

Capítulo 2

Modelagem de dados

2.1 Projeto *top down* de bases de dados

Como em muitas aplicações de engenharia, o projeto de uma base de dados através da técnica *top down* ou de refinamentos sucessivos é largamente utilizado. Ele começa pela análise dos requisitos dos usuários finais da BD e da *visão externa* que eles têm sobre os dados. Esta visão e requisitos dependem da aplicação pretendida da BD, variam de um usuário para outro dentro da organização, e refletem suas necessidades para o trabalho diário. Ela é comumente informal e incompleta, em graus que variam com o nível de informatização da aplicação (ou da organização). É comum procedimentos manuais intercalarem-se ou sobreporem-se a procedimentos automatizados via computador. A descrição desses procedimentos aparece em documentos, relatórios e muitas vezes de forma verbal. Se a BD for corporativa, as interfaces com outras áreas ou departamentos da organização são muitas vezes incompletas ou ineficientes. Um exemplo de ineficiência é a emissão de um relatório por um departamento A para um departamento B, onde B redigita os dados do relatório para integração com seus próprios dados e processamento posterior.

Os objetivos finais dessa análise são: (i) obter uma visão unificada de todos os dados da aplicação, e (ii) definir os *procedimentos funcionais* para operar com os dados. É, portanto, uma sistemática similar à de *análise de sistemas* convencional. Esta visão unificada dos dados é comumente chamada de *modelagem de dados* e corresponde a uma abstração do mundo real contendo o conjunto de informações sobre o mesmo que julgamos importante armazenar e manipular. O projeto *top down* da BD através da *modelagem de dados* consiste em especificar os dados através de refinamentos sucessivos, mapeando os dados definidos num nível mais alto e abstrato para o nível seguinte, menos abstrato e mais detalhado.

No nível mais alto a visão e requisitos da BD ainda é informal e é normal-

mente apresentada sob a forma de documentos textuais. Vamos denominá-la de *visão externa dos dados* ou *modelagem externa de dados*.

O próximo passo consiste em mapear esta visão externa dos dados para um *modelo conceitual de dados* de nível suficientemente alto para esconder detalhes de implementação, mas ao mesmo tempo suficientemente detalhado para descrever os diversos tipos de dados requeridos pela aplicação, os seus inter-relacionamentos e algumas regras de consistência. Esse nível é chamado de *modelagem* ou *projeto conceitual de dados*.

Em 1976 uma forma gráfica e sucinta de modelagem conceitual foi proposta por [Chen76] e denominada de *modelo entidade-relacionamento (MER)*. Ela teve grande aceitação por ser um meio de comunicação do projeto conceitual de fácil compreensão por usuários finais da BD (possivelmente leigos), e ao mesmo tempo ser capaz de formalizar vários aspectos do projeto. A modelagem conceitual utilizando o MER consiste em se projetar uma série de *diagramas entidade-relacionamento* que descrevem os dados da BD e os seus inter-relacionamentos. Seu estudo será o objetivo deste capítulo.

O próximo nível consiste na *especificação lógica* dos dados num formato adequado ao SGBD escolhido para a implementação da BD. Ele é chamado de *projeto lógico de dados* e, dependendo do SGBD escolhido, pode ser de nível suficientemente alto para esconder a maioria dos detalhes de implementação. Fazendo uma analogia com linguagens de programação, ele corresponde, aproximadamente, ao projeto das estruturas de dados de um programa, onde os tipos de dados são completamente definidos.

O mapeamento de *diagramas entidade-relacionamento* para o nível lógico pode ser feito através de ferramentas CASE sofisticadas ou manualmente. Quando estudarmos o modelo relacional veremos que este mapeamento é relativamente simples de ser feito manualmente.

O nível lógico situa-se na fronteira entre a modelagem de dados que é independente do SGBD e aquela que é específica ao SGBD.

O último nível é denominado de *projeto físico dos dados* e corresponde à organização interna do armazenamento dos dados pelo SGBD e à definição de estruturas de dados auxiliares visando uma maior eficiência na recuperação e manipulação dos dados. Dependendo do SGBD uma parte considerável do nível físico fica escondida das aplicações.

A modelagem de dados normalmente não contempla o mapeamento dos *procedimentos funcionais* (vistos na visão externa acima) em programas para manipular a BD, embora esta seja uma das atividades mais complexas do projeto completo de uma aplicação de BD.

A especificação dos dados de uma BD é comumente chamada de *esquema dos dados* e também se pode chamar os níveis vistos acima de *esquema conceitual*, *esquema lógico* e *esquema físico* ou *esquema interno* da BD. É importante distinguir o *esquema de uma BD*, que reflete o projeto e a especificação

atributo determinante deve ser garantida para qualquer conteúdo futuro do CE e não apenas para a instância atual do CE.

Resumindo: um CE fica especificado no modelo entidade-relacionamento, dado o nome do CE, os nomes dos atributos do CE e dentre esses o nome do atributo determinante. Uma forma textual de definir um CE poderia ser, por exemplo:

Funcionários(numf, RG, CPF, nome, endereço, salário)

No exemplo acima o atributo determinante *numf* aparece sublinhado. A forma gráfica de representar um conjunto de entidades consiste em colocar o seu nome dentro de um retângulo e os nomes de cada atributo no final de uma aresta saindo do retângulo ou dentro de uma oval no final da aresta,¹ conforme mostra a Figura 2.2:

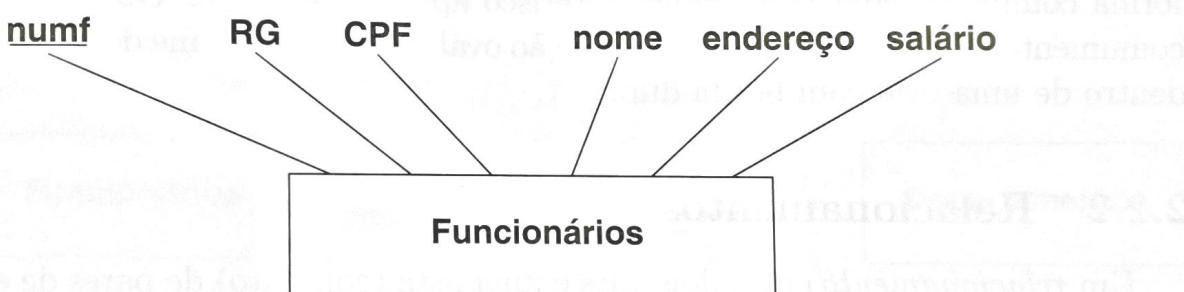


Figura 2.2: Representação de um conjunto de entidades no MER

2.2.1 Tipos de atributos

Um atributo pode conter vários subatributos. Nesse caso ele se diz *composto*. Por exemplo, o atributo “endereço” pode conter os subatributos Local, Cidade e CEP. Local, por sua vez poderia conter os subatributos Rua, Número e Complemento. O valor de endereço é a lista dos valores dos subatributos Rua, Número, Complemento, Cidade e CEP. A representação gráfica desse atributo composto é mostrada na Figura 2.3, adiante.

