

# Unidade 4

## MODELAGEM RELACIONAL DE DADOS

4.1	Processo de Desenvolvimento de Software	02
4.1.1	Software e Processo de Software	02
4.1.2	Ciclo de vida de um projeto	02
4.2	<a href="#">Abstração de Dados</a>	04
4.3	<a href="#">Modelagem Conceitual de Dados</a>	07
4.3.1	Modelo	07
4.3.2	Motivos para a construção de modelos	07
4.3.3	Modelagem de dados	07
4.3.4	Formulação do Obvio	10
4.3.5	Entidade	11
4.3.6	Atributos	13
4.3.7	Relacionamentos	16
4.3.8	Notação a ser adotada	36
4.3.9	Dicionário de Dados	37
4.4	<a href="#">Modelagem Lógica de Dados</a>	40
4.4.1	Introdução	40
4.4.2	Propriedade do modelo relacional normalizado	41
4.4.3	Mapeamento do MCD para o MLD	41
4.4.4	Exemplo de Modelo Conceitual – Entidades-Relacionamentos traduzido para o Modelo Lógico Relacional	50
4.5	<a href="#">Normalização</a>	51
4.5.1	Anomalias de Modificação	51
4.5.2	Formas Normais	52
	<a href="#">Bibliografia</a>	61

Nessa unidade você conhecerá os conceitos associados à Modelagem de Dados a partir dos conceitos de Entidades e Relacionamentos. Conhecerá também uma técnica para traduzir um Modelo ER em um Modelo Lógico adequado a implementação em Bancos de Dados Relacionais. E por fim, conhecerá o processo de Normalização de tabelas.

## UNIDADE 4 – MODELAGEM RELACIONAL DE DADOS

### 4.1 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Este capítulo tratará de uma série de conceitos relacionados ao processo de elaboração do modelo conceitual de dados a partir da utilização do Diagrama Entidade e Relacionamentos.

#### 4.1.1 Software e Processo de Software

Embora o uso de programas de computador já seja algo comum a muitos dos leitores desse material, faremos aqui, como forma de nivelar os conhecimentos que serão tratados ao longo desse capítulo, uma breve explanação do que seja um Software definindo inclusive o processo para elaboração do mesmo, ou seja, Processo de Software.

Segundo Pressman (2002), um Software compreende um **conjunto de programas** que executam em computadores, um **conjunto de dados**, que tanto podem ser números, textos, vídeos, figuras, áudio e um **conjunto de documentos** associados.

São desenvolvidos e não manufaturados como nos casos clássicos de outros produtos. Embora já existam muitas indústrias de software trabalhando com a perspectiva da montagem baseada em componentes, a maior parte continua a ser construída sob encomenda.

A elaboração de um Software compreende um conjunto de atividades que serão realizadas de forma a alcançar o produto ao final. Essas atividades são organizadas na forma de um roteiro **PAREI AQUI**

#### 4.1.2 Ciclo de vida de um Projeto

Refere-se ao Plano do Projeto ou Metodologia que a empresa adota para desenvolver seus sistemas.

Segundo Yourdon (1992), podemos enumerar alguns propósitos para o Ciclo de Vida:

- Definir as atividades a serem executadas em um Projeto de Desenvolvimento de Sistemas.
- Introduzir consistência entre muitos projetos de desenvolvimento de Sistemas na mesma organização.
- Para Introduzir pontos de verificação para o controle gerencial de decisões.

Ao longo da evolução do processo de software, diversos modelos de ciclo de vida surgiram. Cada modelo tenta colocar um pouco de ordem em uma atividade inerentemente caótica que o controle e coordenação de um projeto de software real (PRESSMAN, 2002, p.26).

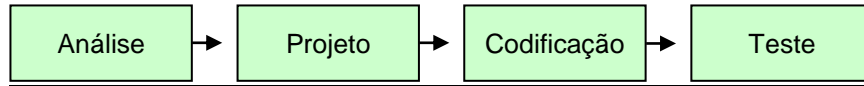
O Modelo Seqüencial Linear (figura 4.1a), também chamado de *ciclo de vida clássico* ou *modelo em cascata* é o modelo mais antigo e também o paradigma de software mais utilizado. Sugere uma abordagem seqüencial que começa no nível de sistema e progride através da análise, projeto, codificação, teste e manutenção.

O Modelo de Prototipagem (figura 4.1b) pode ser adotado quando o cliente define critérios gerais para o software, mas não define detalhadamente requisitos de entrada, processamento ou saída ou porque o desenvolvedor não está seguro quando a questões do ambiente. Dessa forma, o protótipo é utilizado como um mecanismo de levantamento de requisitos.

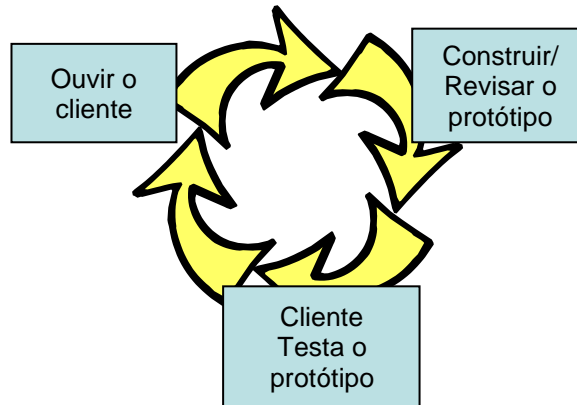
O Modelo RAD (Desenvolvimento Rápido de Aplicação) da Figura 4.1c é um processo incremental que enfatiza um ciclo de desenvolvimento curto, a partir do uso de componentes. Aplicações modularizáveis e que possam ser completada em menos de 3 meses são candidatas ao RAD.

Existem ainda os modelos de processo de software evolucionário que podem ser: Incremental, Espiral, Espiral ganha-ganha e de desenvolvimento concorrente. Temos ainda o Desenvolvimento Baseado em Componentes e o Modelo de Métodos Formais. Não criamos figuras para estes modelos, pois a intenção de apresentar os

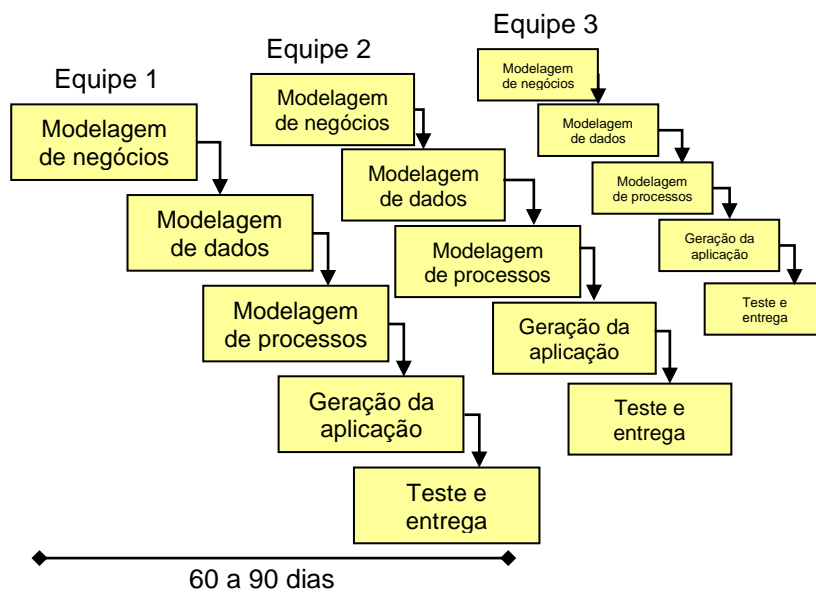
demais modelos foi somente a de conectar os elementos que serão estudados no contexto do desenvolvimento de um software. Todos esses modelos de processo podem ser amplamente estudados na bibliografia citada.



**FIGURA4.1a** Modelo Sequencial Linear (PRESSMAN, 2002).



**FIGURA4.1b** Modelo Prototipagem (PRESSMAN, 2002).



**FIGURA4.1c** Modelo RAD (PRESSMAN, 2002).

## 4.2 ABSTRAÇÃO

Abordaremos aqui os níveis de abstração necessários a produção de um modelo da realidade que possa ser processado e armazenado em um computador. Embora os níveis propostos aqui possam ser utilizados também para o processo de elaboração dos programas (parte funcional do sistema), nos focaremos somente na questão dos dados. A estrutura proposta suporta 5 níveis de abstração: Nível do Mundo Real, Nível Descritivo, Nível Conceitual, Nível Computacional e Nível Interno, que estão detalhados abaixo: (Setzer, Silva, 2005)

### Nível do Mundo Real

O nível do Mundo Real ainda é muito nebuloso, pois representa a vastidão da nossa realidade, mas há os que acreditam que um dia será possível formalizá-lo. Essa realidade é povoada de seres, fatos, coisas, organismos sociais. Para se referenciar a esse elenco de elementos, utiliza-se a denominação **entes**. Além de reconhecer e identificar os entes, é importante que se conheça as *associações* que existem entre os entes no mundo real. Observe que na descrição do mundo real listamos um elenco muito variado de elementos, tanto tangíveis (seres) quanto intangíveis (fatos), ou seja, ao olharmos para o mundo real percebemos tanto coisas concretas como abstratas. Cabe então ao projetista determinar o que é interessante do mundo real para fazer parte do projeto de sistema que está sendo elaborado.

### Nível Descritivo

O segundo nível de abstração é o das descrições informais feitas em uma linguagem natural, tanto escrita quanto falada. O artefato gerado nesse nível é denominado *modelo descritivo*. A elaboração desse modelo se dará pela descrição da realidade por meio de frases, figuras, fotos, preferivelmente sem o uso de conceitos matemáticos formais de forma que as partes (pessoas) envolvidas no processo de elaboração entendam o que será modelado sem a necessidade de conhecimento adicional. As descrições feitas no modelo descritivo devem ser suficientes para se converterem em informação para os que delas fizerem uso. Não há regras para elaboração do Modelo Descritivo, pois tanto o mundo real quanto o modelo descritivo não são formais. No entanto pode-se adotar um conjunto de diretrizes denominadas *Análise de Requisitos* para a elaboração do modelo descritivo. O produto da análise de requisitos será uma descrição formal das **regras de negócio** e as **transações** pertinentes ao ambiente observado. Semelhante ao que ocorre no mundo real, adota-se o termo *ente* para se referir aos elementos que estão sendo observados e modelados.

### Nível Conceitual

O terceiro nível de abstração diferente do segundo já adquire um formalismo matemático para expressar seus modelos. O documento elaborado no terceiro nível é denominado Modelo Conceitual, que já adotam uma conceituação rigorosa, mas que eventualmente não podem ser introduzidos em um computador. Nesse nível, aparecem dois aspectos distintos, em geral misturados nos modelos descritivos: as *estruturas de dados* (que darão forma ao Modelo Conceitual de Dados) e o *tratamento dos dados* (que dão forma ao Modelo Conceitual de Tratamento de dados ou simplesmente Modelo Funcional). No Modelo Conceitual os dados são provenientes dos entes do mundo real de mesma categoria, como constituindo conjuntos abstratos como os da matemática. No nível conceitual estamos atuando no tratamento dos dados que compreende ações de definição, exclusão, atualização e seleção nos metadados, que possuem as informações sobre os dados. Usa-se uma linguagem gráfica para definir as estruturas no nível conceitual apoiada por uma linguagem textual que complementa a linguagem gráfica para não torná-la muito extensa.

Se a abordagem utilizada for a Orientação a Objetos "OO", nesse nível se faz a identificação das "classes de dados". Uma classe envolve as *estruturas* e o *comportamento* (métodos) dos dados. Esse encapsulamento denomina-se classe, pois agrupa em um único elemento a definição das estruturas de dados e das rotinas (métodos) de tratamento dos mesmos. As classes identificadas no contexto da aplicação poderão ser agrupadas em um diagrama denominado Diagrama de Classes.

Na abordagem anterior a OO, o estudo desses dois componentes já era realizado, porém de maneira separada. Nessa época estudavam-se os "Tipos de Dados" e as "rotinas ou procedimentos" que processavam dados de certos tipos. Na abordagem Estruturada, os dados são tratados como entidades e uma das maneiras de expressá-los é a partir de um diagrama denominado Diagrama Entidades-Relacionamentos.

Ainda que se adote alguma notação gráfica e até mesmo alguma ferramenta de software para apoiar a elaboração do Modelo Conceitual, não deve ser preocupação desse nível dos elementos relacionados a parte de implementação, ou seja, com requisitos de hardware e software que serão utilizados para materializar a solução. A mesma regra é utilizada para o Modelo Descritivo. Isso se dá pelo fato de que modelos conceituais não mudam sempre que houver uma mudança de SGBD. Modelos conceituais constituem uma documentação de alto nível sobre as estruturas de dados e os programas que poderão ser implementados.

### Nível Operacional

O quarto nível de abstração é o dos *dados computacionais*. O nível computacional trata da visão externa que a comunidade de usuários possui dos dados e também da visão que estes usuários têm do processamento que é feito pela máquina sobre esses dados. Nesse nível os dados que ainda possuíam uma concepção abstrata proveniente do nível conceitual, já podem ser vistos como um elemento concreto e passível de tratamento por um computador. Os dados no nível conceitual podem ser descritos por qualquer formalismo matemático, podem existir no papel ou na mente. No nível computacional os dados devem ser expressos de forma que um computador possa recebê-los e tratá-los. Se no nível conceitual as informações devem ser expressas o mais próximo possível do mundo real, no nível computacional tais informações devem ser expressas de forma algorítmica, sendo tal algoritmo deve ser escrito em alguma linguagem de programação, de maneira a que seja introduzida para ser executada em um computador.

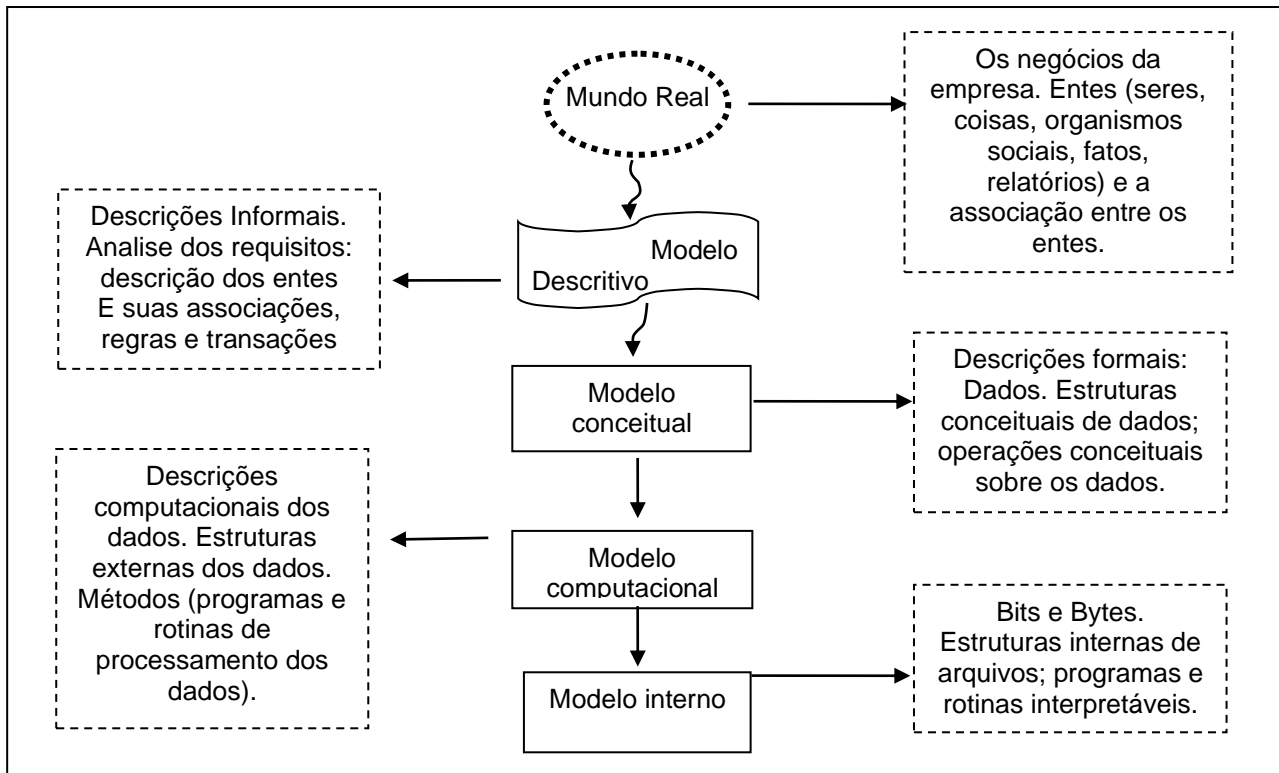
É nesse nível que se faz a adequação dos modelos elaborados nos níveis anteriores para um Sistema de Gerenciamento de Bancos de Dados (SGBD).

Nesse nível temos a correspondência com os dois aspectos do nível conceitual, porém a um nível computacional: o das estruturas de dados, que será tratado por uma linguagem de definição das estruturas de dados (DDL) e o da especificação de tratamentos dos dados, que será tratado por uma linguagem de processamento dos dados (DML). Os tratamentos que os dados sofrerão podem ser: Inclusão, Atualização, Exclusão e de Seleção.

Como o nível computacional é o nível de máquina, já é possível tratar diversos elementos relacionados a eficiência, sendo que para isso os desenvolvedores devam conhecer a estrutura interna dos arquivos gerados pelo SGBD. No entanto, nos dias atuais alguns elementos de desempenho são absorvidos pelos produtos de GBD que a partir de componentes internos escolhe a maneira mais eficiente para o processamento dos dados.

### Nível Interno

O quinto e último nível é o nível de máquina, não mais do ponto de vista do usuário, mas sim dos aspectos internos, ou seja, da representação interna dos dados e dos programas. No nível interno não falamos mais de dados, mas sim de cadeia de bits ou de bytes. É comum chamar o **nível interno** de **nível físico**. Não é muito adequado, pois o nível computacional já é físico, ou seja, já é um nível voltado para a máquina. De forma semelhante, é comum chamar o nível computacional de **nível lógico** o que não parece também ser uma boa alternativa por entendermos que nível lógico estaria mais adequado ao nível conceitual. Enfim, para evitarmos confusões, melhor mantermos os nomes convencionados.



**FIGURA4.2** Níveis de Abstração (SETZER, 2005, p.8)

Em cada nível de abstração, encontramos diversos usuários atuando, desempenhando papéis distintos e complementares:

**Mundo Real** → Este é o nível onde encontramos somente o **usuário final**, desempenhando as atividades relacionadas a sua área de atuação.

**Nível Descritivo** → Aqui participam o **usuário final** e os **analistas de sistemas**. Os usuários finais atuam fornecendo o conhecimento do objeto de estudo para os analistas de sistemas que os registrará de maneira livre de restrições de implementação, de forma que seja compreensível a todos os envolvidos no processo. Eventualmente, dependendo da estrutura organizacional podemos encontrar também **DBAs** atuando nessa fase e em algumas situações, os próprios **programadores**.

