

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 freqüentemente 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 as 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.

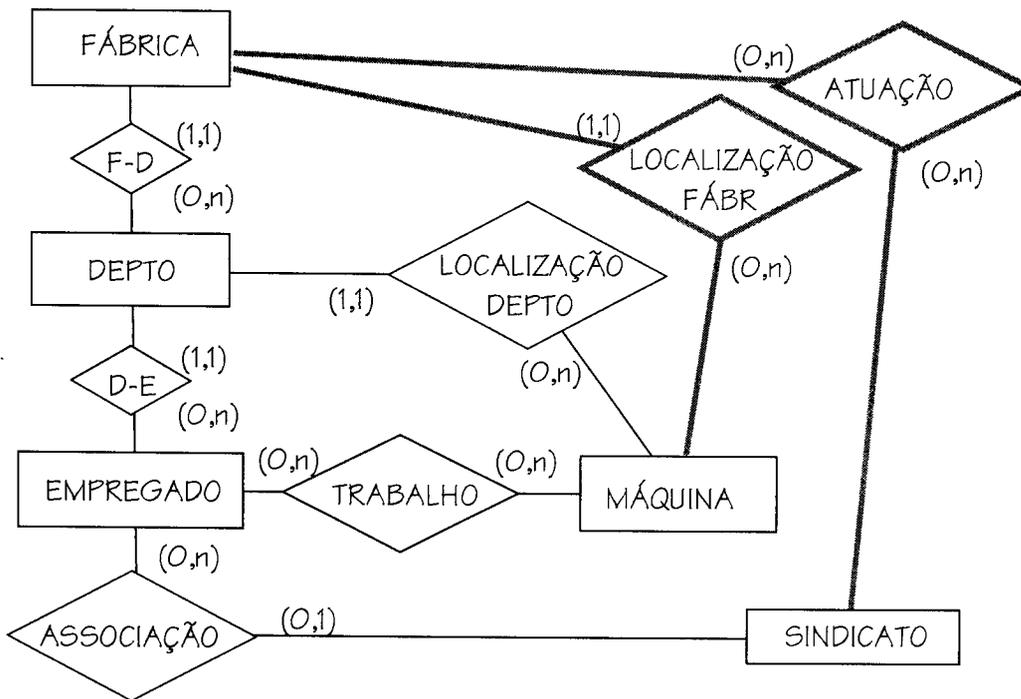


Figura 3.11: Relacionamentos redundantes

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.

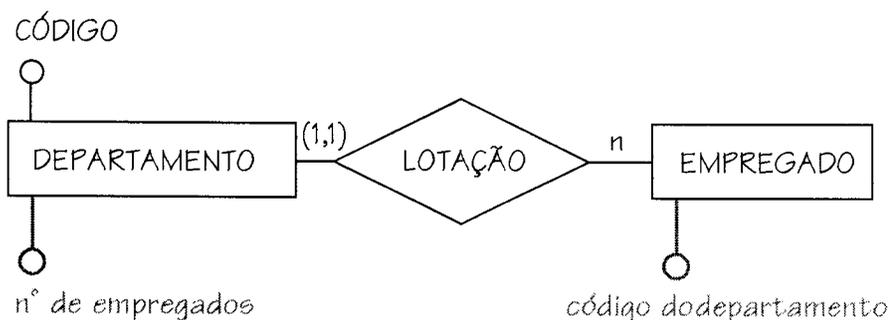


Figura 3.12: Atributos redundantes

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 LOTAÇÃO⁵. 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

⁵Para 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.