Se um atributo de uma entidade pode tomar diversos valores então ele se diz *multivalorado*. Por exemplo, um funcionário pode ter diversos telefones que constituem um atributo multivalorado do CE Funcionários. Outro exemplo: um livro pode ter diversos autores e tratar de diversos assuntos. Esta propriedade pode ser indicada colocando-se um asterisco * após os nomes

dos dados (em geral fixos), do conteúdo específico da BD, que é chamado de *instância da BD*, normalmente variável no tempo.

A Figura 2.1 mostra os níveis da abordagem *top down* no projeto de uma BD.

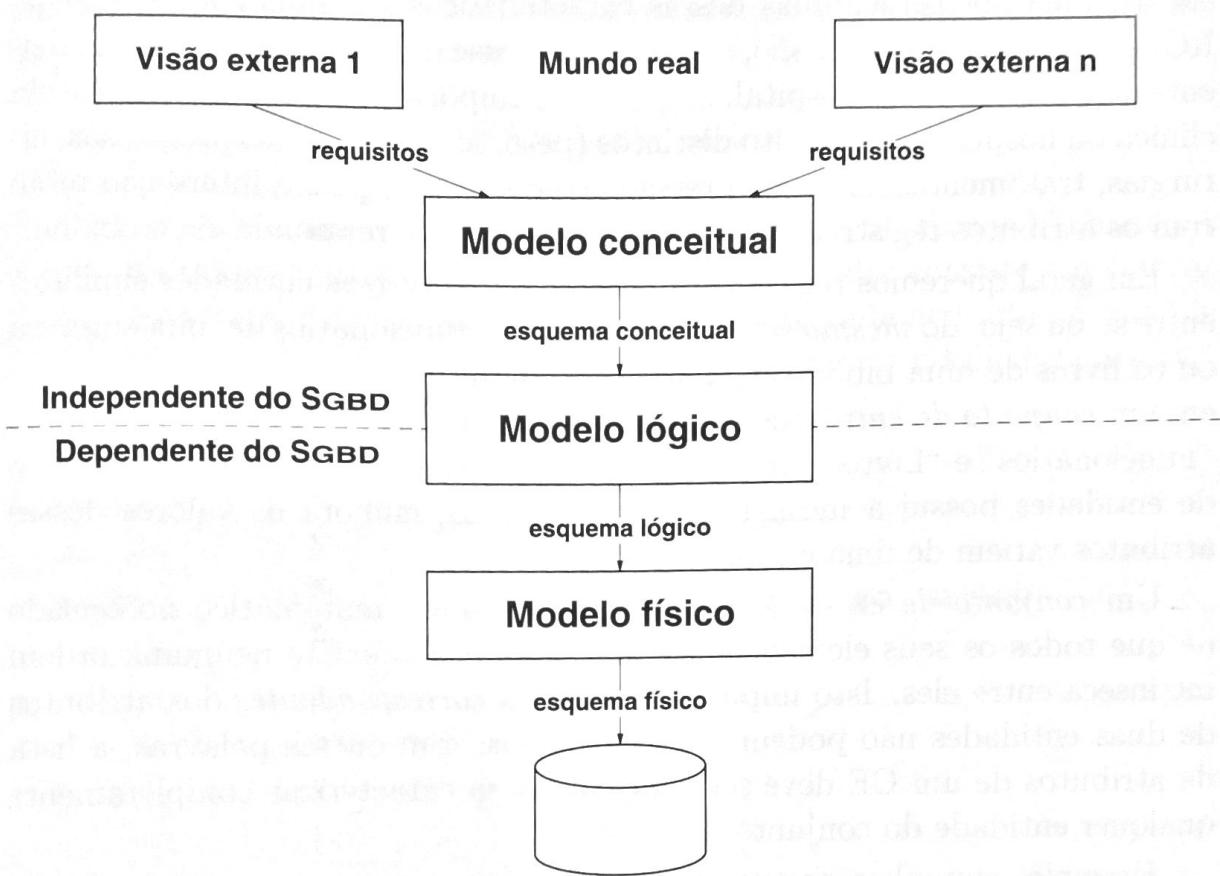


Figura 2.1: Projeto *top down* de uma base de dados

2.2 O modelo entidade–relacionamento

Uma *entidade* é um objeto ou ente do mundo real que possui existência própria e cujas características ou propriedades desejamos registrar. Ela pode ter uma existência física (uma pessoa, um carro, um livro, uma peça) ou abstrata (um departamento, um projeto, um curso).

Uma entidade é caracterizada por algumas propriedades específicas que achamos importante registrar (para os objetivos principais da base de dados) e que serão denominadas de *atributos*. Cada atributo de uma entidade possui um nome e um *valor* específico para a entidade. Por exemplo, o atributo *peso* da entidade *Ana Paula* poderia ter o valor 55 kg.

