversão que descrevem as diferenças com mais precisão.

5.2 Entendendo Relacionamentos

Esta seção explora a notação entidade-relacionamento em mais profundidade ao examinar aspectos importantes dos relacionamentos. A primeira subseção descreve a dependência de identificador, um tipo especializado de dependência de existência. A segunda subseção descreve três padrões importantes de relacionamento: (1) relacionamentos com atributos, (2) auto-relacionamentos e (3) tipos de entidade associativa representando relacionamentos N-ários. A subseção final descreve uma equivalência importante entre relacionamentos M-N e 1-M.

5.2.1 Dependência de Identificador (Entidades Fracas e Relacionamentos Identificadores)

Em um DER, alguns tipos de entidade não podem ter sua própria chave primária. O tipo de entidade sem sua própria chave primária deve tomar emprestada parte (ou toda) sua chave primária de outros tipos de entidade. Os tipos de entidade que tomam emprestado parte de sua chave primária ou sua chave primária inteira são conhecidos como entidades fracas. O relacionamento que fornece componentes da chave primária é conhecido como relacionamento identificador. Assim, uma dependência de identificador envolve uma entidade fraca e um ou mais relacionamentos identificadores.

entidade fraca

um tipo de entidade que toma emprestada toda ou parte de sua chave primária de outro tipo de entidade. Relacionamentos identificadores indicam os tipos de entidade que fornecem componentes da chave primária emprestada.

A dependência de identificador ocorre porque algumas entidades estão fortemente associadas a outras entidades. Por exemplo, uma sala não tem uma identidade separada de seu edifício porque está contida fisicamente dentro do edifício. Você pode se referir a uma sala apenas fornecendo seu identificador de edifício associado. No DER para edifícios e salas (Figura 5.8), o tipo de entidade *Sala* é dependente do identificador do tipo de entidade *Edifício* no relacionamento *Contém*. Uma linha de relacionamento contínua indica um relacionamento identificador. Para entidades fracas, o atributo sublinhado (se presente) é parte da chave primária, mas não a chave primária inteira. Assim, a chave primária de *Sala* é uma combinação de *IDEdifício* e *NumSala*. Como outro exemplo, a Figura 5.9 retrata uma dependência de identificador envolvendo a entidade fraca *UF* e o relacionamento identificador *Contém*.

A dependência de identificador é um tipo especializado de dependência de existência. Lembre-se de que um tipo de entidade dependente de existência tem um relacionamento obrigatório (cardinalidade mínima um). Entidades fracas são dependentes de existência dos relacionamentos identificadores. Além da dependência de existência, uma entidade fraca toma emprestada pelo menos uma parte de sua chave primária inteira. Por causa da dependência de existência e do empréstimo da chave primária, as cardinalidades mínima e máxima de uma entidade fraca são sempre 1.

FIGURA 5.8
Exemplo de Dependência de Identificador

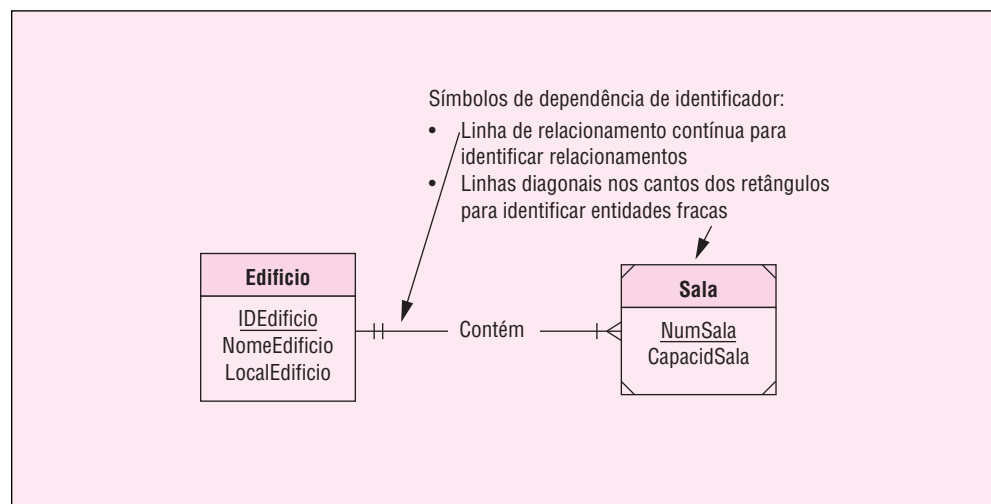


FIGURA 5.9
Outro Exemplo de
Dependência de Identificador

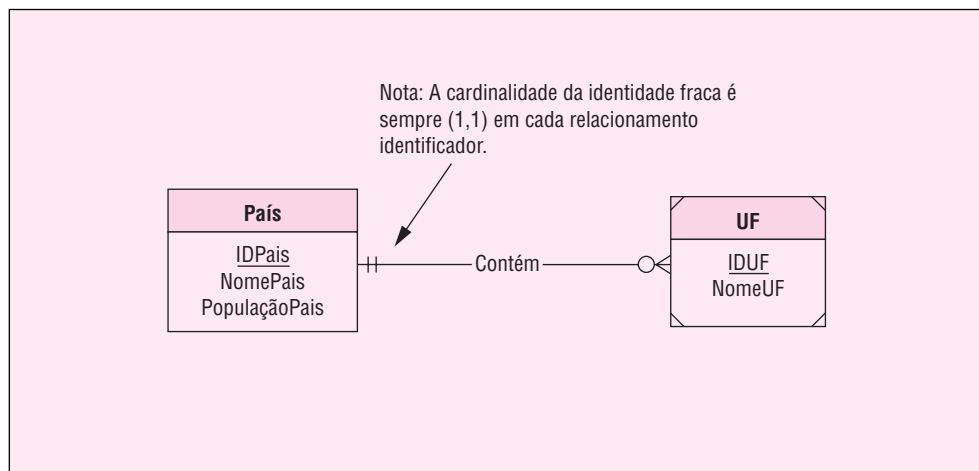
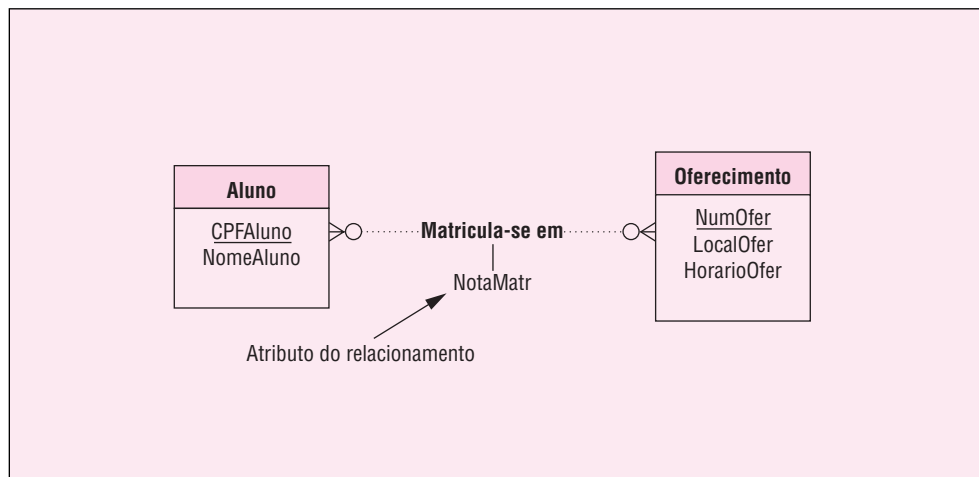


FIGURA 5.10
Relacionamento M-N com
um Atributo



A próxima seção mostra vários exemplos adicionais de dependência de identificador na discussão sobre tipos de entidades associativas e relacionamentos N-ários. O uso da dependência de identificador é necessário para tipos de entidades associativas.

5.2.2 Padrões de Relacionamento

Esta seção discute três padrões de relacionamentos que você pode encontrar nos esforços de desenvolvimento de banco de dados: (1) relacionamentos M-N com atributos, (2) auto-relacionamentos (unários) e (3) tipos de entidades associativas representando relacionamentos N-ários. Embora tais padrões de relacionamento não predominem nos DERs, eles são importantes quando ocorrem. Você precisa estudar esses padrões cuidadosamente a fim de aplicá-los corretamente no desenvolvimento de banco de dados.

Relacionamentos M-N com Atributos

Como mencionado resumidamente na Seção 5.1, os relacionamentos podem ter atributos. Esta situação ocorre em geral com relacionamentos M-N. Em um relacionamento M-N, atributos estão associados à combinação de tipos de entidade, não só a um dos tipos de entidade. Se um atributo é associado a um só tipo de entidade, então ele deve fazer parte desse tipo de entidade, não do relacionamento. As figuras 5.10 e 5.11 representam relacionamentos M-N com atributos. Na Figura 5.10, o atributo *NotaMatr* está associado à combinação de um aluno e um oferecimento, não a qualquer um dos dois sozinhos. Por

FIGURA 5.11
Mais Relacionamentos M-N
com Atributos

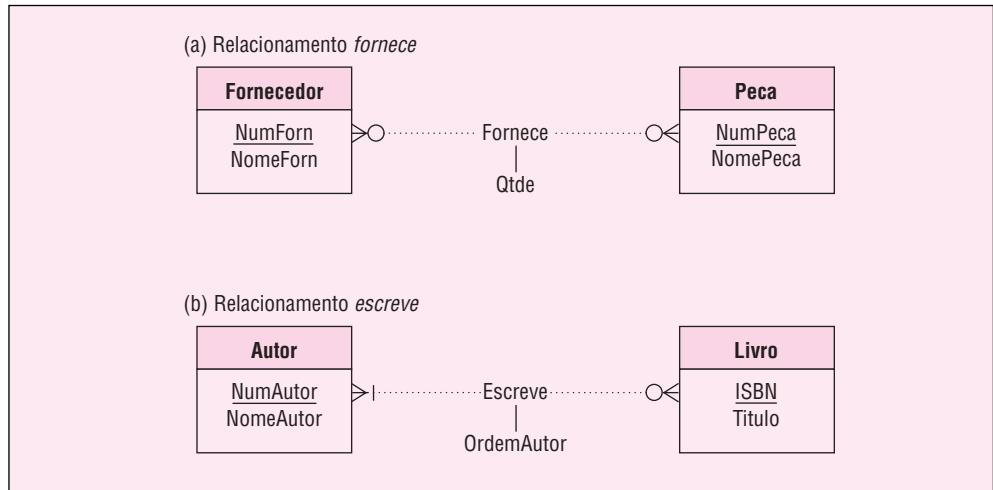
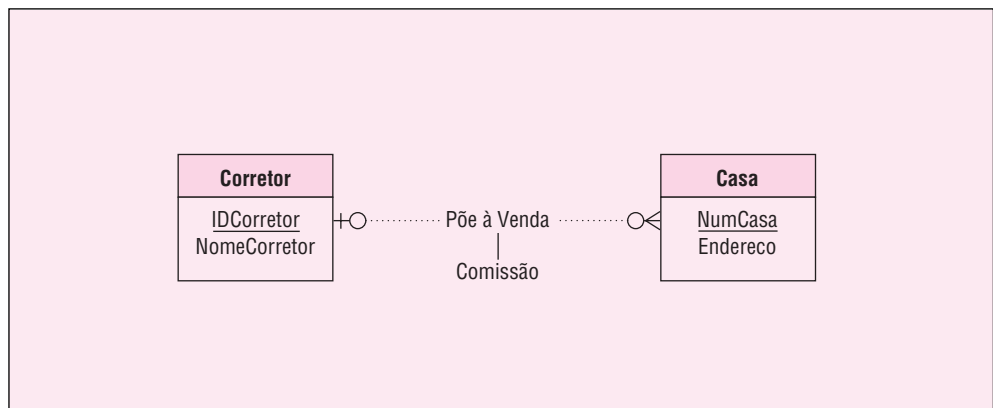


FIGURA 5.12
Relacionamento 1-M com
um Atributo



exemplo, o relacionamento *Matricula-se em* registra o fato de que o estudante com o CPF número 123-77-9993 tem uma nota 3,5 no curso oferecido com o número de oferecimento 1256. Na Figura 5.11(a), o atributo *Qtde* representa a quantidade de uma peça fornecida por um dado fornecedor. Na Figura 5.11(b), o atributo *OrdemAutor* representa a ordem em que o nome do autor aparece no título de um livro. Para reduzir o excesso de informações em um diagrama grande, os atributos de relacionamento podem não ser mostrados.

Os relacionamentos 1-M também podem ter atributos, mas relacionamentos 1-M com atributos são muito menos comuns que relacionamentos M-N com atributos. Na Figura 5.12, o atributo *Comissão* está associado ao relacionamento *Põe à Venda*, não apenas com um *Corretor* ou com o tipo de entidade *Casa*. Uma casa só terá uma comissão se um corretor a puser à venda. Tipicamente, relacionamentos 1-M com atributos são opcionais para o tipo de entidade filha. O relacionamento *Põe à Venda* é opcional para o tipo de entidade *Casa*.

Auto-Relacionamentos (Unários)

Um auto-relacionamento (unário) envolve conexões entre membros de um mesmo conjunto. Os auto-relacionamentos às vezes são chamados de relacionamentos reflexivos porque são como um reflexo em um espelho. A Figura 5.13 mostra dois auto-relacionamentos envolvendo os tipos de entidade *Professor* e *Curso*. Ambos os relacionamentos envolvem dois tipos de entidade que são os mesmos (*Professor* para *Supervisiona* e *Curso* para *PreReqPara*). Estes relacionamentos representam conceitos importantes em um banco de dados de uma universidade. O relacionamento *Supervisiona* denota um organograma, enquanto o relacionamento *PreReqPara* denota as dependências de curso que podem afetar o planejamento do curso de um aluno.

auto-relacionamento

um relacionamento que envolve o mesmo tipo de entidade. Auto-relacionamentos representam as associações entre membros do mesmo conjunto.