**Nível Conceitual** → Aqui participam o **usuário final** e os **analistas de sistemas**. Nessa fase, o usuário atua junto com os analistas na identificação dos elementos relevantes de negócio que mereçam se transformar em futuras estruturas de dados. Em algumas organizações podemos ter a presença dos DBA já nessa fase da modelagem.

**Nível Operacional** → Aqui participam os **analistas de sistemas** e **DBAs**. Nessa fase, os analistas em conjunto com os DBAs irão aplicar regras específicas do produto utilizado para Gerenciar o Banco de Dados de maneira a adequar o Modelo Conceitual elaborado no nível anterior às exigências desse produto específico e aos próprios requisitos relativos ao armazenamento de dados definidos pela organização.

**Nível Interno** → Aqui participam somente os **DBAs**.

Durante as disciplinas de Banco de Dados estaremos abordando elementos teóricos e práticos que representam os níveis Conceitual e Computacional. O Nível Conceitual é tratado na seção 4.3 enquanto o Nível Computacional é tratado na seção 4.4.

### 4.3 MODELAGEM CONCEITUAL DE DADOS

Segundo Cougo (1997), define-se como Modelo Conceitual aquele onde os objetos, suas características e relacionamentos têm a representação fiel ao ambiente observado independentemente de restrições impostas pela tecnologia. Durante a elaboração do mesmo devem ser ignoradas quaisquer particularidades necessárias a implementação. Dessa forma, será possível elaborar modelos que praticamente são imunes ao ambiente que será utilizado para implementá-lo. No entanto, a partir do MCD, será possível derivar para qualquer tipo de estrutura de implementação, como por exemplo, usar um SGBD Hierárquico ou um SGBD Relacional.

A elaboração do Modelo Conceitual dá-se na fase de Análise de um projeto de software, conforme pode ser observado na figura 4.1.

#### 4.3.1 Modelo

Um modelo é a representação abstrata e simplificada de um sistema real, com a qual se pode explicar ou testar o seu comportamento, em seu todo ou em parte. Observe que nessa definição, não estamos condicionando que o modelo seja computadorizado. Qualquer ambiente do mundo real é passível de ser modelado, já que o mesmo busca expressar o objeto observado algumas vezes, mesmo antes da existência física, tangível de tal objeto.

Ex: As plantas de apartamentos dos jornais.

- O manequim da vitrine
- A foto de um conjunto estofado
- Um aeromodelo sendo usado para testes de aerodinâmica
- O desenho em perspectiva de uma nova cozinha
- O molde de uma roupa obtido em uma revista.

O importante, contudo, é perceber que através de algum meio, seja uma maquete, desenho ou descrição, pode-se antecipar ou substituir a existência de uma realidade qualquer.

#### 4.3.2 Motivos Para A Construção De Modelos

Segundo Cougo (1999), os principais motivos para elaborarmos modelos são:

- a) Focalizar características importantes de sistemas deixando de lado as menos importantes.
- b) Discutir alterações e correções nos requisitos do usuário a baixo custo e com mínimo risco.
- c) Para confirmar que o ambiente do usuário foi entendido.
- d) Representar o ambiente observado
- e) Servir de instrumento de comunicação
- f) Favorecer o processo de verificação e validação
- g) Capturar aspectos de relacionamento entre os objetos observados
- h) Servir de referencial para a geração de estruturas de dados

#### 4.3.3 Modelagem De Dados

Tem por objetivo a obtenção de estruturas de dados que nos levem a projetar Banco de Dados.

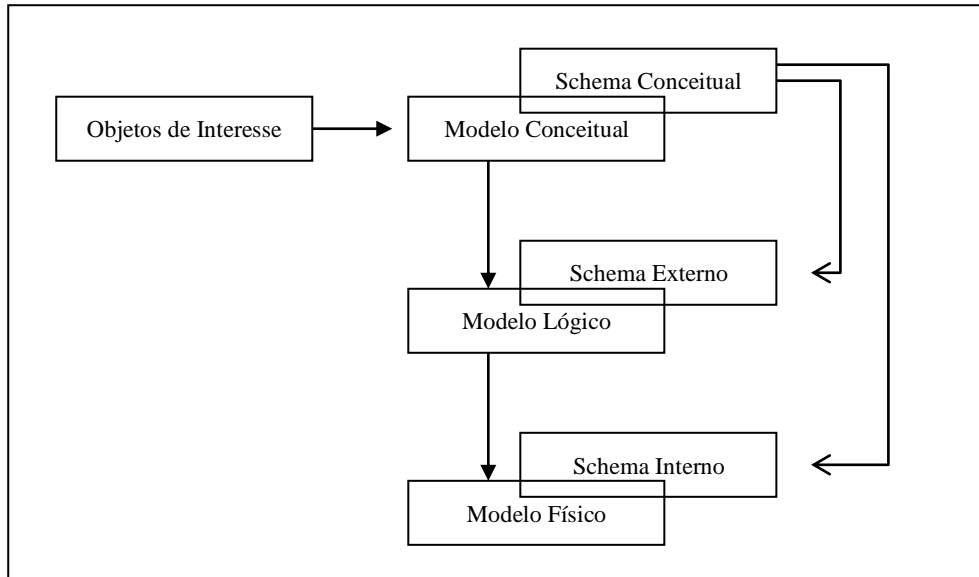
Definem-se basicamente, três níveis de modelagem:

- MCD (Modelo Conceitual de Dados): Objetos, características e relacionamentos, têm a representação fiel do ambiente, independente das limitações impostas da Tecnologia. Equivale ao Nível Conceitual dos Níveis de Abstração apresentados anteriormente.
- MLD (Modelo Lógico de Dados): Objetos, características e relacionamentos, tem a representação de acordo com as regras de implementação e limitantes de alguma tecnologia. Deve observar conceitos de chave de



acesso, controle de chave duplicada, item de repetição, normalização, integridade referencial. Equivale ao Nível Computacional dos Níveis de Abstração apresentados anteriormente.

- MFD (Modelo Físico de Dados): A representação do objeto é feita sob o foco do nível físico de implementação. Equivale ao Nível Interno dos Níveis de Abstração apresentados anteriormente.



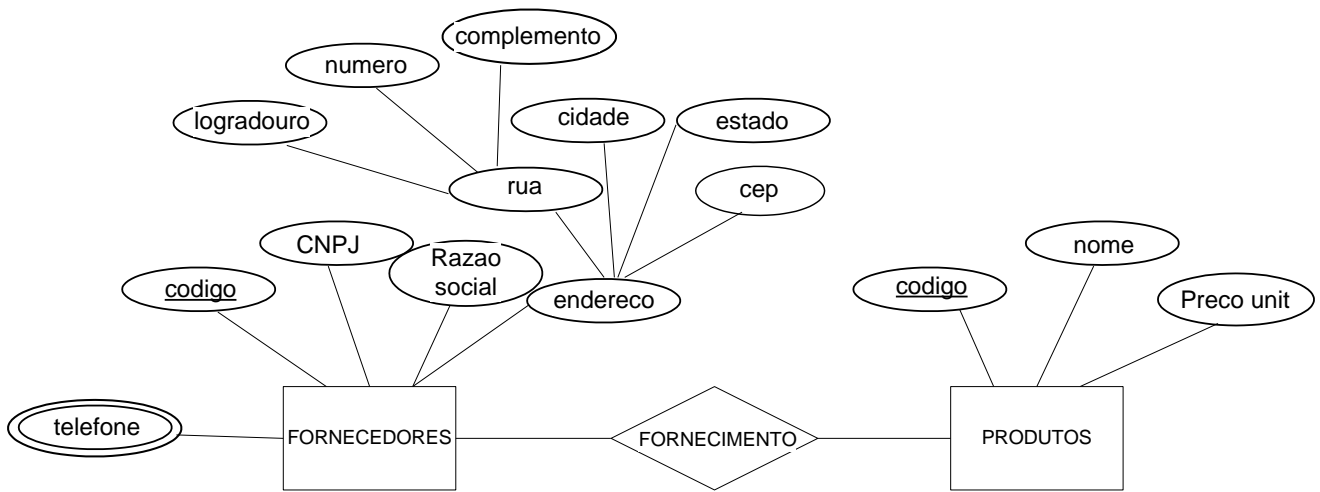
**FIGURA4.3** Relação entre os modelos e esquemas de um banco de dados

### Por que “Modelar Dados”?

- Porque Dados são mais estáveis que os procedimentos.
- Porque Dados existem e podem ser descritos independentemente de como são usados
- Porque os Dados possuem propriedades próprias e bem definidas.

Um importante documento que trata de modelagem de dados foi proposto por Peter P. Chen, em Março de 1976. Intitulado “*The Entity-Relationship Model: Toward the unified view of data*”, tornou-se uma referência para o processo de elaboração de modelos de dados. Com uma abordagem e simbologia simples, o Modelo Entidade-Relacionamento (ER) possui flexibilidade e adaptabilidade. Utiliza-se, basicamente de 3 símbolos para representar as Entidades (Retângulos), os Atributos (Balões) e Relacionamentos (Losangos). A figura 4.4 abaixo demonstra o exemplo de um modelo ER com a simbologia proposta originalmente por Chen. Atualmente, como forma de não poluir o diagrama, optou-se por não demonstrar os atributos com os círculos conectados nas entidades. Os atributos são definidos a parte no Dicionário de Dados, deixando assim o modelo de dados mais limpo, conforme podemos observar na figura 4.5. A figura 4.6 apresenta outra notação para atributos.





Fonte: Silberschatz, 2006, p.143-148

**FIGURA4.4** Notação para Diagrama Entidade-Relacionamento proposto originalmente por Chen.



#### DICIONÁRIO DE DADOS

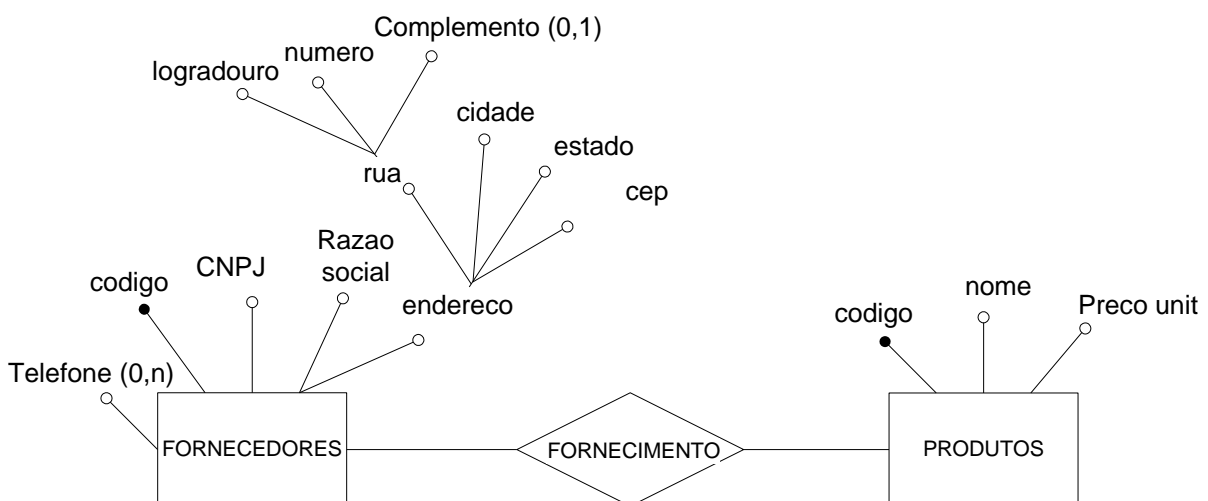
FORNECEDORES = código + cnpj + razão social + ENDERECO + {telefone}

PRODUTOS = código + nome + preco unit

ENDERECO= RUA + cidade + estado + cep

RUA = logradouro + numero + (complemento)

**FIGURA4.5** Notação alternativa para Diagrama Entidade-Relacionamento



**FIGURA4.6** Outra notação para diagrama Entidade-Relacionamento

O modelo proposto por Chen assemelha-se ao que se pode chamar de “Lei do Mundo” (COUGO, 1997).

#### 4.3.4 Formulação do óbvio

**1. Lei do Mundo:** “O mundo está cheio de coisas que possuem características próprias e que se relacionam entre si” (COUGO, 1997).

Analisando a Lei do Mundo, pode-se observar:

- “O mundo está cheio de coisas...”: Dentro de um universo observado, será possível reconhecer objetos (coisas).

Ex.: Num ambiente de uma fábrica, encontramos:

- Máquinas de produção de peças.
- Funcionários operadores dessas máquinas
- Conjunto de Ferramentas utilizadas para operar as máquinas
- Procedimentos de Operação a serem executados.

- “... que possuem características próprias...”: Os objetos identificados no universo terão características que permitirão agrupá-los em conjuntos particulares. Estas características são inerentes a todos os objetos de um mesmo conjunto e assim deverão ser representadas.

Ex.: João, Pedro, José, Maria e Ana são objetos ou elementos individualizados pertencentes ao conjunto denominado FUNCIONÁRIOS.

Ex.: Alicates, Chave de Fenda, Chave Inglesa, Torquês são elementos individualizados pertencentes ao conjunto denominado FERRAMENTAS.

- “... e que se relacionam entre si.”: Tanto objetos como conjunto de objetos se relacionarão em função das suas próprias características.

Ex.: João opera Máquina de Produção X23

João substitui Pedro.

Carlos ajusta Máquina de Produção H124 usando Torquês.

A partir da análise da Lei do Mundo, pode-se partir para a conceituação dos elementos componentes da proposta de Chen – Entidades-Relacionamentos.

#### 4.3.5 Entidade

Para Setzer (2005, p.9) o mundo real é povoado de “**objetos**” dos tipos mais variados, como por exemplo, os seres, os fatos, as coisas e os organismos sociais. Por envolver elementos tão distintos adota a denominação “**entes**” para designá-los, por julgar que para seres e organismos sociais o termo objeto não é muito adequado. Um indivíduo do conjunto representa um objeto ou uma entidade. É algo que possui características que a torna distinguível, que desempenha um papel específico no sistema e sobre o qual desejamos guardar informações, define Pompilho (2002, p.78-79).

A percepção e a identificação de uma entidade dependem diretamente do ambiente analisado e da relevância dela no contexto. Uma pessoa poderá ter papéis distintos em ambientes distintos. Poderá ser um aluno em um sistema acadêmico, um cliente em um sistema comercial e um paciente em um sistema de consultas médica.

Entidades são abstrações de coisas do mundo real de forma que todas as coisas tenham as mesmas **características**, e que estejam sujeitas e em conformidade com as mesmas **normas**. (SHLAER, MELLOR (1990, p.16).

Uma entidade pode ser:

Ex.: Um Ser, um fato, uma coisa, um organismo social.

**4.3.5.1 - Conjunto de Entidades:** O conjunto de cada um desses elementos (seres, fatos, etc...) é denominado **Classe de Entidade** ou **Conjunto de Entidades**. São identificadas a partir dos **Substantivos** e perduram no tempo, podendo cada elemento do conjunto ocorrer uma única vez. A representação gráfica de uma entidade no Modelo ER é feita através de um retângulo, contendo um nome dentro do retângulo que nomeia a entidade. Este nome deverá ser escrito em letras maiúsculas e no plural (SETZER, 2005).

Ex.:

Identificar entidades e enquadrá-las em grupos (conjuntos) é uma tarefa que requer tanto conhecimentos técnicos quanto prática. Para auxiliar nesse processo, Cougo propõe a estratégia descrita abaixo, como meio de apoiar o processo de aprendizado, ou mesmo para validação e verificação do processo de modelagem.

**4.3.5.2 – Estratégia para reconhecimento de Entidades:** Esta estratégia propõe separar os conjuntos de entidades em 5 grandes grupos: Coisas tangíveis, Funções, Eventos ou Ocorrências, Interações e Especificações (COUGO, 1997):

- **As Coisas Tangíveis:** Refere-se a tudo aquilo que pode ser tocado. Refere-se a todos os elementos que tenham existência concreta, que sejam manipuláveis, que possam ser tocados e que ocupem lugar no espaço.

Ex.: “Um avião a jato, um Cavalo puro-sangue, um livro de ficção, uma garrafa térmica, uma chave da porta de entrada”.

“O agrupamento desses objetos em conjuntos (entidades ou classes) dependerá do ponto de vista adotado pelo processo de modelagem. O nível de abstração a ser adotado para o agrupamento poderá ser de maior ou menor grau, o que nos levará a formação de diferentes conjuntos” (COUGO, 1997, p. 38).

A figura 4.7 abaixo, expressa um exemplo de agrupamento.

Conjunto (Entidade)	Objetos que pertencem ao conjunto
MEIOS DE TRANSPORTE	O avião, o automóvel
ANIMAIS	O cavalo, o elefante, o cachorro
UTENSÍLIOS DOMÉSTICOS	A garrafa, a mesa, o telefone, o vidro
UTENSÍLIOS ESCOLARES	O livro, o lápis, o quadro, o disquete
EQUIPAMENTOS	O computador, a máquina de escrever
PERTENCES PESSOAIS	A chave, a mala, a camisa, a carteira

**FIGURA4.7** Modelo de agrupamento de elementos para coisas tangíveis (Adaptado de COUGO, 1997, p.38).

- **As Funções exercidas por elementos:** Algumas vezes, somente se percebe os objetos a partir das funções por eles realizadas. Entendendo por Função, todo o tipo de papel, atribuição, classificação, capacitação.

Ex.: O médico cirurgião, um engenheiro naval, um departamento de compras, seu professor de inglês, o autor de um livro, a gerência de suporte técnico, a recepcionista de um hotel, um médico pediatra, a seção de despacho de materiais.

Conjunto (Entidade)	Objetos que pertencem ao conjunto	Coisas Tangíveis mapeadas como FUNÇÃO
ESPECIALISTAS	O Médico Cirurgião, o Engenheiro Naval, o Autor de um Livro	PESSOA
ORGANIZAÇÕES	O Departamento de Compras, A Gerência de Suporte Técnico	ORGÃO FUNCIONAL
ATENDENTES	O professor de Inglês, O gerente do hotel, A recepcionista do hotel	PESSOA
CLIENTE	Os alunos, O paciente	PESSOA

**FIGURA4.8** Modelo de agrupamento de elementos para funções (COUGO, 1997, p.40)

Pode-se observar ainda, que existem especializações dentro dos grandes grupos que foram definidos acima. Por exemplo, poderíamos distinguir os Especialistas em MÉDICOS, ENGENHEIROS e AUTORES enquadrando cada um no seu devido contexto. Isso está diretamente relacionado ao nível de abstração que se deseja modelar. Níveis mais elevados podem agrupar elementos com algumas características semelhantes, porém, se desejarmos níveis baixos de abstração, o ideal seria uma modelagem onde se diferencie os diversos papéis, conforme proposto no início do parágrafo.

- **Eventos ou Ocorrências:** Alguns objetos só conseguem ser percebidos, enquanto certa ação se desenrola, ou quando algum fato acontece. Nesses casos conseguimos definir características que o tornam materializáveis. Em geral, possuem poucos dados próprios, pois a maioria dos dados a ele pertencentes são na verdade dados de outros objetos participantes do próprio evento (COUGO, 1997, p.41).