Entidades podem ter uma miríade de propriedades, especialmente as que têm existência física, mas apenas algumas são relevantes para a aplicação pretendida da base de dados. Por exemplo, um indivíduo que é funcionário

de uma empresa tem características físicas distintivas (peso, altura, sexo, cor da pele, cor do cabelo, idade), sociais (ascendência, descendência, riqueza, posição social), mentais (QI, formação escolar, línguas etc.), habilidades esportivas, culturais, psicológicas etc. Para o setor de RH da empresa onde ele trabalha apenas algumas dessas características são importantes (nome, RG, CPF, endereço, profissão, sexo etc.). Se esse indivíduo passar a ser paciente de uma clínica ou hospital, os atributos importantes para os registros da clínica ou hospital serão muito distintos (peso, altura, doenças pregressas, cirurgias, tratamentos, alergias, pressão arterial etc.) e pouca interseção terão com os atributos registrados no setor de RH da empresa onde ele trabalha.

Em geral queremos registrar os atributos de diversas entidades similares entre si, ou seja, do *mesmo tipo*; por exemplo, os funcionários de uma empresa ou os livros de uma biblioteca. Para este fim agrupamos entidades similares em um *conjunto de entidades* para o qual damos um nome significativo como “Funcionários” e “Livros” no exemplo acima. Cada entidade de um conjunto de entidades possui a mesma lista de atributos, embora os valores desses atributos variem de uma entidade para outra.

Um *conjunto de entidades* (CE) é um conjunto matemático no sentido de que todos os seus elementos são distintos, e não existe nenhuma ordem intrínseca entre eles. Isto implica que *valores correspondentes* dos atributos de duas entidades não podem ser *todos iguais*. Em outras palavras, a lista de atributos de um CE deve ser suficiente para caracterizar completamente qualquer entidade do conjunto.

Exemplo: suponhamos que queremos registrar para um conjunto de pérolas as seguintes informações: cor, diâmetro, peso, lote; elas podem não ser suficientes para distinguir duas pérolas que podem ter os mesmos valores para cor, diâmetro, peso e lote. Se quisermos que essas pérolas façam parte de um conjunto de entidades, algumas outras propriedades teriam que ser incluídas, como local da extração, empresa, pescador etc. Um exemplo mais realista é o caso de itens fabricados em série cujos atributos mensuráveis são idênticos; nesse caso é comum distingui-los através de um *número de série* único impresso no item.

Estas considerações nos levam ao ponto principal na especificação de um CE que é o conceito de *atributo determinante* (ou *atributo chave*). Um *atributo determinante* de um CE é um dos atributos do CE especialmente projetado para identificar de forma única qualquer entidade do CE, ou seja, dadas duas entidades quaisquer e_1 e e_2 do CE em questão, os valores do atributo determinante são distintos para e_1 e e_2 . Em alguns casos práticos é conveniente estender esta definição de forma que um subconjunto dos atributos do CE seja o *atributo determinante* do CE como no exemplo das pérolas visto acima. Outros exemplos serão vistos posteriormente. É importante enfatizar a expressão acima *especialmente projetado*, porque a unicidade do valor do

departamento, existem N funcionários nele lotados (indicado pelo símbolo ‘N’ ao lado do CE Funcionários)”.



Figura 2.4: Relacionamento N : 1

Se uma entidade de um CE só pode estar associada a uma única entidade de outro CE e vice-versa, dizemos que o relacionamento é de cardinalidade 1 para 1. Por exemplo, Departamentos são gerenciados por Funcionários de tal forma que dado um departamento existe um único funcionário que o gerencia e dado um funcionário ele pode ser o gerente de um único departamento (ou de nenhum). Essas regras estão explicitadas na Figura 2.5.



Figura 2.5: Relacionamento 1 : 1

A cardinalidade de um relacionamento é dita N : N (ou muitos para muitos, ou N : M) quando uma entidade de um CE pode estar associada a várias outras entidades de outro CE e vice-versa. Por exemplo, um funcionário pode participar de diversos projetos e um projeto pode ter a participação de diversos funcionários. Esse relacionamento está representado na Figura 2.6 e tem o nome *Participações*.



Figura 2.6: Relacionamento N : N (muitos para muitos)

Muitas vezes queremos explicitar que *todas* as entidades de um CE devem fazer parte de um dado relacionamento com outro CE. Nesse caso diremos

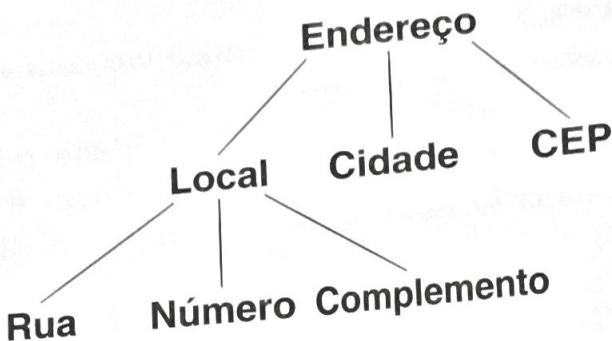


Figura 2.3: Exemplo de atributo composto

Figura 2.3: Exemplo de atributo composto
dos atributos autores e assuntos. O CE Livros poderia então, ser especificado assim:

Livros (num-livro, autores, assuntos*)*

Atributos multivalorados podem ser representados num diagrama ER de forma compacta, colocando-se um asterisco após o seu nome. Outra forma comumente usada quando se usa a notação oval, é colocar o nome do atributo dentro de uma oval com borda dupla.

2.2.2 Relacionamentos

Um *relacionamento* entre dois CEs é uma lista (conjunto) de pares de entidades, onde cada par representa uma associação entre uma entidade de um CE com outra entidade do outro CE e que estabelece uma interdependência entre essas entidades. Por exemplo, podemos representar as “Lotações” de Funcionários em Departamentos através de uma lista de pares ordenados $(f_i, d_j), (f_k, d_l), \dots$, com o significado: o funcionário f_i está lotado no departamento d_j , f_k está lotado no departamento d_l , e assim por diante. Sob o ponto de vista matemático, esse relacionamento é representado por um subconjunto do produto cartesiano Funcionários x Departamentos, que contém todos os possíveis pares (funcionário, departamento).

Neste exemplo, vários funcionários podem estar lotados num único departamento. Dizemos então que o relacionamento “Lotações” de Funcionários com Departamentos é de *cardinalidade* ou *multiplicidade* N para 1 ou N : 1.