FIGURA 5.13
Exemplos de
Auto-
relacionamentos
(Unários)

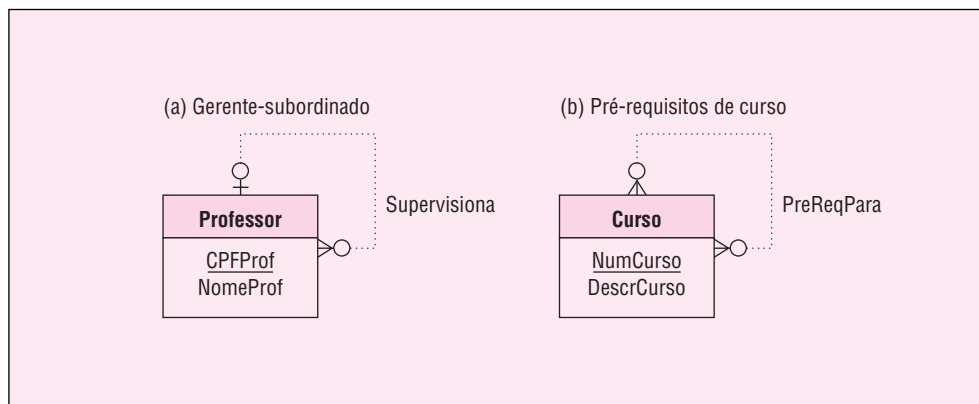
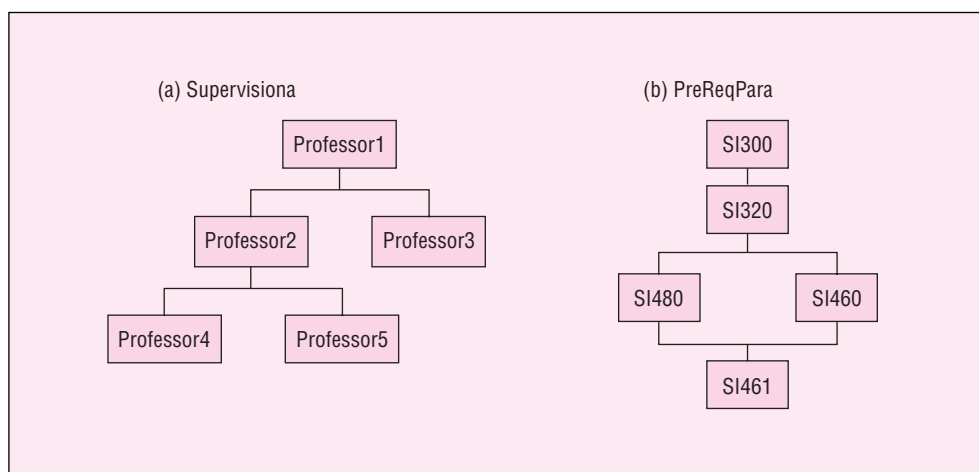


FIGURA 5.14
Diagramas de
Instâncias para Auto-
relacionamentos



Nos auto-relacionamentos, é importante distinguir entre relacionamentos M-N e relacionamentos 1-M. Um diagrama de instâncias pode ajudá-lo a entender a diferença. A Figura 5.14(a) mostra um diagrama de instâncias para o relacionamento *Supervisiona*. Note que cada professor pode ter no máximo um superior. Por exemplo, Professor2 e Professor3 têm Professor1 como seu superior. Portanto, *Supervisiona* é um relacionamento 1-M porque cada professor pode ter no máximo um supervisor. Por sua vez, não há nenhuma restrição similar no diagrama de instâncias para o relacionamento *PreReqPara* [Figura 5.14(b)]. Por exemplo, tanto SI480 como SI460 são pré-requisitos para SI461. Portanto, *PreReqPara* é um relacionamento M-N porque um curso pode ser um pré-requisito para muitos cursos e um curso pode ter muitos pré-requisitos.

Os auto-relacionamentos ocorrem em diversas situações de negócio. Quaisquer dados que podem ser visualizados como na Figura 5.14 podem ser representados como um auto-relacionamento. Exemplos típicos incluem mapas hierárquicos de contas, mapas genealógicos, projetos de peças e rotas de transporte. Nestes exemplos, os auto-relacionamentos são uma importante parte do banco de dados.

Há um outro aspecto nos auto-relacionamentos digno de observação. Às vezes um auto-relacionamento não é necessário. Por exemplo, se você só quer saber se um funcionário é supervisor, não precisa de um auto-relacionamento. Em vez disso, um atributo pode ser usado para indicar que um funcionário é supervisor.

Tipos de Entidades Associativas Representando Relacionamentos N-ários

Algumas notações de DER provêm suporte a relacionamentos envolvendo mais de dois tipos de entidade conhecidos como relacionamentos N-ários (multicaminhos) onde o M significa

FIGURA 5.15
Relacionamento N-ários
(Ternário) Usando a Notação
Chen

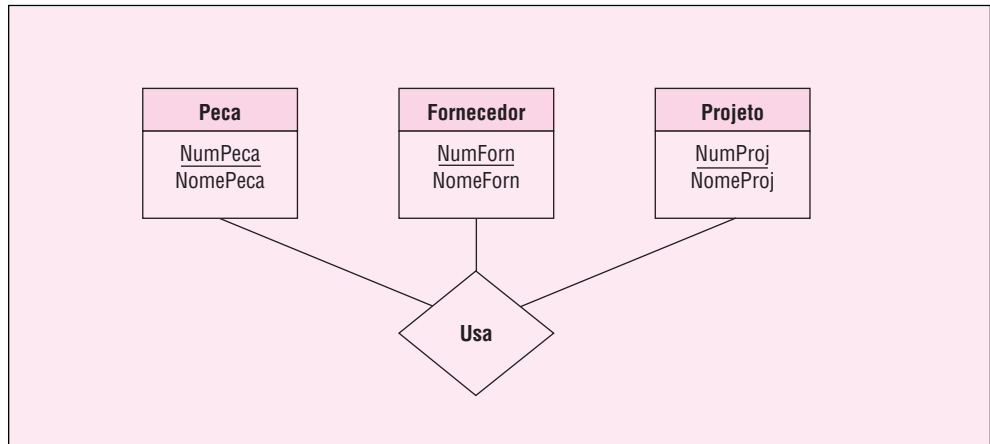
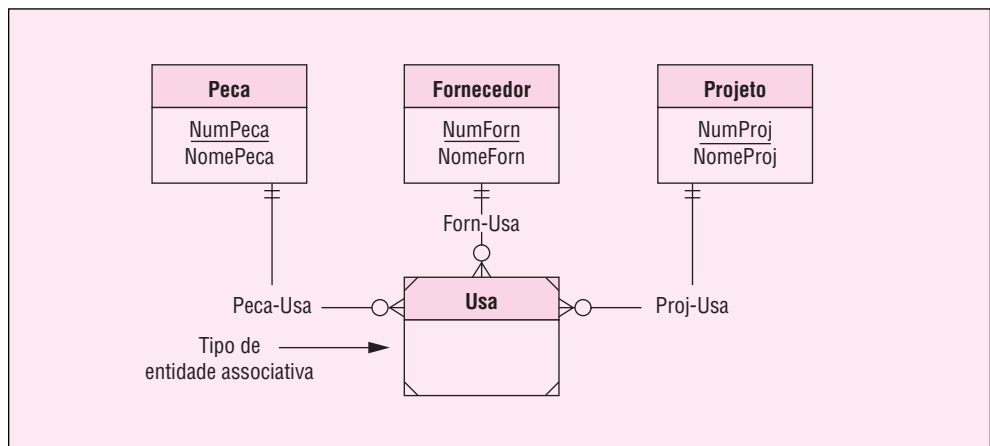


FIGURA 5.16
Tipo de Entidade Associativa
para Representar um
Relacionamento Ternário



mais de dois. Por exemplo, a notação DER Chen (com losangos para representar relacionamentos) permite que os relacionamentos se liguem a mais de dois tipos de entidade, como demonstrado na Figura 5.15². O relacionamento *Usa* lista os fornecedores e as peças usadas nos projetos. Por exemplo, uma instância de relacionamento envolvendo Fornecedor1, Peça1 e Projeto1 indica Fornecedor1 e Peça1 no Projeto1. Um relacionamento N-ário que envolve três tipos de entidade é chamado de relacionamento ternário.

Embora não possa representar diretamente relacionamentos N-ários na notação Pé-de-Galinha, você deve entender como representá-los indiretamente. Você usa um tipo de entidade associativa e um grupo de relacionamentos identificadores 1-M para representar um relacionamento N-ário. Na Figura 5.16, três relacionamentos 1-M se ligam ao tipo de entidade associativa *Usa*, aos tipos de entidade *Peca*, *Fornecedor* e *Projeto*. O tipo de entidade *Usa* é associativa porque seu papel é ligar outros tipos de entidade. Como os tipos de entidade associativa têm um papel de conexão, seus nomes são às vezes verbos ativos. Além disso, tipos de entidades associativas são sempre fracos já que devem tomar emprestado toda a sua chave primária. Por exemplo, o tipo de entidade *Usa* obtém sua chave primária por meio dos três relacionamentos identificadores.

Como outro exemplo, a Figura 5.17 mostra o tipo de entidade associativa *Fornece* que liga tipos de entidades *Funcionario*, *Habilidade* e *Projeto*. Um exemplo de instância do tipo de entidade *Fornece* contém Funcionario1 fornecendo a Habilidade1 no Projeto1.

tipo de entidade associativa

uma entidade fraca que depende de dois ou mais tipos de entidades para sua chave primária. Um tipo de entidade associativa com mais de dois relacionamentos identificadores é conhecido como um tipo de entidade associativa N-ário.

² A notação Chen é assim chamada devido ao Dr. Peter Chen, autor do artigo que definiu o Modelo Entidade-Relacionamento em 1976.

FIGURA 5.17
Tipo de Entidade Associativa
Ligando *Funcionario*,
Habilidade* e *Projeto

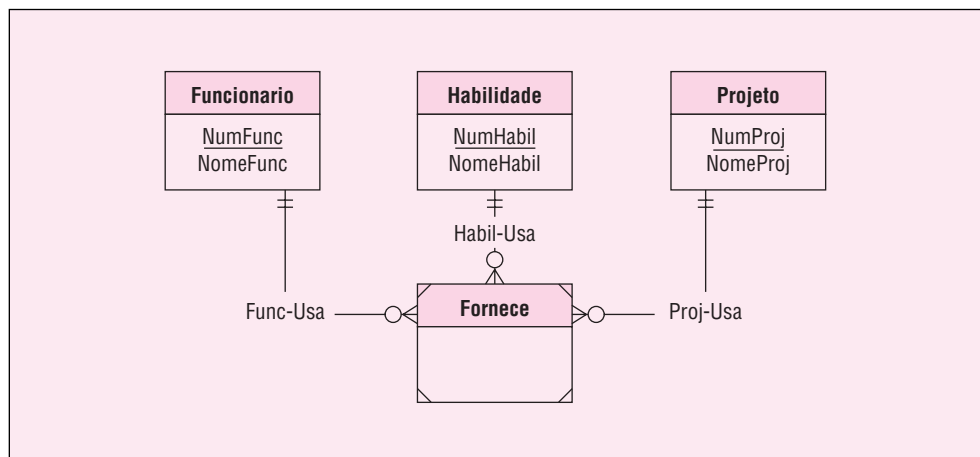
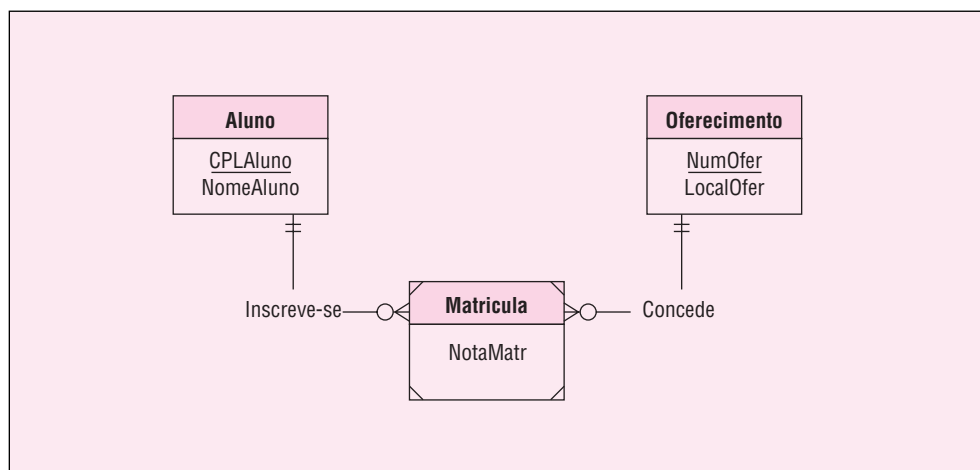


FIGURA 5.18
Relacionamento M-N
***Matricula-se em* (Figura 5.10)**
Transformado em
Relacionamentos 1-M



A questão de quando usar uma entidade associativa N-ária (isto é, um tipo de entidade associativa representando um relacionamento N-ário) pode ser difícil de entender. Se um banco de dados só tem necessidade de registrar fatos em pares, uma entidade associativa N-ária não é necessária. Por exemplo, se um banco de dados só precisa registrar quem fornece uma peça e que projeto usa uma peça, então o tipo de entidade associativa N-ário não deve ser usado. Neste caso, deve haver relacionamentos binários entre *Fornecedor* e *Peça* e entre *Projeto* e *Peça*. Você deve usar um tipo de entidade associativa N-ário quando o banco de dados deve registrar combinações de três (ou mais) entidades em vez de somente combinações de duas entidades. Por exemplo, se um banco de dados precisa registrar qual fornecedor fornece peças em projetos específicos, um tipo de entidade associativa N-ário é necessário. Devido à complexidade dos relacionamentos N-ários, o Capítulo 7 traz uma forma de pensar neles usando restrições, enquanto o Capítulo 12 traz uma forma de pensar neles usando formulários de entrada de dados.