Ex.: Um vôo comercial, um acidente de trânsito, uma apresentação técnica de um fornecedor, uma festa beneficente, uma gincana esportiva.

Conjunto (Entidade)	Objetos que pertencem ao conjunto	Demais objetos que participam da realização do evento
VOO	TAM3195, GOL1404, VRG2133	COMISSÁRIOS, PILOTOS, AERONAVES
ACIDENTE	Batida na av. Fernando Ferrari, atropelamento	VEICULOS, PESSOAS
APRESENTACAO TÉCNICA	Palestra dia 20/12/2009	PALESTRANTE, LOCAL, OUVINTES
FESTA BENEFICENTE	Jantar dançante dos idosos, churrasco da sociedade contra o câncer.	PESSOAS, LOCAL,
GINCANA ESPORTIVA	Gincana de vôlei de praia, gincana de ginástica rítmica, gincana dos alunos de medicina	EQUIPES, ATIVIDADES, PREMIOIS, JUIZES, LOCAL

**FIGURA4.9** Modelo de agrupamento de elementos para eventos ou ocorrências

- **Interações:** Nascem como resultado da associação de objetos em função da execução de um processo. Os objetos-interação passam a existir de maneira individualizada, além dos objetos que participam dessa interação. Em alguns momentos, estes objetos podem ser confundidos, e até mesmo substituídos por Associações (Relacionamentos) entre objetos.

Ex.: A compra de um imóvel, uma adoção de uma criança por um casal, uma venda realizada por um fornecedor.

Objeto-interação	Objetos que participam da	Substituição possível
------------------	---------------------------	-----------------------

	<b>interação</b>	
A compra de um imóvel	Comprador, imóvel, proprietário, corretor, agente financeiro	relacionamento "é comprado por" evento "aquisição" coisa tangível "contrato de compra"
Uma adoção de uma criança	Família cedente, família receptora, criança, órgão regulamentador	Relacionamento "é adotada por" Evento "adoção" Coisa tangível "registro de adoção"
Uma venda realizada por um fornecedor	Fornecedor, produto, cliente	Relacionamento "é vendido por" Evento "venda" Coisa tangível "nota de venda"

**FIGURA4.10** Modelo de agrupamento de elementos para funções (COUGO, 1997, p.43)

- **Especificações:** Refere-se aos objetos que definem características de outros objetos. Isoladamente **não** representam objetos, mas sim especificações que quando aplicadas ou seguidas, darão origem aos objetos. É válida sua supressão no MCD. Serão obtidos no MDL a partir do processo de Normalização.

Certamente que a percepção dos elementos que comporão um Modelo Entidades-Relacionamentos e em especial a descoberta das entidades não é uma atividade trivial. Por isso, algumas dicas podem ser bem boas na hora que você se depara com uma situação real (SQL MAGAZINE, 2004, p.23-24).

**1) Identificar substantivos que designem objetos:** A partir da leitura dos requisitos marque e faça uma lista de todos os substantivos que designem elementos do mundo real como pessoas, coisas, documentos, controles, sistemas. Caso o substantivo apareça mais que uma vez, ou mesmo que o nome não seja o mesmo mas esteja representando o mesmo objeto, considerá-lo apenas uma vez.

**2) Descartar substantivos que representem um único indivíduo:** Para chegar a essa conclusão, uma boa idéia é fazer a seguinte pergunta: "Se este elemento se transformar em uma tabela terá somente uma linha?". Caso a resposta for afirmativa, descartar este substantivo.

**3) Descartar substantivos que estão presentes somente para entendimento do problema:** também nesse caso, podemos identificar melhor estes elementos a partir de uma pergunta: "Preciso guardar informações sobre este objeto?". Se a resposta for não, significa que o mesmo não relevante a ponto de ser mantido no modelo e deverá ser descartado.

**4) Descartar objetos que são referência a uma futura aplicação:** Por tratar-se do processo de elaboração do Modelo Conceitual, não deve haver, nesse momento, preocupação com telas, relatórios, estatísticas, cálculos e portanto, caso tais elementos tenham sido identificados, deverão ser descartados.

**5) Descartar substantivos que se transformados em entidades teriam apenas um atributo:** Analisar se o objeto identificado, o substantivo possui somente uma característica, ou seja, caso venha a se transformar em tabela tenha somente uma coluna. Se sim, possivelmente esse elemento não será mais tratado como entidade, mas possivelmente como atributo de outra entidade.

Após analisados estes elementos, sobrarão da lista realizada conforme a sugestão do item 1 um elenco de substantivos que possivelmente representarão as reais entidades que o modelo poderá ter.

#### 4.3.6 Atributos

A identificação de entidades está diretamente relacionada a necessidade de se armazenar dados (características ou propriedades) a seu respeito. Tais características ou propriedades são denominadas **atributos**. O papel do atributo é o de descrever características ou propriedades relevantes de um conjunto de Entidades.

Os atributos referem-se aquilo que o intelecto percebe a respeito da entidade. É a partir da observação das propriedades que cada objeto do mundo real possui que conseguimos identificá-los, distingui-los, reconhecê-los. Além disso, conforme referenciado por Shlaer e Mellor na seção 4.3.1 todas as entidades de uma determinada classe possuem o mesmo elenco de atributos. Os valores que cada atributo irá assumir podem ser distintos para cada entidade no conjunto, no entanto, tais valores serão retirados de um conjunto denominado domínio (POMPILHO, 2002, p.79).

O nome dos atributos será sempre em letras **minúsculas** e no **singular** quando o atributo for monovalorado ou no **plural** quando o atributo for multivalorado.

Deseja-se que o processo de identificação de atributos alcance um elenco de atributos que sejam **completos, fatorados e independentes** (SHLAER e MELLOR, 1990, p.31).

- **Completo:** “Deve abranger todas as informações pertinentes ao objeto que está sendo definido”. Os atributos devem contemplar todas as características que proporcionem uma perfeita identificação das entidades a qual esteja associado. Assim, ao se definir a entidade PESSOAS, deve-se prever todos os atributos que me permitam caracterizar completamente os elementos (pessoas) que comporão esta entidade.

**Ex.:**       **Entidade:** PESSOAS  
              **Atributos:** nome, rua, número, bairro, cidade, estado, uf, data de nascimento, sexo, estado civil.

- **Fatorado:** Cada atributo deve ser responsável por um aspecto. Não se deve agrupar responsabilidades acessórias aos atributos. Cada parte, cada característica, deve significar um traço, uma particularidade da entidade e não estar agregando um conjunto de informações em um único elemento, em um único atributo.

**Ex.:**       **Entidade:** PESSOAS  
              **Atributos:** nome, rua, número, bairro, cidade, estado, uf, data de nascimento, sexo, estado civil.

Note que o endereço está fracionado nas suas partes constituintes e não agrupado num único elemento denominado endereço.

- **Independente:** Os valores que um atributo pode receber (domínio) devem ser independentes uns dos outros. Isso não significa que não possamos ter atributos distintos com o mesmo domínio. Cada atributo possui um conjunto dos possíveis valores que pode assumir e isso independe dos demais atributos da entidade.

**Ex.:**       **Entidade:** PESSOAS  
              **Atributos:** data de nascimento  
              **Domínio:** deve ter as características de data e ser menor que a data corrente

#### 4.3.6.1 – Classificação dos Atributos quanto a Finalidade:

Segundo Cougo, (1997, p.60-63) não é vital que se estabeleça alguma classificação dos atributos na fase da modelagem conceitual, pois isso não agregará semântica. No entanto, quando chegarmos na fase da modelagem lógica, conhecermos particularidades relacionadas aos atributos irá influenciar o estabelecimento de chaves primárias e chaves estrangeiras. Por isso, após a identificação de cada um dos atributos que as entidades terão, é interessante enquadrar cada atributo em um dos três grupos propostos abaixo:

- **Atributos Descritivos:** “Sob enfoque funcional, aplicam-se a representação das características intrínsecas dos objetos. Todo atributo que seja capaz de demonstrar ou representar, características formadoras, ou pertencentes, a um objeto poderá vir a ser enquadrado como descritivo. Todos os atributos podem ser classificados como descritivos. Alguns atributos poderão deixar de ser enquadrados como Descritivos e passarem a pertencer aos demais grupos em função de possuírem características funcionais adicionais que poderão ser captadas através de suas interpretações” (COUGO, 1997, p.60-61). A figura 4.11 demonstra alguns exemplos de atributos descritivos.

Atributo Descritivo	valor	Objeto
Nome	Maria Silva	PESSOA
Cor	Branca	IMÓVEL

Sabor	Doce	Fruta
-------	------	-------

**FIGURA4.11** Exemplo de atributos descritivos

- **Atributos Nominativos:** “Além de cumprirem a função de Descritivos, servem também como definidores de nomes ou rótulos de identificação aos objetos aos quais pertencem. Os fortes candidatos a atributos normativos serão sempre aqueles que, já no nível conceitual, são reconhecidos e denominados como nome do ..., código do..., número do..., identificador do ..., sigla do .....e que na fase do projeto lógico poderão vir a se tornar chaves candidatas contanto que atendam ao requisito de garantirem a unicidade da entidade (COUGO, 1997, p.61).

Observe que, atributos nominativos não necessariamente devam ser unívocos com relação as entidades, ou seja, não obriga que todas as entidades tenham valores distintos. O importante é que para ser nominativo ele deve rotular uma instância de um objeto, ainda que de maneira não unívoca. A figura 4.12 demonstra alguns exemplos de atributos nominativos.

Atributo Nominativo	valor	Objeto
Numero do CPF	88416618876	EMPREGADO
Sigla do Estado	ES	UNIDADE FEDERATIVA
Numero do Chassis	1ABD333GD55534	VEICULO
Nome	Rita de Cássia	ALUNO

**FIGURA4.12** Exemplo de atributos nominativos

- **Atributos Referenciais:** “Não pertencem propriamente ao objeto onde estão alocados, mas fazem algum tipo de citação, ou ligação desse objeto com outro objeto. Ao identificarmos um atributo referencial estará se assumindo a existência de outro objeto ao qual nos referenciamos. Este outro objeto, o objeto origem do atributo, pode já ter sido percebido claramente e existir previamente em nosso modelo, ou então poderá ser criado para fins de modelagem como local nativo para o atributo em questão” (COUGO, 1997, p.63).

Os atributos referenciais somente devem ser demonstrados em modelos quando nos referenciamos a entidades que não constam no nosso diagrama. Quando todas as entidades envolvidas no relacionamento estiverem presentes no diagrama, as relações são configuradas visualmente pela linha (relacionamento) entre as entidades. A figura 4.13 demonstra alguns exemplos de atributos referenciais.

Atributo Referencial	Exemplo	Objeto no qual está definido	Objeto de origem
Matricula	1234	DEPENDENTE	EMPREGADO
CNPJ	9192847127	NOTA FISCAL	FORNECEDOR
Fabricante	HONDA	VEICULO	FABRICA

**FIGURA4.13** Exemplo de atributos referenciais

- **“Na Fase de Modelo Conceitual:** Nesta fase, a presença de atributos referenciais pode demonstrar não normalização, ou seja, a alocação de dados em lugares não adequados sob o ponto de vista de manipulação de dados”.
- **“Na Fase do Modelo Lógico:** Nesta fase, quando num ambiente Relacional, a existência de atributos referenciais poderá ser normal, desde que estes atributos caracterizem-se como chaves estrangeiras” (COUGO, 1997, p.63-64).

#### Algumas considerações adicionais sobre os atributos:

Ainda sobre a identificação dos atributos é interessante que se distinga alguns elementos adicionais conforme descritos abaixo. (POMPILHO, 2002, p.80-81). Para ilustrar usaremos as entidades da figura 4.4.



- **Atributos Monovalorados ou Univalorado:** É um atributo que recebe um único valor para cada entidade. São atributos que possuem cardinalidade máxima 1.

Ex.: FORNECEDORES (código, CNPJ, razãosocial, logradouro, número, complemento, cidade, estado, codigopostal)

- **Atributos Multivalorados:** É um atributo que pode assumir vários valores para cada entidade. São atributos que possuem cardinalidade máxima N.

Ex.: FORNECEDORES (telefone).

- **Atributos Obrigatórios:** É um atributo que obriga a existência de valor para cada entidade. São atributos que possuem cardinalidade mínima 1.

Ex.: FORNECEDORES (código, CNPJ, razãosocial, logradouro, número, cidade, estado, código postal)

- **Atributos Opcionais:** É um atributo que NÃO obriga a existência de valor para cada entidade. São atributos que possuem cardinalidade mínima 0.

Ex.: FORNECEDORES (complemento)

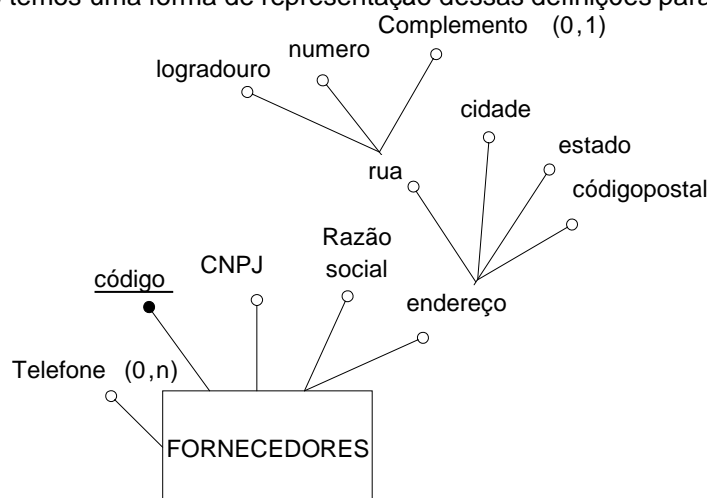
- **Atributo Composto:** São atributos formados por um ou mais atributos, ou seja, são atributos que abrigam uma estrutura de dado. O atributo nome em uma entidade pessoa, também poderá ser visto como composto se for possível e necessário separá-los nas suas várias partes (primeiro-nome, nome-intermediário, último-nome). Quando não são compostos, os atributos são denominados **simples**.

Ex.: FORNECEDORES (endereço e rua).

- **Atributo Determinante ou Identificador:** Atributo que identifica de modo inequivocamente uma Entidade. Pode ser visto como a Chave Primária dessa entidade ou Identificador dessa entidade. Serão sublinhados no diagrama como forma de distingui-lo dos demais. Há que já chame o atributo identificador de Chave Candidata.

Ex.: FORNECEDORES (código)

Na figura 4.14 abaixo temos uma forma de representação dessas definições para atributos.



**FIGURA4.14** notação dos atributos

Na figura 4.15 abaixo temos uma forma alternativa de representação dessas definições para atributos.



FORNECEDORES

Dicionário de Dados:

FORNECEDORES = código + CNPJ + razaosocial + ENDERECO + {telefone}

ENDERECO = RUA + cidade + estado + codigopostal

RUA = logradouro + número + (complemento)

**FIGURA4.15** notação alternativa para atributos

#### 4.3.6.2 – Domínio:

É o conjunto dos possíveis valores que um atributo pode assumir. No domínio de um atributo, especificamos o Formato desse atributo, seu tamanho, os valores especificamente. Por exemplo, um atributo nome para uma classe de entidades PESSOAS, poderá ter como domínio uma “cadeia de caracteres”, já o atributo sexo poderá ter como domínio as opções “Masculino” ou “Feminino”. Será mantido o estudo de domínio até este ponto, já que tal assunto foi estudado em mais detalhes no capítulo 3.

Ex.: Atributo (Sexo do Empregado)

Domínio (Campo Alfanumérico, 1 caracter, só pode assumir os valores “F” e “M”).

#### 4.3.7 Relacionamentos

Pompilho (2002, p.79) define relacionamento como “uma estrutura que indica associações entre elementos de um conjunto de entidades”. Representa um conjunto de conexões entre objetos. Cada Instância de um relacionamento representa uma ou mais associações entre zero ou mais ocorrências de um objeto e zero ou mais ocorrências de outro objeto. Um Relacionamento Binário é um par ordenado (e1, e2), onde e1 e e2 são respectivamente elementos dos conjuntos de Entidades E1 e E2.

Um relacionamento é uma afirmação de ação realizada envolvendo entidades. Por ser uma afirmação, normalmente é dita nos dois sentidos.

Ex.: CLIENTES **COMPRAM** ITENS

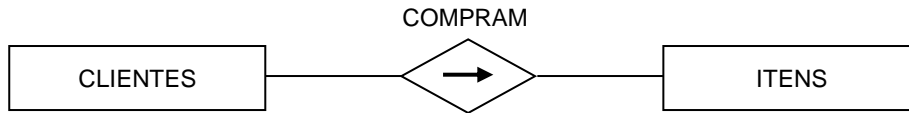
ITENS **SÃO COMPRADOS POR** CLIENTES

##### 4.3.7.1 – Representação Gráfica:

É representado por um Losango (diamante). Na parte externa coloca-se o VERBO que simboliza a ação que vincula os indivíduos das duas entidades. Na parte de dentro do losango, coloca-se uma seta indicativa do sentido de leitura do verbo. Com esses dois elementos (verbo e seta) é possível estabelecer uma forma “ativa” ou “passiva” para leitura do modelo. Observe o exemplo abaixo. Podemos afirmar que “**clientes compram itens**” denotando a voz ativa do verbo referido e por outro lado, “**itens são comprados por clientes**” denotando a voz passiva. Fazer esse exercício na construção dos relacionamentos dará maior semântica ao mesmo, proporcionando a validação e análise dos relacionamentos modelados. Observe que da mesma maneira que uma entidade possui uma denominação (nome) um relacionamento também o terá. Esta denominação, nesse caso, é dada pelo verbo utilizado. Também da mesma forma que um nome inadequado para uma entidade poderá acarretar em uma interpretação incorreta do modelo, um nome inadequado para um relacionamento poderá causar o mesmo transtorno no entendimento e validação do modelo.

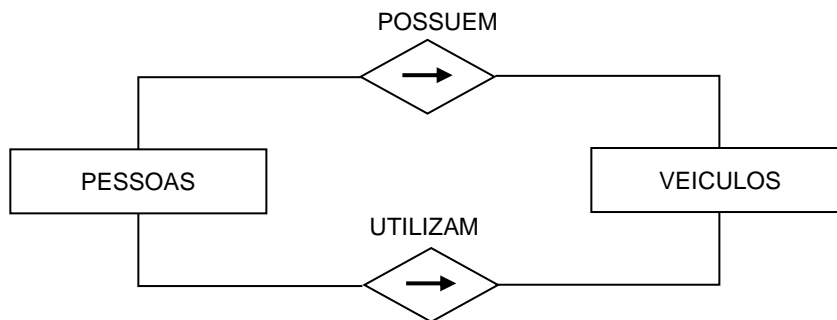


Nas figuras abaixo suprimimos os atributos para não poluir o modelo, já que o foco agora é o estudo dos relacionamentos.