A representação gráfica desse relacionamento é feita através de um losango contendo o nome do relacionamento e unindo ao losango, via segmentos de reta, os CEs Funcionários e Departamentos conforme mostra a Figura 2.4. A cardinalidade do relacionamento está indicada pelos símbolos N e 1. Para determinar sua posição correta pode-se usar o seguinte argumento: “dado um funcionário existe apenas um departamento no qual ele pode estar lotado (indicado pelo símbolo ‘1’ ao lado do CE Departamentos), e dado um

que o relacionamento é *total* entre o primeiro CE e o segundo CE. Por exemplo, no relacionamento Lotações visto acima, podemos requerer que todo funcionário esteja lotado em um departamento. Isto seria indicado por uma marca oval ou circular conforme mostra a Figura 2.7. A ausência desta marca no ângulo do losango voltado para Departamentos indica que o relacionamento de Departamentos com Funcionários é *parcial*. Isto permitiria, por exemplo, um novo departamento ser criado sem ter funcionários. Este é um exemplo de uma *regra de consistência* ou *regra de negócio* suportada pelo MER.



Figura 2.7: Relacionamento total

Relacionamentos podem ter atributos. Por exemplo, o relacionamento N : N para indicar a associação de Materiais com seus Fornecedores pode indicar para cada par (m_i, f_j) do relacionamento, o preço, o prazo e o lote (quantidade) que o fornecedor f_j estabelece para fornecer o material m_i . A Figura 2.8 mostra a representação gráfica desse relacionamento.

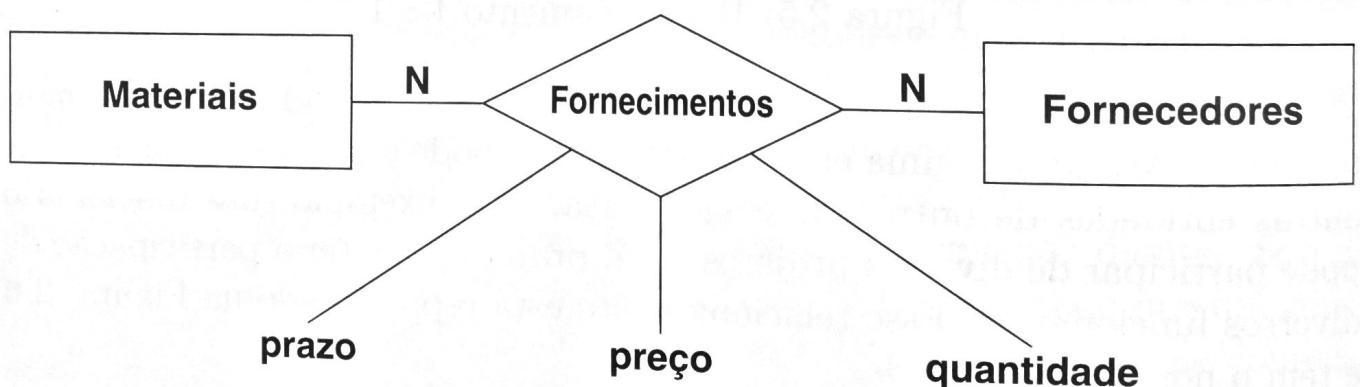


Figura 2.8: Relacionamento com atributos

Poderíamos questionar nesse exemplo se um dado atributo, digamos, preço, pertence de fato ao relacionamento em vez de a um dos conjuntos de entidades envolvidos. Um teste simples é capaz de esclarecer a dúvida. Fixe o material e varie o fornecedor: se o preço varia, então o atributo não é do material; em seguida, fixe o fornecedor e varie o material: se o preço varia, então o atributo não é do fornecedor. Fica claro, nesse caso, que o atributo preço é do relacionamento, isto é, cada par (material, fornecedor) possui um preço. Idem para os atributos prazo e lote.

2.2.3 Conjuntos de entidades fracos

Há casos em que a existência de um CE está vinculada à existência de outro CE. Um exemplo típico é o registro, para fins de seguro-saúde ou imposto de renda, dos dependentes de um funcionário. Nesse caso o registro só faz sentido para a empresa porque o dependente está *ligado* ao funcionário. Diz-se, então, que o CE “Dependentes” é um *conjunto de entidades fraco*. O CE Funcionários é às vezes chamado de *conjunto pai* ou *conjunto mestre* e Dependentes é às vezes chamado de *conjunto detalhe*.

CEs fracos não possuem um atributo determinante *per se*, mas em geral utilizam o atributo determinante do *conjunto pai* para construir o seu atributo determinante. Por exemplo, o atributo determinante de Dependentes poderia ser o par (numf, nome-dependente).

CEs fracos são representados no MER por retângulos de borda dupla e seu relacionamento com o *conjunto pai* por um losango também de borda dupla. A Figura 2.9 mostra o diagrama ER para esse exemplo.

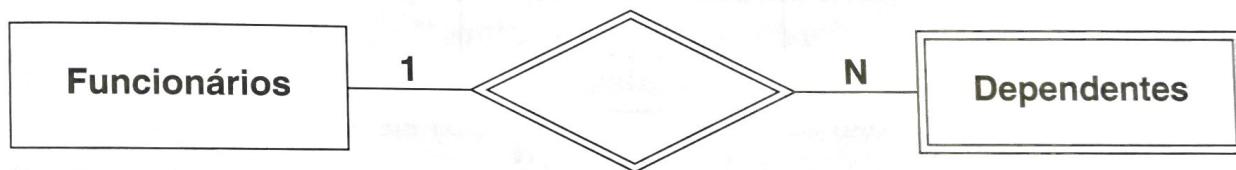


Figura 2.9: Exemplo de conjunto de entidades fraco

2.2.4 Auto-relacionamentos

Muitas vezes queremos fazer o relacionamento de um CE consigo mesmo. Por exemplo, dado o CE “Peças” queremos saber quais peças são componentes de uma dada peça ou, dada uma peça quais peças a têm como componente. Esses dois relacionamentos podem ser representados pelo diagrama da Figura 2.10. Observe que cada uma das ligações do losango com o CE Peças recebeu um rótulo. O primeiro rótulo significa: “uma peça é componente” de outra peça, e o segundo rótulo significa “uma peça tem como componente” outra peça. Os rótulos das ligações explicitam o *papel* (em inglês: *role*) que a peça desempenha no relacionamento. Este papel é normalmente evidente nos relacionamentos normais, mas precisa ser explicitado nos auto-relacionamentos. Observe também que o relacionamento é de cardinalidade $N : N$, isto é, uma peça pode ter vários componentes e uma dada peça pode ser componente de várias peças, conforme mostra a Figura 2.10.

