

# Abordagem Entidade Relacionamento

Como vimos no Capítulo 1, a primeira etapa do projeto de um banco de dados é a construção de um modelo conceitual, a chamada *modelagem conceitual*. O objetivo da modelagem conceitual é obter uma descrição abstrata, independente de implementação em computador, dos dados que serão armazenados no banco de dados.

A técnica de modelagem de dados mais difundida e utilizada é a *abordagem entidade-relacionamento (ER)*. Nesta técnica, o modelo de dados é representado através de um *modelo entidade-relacionamento (modelo ER)*. Usualmente, um modelo ER é representado graficamente, através de um *diagrama entidade-relacionamento (DER)*. A abordagem ER foi criada em 1976 por Peter Chen. Ela pode ser considerada como um padrão de fato para modelagem conceitual. Mesmo as técnicas de modelagem orientada a objetos que têm surgido nos últimos anos baseiam-se nos conceitos da abordagem ER.

Este capítulo tem por objetivo apresentar os conceitos centrais da abordagem ER: *entidade, relacionamento, atributo, generalização/especialização e entidade associativa*. Junto com os conceitos, é apresentada uma notação gráfica para diagramas ER. A notação gráfica usada no capítulo é a notação originalmente introduzida por Peter Chen. No Capítulo 3, são discutidas outras notações para representar diagramas ER.

## 2.1 ENTIDADE

O conceito fundamental da abordagem ER é o conceito de *entidade*.

entidade
=
conjunto de objetos <sup>1</sup> da realidade modelada sobre os quais deseja-se manter informações no banco de dados

Uma entidade representa, no modelo conceitual, um conjunto de objetos da realidade modelada. Como o objetivo de um modelo ER é modelar de forma abstrata um BD, interessam-nos somente os objetos sobre os quais deseja-se manter informações. Vejamos alguns exemplos. No sistema de informações industrial que usamos no Capítulo 1, alguns exemplos de entidades poderiam ser os produtos, os tipos de produtos, as vendas ou as compras. Já em um sistema de contas correntes, algumas entidades podem ser os clientes, as contas correntes, os cheques e as agências. Observe que uma entidade pode representar tanto objetos concretos da realidade (uma pessoa, um automóvel), quanto objetos abstratos (um departamento, um endereço<sup>2</sup>).

Em um DER, uma entidade é representada através de um retângulo que contém o nome da entidade. Alguns exemplos são mostrados na Figura 2.1.



Figura 2.1: Representação gráfica de entidades

Como dito acima, cada retângulo representa um *conjunto* de objetos sobre os quais deseja-se guardar informações. Assim, no exemplo, o primeiro retângulo designa o conjunto de todas as pessoas sobre as quais se deseja manter informações no banco de dados, enquanto o segundo retângulo designa o conjunto de todos os departamentos sobre os quais se deseja manter informações. Caso seja necessário referir um objeto particular (uma determinada pessoa ou um determinado departamento) fala-se em *ocorrência* de entidade (alguns autores usam também o anglicismo “*instância*” de entidade).

Há autores que preferem usar o par de termos *conjunto de entidades* e *entidade* para designar respectivamente o conjunto de objetos e cada objeto individual. Essa terminologia é a primeira vista mais adequada pois corresponde ao uso dos termos na linguagem natural, onde “entidade” é um indivíduo e não um coletivo. Entretanto, esta terminologia não é adequada no

---

<sup>1</sup> O termo “objeto” possui aqui a conotação que é lhe dada na linguagem natural, de “coisa, tudo que é perceptível ou manipulável”. Não estamos falando aqui do termo “objeto” como usado na modelagem e na programação orientadas a objetos, onde o termo tem uma conotação mais precisa.

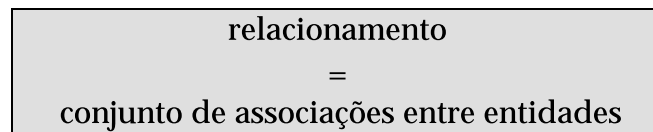
<sup>2</sup> Neste ponto, aquele que já possui conhecimento sobre a modelagem ER poderá estar pensando: “Obviamente endereço é um atributo e não uma entidade!”. Se você é um desses leitores, peço um pouco de paciência, pois esta questão será esclarecida mais adiante.

projeto de BD, no qual falamos com grande frequência sobre conjuntos de objetos e raramente sobre indivíduos. Por esse motivo preferimos usar o par de termos *entidade* e *ocorrência de entidade*.

## 2.2 RELACIONAMENTO

### 2.2.1 Conceituação

Além de especificar os objetos sobre os quais deseja-se manter informações, o DER deve permitir a especificação das propriedades dos objetos que serão armazenadas no BD. Uma das propriedades sobre as quais pode ser desejável manter informações é a associação entre objetos. Exemplificando, pode ser desejável saber quais pessoas estão associadas a quais departamentos em uma organização.



Em um DER, um relacionamento é representado através de um losango, ligado por linhas aos retângulos representativos das entidades que participam do relacionamento. A Figura 2.2 apresenta um DER contendo duas entidades, **PESSOA** e **DEPARTAMENTO**, e um relacionamento, **LOTAÇÃO**.



Figura 2.2: Representação gráfica de relacionamento

Este modelo expressa que o BD mantém informações sobre:

- ☐ um conjunto de objetos classificados como pessoas (relacionamento **PESSOA**)
- ☐ um conjunto de objetos classificados como departamentos (relacionamento **DEPARTAMENTO**)
- ☐ um conjuntos de associações, que ligam um departamento a uma pessoa. (relacionamento **LOTAÇÃO**).

Da mesma forma que fizemos com entidades, quando quisermos nos referir a associações particulares dentro de um conjunto, vamos nos referir a *ocorrências de relacionamentos*. No caso do relacionamento **LOTAÇÃO** uma ocorrência seria um par específico formado por uma determinada ocorrência da entidade **PESSOA** e por uma determinada ocorrência da entidade **DEPARTAMENTO**.

Para fins didáticos, pode ser útil construir um *diagrama de ocorrências*, como o apresentado na Figura 2.3. Este diagrama refere-se ao modelo ER da Figura 2.2. Em um diagrama de ocorrências, ocorrências de entidades são representadas por círculos brancos e ocorrências de relacionamentos por círculos negros. As ocorrências de entidades participantes de uma ocorrência de relacionamento são indicadas pelas linhas que ligam o círculo negro repre-

sentativo da ocorrência de relacionamento aos círculos brancos representativos das ocorrências de entidades relacionadas. Assim, a Figura 2.3 representa que há uma ocorrência de **LOTAÇÃO** que liga a pessoa **p1** com o departamento **d1**. Observe-se que, na forma como está, o modelo da Figura 2.2 não informa quantas vezes uma entidade é associada através de um relacionamento (veremos como isso pode ser representado mais abaixo). O modelo apresentado permite que uma ocorrência de entidade (por exemplo, a pessoa **p3**) não esteja associada a nenhuma ocorrência de entidade através do relacionamento, ou uma ocorrência de entidade (por exemplo, a pessoa **p1**) esteja associada a exatamente uma ocorrência de entidade através do relacionamento, ou ainda, que uma ocorrência de entidade (por exemplo, o departamento **d1**) esteja associada a mais de uma ocorrência de entidade através do relacionamento.

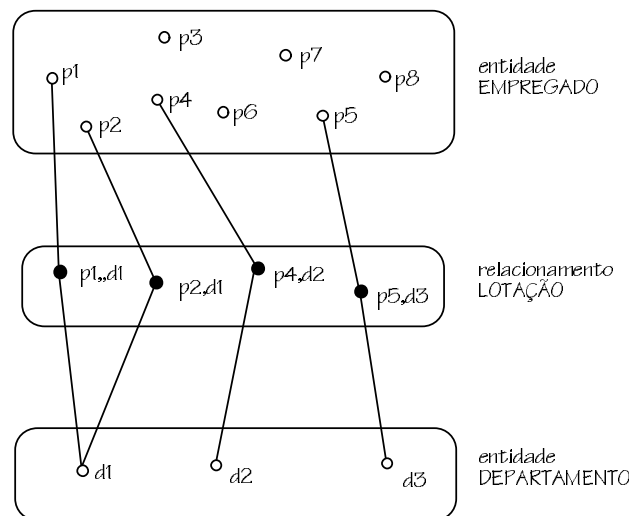


Figura 2.3: Diagrama de ocorrências

Não necessariamente um relacionamento associa entidades diferentes. A Figura 2.4 mostra um DER que contém um *auto-relacionamento*, isto é, um relacionamento entre ocorrências de uma mesma entidade. Neste caso, é necessário um conceito adicional, o de *papel* da entidade no relacionamento. No caso do relacionamento de casamento, uma ocorrência de pessoa exerce o papel de *marido* e a outra ocorrência de pessoa exerce o papel de *esposa*. Papéis são anotados no DER como mostrado na Figura 2.4. No caso de relacionamentos entre entidades diferentes, como o de **LOTAÇÃO** mostrado acima, não é necessário indicar os papéis das entidades, já que eles são óbvios.

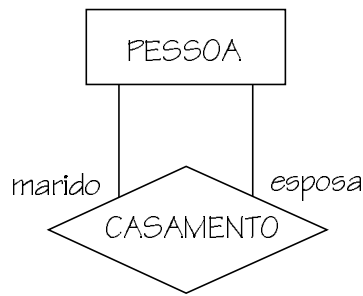


Figura 2.4: Auto-relacionamento

A Figura 2.5 apresenta um possível diagrama de ocorrências para o DER da Figura 2.4. Os papéis (marido e esposa) das ocorrências de entidades em cada ocorrência de relacionamento foram anotadas nas linhas que ligam os círculos representativos das ocorrências de entidades e relacionamentos.

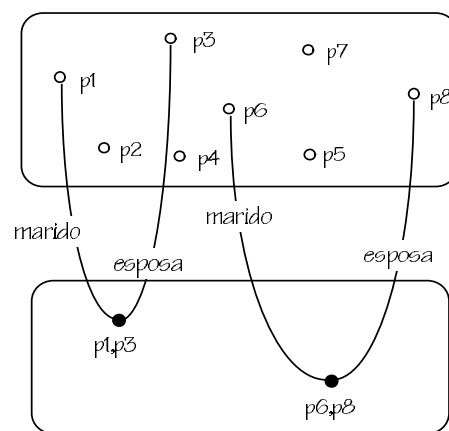


Figura 2.5: Diagrama de ocorrências para o relacionamento casamento

## 2.2.2 Cardinalidade de relacionamentos

Para fins de projeto de banco de dados, uma propriedade importante de um relacionamento é a de quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência através do relacionamento. Esta propriedade é chamada de *cardinalidade* de uma entidade em um relacionamento. Há duas cardinalidades a considerar: a cardinalidade *máxima* e a cardinalidade *mínima*.

cardinalidade (mínima, máxima) de entidade em relacionamento
=
número (mínimo, máximo) de ocorrências de entidade associadas a uma ocorrência da entidade em questão através do relacionamento

### 2.2.3 Cardinalidade máxima

Para exemplificar o conceito de cardinalidade vamos retomar o exemplo da Figura 2.2. Vamos considerar as seguintes cardinalidade máximas:

- ❑ Entidade **EMPREGADO** tem cardinalidade máxima **1** no relacionamento **LOTAÇÃO**:

Isso significa que uma ocorrência de **EMPREGADO** pode estar associada a no máximo uma ocorrência de **DEPARTAMENTO**, ou em outros termos, que um empregado pode estar lotado em no máximo um departamento

- ❑ Entidade **DEPARTAMENTO** tem cardinalidade máxima **120** no relacionamento **LOTAÇÃO**:

Isso significa que uma ocorrência de **DEPARTAMENTO** pode estar associada a no máximo 120 ocorrências de **EMPREGADO**, ou em outros termos, que um departamento pode ter nele lotado no máximo 120 empregados.

Para fins práticos, não é necessário distinguir entre diferentes cardinalidades máximas maiores que 1. Por este motivo, apenas duas cardinalidades máximas são relevantes: a cardinalidade máxima **1** e a cardinalidade máxima “muitos”, referida pela letra **n**. Assim, no exemplo acima, diz-se que a cardinalidade máxima da entidade **DEPARTAMENTO** no relacionamento **LOTAÇÃO** é **n**.

A cardinalidade máxima é representada no DER conforme indicado na Figura 2.6. Observe a convenção usada. À primeira vista, ela pode parecer pouco natural, já que vai anotada “do outro lado” do relacionamento a qual se refere. Exemplificando, a cardinalidade máxima da entidade **EMPREGADO** no relacionamento **LOTAÇÃO** é anotada junto ao símbolo da entidade **DEPARTAMENTO**.

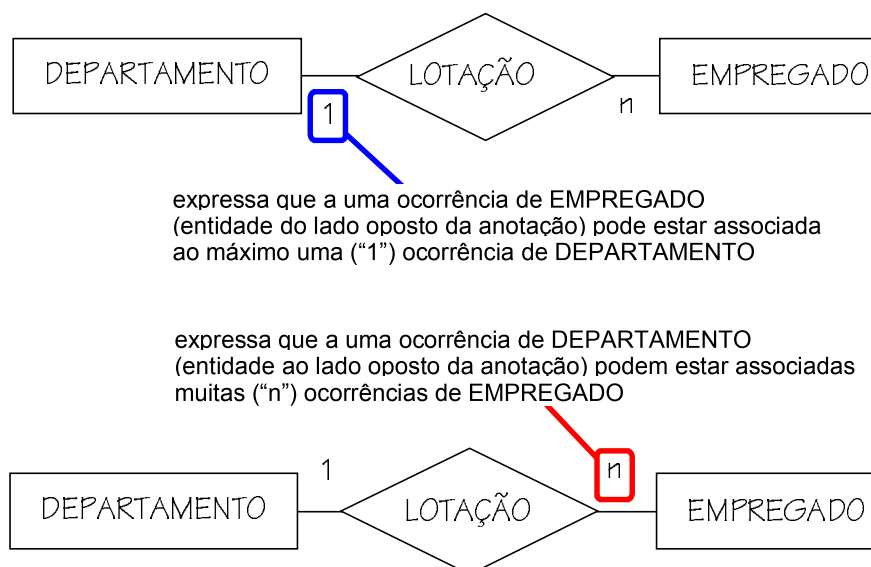


Figura 2.6: Cardinalidade máxima de relacionamento

## 2.2.4 Classificação de relacionamentos binários

A cardinalidade máxima pode ser usada para classificar relacionamentos binários. Um *relacionamento binário* é aquele cujas ocorrências envolvem duas entidades, como todos vistos até aqui. Podemos classificar os relacionamentos em **n:n** (muitos-para-muitos), **1:n** (um-para-muitos) e **1:1** (um-para-um). A Figura 2.7, a Figura 2.8 e a Figura 2.9 apresentam exemplos de relacionamentos com cardinalidades máximas 1:1, 1:n e n:n, respectivamente. A seguir comentamos a interpretação de alguns relacionamentos apresentados nestas figuras.

Na Figura 2.7, no relacionamento **CASAMENTO**, as cardinalidades máximas expressam que uma pessoa pode possuir no máximo um marido (uma instância de pessoa pode estar associada via relacionamento a no máximo outra pessoa no papel de esposa) e no máximo uma esposa.

Observe que este relacionamento, apesar de envolver apenas uma entidade, é também considerado como um relacionamento binário. O que determina o fato de o relacionamento ser binário é o número de ocorrências de entidade que participam de cada ocorrência do relacionamento. De cada ocorrência de **CASAMENTO** participam exatamente duas ocorrências da entidade **PESSOA** (um marido e uma esposa). Por este motivo, o relacionamento de **CASAMENTO** é classificado como sendo binário.

A Figura 2.8 mostra outros exemplos de relacionamentos 1:n, além do relacionamento **LOTAÇÃO** que já havia sido visto acima. O relacionamento **INSCRIÇÃO** modela a inscrição de alunos em uma universidade pública, onde existe a restrição de que um aluno pode estar inscrito em no máximo um curso.

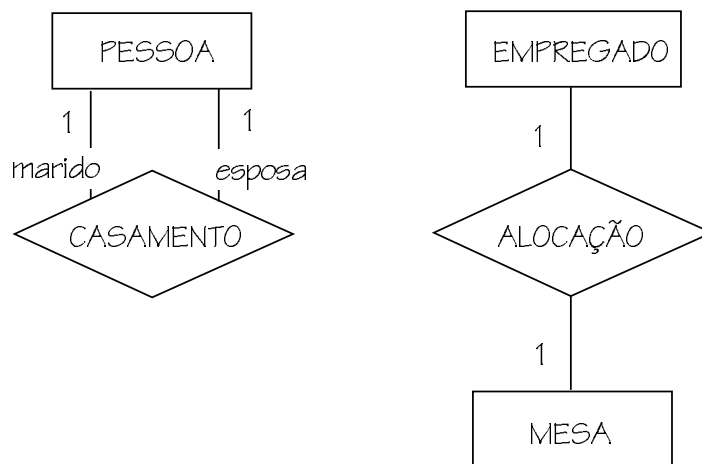


Figura 2.7: Relacionamentos 1:1

O relacionamento **INSCRIÇÃO** da Figura 2.8 representa a associação entre cursos de uma Universidade pública e seus alunos. Por tratar-se de uma universidade pública, cada aluno pode estar vinculado a um curso no máximo.

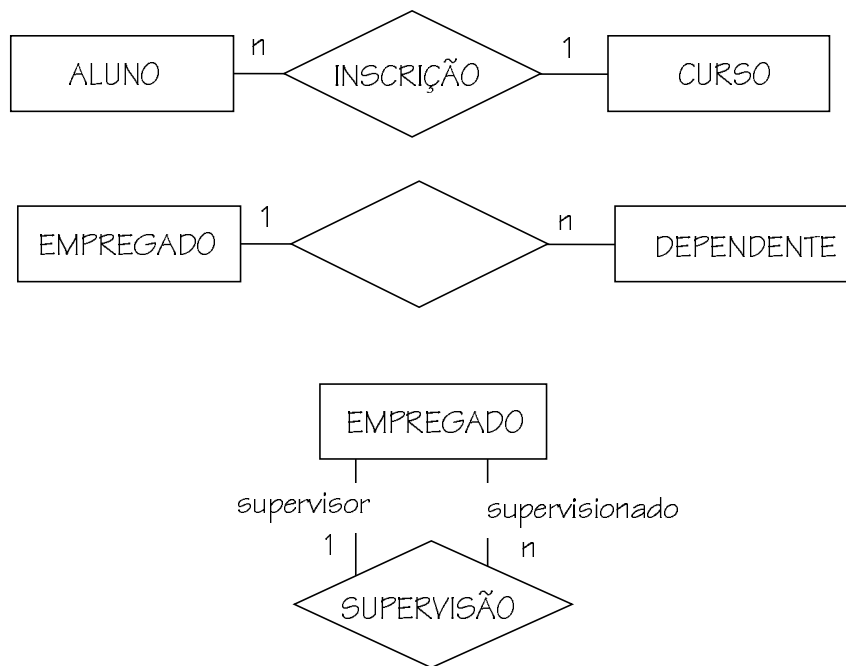


Figura 2.8: Relacionamentos 1:n

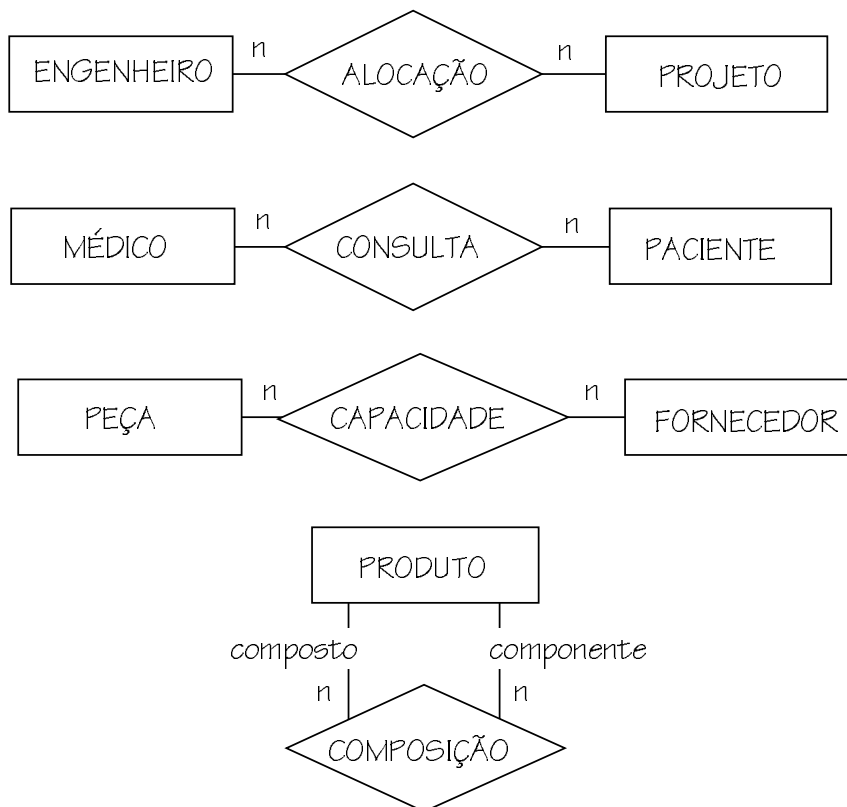


Figura 2.9: Relacionamentos n:n

O relacionamento entre as entidades **EMPREGADO** e **DEPENDENTE** (Figura 2.8) modela a associação entre um empregado e seus dependentes para fins de imposto de renda. Neste caso, um dependente pode estar associado a no máximo um empregado. Cabe observar que no DER, não foi anotado o nome do relacionamento. No caso de no DER não constar o nome do relaci-



onamento, este é denominado pela concatenação de nomes das entidades participantes. Assim, neste caso, o relacionamento é denominado **EMPREGADO-DEPENDENTE**.