**FIGURA4.16** Notação gráfica para relacionamento

Pode existir mais que um tipo de relacionamento entre duas entidades:

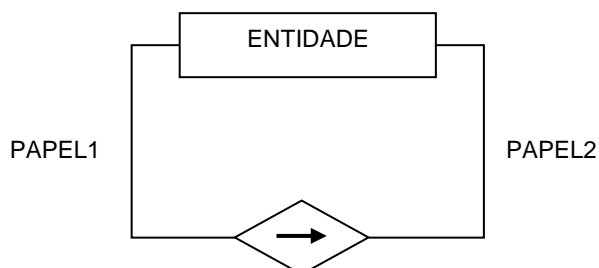


**FIGURA4.17** Dois relacionamentos envolvendo o mesmo conjunto de entidades

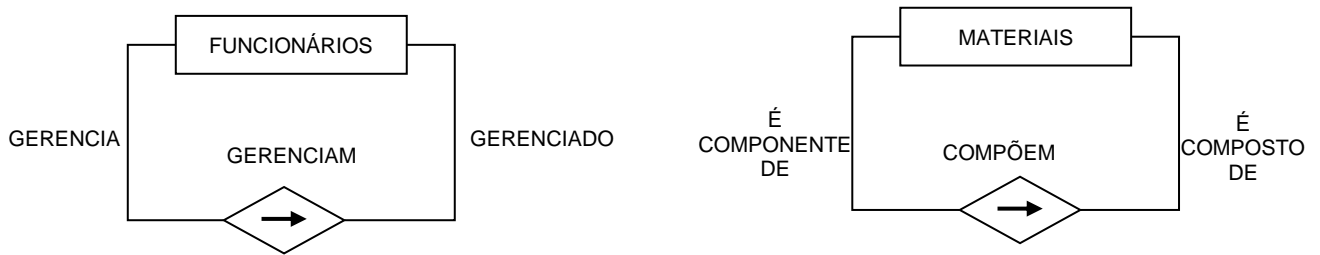
O relacionamento poderá ocorrer tanto entre elementos do **mesmo tipo** (as instancias que participam do relacionamento pertencem a mesma classe), quanto entre elementos de **tipos diferentes** (as instâncias que participam do relacionamento pertencem as classes diferentes).

#### 4.3.7.2 – Relacionamento entre objetos do mesmo tipo – Auto-relacionamento

Também chamado de Relacionamento Recursivo, ocorre quando estão envolvidos objetos da mesma entidade. Isso faz com que tenhamos que reconhecer diferentes papeis para um mesmo objeto. Se R é um relacionamento que relaciona elementos de um conjunto de entidades E a elementos desse mesmo conjunto E, R é denominado de um Auto-Relacionamento (COUGO, 1997, p.70).



**FIGURA4.18** Notação genérica para auto-relacionamento

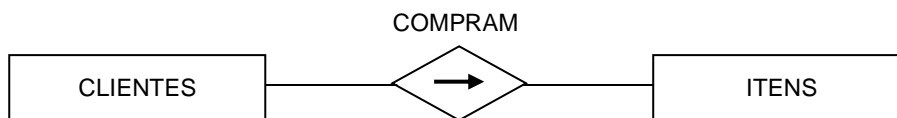


**FIGURA4.19** Exemplos de auto-relacionamento

Observe que o auto-relacionamento faz com que tenhamos que perceber diferentes papéis para os indivíduos de uma mesma classe.

#### 4.3.7.3 – Relacionamento entre objetos de tipo diferente

São os mais frequentemente observados, pois ocorrem quando estão envolvidos objetos de entidades diferentes. Aqui, instâncias de um tipo de objeto associam-se a instâncias de outro tipo. (COUGO, 1997, p.67).



**FIGURA4.20** Relacionamento entre indivíduos de entidades diferentes

#### 4.3.7.4 – Caracterização dos Relacionamentos

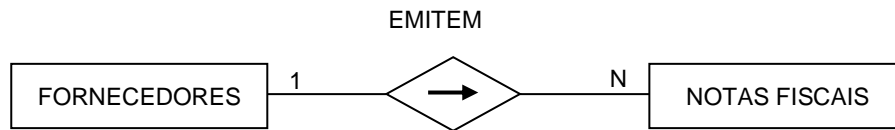
Um relacionamento demonstra uma associação entre objetos, de igual ou diferente tipo. Embora pareça extremamente simples, a fase de caracterização dos relacionamentos é um dos pontos vitais para a construção de modelos íntegros. Tal caracterização deve ser baseada no entendimento de alguns requisitos (COUGO, 1997, p.75):

- Grau ou Cardinalidade do Relacionamento
- Número de elementos que participam do relacionamento
- Condição de participação do elemento no relacionamento
- Condição de estabelecimento do relacionamento

Aqui, por questões didáticas, será tratado cada requisito isoladamente, porém nos modelos deverão ser analisados de maneira integrada.

##### a) Grau ou Cardinalidade do Relacionamento:

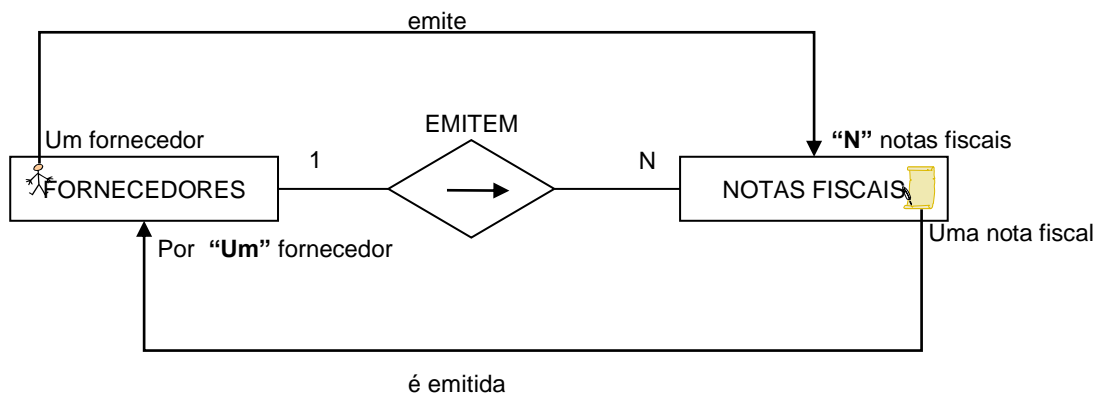
Refere-se ao número **máximo** de possíveis associações de um relacionamento. Busca definir o número **máximo** de pares que cada indivíduo de uma entidade poderá fazer com os indivíduos de outra entidade (ou da mesma, para o caso de auto-relacionamento) (COUGO, 1997, p.75). Tratam das regras de negócio que irão impactar diretamente o projeto lógico do banco de dados, pois a análise das cardinalidade influencia diretamente a construção do projeto lógico do banco de dados.



**FIGURA4.21** Notação do grau ou cardinalidade do relacionamento

Tomando como exemplo o modelo acima, podemos estabelecer a seguinte regra para notação da cardinalidade:

Fixar “um” elemento fictício da entidade FORNECEDORES e perguntar a ele: Um fornecedor pode emitir quantas notas fiscais? A resposta a esta pergunta será anotada ao lado da entidade NOTAS FISCAIS. De forma semelhante, deve-se fazer a pergunta no sentido contrário, pois também precisamos saber quantos pares NOTAS FISCAIS faz com FORNECEDORES. Então, faça novamente a pergunta: Uma nota fiscal pode ser emitida por quantos fornecedores? A resposta deverá ser anotada ao lado da entidade FORNECEDORES.

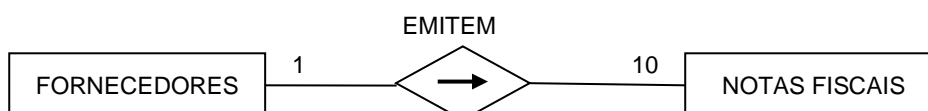


**FIGURA4.22** Notação do grau ou cardinalidade do relacionamento

Interpretação do modelo proposto:

- Um fornecedor pode emitir no **máximo** várias notas fiscais.
- Uma nota fiscal pode ser emitida por no **máximo** um fornecedor.

Em alguns casos é possível identificar o limite máximo de ocorrências de determinado objeto, no relacionamento, como por exemplo:



**FIGURA4.23** relacionamento com limite máximo de cardinalidade

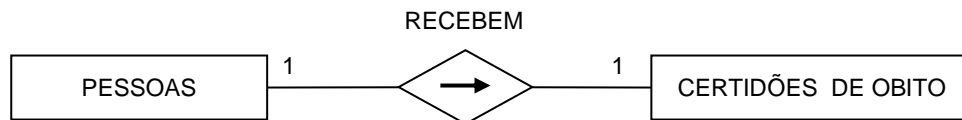
Interpretação:

- Um fornecedor emite no máximo 10 notas fiscais.
- Uma nota fiscal é emitida por no máximo 1 fornecedor.

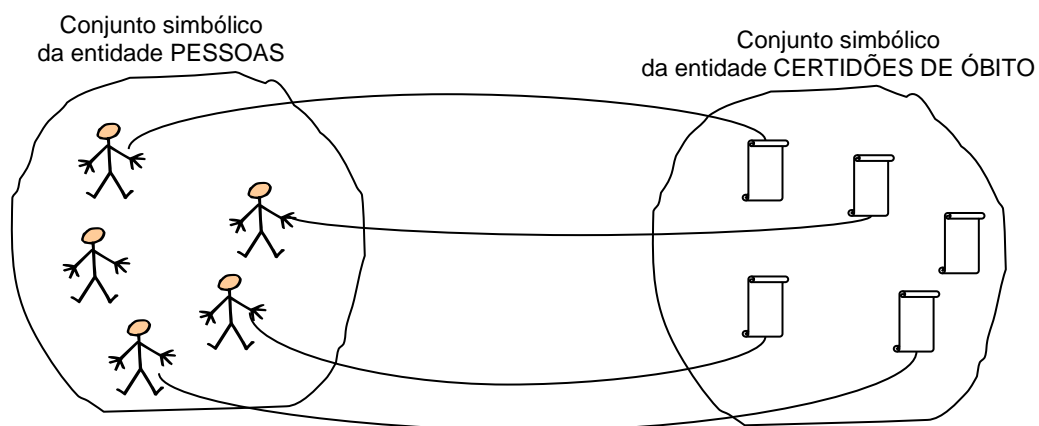
Os graus dos relacionamentos podem ser enquadrados em três grandes grupos:

- 1:1 (Um-para-um)
- 1:N (Um-para-Muitos)
- M:N (Muitos-para-Muitos)

→ **Relacionamento 1:1:** É um caso especial de relacionamento 1:N onde o N é fixado com o valor Máximo de 1. Significa que 1 elemento do Tipo A se relaciona com no máximo 1 elemento do Tipo B e que 1 elemento do Tipo B se relaciona com no máximo 1 elemento do Tipo A.



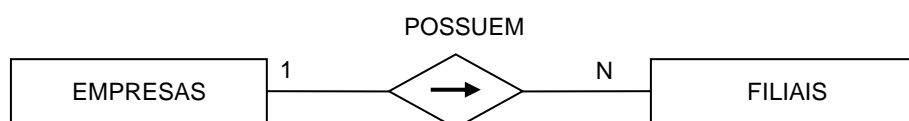
**FIGURA4.24** relacionamento de cardinalidade 1:1 (um para um)



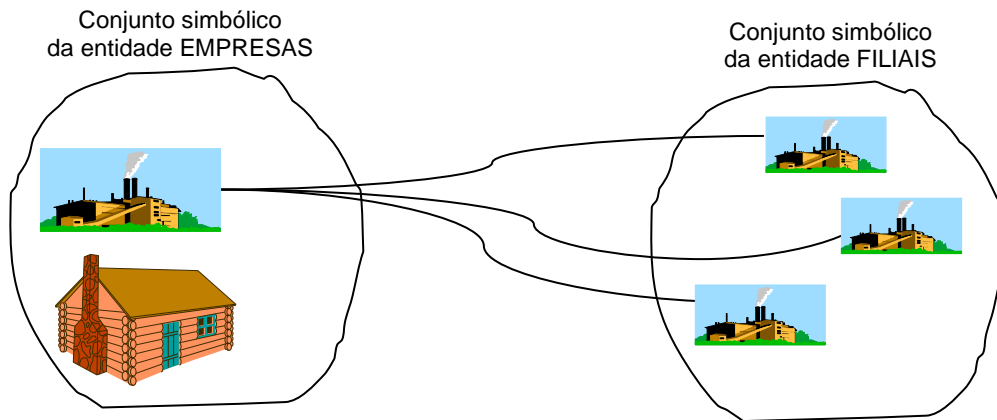
**FIGURA4.25** representação simbólica de relacionamento 1:1 (um para um)

Observe que não estamos tratando nesse momento do grau mínimo, que diz se todo elemento do grupo A deve obrigatoriamente se relacionar a no mínimo um elemento do grupo B e vice-versa. Este dado será tratado mais adiante no item “c) condição de participação do elemento no relacionamento”.

→ **Relacionamento 1:N:** Este relacionamento é bastante comum, sendo encontrado na maior parte dos modelos. Significa que 1 elemento do Tipo A se relaciona com no máximo N elementos do Tipo B e que 1 elemento do Tipo B se relaciona com no máximo 1 elemento do Tipo A. Note que novamente não estamos tratando da obrigatoriedade do relacionamento, ou seja, de que todo A esteja relacionado a no mínimo um B ou que todo B esteja relacionado a no mínimo um de A. Mas é importante que, para respeitar os números de pares possíveis idealizados na combinação 1:N, uma vez que um elemento de B faça par com algum elemento de A, ele não poderá formar par com nenhum outro elemento de A.

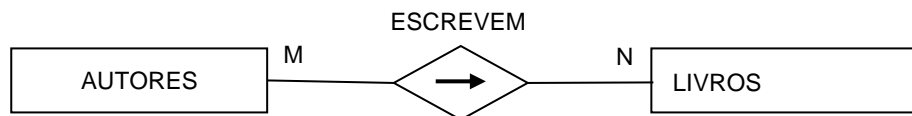


**FIGURA4.26** relacionamento de cardinalidade 1:N (um para muitos)

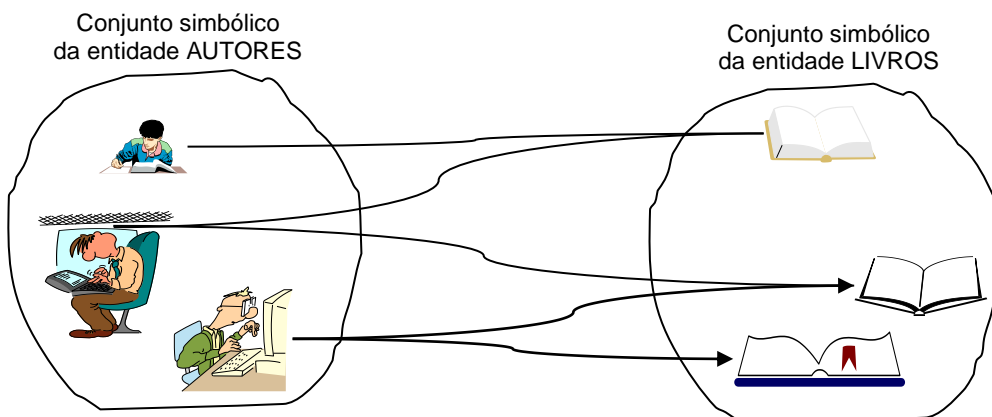


**FIGURA4.27** representação simbólica de relacionamento 1:n (um para muitos)

→ **Relacionamento M:N:** Nesse tipo de relacionamento, teoricamente, não há restrição quanto às possíveis ligações a serem estabelecidas entre os conjuntos A e B. Num relacionamento M:N, um elemento de um conjunto A pode se associar a 0 (nenhum) , 1 (um) ou N (muitos) elementos do conjunto B. Por sua vez, um elemento de um conjunto B pode se associar a 0 (nenhum) , 1 (um) ou N (muitos) elementos do conjunto A.



**FIGURA4.28** relacionamento de cardinalidade 1:N (um para muitos)



**FIGURA4.29** relacionamento de cardinalidade M:N (muitos para muitos)

Observe que não há obrigatoriedade de que todos os elementos do conjunto A tenham vários elementos do conjunto B associados a ele. Alguns podem ter nenhum, outros podem ter somente 1 e outros podem ter vários. O mesmo também é válido para relação do conjunto B.



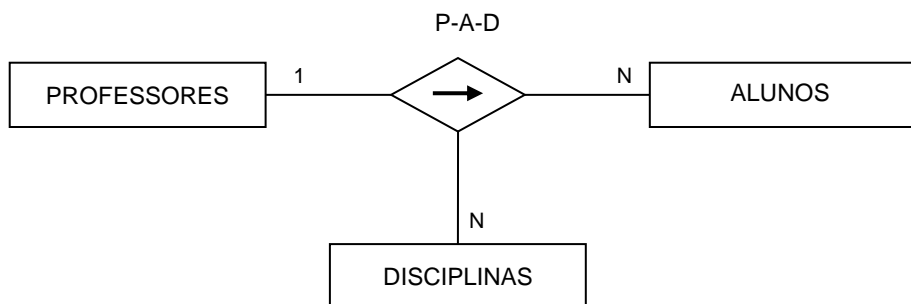
Outra observação importante, é o porquê da denominação desse gênero como M:N e não N:N. Utiliza-se as duas letras para que fique explícito que as duas grandezas são totalmente diferentes. Isso quer dizer que se um elemento do conjunto A pode se relacionar com até 10 elementos do conjunto B, o inverso não é obrigatório, um elemento do conjunto B não necessariamente precisará se relacionar com 10 elementos do conjunto B.

#### b) Número de elementos que participam do relacionamento:

Até agora, a modelagem baseou-se no relacionamento que envolve somente dois conjuntos de entidades, ou somente nos objetos pertencentes a duas entidades. Na verdade, um relacionamento pode se estabelecer entre duas ou mais entidades (SETZER, 2005, p.52). Do ponto de vista conceitual, o estabelecimento de relacionamentos Ternários (3 conjuntos de entidades), Quaternários (4 conjuntos de entidades) ou maiores não é muito freqüente. Algumas vezes, mas não sempre, no lugar de um relacionamento ternário ou maior, pode existir um objeto que não está explícito e que agrega em si um conjunto de outras informações bem como de relacionamento com outros objetos (COUGO, 1997, p.85).

- Relacionamentos Binários: AVIÃO transporta CARGA.
- Relacionamentos Ternários: PROFESSORES ministram DISCIPLINAS para ALUNOS.

Na falta de um nome adequado para o relacionamento, podemos adotar outra solução como por exemplo, colocar as iniciais dos nomes das entidades envolvidas ou mesmo colocar os três nomes por extenso (SETZER, 2005, p.51).



**FIGURA4.30** relacionamento ternário

- Dado um aluno em uma disciplina, há um **SÓ** professor associado a eles, isto é, um aluno **NÃO** pode ter em uma certa disciplina mais do que um professor.
- Por outro lado, um professor pode ministrar uma disciplina para um número qualquer de alunos;
- E um professor pode dar para um certo aluno mais do que uma disciplina.