Consideremos um outro exemplo onde temos um auto-relacionamento entre funcionários com os seguintes *papéis*: o primeiro, “Gerencia”, de cardi-

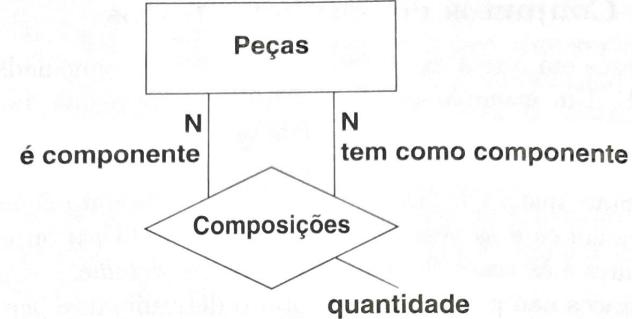


Figura 2.10: Exemplo de auto-relacionamento

nalidade 1 : N, significa que um funcionário pode ser o gerente de vários funcionários e o segundo, “É gerenciado”, de cardinalidade N : 1, significa que vários funcionários podem ser gerenciados por um gerente (Figura 2.11).

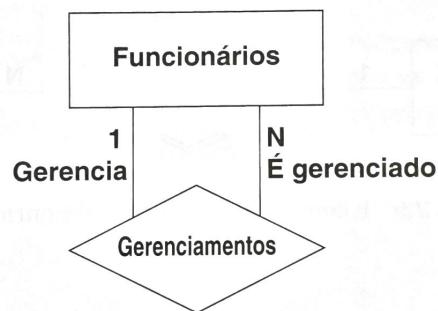


Figura 2.11: Outro exemplo de auto-relacionamento

Os exemplos vistos até agora são de relacionamentos envolvendo dois CEs. Eles são ditos binários ou de grau 2 e são os mais comuns na prática. O *grau* de um relacionamento é o número de CEs envolvidos no relacionamento. A Figura 2.12 mostra um relacionamento de grau 3, ou triplo, entre Professores, Alunos e Disciplinas.

A cardinalidade desse relacionamento, 1 : N : N, pode ser interpretada da seguinte forma:

- dado um professor e uma determinada disciplina temos diversos alunos;
- dado um professor e um determinado aluno temos diversas disciplinas;
- dado um aluno e uma certa disciplina, temos um único professor.

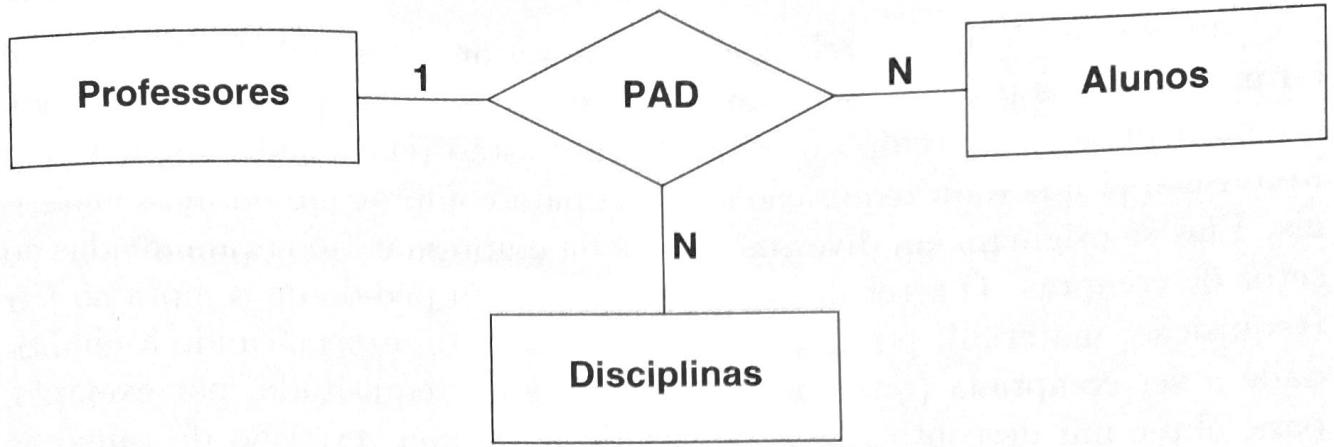


Figura 2.12: Exemplo de relacionamento triplo

A Figura 2.13 mostra um relacionamento triplo chamado MRP entre Materiais, Pedidos e Requisições, típico de um sistema de compras de materiais, onde requisições vindas de diversos setores são agrupadas em pedidos ou ordens de compra e se deseja relacionar cada pedido com as requisições originais. O relacionamento possui dois atributos: quantidade requisitada e quantidade pedida. O relacionamento tem cardinalidade N : N : N e é total do lado de Pedidos e de Requisições (toda requisição está associada a um ou mais pedidos e pode conter vários materiais, idem para pedidos), mas é parcial do lado de Materiais (um dado material pode não estar sendo comprado no momento).