O relacionamento **SUPERVISÃO** (Figura 2.8) é um exemplo de auto-relacionamento 1:n. Ele modela a associação entre um empregado (supervisor) e seus supervisionados imediatos. A cardinalidade máxima expressa que um empregado pode possuir no máximo um supervisor, mas muitos supervisionados.

O tipo menos restrito de relacionamento é o de cardinalidade **n:n**. A Figura 2.9 apresenta alguns relacionamentos deste tipo, inclusive o de um auto-relacionamento.

### 2.2.5 Relacionamento ternário

Todos exemplos até aqui mostrados são de relacionamentos binários, ou seja, de relacionamentos que associam exatamente duas entidades. A abordagem ER permite que sejam definidos relacionamentos de grau maior do que dois (relacionamentos ternários, quaternários,...). O DER da Figura 2.10 mostra um exemplo de um relacionamento ternário.

Cada ocorrência do relacionamento **DISTRIBUIÇÃO** associa três ocorrências de entidade: um produto a ser distribuído, uma cidade na qual é feita a distribuição e um distribuidor.

No caso de relacionamentos de grau maior que dois, o conceito de cardinalidade de relacionamento é uma extensão não trivial do conceito de cardinalidade em relacionamentos binários. Lembre-se que, em um relacionamento binário R entre duas entidades A e B, a cardinalidade máxima de A em R indica quantas ocorrências de B podem estar associadas a cada ocorrência de A. No caso de um relacionamento ternário, a cardinalidade refere-se a *pares de entidades*. Em um relacionamento R entre três entidades A, B e C, a cardinalidade máxima de A e B dentro de R indica quantas ocorrências de C podem estar associadas a um par de ocorrências de A e B.

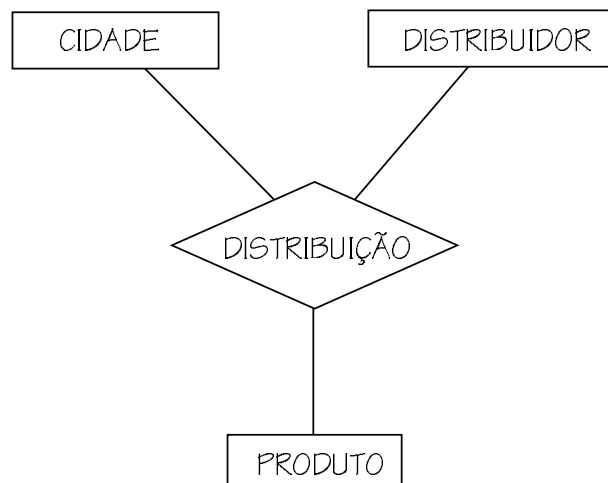


Figura 2.10: Relacionamento ternário

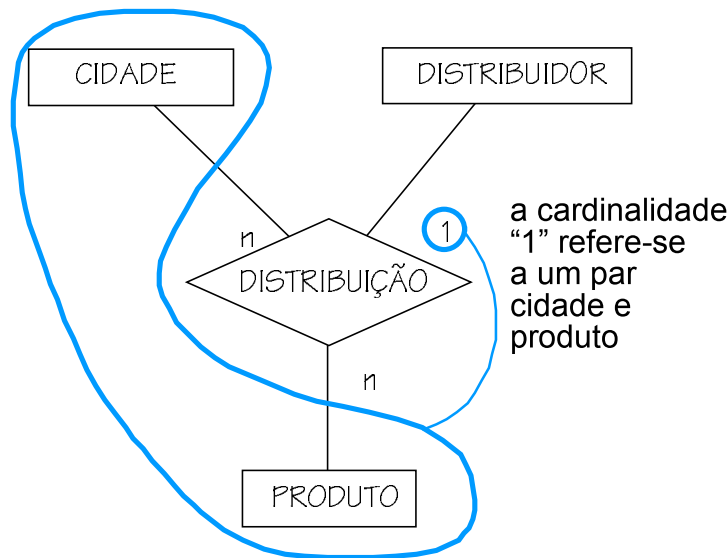


Figura 2.11: Cardinalidade em relacionamentos ternários

Exemplificando, na Figura 2.11, o “1” na linha que liga o retângulo representativo da entidade **DISTRIBUIDOR** ao losango representativo do relacionamento expressa que cada par de ocorrências (cidade, produto) está associado a no máximo um distribuidor. Em outros termos, não há concorrência pela distribuição de um produto em uma cidade.

Já os dois “n” expressam que:

- ❑ A um par (cidade, distribuidor) podem estar associados muitos produtos, ou em outros termos, um distribuidor pode distribuir em uma cidade muitos produtos.
- ❑ A um par (produto, distribuidor) podem estar associadas muitas cidades, ou em outros termos um distribuidor pode distribuir um produto em muitas cidades.

## 2.2.6 Cardinalidade mínima

Além da cardinalidade máxima, uma outra informação que pode ser representada por um modelo ER é o número mínimo de ocorrências de entidade que são associadas a uma ocorrência de uma entidade através de um relacionamento. Para fins de projeto de BD, consideram-se apenas duas cardinalidades mínimas: a cardinalidade mínima 0 e a cardinalidade mínima 1.

A cardinalidade mínima 1 também recebe a denominação de “associação *obrigatória*”, já que ela indica que o relacionamento deve obrigatoriamente associar uma ocorrência de entidade a cada ocorrência da entidade em questão. Com base na mesma linha de raciocínio, a cardinalidade mínima 0 também recebe a denominação de “associação *opcional*”.

A cardinalidade mínima é anotada no diagrama junto a cardinalidade máxima, conforme mostrado na Figura 2.12. Nesta figura, aparece novamente o exemplo da alocação de empregados a mesas que já foi apresentado anteriormente. Aqui, a cardinalidade mínima é usada para especificar que cada empregado deve ter a ele alocada obrigatoriamente uma mesa

(cardinalidade mínima 1) e que uma mesa pode existir sem que a ela esteja alocado um empregado (cardinalidade mínima 0).

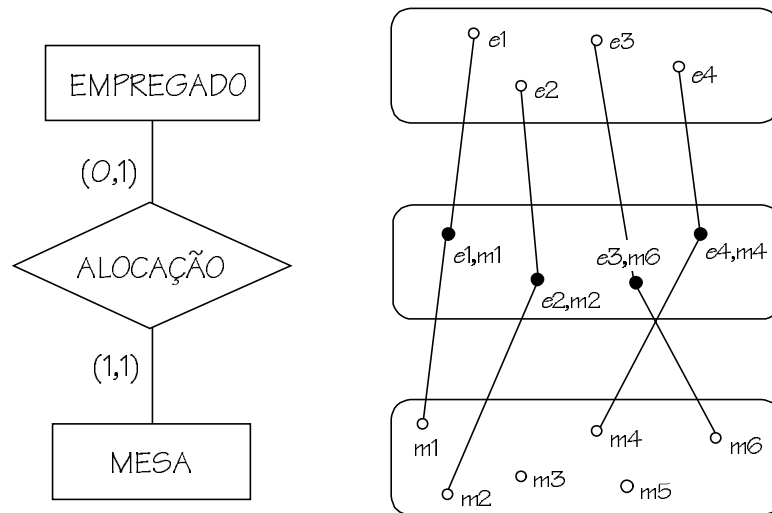


Figura 2.12: Cardinalidade mínima de relacionamento

### 2.2.7 Exemplo de uso de entidades e relacionamentos

A Figura 2.13 apresenta um exemplo de um diagrama entidade-relacionamento mais abrangente que os anteriores, envolvendo diversas entidades e relacionamentos. Como se vê, um diagrama ER é apresentado na forma de um grafo. A distribuição dos símbolos de DER no papel é totalmente arbitrária e não tem maior significado do ponto de vista formal. Entretanto, para tornar o diagrama mais legível é usual evitar-se cruzamentos de linhas. Para isso, a recomendação geral é a de posicionar os retângulos representativos de entidades que participam de muitos relacionamentos no centro do diagrama.

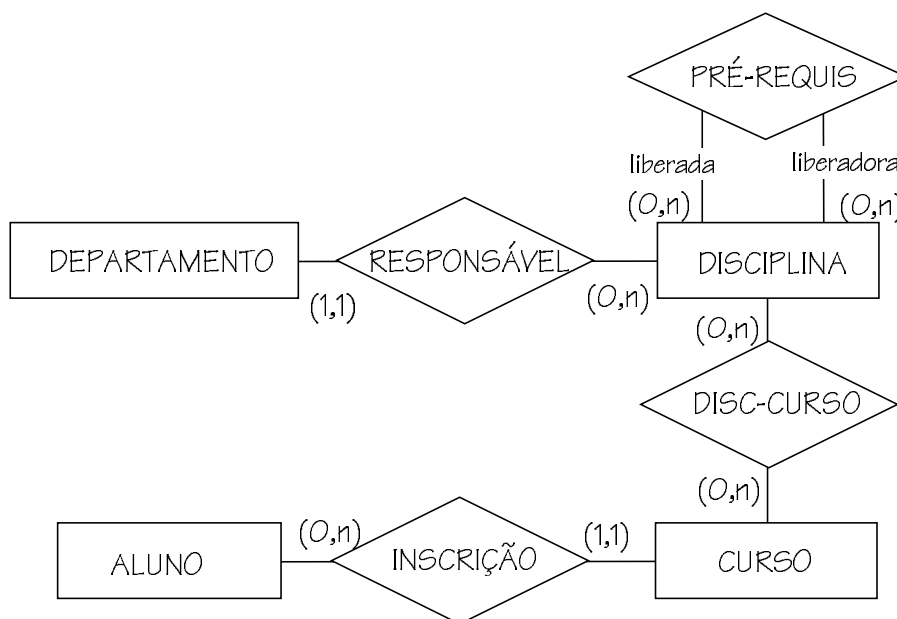


Figura 2.13: DER para o controle acadêmico de uma universidade

O modelo da Figura 2.13 é uma parte do modelo de dados de um sistema de controle acadêmico de uma universidade fictícia. O modelo descreve o seguinte:

- ❑ Deseja-se manter informações sobre alunos, cursos, disciplinas e departamentos.
- ❑ Além disso, deseja-se manter informações sobre a associação de alunos a cursos, de disciplinas a cursos, de disciplinas a departamentos, bem como de disciplinas a suas disciplinas pré-requisitos.
- ❑ Através das cardinalidades expressa-se que:
  - Cada disciplina possui exatamente um departamento responsável, e um departamento é responsável por muitas disciplinas, inclusive por nenhuma. Note-se que, apesar de sabermos que os departamentos em uma universidade existem para ser responsáveis por disciplinas, especificamos a cardinalidade mínima de DEPARTAMENTO em RESPONSÁVEL como sendo “0”. Com isso admitimos a possibilidade de existirem departamentos vazios. Esta cardinalidade foi especificada considerando o estado do banco de dados imediatamente após a criação de um novo departamento, bem como o estado imediatamente antes da eliminação de um departamento. Da forma como a restrição foi especificada, é possível incluir o departamento em uma transação, para, depois, em transações subsequentes, vinculá-lo às disciplinas sob sua responsabilidade. Se tivesse sido especificada a cardinalidade mínima “1”, ao menos uma disciplina teria que ser vinculada ao departamento já na própria transação de inclusão do departamento. Como observa-se da discussão acima, para especificar as cardinalidades mínimas é necessário possuir conhecimento sobre as transações de inclusão e exclusão das entidades.
  - Uma disciplina pode possuir diversos pré-requisitos, inclusive nenhum. Uma disciplina pode ser pré-requisito de muitas outras disciplinas, inclusive de nenhuma.
  - Uma disciplina pode aparecer no currículo de muitos cursos (inclusive de nenhum) e um curso pode possuir muitas disciplinas em seu currículo (inclusive nenhuma).
  - Um aluno está inscrito em exatamente um curso e um curso pode ter nele inscritos muitos alunos (inclusive nenhum).

## 2.3 ATRIBUTO

Para associar informações a ocorrências de entidades ou de relacionamentos usa-se o conceito de *atributo*.

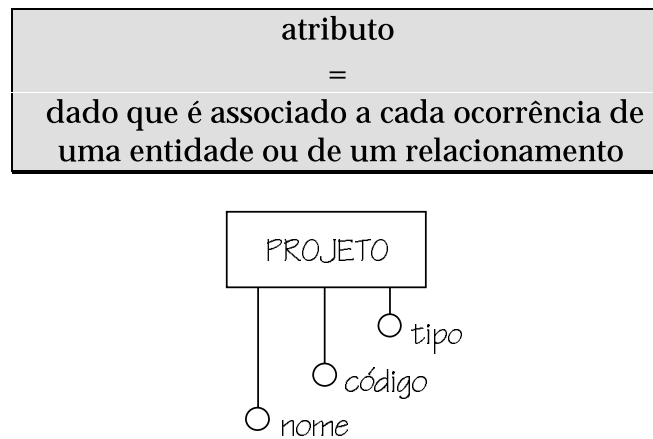


Figura 2.14: Atributos de uma entidade

Atributos são representados graficamente conforme mostra a Figura 2.14. A figura expressa que a cada ocorrência de **PROJETO** é associado exatamente um nome, um código e um tipo.

Na prática, atributos não são representados graficamente, para não sobrecarregar os diagramas, já que muitas vezes entidades possuem um grande número de atributos. Prefere-se usar uma representação textual que aparece separadamente do diagrama ER. Ao final deste capítulo, é fornecida uma possível sintaxe para uma representação textual dos atributos. No caso de ser usado um software para construção de modelos ER, o próprio software encarrega-se do armazenamento da lista de atributos de cada entidade em um *dicionário de dados*

Um atributo pode possuir uma cardinalidade, de maneira análoga a uma entidade em um relacionamento. A cardinalidade de um atributo define quantos valores deste atributo podem estar associados a uma ocorrência da entidade/relacionamento a qual ele pertence. A representação diagramática da cardinalidade de atributos é derivada da representação da cardinalidade de entidades em relacionamentos, conforme mostra a Figura 2.15. No caso de a cardinalidade ser (1,1) ela pode ser omitida do diagrama. Assim, o exemplo da Figura 2.15 expressa que **nome** e **código** são atributos *obrigatórios* (cardinalidade mínima “1” – cada entidade possui no mínimo um valor associado) e *mono-valorados* (cardinalidade máxima “1” – cada entidade possui no máximo um valor associado). Já o atributo **telefone**, é um atributo *opcional* (cardinalidade mínima 0) e *multi-valorado* (cardinalidade máxima n).

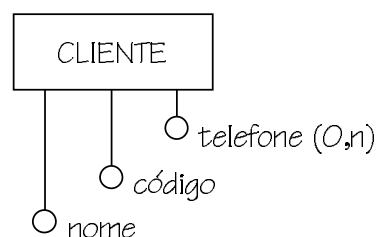


Figura 2.15: Cardinalidade de Atributos

Assim como entidades possuem atributos, também relacionamentos podem possuir atributos. A Figura 2.16 mostra um DER no qual um relacionamento, **ATUAÇÃO**, possui um atributo, a função que um engenheiro exerce dentro de um projeto. Esta não pode ser considerada atributo de **ENGENHEIRO**, já que um engenheiro pode atuar em diversos projetos exercendo diferentes funções. Também, não é atributo de **PROJETO**, já que, em um projeto, podem atuar diversos engenheiros com funções diferentes.

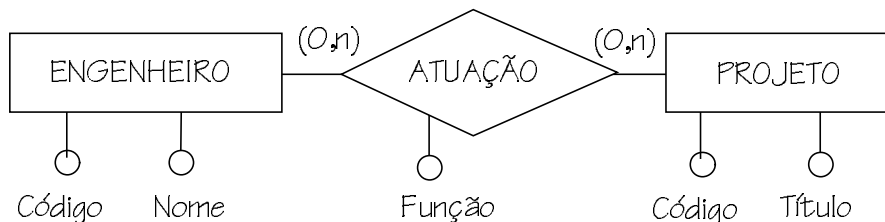


Figura 2.16: Atributo de relacionamento n:n

Outro exemplo de atributo em relacionamento, agora em um relacionamento 1:n, é mostrado na Figura 2.17. Este diagrama modela vendas em uma organização comercial. Algumas vendas são à vista, outras à prazo. Vendas à prazo são relacionadas a uma financeira, através do relacionamento **FINANCIAMENTO**. Os atributos **nº de parcelas** e **taxa de juros** são atributos do relacionamento.

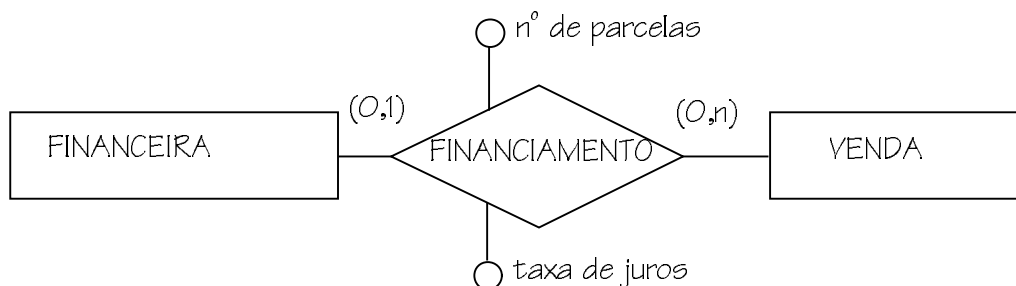


Figura 2.17: Atributo de relacionamento 1:n

Estes dois atributos poderiam ter sido incluídos na entidade **VENDA**. Neste caso, seriam atributos opcionais, já que nem toda venda é à prazo e possui estes atributos. Assim, preferiu-se usar o modelo da figura, exatamente para explicitar o fato de os atributos **nº de parcelas** e **taxa de juros** pertencerem somente a vendas à prazo.

### 2.3.1 Identificando entidades

Cada entidade deve possuir um *identificador*. Um identificador é um conjunto de um ou mais atributos (e possivelmente relacionamentos, como visto abaixo) cujos valores servem para distinguir uma ocorrência da entidade das demais ocorrências da mesma entidade.

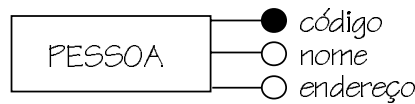


Figura 2.18: Identificador simples

O caso mais simples é o da entidade que possui um único atributo como identificador. No DER, atributos identificadores são representados por um círculo preto. No exemplo da Figura 2.18, o atributo **código** é identificador. Isso significa que cada pessoa possui um código diferente. Já os atributos **nome** e **endereço** não são identificadores – o mesmo nome (ou o mesmo endereço) pode ser associados a diferentes pessoas.

A Figura 2.19 mostra um exemplo no qual o identificador da entidade é composto por diversos atributos. Considera-se um almoxarifado de uma empresa de ferragens organizado como segue. Os produtos ficam armazenados em prateleiras. Estas prateleiras encontram-se em armários organizados em corredores. Os corredores são numerados sequencialmente a partir de um e as prateleiras são numeradas sequencialmente a partir de um dentro de um corredor. Assim, para identificar uma prateleira é necessário conhecer seu número e o número do corredor em que se encontra. Para cada prateleira deseja-se saber sua capacidade em metros cúbicos.

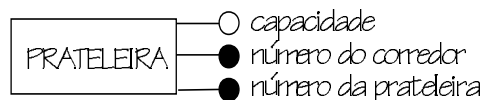


Figura 2.19: Identificador composto

Finalmente, há casos em que o identificador de uma entidade é composto não somente por atributos da própria entidade mas também por relacionamentos dos quais a entidade participa (*relacionamento identificador*). Um exemplo deste caso é mostrado na Figura 2.20. Este diagrama apresenta empregados de uma organização, relacionados com os seus dependentes para fins de imposto de renda. Cada dependente está relacionado a exatamente um empregado. Um dependente é identificado pelo empregado ao qual ele está relacionado e por um número de seqüência que distingue os diferentes dependentes de um mesmo empregado. No DER, o relacionamento usado como identificador é indicado por uma linha mais densa, conforme mostra a Figura 2.20.

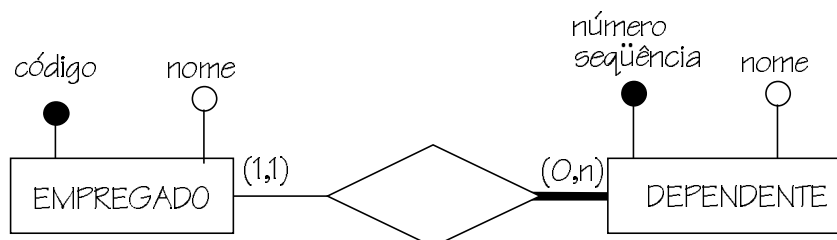


Figura 2.20: Relacionamento identificador

Nesse caso, alguns autores dizem que a entidade **DEPENDENTE** é uma entidade *fraca*. O termo “fraca” deriva-se do fato de a entidade somente existir quando relacionada a outra entidade e de usar como parte de seu identificador, entidades relacionadas. Entretanto, os autores de livros mais recentes preferem não utilizar o conceito, já que as entidades chamadas “fracas” por este critério podem, dependendo da realidade modelada, ser centrais a um modelo. Considere o fragmento de um DER sobre empresas, mostrado na Figura 2.21. No exemplo, é representada a divisão de grupos de empresas em empresas e de empresas em filiais de empresas. Para identificar um grupo de empresas é usado um código. Já uma empresa é identificada pelo grupo ao qual está relacionada e por um número da empresa dentro do grupo. Finalmente, uma filial é identificada pela empresa a qual está vinculada e por um número de filial dentro da empresa. Pela definição acima, tanto **EMPRESA** quanto **FILIAL** são entidades fracas. Entretanto, ao considerarmos que a maior parte das entidades que eventualmente comporiam o restante do modelo estariam ligadas a **EMPRESA** ou **FILIAL** vemos que o conceito de “fraqueza” introduzido acima não se aplica ao caso em questão.