Em um relacionamento ternário, “a leitura da cardinalidade refere-se a *pares de entidades*. Em um relacionamento R entre as 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” (HEUSER, 2001, p.19-20). Tomando o exemplo acima, faz-se a seguinte leitura:

- Toma-se o par formado por um elemento da entidade PROFESSORES e um elemento da entidade ALUNOS e faz-se a leitura de quantos elementos da entidade DISCIPLINAS poderão participar do par formado por PROFESSORES-ALUNOS.
- Feita a primeira análise considerando-se esta lógica, faz-se o rodízio das entidades: Agora se toma um elemento de ALUNOS e um elemento de DISCIPLINAS e perguntam-se quantos PROFESSORES poderão participar do par de ALUNOS-DISCIPLINAS;
- E por fim, toma-se o par formado por um elemento de DISCIPLINAS e um elemento de PROFESSORES e perguntam-se quantos ALUNOS poderão estar presentes no par de DISCIPLINAS-PROFESSORES.

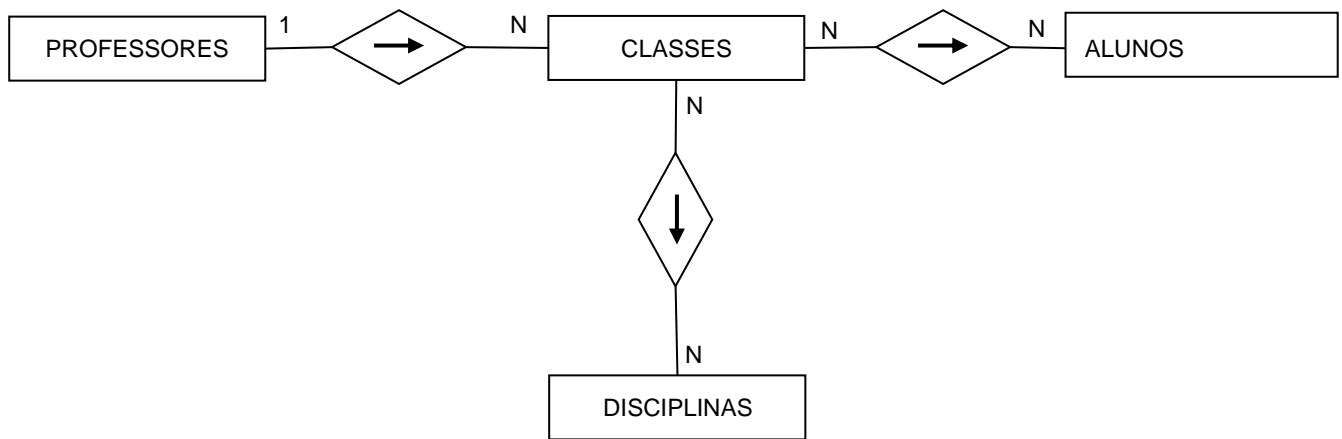
No exemplo acima, leríamos:

- para um professor e uma disciplina, podemos ter N alunos associados.

- para um professor e um aluno, podemos ter várias disciplinas.
- para uma disciplina e um aluno, podemos ter somente 1 professor.

Um dado muito importante nesse tipo de relacionamento, é que todos os elementos envolvidos têm que estar presente simultaneamente para que o relacionamento ocorra.

Segundo Silberschatz (2006, p.148-149) algumas vezes é possível desenvolver um modelo de dados que só possua relacionamentos binários. Dessa forma, situações onde se mapeia inicialmente relacionamentos ternários ou de grau maior, poderiam ser transformados em vários relacionamentos binários. Observando a figura do relacionamento PROFESSORES, ALUNOS e DISCIPLINAS demonstrado anteriormente poderíamos ter como solução o modelo proposto abaixo.



**FIGURA4.31** solução alternativa para relacionamento ternário

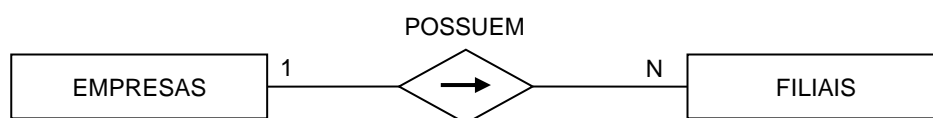
No entanto, que fique claro que isso nem sempre é possível e nesses casos, tentar transformar qualquer relacionamento que não seja binário em relacionamento binário pode não ser uma boa alternativa. Mesmo assim, fica aqui a reflexão, para efeito de enriquecimento do aprendizado e do processo de modelagem.

### c) Condição de participação do elemento no relacionamento:

Vimos anteriormente que o grau do relacionamento trata do número de pares possíveis entre os indivíduos das entidades. Naquele momento, abordado no item “a) Grau ou Cardinalidade do Relacionamento” tratamos somente da **cardinalidade máxima**, ou seja, do número máximo de pares possíveis. Nesse momento, trataremos da análise da obrigatoriedade ou não do relacionamento. A obrigatoriedade ou não do relacionamento trata da possibilidade de que elementos de um conjunto A possam não formar par com nenhum elemento de um conjunto B e vice-versa.

Tomemos novamente o exemplo utilizado anteriormente para ilustrarmos a questão proposta. O Grau estudado anteriormente, determinada que:

- uma empresa pode ter muitas filiais
- uma filial deve estar associada a somente uma empresa



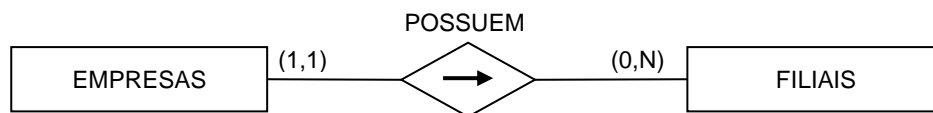
**FIGURA4.32** condição de participação no relacionamento

Porém, algumas outras questões muito importantes para o projeto não estão sendo respondidas, como por exemplo:

- toda empresa deve ter alguma filial, nem que seja uma?
- uma filial pode estar associada a nenhuma empresa?

Observe que tais questões não estão sendo respondidas pelo modelo. Certamente que as respostas podem variar de acordo com o contexto, mas haverá uma resposta e essa deverá estar mapeada no modelo. Para expressar tais informações, ampliaremos o conceito de **Grau do Relacionamento** estudado anteriormente para abrigar o **grau mínimo**. Nos exemplos mostrados na seção que tratou do grau, quando expressamos um relacionamento como sendo 1:N, nos referíamos aos graus máximos de cada lado do relacionamento. O grau mínimo trata do limite mínimo de pares possíveis ou o “menor valor de participação dos elementos do conjunto A e B no relacionamento” (COUGO, 1997, p.90).

Para expressar graficamente os graus máximos e mínimos, modificaremos levemente a notação vista até aqui, acrescentando o valor mínimo, antes do valor máximo, separados por uma vírgula. Assim, o exemplo de EMPRESAS e FILIAIS visto acima cujo grau foi expresso como sendo 1:N (representando somente os máximos), para abrigar o grau mínimo passará a ser notado 1,1:0,N. Observe o modelo abaixo com a modificação proposta.



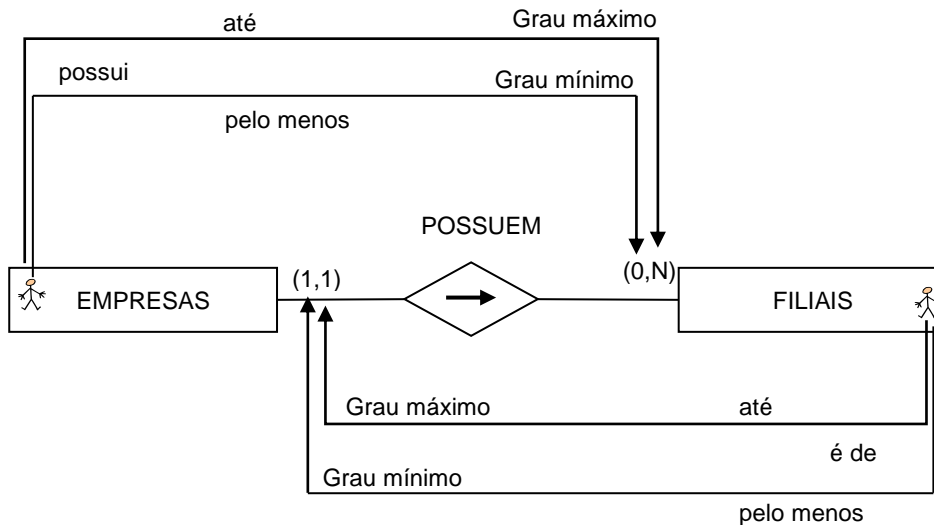
**FIGURA4.33** relacionamento com cardinalidade máxima e mínima

Interpretação:

- Uma empresa possui no mínimo nenhuma e no máximo muitas filiais.
- Uma filial é de no mínimo 1 e no máximo 1 empresa.

Também podemos ler:

- A empresa pode não ter filiais ou poderá ter muitas.
- toda filial deve estar vinculada a obrigatoriamente uma e somente uma empresa.

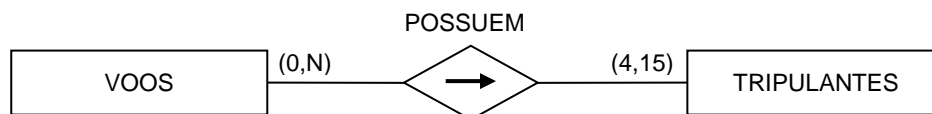


**FIGURA4.34** detalhamento da cardinalidade máxima e mínima

Dessa forma, toda vez que desejamos expressar que um indivíduo de uma entidade pode não participar de qualquer relacionamento (**relacionamento condicional**), representaremos com a cardinalidade mínima 0, ao passo que para representar a obrigatoriedade de participação em pelo menos um relacionamento (**relacionamento incondicional**), usaremos cardinalidade mínima 1.

Outra informação importante sobre a notação adotada é a de que o valor referente à cardinalidade mínima (obrigatoriedade ou opcionalidade de participação no relacionamento) está sempre escrita no lado esquerdo da vírgula, dentro do parêntesis, ao passo que a cardinalidade máxima está sempre escrita no lado direito da vírgula.

Em algumas situações, será possível determinar um grau mínimo e máximo específico, como o exemplo abaixo irá expressar. Nesse exemplo, o número mínimo e máximo de tripulantes que devem estar presentes em um voo é conhecido, ou seja, um voo deve ter no mínimo 4 e no máximo 15 tripulantes. Embora semanticamente possa estar correto, modelos com esta característica podem ser mais frágeis, pois mudanças nesses valores podem impactar o projeto.



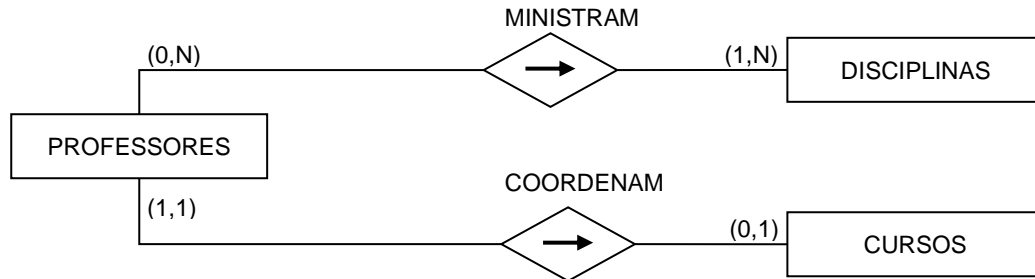
**FIGURA4.35** limite restrito de cardinalidade máxima e mínima

#### d) Condição de estabelecimento do relacionamento:

Nessa fase, analisaremos quando o relacionamento de elementos de duas entidades impacta os possíveis relacionamentos desses mesmos indivíduos com estas ou outras entidades. As condições possíveis podem ser enquadradas em três tipos (COUGO, 1997, p.94):

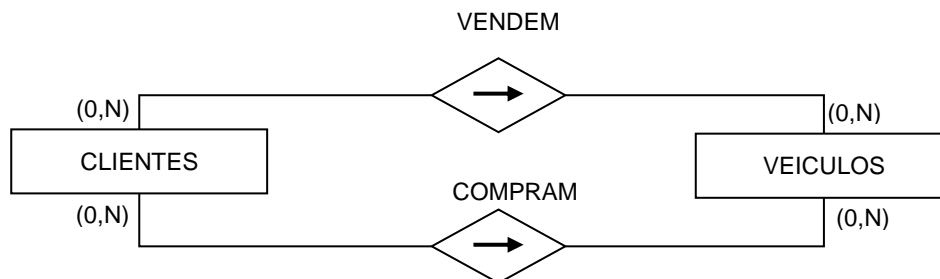
- Relacionamentos Independentes
- Relacionamentos Contingentes
- Relacionamentos mutuamente exclusivos

- **Relacionamentos Independentes:** Permitem que ocorra relacionamento entre elementos das entidades, sem que se tenha que avaliar qualquer outro relacionamento existente (COUGO, 1997, p.94).



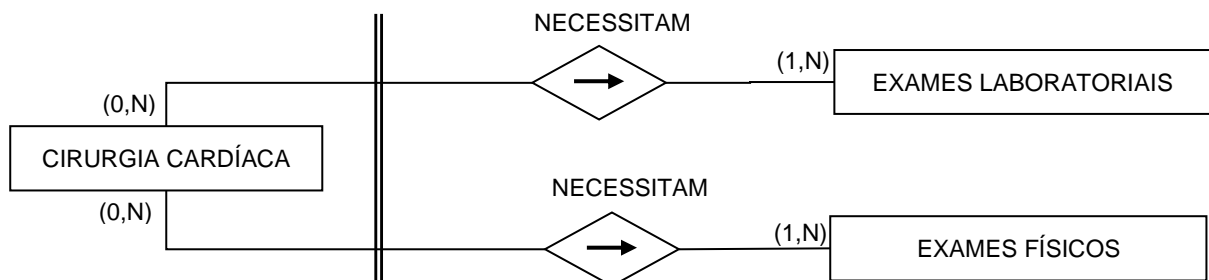
**FIGURA4.36** exemplo de relacionamento independente com mesma entidade

Podem ocorrer tanto entre entidades distintas, como entre as mesmas entidades.



**FIGURA4.37** relacionamento independente com entidades distintas

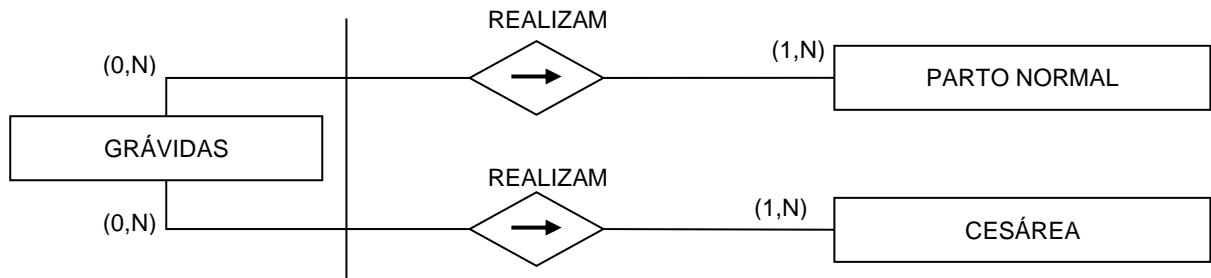
- **Relacionamentos Contingentes:** São relacionamentos que, “tendo dependência uns com os outros, impõem o estabelecimento simultâneo de associações entre os vários elementos envolvidos” (COUGO, 1997, p.94). A presença desse tipo de relacionamento é mais comum quando se percebe associações com a mesma finalidade porém envolvendo elementos diferentes. Nessa situação, mais de um relacionamento deve ocorrer em um dado instante. A notação adotada são **duas barras verticais** sobre as linhas que expressam os relacionamentos que serão tratados como contingentes.



**FIGURA4.38** relacionamento contingente

- **Relacionamentos mutuamente exclusivos:** Nesse tipo de relacionamento, se a associação de um elemento for estabelecida, não poderá ser estabelecida pelos demais (COUGO, 1997, p.97). A notação adotada é **uma**

**barra vertical** sobre as linhas que expressam os relacionamentos que serão tratados como mutuamente exclusivos.



**FIGURA4.39** relacionamento mutuamente exclusivo

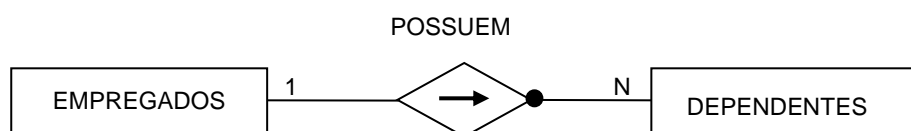
#### 4.3.7.5 – Relacionamentos Parciais ou Totais

A noção de Parcialidade e Totalidade está ligada a Cardinalidade Mínima de um relacionamento. Trata-se de uma restrição muito comum e importante na modelagem de casos práticos (SETZER, 2005, p.38).

Diz-se que um relacionamento é **parcial**, quando no par ordenado dos elementos  $x$  e  $y$  oriundos das Entidades  $X$  e  $Y$ , **não** existe a obrigatoriedade de que para todo  $x$  exista um  $y$  relacionado, ou seja, você pode encontrar elementos na Entidade  $X$  para o qual não exista relação com nenhum elemento da Entidade  $Y$ .

Diz-se que um relacionamento é **total**, quando no par ordenado dos elementos  $x$  e  $y$  oriundos das Entidades  $X$  e  $Y$ , **existe** a obrigatoriedade de que para todo  $x$  exista um  $y$  relacionado, ou seja, todo elemento na Entidade  $X$  deve possuir uma relação com no mínimo um elemento da Entidade  $Y$ .

Notação: Representamos a totalidade acrescentando uma **bolinha cheia** na saída da aresta do Relacionamento que une o Losango ao ângulo da Entidade, no caso desse Relacionamento ser **Total** para esta Entidade.

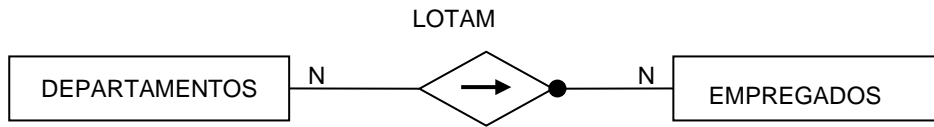


**FIGURA4.40** notação para totalidade de relacionamento

Analisando o Modelo:

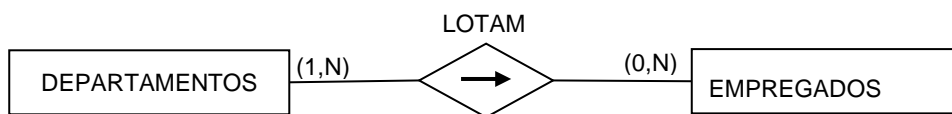
- Um Empregado pode ter no mínimo 0 e no máximo  $N$  Dependentes. Porém todo Dependente deve estar associado à no mínimo 1 e no máximo 1 Empregado.
- Diz-se que o relacionamento **POSSUEM** é **Total** em relação a **DEPENDENTES** e **Parcial** em relação a **EMPREGADOS**

Dado o conjunto de Entidades **EMPREGADOS** e **DEPARTAMENTOS** e o Relacionamento **LOTAM**, a restrição de totalidade impõe que todo elemento de **EMPREGADOS** esteja no relacionamento **LOTAM**. No entanto, podemos ter um departamento que não esteja em nenhum relacionamento em **LOTAM**, ou seja, um departamento que não possua nenhum empregado lotado. Se ambas restrições são verdade, diz-se que o Relacionamento é **Total** em **EMPREGADOS** e **Parcial** em **DEPARTAMENTOS**.