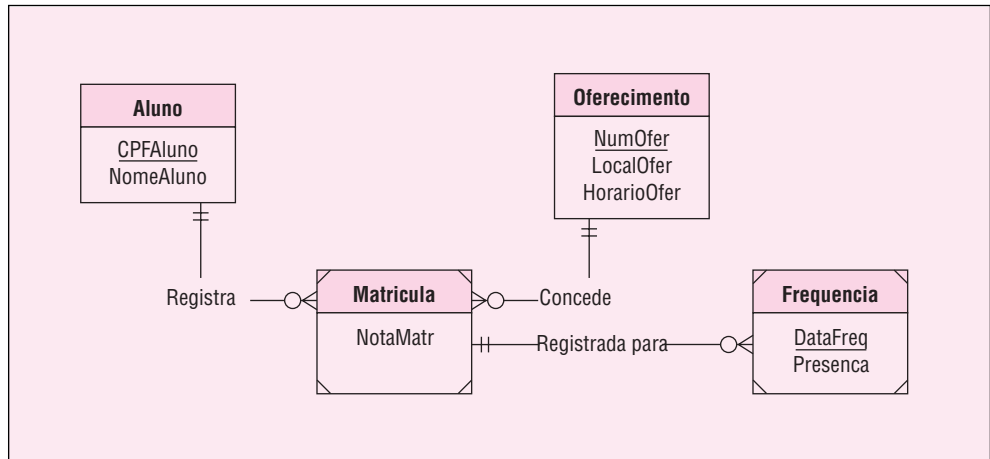
equivalência de relacionamento

um relacionamento M-N pode ser substituído por um tipo de entidade associativa e dois relacionamentos identificadores 1-M. Na maioria dos casos, a escolha entre um relacionamento M-N e o tipo de entidade associativa é uma preferência pessoal.

5.2.3 Equivalência entre Relacionamentos 1-M e M-N

Para melhorar seu entendimento sobre os relacionamentos M-N, você deve conhecer uma equivalência importante para os relacionamentos M-N. Um relacionamento M-N pode ser substituído por um tipo de entidade associativa e dois relacionamentos identificadores 1-M. A Figura 5.18 mostra o relacionamento *Matricula-se em* (Figura 5.10) convertido nesse estilo 1-M. Na Figura 5.18, dois relacionamentos identificadores e um tipo de entidade associativa substituem o relacionamento de *Matricula-se em*. O nome do relacionamento (*Matricula-se em*) foi mudado para um substantivo (*Matricula*) para seguir a convenção do uso de substan-

FIGURA 5.19
Tipo de Entidade Frequência
 Adicionado ao DER da
 Figura 5.18



tivos para nomes de tipo de entidade. O estilo 1-M é semelhante à representação em um diagrama de banco de dados relacional. Se você se sente mais confortável com o estilo 1-M, então use-o. Em termos de DER, os estilos M-N e 1-M têm o mesmo significado.

A transformação de um relacionamento M-N em relacionamentos 1-M é semelhante a representar um relacionamento N-ário usando relacionamentos 1-M. Sempre que um relacionamento M-N é representado como um tipo de entidade associativa e dois relacionamentos 1-M, o novo tipo de entidade é dependente de identificador em ambos os relacionamentos 1-M, como mostrado na Figura 5.18. Da mesma forma, ao representar relacionamentos N-ários, o tipo de entidade associativa é dependente de identificador em todos os relacionamentos 1-M como nas Figuras 5.16 e 5.17.

Há uma situação quando se dá preferência ao estilo 1-M e não ao estilo M-N. Quando um relacionamento M-N deve estar relacionado a outros tipos de entidade em relacionamentos, você deve usar o estilo 1-M. Por exemplo, suponha que além da matrícula em um oferecimento de curso, deve ser registrada a frequência em cada aula. Nesta situação, o estilo 1-M é mais recomendado porque é necessário ligar uma matrícula aos registros de presença. A Figura 5.19 mostra o tipo de entidade *Frequencia* adicionado ao DER da Figura 5.18. Note que um relacionamento M-N entre os tipos de entidade *Aluno* e *Oferecimento* não teria permitido outro relacionamento com *Frequencia*.

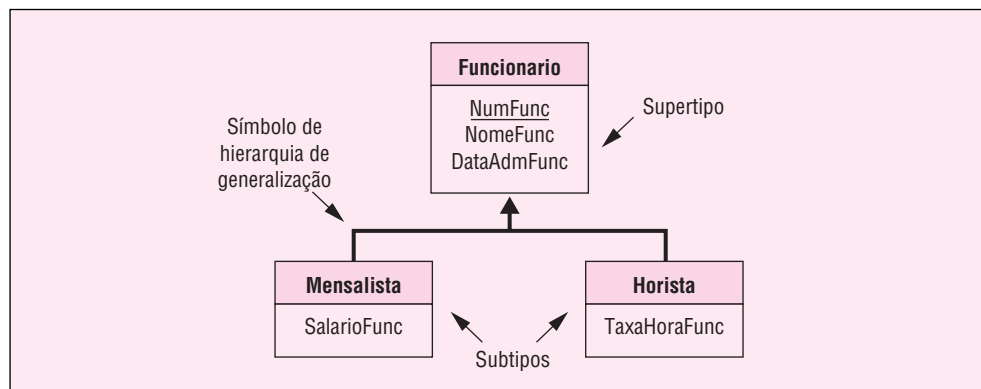
A Figura 5.19 fornece outros exemplos de dependência de identificador. *Frequencia* é dependente de identificação de *Matricula* no relacionamento *Registrada para*. A chave primária de *Frequencia* consiste em *DataFreq* junto com a chave primária de *Matricula*. Semelhantemente, *Matricula* é dependente de identificação tanto de *Aluno* como *Oferecimento*. A chave primária de *Matricula* é uma combinação de *CPFAluno* e *NumO*.

5.3 Classificação no Modelo Entidade–Relacionamento

As pessoas classificam entidades para entender melhor seu ambiente. Por exemplo, animais são classificados em mamíferos, répteis e outras categorias a fim de compreender semelhanças e diferenças entre espécies diferentes. Nos negócios, o ato de classificar também está disseminado. Podemos aplicar classificações a investimentos, funcionários, clientes, empréstimos, peças e assim por diante. Por exemplo, quando se solicita um financiamento imobiliário, uma distinção importante está entre financiamentos com taxas prefixadas ou pós-fixadas. Dentro de cada tipo de financiamento, há muitas variações distinguidas por características tal como o prazo de financiamento, as multas e o valor financiado.

Essa seção descreve a notação DER que dá suporte a classificações. Você aprenderá a usar hierarquias de generalização, restrições específicas de cardinalidade para hierarquias de generalização e usar hierarquias de generalização de múltiplos níveis para classificações complexas.

FIGURA 5.20
Hierarquia de Generalização
para Funcionários



hierarquia de generalização

uma coleção de tipos de entidades organizados em uma estrutura hierárquica para mostrar semelhança nos atributos. Cada subtipo ou tipo de entidade filha representa um subconjunto de seu supertipo ou tipo de entidade pai.

herança

uma funcionalidade de modelagem de dados que provê suporte ao compartilhamento de atributos entre um supertipo e um subtipo. Os subtipos herdam os atributos de seus supertipos.

5.3.1 Hierarquias de Generalização

Hierarquias de generalização permitem que tipos de entidade estejam relacionados pelo nível de especialização. A Figura 5.20 retrata uma hierarquia de generalização para classificar funcionários como mensalistas ou horistas. Funcionários tanto mensalistas e horistas são tipos especializados de funcionários. O tipo de entidade *Funcionario* é conhecido como o supertipo (ou pai). Os tipos de entidade *Mensalista* e *Horista* são conhecidos como os subtipos (ou filhos). Como cada entidade de subtipo é uma entidade do supertipo, o relacionamento entre um subtipo e supertipo é conhecido como ISA (em inglês, “*is a*” significa “é um”). Por exemplo, um funcionário mensalista é um funcionário. Já que o nome do relacionamento (ISA) é sempre o mesmo, ele não é mostrado no diagrama.

A herança suporta o compartilhamento entre um supertipo e seus subtipos. Porque cada entidade de subtipo também é uma entidade de supertipo, os atributos do supertipo também se aplicam a todos os subtipos. Por exemplo, cada entidade *Mensalista* tem um número de funcionário, nome e data de admissão porque é também uma entidade de *Funcionario*. Herança significa que os atributos de um supertipo são automaticamente parte de seu subtipo, isto é, cada subtipo herda os atributos de seu supertipo. Por exemplo, os atributos do tipo de entidade de *Mensalista* são seu atributo direto (*Mensalista*) mais seus atributos herdados de *Funcionario* (*NumFunc*, *NomeFunc*, *DataAdmFunc* etc.). Atributos herdados não são mostrados em um DER. Sempre que você tem um subtipo, supõe-se que ele herda os atributos de seu supertipo.

5.3.2 Restrições de Disjunção e de Completude

As hierarquias de generalização não mostram as cardinalidades porque são sempre as mesmas. Já as restrições de disjunção e de completude podem ser mostradas. Disjunção significa que os subtipos em uma hierarquia de generalização não têm nenhuma entidade em comum³. Na Figura 5.21, na hierarquia de generalização há disjunção porque um valor mobiliário não pode ser ao mesmo tempo uma ação e um título. Por sua vez, na hierarquia de generalização da Figura 5.22 não há disjunção porque os assistentes podem ser tanto de alunos como de professores. Assim, o conjunto de alunos se sobrepõe ao conjunto de professores. Completude significa que cada entidade de um supertipo deve ser uma entidade em um dos subtipos na hierarquia de generalização⁴. A restrição de completude na Figura 5.21 significa que cada valor mobiliário deve ser ou uma ação ou um título.

Algumas hierarquias de generalização não possuem nem restrições de disjunção nem de completude. Na Figura 5.20, a ausência de uma disjunção significa que alguns funcionários podem ser tanto mensalistas quanto horistas. A falta de uma restrição de completude indica que alguns funcionários não são pagos por mês nem por hora (talvez por comissão).

³ NRT: Também conhecida como especialização mutuamente exclusiva ou categoria, ou seja, uma entidade de um subtipo não pode participar de qualquer outro subtipo no mesmo nível de hierarquia, sendo desta forma exclusivo a um único subtipo ou categoria.

⁴ NRT: Também conhecida como especialização total, ou seja, todos os subtipos identificados foram representados, assim uma entidade do supertipo sempre terá representação como um subtipo. Do contrário, é chamada especialização parcial.

FIGURA 5.21
Hierarquia de Generalização
para Valores Mobiliários

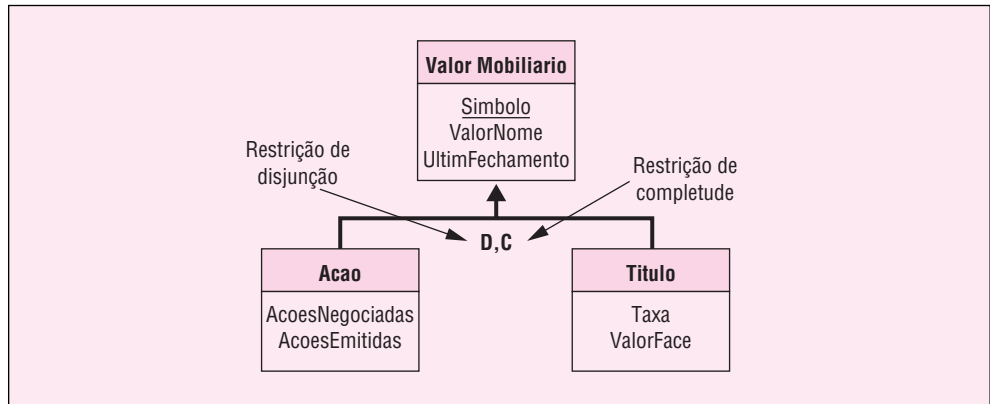


FIGURA 5.22
Hierarquia de Generalização
para Público Universitário

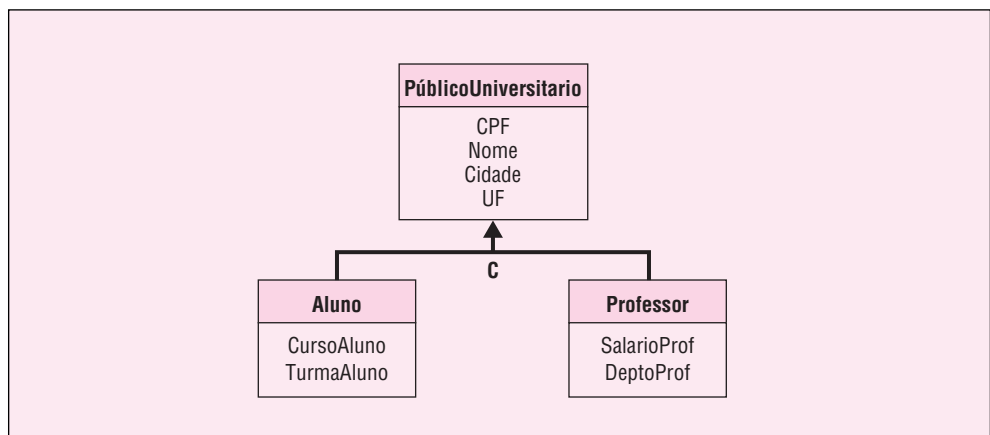
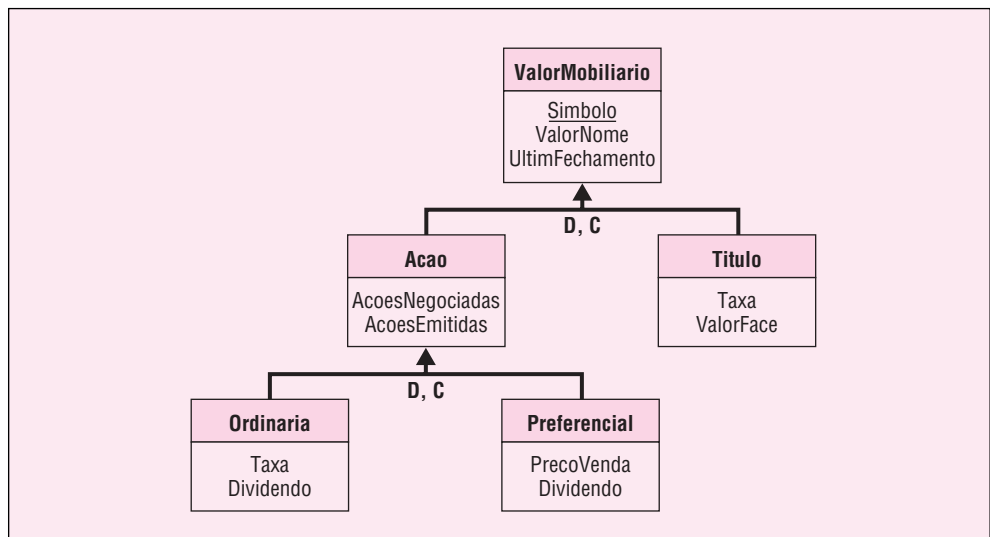


FIGURA 5.23
Níveis Múltiplos para
Hierarquias de
Generalização



5.3.3 Múltiplos Níveis de Generalização

As hierarquias de generalização podem ser estendidas a mais um nível. Esta prática pode ser útil em áreas como a de investimentos, onde o conhecimento é altamente estruturado. Na Figura 5.23, há dois níveis de subtipos abaixo de Valor Mobiliário. A herança se estende

a todos os subtipos, diretos e indiretos. Assim, ambos os tipos de entidade *Preferencial* e *Ordinaria* herdam os atributos de *Acao* (o pai imediato) e *Valor Mobiliario* (o pai indireto). Note que as restrições de disjunção e de completude podem ser feitas para cada grupo de subtipos.