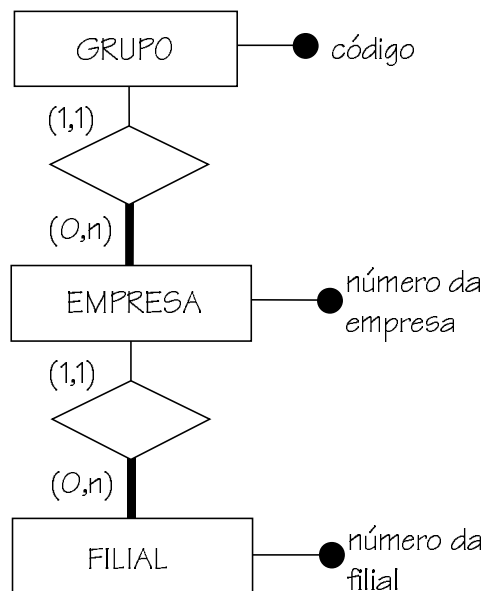


Figura 2.21: Entidades com relacionamentos identificadores

identificador de entidade
=
conjunto de atributos e relacionamentos cujos valores distinguem uma ocorrência da entidade das demais

O identificador de uma entidade, seja ele simples, composto por diversos atributos, ou composto por identificadores externos, deve obedecer duas propriedades:

- ❑ O identificador deve ser *mínimo*. Isso significa que o identificador de uma entidade deve ser composto de tal forma que, retirando um dos



atributos ou relacionamentos que o compõe, ele deixa de ser identificador. Exemplificando, na entidade **PESSOA** na Figura 2.18, o par **código** e **nome** poderia ser usado para distinguir uma ocorrência de **PESSOA** das demais. Entretanto, estes atributos não formam um identificador mínimo, já que **código** é suficiente para distinguir as ocorrências de **PESSOA**.

- ❑ Cada entidade deve possuir um *único* identificador. Em alguns casos, diferentes conjuntos de atributos podem servir para distinguir as ocorrências da entidade. Exemplificando, a entidade **EMPREGADO** da Figura 2.22 poderia possuir como identificador tanto o atributo **código**, quanto o atributo **CIC** (identificador único do contribuinte junto a Receita Federal). Cabe ao modelador decidir qual dos dois atributos será usado como identificador da entidade. Alguns critérios para esta decisão aparecem mais adiante, no capítulo relativo ao projeto de BD a partir do modelo ER.

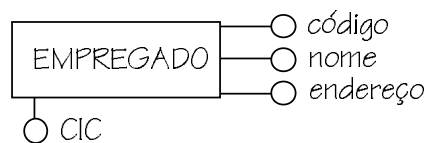


Figura 2.22: Identificadores alternativos

### 2.3.2 Identificando relacionamentos

Em princípio, uma ocorrência de relacionamento diferencia-se das demais do mesmo relacionamento pelas ocorrências de entidades que dela participam. Exemplificando, uma ocorrência de **ALOCUÇÃO** (Figura 2.23) é identificada pela ocorrência de **ENGENHEIRO** e pela ocorrência de **PROJETO** que ela relaciona. Em outros termos, para cada par (engenheiro, projeto) há no máximo um relacionamento de alocação.



Figura 2.23: Relacionamento identificado por suas entidades

Entretanto, há casos nos quais entre as mesmas ocorrências de entidade podem existir diversas ocorrências de relacionamento. Um exemplo é o relacionamento **CONSULTA** entre entidades de **MÉDICO** e de **PACIENTE** (Figura 2.24). Entre um determinado médico e um determinado paciente podem haver diversas consultas. Neste caso, é necessário algo que distinga uma consulta entre um médico e seu paciente das demais consultas entre este médico e seu paciente. A diferenciação dá-se através de *atributos identificadores de relacionamentos*. No caso da Figura 2.24, o atributo identificador do relacionamento é **data/hora**.

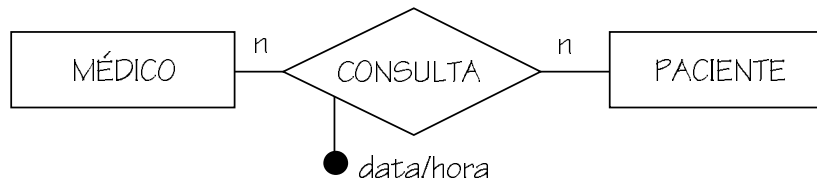


Figura 2.24: Identificador de relacionamento

Assim, de forma geral, um relacionamento é identificado pelas entidades dele participantes, bem como pelos atributos identificadores eventualmente existentes.

## 2.4 GENERALIZAÇÃO/ESPECIALIZAÇÃO

Além de relacionamentos e atributos, propriedades podem ser atribuídas a entidades através do conceito de *generalização/especialização*. Através deste conceito é possível atribuir propriedades particulares a um subconjunto das ocorrências (*especializadas*) de uma entidade *genérica*. O símbolo para representar generalização/especialização é um triângulo isósceles, conforme mostra a Figura 2.25. A generalização/especialização mostrada nesta figura expressa que a entidade **CLIENTE** é dividida em dois subconjuntos, as entidades **PESSOA FÍSICA** e **PESSOA JURÍDICA** cada um com propriedades próprias.

Associada ao conceito de generalização/especialização está a idéia de *herança de propriedades*. Herdar propriedades significa que cada ocorrência da entidade especializada possui, além de suas próprias propriedades (atributos, relacionamentos e generalizações/especializações), também as propriedades da ocorrência da entidade genérica correspondente. Assim, segundo o DER da Figura 2.25, a entidade **PESSOA FÍSICA** possui, além de seus atributos particulares, **CIC** e **sexo**, também todas as propriedades da ocorrência da entidade **CLIENTE** correspondente, ou seja, os atributos **nome** e **código**, o seu identificador (atributo **código**), bem como o relacionamento com a entidade **FILIAL**. Resumindo, o diagrama expressa que toda pessoa física tem como atributos **nome**, **código**, **CIC** e **sexo**, é identificada pelo **código** e está obrigatoriamente relacionada a exatamente uma filial. Da mesma maneira, toda pessoa jurídica tem como atributos **nome**, **código**, **CGC** e **tipo de organização**, é identificada pelo **código** e está obrigatoriamente relacionada a exatamente uma filial.

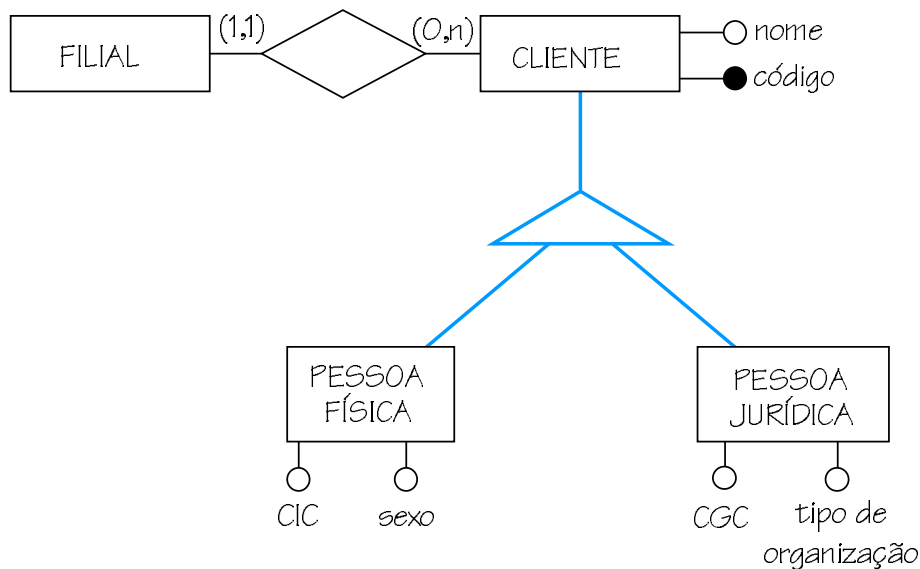


Figura 2.25: Generalização/especialização

A generalização/especialização pode ser classificada em dois tipos, *total* ou *parcial*, de acordo com a obrigatoriedade ou não de a uma ocorrência da entidade genérica corresponder uma ocorrência da entidade especializada.

Em uma generalização/especialização *total* para cada ocorrência da entidade genérica existe sempre uma ocorrência em uma das entidades especializadas. Esse é o caso do exemplo da Figura 2.25, no qual a toda ocorrência da entidade **CLIENTE** corresponde uma ocorrência em uma das duas especializações. Esse tipo de generalização/especialização é simbolizado por um “t” conforme mostrado na Figura 2.26.

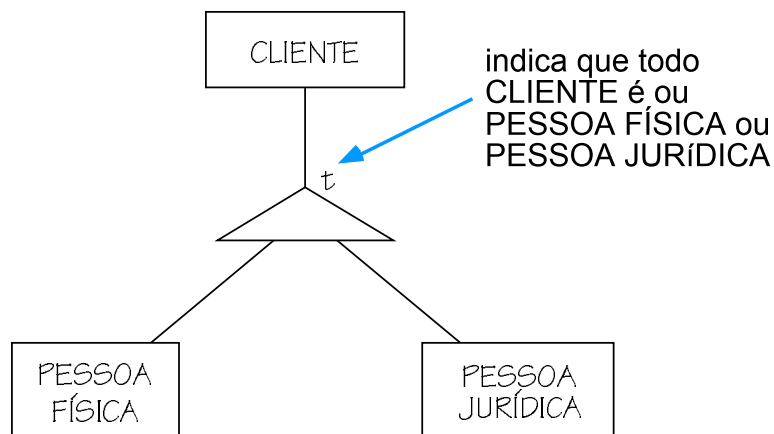


Figura 2.26: Generalização/especialização total

Em uma generalização/especialização *parcial*, nem toda ocorrência da entidade genérica possui uma ocorrência correspondente em uma entidade especializada. Esse é o caso do exemplo da Figura 2.27, no qual nem toda entidade **FUNCIONÁRIO** possui uma entidade correspondente em uma das duas especializações (nem todo o funcionário é motorista ou secretária). Esse tipo de generalização/especialização é simbolizado por um “p” conforme mostrado na figura. Usualmente, quando há uma especialização parcial, na

entidade genérica (no caso do exemplo, em **FUNCIONÁRIO**) aparece um atributo que identifica o tipo de ocorrência da entidade genérica (no caso do exemplo, trata-se do atributo **tipo de funcionário**). Este atributo não é necessário no caso de especializações totais, já que a presença da ocorrência correspondente a entidade genérica em uma de suas especializações é suficiente para identificar o tipo da entidade.

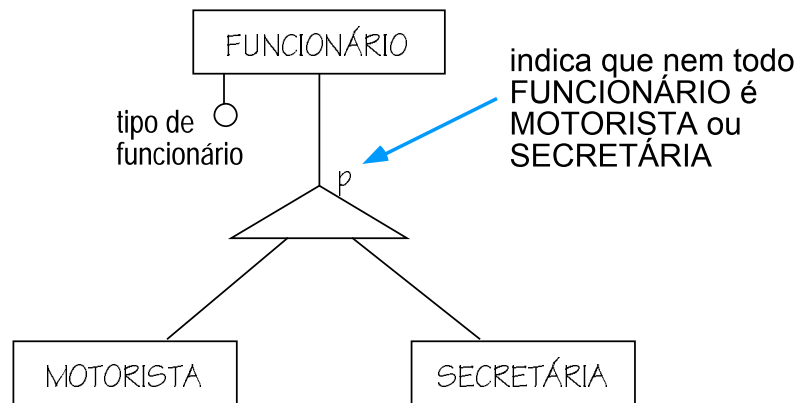


Figura 2.27: Generalização/especialização parcial

Uma entidade pode ser especializada em qualquer número de entidades, inclusive em uma única. Exemplificando, se no exemplo da Figura 2.27, apenas os motoristas possuísem propriedades particulares, haveria apenas uma entidade especializada, a de motoristas.

Além disso, não há limite no número de níveis hierárquicos da generalização/especialização. Uma entidade especializada em uma generalização/especialização, pode, por sua vez, ser entidade genérica em uma outra generalização/especialização. É admissível, inclusive, que uma mesma entidade seja especialização de diversas entidades genéricas (a chamada *herança múltipla*). A Figura 2.28 apresenta um DER em que aparecem múltiplos níveis de generalização/especialização, bem como o conceito de herança múltipla. Exemplificando, uma entidade **BARCO** é uma especialização de um **VEÍCULO AQUÁTICO**, que por sua vez é uma especialização de **VEÍCULO**. Assim um barco tem, além de suas propriedades específicas, também as propriedades de um veículo aquático, bem como as propriedades de um veículo em geral. O exemplo de herança múltipla aparece na entidade **VEÍCULO ANFÍBIO**. Um veículo anfíbio, possui, além de suas propriedades específicas, tanto as propriedades de um veículo aquático, quanto as propriedades de um veículo terrestre, já que é especialização destas duas últimas entidades.

A generalização/especialização como tratada neste livro é *exclusiva*. Generalização/especialização exclusiva significa que uma ocorrência de entidade genérica aparece, para cada hierarquia generalização/especialização, no máximo uma vez, nas folhas da árvore de generalização/especialização. Exemplificando, na Figura 2.28, um veículo é automóvel ou veículo anfíbio ou barco e somente um destes.

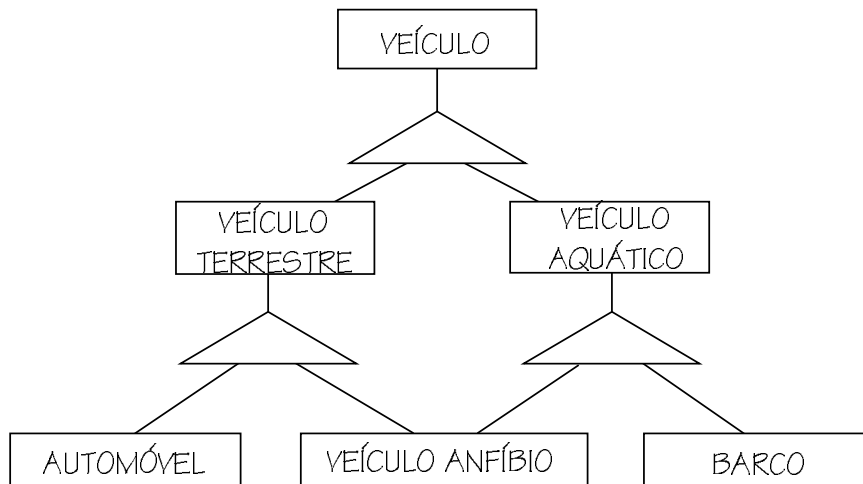


Figura 2.28: Generalização/especialização em múltiplos níveis e com herança múltipla

Há autores que permitem também especializações não exclusivas. Um exemplo é mostrado na Figura 2.29. Neste diagrama, considera-se o conjunto de pessoas vinculadas a uma universidade. Neste caso, a especialização não é exclusiva, já que a mesma pessoa pode aparecer em múltiplas especializações. Uma pessoa pode ser professor de um curso e ser aluno em outro curso, ou ser aluno de pós-graduação. Por outro lado, uma pessoa pode acumular o cargo de professor em tempo parcial com o cargo de funcionário, ou, até mesmo, ser professor de tempo parcial em dois departamentos diferentes, sendo portanto duas vezes professor. O principal problema que este tipo de generalização/especialização apresenta é que neste caso as entidades especializadas não podem herdar o identificador da entidade genérica. No caso, o identificador de pessoa não seria suficiente para identificar professores, já que uma pessoa pode ser múltiplas vezes professor.

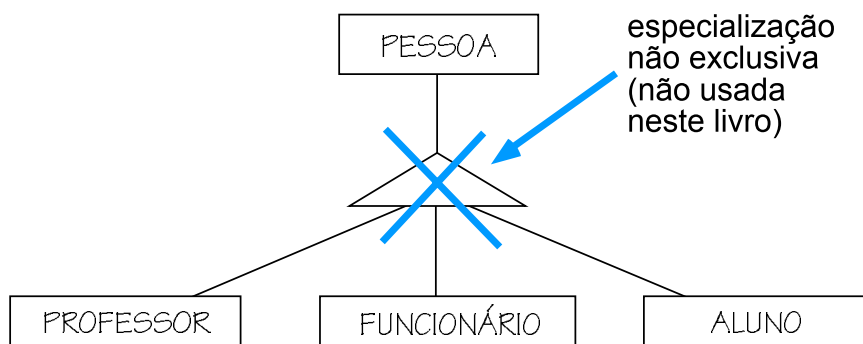


Figura 2.29: Generalização/especialização não exclusiva

Em casos como o mostrado, usa-se o conceito de relacionamento, ao invés do conceito de generalização/especialização. O modelo deveria conter três relacionamentos, associando a entidade **PESSOA** com as entidades correspondentes a cada um dos papéis de **PESSOA** (**PROFESSOR**, **FUNCIONÁRIO** e **ALUNO**).

## 2.5 ENTIDADE ASSOCIATIVA

Um relacionamento é uma associação entre entidades. Na modelagem ER não foi prevista a possibilidade de associar uma entidade com um relacionamento ou então de associar dois relacionamentos entre si. Na prática, quando está-se construindo um novo DER ou modificando um DER existente, surgem situações em que é desejável permitir a associação de uma entidade a um relacionamento. A título de exemplo, considere-se o diagrama da Figura 2.30.



Figura 2.30: DER a ser modificado

Suponha que seja necessário modificar este diagrama com a adição da informação de que, em cada consulta, um ou mais medicamentos podem ser prescritos ao paciente. Para tal, criar-se-ia uma nova entidade, **MEDICAMENTO**. A questão agora é: com que entidade existente deve estar relacionada a nova entidade? Se **MEDICAMENTO** fosse relacionado a **MÉDICO**, ter-se-ia apenas a informação de que médico prescreveu que medicamentos, faltando a informação do paciente que os teve prescritos. Por outro lado, se **MEDICAMENTO** fosse relacionado a **PACIENTE**, faltaria a informação do médico que prescreveu o medicamento. Assim, deseja-se relacionar o medicamento à consulta, ou seja deseja-se relacionar uma entidade (**MEDICAMENTO**) a um relacionamento (**CONSULTA**), o que não está previsto na abordagem ER. Para tal, foi criado um conceito especial, o de *entidade associativa*. Uma entidade associativa nada mais é que a redefinição de um relacionamento, que passa a ser tratado como se fosse também uma entidade. Graficamente, isso é feito como mostrado na Figura 2.31. O retângulo desenhado ao redor do relacionamento **CONSULTA** indica que este relacionamento passa a ser visto como uma entidade (associativa, já que baseada em um relacionamento). Sendo **CONSULTA** também uma entidade, é possível associá-la através de relacionamentos a outras entidades, conforme mostra a figura.

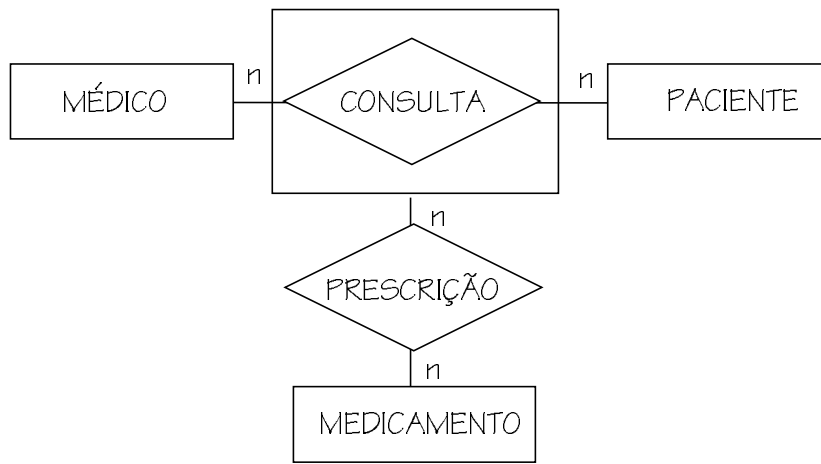


Figura 2.31: Entidade associativa

Observe-se que, caso não se desejasse usar o conceito de entidade associativa, seria necessário transformar o relacionamento **CONSULTA** em uma entidade, que então poderia ser relacionada a **MEDICAMENTO**, conforme mostrado na Figura 2.32.

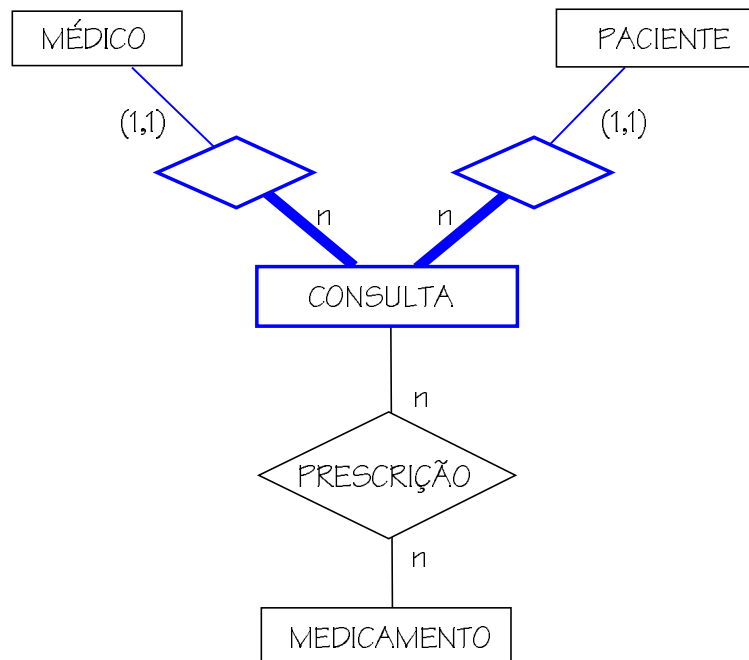


Figura 2.32: Substituindo relacionamento por entidade

Na figura, o relacionamento foi substituído por uma entidade homônima, junto com dois relacionamentos (parte representada em linhas densas). Observe-se que, para manter equivalência com o diagrama anterior, uma consulta está relacionada com exatamente um médico e exatamente um paciente (cardinalidade mínima e máxima é um). Uma consulta é identificada pelo paciente e pelo médico a ela ligados. Tendo substituído o relacionamento **CONSULTA** pela entidade, basta relacionar a entidade **CONSULTA** com a entidade **MEDICAMENTO**. Observe-se que o diagrama da Figura 2.32 é equivalente ao diagrama da Figura 2.31. Equivalente aqui significa que ambos geram

o mesmo banco de dados relacional. A transformação de modelos ER em descrições de BD relacionais, necessária à compreensão do conceito de equivalência, é apresentada em um dos próximos capítulos.