**FIGURA4.41** outro exemplo de totalidade de relacionamento

Uma notação alternativa para a totalidade é a de explicitar na própria cardinalidade os valores máximos e mínimos conforme é adotado nos demais exemplos desse material. Nesse caso, a totalidade é representada pela cardinalidade mínima igual a 1 ao passo que a parcialidade é representada pela cardinalidade mínima igual a 0.



**FIGURA4.42** totalidade utilizando cardinalidade mínima

#### 4.3.7.6 – Relacionamentos com Atributos

Assim como um conjunto de entidades possuem características, em várias situações de modelagem podemos encontrar atributos que pertencem não a um conjunto de entidades, mas sim a um conjunto de relacionamento. Atributos de relacionamento são atributos que não pertencem exclusivamente às entidades presentes em um modelo, mas sim, ao relacionamento que vincula estas entidades. O dado para este atributo só existirá quando o relacionamento ocorrer, já que “Esses dados denotam a existência de informações que só podem ser estabelecidas ou consideradas quando da existência de uma associação entre elementos” (COUGO, 1997, p.100). Somente pela existência das entidades separadamente não é garantia de valor para este tipo de atributo.

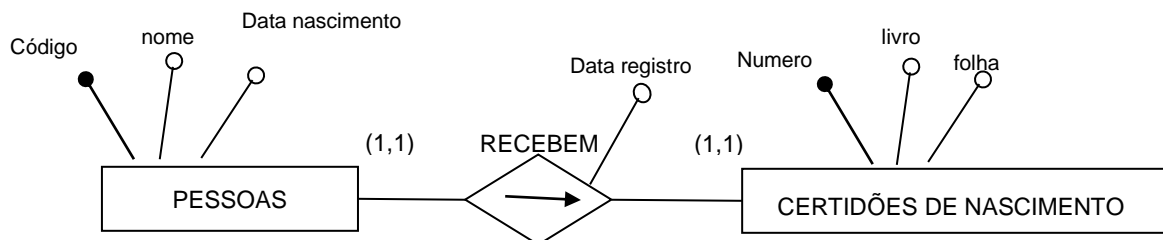
No entanto, alguns tipos de atributos de relacionamento podem ser perfeitamente alocados nas entidades participantes do mesmo. O que determinará se o atributo do relacionamento poderá ser alocado em alguma entidade pertencente ao relacionamento será a **cardinalidade máxima** desse relacionamento (SILBERSCHATZ, 2006, p.149-150).

- Para relacionamentos de cardinalidade 1:1 o atributo do relacionamento dessas duas entidades **poderá** ser alocado em qualquer uma das entidades participantes.
- Para relacionamentos de cardinalidade 1:N, o atributo do relacionamento somente **poderá** ser alocado na entidade que possui a cardinalidade N.
- Para relacionamentos de cardinalidade M:N não teremos esta liberdade de alocar tais atributos nas entidades participantes e nesse caso, **deverão** obrigatoriamente ser alocados no relacionamento.

Para ilustrar as três situações de cardinalidade, vamos analisar os três exemplos abaixo que possuem cardinalidade 1:1, 1:N e M:N

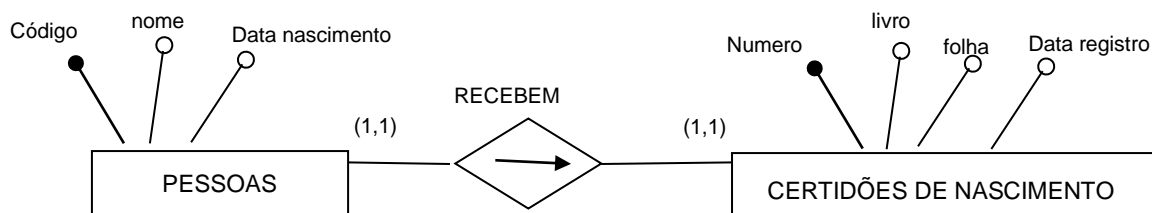
Exemplo 1 – Relacionamento com cardinalidade máxima 1:1 com atributo alocado no relacionamento.





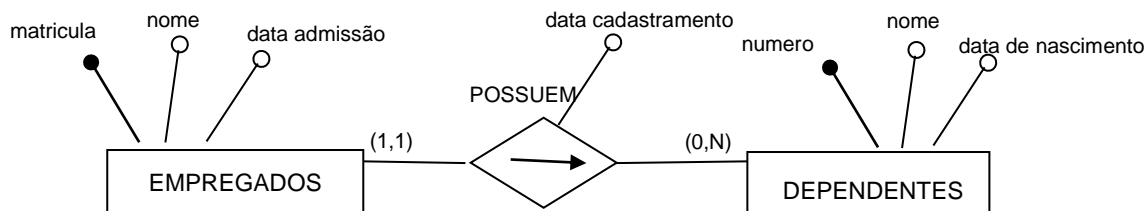
**FIGURA4.43** relacionamento de cardinalidade 1:1 – notação do atributo de relacionamento

Relacionamento com cardinalidade 1:1 com atributo alocado na entidade da direita.



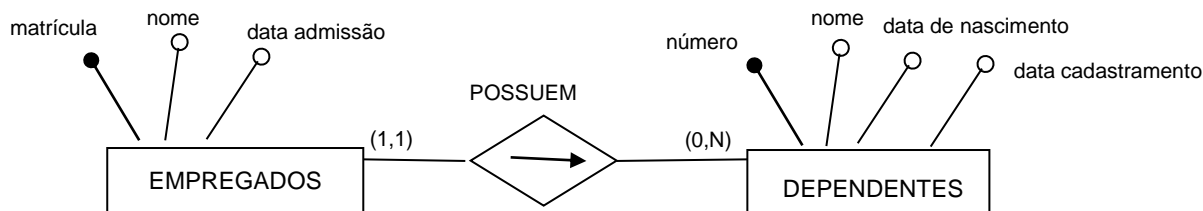
**FIGURA4.44** relacionamento de cardinalidade 1:1 - notação do atributo do relacionamento

Exemplo 2 – Relacionamento com cardinalidade máxima 1:N com atributo alocado no relacionamento.



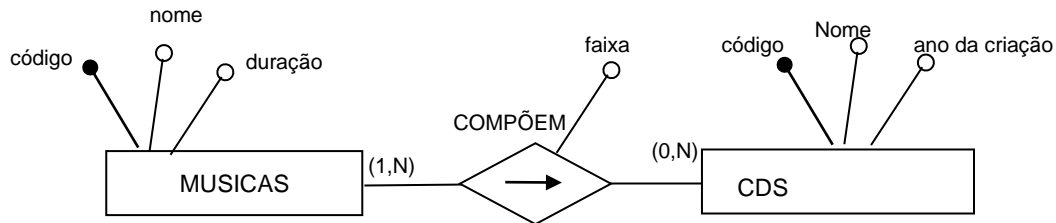
**FIGURA4.45** relacionamento de cardinalidade 1:N - notação do atributo do relacionamento

Relacionamento com cardinalidade 1:N com atributo alocado na entidade de cardinalidade máxima N.



**FIGURA4.46** relacionamento de cardinalidade 1:N - notação do atributo do relacionamento

Exemplo 3 – Relacionamento com cardinalidade máxima M:N com atributo alocado no relacionamento.



**FIGURA4.47** relacionamento de cardinalidade M:N - notação do atributo do relacionamento

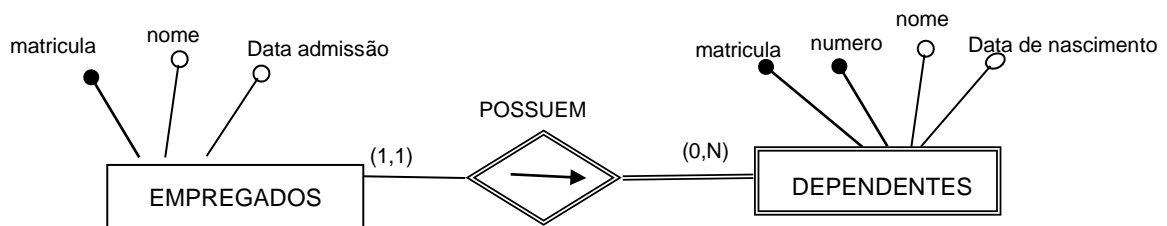
**Em um relacionamento de cardinalidade M:N, pode-se aplicar o seguinte teste para analisar onde deverá ser alocado o atributo do relacionamento.**

- Fixa o um indivíduo da entidade da esquerda e variam-se os indivíduos da entidade da direita.
  - Se o conteúdo do atributo em questão se mantém o mesmo, significa que ele pertence à entidade da esquerda.
  - Se o conteúdo de atributo se modifica, significa que ele **não** pertence à entidade da esquerda.
- Fixa um indivíduo da entidade da direita e variam-se os indivíduos da entidade da esquerda.
  - Se o conteúdo do atributo se mantém o mesmo, significa que ele pertence à entidade da direita.
  - Se o conteúdo do atributo se modifica, significa que ele **não** pertence à entidade da direita.

Se ao final do teste, conclui-se que o atributo analisado não pode pertencer nem na entidade a esquerda nem a entidade a direita do relacionamento, e que este atributo é essencial ao modelo, significa que o atributo deve pertencer ao relacionamento.

#### 4.3.7.7 – Entidade Fraca

São entidades que não possuem identidade própria, ou seja, não possuem atributos suficientes para formar uma chave. Sua existência está condicionada a existência de outra entidade dita **Forte**. O relacionamento entre a entidade forte e a fraca é de grau **1:N** e a participação da entidade fraca é **total**. Por essa razão, o atributo determinante da entidade fraca será formado pelo atributo determinante da entidade forte mais um atributo diferenciador pertencente a entidade fraca (SILBERSCHATZ, 2006, p.150-151).



**FIGURA4.48** entidade fraca

No exemplo acima, a entidade DEPENDENTES é fraca e depende da existência da entidade EMPREGADOS para que possa existir. O atributo determinante de DEPENDENTES será formado pelo atributo determinante de EMPREGADOS (matricula) mais um atributo diferenciador pertencente a ela (numero). A representação gráfica nesse caso é modificada tanto na entidade fraca, no relacionamento e na aresta que a vincula ao relacionamento. Todos estes elementos passaram a ter linha dupla.

**4.3.7.8 – Recursos estendidos de Entidades e Relacionamentos:** Embora seja possível construir modelos bastante completos somente pelo uso de entidades, atributos e relacionamentos, o tempo demonstrou que novas figuras de modelagem se tornaram necessárias para melhor expressar a realidade observada. Nessa seção abordaremos dois importantes elementos. São eles: **Agregações** e **Estrutura de Generalização-Especialização** (SILBERSCHATZ, 2006, p.156).

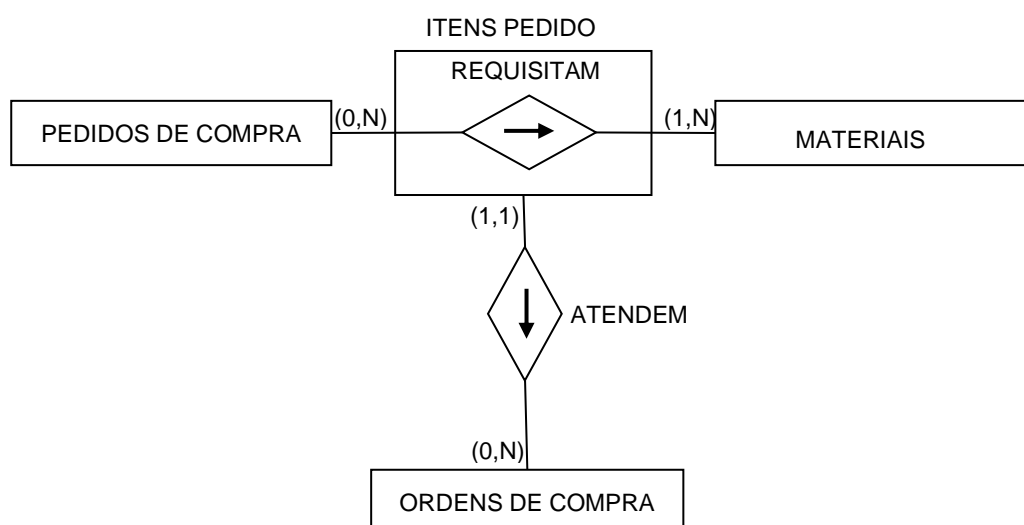
### Agregações

Embora o MER seja bastante expressivo para representar uma gama vasta de situações da vida real, ele possui a limitação de não permitir o relacionamento entre relacionamentos. Em alguns casos (e não raros) necessitamos vincular uma entidade C a um par, formado pelo relacionamento de outras entidades A e B. Ou seja, precisamos construir um relacionamento entre a C com o relacionamento de A e B. Para resolver este problema, adotamos uma estrutura denominada **Agregação**. “Agregação é uma abstração pela qual os relacionamentos são tratados como entidades de nível superior” (SILBERSCHATZ, 2006, p.156).

Agregação é então o tratamento de um relacionamento como sendo um conjunto de entidades. É importante ressaltarmos que os dois relacionamentos ocorrem em **momentos distintos do tempo** (SETZER, 2005). Analisando o modelo proposto abaixo e tomando por base a linha do tempo, está sendo demonstrado que o relacionamento REQUISITAM envolvendo as entidades PEDIDOS DE COMPRA e MATERIAIS dá-se em um primeiro momento. Em um segundo momento, dá-se o relacionamento entre o agregado ITENS DE PEDIDO e a entidade ORDENS DE COMPRA.

Dessa forma, existe uma dupla formada pelo primeiro relacionamento que existirá por um período de tempo e ao juntar-se com o segundo relacionamento dará vida a uma tripla (SETZER, 2005, p.56).

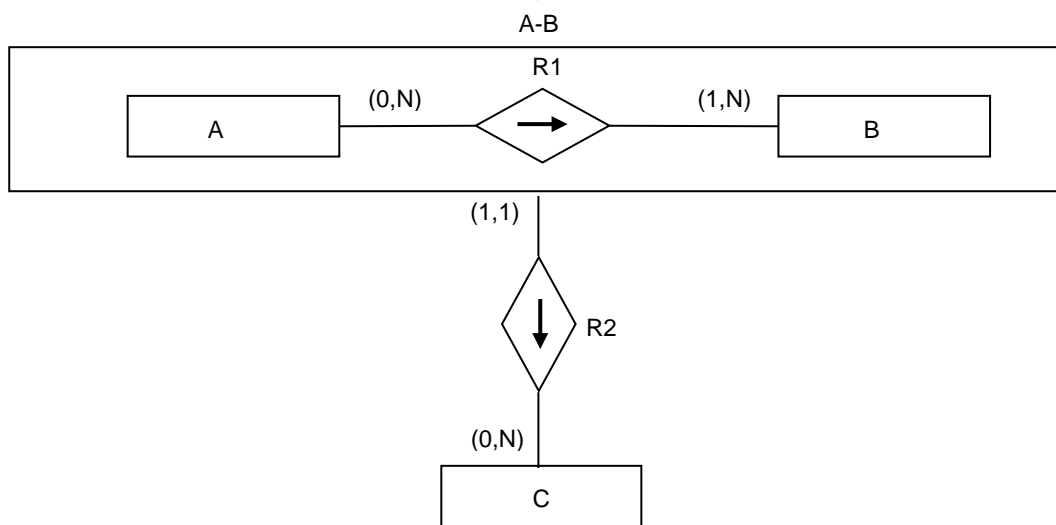
Notação: Um agregado pode ser representado por um retângulo envolvendo o relacionamento das entidades.



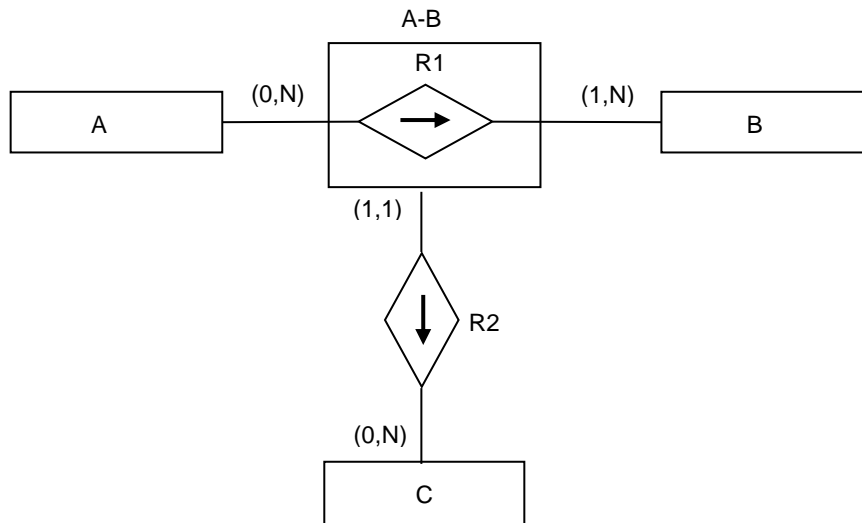
**FIGURA4.49** Agregação

**Algumas regras sobre agregações (SETZER, 2005, p.58):**

- se a cardinalidade mínima do relacionamento entre a agregação e a outra entidade for 1, estaremos na verdade tratando de um relacionamento ternário ao invés de uma agregação. Em outras palavras, quando uma agregação se relacionar com uma outra entidade, a cardinalidade mínima desse relacionamento com relação a nova entidade deverá ser 0, para que fique configurado que o primeiro relacionamento (gerador da agregação) ocorre em um momento diferente do segundo relacionamento (da agregação com a outra entidade).
- Agregações obrigatoriamente são oriundas de relacionamento M:N.
- Agregações não possuem atributos. O que pode ocorrer é a presença de atributos do relacionamento que dará origem ao agregado.
- Agregações também poderão ocorrer em relacionamentos ternários, porém com o mesmo objetivo, que é o de permitir que uma “quarta” entidade se relacione ao trio formado anteriormente, em um momento distinto do momento em que o trio se relacionou inicialmente.
- Podemos representar a agregação de duas maneiras, conforme pode ser visto das figuras 4.50 e 4.51 abaixo. Preferimos, no entanto, a segundo hipótese (figura 4.51), envolvendo somente o relacionamento.



**FIGURA4.50** Notação alternativa para agregação



**FIGURA4.51** Notação alternativa para agregação

Observe que a linha que vincula a entidade C ao relacionamento A-B, em ambos os casos, NUNCA ultrapassa o retângulo simbólico do agregado. É fundamental que se observe este detalhe, sob pena de se ter um modelo incorreto, com um relacionamento nem binário, nem ternário.