5.4 Resumo da Notação e Regras de Diagrama

Você viu muitos aspectos da notação DER nas seções anteriores deste capítulo. Para que você não sofra pelo excesso de informação, esta seção fornece um conveniente resumo, bem como regras para ajudá-lo a evitar erros comuns de diagramação.

5.4.1 Resumo da Notação

Para ajudá-lo a lembrar a notação introduzida nas seções anteriores, a Tabela 5.2 apresenta um resumo, enquanto a Figura 5.24 mostra a notação para o banco de dados da universidade do Capítulo 4. A Figura 5.24 difere em alguns aspectos do banco de dados da universidade do Capítulo 4 para apresentar a maior parte da notação Pé-de-Galinha. A Figura 5.24 contém uma hierarquia de generalização para mostrar as semelhanças entre alunos e professores. Você deve observar que a chave primária dos tipos de entidade *Aluno* e *Professor* é *CPF*, um atributo herdado do tipo de entidade *Publico Universitario*. O tipo de entidade *Matricula* (associativa) e os relacionamentos identificadores (*Inscribe-se* e *Concede*) podem aparecer como um relacionamento M-N, como mostrado previamente na Figura 5.10. Além destas questões, a Figura 5.24 omite alguns atributos por uma questão de concisão.

TABELA 5.2 Resumo da Notação de Pé-de-Galinha

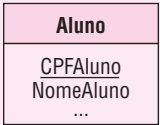
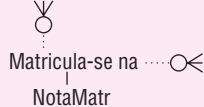
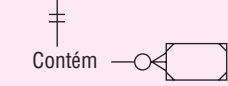




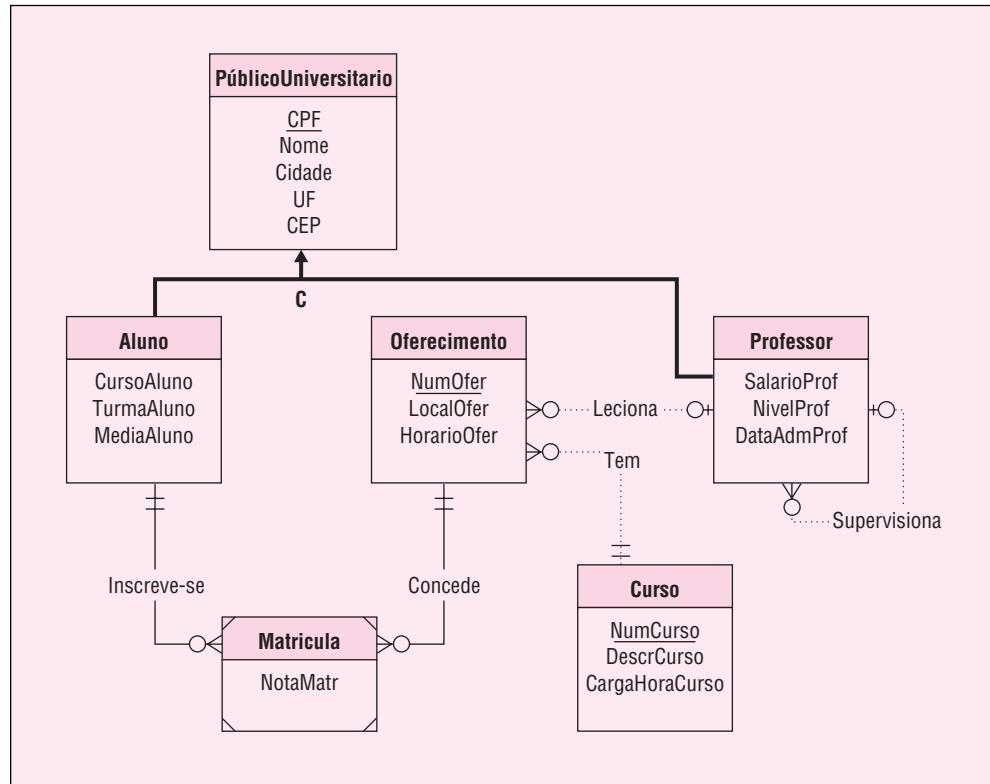
Símbolo	Significado
	Tipo de entidade com atributos (chave primária sublinhada).
	Relacionamento M-N com atributos: atributos são mostrados se há espaço; caso contrário, os atributos são listados separadamente.
	Dependência de identificador: relacionamento(s) identificador (linhas contínuas de relacionamento) e entidade fraca (linhas diagonais no canto do retângulo). Tipos de entidade associativa também são fracos porque são (por definição) dependentes de identificador.
	Hierarquia de generalização com restrições de disjunção e completude.
	Cardinalidade dependente de existência (cardinalidade mínima 1): símbolo interno é uma linha perpendicular à linha de relacionamento.
	Cardinalidade opcional (cardinalidade mínima 0): símbolo interno é um círculo.
	Cardinalidade de valor único (cardinalidade máxima 1): símbolo exteno é uma linha perpendicular.

FIGURA 5.24
DER para o Banco de Dados
da Universidade



Representação de Regras de Negócio em um DER

À medida que desenvolve um DER, você deve lembrar que todo DER contém regras de negócio que impõem políticas organizacionais e promovem uma comunicação eficiente entre os *stakeholders*⁵ do negócio. Um DER contém importantes regras de negócio representadas como chaves primárias, relacionamentos, cardinalidades e hierarquias de generalização. As chaves primárias provêm a identificação da entidade, um requisito importante para a comunicação em um negócio. A dependência de identificador envolve uma entidade que depende de outras entidades para ser identificada, um requisito em algumas comunicações em um negócio. Os relacionamentos indicam conexões diretas entre unidades de comunicação de um negócio. As cardinalidades restringem o número de entidades relacionadas em relacionamentos, dando suporte às políticas organizacionais e a uma comunicação coerente no negócio. As hierarquias de generalização com restrições de disjunção e de completude prestam suporte à classificação de entidades do negócio e das políticas organizacionais. Logo, os elementos de um DER são cruciais para execução das políticas organizacionais e para uma comunicação eficiente no negócio.

Para tipos adicionais de restrições de negócio, um DER pode ser aprimorado com documentação informal ou com uma linguagem de regras formais. Como o SQL:2003 suporta uma linguagem de regras formais (capítulos 11 e 14), não se propõe uma linguagem aqui. Na ausência de uma linguagem de regras formais, as regras de negócio podem ser armazenadas como documentação informal associada a tipos de entidade, atributos e relacionamentos. Espécies típicas de regras de negócios para especificar como documentação informal são restrições de chave candidata, restrições de comparação de atributo, restrições de valor nulo e valores padrão. As chaves candidatas fornecem meios alternativos para identificar entidades de negócio. As restrições de comparação de atributo restringem os valores de atributos tanto para um conjunto fixo de valores quanto para os valores de outros atributos. As restrições de

⁵ NRT: São pessoas ou organizações que serão afetadas pelo sistema e que têm influência direta ou indireta nos requisitos do sistema.

TABELA 5.3
Resumo de Regras de
Negócio em um DER

Regra de Negócio	Representação no DER
Identificação de entidade	Chaves primárias para tipos de entidade, dependência de identificador (entidades fracas e relacionamentos identificadores), documentação informal sobre atributos únicos.
Conexões entre entidades de negócio	Relacionamentos
Número de entidades relacionadas	Cardinalidades mínima e máxima
Inclusão entre grupos de entidades	Hierarquias de generalização
Valores razoáveis	Documentação informal sobre restrições de atributo (comparação a valores constantes ou outros atributos)
Completude da coleta de dados	Documentação informal sobre valores nulos e valores padrão

TABELA 5.4
Regras de Completude e
Consistência

Tipo de Regra	Descrição
<i>Completude</i>	<ol style="list-style-type: none"> 1. Regra da chave primária: Todos os tipos de entidade têm uma chave primária (direta, emprestada ou herdada). 2. Regra do nome: Todos os tipos de entidade, relacionamentos e atributos têm um nome. 3. Regra da cardinalidade: É dada uma cardinalidade para os dois tipos de entidade em um relacionamento. 4. Regra da participação da entidade: Todos os tipos de entidade exceto aqueles em uma hierarquia de generalização que participam em pelo menos um relacionamento. 5. Regra da participação da hierarquia de generalização: Cada hierarquia de generalização participa em pelo menos um relacionamento com um tipo de entidade que não está na hierarquia de generalização.
<i>Consistência</i>	<ol style="list-style-type: none"> 1. Regra do nome da entidade: Os nomes dos tipos de entidade são únicos. 2. Regra do nome do atributo: Os nomes dos atributos são únicos dentro dos tipos de entidade e relacionamentos. 3. Regra do nome de atributo herdado: Os nomes de atributo em um subtipo não correspondem aos nomes de atributo herdados (diretos ou indiretos). 4. Regra de conexão de relacionamento/tipo de entidade: Todos os relacionamentos ligam dois tipos de entidade (não necessariamente distintos). 5. Regra de conexão relacionamento/relacionamento: Os relacionamentos não se ligam a outros relacionamentos. 6. Regra da entidade fraca: Entidades fracas têm pelo menos um relacionamento identificador. 7. Regra do relacionamento identificador: Para cada relacionamento identificador, pelo menos um tipo de entidade participante deve ser fraco. 8. Regra de cardinalidade da dependência de identificador: Para cada relacionamento identificador, as cardinalidades mínima e máxima devem ser 1 na direção da filha (entidade fraca) para o tipo de entidade pai. 9. Regra da chave estrangeira redundante: Chaves estrangeiras redundantes não são usadas.

valor nulo e os valores padrão apóiam políticas sobre a completude das atividades de coleta de dados. A Tabela 5.3 resume as espécies comuns de regras de negócios que podem ser especificadas tanto formal quanto informalmente em um DER.

5.4.2 Regras de Diagrama

Para proporcionar orientações sobre o uso correto da notação, a Tabela 5.4 apresenta regras de completude e de consistência. Você deve aplicar tais regras ao completar um DER para assegurar-se de que não há nenhum erro de notação em seu DER. Desse modo, as regras de diagrama servem a um propósito semelhante ao das regras de sintaxe para uma linguagem de computação. A ausência de erros de sintaxe não quer dizer que um programa de computador execute suas tarefas corretamente. De igual maneira, a ausência de erros de notação não quer dizer que um DER fornece uma representação adequada dos dados. As regras de diagrama *não* asseguram que foram consideradas as múltiplas alternativas, que foram representados corretamente os requisitos dos usuários nem que seu projeto foi adequadamente documentado. O Capítulo 6 discute essas questões para aumentar suas habilidades na modelagem de dados.

A maioria das regras na Tabela 5.4 não exige muita elaboração. As três primeiras regras de completude e as cinco primeiras regras de consistência são simples de entender. Mesmo que as regras sejam simples, você ainda deve verificar seu DER para checar sua conformidade, já que é fácil deixar passar uma violação em um DER de tamanho médio.

As regras de consistência não exigem nomes únicos para os relacionamentos porque os tipos de entidade participantes fornecem um contexto para os nomes dos relacionamentos. No entanto, é uma boa prática usar nomes únicos de relacionamento tanto quanto possível para tornar os relacionamentos fáceis de distinguir. Além disso, dois ou mais relacionamentos envolvendo os mesmos tipos de entidade devem ser únicos porque os tipos de entidade não podem mais proporcionar o contexto para distinguir os relacionamentos. Como é incomum ter mais de um relacionamento entre os mesmos tipos de entidade, as regras de consistência não incluem esta provisão.

As regras de completude 4 (regra da participação da entidade) e 5 (regra da participação da hierarquia de generalização) exigem elaboração. Quebrar tais regras é um alerta, não necessariamente um erro. Na maioria dos DERs, todos os tipos de entidade que não estão em uma hierarquia de generalização e todas as hierarquias de generalização estão ligados a pelo menos um outro tipo de entidade. Em situações raras, um DER contém um tipo de entidade não relacionado somente para armazenar uma lista de entidades. A regra 5 se aplica a uma hierarquia de generalização inteira, não a cada tipo de entidade em uma hierarquia de generalização. Em outras palavras, pelo menos um tipo de entidade em uma hierarquia de generalização deve estar ligado a pelo menos um tipo de entidade que não está na hierarquia de generalização. Em muitas hierarquias de generalização, múltiplos tipos de entidade participam em relacionamentos. As hierarquias de generalização permitem que subtipos participem em relacionamentos, restringindo assim a participação em relacionamentos. Por exemplo, na Figura 5.24, *Aluno* e *Professor* participam em relacionamentos.