## 2.6 ESQUEMAS GRÁFICOS E TEXTUAIS DE MODELOS ER

A descrição de um modelo é chamada, na terminologia de banco de dados, de *o esquema do banco de dados*.

Até este ponto os esquemas ER sempre foram diagramas ER, isto é sempre estão apresentados na forma gráfica. A Figura 2.33 resume os símbolos usados neste livro para a representação gráfica de esquemas ER.

Um esquema ER pode ser um texto. Na Figura 2.34 aparece a sintaxe de uma linguagem textual para definição de esquemas ER. A sintaxe é dada na forma de uma gramática BNF<sup>3</sup>. Nesta sintaxe, são usadas as seguintes convenções: colchetes denotam opcionalidade, chaves denotam repetição, o sufixo **LISTA** denota uma seqüência de elementos separados por vírgulas e o sufixo **NOME** denota identificadores.

---

<sup>3</sup> Se você não sabe o que é uma gramática BNF consulte um livro sobre linguagens de programação ou sobre construção de compiladores.



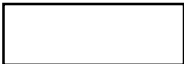
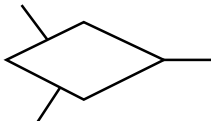


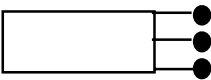
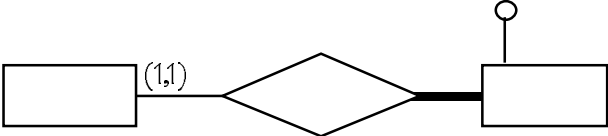
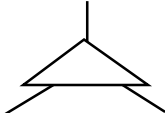
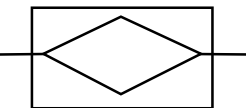
Conceito	Símbolo
Entidade	
Relacionamento	
Atributo	
Atributo identificador	 
Relacionamento identificador	
Generalização/especialização	
Entidade associativa	

Figura 2.33: Símbolos usados na construção de esquemas ER

A Figura 2.35 apresenta um esquema ER textual correspondente ao esquema ER gráfico da Figura 2.20. Note-se que a representação gráfica e a textual aqui usadas não são exatamente equivalentes. A notação textual, em nosso caso, é mais rica, pois inclui a possibilidade de definir um *tipo de atributo* (declaração [DECL\\_TIPO](#)).

Na prática é usual combinar as duas formas de representar esquemas ER: a diagramática e a textual. Escolhe-se a forma de representar de acordo com a sua adequação. Entidades e relacionamentos, bem como hierarquias de generalização/especialização são normalmente representadas de forma gráfica, pois a representação textual de grafos é difícil de ler. Já os atributos das entidades e dos relacionamentos, bem como a definição de atributos identificadores é feita de forma textual, pois sobrecarregaria o diagrama.

ESQUEMA → Esquema: ESQUEMA\_NOME  
           SEÇÃO\_ENTIDADE  
           SEÇÃO\_GENERALIZAÇÃO  
           SEÇÃO\_ENT\_ASSOCIATIVA  
           SEÇÃO\_RELACIONAMENTO

SEÇÃO\_ENTIDADE → {DECL\_ENT}

DECL\_ENT → Entidade: ENTIDADE\_NOME  
           [SEÇÃO\_ATRIBUTO]  
           [SEÇÃO\_IDENTIFICADOR]

SEÇÃO\_ATRIBUTO → Atributos: {DECL\_ATRIB}

DECL\_ATRIB → [(MIN\_CARD,MAX\_CARD)] ATRIBUTO\_NOME [: DECL\_TIPO]

MIN\_CARD → 0 | 1

MAX\_CARD → 1 | n

DECL\_TIPO → inteiro | real | boolean | texto(INTEIRO) |  
           enumeração(LISTA\_VALORES)

SEÇÃO\_IDENTIFICADOR → Identificadores: {DECL\_IDENT}

DECL\_IDENT → {IDENTIFICADOR}

IDENTIFICADOR → ATRIBUTO\_NOME |  
           ENTIDADE\_NOME (via RELACIONAMENTO\_NOME)

SEÇÃO\_GENERALIZAÇÃO → {DECL\_HIERARQUIA\_GEN}

DECL\_HIERARQUIA\_GEN → Generalização [(COBERTURA)]: NOME\_GEN  
           PAI: NOME\_ENTIDADE  
           FILHO: LISTA\_NOME\_ENTIDADE

COBERTURA → t | p

SEÇÃO\_ENT\_ASSOCIATIVA → {DECL\_ENT\_ASSOC}

DECL\_ENT\_ASSOC → EntidadeAssociativa: NOME\_RELACIONAMENTO

SEÇÃO\_RELACIONAMENTO → {DECL\_RELACION}

DECL\_RELACION → Relacionamento: NOME\_RELACIONAMENTO  
           Entidades: {DECL\_ENT-RELACIONADA}  
           [Atributos: {DECL\_ATRIB}]  
           [Identificadores: {DECL\_IDENT}]

DECL\_ENT-RELACIONADA → [(CARD\_MIN,CARD\_MAX)] NOME\_ENTIDADE

Figura 2.34: Gramática BNF de uma linguagem para definição de esquema ER textual

```

Esquema: EMP_DEP

Entidade: EMPREGADO
  Atributos: CÓDIGO: inteiro
  Identificadores: CÓDIGO

Entidade: DEPENDENTE
  Atributos: NÚMERO_SEQUENCIA: inteiro
             NOME: texto(50)
  Identificadores: EMPREGADO via EMP_DEP
                  NÚMERO_SEQUENCIA

Relacionamento: EMP_DEP
  Entidades: (1,1) EMPREGADO
             (0,n) DEPENDENTE

```

Figura 2.35: Esquema ER textual correspondente à Figura 2.20

## EXERCÍCIOS

**Exercício 2.1:** Dê ao menos cinco exemplos dos conceitos básicos da abordagem ER apresentados neste capítulo: entidade, relacionamento, atributo, generalização/especialização.

**Exercício 2.2:** Explique a diferença entre uma entidade e uma ocorrência de entidade. Exemplifique.

**Exercício 2.3:** O que é o papel de uma entidade em um relacionamento. Quando é necessário especificar o papel das entidades de um relacionamento?

**Exercício 2.4:** Considere o relacionamento **CASAMENTO** que aparece no DER da Figura 2.7. Segundo este DER o banco de dados poderia conter um casamento em que uma pessoa está casada consigo mesma? O DER permite que a mesma pessoa apareça em dois casamentos diferentes, uma vez como marido e outra vez com esposa? Caso uma destas situações possa ocorrer, como deveria ser modificado o DER para impedi-las?

**Exercício 2.5:** Confeccione um possível diagrama de ocorrências para o relacionamento **SUPERVISÃO** (Figura 2.8) e suas respectivas entidades.

**Exercício 2.6:** Confeccione um possível diagrama de ocorrências para o relacionamento **COMPOSIÇÃO** (Figura 2.9) e suas respectivas entidades.

**Exercício 2.7:** Mostre como o modelo ER da Figura 2.11 pode ser representado sem uso de relacionamentos ternários, apenas usando relacionamentos binários.

**Exercício 2.8:** Dê um exemplo de um relacionamento ternário. Mostre como a mesma realidade pode ser modelada somente com relacionamentos binários.

**Exercício 2.9:** Para o exemplo de relacionamento ternário da questão anterior, justifique a escolha das cardinalidades mínima e máxima.

**Exercício 2.10:** Considere o DER da Figura 2.12. Para que a restrição de cardinalidade mínima seja obedecida, que ocorrências de entidade devem existir no banco de dados, quando for incluída uma ocorrência de **EMPREGADO**? E quando for incluída uma ocorrência de **MESA**?

**Exercício 2.11:** Construa um DER que modela a mesma realidade que a mostrada no DER da Figura 2.16, usando apenas relacionamentos 1:n.

**Exercício 2.12:** Considere o relacionamento **EMPREGADO-DEPENDENTE** que aparece na Figura 2.20. Considere que um dependente de um empregado possa ser também empregado. Como o modelo deveria ser modificado para evitar o armazenamento redundante das informações das pessoas que são tanto dependentes quanto empregados?

**Exercício 2.13:** Construa um DER em que o conceito de entidade associativa é usado.

**Exercício 2.14:** Dê ao menos três exemplos de entidades com relacionamentos identificadores (entidades fracas).

**Exercício 2.15:** Considere o exemplo da Figura 2.13. Modifique as cardinalidades mínimas de forma a especificar o seguinte:

- ☐ Um curso não pode estar vazio, isto é, deve possuir ao menos uma disciplina em seu currículo
- ☐ Um aluno, mesmo que não inscrito em nenhum curso, deve permanecer por algum tempo no banco de dados.

**Exercício 2.16:** Sem usar atributos opcionais, nem atributos multi-valorados, construa um DER que contenha as mesmas informações do DER da Figura 2.15

**Exercício 2.17:** O DER da Figura 2.29 modela uma generalização/especialização não exclusiva. Como dito no texto do capítulo que descreve este DER, generalizações/especializações deste tipo não são usadas neste livro. Construa um DER que modela a realidade descrita sem usar o conceitos de generalização/especialização não exclusiva.

**Exercício 2.18:** A Figura 2.36 apresenta um modelo de dados para uma farmácia. Descreva em português tudo o que está representado neste diagrama.

**Exercício 2.19:** Invente nomes para os relacionamentos da Figura 2.36.

**Exercício 2.20:** Dê uma justificativa para as cardinalidades mínimas do relacionamento entre **FORNECEDOR** e **FABRICANTE** no DER da Figura 2.36.

**Exercício 2.21:** Explique o significado das cardinalidades mínima e máxima do relacionamento ternário (entre **MEDICAMENTO**, **VENDA** e **RECEITA MÉDICA**) no DER da Figura 2.36.

**Exercício 2.22:** Em princípio, uma venda deve envolver ao menos um produto. Entretanto, isso não é exigido pelas cardinalidades mínimas dos relacionamentos entre **VENDA** e **MEDICAMENTO** e entre **VENDA** e **PERFUMARIA** no DER da Figura 2.36. Explique porque.

**Exercício 2.23:** Para cada entidade e cada relacionamento no DER da Figura 2.36 defina, quando possível, atributos. Para cada entidade, indique o(s) atributo(s) identificador(es).

**Exercício 2.24:** Escreva um esquema ER textual para o esquema diagramático da Figura 2.36.

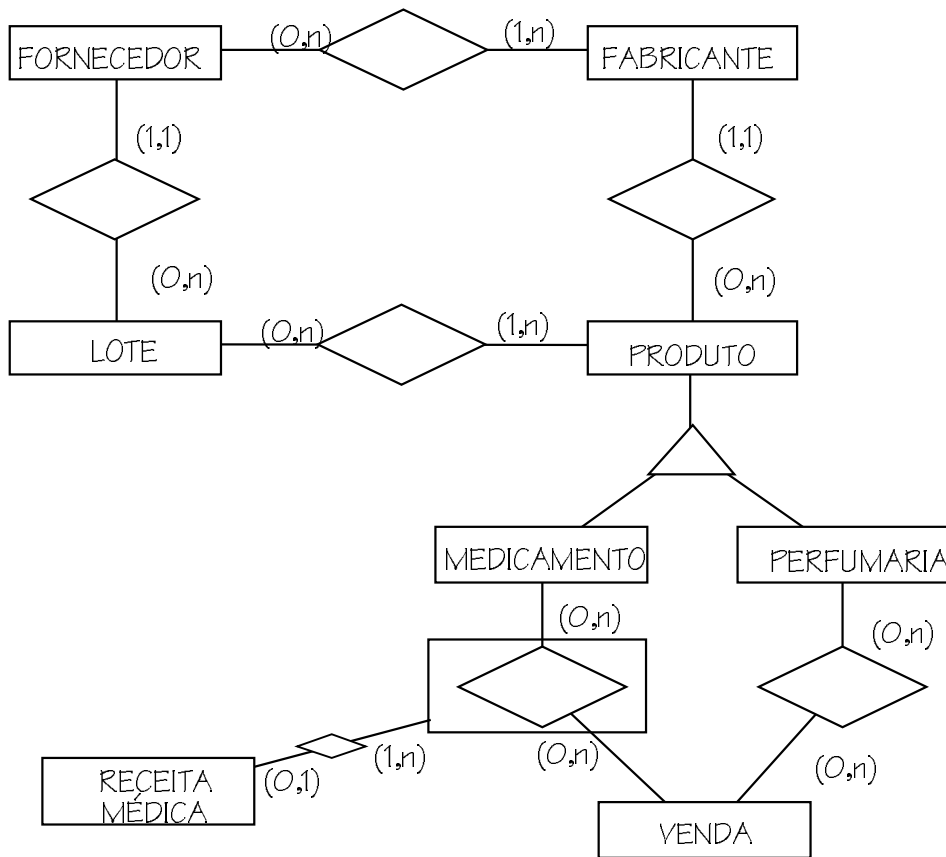


Figura 2.36: Diagrama ER de uma farmácia

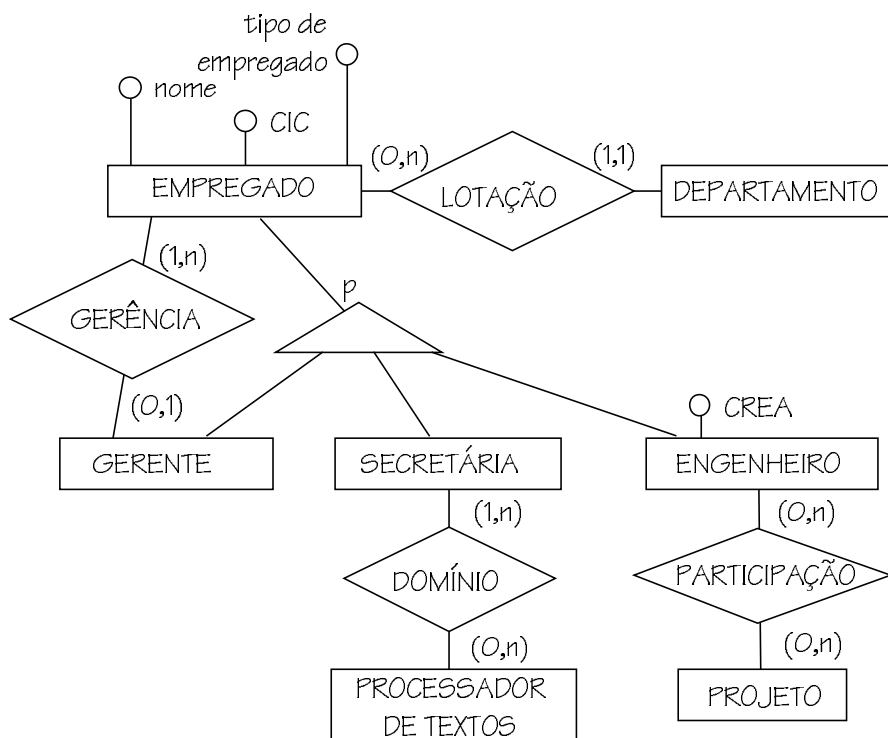


Figura 2.37: Diagrama ER para sistema de recursos humanos

**Exercício 2.25:** A Figura 2.37 apresenta um DER de parte de um sistema de recursos humanos em uma organização. Descreva em português tudo que está representado neste diagrama.

**Exercício 2.26:** Para cada entidade e cada relacionamento do DER da Figura 2.37 defina, quando possível, atributos. Para cada entidade, indique o(s) atributo(s) identificador(es).

**Exercício 2.27:** Escreva um esquema ER textual para o esquema diagramático da Figura 2.37

**Exercício 2.28:** De acordo com o DER da Figura 2.37, que ações devem ser tomadas ao excluir-se do banco de dados uma secretária?

**Exercício 2.29:** De acordo com o DER da Figura 2.37, uma secretária ou um engenheiro não podem ser gerentes. Porque? Como o DER deveria ser modificado para permitir que tanto uma secretária, quanto um engenheiro pudessem ser também gerentes?

## REFERÊNCIAS BIBLIOGRÁFICAS

O artigo original sobre a abordagem ER é o trabalho de Chen [3]. Na abordagem original de Chen apareciam apenas os conceitos de entidade, relacionamento e atributo. Ainda não apareciam os conceitos de generalização/especialização nem de entidade associativa. Estes foram introduzidos mais tarde em diversos trabalhos como [7,8].

O trabalho de Chen baseou-se em diversas propostas de modelos de dados que foram feitas à época. Um dos artigos fundamentais sobre o assunto modelagem de dados é o de Abrial [1]. Este é historicamente o primeiro a tratar do problema de modelagem semântica de dados, isto é da modelagem sem considerar aspectos de implementação. Os trabalhos de Kent [4,5] abordam em detalhe a diferença entre um modelo lógico que inclui detalhes de implementação e um modelo semântico de dados, abstrato e independente de implementação, como é a abordagem ER. Já o trabalho de Smith e Smith [9] apresenta os conceitos sobre os quais a maioria das abordagens de modelagem de dados, inclusive a abordagem ER, baseiam-se.

Além da abordagem ER, diversas outras abordagens de modelagem foram propostas na literatura. Apesar de a abordagem ER continuar sendo a abordagem de modelagem de dados mais aceita, é interessante estudar as propostas concorrentes, para compreender os pontos fracos da abordagem ER e as propostas para corrigi-los. A técnica NIAM [11] é provavelmente o concorrente mais importante da abordagem ER, pois teve, principalmente durante a década de 80 e na Europa, um número considerável de usuários. A principal característica de NIAM que a diferencia da abordagem ER é a ausência do conceito de atributo.

A abordagem ER é apresentada em vários livros texto sobre projeto de banco de dados. Um texto bastante completo e detalhado é o de Batini, Ceri e Navathe [2]. Outros livros conhecidos são o de Teorey [10] e o de Maninila e Rähä [6], este último mais dedicado a aspectos teóricos.

- [1] Abrial, J.R. Data Semantics, in Klimbie, J.W.. and Koffman, K.L. (editors) *Data Base Management*, North-Holland, 1974, 1–59.

- [2] Batini, C.; Ceri, S.; Navathe, S.B. *Conceptual Database Design - an Entity Relationship Approach*. , Benjamin/Cummings, Redwood City, CA, 1992
- [3] Chen, P. The Entity Relationship Model—Toward a Unified View of Data, *ACM Transactions on Database Systems*, 1:1, March 1976, pp 9–37.
- [4] Kent, W. *Data and Reality*, North-Holland, 1978.
- [5] Kent, W. Limitations of Record-Based Information Models”, *ACM Transactions on Database Systems*, 4:1, March 1979, pp 107–131.
- [6] Maninila, H.; Rähkä, K.-J. *The Design of Relational Databases*. Addison-Wesley, Wokingham, England, 1992.
- [7] dos Santos, C.S., Neuhold, E., & Furtado, A. A Data Type Approach to the Entity Relationship Model, in Chen, P. (editor) *Entity-Relationship Approach to Systems Analysis and Design*, (Proceedings of the First International Conference on Entity-Relationship Approach, Los Angeles, California, December 1979), North-Holland, 1980, pp 103–120.
- [8] Scheuermann, P., Schiffner, G., & Weber, H. Abstraction Capabilities and Invariant Properties Modeling within the Entity-Relationship Approach, in Chen, P. (editor) *Entity-Relationship Approach to Systems Analysis and Design*, (Proceedings of the First International Conference on Entity-Relationship Approach, Los Angeles, California, December 1979), North-Holland, 1980, pp 121-140.
- [9] Smith, J. & Smith, D. Database Abstractions: Aggregation and Generalization, *ACM Transactions on Database Systems*, 2:2, June 1977, pp 105–133.
- [10] Teorey, T.J. *Database Modeling & Design - The Fundamental Principles*. Second Edition. Morgan Kaufmann, San Francisco, CA, 1994
- [11] Verheijen, G.M.A. and VanBekkum, J. NIAM: An Information Analysis Method, in Olle, T., Sol, H., and Verrijn-Stuart, A. (editors) *Information System Design Methodologies: a Comparative Review*, North-Holland, 1982, pp. 537–590.





## Construindo modelos ER

No capítulo anterior, vimos como um diagrama ER é composto. Neste capítulo, vamos nos concentrar na construção de modelos ER, isto é, no problema de como, dada uma determinada realidade, obter o modelo ER.

O capítulo inicia apresentando uma coletânea de conselhos práticos e heurísticas a usar durante a modelagem conceitual. A seguir são apresentadas notações alternativas a de Peter Chen para confecção de diagramas ER. O capítulo finaliza apresentando um processo de modelagem e discutindo alternativas a este processo.

### 3.1 PROPRIEDADES DE MODELOS ER

Nesta seção, discutimos algumas propriedades de modelos ER que são importantes ter em mente quando da modelagem ER.

#### 3.1.1 Um modelo ER é um modelo formal

Um DER é um modelo formal, preciso, não ambíguo. Isto significa que diferentes leitores de um mesmo DER devem sempre entender exatamente o mesmo. Tanto é assim, que um DER pode ser usado como entrada a uma ferramenta CASE<sup>4</sup> na geração de um banco de dados relacional. Por isso, é de fundamental importância que todos os envolvidos na confecção e uso de diagramas ER estejam treinados na sua perfeita compreensão.