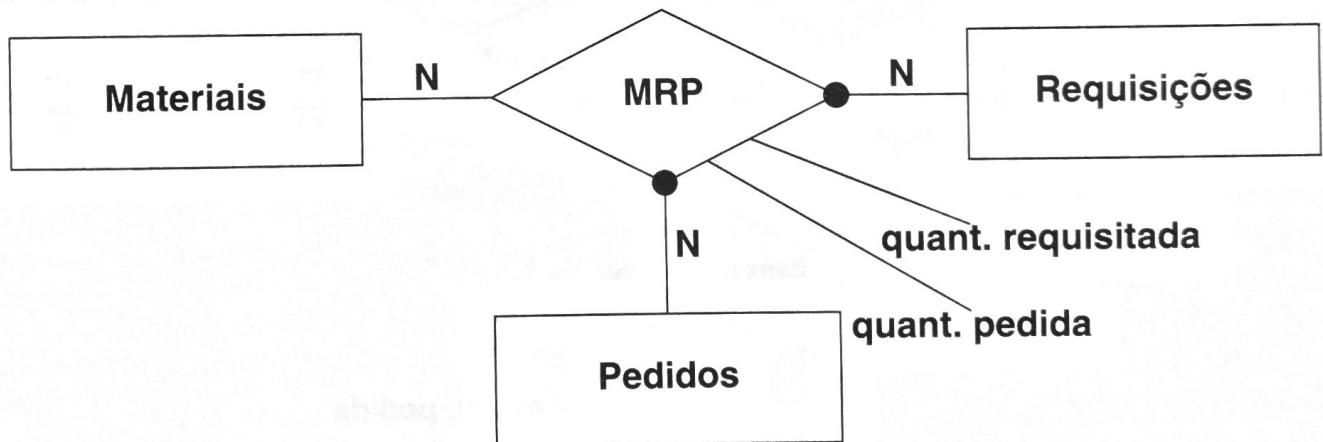


Figura 2.13: Outro exemplo de relacionamento triplo

2.2.5 Extensões ao modelo entidade-relacionamento Agregações

Há casos em que relacionamentos de grau superior a 2 não capturam as *regras de negócio* desejadas. Por exemplo, no relacionamento triplo MRP visto anteriormente entre Materiais, Requisições e Pedidos (ordens de com-

Especialização

As técnicas de orientação a objetos tiveram várias influências sobre o projeto de bases de dados. Uma delas é o conceito de subclasse e herança. Muitas vezes queremos registrar características especiais de certos subconjuntos de um CE. Por exemplo, no CE Funcionários temos secretárias, técnicos, engenheiros, gerentes etc., e para cada uma dessas categorias queremos guardar alguns atributos específicos, como habilidades das secretárias em digitação, informática, línguas etc. Outros atributos seriam requeridos para engenheiros e assim por diante. Seria ineficiente estender o CE Funcionários com todos esses atributos que só teriam valores para cada grupo específico de funcionários ficando vazios os não correspondentes ao grupo. Para este fim criamos os CEs denominados Secretárias, Técnicos, Engenheiros etc., e um tipo especial de relacionamento desses CEs com Funcionários chamado *é-um* (do inglês *is-a*) e representado por um triângulo conforme mostra a Figura 2.15.

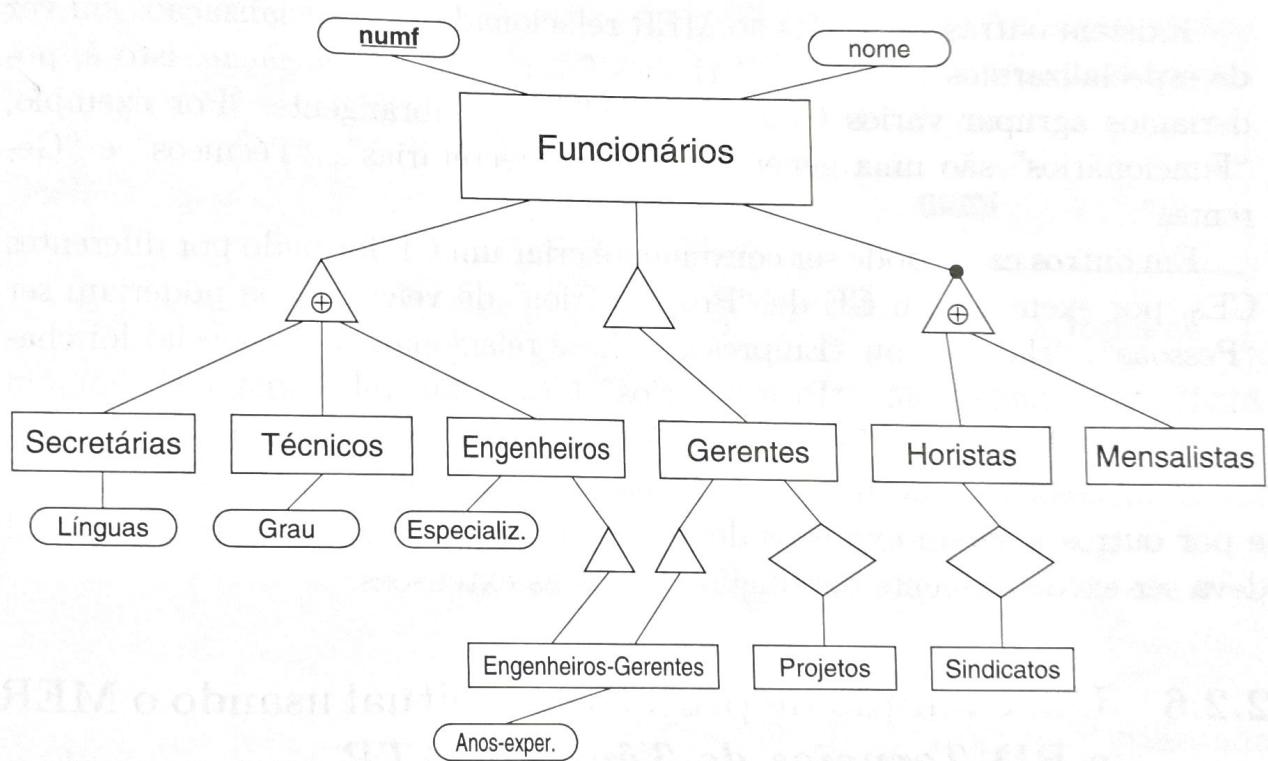


Figura 2.15: Exemplo de especialização

Esses novos CEs são também chamados de *especializações* do CE Funcionários. Como o nome *é-um* indica, o relacionamento significa que uma secretária “é uma” funcionária, isto é, possui todos os atributos do CE Funcionários. Além disso secretárias possuem atributos específicos que aparecem no diagrama ligados ao CE Secretárias. Este é o mesmo conceito de herança em sistemas orientados a objeto.