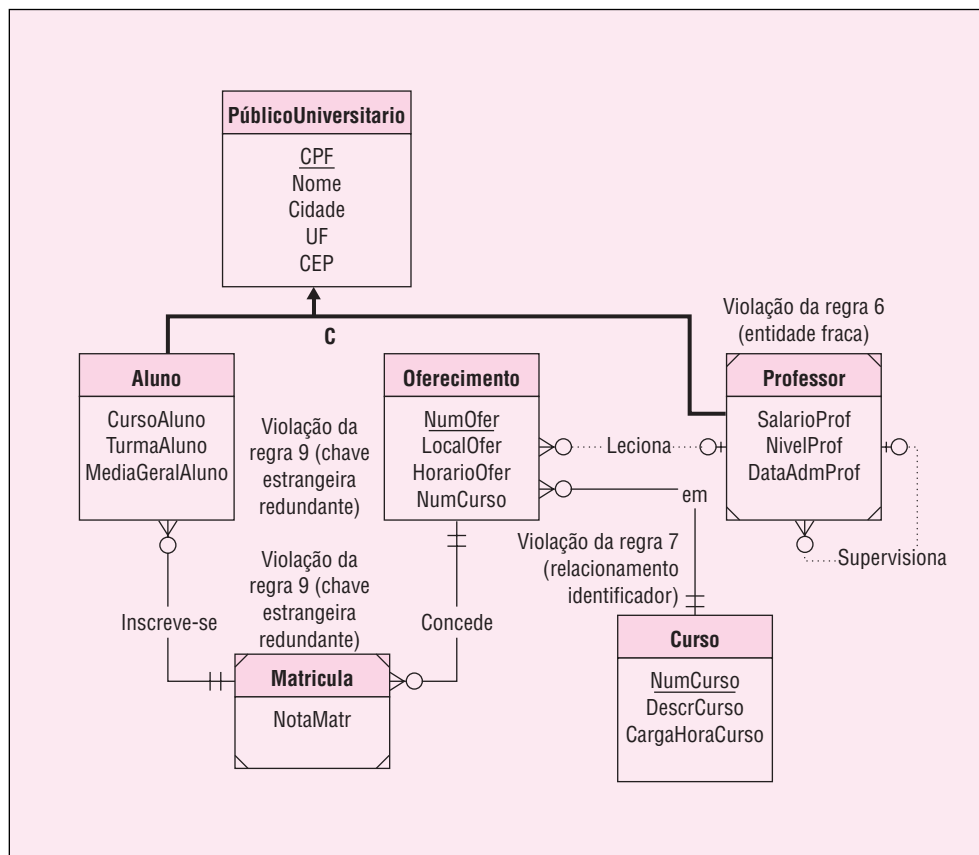
As regras de consistência 6 a 9 envolvem erros comuns nos DERs de modeladores de dados iniciantes. Os modeladores de dados iniciantes violam as regras de consistência 6 a 8 por causa da complexidade da dependência de identificador. A dependência de identificador envolvendo uma entidade fraca e relacionamentos identificadores enseja mais erros do que outras partes da notação Pé-de-Galinha. Adicionalmente, cada relacionamento identificador também exige cardinalidades mínima e máxima 1 na direção da filha (entidade fraca) para o tipo de entidade pai. Os modeladores de dados iniciantes violam a regra de consistência 9 (regra da chave estrangeira redundante) por causa de confusão entre um DER e o modelo de dados relacionais. O processo de conversão transforma os relacionamentos 1-M em chaves estrangeiras.

Exemplo de Violação de Regra e Resoluções

Como as regras da dependência de identificador e a regra da chave estrangeira redundante são uma fonte freqüente de erros para projetistas iniciantes, esta seção traz um exemplo para ilustrar as violações de regra e resoluções. A Figura 5.25 mostra as violações das regras da dependência de identificador (regras de consistência 6 a 9) e da regra da chave estrangeira redundante (regra de consistência 9) para o DER do banco de dados da universidade. A seguinte lista explica as violações:

- **Violação da regra de consistência 6 (regra da entidade fraca):** *Professor* não pode ser uma entidade fraca sem pelo menos um relacionamento identificador.
- **Violação da regra de consistência 7 (regra do relacionamento identificador):** O relacionamento *Tem* identifica, mas nem *Oferecimento* nem *Curso* são entidades fracas.
- **Violação da regra de consistência 8 (regra da cardinalidade da dependência de identificador):** A cardinalidade do relacionamento *Inscreve-se* de *Matricula* a *Aluno* deve ser (1, 1) e não (0, Muitos).
- **Violação da regra de consistência 9 (regra da chave estrangeira redundante):** O atributo *NumCurso* no tipo de entidade *Oferecimento* é redundante com o relacionamento *Tem*. Como *NumCurso* é a chave primária de *Curso*, não deve ser um atributo de *Oferecimento* para ligar um *Oferecimento* a um *Curso*. O relacionamento *Tem* fornece a ligação a *Curso*.

FIGURA 5.25
DER com Violações das
Regras de Consistência 6 a 9



Para a maioria das regras, resolver as violações é fácil. A tarefa mais importante é reconhecer a violação. Para as regras de dependência de identificador, a solução pode depender dos detalhes do problema. A seguinte lista sugere possíveis ações corretivas para erros de diagrama:

- **Solução para a regra de consistência 6 (regra da entidade fraca):** O problema pode ser resolvido adicionando um ou mais relacionamentos identificadores ou transformando a entidade fraca em uma entidade regular. Na Figura 5.25, o problema é resolvido tornando *Professor* uma entidade regular. A solução mais comum é adicionar um ou mais relacionamentos identificadores.
- **Solução para a regra de consistência 7 (regra do relacionamento identificador):** O problema pode ser resolvido adicionando uma entidade fraca ou fazendo com que o relacionamento deixe de ser identificador. Na Figura 5.25, o problema é resolvido tornando o relacionamento *Tem* como não identificador. Se há mais de um relacionamento identificador envolvendo o mesmo tipo de entidade, a solução típica envolve transformar o tipo de entidade comum em uma entidade fraca.
- **Solução para a regra de consistência 8 (regra da cardinalidade da dependência de identificador):** O problema pode ser resolvido transformando a cardinalidade da entidade fraca para (1,1). Tipicamente, a cardinalidade do relacionamento identificador é invertida. Na Figura 5.25, a cardinalidade do relacionamento *Inscreve-se* deve ser invertida ((1,1) próxima a *Aluno* e (0, Muitos) próxima a *Matricula*).
- **Solução para a regra de consistência 9 (regra da chave estrangeira redundante):** Normalmente o problema pode ser resolvido retirando a chave estrangeira redundante. Na Figura 5.25, *NumCurso* deve ser retirado como um atributo de *Oferecimento*. Em alguns casos, o atributo não pode representar uma chave estrangeira. Se o atributo não representa uma chave estrangeira, este deve ser renomeado e não retirado.

TABELA 5.5
Organização Alternativa
das Regras

Categoria	Regras
<i>Nomes</i>	<p>Todos os tipos de entidade, relacionamentos e atributos devem ter um nome. (Regra de completude 2)</p> <p>Os nomes do tipo de entidade são únicos. (Regra de consistência 1)</p> <p>Os nomes dos atributos são únicos dentro de tipos de entidade e relacionamentos. (Regra de consistência 2)</p> <p>Os nomes dos atributos em um subtipo não correspondem aos nomes de atributos herdados (nem diretos, nem indiretos). (Regra de consistência 3)</p>
<i>Conteúdo</i>	<p>Todos os tipos de entidade têm uma chave primária (direta, emprestada ou herdada). (Regra de completude 1)</p> <p>A cardinalidade deve ser atribuída a ambos os tipos de entidade em um relacionamento. (Regra de completude 3)</p>
<i>Conexão</i>	<p>Todos os tipos de entidade exceto aqueles que estão em uma hierarquia de generalização participam pelo menos de um relacionamento. (Regra de completude 4)</p> <p>Cada hierarquia de generalização participa pelo menos de um relacionamento com um tipo de entidade que não está na hierarquia de generalização. (Regra de completude 5)</p> <p>Todos os relacionamentos ligam dois tipos de entidade. (Regra de consistência 4)</p> <p>Os relacionamentos não se ligam a outros relacionamentos. (Regra de consistência 5)</p> <p>Não se usam chaves estrangeiras redundantes. (Regra de consistência 9)</p>
<i>Dependência de identificador</i>	<p>Entidades fracas têm pelo menos um relacionamento identificador. (Regra de consistência 6)</p> <p>Para cada relacionamento identificador, pelo menos um tipo de entidade participante deve ser fraca. (Regra de consistência 7)</p> <p>Para cada entidade fraca, as cardinalidades mínima e máxima devem ser iguais a 1 para cada relacionamento identificador. (Regra de consistência 8)</p>

Organização Alternativa de Regras

A organização das regras da Tabela 5.4 pode ser difícil de memorizar. A Tabela 5.5 fornece uma alternativa ao agrupar as regras por finalidade. Se acha essa organização mais intuitiva, você deve usá-la. Seja qual for o modo que preferir para se lembrar das regras, o ponto importante é aplicá-las depois de completar um DER. Para ajudá-lo a aplicar as regras de diagrama, a maioria das ferramentas CASE executa verificações específicas das notações suportadas pelas ferramentas. A próxima seção descreve a verificação de regras de diagrama pelo ER Assistant, a ferramenta de modelagem de dados disponibilizada neste livro.

Suporte no ER Assistant

Para melhorar a produtividade de modeladores de dados iniciantes, o ER Assistant fornece suporte às regras de consistência e completude listadas na Tabela 5.4. O ER Assistant suporta as regras de consistência 4 e 5 por meio de suas ferramentas de diagramação. Os relacionamentos devem estar ligados a dois tipos de entidade (não necessariamente distintos) proibindo violações das regras de consistência 4 e 5. Para as outras regras de completude e consistência, o ER Assistant fornece o botão de *Check Diagram* que gera um relatório de violações de regras. Como o botão *Check Diagram* pode ser usado quando um DER ainda não está completo, o ER Assistant não exige que você conserte as violações de regra encontradas em um DER. Antes de completar um DER, você deve resolver cada violação apontada pelo ER Assistant.

Para a regra da chave estrangeira redundante (regra de consistência 9), o ER Assistant usa uma implementação simples para determinar se um DER contém uma chave estrangeira redundante. O ER Assistant verifica o tipo de entidade filha (tipo de entidade no lado “muitos” do relacionamento) para verificar se há um atributo com o mesmo nome e tipo de dados que a chave primária no tipo de entidade pai (tipo de entidade no lado “um” do relacionamento). Se o ER Assistant acha um atributo com o mesmo nome e tipo de dados, ele inclui uma violação no relatório do *Check Diagram*.

5.5 Comparação com Outras Notações

A notação DER apresentada neste capítulo é semelhante, mas não idêntica, às que você pode encontrar mais tarde. Não há nenhuma notação padrão para DERs. Há talvez seis notações razoavelmente populares de DER, cada uma com suas próprias pequenas variações que aparecem na prática. A notação neste capítulo vem da matriz do Pé-de-Galinha no Visio Profissional 5 com a adição de notações de generalização. As notações que você encontrará na prática dependerão de fatores como a ferramenta de modelagem de dados (se houver uma) usada em sua organização e setor. Uma coisa é certa: você deve se preparar para se adaptar à notação em uso. Esta seção descreve as variações de DER que você pode encontrar assim como a notação de Diagrama de Classe da Unified Modeling Language (UML – Linguagem de Modelagem Unificada), um padrão emergente para modelagem de dados.

5.5.1 Variações de DER

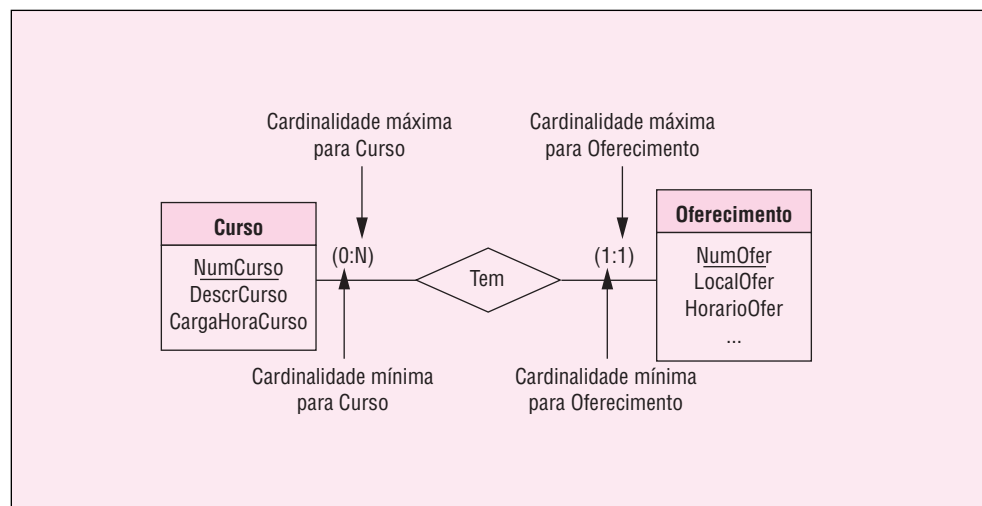
Como não há um padrão amplamente aceito de DER, símbolos diferentes podem ser usados para representar o mesmo conceito. As cardinalidades do relacionamento são fonte de muitas variações. Você deve prestar atenção à colocação dos símbolos de cardinalidade. A notação deste capítulo coloca os símbolos perto do tipo de entidade “distante”, enquanto outras notações colocam os símbolos de cardinalidade perto do tipo de entidade “próximo”. A notação deste capítulo usa uma representação visual das cardinalidades com as cardinalidades mínima e máxima dadas por três símbolos. Outras notações usam uma representação de texto com letras e números inteiros em vez de símbolos. Por exemplo, a Figura 5.26 mostra um DER Chen com a posição das cardinalidades invertidas, cardinalidades demonstradas pelo texto e relacionamentos indicados por losangos.

Outras variações de símbolo são representações visuais para certas espécies de tipos de entidades. Em algumas notações, entidades fracas e relacionamentos M-N têm representações especiais. Entidades fracas às vezes são incluídas em retângulos duplos. Relacionamentos identificadores às vezes são incluídos em losangos duplos. Os relacionamentos M-N com atributos às vezes são mostrados como um retângulo com um losango dentro para indicar as qualidades duplas (tanto relacionamento como tipo de entidade).

Além das variações de símbolo, há também variações de regra, como mostramos na lista a seguir. Para cada restrição, há um remédio. Por exemplo, se apenas há suporte para relacionamentos binários, os relacionamentos N-ários devem ser representados como um tipo de entidade associativa a relacionamentos 1-M.

1. Algumas notações não têm suporte a relacionamentos N-ários.
2. Algumas notações não têm suporte a relacionamentos M-N.
3. Algumas notações não têm suporte a relacionamentos com atributos.