Observa-se em certas organizações, que modelos ER são sub-utilizados, servindo apenas como ferramenta para apresentação informal de idéias. Isso pode ser evitado com treinamento formal de todos envolvidos na modelagem e projeto do banco de dados.

Note-se que é importante que efetivamente *todos* os que manipulam modelos ER estejam treinados em sua compreensão. O fato de um DER ser gráfico e intuitivo pode transmitir a falsa impressão de ser compreensível até por alguém não treinado.

Para exemplificar o tipo de problemas que surgem ao usar diagramas ER sem treinar as pessoas envolvidas, descrevo uma situação que já observei em algumas organizações. Os técnicos em computação da organização desenvolveram um DER a partir de sua compreensão sobre o sistema a ser construído, obtida a partir de entrevistas com os futuros usuários do sistema. Para validar o modelo, este foi apresentado aos usuários. Entretanto, os usuários não foram treinados em modelagem e compreendiam o modelo apenas como uma descrição gráfica informal. Os usuários concordaram com o DER que lhes foi apresentado. Mais tarde, já com o banco de dados em funcionamento, descobriu-se que os usuários não haviam entendido efetivamente o que foi modelado. O BD implementado não era exatamente aquele desejado pelos usuários, apesar de corresponder exatamente ao DER apresentado. Assim, para que modelos ER possam atingir seus objetivos, é necessário treinamento formal. Não estou sugerindo aqui que os usuários tenham que ser treinados como modeladores. Basta que eles recebam algumas horas de treinamento na leitura e compreensão de diagramas ER.

#### 3.1.2 Abordagem ER têm poder de expressão limitado

Em um modelo ER, são apresentadas apenas algumas propriedades de um banco de dados. Em realidade, a linguagem dos DER é uma linguagem muito pouco poderosa e muitas propriedades desejáveis do banco de dados necessitam ser anotadas adicionalmente ao DER. Abaixo mostramos dois exemplos

---

<sup>4</sup>Ferramenta CASE: software que dá suporte a construção de software (CASE vem do inglês “computer aided software engineering” que significa “engenharia de software suportada por computador”)

de DER, já vistos no capítulo anterior, para salientar o quão incompletos são estes modelos.

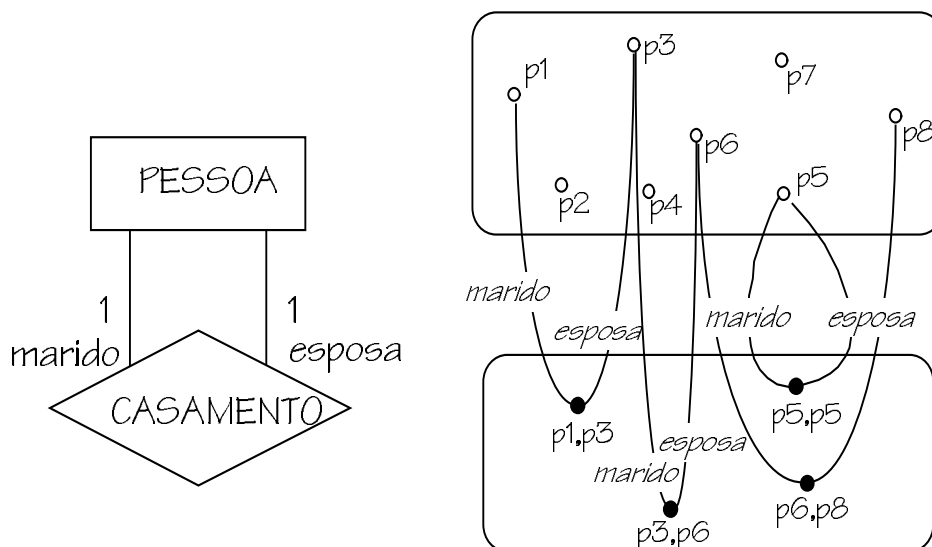


Figura 3.1: DER e diagrama de ocorrências para CASAMENTO

A Figura 3.1 mostra o DER que modela pessoas e casamentos já mostrado no capítulo anterior. Ao lado, são mostrados possíveis ocorrências de **PESSOA** e **CASAMENTO**. Entretanto, as ocorrências de **CASAMENTO** não correspondem ao nosso conhecimento da realidade (pelo menos se considerarmos a legislação brasileira). A pessoa **p3** aparece em dois casamentos, uma vez no papel de esposa e outra vez no papel de marido. Além disso, a pessoa **p5** aparece casada consigo mesma (o que poderia ser bom para um contribuinte do imposto de renda - ter-se-ia os descontos referentes a um dependente, sem incorrer nos custos referentes a um dependente). Assim, para fazer com que o DER corresponda a realidade que deseja-se modelar, é necessário modificá-lo ou então definir restrições adicionais. No caso em questão, é possível modificar o DER para excluir os casamentos indesejáveis (ver exercício no capítulo anterior).

Aqui cabe a pergunta: até onde deve ser modificado um DER para introduzir restrições de integridade? A resposta não é trivial. Aqui entra o bom senso do modelador. É necessário lembrar o objetivo que se tem ao construir um diagrama ER: o de projetar um banco de dados. Neste contexto, o DER nada mais é do que uma descrição abstrata das estruturas do banco de dados (das tabelas, no caso de um banco de dados relacional). O objetivo do diagrama não é o de especificar todas restrições de integridade. Assim, somente são incluídas construções em um DER, quando estas possuem uma correspondência no banco de dados a ser implementada. Construções artificiais, isto é, construções incluídas no modelo apenas para satisfazer determinadas restrições de integridade são indesejáveis, pois distorcem os objetivos que se tem ao construir o DER.

Entretanto, há situações em que, mesmo através de modificações no DER, é impossível incluir no diagrama as restrições desejadas.

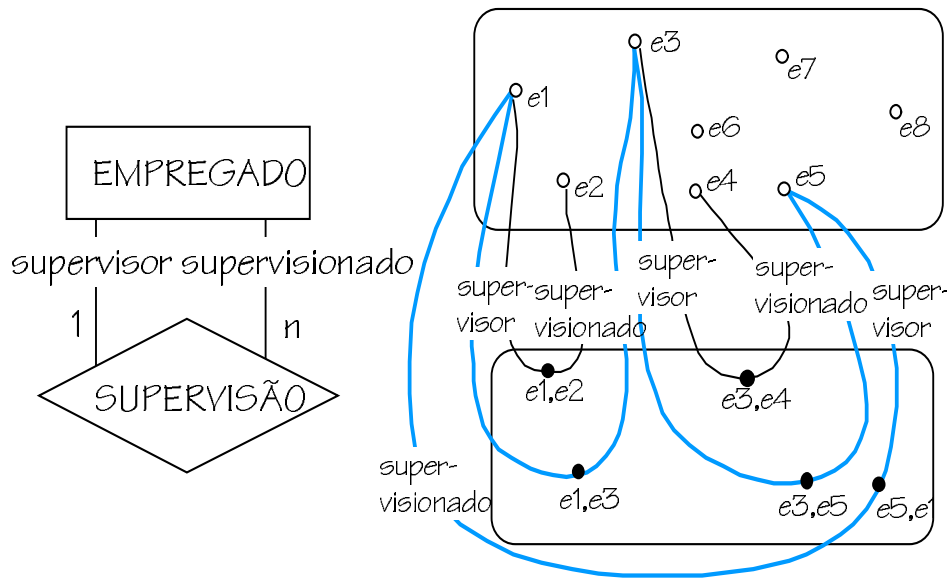


Figura 3.2: Relacionamento recursivo

A Figura 3.2 apresenta um exemplo de um DER modelando empregados e a hierarquia de supervisão em uma organização. O relacionamento **SUPERVISÃO** possui cardinalidade 1:n indicando que um empregado pode supervisionar muitos outros, mas possui no máximo um supervisor. Como está especificado, o modelo admite o diagrama de ocorrências que aparece na figura. Os relacionamentos mostrados informam que o empregado **e1** é supervisor do empregado **e3**, que por sua vez é supervisor de **e5**, o qual, por sua vez é supervisor de **e1**. Isso obviamente contraria nosso conhecimento sobre a realidade modelada, já que, em uma hierarquia de supervisão, não é permitido que um superior hierárquico (no caso **e1**) apareça como supervisionado em um nível mais baixo da hierarquia (no caso, **e1** aparece como supervisionado de **e5**). Essa restrição, ao contrário do exemplo do casamento apresentado acima, não é possível de se introduzir no DER através de modificações. O problema é que a restrição é uma restrição de integridade *recursiva* e estas não podem ser representadas através de diagramas ER. Neste caso, resta apenas especificar a restrição a parte do DER.

### 3.1.3 Diferentes modelos podem ser equivalentes

Na prática, muitas vezes observa-se analistas em acirradas discussões a fim de decidir como um determinado objeto da realidade modelada deve aparecer no modelo. Às vezes, tais discussões são absolutamente supérfluas, pois os diferentes modelos ER, em qualquer das opções defendidas pelos diferentes analistas, geram o mesmo banco de dados.

Há um conceito de *equivalência* entre modelos ER. De maneira informal, diz-se que dois modelos são equivalentes, quando expressam o mesmo, ou seja quando modelam a mesma realidade.

Para definir o conceito de equivalência de forma mais precisa, é necessário considerar o BD que é projetado a partir do modelo ER. Para fins de projeto de BD, dois modelos ER são equivalentes, quando ambos geram o mesmo esquema de BD. Assim, para analisar se dois modelos são equivalentes, é ne-

É claro que para entender perfeitamente este conceito de equivalência de modelos, o leitor deve conhecer as regras de tradução apresentadas no Capítulo 5. Mesmo assim, considerando a definição informal de equivalência (dois modelos que expressam o mesmo), é possível compreender alguns casos da aplicação do conceito.

Os dois modelos são equivalentes, pois expressam o mesmo e geram o mesmo banco de dados.



A transformação de um relacionamento n:n em entidade segue o seguinte processo:

1. O relacionamento  $n:n$  é representado como uma entidade.
2. A entidade criada é relacionada às entidades que originalmente participavam do relacionamento.
3. A entidade criada tem como identificador:
  - as entidades que originalmente participavam do relacionamento
  - os atributos que eram identificadores do relacionamento original (caso o relacionamento original tivesse atributos identificadores)
4. As cardinalidades da entidade criada nos relacionamentos de que participa é sempre (1,1).
5. As cardinalidades das entidades que eram originalmente associadas pelo relacionamento transformado em entidade são transcritas ao novo modelo conforme mostrado na Figura 3.3.

Como um relacionamento  $n:n$  pode ser transformado em entidade, é possível construir modelos sem relacionamentos  $n:n$ . Neste fato, baseiam-se diversas variantes da abordagem ER, que excluem relacionamentos  $n:n$ , ou excluem apenas relacionamentos  $n:n$  com atributos. Um exemplo são várias abordagens baseadas na Engenharia de Informações (ver Seção 3.4.1.1).

Um outro caso de equivalência, é entre um relacionamento de cardinalidade 1:1 e com cardinalidade mínima “1” em ambos os lados pode ser substituído por uma única entidade.

## 3.2 IDENTIFICANDO CONSTRUÇÕES

Apenas observando um determinado objeto da realidade modelada, é difícil determinar se tal objeto deve ser modelado como uma entidade, um atributo ou um relacionamento. Para definir que construção de ER será usada na modelagem do objeto, é necessário conhecer o seu contexto, isto é, o modelo dentro do qual ele aparece. Assim, a recomendação geral que se dá para a identificação de objetos é a de considerar a decisão por um ou outro tipo de representação no modelo (entidade, relacionamento, atributo) como sujeita a alteração durante a modelagem. Não é aconselhável despendar um tempo excessivo em longas discussões sobre como modelar um objeto. O próprio desenvolvimento do modelo e o aprendizado sobre a realidade irão refinando e aperfeiçoando o modelo. Mesmo assim, existem alguns indicativos para a escolha de construções de modelagem. Estes indicativos passam a ser descritos a seguir.

### 3.2.1 Atributo versus entidade relacionada

Uma questão que às vezes surge na modelagem de um sistema é entre modelar um objeto como sendo um atributo de uma entidade ou como sendo uma entidade autônoma relacionada a essa entidade.

Exemplificando, no caso de uma indústria de automóveis, como devemos registrar a cor de cada automóvel que sai da linha de produção? Caso considerarmos que cada automóvel possui uma única cor predominante, pode-se pensar em modelar a cor como um atributo da entidade AUTOMÓVEL (primeira opção da Figura 3.4). Outra opção seria modelar a cor

como uma entidade autônoma, que está relacionada à entidade **AUTOMÓVEL** (segunda opção da Figura 3.4).

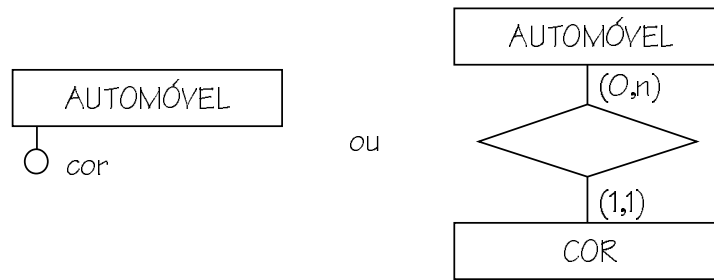


Figura 3.4: Escolhendo entre atributo e entidade relacionada

Alguns critérios para esta decisão são:

- ❑ Caso o objeto cuja modelagem está em discussão esteja vinculado a outros objetos (atributos, relacionamentos, entidades genéricas ou especializadas), o objeto deve ser modelado como entidade, já que um atributo não pode ter atributos, nem estar relacionado a outras entidades, nem ser generalizado ou especializado. Caso contrário, o objeto pode ser modelado como atributo. Assim, no caso do exemplo das cores dos automóveis, poder-se-ia optar por modelar a cor como uma entidade, caso se tivesse que registrar no banco de dados os possíveis fabricantes da tinta da referida cor (entidades relacionadas a cor), ou caso se quisesse registrar as datas de início e fim (atributos) do uso de uma determinada cor. Caso não houvesse nenhum objeto relacionado à cor do automóvel, poder-se-ia modelá-la como atributo da entidade automóvel.
- ❑ Quando o conjunto de valores de um determinado objeto é fixo durante toda a vida do sistema ele pode ser modelado como atributo, visto que o domínio de valores de um atributo é imutável. Quando existem transações no sistema que alteram o conjunto de valores do objeto, o mesmo não deve ser modelado como atributo. Assim, retomando o exemplo das cores dos automóveis, caso existissem transações de criação/eliminação de cores, seria preferível a modelagem de cor como entidade relacionada a entidade automóvel.

### 3.2.2 Atributo versus generalização/especialização

Um outro conflito de modelagem para o qual há algumas regras de cunho prático é entre modelar um determinado objeto (por, exemplo, a categoria funcional de cada empregado de uma empresa) como atributo (categoria funcional como atributo da entidade **EMPREGADO** - Figura 3.5) ou através de uma especialização (cada categoria funcional corresponde a uma especialização da entidade empregado - Figura 3.6).

Uma especialização deve ser usada quando sabe-se que as classes especializadas de entidades possuem propriedades (atributos, relacionamentos, generalizações, especializações) particulares. Assim, no caso do exemplo acima, faz sentido especializar a entidade empregado de acordo com a categoria funcional, no caso de as classes particulares possuírem atributos ou relacionamentos próprios (um exemplo é o caso da Figura 3.5).

Ainda no exemplo dos empregados, o sexo do empregado é melhor modelado como atributo de empregado, caso não existam propriedades particulares de homens e mulheres a modelar na realidade considerada.

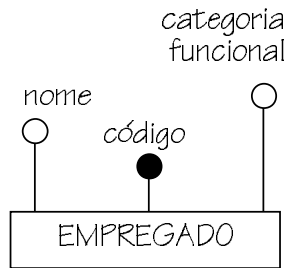


Figura 3.5: Modelando categoria funcional como atributo

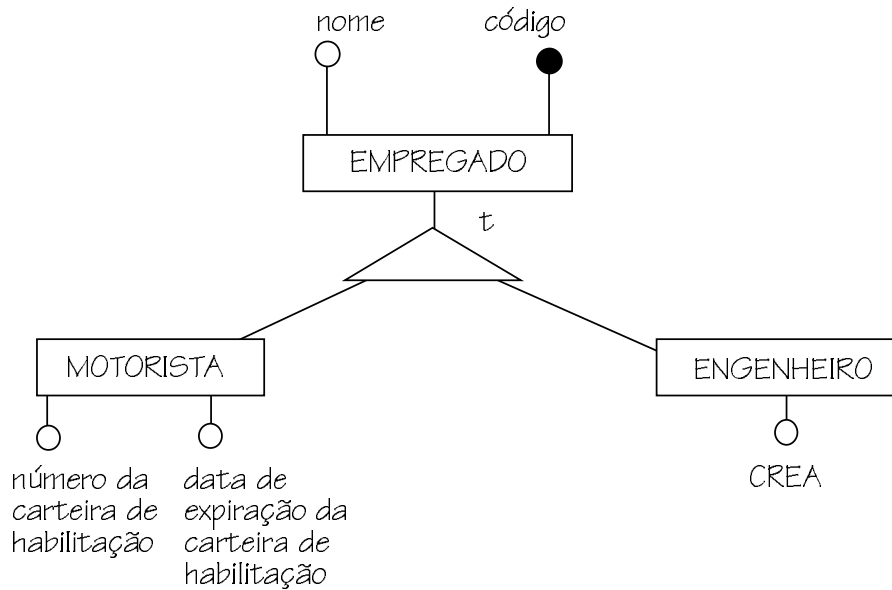


Figura 3.6: Modelando categoria funcional como especialização

### 3.2.3 Atributos opcionais e multi-valorados

Conforme vimos no capítulo anterior (Seção 2.3), atributos podem ser *opcionais* (nem toda ocorrência da entidade possui um valor do atributo) ou *multi-valorados*. Entretanto, quando se inicia o processo de modelagem é aconselhável tentar restringir-se ao uso de atributos *obrigatórios* e *mono-valorados* pelas razões abaixo listadas.

#### 3.2.3.1 Atributo opcional

Em muitos casos, atributos opcionais indicam subconjuntos de entidades que são modelados mais corretamente através de especializações. O DER da Figura 3.7 apresenta uma entidade com diversos atributos opcionais. Segundo este DER, nem todo empregado possui um registro no CREA (Conselho Regional de Engenharia e Arquitetura), nem todo empregado possui um registro no CRM (Conselho Regional de Medicina), etc. Entretanto, o modelo não representa que combinações de atributos são válidas. Será que um empregado



pode possuir atributos CREA, CRM, data de expiração da carteira de habilitação mas não possuir o número da carteira?

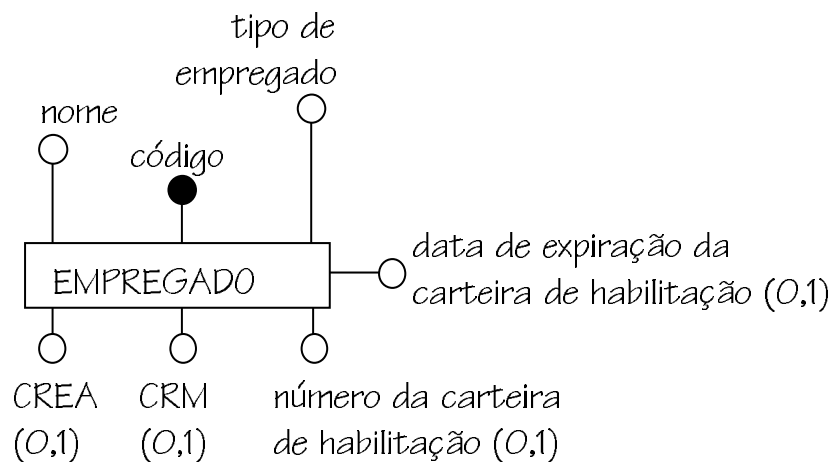


Figura 3.7: Atributos opcionais

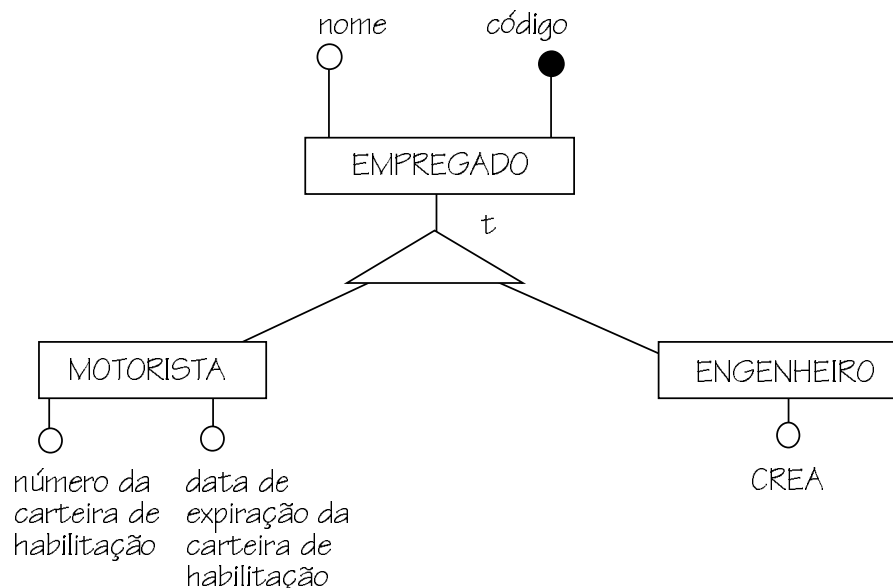


Figura 3.8: Aumentando a fidelidade do modelo através de especializações

O problema que ocorre no exemplo, é que o uso de atributos opcionais esconde as diferentes categorias de empregados e suas entidades. A realidade é modelada com maior fidelidade, caso a entidade **EMPREGADO** seja especializada conforme mostra a figura 4.7. Neste modelo fica claro quais os atributos de cada um dos subconjuntos particulares de **EMPREGADO**. Assim, toda vez que aparecer um atributo opcional é aconselhável verificar se a modelagem através de entidades especializadas não é mais conveniente.