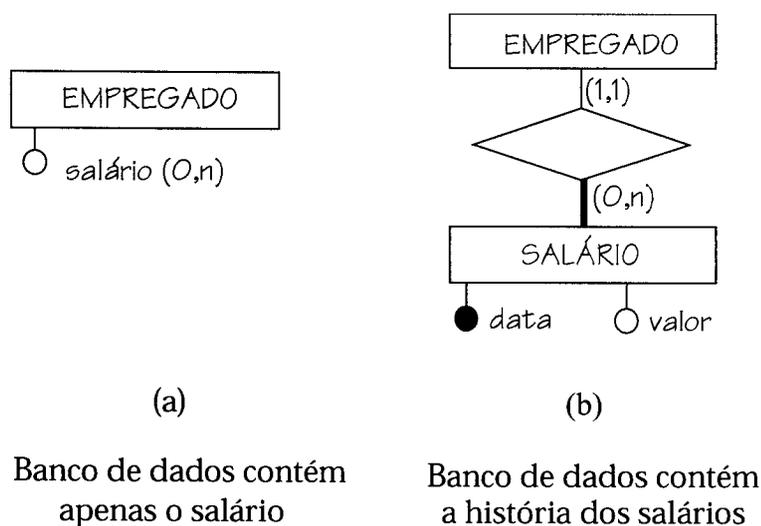


Figura 3.13: Modelando a dimensão temporal de atributos

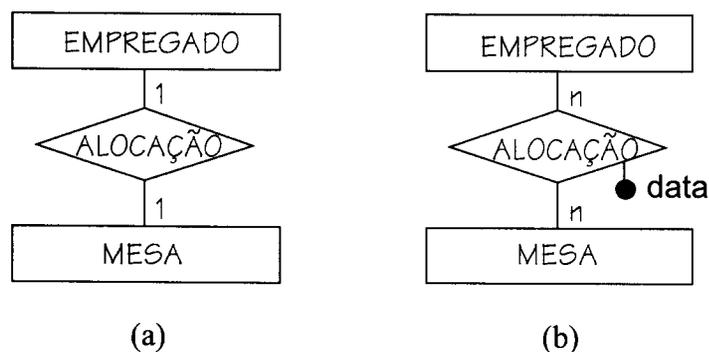
3.3.4.2 Relacionamentos que modificam ao longo do tempo

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.



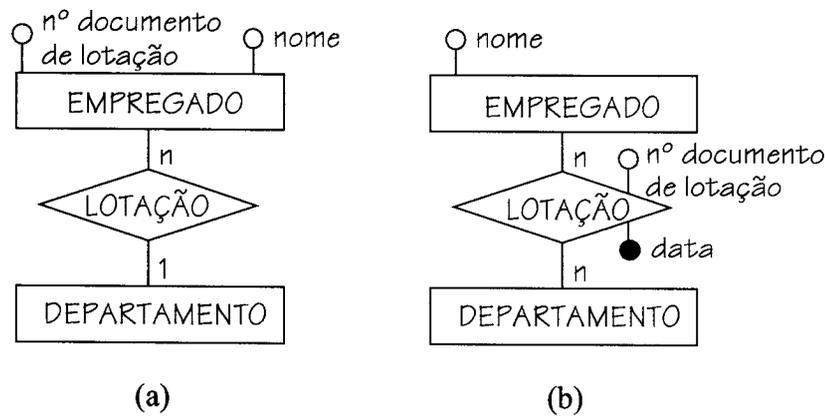
(a)
 Base de dados contém apenas a alocação atual

(b)
 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.

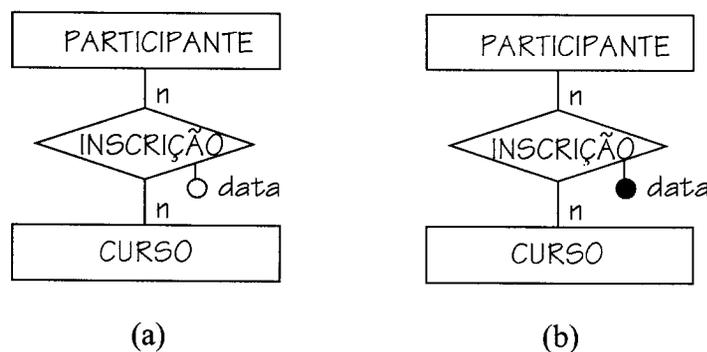
A Figura 3.16 apresenta mais um caso de consideração de história de relacionamentos. Neste caso, trata-se de um relacionamento de cardinalidade n:n. Modela-se a inscrição de participantes nos cursos oferecidos por uma empresa de treinamento. Na primeira versão, considera-se apenas os cursos nos quais 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