FIGURA 5.26
Notação Chen para o DER
Curso-Oferecimento



4. Algumas notações não têm suporte a auto-relacionamentos (unários).
5. Algumas notações permitem que relacionamentos estejam ligados a outros relacionamentos.
6. Algumas notações mostram chaves estrangeiras como atributos.
7. Algumas notações permitem que os atributos tenham mais de um valor (atributos multi-valorados).

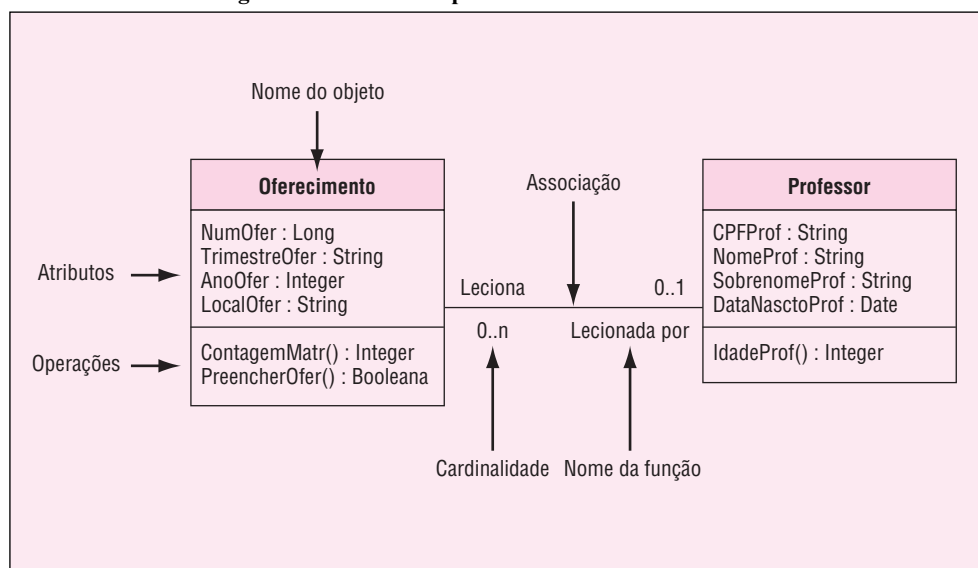
As restrições em uma notação de DER necessariamente não tornam a notação menos expressiva que outras notações sem as restrições. Podem ser necessários símbolos adicionais em um diagrama, mas os mesmos conceitos ainda podem ser representados. Por exemplo, a notação Pé-de-Galinha não oferece suporte a relacionamentos N-ários. No entanto, relacionamentos N-ários podem ser representados usando tipos N-ários de entidades associativas. Os tipos N-ários de entidades associativas requerem símbolos adicionais em relação aos relacionamentos N-ários, mas os mesmos conceitos são representados.

5.5.2 A Notação de Diagrama de Classe da Linguagem de Modelagem Unificada (UML)

A Linguagem de Modelagem Unificada tornou-se a notação padrão para modelagem orientada a objetos. A modelagem orientada a objetos dá ênfase aos objetos e não aos processos, como enfatizado geralmente nas abordagens tradicionais de desenvolvimento de sistemas. Na modelagem orientada a objetos, definem-se os objetos primeiro, seguidos pelas características (atributos e operações) dos objetos e, então, a interação dinâmica entre os objetos. A UML contém diagramas de classe, diagramas de interface e diagramas de interação para apoiar a modelagem orientada a objetos. A notação de diagrama de classe fornece uma alternativa às notações de DER apresentadas neste capítulo.

Os diagramas de classe contêm classes (grupos de objetos), associações (relacionamentos binários) entre classes e características dos objetos (atributos e operações). A Figura 5.27 mostra um diagrama de classe simples contendo as classes *Professor* e *Oferecimento*. O diagrama foi desenhado com o estêncil UML no Visio Professional. A associação na Figura 5.27 representa um relacionamento 1-M. A UML tem suporte a nomes de funções e cardinalidades (mínima e máxima) para cada direção em uma associação. A cardinalidade 0..1 significa que um objeto de oferecimento pode estar relacionado a no mínimo zero objetos de Professor e no máximo um objeto de Professor. As operações estão listadas abaixo dos atributos. Cada operação contém uma lista de parâmetros entre parênteses junto com o tipo de dados retornado pela operação.

FIGURA 5.27 Diagrama de Classe Simples



As associações na UML são semelhantes aos relacionamentos na notação Pé-de-Galinha. As associações podem representar relacionamentos binários ou unários. Para representar um relacionamento N-ário são necessários uma classe e um grupo de associações. Para representar um relacionamento M-N com atributos, a UML fornece a classe de associação para permitir que as associações tenham atributos e operações. A Figura 5.28 mostra uma classe de associação que representa um relacionamento M-N entre o *Aluno* e as classes de *Oferecimento*. A classe de associação contém os atributos do relacionamento.

Diferentemente da maioria das notações DER, o suporte à generalização foi construído na UML desde seu princípio. Na maioria das notações DER, a generalização foi criada como uma característica adicional depois que uma notação já estava estabelecida. Na Figura 5.29, a seta vazia grande denota uma classificação da classe de *Aluno* nas classes *Graduacao* e *Pos-Graduacao*. A UML suporta nomes e restrições de generalização. Na Figura 5.29, a generalização de *Status* está completa, querendo dizer que cada aluno deve ser de graduação ou de pós-graduação.

FIGURA 5.28
Associação de Classe
Representando um
Relacionamento M-N
com Atributos

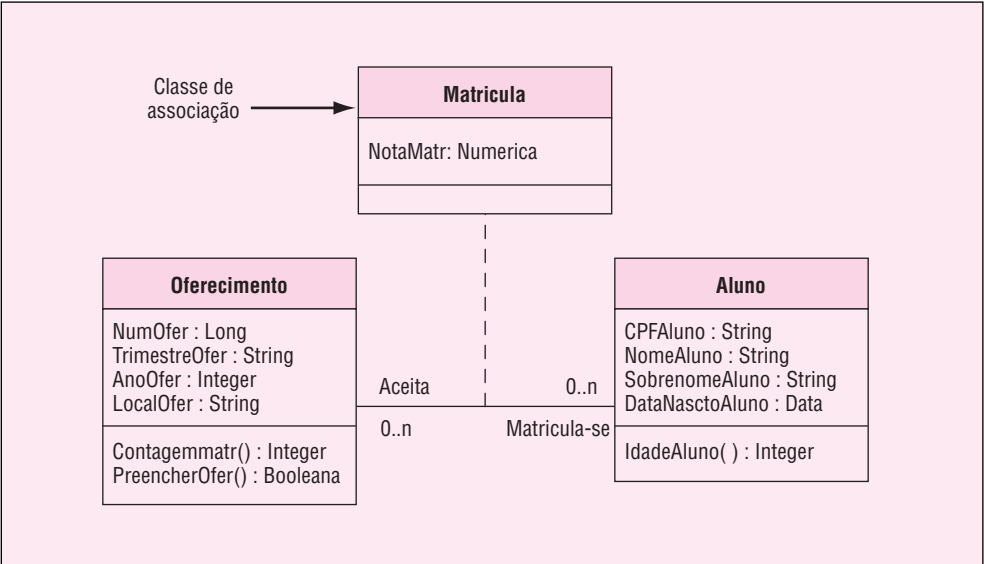


FIGURA 5.29
Diagrama de Classe com
um Relacionamento de
Generalização

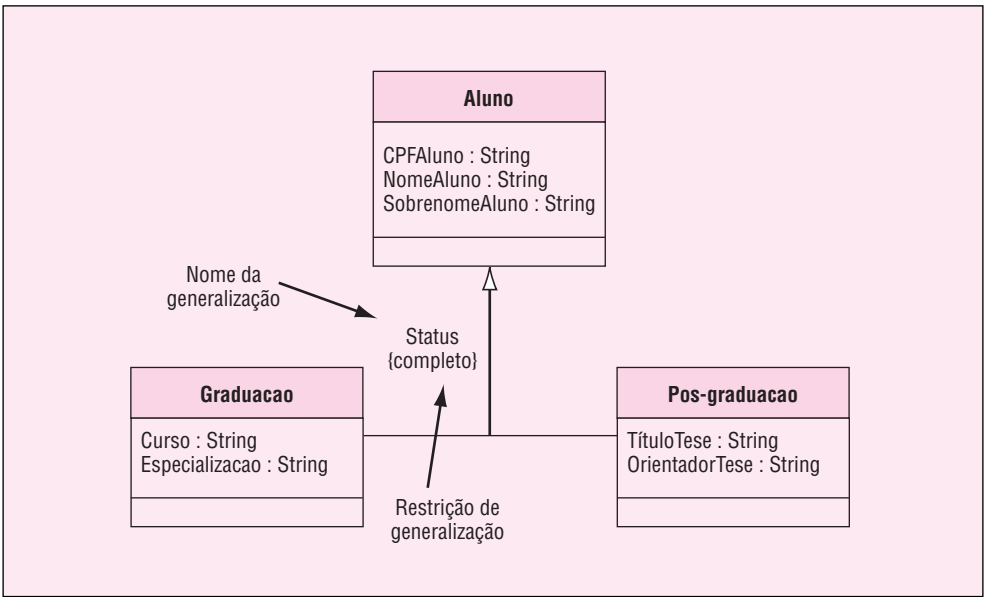
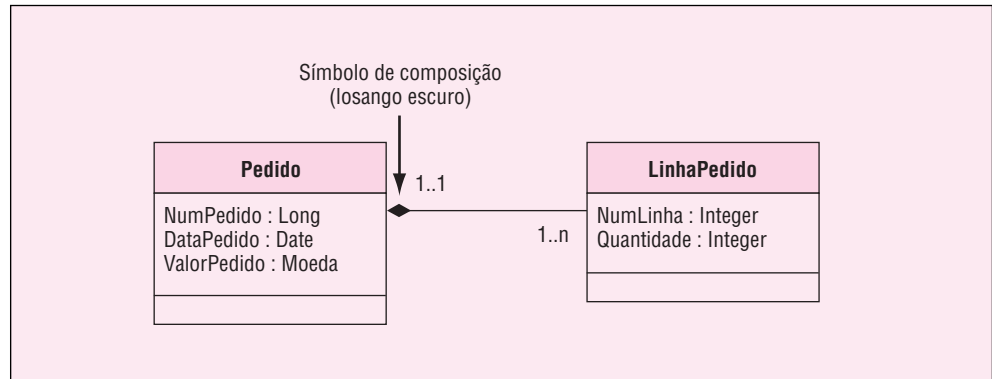


FIGURA 5.30
Diagrama de Classe com um
Relacionamento de
Composição



A UML também fornece um símbolo especial para relacionamentos de composição, semelhante às dependências de identificador nas notações DER. Em um relacionamento de composição, os objetos em uma classe filha pertencem somente a objetos na classe pai. Na Figura 5.30, cada objeto de *LinhaPedido* pertence a um objeto de *Pedido*. A exclusão de um objeto pai causa a exclusão dos objetos filhos relacionados. Em consequência, o objeto filho normalmente toma emprestada parte de sua chave primária do objeto pai. No entanto, a UML não exige esta dependência de identificador.

Os diagramas de classe da UML proporcionam muitas outras funcionalidades não apresentadas nessa breve visão geral. A UML tem suporte a tipos diferentes de classes para integrar questões de linguagem de programação com aspectos da modelagem de dados. Outros tipos de classes incluem classes de valor, classes de estereótipo, classes parametrizadas e classes abstratas. Para generalização, a UML oferece suporte a restrições adicionais tais como classificação estática e dinâmica e diferentes interpretações de relacionamentos de generalização (subtipo e subclasse). Para garantir a integridade de dados, a UML suporta a especificação de restrições em um diagrama de classe.

Você deve notar que diagramas de classe são somente uma parte da UML. Até certo ponto, os diagramas de classe devem ser entendidos no contexto da modelagem orientada a objetos e de toda a UML. Você deve ter em mente que deveria dedicar um trimestre acadêmico inteiro à modelagem orientada a objetos e à UML.

Considerações Finais

Este capítulo explicou a notação dos diagramas entidade–relacionamento como um requisito para aplicar diagramas entidade–relacionamento no processo de desenvolvimento de banco de dados. Usando a notação Pé-de-Galinha, este capítulo descreveu símbolos, padrões importantes de relacionamento e hierarquias de generalização. Os símbolos básicos são tipos de entidade, relacionamentos, atributos e cardinalidades a fim de representar o número de entidades participando de um relacionamento. Quatro importantes padrões de relacionamento foram descritos: relacionamentos muitos-para-muitos (M-N) com atributos, tipos de entidades associativas representando relacionamentos N-ários, relacionamentos identificadores fornecendo chaves primárias para entidades fracas e auto-relacionamentos (unários). As hierarquias de generalização permitem que a classificação dos tipos de entidade demonstre as semelhanças entre os tipos de entidade.

Para melhorar o seu uso da notação Pé-de-Galinha, apresentamos representações de regras de negócio, regras de diagrama e comparações com outras notações. Este capítulo apresentou a representação informal e formal de regras de negócio em um diagrama entidade–relacionamento de modo a fornecer um contexto organizacional para diagramas entidade–relacionamento. As regras de diagrama envolvem requisitos de completude e de consistência. As regras de diagrama asseguram que um DER não contém erros óbvios. Para ajudá-lo a aplicar as regras, o ER Assistant fornece uma ferramenta para verificar as regras em um DER completo. Com a finalidade de ampliar seu conhecimento em notações DER, este capítulo apresentou as variações comuns que você pode encontrar assim como a notação do Diagrama de Classe da Linguagem de Modelagem Unificada (UML), uma notação-padrão para a modelagem orientada a objetos.

Este capítulo enfatizou a notação dos DERs a fim de fornecer uma base sólida para o estudo mais difícil que é aplicar a notação em problemas administrativos. Para dominar a modelagem de dados, você necessita entender a notação DER e conseguir uma ampla prática construindo DERs. O Capítulo 6 aborda a prática de construir DERs para problemas de negócios. Aplicar a Anotação envolve uma representação completa e coerente dos requisitos dos usuários, a geração de projetos alternativos e a documentação de decisões de projeto. Além dessas habilidades, o Capítulo 6 traz as regras para converter um DER em um projeto de tabelas. Com um estudo cuidadoso, os capítulos 5 e 6 fornecem uma base sólida para executar modelagem de dados em bancos de dados de negócio.