### 3.2.3.2 Atributo multi-valorado

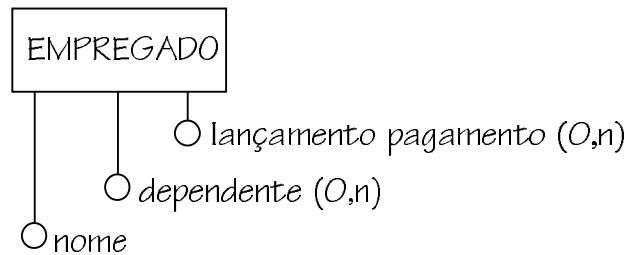


Figura 3.9: Usando atributos multi-valorados

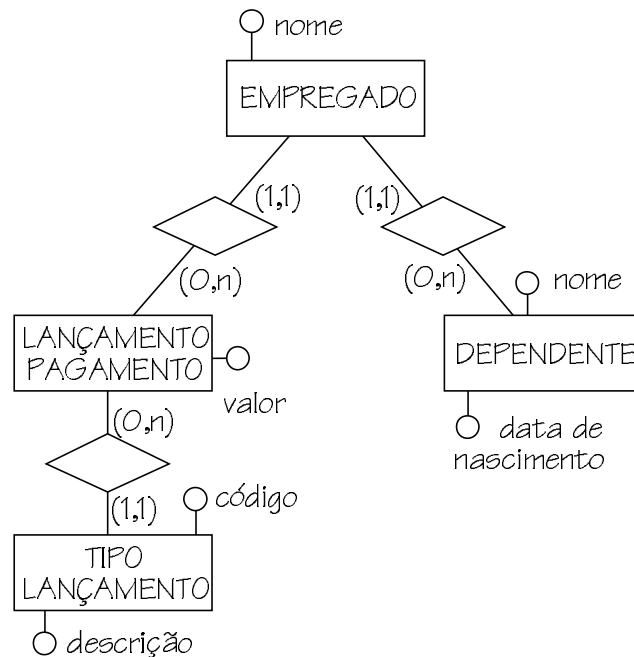


Figura 3.10: Substituindo atributos multi-valorados por entidades relacionadas

Atributos multi-valorados são indesejáveis por duas razões:

- ❑ Nos SGBD relacionais que seguem o padrão SQL/2, atributos multi-valorados não possuem implementação direta. Não existe em um SGBD relacional uma construção como os “arrays” de Pascal, nem como os grupos repetidos de COBOL. Assim, caso esteja sendo projetada um banco de dados relacional, é aconselhável usar apenas atributos mono-valorados.
- ❑ Atributos multi-valorados podem induzir a um erro de modelagem, que é o de ocultar entidades e relacionamentos em atributos multi-valorados. A Figura 3.9 mostra um DER em que são modelados empregados. Como atributos de **EMPREGADO** aparecem o **nome** do empregado, seus **dependentes** e os **lançamentos de pagamento** que compõem seu contracheque. Entretanto, ao considerar a entidade **EMPREGADO** mais detalhadamente, observa-se que tanto **dependentes**, quanto os **lançamentos** possuem propriedades particulares. Cada dependente possui um nome e uma data de nascimento. Já um lançamento de pagamento possui um valor e um tipo

de lançamento, sobre o qual também nos interessa manter informações, como o código do tipo e sua descrição. Assim, o modelo correto para a realidade em questão é o apresentado na Figura 3.10.

### 3.3 VERIFICAÇÃO DO MODELO

Uma vez confeccionado, um modelo ER deve ser validado e verificado. A verificação é o controle de qualidade que procura garantir que o modelo usado para a construção do banco de dados gerará um bom produto. Um modelo, para ser considerado bom, deve preencher uma série de requisitos, como ser completo, ser correto e não conter redundâncias.

#### 3.3.1 Modelo deve ser correto

Um modelo está correto quando não contém erros de modelagem, isto é, quando os conceitos de modelagem ER são corretamente empregados para modelar a realidade em questão. Pode-se distinguir entre dois tipos de erros, os erros *sintáticos* e os erros *semânticos*. Erros sintáticos ocorrem quando o modelo não respeita as regras de construção de um modelo ER. Exemplos de erros sintáticos são o de associar atributos a atributos, o de associar relacionamentos a atributos, o de associar relacionamentos através de outros relacionamentos ou de especializar relacionamentos ou atributos. Já erros semânticos ocorrem quando o modelo, apesar de obedecer as regras de construção de modelos ER (estar sintaticamente correto) reflete a realidade de forma inconsistente. Alguns exemplos de erros semânticos praticados frequentemente são:

- ❑ *Estabelecer associações incorretas.*

Um exemplo é associar a uma entidade um atributo que na realidade pertence a outra entidade. Por exemplo, em um modelo com entidades **CLIENTE** e **FILIAL**, associar a **CLIENTE** o nome da filial com o qual o cliente trabalha usualmente (nome de filial é um atributo de **FILIAL**).

- ❑ *Usar uma entidade do modelo como atributo de outra entidade.*

Um exemplo seria ter, em um modelo, uma entidade **BANCO** e usar banco como atributo de uma outra entidade **CLIENTE**. Cada objeto da realidade modelada deve aparecer uma única vez no modelo ER.

- ❑ *Usar o número incorreto de entidades em um relacionamento.*

Um exemplo é o de fundir em um único relacionamento ternário dois relacionamentos binários independentes.

As regras de normalização de bases de dados relacionais apresentadas no próximo capítulo servem também para verificar a correção de modelos ER.

#### 3.3.2 Modelo deve ser completo

Um modelo completo deve fixar todas propriedades desejáveis do banco de dados. Isso obviamente somente pode ser verificado por alguém que conhece profundamente o sistema a ser implementado. Uma boa forma de verificar se o modelo é completo é verificar se todos os dados que devem ser obtidos do banco de dados estão presentes e se todas as transações de modificação do banco de dados podem ser executadas sobre o modelo.

Este requisito é aparentemente conflitante com a falta de poder de expressão de modelos ER que foi citada acima. Quando dizemos que um mo-

delo deve ser completo, estamos exigindo que todas propriedades *expressáveis* com modelos ER apareçam no modelo.

### 3.3.3 Modelo deve ser livre de redundâncias

Um modelo deve ser mínimo, isto é não deve conter conceitos redundantes.

Um tipo de redundância que pode aparecer é a de relacionamentos redundantes. *Relacionamentos redundantes* são relacionamentos que são resultado da combinação de outros relacionamentos entre as mesmas entidades. A Figura 3.11 apresenta um DER com relacionamentos redundantes.

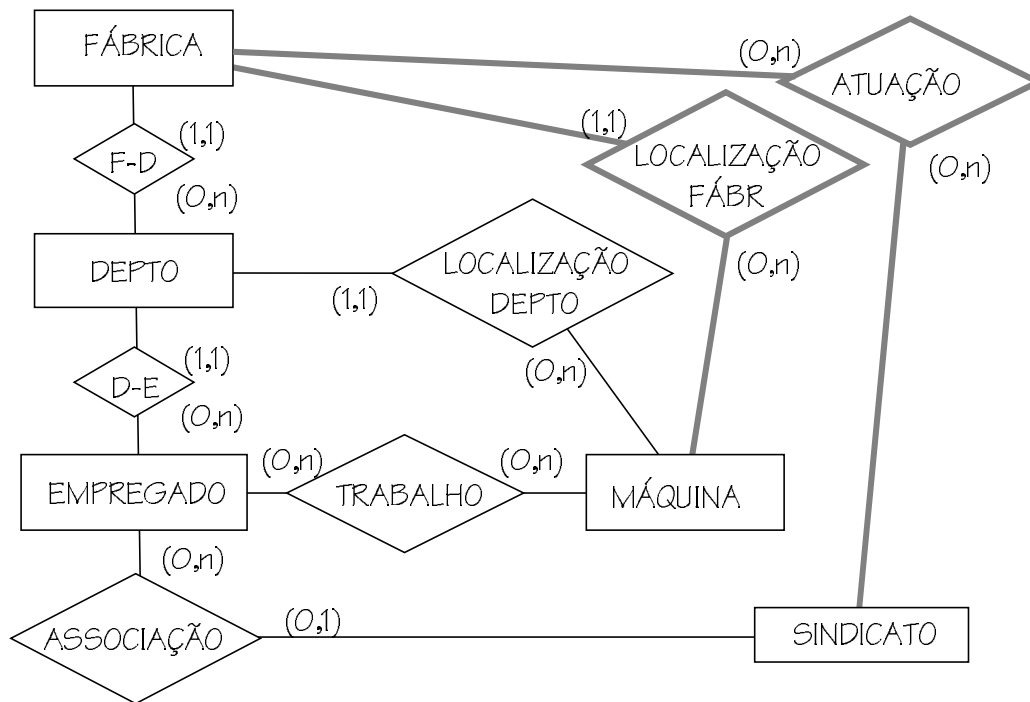


Figura 3.11: Relacionamentos redundantes

Os relacionamentos desenhados em linhas densas no DER da Figura 3.11 são redundantes. O relacionamento **LOCALIZAÇÃO-FÁBR** entre **MÁQUINA** e **FÁBRICA** é redundante. Um relacionamento é redundante, quando é possível eliminá-lo do diagrama ER, sem que haja perda de informações no banco de dados. No caso do relacionamento **LOCALIZAÇÃO-FÁBR**, a associação entre entidades por ele expressa já está contida nos relacionamentos **LOCALIZAÇÃO-DEPT** e **F-D**. Em outros termos, como o banco de dados informa em que departamento uma máquina está localizada e em que fábrica o departamento está localizado, ela informa por consequência em que fábrica uma máquina está localizada. Seguindo raciocínio análogo é fácil verificar que o relacionamento **ATUAÇÃO**, entre **FÁBRICA** e **SINDICATO** é redundante, pois a informação nele contida é derivada da informação fornecida pelos relacionamentos **ASSOCIAÇÃO**, **D-E** e **F-D**.

Um outro tipo de redundância que pode aparecer em modelos ER é a de atributos redundantes. *Atributos redundantes* são atributos deriváveis a partir da execução de procedimentos de busca de dados e/ou cálculos sobre o banco de dados.

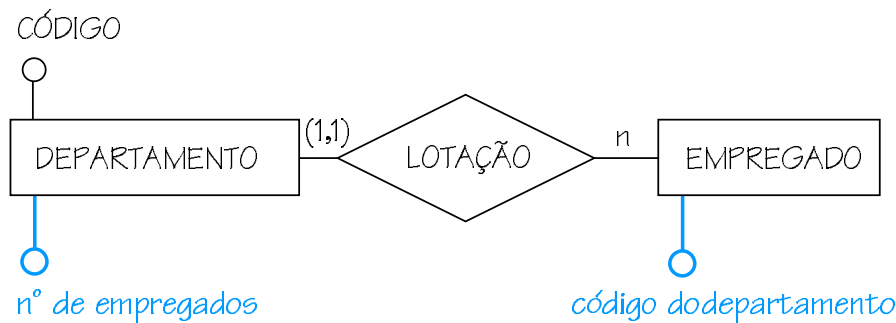


Figura 3.12: Atributos redundantes

A Figura 3.12 apresenta um DER que contém dois atributos redundantes (*n° de empregados* e *código do departamento*). O atributo *código do departamento* é redundante pois pode ser obtido através do acesso à entidade **DEPARTAMENTO** associada à entidade **EMPREGADO** através do relacionamento de **LOTAÇÃO**<sup>5</sup>. Já o atributo *n° de empregados* é redundante pois pode ser obtido através de um processo de contagem sobre o relacionamento **LOTAÇÃO**.

Construções redundantes devem ser omitidas do modelo ER. Como o diagrama ER não distingue construções derivadas das demais construções, o projetista do banco de dados poderia ser levado a implementá-las gerando redundância não controlada de dados. Redundância não controlada de dados em um banco de dados é indesejável por diversas razões. Construções redundantes normalmente implicam desperdício de espaço de armazenamento. Construções redundantes podem resultar em informações incorretas, caso não sejam alteradas em sincronia com as informações das quais são derivadas. Observe que isso não significa que redundância *controlada* de dados (redundância de dados da qual programas e usuários têm conhecimento) deva também ser necessariamente evitada. Às vezes, construções redundantes em um banco de dados podem servir para aumentar a performance de operações de busca de informações no banco de dados, mas nem por isso devem aparecer no modelo conceitual do banco de dados.

### 3.3.4 Modelo deve refletir o aspecto temporal

Usualmente, ao iniciar a modelagem ER, a preocupação é obter um modelo que descreva os estados válidos e corretos do banco de dados. O primeiro modelo tende a refletir um estado momentâneo do banco de dados. Entretanto, é necessário lembrar que assim como informações são incluídas no banco de dados, elas também podem ter que ser eliminadas do banco de dados. Um banco de dados não pode crescer indefinidamente. Informações ultrapassadas ou desnecessárias podem ser eliminadas. Portanto, é necessário considerar o *aspecto temporal* na modelagem de dados. Não há regras gerais de

<sup>5</sup>Para o leitor afeito a terminologia de bancos de dados relacionais, *código do departamento* é uma assim chamada *chave estrangeira*. Este atributo é usado em um banco de dados relacional, como veremos nos próximos capítulos, para implementar o relacionamento **LOTAÇÃO**. Como no modelo ER ainda não entramos em detalhes de implementação com SGBD relacional, o atributo deve ser omitido.

como proceder neste caso, mas é possível identificar alguns padrões que repetem-se freqüentemente na prática.

### 3.3.4.1 Atributos cujos valores modificam ao longo do tempo

Alguns atributos de uma entidade, normalmente aqueles que não são identificadores da entidade, podem ter seus valores alterados ao longo do tempo (por exemplo, o endereço de um cliente pode ser modificado). Algumas vezes, por questões de necessidades futuras de informações, ou até mesmo por questões legais, o banco de dados deve manter um registro histórico das informações. Um exemplo é o valor do salário de um empregado. Num sistema de pagamento, não interessa saber apenas o estado atual, mas também o salário durante os últimos meses, por exemplo, para emitir uma declaração anual de rendimentos de cada empregado. Assim, salário não pode ser modelado como um atributo, mas sim como uma entidade. A figura 4.12 apresenta as duas alternativas de modelagem.

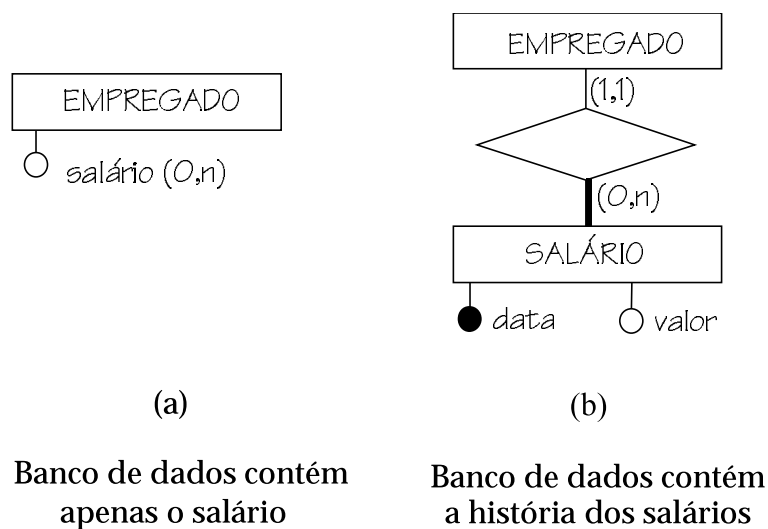


Figura 3.13: Modelando a dimensão temporal de atributos

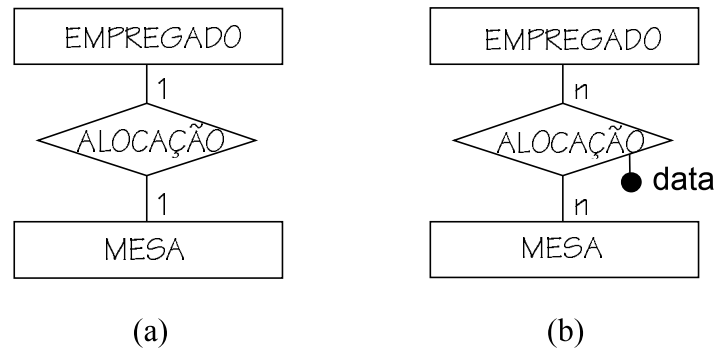
### 3.3.4.2 Relacionamentos que modificam ao longo do tempo

Assim como atributos podem ter seus valores modificados ao longo do tempo, também relacionamentos podem ser modificados e também neste caso pode ser requerido que o banco de dados mantenha um registro histórico das alterações.

Em geral, relacionamentos que, ao considerar apenas o estado atual do banco de dados, possuem cardinalidade 1:1 ou 1:n são transformados em cardinalidade n:n, quando é considerada a história das alterações de relacionamento.

Para exemplificar, consideramos o relacionamento **ALOCAÇÃO** da Figura 3.14(a). Este relacionamento possui cardinalidade 1:1, ou seja, cada empregado está alocado a no máximo uma mesa e cada mesa tem a ela alocado no máximo um empregado. Este modelo está correto caso deseje-se armazenar no banco de dados apenas a alocação atual de cada mesa. Entretanto, caso deseje-se armazenar também a história das alocações, isto é, que empregados

estiveram alocados a que mesas ao longo do tempo, é necessário modificar o modelo (Figura 3.14 (b)). O relacionamento passa a ter cardinalidade  $n:n$ , já que, ao longo do tempo um empregado pode ter sido alocado a diversas mesas e uma mesa pode ter tido a ela alocados muitos empregados. Como um mesmo empregado pode ter sido alocado a mesma mesa múltiplas vezes, torna-se necessário um atributo identificador do relacionamento, afim de distinguir uma alocação de um determinado empregado a uma mesa, das demais alocações deste empregado à mesma mesa. Com isso surge o atributo identificador **data**.



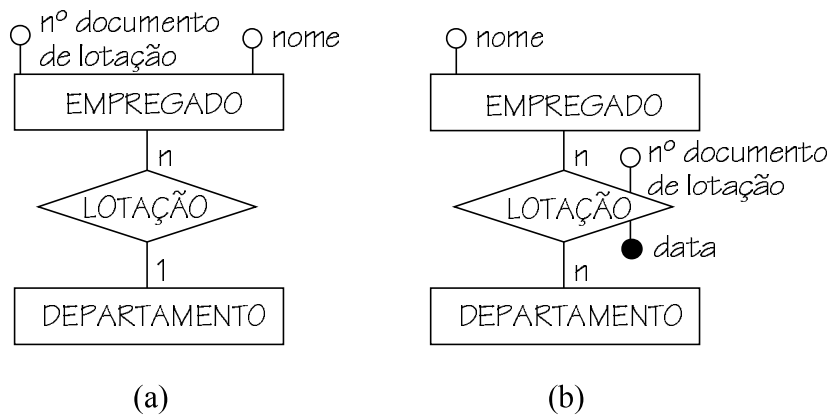
Base de dados contém apenas a alocação atual

Base de dados contém a história das alocações

Figura 3.14: Modelando a dimensão temporal de relacionamentos 1:1

A Figura 3.15 apresenta uma situação semelhante, agora considerando um relacionamento de cardinalidade 1:n. Se quisermos considerar a história das lotações de empregados ao longo do tempo, é necessário transformar o relacionamento para a cardinalidade  $n:n$ , já que ao longo do tempo um empregado pode ter atuado em diferentes departamentos. Neste caso, pode ocorrer também que atributos da entidade **EMPREGADO** migrem para o relacionamento. No caso do exemplo, isto ocorre com o atributo **nº documento de lotação**. Este atributo, que, na primeira versão do modelo, aparece na entidade **EMPREGADO**, migra, na nova versão, para o relacionamento, já que na nova versão um empregado pode estar relacionado com múltiplos departamentos.

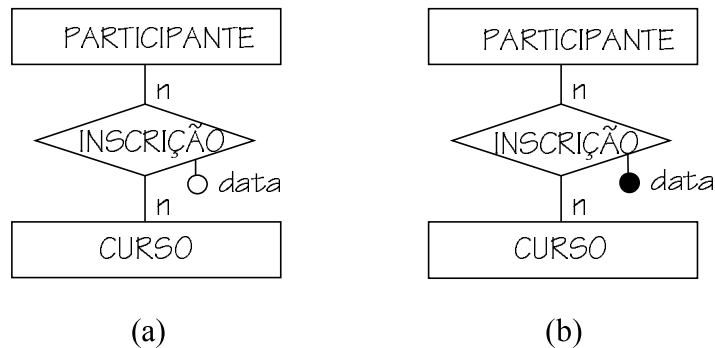
A Figura 3.16 apresenta mais um caso de consideração de história de relacionamentos. Neste caso, trata-se de um relacionamento de cardinalidade  $n:n$ . Modela-se a inscrição de participantes nos cursos oferecidos por uma empresa de treinamento. Na primeira versão, considera-se apenas os cursos nos quais uma pessoa está inscrita em um determinado instante no tempo. Na segunda versão, considera-se todas as inscrições, inclusive as do passado. A modificação de uma versão para a outra consta da transformação do atributo **data** em atributo identificador. Isto ocorre porque, na segunda versão, um participante pode aparecer relacionado múltiplas vezes a um determinado curso (caso ele tenha se inscrito múltiplas vezes no curso). O atributo passa a distinguir uma inscrição de uma pessoa em um curso, das demais inscrições desta pessoa no mesmo curso.