Este processo de especialização pode ser estendido recursivamente, se preciso: por exemplo, Engenheiros poderiam ser subdivididos em *Engenheiros-*

pra), uma requisição está relacionada com um ou mais materiais e com um ou mais pedidos (ou com nenhum deles), o que é artificial. Requisições precedem a pedidos no tempo e originalmente estão relacionadas apenas com Materiais, já que toda requisição é criada para comprar um ou mais materiais. Elas se originam em diversos setores da empresa e são encaminhadas ao setor de compras. O setor de compras associa um pedido de compra ao par (requisição, material), portanto, ao relacionamento, especificando a quantidade a ser comprada (que pode ser diferente da requisitada, por exemplo, para obter um desconto maior ou devido a alguma restrição de tamanho de lote de venda). Esta separação de funções implica na existência de dois relacionamentos distintos; o segundo é chamado de *agregação* porque o relacionamento de Materiais com Requisições é *agregado* em um pseudo CE, que por sua vez se relaciona com Pedidos através do relacionamento “Itens de pedidos”. A Figura 2.14 mostra como representar esse conceito num diagrama ER.

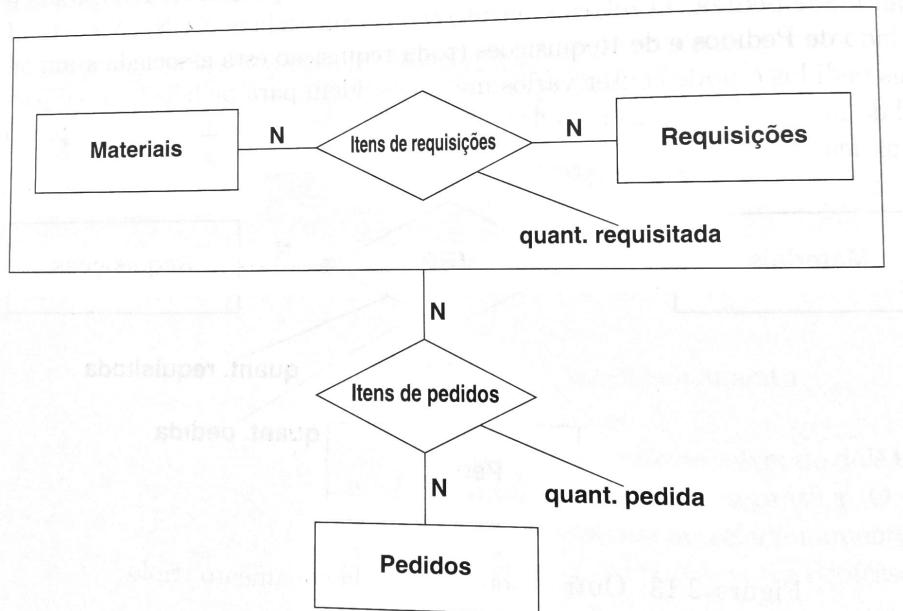


Figura 2.14: Exemplo de agregação

Posteriormente, quando estudarmos o mapeamento do modelo conceitual para o modelo relacional, veremos que relacionamentos N : N serão representados por *tabelas auxiliares* e, portanto, se comportam como CEs, o que torna muito simples representar esse relacionamento especial no modelo relacional.

para cada um dos atributos dos diversos CEs etc.

2.2.7 Convenções para nomes — considerações práticas

A atribuição de nomes para CEs e relacionamentos deve ser feita de forma a exprimir a semântica (ou seja, o significado) dos objetos sendo definidos. Nomes apropriados para CEs aparecem naturalmente na modelagem. Nomes apropriados para CEs aparecem naturalmente na modelagem. Nomes apropriados para CEs aparecem naturalmente na modelagem. Nomes apropriados para CEs aparecem naturalmente na modelagem.

No exemplo anterior, a alocação de funcionários em departamentos foi expressa através do termo simétrico ou neutro, “Lotações”, enquanto os papéis seriam expressos por: funcionário “está lotado no” departamento e departamento “emprega” ou “lota” funcionário.

Observe que adotamos a convenção de expressar nomes de CEs e relacionamentos sempre no plural (para enfatizar que são conjuntos contendo vários elementos similares), embora alguns livros adotem a convenção contrária (para enfatizar que o relacionamento se dá entre *uma* entidade de um CE com *uma* entidade do outro CE!). De qualquer forma, a uniformidade da notação é a consideração mais importante para o leitor da documentação do projeto.

No caso de relacionamentos triplos ou quádruplos, na prática menos freqüentes do que relacionamentos binários, dificilmente se consegue um nome significativo, e uma solução de compromisso é usar as iniciais dos CEs participantes e incluir uma boa descrição textual do seu significado, nem sempre óbvio simplesmente olhando o diagrama. Felizmente, quando passarmos do projeto conceitual para o projeto lógico tais dificuldades de interpretação vão desaparecer, pois, como veremos, a representação do relacionamento no modelo lógico esclarece a maior parte das dúvidas sobre o seu significado.

Um último comentário sobre o MER: a principal utilidade da modelagem conceitual utilizando o MER é ser um meio de comunicação do projeto da BD inteligível a usuários leigos, especialmente aqueles que têm a responsabilidade de definir os objetivos e usos da BD, como gerentes, supervisores etc. Esses usuários devem, portanto, ter uma visão clara do projeto e, principalmente, projetista corre o perigo de implantar a aplicação que “não era bem aquela que o usuário queria”. Isto, infelizmente, ocorre na prática com muito mais freqüência do que seria desejável, daí a nossa ênfase nesse ponto.

costuma chamar informalmente de *arquivo mestre* em projetos voltados para arquivos. Contém informações típicas de um cadastro de pessoas. O atributo determinante será denominado *numj*. Outro CE corresponde ao registro de dados cadastrais dos torneios do Grand Slam e do Master Series e que chamaremos de “*Lista_torneios*”, com atributo determinante *numt*, e atributo nome do torneio, país hospedeiro, categoria do torneio, tipo de quadra e número inicial de participantes. Finalmente, queremos representar num relacionamento entre “*Jogadores*” e “*Lista_torneios*”, dados históricos das realizações dos torneios, ou seja, para cada torneio e ano, queremos registrar o jogador que venceu o torneio e o valor do prêmio recebido. Vamos chamar esse relacionamento de “*Vencedores_Torneios*”.