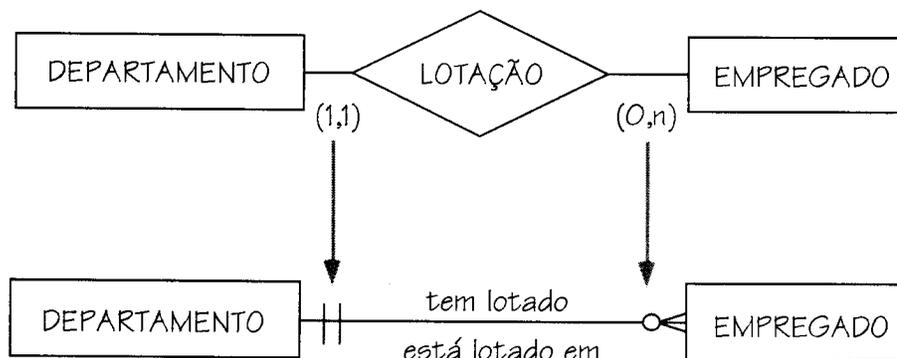
Quando de seu surgimento na literatura, a abordagem ER não era ainda suportada por ferramentas em computador de edição e manipulação, como as

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 freqüentemente 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 conseqüências:
 - 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)

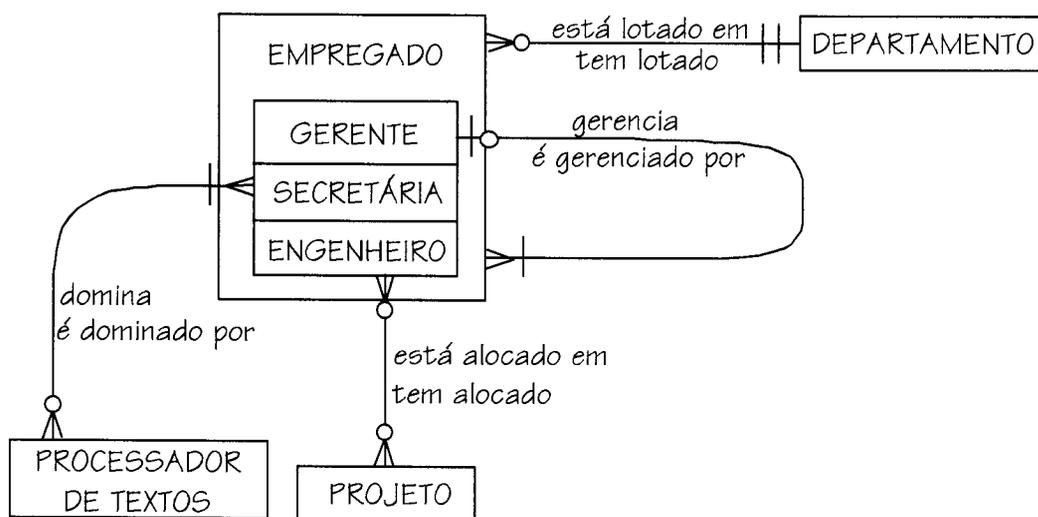


Figura 3.18: Modelo ER da Figura 2.37 na notação de Engenharia de Informações

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.

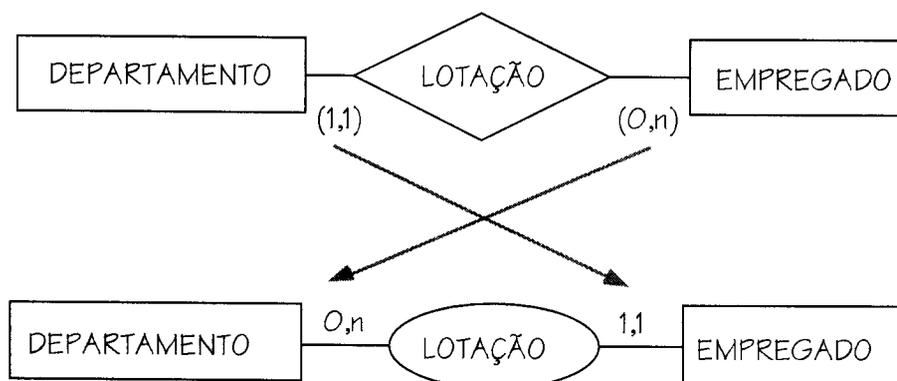


Figura 3.19: Notação Chen e correspondente notação MERISE

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 seqüência 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