Base de dados contém apenas a lotação atual

Base de dados contém a história das lotações

Figura 3.15: Modelando a dimensão temporal de relacionamentos 1:n



Base de dados contém apenas a inscrição atual

Base de dados contém a história das inscrições

Figura 3.16: Modelando a dimensão temporal de relacionamentos n:n

### 3.3.4.3 Consultas a dados referentes ao passado

Muitas vezes, para evitar o crescimento desmedido do banco de dados, informações referentes ao passado são eliminadas. Entretanto, estas informações podem ser necessárias no futuro, por exemplo, por motivos legais, para realização de auditorias ou para tomada de decisões. Portanto, é necessário planejar desde a modelagem, por quanto tempo as informações ficarão armazenadas no banco de dados. Caso informações antigas fiquem no banco de dados, podem ser necessários atributos para indicar o status da informação, se atual ou antiga.

#### ❑ Planejar o arquivamento de informações antigas

Para as informações que serão retiradas do banco de dados e armazenadas em arquivos convencionais, é necessário fazer um planejamento de como estas informações serão acessadas no futuro, caso venham a ser necessárias. Uma solução que poderia ser considerada, é a de reincluir as informações no banco de dados, quando elas forem necessárias no futuro. Isso permite que, para buscar as informações passadas, sejam usados os



mesmos procedimentos que são usados para acessar as informações atuais. Entretanto, é necessário considerar que as informações em um banco de dados estão normalmente relacionadas a outras. Caso as ocorrências de entidade que se deseja devolver à base de dados estejam relacionadas a outras ocorrências, é necessário que estas estejam presentes. Se elas também tiverem sido excluídas, deverão ser igualmente devolvidas à base de dados. Essa inclusão pode ser propagada em cascata para outras entidades.

#### □ *Planejar informações estatísticas*

Em alguns casos, informações antigas são necessárias apenas para tomada de decisões. Neste caso, muitas vezes deseja-se apenas dados resultantes de cálculos ou estatísticas sobre as informações, como totais, contagens, médias,... Assim pode ser conveniente manter no banco de dados estas informações compiladas e eliminar as informações usadas na compilação.

### **3.3.5 Entidade isolada e entidade sem atributos**

Uma entidade isolada é uma entidade que não apresenta nenhum relacionamento com outras entidades. Em princípio, entidades isoladas não estão incorretas. Uma entidade que muitas vezes aparece isolada é aquela que modela a organização na qual o sistema implementado pelo BD está embutida. Tomemos como exemplo o BD de uma universidade que aparece na Figura 2.13. A entidade **UNIVERSIDADE** pode ser necessária, caso se deseje manter no BD alguns atributos da universidade. Mesmo assim, o modelo não deveria conter o relacionamento desta entidade com outras, como **ALUNO** ou **CURSO**. Como o BD modela uma única universidade, não é necessário informar no BD em que universidade o aluno está inscrito ou a qual universidade o curso pertence.

Mesmo assim, a ocorrência de entidades isoladas em modelos na prática é rara e por isso deve ser investigada em detalhe, para verificar se não foram esquecidos relacionamentos.

Uma outra situação que não está incorreta, mas deve ser investigada, é a de uma entidades sem atributos.

## **3.4 ESTABELECENDO PADRÕES**

Modelos de dados são usados para comunicação entre as pessoas da organização (usuários, analistas, programadores,...) e até mesmo para a comunicação com programas (ferramentas CASE, geradores de código,...). Assim, ao usar modelagem de dados, é necessário estabelecer padrões de confecção de modelos. Infelizmente, na prática e na literatura não aparece uma versão apenas de modelo ER, mas muitas, que distinguem-se umas das outras não só na representação gráfica, isto é em sua sintaxe, mas também na semântica. Nesta seção discutimos algumas variantes da abordagem ER e como garantir a utilização de uma variante escolhida.

### **3.4.1 Variantes de modelos ER**

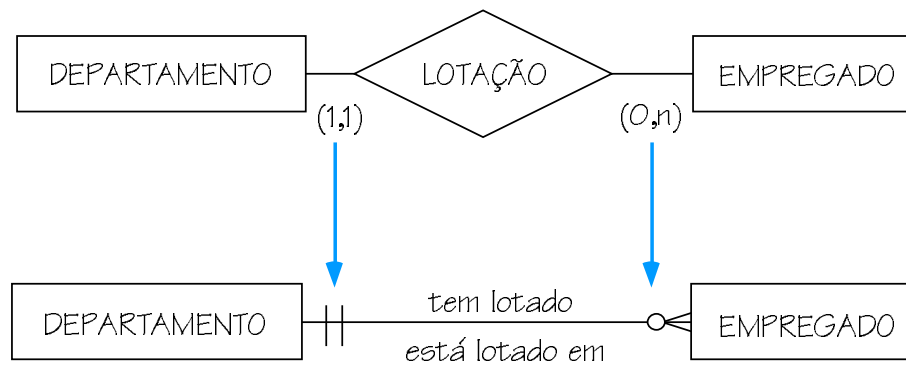
Quando de seu surgimento na literatura, a abordagem ER não era ainda suportada por ferramentas em computador de edição e manipulação, como as

ferramentas CASE hoje existentes. Com isso, cada autor de livro sobre o assunto, bem como cada usuário da abordagem tinha total liberdade para escolher uma representação gráfica e até mesmo uma semântica para a abordagem. A consequência é que hoje observa-se uma grande variedade de abordagens que levam o título de entidade-relacionamento. Até este ponto do livro, apresentamos a abordagem ER na forma que aparece mais frequentemente na literatura e que é muitas vezes usada na prática. Essa notação é às vezes referida como notação tipo “Chen” pois, com algumas extensões, segue a notação proposta por Peter Chen em seu primeiro artigo sobre a abordagem.

Além desta notação duas famílias de notações têm importância, a notação *Engenharia de Informações* e a notação *MERISE*.

### 3.4.1.1 Notação Engenharia de Informações

A Engenharia de Informações é uma metodologia de desenvolvimento de sistemas de informação proposta no início da década de 80 por Martin e Finkelstein. Essa metodologia tem uma importância considerável na prática, comprovada pela existência de ferramentas CASE que a suportam. A característica mais importante desta metodologia é seu embasamento na modelagem de dados. A organização é vista fundamentalmente através do modelo de dados.



Notação para cardinalidade máxima e mínima:

- | Cardinalidade (mínima, máxima) 1
- Cardinalidade mínima 0
- ≧ Cardinalidade máxima n

Figura 3.17: Notação Engenharia de Informações

Para a Engenharia de Informações, foi definida uma notação especial de diagramas ER, conhecida como notação Engenharia de Informações, ou também notação James Martin.

A Figura 3.17 apresenta um exemplo de um DER na notação Chen até aqui empregada e o correspondente DER na notação Engenharia de Informações. As diferenças principais são as seguintes:

- ❑ Na notação Engenharia de Informações, relacionamentos são representados apenas por uma linha que liga os símbolos representativos das entidades associadas. Isso têm as seguintes consequências:
  - A notação admite apenas relacionamentos binários, já que uma linha apenas conecta duas entidades. Relacionamentos ternários ou de grau maior são modelados através de uma entidade associada através de relacionamentos binários a cada uma das entidades que participam do relacionamento ternário.
  - Atributos aparecem exclusivamente em entidades. Com isso, objetos que seriam modelados como relacionamentos n:n na notação de Chen tendem a ser modelados como entidades na notação de Engenharia de Informações.
- ❑ A denominação de um relacionamento é escrita na forma de verbos em ambas direções de leitura (DEPARTAMENTO *tem lotado* EMPREGADO, EMPREGADO *está lotado em* DEPARTAMENTO).
- ❑ A notação para cardinalidade máxima e mínima é gráfica. O símbolo mais próximo do retângulo representativo da entidade corresponde a cardinalidade máxima, o mais distante a cardinalidade mínima.
- ❑ A generalização/especialização é chamada de subconjunto (subtipo) de entidades e é representada através do aninhamento dos símbolos de entidade conforme mostra a Figura 3.18.

A Figura 3.18 apresenta um exemplo mais abrangente representado com a notação da Engenharia de Informações (trata-se do mesmo modelo ER apresentado na Figura 2.37)

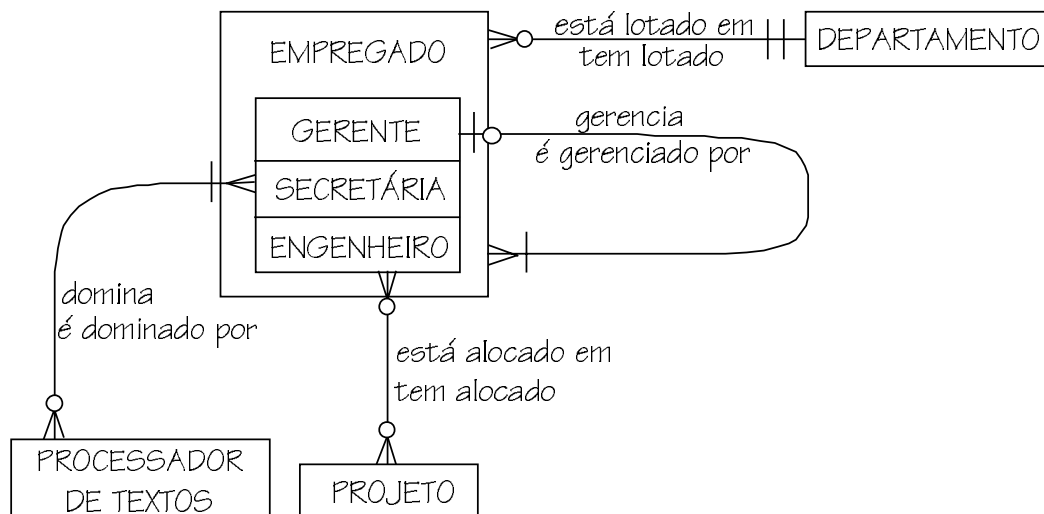


Figura 3.18: Modelo ER da Figura 2.37 na notação de Engenharia de Informações

### 3.4.1.2 Notação MERISE

MERISE é a metodologia de desenvolvimento de sistemas muito popular na França. Uma etapa desta metodologia, é a de modelagem de dados. Na modelagem de dados, MERISE adota diagramas ER, com uma notação um pouco

diferente da usual. Em princípio, esta notação não seria de maior importância para o leitor deste texto. Entretanto, há um grupo padronização na ISO (“International Standards Organization”) que está estudando a padronização de notações de DER. Relatórios preliminares deste grupo indicam que uma das técnicas que pode vir a ser adotada é a empregada em MERISE.

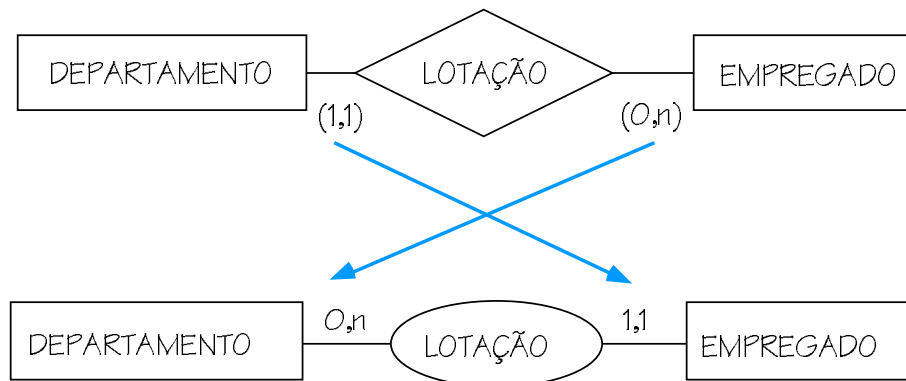


Figura 3.19: Notação Chen e correspondente notação MERISE

A Figura 3.19 apresenta um DER na notação Chen e o DER correspondente na notação MERISE. Como se observa, além da modificação cosmética de representar o relacionamento por uma elipse, ao invés de um losango, a posição em que são indicadas as cardinalidades mínima e máxima mudou. O motivo é que a interpretação dada a cardinalidade em MERISE é diferente da usual. Na interpretação usual de cardinalidade, ela indica quantas ocorrências de entidade (no mínimo e no máximo) podem estar associadas a uma ocorrência de determinada entidade. Esta interpretação é a chamada de semântica *associativa*. Em MERISE usa-se a semântica *participativa*. Nesta a cardinalidade indica quantas vezes uma ocorrência de entidade *participa* de um relacionamento. Exemplificando, na figura acima na notação Chen, a cardinalidade (1,1) representa o fato de um empregado estar vinculado a no mínimo um e no máximo um departamento. Já no diagrama em MERISE, a anotação 1,1 indica que uma ocorrência da entidade EMPREGADO participa no mínimo uma e no máximo uma vez do relacionamento LOTAÇÃO.

### 3.4.2 Uso de ferramentas de modelagem

Na prática, não é aceitável que o diagrama ER seja confeccionado manualmente. A criação de um DER à mão é muito trabalhosa, pois, durante o processo de modelagem, as revisões são frequentes. Além disso, dificilmente diagramas feitos à mão serão atualizados, quando de alterações do modelo durante a sua vida útil. Portanto, é recomendável que desde o início da confecção do DER, seja usada uma ferramenta em computador para apoio à modelagem. Podem ser consideradas duas alternativas:

#### 3.4.2.1 Uso de uma ferramenta CASE

O software ideal para acompanhar o projeto de um banco de dados é uma ferramenta CASE (CASE - “computer aided software engineering”). Uma ferramenta CASE pode apoiar o desenvolvimento de um banco de dados tanto a

nível de modelagem quanto a nível de projeto do banco de dados. Do ponto de vista da modelagem o mínimo que se deve exigir de uma ferramenta CASE é:

☐ *Capacidade de edição diagramática*

A ferramenta CASE deve oferecer uma interface gráfica poderosa e fácil de usar, que permita construir o DER diretamente no computador. A modificação do DER (inclusão/eliminação/movimentação) de símbolos deve ser simplificada.

☐ *Dicionário de dados*

A ferramenta deve possuir um *dicionário de dados*, isto é, um pequeno banco de dados, onde toda descrição do DER está armazenada.

☐ *Integração entre o diagrama ER e o dicionário de dados*

O dicionário de dados deve ser integrado ao DER. A partir do DER deve ser possível visualizar e editar as entradas do dicionário de dados correspondentes a elementos selecionados do DER.

### 3.4.2.2 Uso de programas de propósito geral

Em organizações com pequeno orçamento para a Informática, pode ser difícil obter os recursos necessários à aquisição de uma ferramenta CASE. Neste caso, pode-se usar programas de propósito geral para editar o DER e montar o dicionário de dados:

☐ *Edição do DER*

Para editar o DER pode-se utilizar um programa de desenho de propósito geral. Há programas de desenho que trabalham com os conceitos de nodo e de arco. Nestes programas, quando o usuário mover um nodo do diagrama, todos arcos ligados a este nodo são também movimentados. Este tipo de programas são particularmente adequados para edição de diagramas ER.

☐ *Dicionário de dados*

Para construir o dicionário de dados pode-se utilizar um processador de textos, uma planilha eletrônica ou um banco de dados (esta é a melhor opção). A escolha provavelmente irá recair sobre o programa que for mais conhecido e dominado na organização em questão.

## 3.5 ESTRATÉGIAS DE MODELAGEM

O processo de construção de um modelo é um processo incremental, isto é, um modelo de um sistema não é construído em um único passo, mas em muitos passos pequenos, muitas pequenas transformações do modelo inicial até o modelo completo. Gradativamente, o modelo vai sendo enriquecido com novos conceitos e estes vão sendo ligados aos existentes ou os existentes vão sendo aperfeiçoados. Uma *estratégia* de modelagem ER é uma sequência de passos (uma “receita-de-bolo”) de transformação de modelos. Estudando o processo de construção de modelos ER, diferentes autores propuseram diferentes estratégias, que apresentamos abaixo.

Na prática, observa-se que nenhuma das estratégias propostas na literatura é universalmente aceita. Normalmente, é aplicada uma combinação das

diversas estratégias de modelagem. Isso é compreensível, se considerarmos que o processo de modelagem é um processo de aprendizagem. Quando construímos o modelo de um sistema, estamos aprendendo fatos sobre aquele sistema. A sequência de idéias que se tem durante um processo de aprendizagem é dificilmente controlável por uma estratégia.

Para definir qual a estratégia a usar na construção de um modelo ER, deve-se identificar qual a fonte de informações principal para o processo de modelagem. São duas as fontes de informação que iremos considerar: descrições de dados existentes e o conhecimento que pessoas possuem sobre o sistema.

### **3.5.1 Partindo de descrições de dados existentes**

Uma opção de fonte de informações para o processo de modelagem de dados são descrições de dados já existentes.

Exemplificando, esta situação ocorre, quando deseja-se obter um modelo de dados para um sistema em computador existente. Neste caso usa-se como descrição dos dados as descrições dos arquivos utilizados pelo sistema em computador. Este caso é conhecido por *engenharia reversa*, pois objetiva obter uma especificação (o modelo) a partir de um produto existente (o sistema em computador).

Outro exemplo do uso de descrições de dados já existentes é quando são utilizadas descrições dos documentos (pastas, fichas, documentos fiscais,...) usados em um sistema não automatizado.

Para estes casos, a estratégia mais adequada é a estratégia “*bottom-up*” (de baixo para cima). Esta estratégia consta de partir de conceitos mais detalhados (“embaixo” em termos de níveis de abstração) e abstrair gradativamente. O processo de modelagem inicia com a identificação de atributos (conceitos mais detalhados). A partir daí, os atributos são agregados em entidades e as entidades são relacionadas e generalizadas. Essa forma de trabalhar é adequada quando já dispõe-se dos atributos. Normalmente, isso ocorre, quando já possui-se uma definição dos dados que deverão estar armazenados no banco de dados. Iremos tratar essa estratégia em detalhe em um capítulo a parte.

### **3.5.2 Partindo do conhecimento de pessoas**

A outra opção é partir do conhecimento que pessoas possuem sobre o sistema sendo modelado. Este é o caso, quando novos sistemas, para os quais não existem descrições de dados, estão sendo concebidos. Para este caso, podem ser aplicadas duas estratégias, a “*top-down*” e a “*inside-out*”.

#### **3.5.2.1 Estratégia “top-down”**

Esta estratégia consta de partir de conceitos mais abstratos (“de cima”) e ir gradativamente refinando estes conceitos em conceitos mais detalhados. Ou seja, segue-se o caminho inverso da estratégia “*bottom-up*”. Aqui o processo de modelagem inicia com a identificação de entidades genéricas (conceitos mais abstratos). A partir daí, são definidos os atributos das entidades, seus

relacionamentos, os atributos dos relacionamentos, e as especializações das entidades.

Uma sequência de passos para obter um modelo usando a estratégia “top-down” é a mostrada abaixo:

1. *Modelagem superficial* - Nesta primeira etapa, é construído um DER pouco detalhado (faltando domínios dos atributos e cardinalidades mínimas de relacionamentos) na seguinte sequência:
  - a) Enumeração das *entidades*.
  - b) Identificação dos *relacionamentos* e *hierarquias* de generalização/especialização entre as entidades.  
Para cada relacionamento identifica-se a *cardinalidade máxima*.
  - c) Determinação dos *atributos* de entidades e relacionamentos.
  - d) Determinação dos *identificadores* de entidades e relacionamentos.
  - e) O banco de dados é verificado quanto ao *aspecto temporal*.
2. *Modelagem detalhada* - Nesta etapa, completa-se o modelo com os domínios dos atributos e cardinalidades mínimas dos relacionamentos:
  - a) Adiciona-se os domínios dos atributos.
  - b) Define-se as cardinalidades mínimas dos relacionamentos. É preferível deixar estas cardinalidades por último, já que exigem conhecimento bem mais detalhado sobre o sistema, inclusive sobre as transações.
  - c) Define-se as demais restrições de integridade que não podem ser representadas pelo DER.
3. *Validação do modelo*
  - a) Procura-se construções redundantes ou deriváveis a partir de outras no modelo.
  - b) Valida-se o modelo com o usuário.

Em qualquer destes passos é possível retornar a passos anteriores. Exemplificando, durante a identificação de atributos é possível que sejam identificadas novas entidades, o que faria com que o processo retornasse ao primeiro passo.

### 3.5.2.2 Estratégia “inside-out”

A estratégia “inside-out” (de dentro para fora) consta de partir de conceitos considerados mais importantes (centrais, parte-se “de dentro”) e ir gradativamente adicionando conceitos periféricos a eles relacionados (ir “para fora”). Aqui, o processo inicia com a identificação de uma entidade particularmente importante no modelo e que, supõe-se, estará relacionada a muitas outras entidades. A partir daí, são procurados atributos, entidades relacionadas, generalizações e especializações da entidade em foco, e assim recursivamente até obter-se o modelo completo. A denominação da estratégia provém da idéia de que entidades importantes em um modelo e relacionadas a muitas outras são usualmente desenhadas no centro do diagrama, afim de evitar cruzamento de linhas. Um exemplo da aplicação desta estratégia é mostrado na Figura 3.20.