Revisão para Conceitos

- Conceitos Básicos: tipos de entidade, relacionamentos e atributos.
- Cardinalidades mínima e máxima para restringir a participação em um relacionamento.
- Classificação de cardinalidades como opcional, obrigatória e funcional.
- Dependência de existência para entidades que não podem ser armazenadas sem o armazenamento de entidades relacionadas.
- Dependência de identificador envolvendo entidades fracas e relacionamentos identificadores para o suporte a tipos de entidade que tomam emprestada pelo menos parte de suas chaves primárias.
- Relacionamentos M-N com atributos: atributos são associados à combinação de tipos de entidade, não apenas com um dos tipos de entidade.
- Equivalência entre um relacionamento M-N e um tipo de entidade associativa a relacionamento identificador 1-M.
- Tipos de entidades associativas N-árias para representar relacionamentos N-ários entre mais de dois tipos de entidade.
- Auto-relacionamentos (unários) para representar as associações entre entidades do mesmo tipo de entidade.
- Diagramas de instâncias para ajudar a distinguir entre auto-relacionamentos M-N e 1-M.
- Hierarquias de generalização para mostrar semelhanças entre tipos de entidade.
- Representação de regras de negócio em um DER: identificação de entidade, conexões entre entidades de negócio, número de entidades relacionadas, inclusão entre conjuntos de entidades, valores razoáveis e completude da coleta de dados.
- Regras de diagrama para prevenir erros óbvios na modelagem de dados.
- Fontes comuns de erros de diagrama: dependência de identificador e chaves estrangeiras redundantes.
- Suporte para as regras de diagrama no ER Assistant.
- Variações de DER: símbolos e regras de diagrama.
- Notação do Diagrama de Classe da Linguagem de Modelagem Unificada (UML) como uma alternativa ao Modelo Entidade-Relacionamento.

Questões

1. O que é um tipo de entidade?
2. O que é um atributo?
3. O que é um relacionamento?
4. Qual é a correspondência em linguagem natural para tipos de entidades e relacionamentos?
5. Qual é a diferença entre um DER e um diagrama de instâncias?
6. Que símbolos são as contrapartes em um DER das chaves estrangeiras no Modelo Relacional?
7. Quais cardinalidades indicam relacionamentos obrigatórios, opcionais e funcionais?

8. Quando é importante converter um relacionamento M-N em relacionamentos 1-M?
9. Como um diagrama de instâncias pode ajudar a determinar se um auto-relacionamento é um relacionamento 1-M ou M-N?
10. Quando um DER deve conter entidades fracas?
11. Qual é a diferença entre dependência de existência e tipo de entidade fraca?
12. Por que a classificação é importante nos negócios?
13. O que é a herança nas hierarquias de generalização?
14. Qual a finalidade das restrições de disjunção e de completude para uma hierarquia de generalização?
15. Que símbolos são usados para cardinalidades na notação Pé-de-Galinha?
16. Quais são os dois componentes da dependência de identificador?
17. Como os relacionamentos N-ários são representados na notação Pé-de-Galinha?
18. O que é tipo de entidade associativa?
19. Qual é a equivalência entre um relacionamento M-N e relacionamentos 1-M?
20. O que significa dizer que parte de uma chave primária é emprestada?
21. Qual é a finalidade das regras de diagrama?
22. Quais são as limitações das regras de diagrama?
23. Que regras de consistência comumente são transgredidas por modeladores de dados iniciantes?
24. Por que modeladores de dados iniciantes violam as regras de dependência de identificador (regras de consistência 6 a 8)?
25. Por que modeladores de dados iniciantes violam a regra de consistência 9 sobre chaves estrangeiras redundantes?
26. Por que uma ferramenta CASE deve fornecer suporte a regras de diagrama?
27. Como o ER Assistant fornece suporte às regras de consistência 4 e 5?
28. Como o ER Assistant fornece suporte a todas as regras de consistência exceto as regras 4 e 5?
29. Por que o ER Assistant não exige a correção de todos os erros de diagrama encontrados em um DER?
30. Como o ER Assistant implementa a regra de consistência 9 sobre chaves estrangeiras redundantes?
31. Liste algumas diferenças de símbolo na notação DER que você pode experimentar em sua carreira.
32. Liste algumas diferenças de regra de diagrama na notação DER que você pode experimentar em sua carreira.
33. O que é a Linguagem de Modelagem Unificada (UML)?
34. Quais são os elementos de modelagem em um diagrama de classe UML?
35. Quais tipos de regras de negócios são formalmente representados na notação Pé-de-Galinha para DERs?
36. Que tipos de regras de negócios são definidos por documentação informal na ausência de uma linguagem de regras para um DER?

Problemas

Os problemas enfatizam o uso correto da notação Pé-de-Galinha e a aplicação das regras de diagrama. Essa ênfase é coerente com a pedagogia do capítulo. Os problemas que apresentam maiores desafios no Capítulo 6 enfatizam os requisitos de usuários, transformações de diagrama, documentação de projeto e conversão de esquemas. Para obter um bom entendimento da modelagem de dados, você deve completar os problemas em ambos os capítulos.

1. Desenhe um DER contendo os tipos de entidade Pedido e Cliente ligados por um relacionamento 1-M de Cliente a Pedido. Escolha um nome de relacionamento apropriado usando seu conhecimento usual das interações entre clientes e pedidos. Defina cardinalidades mínimas de modo que um pedido seja opcional para um cliente e que um cliente seja obrigatório para um pedido. Para o tipo de entidade Cliente, adicione atributos NumCli (chave primária), NomeCli, SobrenomeCli, EndCli, CidadeCli, UF Cli, CEP Cli e SaldoCli (saldo). Para o tipo de entidade Pedido, adicione atributos para o NumPed (chave primária), DataPed, NomePed, EndPed, CidadePed, UFPed, e CEP Ped. Se você estiver usando o ER Assistant ou outra ferramenta de projeto que tenha suporte à especificação de tipos de dados, escolha tipos de dados apropriados para os atributos com base em seu conhecimento usual.
2. Amplie o DER do problema 1 com o tipo de entidade Funcionário e um relacionamento 1-M de Funcionário a Pedido. Escolha um nome de relacionamento apropriado usando seu conhecimento

usual das interações entre funcionários e pedidos. Defina cardinalidades mínimas de modo que um funcionário seja opcional a um pedido e que um pedido seja opcional a um funcionário. Para o tipo de entidade *Funcionario*, adicione atributos *NumFunc* (chave primária), *FuncNomeFunc*, *SobrenomeFunc*, *FoneFunc*, *EmailFunc*, *TaxaComFunc* (taxa de comissão) e *NomeDeptoFunc*. Se você estiver usando o ER Assistant ou outra ferramenta de projeto que dê suporte à especificação de tipos de dados, escolha tipos de dados apropriados para os atributos com base em seu conhecimento usual.

3. Amplie o DER do problema 2 com auto-relacionamento 1-M envolvendo o tipo de entidade *Funcionario*. Escolha um nome de relacionamento apropriado usando seu conhecimento usual de relacionamentos de organização entre funcionários. Defina cardinalidades mínimas de modo que o relacionamento seja opcional em ambas as direções.
4. Amplie o DER do problema 3 com o tipo de entidade *Produto* e um relacionamento M-N entre *Produto* e *Pedido*. Escolha um nome de relacionamento apropriado usando seu conhecimento usual das conexões entre produtos e pedidos. Defina cardinalidades mínimas de modo que uma ordem seja opcional a um produto e um produto seja obrigatório a uma ordem. Para o tipo de entidade *Produto*, adicione atributos *NumProd* (chave primária), *NomeProd*, *QtdeEstqProd*, *PreçoProd* e *DataProxExpedProd*. Para o relacionamento M-N, adicione um atributo para a quantidade pedida. Se você estiver usando o ER Assistant ou outra ferramenta de projeto que tenha suporte a especificação de tipos de dados, escolha tipos de dados apropriados para os atributos com base em seu conhecimento usual.
5. Revise o DER do problema 4 transformando o relacionamento M-N em um tipo de entidade associativa e dois relacionamentos identificadores 1-M.
6. Verifique seus DERs dos problemas 4 e 5 para ver se há violações às regras de diagrama. Se você seguiu as instruções do problema, seus diagramas não devem ter erros. Execute a verificação sem usar o ER Assistant. Depois, use a função Check Diagram do ER Assistant.
7. Usando seu DER corrigido do problema 6, adicione as violações de regras de consistência 6 a 9. Use a função Check Diagram do ER Assistant para identificar os erros.
8. Projete um DER para o tipo de entidade *Tarefa* e um auto-relacionamento M-N. Para o tipo de entidade *Tarefa*, adicione atributos *NumTarefa* (chave primária), *DescrTarefa*, *DuraçãoEstimadaTarefa*, *SituaçãoTarefa*, *HoraInicioTarefa* e *HoraTerminoTarefa*. Escolha um nome de relacionamento apropriado usando seu conhecimento usual das conexões de precedência entre tarefas. Defina cardinalidades mínimas de modo que o relacionamento seja opcional em ambas as direções.
9. Revise o DER do problema 8 transformando o relacionamento M-N em um tipo de entidade associativa e dois relacionamentos identificadores 1-M.
10. Defina uma hierarquia de generalização contendo o tipo de entidade *Aluno*, o tipo de entidade *Graduação* e o tipo de entidade *Pós-Graduação*. O tipo de entidade *Aluno* é o supertipo e *Graduação* e *Pós-Graduação* são subtipos. O tipo de entidade de *Aluno* tem atributos *NumAluno* (chave primária), *NomeAluno*, *SexoAluno*, *DataNasctoAluno* (data de nascimento), *EmailAluno* e *DataAdmAluno*. O tipo de entidade *Graduação* tem os atributos *CursoGrad*, *OpcionaisGrad* e *TurmaGrad*. O tipo de entidade *Pós-Graduação* tem os atributos *OrientadorPosGrad*, *TesePosGrad* e *SituaçãoAssistPosGrad* (situação de assistente). A hierarquia de generalização deve ser completa e com disjunção.
11. Defina uma hierarquia de generalização contendo o tipo de entidade *Funcionario*, o tipo de entidade *Professor* e o tipo de entidade *Administrador*. O tipo de entidade *Funcionario* é o supertipo e *Professor* e *Administrador* são subtipos. O tipo de entidade *Funcionario* tem os atributos *NumFunc* (chave primária), *FuncNomeFunc*, *FuncSexoFunc*, *DataNasctoFunc* (data de nascimento), *FoneFunc*, *EmailFunc* e *DataAdmFunc*. O tipo de entidade *Professor* tem os atributos *ClassificacaoProf*, *PeriodoPagtoProf* e *SituaçãoProf*. O tipo de entidade *Administrador* tem os atributos *TituloAdm*, *DuraçãoContratoAdm* e *DtNomeacaoAdm*. A hierarquia de generalização deve ser completa e com sobreposição.
12. Combine as hierarquias de generalização dos problemas 10 e 11. A raiz da hierarquia de generalização é o tipo de entidade *PublicoUniversitario*. A chave primária de *PublicoUniversitario* é *CPFUniv*. Os outros atributos do tipo de entidade *PublicoUniversitario* devem ser os atributos comuns a *Funcionario* e *Aluno*. Você deve renomear os atributos para ser coerente com inclusão no tipo de entidade *PublicoUniversitario*. A hierarquia de generalização deve ser completa e com disjunção.
13. Desenhe um DER contendo os tipos de entidade *Paciente*, *Medico* e *Consulta* ligados por relacionamentos 1-M de *Paciente* a *Consulta* e *Medico* a *Consulta*. Escolha nomes apropriados para os



relacionamentos. Defina cardinalidades mínimas de modo que pacientes e médicos sejam obrigatórios para uma consulta, mas consultas sejam opcionais para pacientes e médicos. Para o tipo de entidade *Paciente*, adicione os atributos *NumPac* (chave primária), *NomePac*, *SobrenomePac*, *EndPac* (endereço), *CidadePac*, *UFPac*, *CEPPac* e *PlanoSaudePac*. Para o tipo de entidade *Medico*, adicione os atributos *NumMed* (chave primária), *NomeMed*, *SobrenomeMed*, *EspecialidadeMed*, *FoneMed*, *EmailMed*, *HospitalMed* e *CertificacaoMed*. Para o tipo de entidade *Consulta*, adicione os atributos *NumConsulta* (chave primária), *DtConsulta*, *FormaPagtoConsulta* (dinheiro, cheque ou cartão de crédito) e *ValorConsulta*. Se você usar o ER Assistant ou outra ferramenta de projeto que tenha suporte a especificação de tipos de dados, escolha os tipos de dados apropriados para os atributos com base em seu conhecimento usual.