### Generalização e Especialização

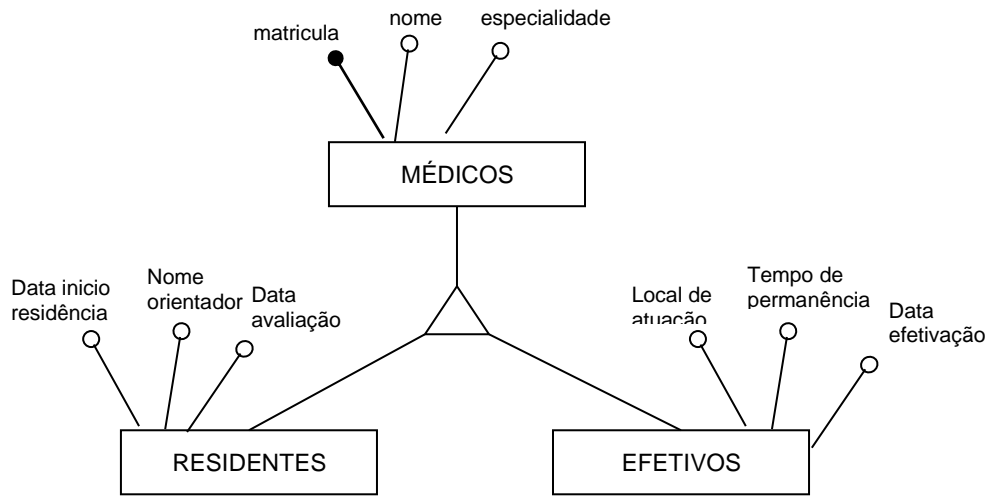
A partir do processo de Abstração é possível analisar o Mundo Real e dele extrair objetos, classes. Tal extração baseia-se na capacidade de se conseguir encontrar semelhanças entre determinados objetos de forma a agrupá-los em conjuntos. Através desse processo identificamos as Entidades que propiciam a elaboração de Modelos Entidades e Relacionamentos.

Avançando no processo de modelagem, depois da tomada inicial de conhecimento dos objetos (entidades) existentes no ambiente, parte-se para uma análise mais profunda de tais elementos que permitirá visualizar sub-conjuntos dentro dos conjuntos inicialmente identificados. Outras vezes, se perceberá semelhantes em elementos colocados em objetos distintos, dando a idéia de que esses objetos pertençam a um conjunto maior. Surge então uma nova figura de modelagem denominada **Generalização-Especialização**. **Generalizamos** objetos que foram modelados separados inicialmente, mas que pelas características observadas poderão fazer parte de um conjunto maior. Assim como, **especializamos** objetos que, tendo sido modelados inicialmente como pertencentes a um único conjunto, possuem algumas características distintas de outros indivíduos do mesmo conjunto. As características que permitem tais modificações no modelo são tanto a presença de **atributos** como de **relacionamentos**.

Por exemplo, ao analisar um ambiente hospitalar verificamos os seguintes elementos: MÉDICOS, PACIENTES, QUARTOS, SALAS DE CIRURGIA. Observamos que cada conjunto identificado, conterá um número X de elementos que se integram. Porém, numa análise posterior, distingue-se no conjunto MEDICOS as categorias MEDICOS RESIDENTES e MEDICOS EFETIVOS. Cada uma dos elementos da entidade MEDICOS tem apesar de diversas características comuns algumas que são particulares a cada especialização, como por exemplo (COUGO, 1997):

- todos os **médicos**, indistintamente possuem uma matrícula, um nome e uma especialidade.
- Quando tratar-se de um **Efetivo** tem-se, além dos atributos descritos acima, um local de atuação, tempo de permanência e data de efetivação.
- Por outro lado, quando tratar-se de um **Residente** tem-se também uma data início da residência, nome do orientador, data avaliação.

Para resolver este problema, passamos a adotar uma nova figura no Modelo que representa esta presença de Grupos e Subgrupos.



**FIGURA4.52** Estrutura de Generalização-Especialização

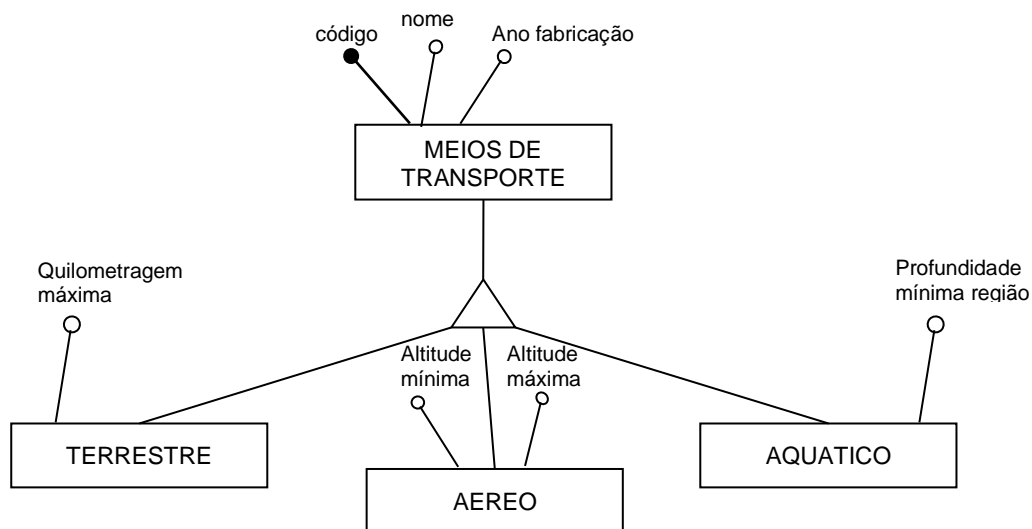
Esta notação nos permite visualizar o conjunto maior (Entidade Genérica), MEDICOS, conectada aos seus subconjuntos (Entidades Específicas) através de um triângulo. Dessa forma não temos três entidades distintas, mas sim, uma entidade representando o conjunto completo (MEDICOS) e duas entidades representando subconjuntos extraídos desse conjunto maior (RESIDENTES e EFETIVOS).

A partir dessa figura, podemos realizar a seguinte leitura.

- Todo RESIDENTE é naturalmente um MEDICO, assim como,
- Todo EFETIVO é também um MEDICO
- Todo MEDICO **ou** é EFETIVO **ou** é RESIDENTE

Um indivíduo, Residente ou Efetivo, terá o conjunto de suas características (dados), formado pelas características presentes na entidade genérica (MEDICOS) mais as características presentes na entidade específica ao qual se enquadra (RESIDENTES ou EFETIVOS).

Abaixo se encontra outro exemplo do uso da estrutura de Generalização e Especialização, porém agora com 3 ramificações de especialização.



**FIGURA4.53** Estrutura de Generalização-Especialização

Regras para se identificar à necessidade de Especialização (COUGO, 1997, p.122):

**Regra1:** Existe algum atributo que seja aplicável a somente um subconjunto de elementos e não a todos.

**Regra2:** Existe algum relacionamento que seja aplicável a somente um subconjunto de elementos e não a todos?

**Regra3:** Não poluir o modelo inserindo detalhes desnecessários.

**Regra4:** Garantir a máxima compreensão do modelo com o mínimo de uso de recursos adicionais, como dicionários de dados, documentos, anexos.

Os dois modelos propostos acima (MEDICOS e MEIOS DE TRANSPORTE) refletem a situação de um processo de projeto realizado com uma abordagem **TOP DOWN**, onde parte-se de um conjunto de entidades iniciais e vai se refinando até alcançar níveis mais detalhados e representativos da realidade observa. Nesses dois exemplo, realizou-se a **Especialização** das Entidades inicialmente propostas, quebrando-as em subgrupos. Outra abordagem denominada **BOTTOM UP**, parte da análise de elementos mais detalhados, e novamente a medida que se analisa em detalhes cada um desses elementos, percebe-se semelhanças (atributos ou relacionamentos) entre eles que permitam agrupá-los em uma estrutura superior, resguardando suas características particulares. Este processo de agrupamento é denominado **Generalização**. Mas não importa, se o processo foi a Especialização ou a Generalização. Ambos produzem modelos semelhantes. O ponto de partida para a análise é que foi diferente. De fato, a generalização é uma inversão da especialização (SILBERSCHATZ, 2006, p.152-153).

Os modelos propostos nos dois exemplos acima retratam um tipo de especialização no qual os subconjuntos são **mutuamente exclusivos**, ou seja, dado um determinado indivíduo pertencente à Entidade Genérica, ele só poderá pertencer a uma única Entidade Específica. Um Médico ou é Residente ou é Efetivo. Um Meio de Transporte ou é Terrestre ou é Aéreo ou é Aquático. Nesses casos, diz-se que as especializações são **disjuntas**, pois define que, um indivíduo, poderá pertencer a no máximo uma entidade especializada a uma **Categoria**. (COUGO, 1997, p.112-114; SETZER, 2005, p.63-64)

Em outros casos, encontramos especialização não-mutuamente exclusiva, ou seja, um indivíduo da Entidade Genérica poderá pertencer a mais que uma entidade específica. Nesse caso dizemos que a especialização é **não-disjunta** ou **Inclusiva**, pois reconhece a possibilidade de um mesmo indivíduo realizar **papéis** distintos no contexto. (COUGO, 1997, p.112-114; SETZER, 2005, p.63-64)

Como o símbolo utilizado para representar a especialização é o mesmo, ou seja, um triângulo que vincula a entidade genérica as suas entidades específicas, precisamos complementá-lo de forma a distinguir as especializações **disjuntas (categoria)** das **não-disjuntas (papeis)**. Na bibliografia consultada encontramos as seguintes sugestões, ambas a serem escritas dentro do triângulo:

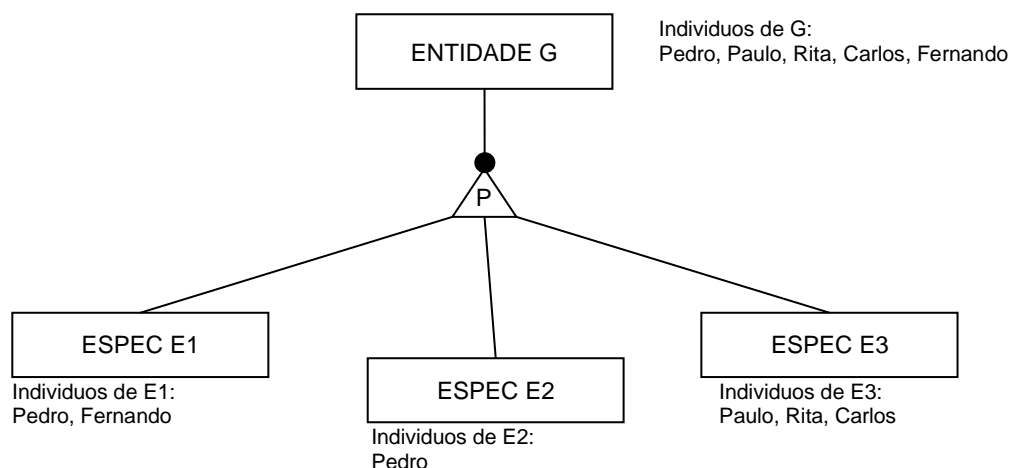
- Um **"X"** para especialização **disjunta** e um **"O"** para **não-disjunto**, ou
- Um **"C"** para especialização **disjunta** e um **"P"** para **não-disjunto**

Como as especializações **disjuntas** são as mais comuns, podemos convencionar não acrescentar nenhuma característica nesses casos e adotar um P para representar a especialização **não-disjunta**.

Além disso, precisamos distinguir a **totalidade** do vínculo entre a entidade genérica e as entidades específicas. Se cada entidade do nível superior precisa pertencer a um conjunto de entidades de nível inferior, diz-se que existe **generalização ou especialização total**. No entanto, se alguma entidade do nível superior pode **não** pertencer a um conjunto de entidades de nível inferior, diz-se que existe **generalização ou especialização parcial**. Podemos omitir qualquer simbologia adicional nos casos de generalização parcial por ser o mais comumente encontrado e podemos adotar um símbolo (**bolinha cheia**) somente para as totais. Nesse caso, podemos optar por colocar a bolinha na linha que liga a entidade genérica ao triângulo.



A figura 4.54 abaixo demonstra uma estrutura de generalização-especialização contendo os elementos gráficos adicionais: Uma bolinha cheia para indicar a parcialidade e a letra P para indicar que as entidades específicas são **papeis**. Nesse exemplo, vemos que na entidade genérica “G” existem 5 indivíduos. Um indivíduo (Pedro), aparece em duas entidades específicas. No entanto, que fique claro, de que todo indivíduo que apareça em uma entidade específica, obrigatoriamente deve existir na entidade genérica.


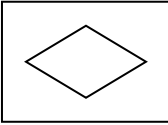


**FIGURA4.54** Estrutura de Generalização-Especialização não-disjunta

### 4.3.8 Notação a ser adotada

Diversos são as notações que podem ser utilizadas para representar um Modelo Conceitual. As mais difundidas são as de James Martin e as de Peter Chen. Nesse trabalho estaremos adotando a notação de Peter Chen com alguns elementos adicionais propostos ao longo desse material.

Elemento	Descrição	Símbolo	
Entidade forte	Retângulo simples		
Entidade Fraca	Retângulo duplo		
Atributo	Serão declarados a parte no dicionário de dados	Monovalorado	Sem símbolo
		Multivalorado	x{nome}y, onde x e y referem-se a cardinalidade mínima e máxima de ocorrências.
		Opcional	(nome)
		Composto	
		Determinante	sublinhado
Relacionamento	Um losango com uma seta dentro indicando o sentido de leitura do verbo		
Nome do relacionamento	Verbo no infinitivo Escrito do lado de fora do losango	COMPRAM 	

Elemento	Descrição	Símbolo
Cardinalidade do relacionamento	Mínimos e máximos expressos dentro de parêntesis	(X:Y) representando a cardinalidade máxima para cada entidade pertencente ao relacionamento.
Limites de cardinalidade	Serão alocados ao lado de cada entidade pertencente ao relacionamento	(X,Y) onde X representa a cardinalidade mínima e Y representa a cardinalidade máxima.
Totalidade do relacionamento	Será demonstrada pela cardinalidade mínima do relacionamento	
Generalização-especialização (Gen-Espec)	Um triângulo tendo no topo a entidade Genérica e na base as entidades Específicas	
Totalidade da Gen-Espec	Uma bolinha cheia ligando a entidade Genérica ao Triângulo somente para o caso de <b>totalidade</b> . Quando for parcial não será usado nenhum símbolo. No entanto, aqui a totalidade será expressa pela cardinalidade mínima.	●
Disjunção e Não-disjunção da Gen-Espec	Dentro do triângulo será colocado: C – para Gen-Espec disjunto P – para Gen-Espec não-disjunto	C P
Agregação	Um retângulo envolvendo o losango do relacionamento a ser agregado	
Relacionamento Independente	Nenhum símbolo será adotado	
Relacionamento Contigente	Será representado por duas barras na vertical cortando os relacionamentos com esta característica	
Relacionamento Mutuamente Exclusivo	Será representado por uma barra na vertical cortando os relacionamentos com esta característica	

**FIGURA4.55** notações utilizadas

### 4.3.9 Dicionário de Dados

O dicionário de dados é uma relação de todos os elementos de dados componentes do sistema.

Notação do Dicionário de Dados

SÍMBOLO	SIGNIFICADO
=	É composto de
+	E
()	Opcional (pode estar presente ou ausente)
x{}y	Iteração
[ ]	Escolha uma das opções alternativas
**	Comentário
@	Identificador (campo chave)

**FIGURA4.56** notações para dicionário de dados

Algumas Explicações:

= **Definições:** Símbolo que representa a definição de um elemento de dado. É lido como “é definido como” ou “é composto de” ou simplesmente “significa”.

Ex.: A = B + C , interpretado como A é composto de B mais C ou ainda A é definido como B e C.

() **Elemento de dados opcionais:** É o elemento que pode ou não estar presente como um componente de um elemento de dados composto.

Ex.: Endereço-cliente = (endereço-remessa) + (endereço-cobrança). Pode ser lido como sendo, um Endereço-cliente pode ter endereço-remessa, ou endereço-cobrança ou ambos ou nenhum.

{ } **Iteração:** Indica a ocorrência repetida de um componente de um elemento de dados, onde X indica a ocorrência mínima e Y a ocorrência máxima.

Ex.: Pedido = nome-cliente + endereço-remessa + 1{item}10. Significa que um Pedido deve ter o nome do cliente, o endereço de remessa e no mínimo 1 e no máximo 10 itens. Pode-se ter as seguintes combinações para os limites da iteração:

1. X{ }Y Informando limite inferior e superior
2. { }Y Informando somente limite superior
3. X{ } Informando somente limite inferior
4. { } Não informando nenhum limite

[ | ] **Seleção:** Indica que o elemento de dados consiste em exatamente uma escolha de um conjunto de opções alternativas. As opções são delimitadas por colchetes “[” e “]” e separadas pelo caracter de barra vertical.

Ex.: Sexo [masculino | feminino]

Exemplo1:

**NOME** = TITULO-CORTESIA + PRIMEIRO-NOME + (NOME-INTERMEDIÁRIO) + ULTIMO-NOME

**TITULO-CORTESIA** = [ Sr. | Srta. | Sra. | Sras. | Dr. | Professor ]

**PRIMEIRO-NOME** = {CARACTER-VALIDO}

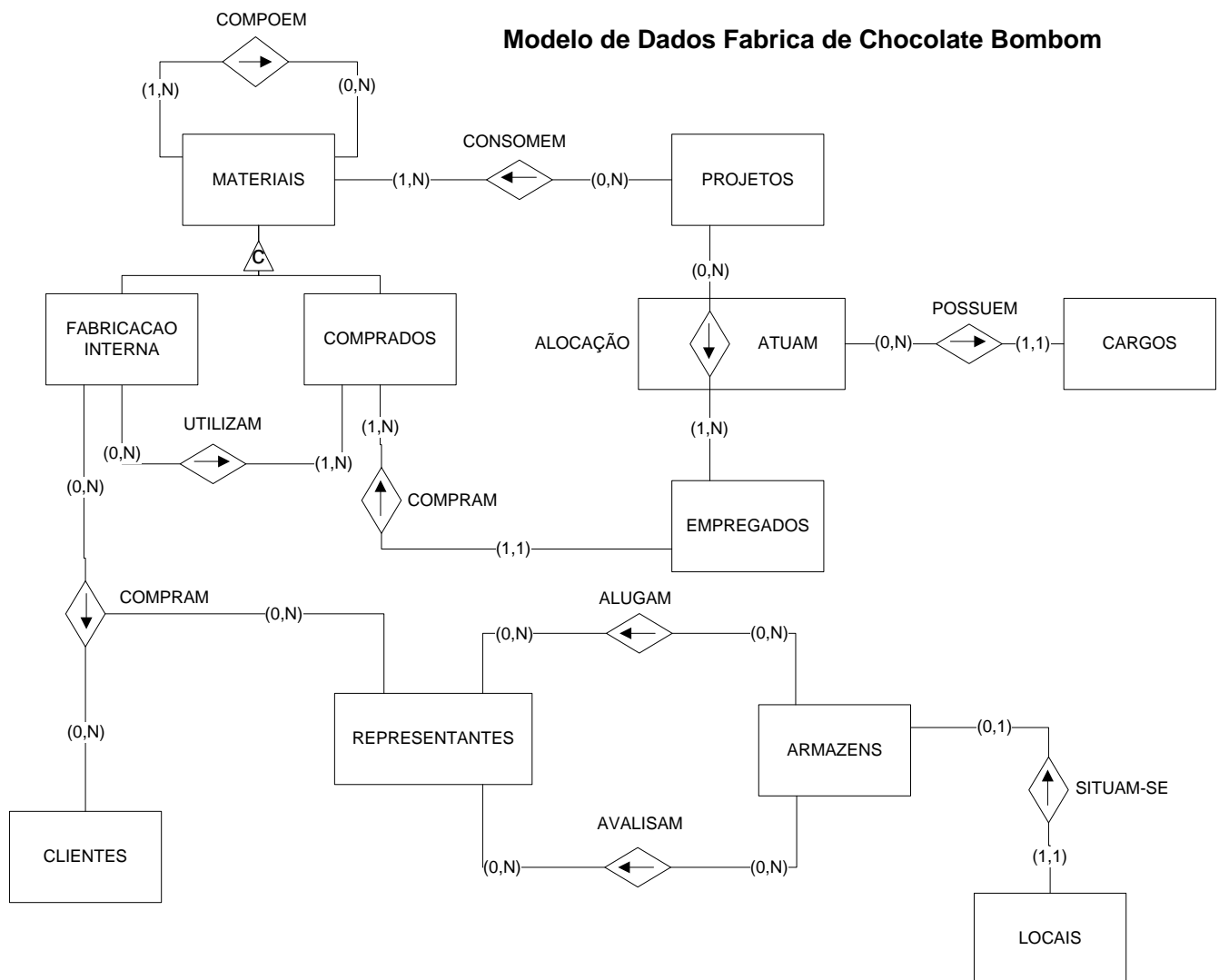
**NOME-INTERMEDIARIO** = {CARACTER-VALIDO}

**ULTIMO-NOME** = {CARACTER-VALIDO}

**CARACTER-VALIDO** = [ A – Z | a – z | 0 – 9 | ‘ | - | | ]

#### 4.3.10 Exemplo de Modelo Conceitual – Entidades-Relacionamentos.

O modelo de dados abaixo tem por objetivo demonstrar o produto do processo de modelagem de dados a nível conceitual tomando por base o modelo entidade-relacionamento. Este é um cenário hipotético, ou seja, não reflete nenhuma situação real justamente para garantir que teríamos condições de modelar diversos elementos teóricos abordados ao longo do capítulo.