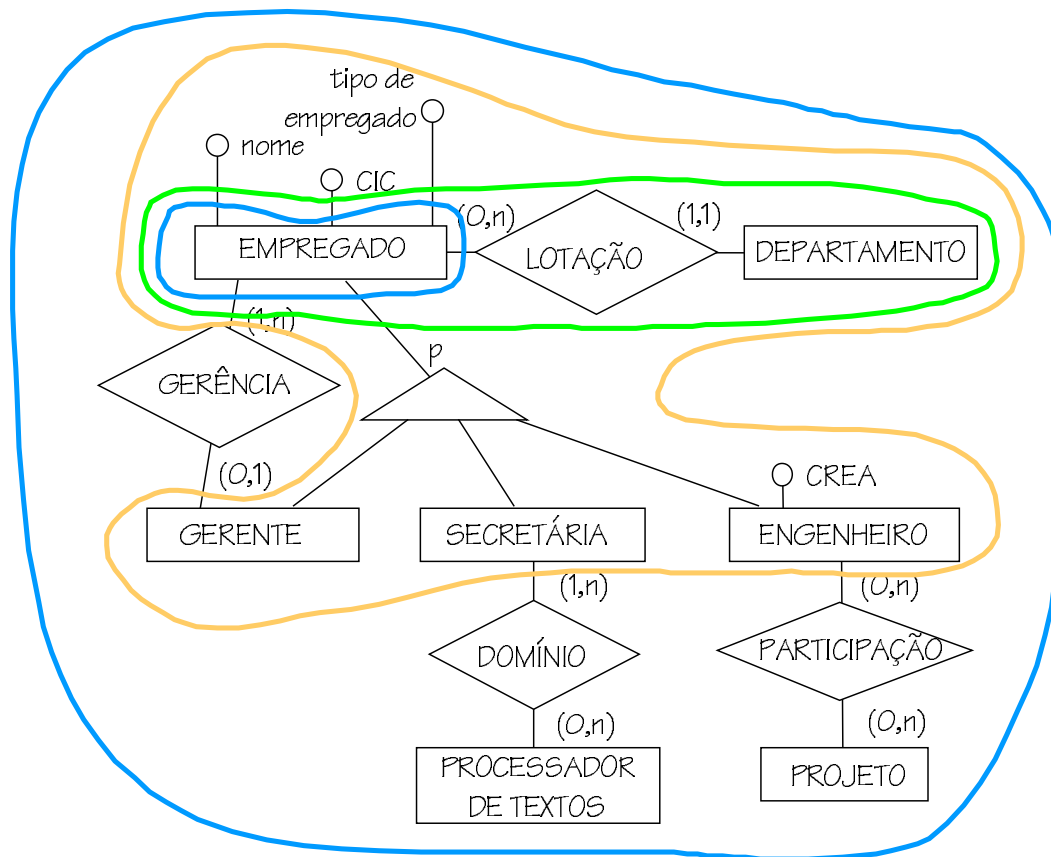


Figura 3.20: Aplicação da estratégia "de dentro para fora"

Na Figura 3.20, está mostrada a modelagem de uma parte de um sistema de recursos humanos. As linhas em tons de cinza indicam os passos da construção do modelo. No caso, considerou-se como conceito central e ponto de partida da modelagem a entidade EMPREGADO. Após, procurou-se entidades relacionadas a EMPREGADO, tendo sido identificada a entidade DEPARTAMENTO. No próximo passo, foram identificados os atributos e especializações de EMPREGADO e finalmente foram identificados os atributos e as entidades relacionados às especializações de EMPREGADO.

Os passos que ocorrem no processo "inside-out" são os mesmos que ocorrem no processo "top-down" ocorrendo, entretanto, uma iteração muito maior entre os passos 1.a, 1.b e 1.c.

## EXERCÍCIOS

**Exercício 3.1:** A Figura 3.21 apresenta um relacionamento que associa um produto de uma indústria com seus componentes (em inglês, "bill-of-materials"). Cada produto pode possuir diversos componentes e cada componente pode aparecer dentro de diversos produtos. Assim trata-se de um relacionamento do tipo n:n. Uma restrição que deve ser imposta sobre o banco de dados em questão é a de que um produto não pode aparecer na lista de seus componentes. Pergunta-se:

a) O modelo apresentado na figura contém esta restrição?



- b) Caso negativo, é possível alterar o modelo em questão para incluir esta restrição, se considerarmos que o nível de profundidade da hierarquia de composição de cada produto não excede três (tem-se apenas produtos prontos, produtos semi-acabados e matérias-primas)? Caso afirmativo, apresente a solução.
- c) É possível estender a solução do quesito anterior para uma hierarquia não limitada de níveis de composição?

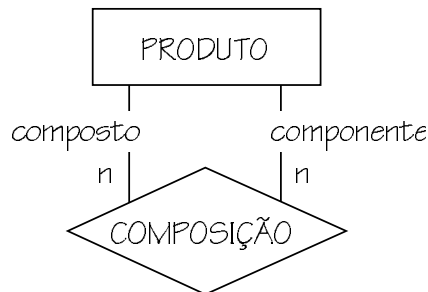


Figura 3.21: Composição de produtos

**Exercício 3.2:** Deseja-se modelar os clientes de uma organização. Cada cliente possui um identificador, um nome, um endereço e um país. Discuta os prós e contras das duas alternativas de modelagem de país:

- Como atributo da entidade cliente
- Como entidade relacionada a cliente.

**Exercício 3.3:** A Figura 3.22 apresenta uma entidade e respectivos atributos, muitos deles opcionais e um multi-valorado. Considere que há dois tipos de clientes, pessoas físicas e pessoas jurídicas. Pessoas físicas possuem código, CIC, nome, sexo (opcional), data de nascimento (opcional) e telefones (opcionais). Pessoas jurídicas possuem código, CGC, razão social e telefones (opcionais). Apresente um diagrama ER que modele mais precisamente esta realidade. Explique no que seu diagrama é mais preciso que o mostrado na figura 4.21.

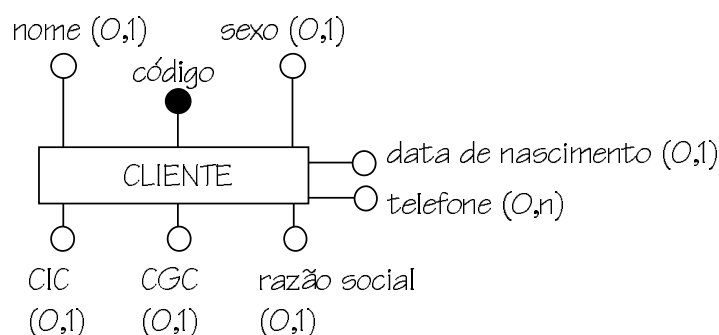


Figura 3.22: Cliente com atributos

**Exercício 3.4:** Transforme o modelo ER resultante do exercício anterior da notação Chen para a notação da Engenharia de Informações.

**Exercício 3.5: Estudo de caso - Administradora de imóveis**

Construa um diagrama ER (apenas entidades e relacionamentos com cardinalidades máximas) para a administradora de imóveis descrita abaixo.

A administradora trabalha tanto com administração de condomínios, quanto com a administração de aluguéis.

Uma entrevista com o gerente da administradora resultou nas seguintes informações:

- a) A administradora administra condomínios formados por unidades condominiais.
- b) Cada unidade condominial é de propriedade de uma ou mais pessoas. Uma pessoa pode possuir diversas unidades.
- c) Cada unidade pode estar alugada para no máximo uma pessoa. Uma pessoa pode alugar diversas unidades.

### **Exercício 3.6: Estudo de caso - Locadora de vídeos**

(adaptado do material de um curso de modelagem de dados da Oracle)

Uma pequena locadora de vídeos possui ao redor de 2.000 fitas de vídeo, cujo empréstimo deve ser controlado.

Cada fita possui um número. Para cada filme, é necessário saber seu título e sua categoria (comédia, drama, aventura, ...). Cada filme recebe um identificador próprio. Para cada fita é controlado que filme ela contém. Para cada filme há pelo menos uma fita, e cada fita contém somente um filme. Alguns poucos filmes necessitam duas fitas.

Os clientes podem desejar encontrar os filmes estrelados pelo seu ator predileto. Por isso, é necessário manter a informação dos atores que estrelam em cada filme. Nem todo filme possui estrelas. Para cada ator os clientes às vezes desejam saber o nome real, bem como a data de nascimento.

A locadora possui muitos clientes cadastrados. Somente clientes cadastrados podem alugar fitas. Para cada cliente é necessário saber seu prenome e seu sobrenome, seu telefone e seu endereço. Além disso, cada cliente recebe um número de associado.

Finalmente, desejamos saber que fitas cada cliente tem emprestadas. Um cliente pode ter várias fitas em um instante no tempo. Não são mantidos registros históricos de aluguéis.

### **Exercício 3.7: Estudo de caso - Sistema de reserva de passagens aéreas**

O objetivo do trabalho é projetar um sistema de reservas para uma companhia de aviação. O sistema contará com um banco de dados central, que será acessado por aplicações clientes, rodando tanto dentro da própria companhia, quanto fora dela.

A transação central do sistema é a reserva. Uma reserva é identificada por um código gerado pelo sistema em computador. A reserva é feita para um único passageiro, do qual se conhece apenas o nome. A reserva compreende um conjunto de trechos de vôos, que acontecerão em determinada data/hora. Para cada trecho, a reserva é feita em uma classe (econômica, executiva, etc.).

Um vôo é identificado por um código e possui uma origem e um destino. Por exemplo, o vôo 595 sai de Porto Alegre com destino a São Paulo. Um vôo é composto de vários trechos, correspondendo às escalas intermediárias

do voo. Por exemplo, o voo 595 é composto de dois trechos, um de Porto Alegre a Londrina, o outro de Londrina a São Paulo. Cabe salientar que há cidades que são servidas por vários aeroportos. Por isso, é importante informar ao passageiro que faz a reserva, qual é o aeroporto no qual o voo passa.

Às vezes os clientes, ao fazer a reserva querem saber qual é o tipo de aeronave que será utilizada em determinado trecho de voo. Alguns poucos voos, principalmente internacionais, têm troca de aeronave em determinadas escalas.

Nem todos voos operam em todos dias de semana. Inclusive, certos voos têm pequenas mudanças de horário em certos dias da semana.

Cada reserva possui um prazo de validade. Caso os bilhetes não tenham sido emitidos, até esgotar-se o prazo da reserva, a mesma é cancelada. Reservas podem ser prorrogadas.

Como o “check-in” de todos os voos está informatizado, a companhia possibilita a reserva de assento para o passageiro. Reservas de assento podem ser feitas com até três meses de antecedência

Além de efetivar reservas, o sistema deve servir para vários tipos de consultas que os clientes podem querer fazer:

- a) possibilidades de viagem de uma cidade ou de um aeroporto para outro
- b) o mesmo, mas restrito a determinados dias da semana
- c) horários de chegada ou de saída em determinados voos
- d) disponibilidade de vagas em um trecho de voo
- e) disponibilidade de determinados assentos em um trecho de voo.

### **Exercício 3.8: Estudo de caso - Sistema para locadora de veículos**

O objetivo deste estudo de caso é construir um modelo ER para o BD de uma empresa de locação de veículos. A empresa em questão aluga automóveis, camionetas de passageiros e camionetas de carga.

Ela atende a dois mercados, o das pessoas físicas e o das pessoas jurídicas. Para acelerar o atendimento, é importante conhecer os dados de clientes que já tenham usado a locadora no passado. Para cada pessoa física é necessário conhecer seu nome, sexo, data de nascimento, endereço e CIC. Já para as pessoas jurídicas é necessário conhecer seu nome, CGC, inscrição estadual e endereço. Os clientes são identificados por um código interno a locadora.

A empresa tem uma grande rede de filiais, espalhada pelo sul do país. Em um momento no tempo, um veículo encontra-se sob responsabilidade de uma filial. Entretanto, como veículos podem ser alugados para viagens em um sentido somente, eles podem mudar de filial. Um veículo é identificado pela sua placa. Além disso, é necessário conhecer o número do chassi, o número do motor, o tipo de veículo e a cor de cada veículo.

O sistema em computador deverá registrar:

- a) os veículos disponíveis em determinada filial na data corrente,
- b) as reservas para veículos em uma filial, com previsão de que veículos estarão disponíveis em uma data futura,
- c) os veículos presentemente alugados pela filial, o ponto de entrega (caso seja diferente do de locação) e data de entrega prevista.

Os veículos são classificados por uma tabela de tipos. Por exemplo, P3 corresponde a automóveis pequenos, de quatro portas e com ar-condicionado (Uno, Palio, etc.) e G4 a grandes automóveis de luxo (Omega ou similar). As reservas não são feitas para uma marca ou modelo de veículo, mas para um tipo de veículo.

Para tipos de automóveis, os clientes desejam saber o tamanho, classificado em pequeno, médio e grande, o número de passageiros, o número de portas, bem como se possui os seguintes acessórios: ar-condicionado, rádio, toca-fitas, CD, direção hidráulica e câmbio automático. Para tipos de camionetas de passageiros, as informações são as mesmas que para automóveis. Já para tipos de camionetas de carga, as informações acima não são relevantes. Neste caso, os clientes desejam saber a capacidade de carga da camioneta.

Para cada tipo de veículo, há um determinado número de horas necessário para limpeza e revisão de entrega, entre uma reserva e outra.

Além disso, o sistema deve programar as revisões dos veículos, impedindo que sejam reservados quando há revisões pendentes. Esta programação é feita com base em um conjunto de parâmetros que são a quilometragem atual do veículo, a quilometragem média diária de um veículo do tipo, bem como em uma tabela de revisões do tipo de veículo.

A seguradora que segura os veículos, exige que, para cada veículo alugado, seja mantida a identificação do motorista, o número de sua habilitação e data de vencimento da mesma. A habilitação não pode vencer dentro do prazo da locação.

### **Exercício 3.9: Estudo de caso - Sistema de preparação de congressos da IFIP**

(Esta é a adaptação de um conhecido estudo de caso proposto em um congresso de um grupo de trabalho da IFIP que tinha por objetivo comparar diferentes metodologias de desenvolvimento de sistemas de informação. O estudo de caso foi concebido como estudo de caso padrão em que cada metodologia a comparar deveria ser aplicada.)

Deseja-se construir um sistema em computador para apoiar a organização de congressos de trabalho da IFIP (Federação Internacional de Processamento de Informações). A IFIP é uma sociedade que congrega as sociedades nacionais de informática de diversos países.

Dentro da IFIP, atuam grupos de trabalho (GT). Cada GT trata de um determinado assunto. Cada GT recebe um código e é denominado segundo o assunto de que trata. Um dos objetivos de um GT é o de promover de forma regular congressos de trabalho.

Um congresso de trabalho é um encontro com número limitado de participantes (50-100) dos quais espera-se participação ativa. No congresso, são apresentados e discutidos trabalhos originais dos participantes. Para possibilitar que os participantes obtenham financiamento para sua ida ao congresso, é importante assegurar que a seleção dos trabalhos apresentados seja feita de maneira imparcial, por especialistas reconhecidos na área do congresso. O congresso é identificado por um código e possui um nome, um local e uma data de realização.

A IFIP recomenda que a preparação do congresso seja feita por dois comitês, o comitê organizador e o comitê de programa, montados especialmente para cada congresso.

O *comitê de programa* (CP) executa as funções listadas abaixo:

- ❑ O CP está encarregado da recepção de artigos submetidos, controlando se eles estão dentro do prazo. Um artigo é um trabalho que um ou mais autores pretendem apresentar no congresso e que será publicado em seus anais. Os anais são normalmente publicados por editoras comerciais. Por isso, os artigos devem ser originais, isto é, cada artigo somente pode ser submetido a um congresso. Os artigos são numerados sequencialmente dentro de um congresso, à medida que vão sendo recebidos. É necessário saber o título do artigo, bem como seus autores.
- ❑ Antes do início da avaliação dos artigos, o CP deve montar, um corpo de avaliadores externos, que irá ler cada artigo e emitir uma opinião sobre o artigo. Normalmente, mas não necessariamente avaliadores externos são pessoas que já participaram de congressos anteriores sobre o mesmo tema. Somente será cadastrado como avaliador de um congresso aquele que, após convidado para tal, o aceitar explicitamente.
- ❑ À medida que os artigos forem sendo recebidos, eles devem ser distribuídos aos avaliadores previamente cadastrados. Cada artigo deve ser distribuído a pelo menos três avaliadores. Um avaliador nunca deve estar com mais que um artigo em um instante no tempo.
- ❑ O CP está encarregado de fazer o julgamento dos artigos. No julgamento, o CP decide quais artigos serão aceitos e quais serão rejeitados, com base nos pareceres dos avaliadores. O julgamento acontece um certo tempo depois do fim do prazo de recepção de artigos.
- ❑ Após o julgamento, o CP deve montar o programa do congresso, isto é, agrupar os artigos aceitos em sessões e escolher um moderador para cada sessão. Uma sessão é um grupo de artigos que tratam de um assunto comum e que são apresentados em um determinado horário. Por tratar-se de um congresso de trabalho, não há sessões em paralelo, isto é, em um congresso, em um momento no tempo, não pode ocorrer mais que uma sessão. Assim, a sessão é identificada pelo congresso e pelo horário em que ocorre.
- ❑ Além das tarefas acima, o CP está encarregado de toda comunicação com os autores e com os avaliadores.

O *comitê organizador* (CO) está encarregado de toda parte financeira e organização local do congresso. Para fins do estudo de caso, vamos considerar apenas as seguintes atividades:

- ❑ Com boa antecedência, o CO deve emitir as chamadas de trabalho do congresso. A chamada de trabalhos é um texto que divulga o congresso e que convida as pessoas a submeter artigos ao congresso. Esta chamada é amplamente divulgada em listas eletrônicas, periódicos, etc. Além disso ela deve ser enviada pessoalmente, por correio eletrônico, às seguintes pessoas:
  - membros dos GT que promovem o congresso

- pessoas que de alguma forma já participaram de congressos promovidos pelos GT que promovem o congresso
- pessoas sugeridas para tal pelos membros dos GT que promovem o congresso

Neste envio, devem ser evitadas redundâncias.

- ❑ Após montado o programa pelo CP, o CO deve emitir os convites para participação no congresso. Por tratar-se de congressos de trabalho, apenas pessoas convidadas podem inscrever-se. São convidados:
  - os autores de trabalhos submetidos ao congresso,
  - os avaliadores externos do congresso,
  - os membros do CO e do CP do congresso, e
  - os membros dos GT que promovem o congresso.
- ❑ Finalmente, cabe ao CO executar as inscrições dos convidados que aceitarem o convite, controlando o prazo de inscrição e verificando se a pessoa é efetivamente convidada.

Observar que uma mesma pessoa pode atuar em muitas funções, mesmo em um mesmo congresso. A única restrição importante é que uma pessoa não pode avaliar o seu próprio artigo.

Como na preparação do congresso há considerável comunicação entre as diversas pessoas envolvidas, é necessário manter sempre atualizados os dados de acesso a pessoa, como seu endereço, seu telefone, seu número de fax e principalmente seu endereço de correio eletrônico.

O objetivo do exercício é construir um modelo ER de um banco de dados que será compartilhado via Internet entre os diversos comitês dos congressos em organização pela IFIP. Neste BD, estarão não só as informações sobre os congressos em organização, mas também sobre os do passado.

### **Exercício 3.10: Estudo de caso - Sistema de almoxarifado**

O enunciado deste trabalho foi adaptado de um livro de Peter Coad sobre projeto orientado a objetos.

O almoxarifado pertence a um grupo de empresas do ramo industrial e serve para estocar peças destinadas às várias empresas do grupo. Cada empresa do grupo é considerada um cliente do almoxarifado.

O almoxarifado está organizado em corredores. Cada corredor possui vários receptáculos para peças (um receptáculo é uma bacia retangular de material plástico). Os receptáculos são todos do mesmo tamanho. Os corredores são numerados e os receptáculos são numerados por corredor. Por exemplo, o receptáculo 2-10 é o décimo receptáculo do segundo corredor.

Em uma das extremidades do almoxarifado encontra-se o setor de recepção de peças. Lá chegam as peças entregues pelos fornecedores. Quando ocorre a chegada de peças, a primeira atividade é registrar na ordem de compra a chegada das peças. Uma cópia de toda ordem de compra é sempre enviada ao setor de recepção. Assim, neste setor sempre sabe-se quais as peças que estão por ser entregues. As ordens de compra são geradas no setor de compras e apenas repassadas ao almoxarifado.

Uma entrega corresponde sempre a uma ordem de compra. Entretanto, são admitidas entregas parciais, isto é, entregas que não completam a ordem de compra. Em uma entrega podem ser entregues diferentes quantidades de diferentes peças.

As peças recebidas são colocadas sobre um estrado. Este estrado é então levado para o almoxarifado por uma empilhadeira e as peças são distribuídas nos receptáculos. Um estrado pode conter diferentes peças. Para cada peça, procura-se um receptáculo que já contenha unidades da peça em questão e que ainda tenha espaço para a carga chegada. Caso não haja um receptáculo nestas condições, procura-se um receptáculo vazio.

A saída do almoxarifado se dá contra pedidos de clientes. Um pedido pode solicitar vários tipos de peças. Todas peças que atendem um pedido são juntadas, embaladas e colocadas em uma rampa de carga (numerada) onde encosta o caminhão do cliente. Não há pedidos pendentes, isto é, os clientes sempre pedem quantidades de peças que há em estoque.

O objetivo do sistema é o de aumentar o lucro do almoxarifado, ajudando sua equipe a guardar e recuperar itens mais rapidamente e a conhecer as quantidades estocadas.

O almoxarifado é de grande porte e constantemente há várias empilhadeiras circulando por ele tanto para estocar entregas quando para buscar peças referentes a um pedido.

Outros detalhes do sistema são fornecidos a seguir.

O almoxarifado somente atende empresas. É necessário manter um cadastro de clientes com CGC, nome, endereço e telefone de contato. Para cada peça é necessário conhecer seu UPC (“Universal Product Code”), descrição e número interno à organização.

Para cada entrega, o setor de recepção monta uma lista de distribuição, que instrui o operador sobre que peças, em quantidade ele deve estocar em que receptáculos.

Para cada pedido, o setor de saída monta uma lista de busca, que instrui o operador sobre que peças, em quantidade ele deve buscar em que receptáculos.

Em termos de processos, é necessário que o sistema processe o seguinte:

- ☐ Dê as ordens de distribuição de peças chegadas para cada chegada.
- ☐ Dê as ordens para busca para cada pedido.
- ☐ Mantenha a quantidade estocada de cada item e de cada receptáculo.
- ☐ Informe que peças em que quantidade devem ser estocadas ou buscadas em que receptáculos.

Em termos específicos de transações devem ser consideradas:

- ☐ Transações de chegada
  - Registro da chegada de produtos
  - Instruções para estocagem (em que estrado, em que receptáculos)
  - Confirmação da estocagem em um receptáculo
- ☐ Transações de saída de produtos
  - Registro de um pedido