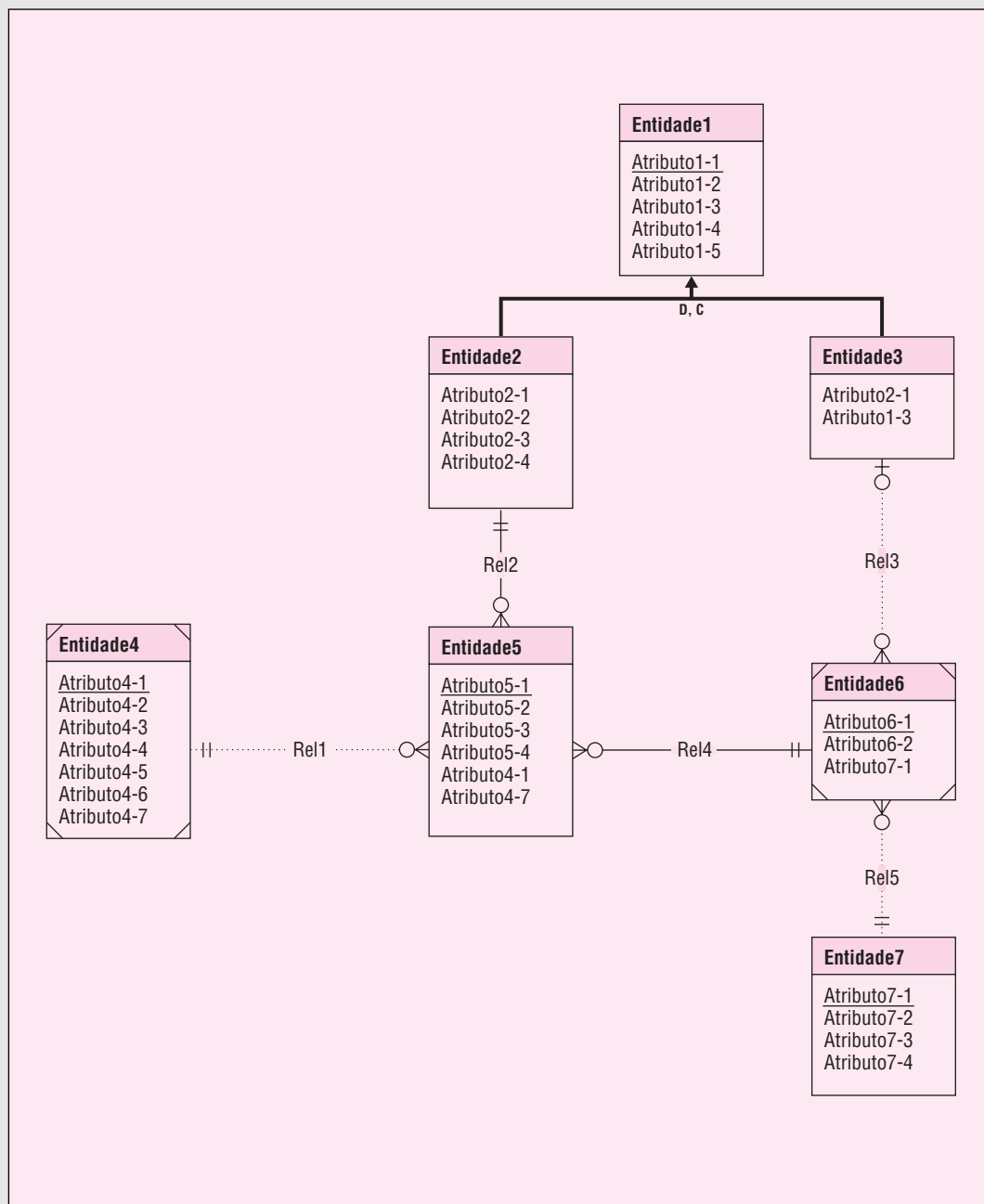
14. Amplie o DER do problema 13 com os tipos de entidade *Enfermeira*, *Item* e *DetalheConsulta* ligados por relacionamentos 1-M de *Consulta* a *DetalheConsulta*, de *Enfermeira* a *DetalheConsulta* e de *Item* a *DetalheConsulta*. O *DetalheConsulta* é uma entidade fraca, sendo o relacionamento 1-M de *Consulta* a *DetalheConsulta* um relacionamento identificador. Escolha nomes apropriados para os relacionamentos. Defina cardinalidades mínimas de modo que uma enfermeira seja opcional para um detalhamento da consulta, que um item seja obrigatório para um detalhamento da consulta e que detalhamento da consulta seja opcional para enfermeiras e itens. Para o tipo de entidade *Item*, adicione os atributos *NumItem* (chave primária), *DescrItem*, *PrecoItem* e *TipoItem*. Para o tipo de entidade *Enfermeira*, adicione os atributos *NumEnf* (chave primária), *NomeEnf*, *SobrenomeEnf*, *TituloEnf*, *FoneEnf*, *EspecialidadeEnf* e *CategoriasalarialEnf*. Para o tipo de entidade *DetalheConsulta*, adicione os atributos *NumDetalhe* (parte da chave primária) e *ValorDetalhe*. Se você estiver usando o ER Assistant ou outra ferramenta de projeto que tenha suporte a especificação de tipos de dados, escolha tipos de dados apropriados para os atributos com base em seu conhecimento usual.
15. Refine o DER do problema 14 com uma hierarquia de generalização consistindo em *Profissional*, *Medico* e *Enfermeira*. A raiz da hierarquia de generalização é o tipo de entidade *Profissional*. A chave primária de *Profissional* é *NumProf* substituindo os atributos *NumMed* e *NumEnf*. Os outros atributos do tipo de entidade *Profissional* devem ser os atributos comuns a *Enfermeira* e *Medico*. Você deve renomear os atributos para ser coerente com inclusão no tipo de entidade *Profissional*. A hierarquia de generalização deve ser completa e com disjunção.
-  16. Verifique seu DER do problema 15 com relação a violações das regras de diagrama. Se você seguiu as instruções do problema, seu diagrama não deve ter erros. Aplique as regras de consistência e de completude a fim de assegurar que seu diagrama não tem erros. Se você estiver usando o ER Assistant, você pode usar a função Check Diagram depois de verificar você mesmo as regras.
-  17. Use seu DER corrigido do problema 16, adicione violações das regras de consistência 3 e 6 a 9. Se você estiver usando o ER Assistant, pode usar a função Check Diagram depois de verificar você mesmo as regras.
-  18. Para cada erro de consistência na Figura 5.P1, identifique a regra de consistência violada e sugira possíveis correções do erro. O DER tem nomes genéricos para que você se concentre nos erros de diagrama encontrados em vez de focar no significado do diagrama. Se você estiver usando o ER Assistant, pode comparar sua solução ao resultado obtido pela função Check Diagram.
-  19. Para cada erro de consistência na Figura 5.P2, identifique a regra de consistência violada e sugira possíveis correções do erro. O DER tem nomes genéricos para que você se concentre nos erros de diagrama encontrados em vez de focar no significado do diagrama. Se estiver usando o ER Assistant, você pode comparar sua solução ao resultado obtido pela função Check Diagram.
-  20. Para cada erro de consistência na Figura 5.P3, identifique a regra de consistência violada e sugira possíveis correções do erro. O DER tem nomes genéricos para que você se concentre nos erros de diagrama encontrados em vez de focar no significado do diagrama. Se estiver usando o ER Assistant, você pode comparar sua solução ao resultado obtido pela função Check Diagram.
21. Desenhe um DER contendo os tipos de entidade *Funcionario* e *Compromisso* ligados por um relacionamento M-N. Escolha um nome de relacionamento apropriado usando seu conhecimento usual das interações entre funcionários e compromissos. Defina cardinalidades mínimas de modo que um compromisso seja opcional para funcionário e um funcionário seja obrigatório para um compromisso. Para o tipo de entidade *Funcionario*, adicione os atributos *NumFunc* (chave primária), *NomeFunc*, *SobrenomeFunc*, *CargoFunc*, *FoneFunc* e *EmailFunc*. Para o tipo de entidade *Compromisso*, adicione os atributos *NumCompr* (chave primária), *AssuntoCompr*, *HoraInicioCompr*, *HoraTerminoCompr* e *AnotacoesCompr*. Para o relacionamento M-N, adicione um atributo

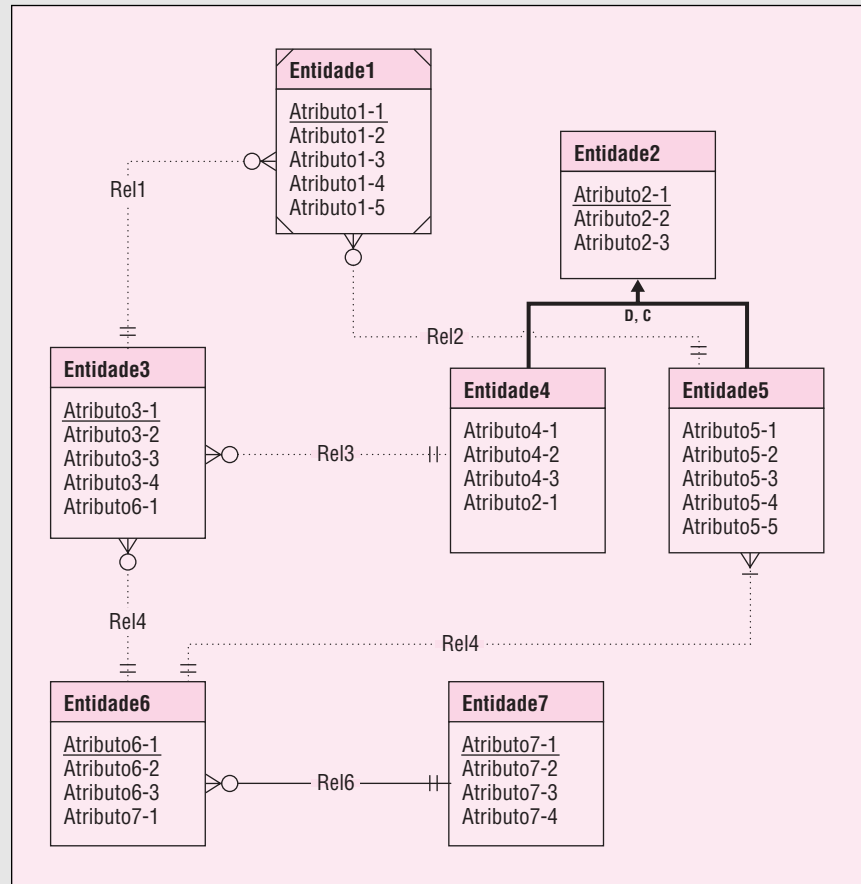
FIGURA 5.P1 DER para o Problema 18



Presenca indicando se o funcionário compareceu ao compromisso. Se você estiver usando o ER Assistant ou outra ferramenta de projeto que tenha suporte à especificação de tipos de dados, escolha os tipos de dados apropriados para os atributos com base em seu conhecimento usual.

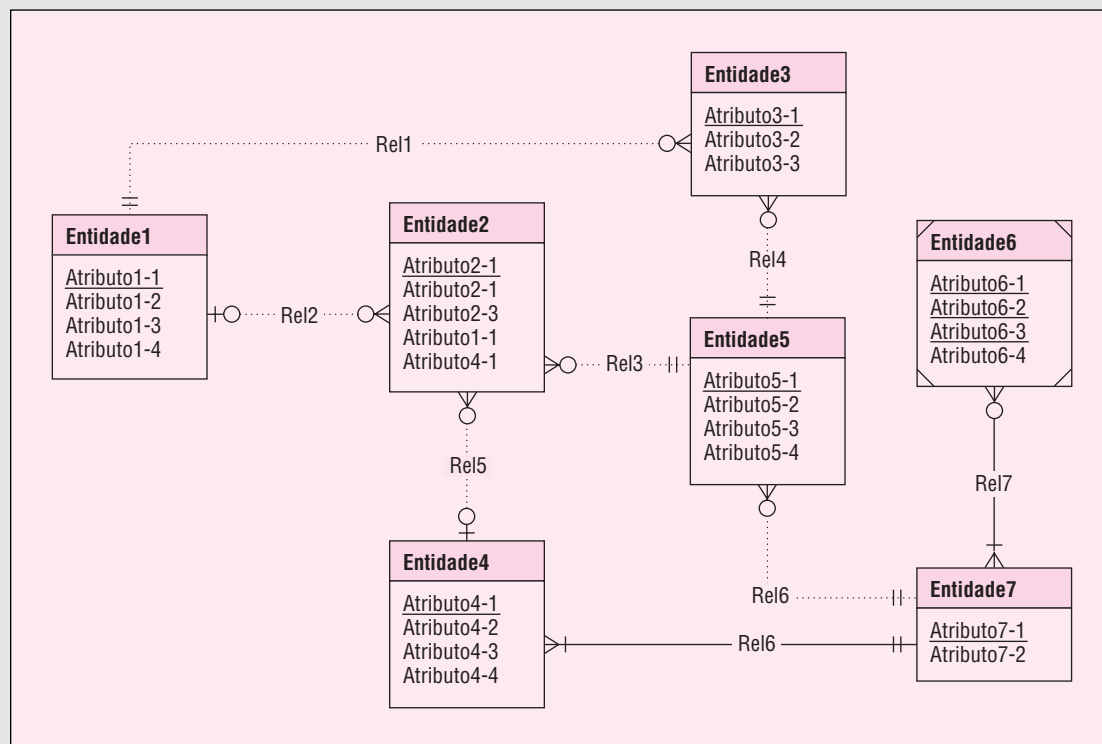
22. Amplie o DER do problema 21 com o tipo de entidade *Local* e um relacionamento 1-M de *Local* a *Compromisso*. Escolha um nome de relacionamento apropriado usando seu conhecimento usual das interações entre locais e compromissos. Defina cardinalidades mínimas de modo que um local seja opcional para um compromisso e um compromisso seja opcional para um local. Para o tipo de entidade *Local*, adicione os atributos *NumLoc* (chave primária), *PredioLoc*, *NumSalaLoc* e *CapacidadeLoc*. Se você estiver usando o ER Assistant ou outra ferramenta de projeto que tenha suporte à especificação de tipos de dados, escolha os tipos de dados apropriados para os atributos com base em seu conhecimento usual.

FIGURA 5.P2
DER para o Problema 19



23. Amplie o DER do problema 22 com o tipo de entidade *Calendario* e um relacionamento M-N de *Compromisso* a *Calendario*. Escolha um nome de relacionamento apropriado usando seu conhecimento usual das interações entre compromissos e calendários. Defina cardinalidades mínimas de modo que um compromisso seja opcional para um calendário e um calendário seja obrigatório para um compromisso. Para o tipo de entidade *Calendario*, adicione os atributos *NumCalend* (chave primária), *DataCalend* e *HoraCalend*. Se você estiver usando o ER Assistant ou outra ferramenta de projeto que tenha suporte à especificação de tipos de dados, escolha os tipos de dados apropriados para os atributos com base em seu conhecimento usual.
24. Revise o DER do problema 23 transformando o relacionamento M-N entre *Funcionario* e *Compromisso* em um tipo de entidade associativa com dois relacionamentos identificadores 1-M.

FIGURA 5.P3 DER para o Problema 20



Referências para Estudos Adicionais

Podemos citar quatro livros especializados em projetos de banco de dados. São eles: Batini, et al. (1992); Nijssen e Halpin (1989); Teorey (1999) e Carlis e Maguire (2001). O site DBAZine (www.dbazine.com) e o site de ajuda de DevX (www.devx.com) tem bastantes conselhos práticos sobre desenvolvimento de banco de dados e modelagem de dados. Se você quiser saber mais detalhes sobre a UML, consulte o UML Center (umlcenter.visual-paradigm.com/index.html) para obter tutoriais e outros recursos.