A Figura 2.16 mostra o projeto conceitual através de um diagrama entidade-relacionamento. Observe que um jogador pode ter ganho um certo torneio mais de uma vez, mas dado um torneio e um ano só existe um vencedor. O CE “*Anos_realização*” foi colocado no projeto conceitual unicamente para refletir esta restrição, o relacionamento N : N : 1 entre *Anos_realização*, *Lista_torneios* e *Jogadores*. O diagrama foi intencionalmente simplificado, omitindo os atributos dos CEs envolvidos.

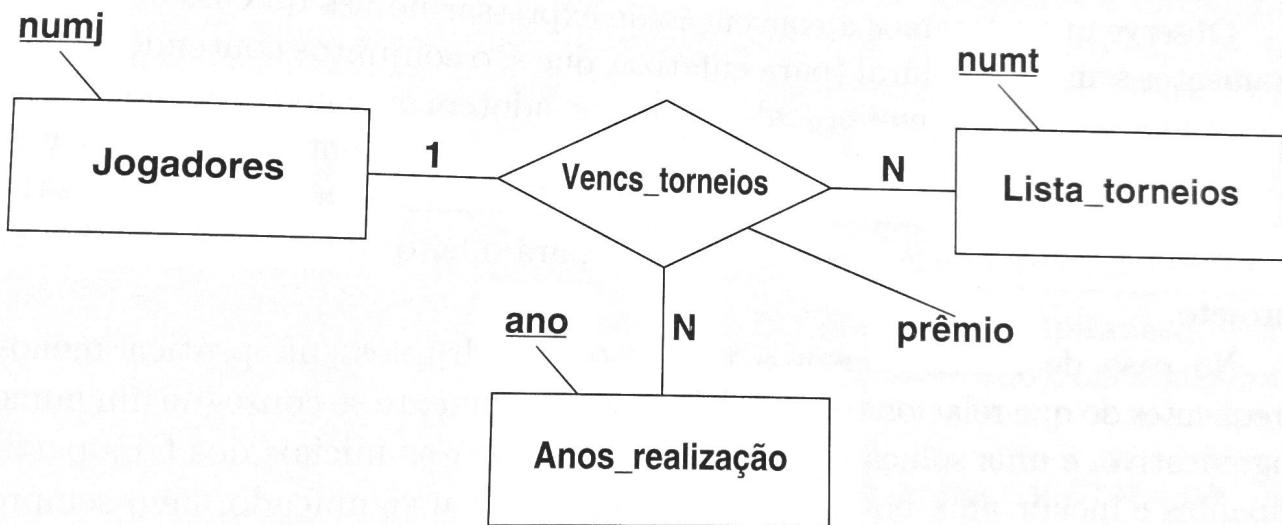


Figura 2.16: Projeto conceitual da BD *Torneios de Tênis da ATP* utilizando o modelo entidade-relacionamento

Embora extremamente simples, várias dúvidas colocadas anteriormente na visão externa foram esclarecidas nesse projeto conceitual, com exceção da que diz respeito a quais dados manter quando um jogador se aposenta e que relatórios devem ser emitidos pelo sistema. A última corresponde a procedimentos funcionais não cobertos nesse nível do projeto. Posteriormente faremos o mapeamento desse projeto conceitual para o *projeto lógico*, voltado para o modelo relacional de BD, onde a primeira dúvida será esclarecida e informações adicionais serão colocadas, como, por exemplo, o tipo e unidade monetária do prêmio concedido ao vencedor de um torneio, os tipos de dados

Gerentes. É possível também a um funcionário pertencer a mais de uma especialização como por exemplo a Engenheiros e Gerentes. Poderíamos até ter herança múltipla como é o caso de Engenheiros-Gerentes no diagrama.

O símbolo \oplus indica que a especialização é do tipo *ou exclusivo*, ou seja, um funcionário pode ser especializado em apenas um dentre “Secretárias”, “Técnicos” e “Gerentes”. Idem para a especialização em “Horistas” e “Mensalistas”. A especialização pode também ser total ou parcial. Por exemplo, suponha que funcionários necessariamente devem ser ou horistas ou mensalistas: isto foi indicado pelo símbolo circular no vértice superior do triângulo. Observe também que certos relacionamentos podem se aplicar apenas a certas especializações como é o caso do relacionamento de Horistas com Sindicatos e o de Gerentes com Projetos.

Quando fizermos o mapeamento do MER para o modelo relacional veremos que é bastante simples representar os relacionamentos *é-um* através de *tabelas auxiliares*.

Existem outras extensões ao MER relacionadas a especialização: em vez de especializarmos um CE poderíamos fazer uma *generalização*, isto é, poderíamos agrupar vários CEs em um CE mais abrangente. Por exemplo, “Funcionários” são uma generalização de “Secretárias”, “Técnicos” e “Gerentes”.

Em outros casos pode ser conveniente criar um CE formado por diferentes CEs, por exemplo um CE de “Proprietários” de veículos que poderiam ser “Pessoas”, “Bancos” ou “Empresas”. Esse relacionamento especial foi chamado de *categorização*. “Proprietários” teriam um relacionamento comum com o CE “Veículos”. Estas extensões são, ao nosso ver, menos úteis, por serem menos claras ao usuário leigo em bancos de dados. Por este motivo, e por outros a serem expostos depois, não cremos que o projeto conceitual deva ser excessivamente detalhado com essas extensões.

2.2.6 Um exemplo de projeto conceitual usando o MER: a BD *Torneios de Tênis da ATP*

Vamos agora detalhar melhor a visão externa da BD *Torneios de Tênis da ATP* vista anteriormente. Após muita interação com o *gerente do portal* que está contratando o projeto, chegamos à conclusão de que inicialmente serão colocadas na BD informações sobre jogadores que venceram torneios do Grand Slam a partir de 1992, e de alguns jogadores brasileiros. Deseja-se também registrar dados cadastrais sobre todos os torneios do Grand Slam e do Master Series, e informações sobre os jogadores que venceram esses torneios a partir de 1992. Chegamos à conclusão de que apenas dois CEs e um relacionamento são necessários: um CE contendo dados cadastrais dos jogadores e que vamos chamar de “Jogadores”. Ele corresponde ao que se