#### DICIONÁRIO DE DADOS

MATERIAIS = codigom + nome + quantidade  
 FABRICAÇÃO INTERNA = codigom + modelo + precocusto  
 COMPRADOS = codigom + precocompra + dataultima compra  
 PROJETOS = codigop + nome + dataelaboração + objetivo  
 CLIENTES = codigoc + razaosocial + ENDEREÇO + (e\_mail) + 1{telefones}  
 REPRESENTANTES = codigor + razãosocial + endereço + (e\_mail) + 1{telefones}  
 ARMAZENS = codigoa + telefone  
 LOCAIS = codigol + bairro + cidade  
 EMPREGADOS = codigoe + nome + dataadmissao  
 ALOCAÇÃO = datainicio + datafim  
 CARGOS = codigoc2 + nome + salario  
 ENDERECO=logradouro + numero + (complemento) + bairro + cidade + UF + CEP

**FIGURA4.57** Exemplo de Modelo Conceitual Entidade-Relacionamento

## 4.4 MODELAGEM LÓGICA DE DADOS

### 4.4.1 Introdução

Uma vez obtido o Modelo Conceitual conforme descrito na seção 3.4, já temos condições de partir para um modelo de implementação, ou seja, para um Modelo Lógico de Dados (MLD). Um Modelo Lógico é aquele em que os objetos, características e relacionamentos são representados de acordo com as regras de implementação a partir das restrições impostas por uma tecnologia. A obtenção do modelo lógico se dá pela tradução dos elementos presentes no Modelo Conceitual (COUGO, 1997).

Tal conversão baseia-se num conjunto de regras, que são aplicadas sobre o Modelo Conceitual. Cada regra é direcionada a atender a determinados requisitos associados à tecnologia na qual o modelo será construído. Esta tecnologia pode ser Relacional, Rede-CodasyI, Orientada a Objetos (COUGO, 1997).

Nesse caso, adota-se a Tecnologia Relacional e mais especificamente, Relacional Normalizada e por esta razão, a conversão do MCD para o MLD deverá usar Regras para Bancos de Dados Relacionais. O produto gerado pela tradução será um Diagrama Relacional.

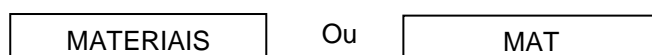
Modelos Lógicos estão associados a fase de Projeto de um projeto de software.

Com isso, pode-se obter um MLD a partir de um MCD seguindo os seguintes passos (COUGO, 1997):

- Obter o MCD
- Definir o tipo de Implementação (Relacional, Rede-codasyI, OO).
- Aplicar as regras de derivação
- Adaptar modelo às necessidades.

A simbologia utilizada no Diagrama Relacional é a seguinte:

- **Tabelas (tradução das Entidades)** → são representadas por um Retângulo, com o nome da tabela escrito dentro do retângulo. Observe que por se tratar de um objeto a ser compreendido e armazenado dentro de um SGBD, aceita-se abreviaturas.



- **Colunas (tradução dos atributos)** → são representadas dentro do retângulo da tabela, com suas respectivas características. Ex.: Tabela MATERIAIS representando a entidade MATERIAIS e as colunas CODIGO, NOME e QUANTIDADE representando atributos com mesma nomenclatura.

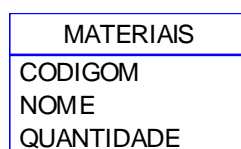






FIGURA4.58 Tabelas e suas colunas

- **Ligações (tradução dos relacionamentos)** → são representadas conforme a tabela abaixo. As regras para análise e tradução dos relacionamentos serão estudadas em detalhes na seção 4.4.3 – Mapeamento do MCD para o MLD.

Cardinalidade	Ligação
0,1	
1,1	
0,N	
1,N	

**FIGURA4.59** Notação da cardinalidade para modelo lógico (pé-de-galinha)

#### 4.4.2 Propriedades do Modelo Relacional Normalizado

Para se garantir a perfeita adequação do Modelo Conceitual sem com isso perder as características modeladas, precisamos respeitá-las e considerar as seguintes regras do Modelo Relacional.

- Cada célula pode ser vazia ou conter no Máximo 1 valor.
- A ordem das linhas é irrelevante do ponto de visto do usuário.
- Não há duas linhas iguais
- Cada coluna tem um nome
- Duas colunas distintas devem ter nomes diferentes.
- A partir do nome, a ordem das colunas é irrelevante.
- Cada relação tem um nome próprio diferente dos demais nomes de relação.
- Os valores de uma coluna são retirados de um conjunto denominado "Domínio".
- Duas ou mais colunas podem ter o mesmo domínio

#### 4.4.3 Mapeamento do MCD para o MLD (ELMASRI/NAVATHE, 2005, p.139-141; SILBERSCHATZ, 2006, p.161-165)

**4.4.3.1 Entidade Regulares** → Para cada Entidade Forte *E* no MCD será criada uma tabela *T* no MLD. Incluir todos os atributos simples e os atributos compostos já decompostos como colunas em *T*. Por exemplo, tomando a entidade CLIENTES e seus respectivos atributos presentes no MCD da figura 4.57 teríamos a seguinte tradução:

CLIENTES		
<u>CODIGOC</u>	<pk>	not null
RAZAO_SOCIAL		not null
LOGRADOURO		not null
NUMERO		not null
COMPLEMENTO		null
BAIRRO		not null
CIDADE		not null
ESTADO		not null
CEP		not null
E_MAIL		null

**FIGURA4.60** Tabela derivada de entidade regular

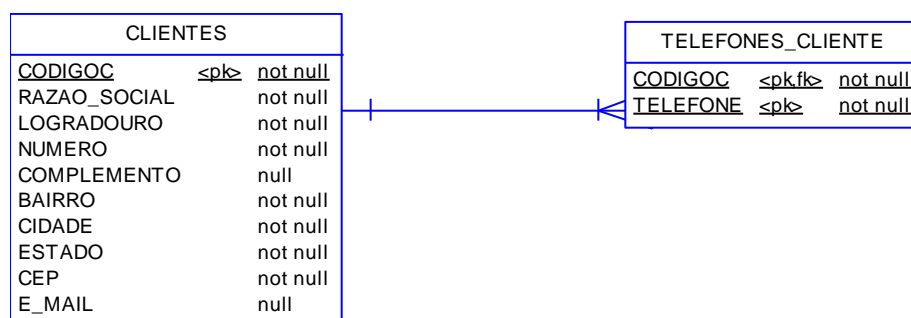
Observando-se esta figura percebemos as seguintes características: A entidade CLIENTES deu origem a tabela denominada CLIENTES. Cada atributo simples (codigoc, razão social, e\_mail) da entidade CLIENTES tornou-se um atributo simples na tabela CLIENTES. O atributo composto Endereço, foi desmembrado em seus vários

componentes (logradouro, numero, complemento, bairro, cidade, estado e CEP). O atributo telefone aparecerá em outra tabela conforme será definido no item 4.4.3.4.

**4.4.3.2 Atributos Determinantes** → O Atributo Determinante de uma relação será transformado na Chave Primária da Tabela. Se o atributo determinante for composto na forma  $A=(a_1, a_2, a_3)$ , cada atributo que o compõe será parte da Chave Primária da tabela. Ao observarmos a tabela CLIENTES da figura 4.60, percebe-se que o atributo determinante “codigoc” tornou-se a Chave Primária da tabela.

**4.4.3.3 Atributos Monovalorados** → Todos os atributos simples, monovalorados serão transformados em colunas nas tabelas. De forma semelhante, percebe-se que todo atributo monovalorado presente na entidade CLIENTES (codigoc, razão social e e-mail) tornou-se uma coluna na tabela clientes.

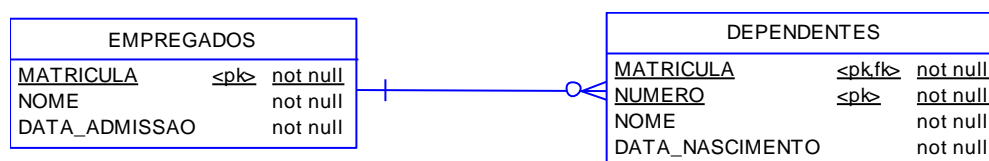
**4.4.3.4 Atributos Multivalorados** → Para cada atributo multivalorado, criar uma nova tabela. Nessa nova tabela o atributo não será multivalorado. Também deverá estar nessa nova tabela a PK da tabela origem de onde o atributo multivalorado saiu. Essa PK, junto com o atributo que deixou de ser multivalorado formarão a PK da nova tabela. Para preservar a multivaloração do atributo, a nova tabela terá uma cardinalidade N para 1 com a tabela origem. Na entidade CLIENTES havia o atributo 1{telefones}, um multivalorado obrigatório. A tradução desse atributo produzirá o modelo relacional da figura 4.61. Nesse caso, o atributo telefone deu origem a uma nova tabela (TELEFONES\_CLIENTE) contendo o numero do telefone e a Chave Primária da tabela ao qual ele está vinculado. As duas colunas (codigoc e telefone) formarão a Chave Primária dessa nova tabela.



**FIGURA4.61** Tradução de Atributo Multivalorado

**4.4.3.5 Atributos Compostos** → Cada atributo composto deverá ser desmembrado em atributos simples. Conforme observa-se na figura 4.60, o atributo composto endereço foi desmembrado nos seus vários componentes (logradouro, numero, complemento, bairro, cidade, estado, CEP).

**4.4.3.6 Entidades Fracas** → Para cada entidade fraca será criada uma tabela. Incluir todos os atributos simples e os atributos compostos já decompostos como colunas nessa tabela. Incluir também como Chave Estrangeira a Chave Primária da tabela que corresponde à Entidade Forte do relacionamento. Esta Chave Estrangeira, junto com a chave parcial da entidade fraca formação a Chave Primária dessa tabela. Se a Entidade Forte do relacionamento de uma determinada Entidade Fraca for também uma Entidade Fraca em outro relacionamento, então a Chave Primária desse primeiro relacionamento deverá ser identificada para então derivar esta Chave para o segundo relacionamento. Tomaremos como exemplo a situação proposta na seção 4.3.7.7 – Entidade Fraca, figura 4.48. A figura 4.62 abaixo expressa a solução para derivação de entidades fracas.

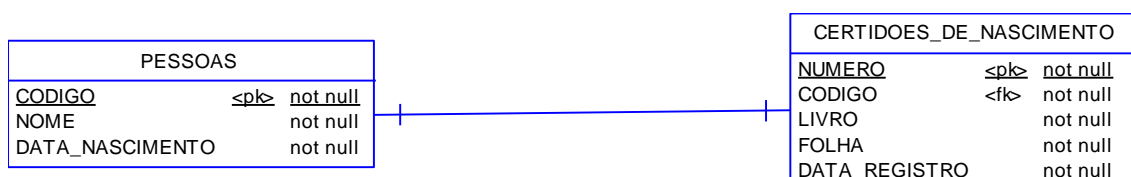


**FIGURA4.62** Tradução de Entidade Fraca

**4.4.3.7 Relacionamentos Binários** → Os relacionamentos serão traduzidos conforme a cardinalidade dos mesmos.

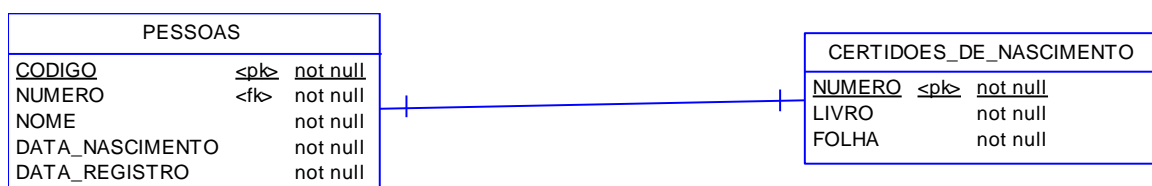
**Relacionamento de Cardinalidade 1:1:** Para ilustrar esse relacionamento usaremos a situação proposta na figura 4.43. Nesse tipo de cardinalidade, podemos adotar 3 soluções possíveis:

- Transposição de Chave Estrangeira → A coluna Chave Primária de uma das tabelas pertencentes ao relacionamento será transposta para a outra tabela onde passará a ser tratada como chave estrangeira.



**FIGURA4.63** Tradução de Relacionamento 1:1 – Transposição de Chave

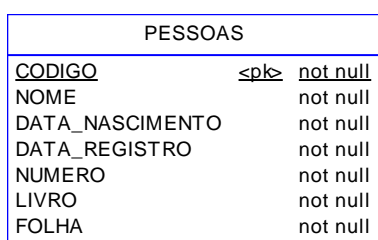
Na figura 4.63 acima, o relacionamento foi traduzido transpondo-se a Chave Primária da tabela PESSOAS (código) para a tabela CERTIDOES\_DE\_OBITO, onde será Chave Estrangeira. Além disso, o atributo de relacionamento (se houver) também poderá ser alocado em CERTIDOES\_DE\_OBITO



**FIGURA4.64** Tradução de Relacionamento 1:1 – Transposição de Chave (outra direção)

De forma semelhante, a figura 4.64 expressa a tradução do relacionamento transpondo a Chave Primária da tabela CERTIDOES\_DE\_OBITO (numero) para a tabela PESSOAS, onde será Chave Estrangeira. Além disso, o atributo de relacionamento (se houver) também poderá ser alocado em PESSOAS.

- Unificar em uma única tabela → Nessa opção, as entidades participantes do relacionamento são fundidas em uma única tabela. Esta opção pode ser apropriada quando o relacionamento é total nos dois sentidos. Caso um dos lados do relacionamento não seja total, os atributos referentes a esta entidade deverão ser nulos na tabela oriunda da fusão.



**FIGURA4.65** Tradução de Relacionamento 1:1 – Fusão na tabela da esquerda

Na figura 4.65 acima, a tradução do relacionamento deu-se pela eliminação da tabela da direita e a transposição dos atributos da mesma para a tabela da esquerda. Ao olhar a tabela PESSOAS, vemos que as colunas código, nome e data\_nascimento pertenciam a entidade PESSOAS, a



coluna data\_registro pertencia ao relacionamento e as colunas numero, livro e folha pertencem a entidade CERTIDOES\_DE\_OBITO que já não existe mais.

CERTIDOES_DE_NASCIMENTO		
<u>NUMERO</u>	<pk>	not null
LIVRO		not null
FOLHA		not null
CODIGO		not null
NOME		not null
DATA_DE_NASCIMENTO		not null
DATA_REGISTRO		not null

**FIGURA4.66** Tradução de Relacionamento 1:1 – Fusão na tabela da direita

De forma semelhante ao realizada na figura 4.65, a tradução do relacionamento também poderá ser feita pela fusão na tabela da esquerda do relacionamento. Ao olhar a tabela CERTIDOES\_DE\_OBITO, vemos que as colunas numero, livro e folha pertenciam a entidade CERTIDOES\_DE\_OBITO, as colunas código, nome e data\_nascimento pertenciam a entidade PESSOAS e a coluna data\_registro pertencia ao relacionamento.

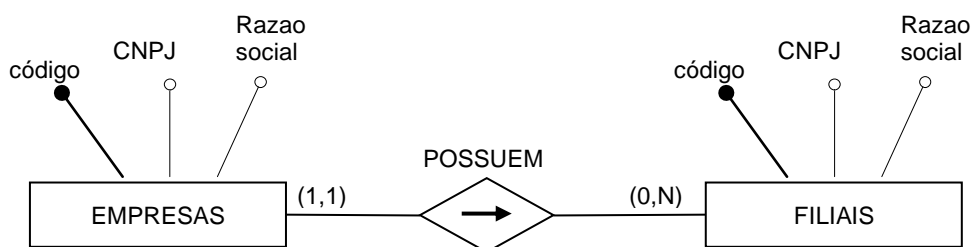
- Tabela de Relacionamento → a terceira solução será criar uma nova tabela para abrigar o relacionamento. Esta solução é mais indicada para relacionamento de cardinalidade M:N.



**FIGURA4.67** Tradução de Relacionamento 1:1 – criar tabela de relacionamento

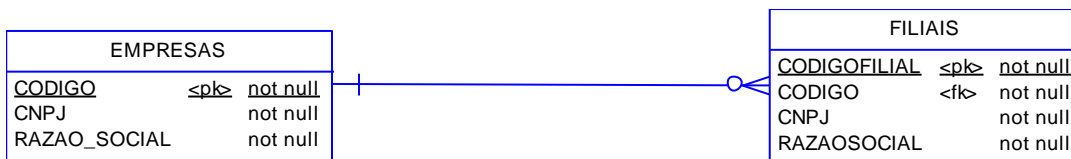
Aqui surge a tabela REGISTRO que representa o relacionamento entre PESSOAS e CERTIDOES\_DE\_OBITO. Também nessa nova tabela será abrigado o atributo de relacionamento (data\_registro) se houver.

- **Relacionamento de Cardinalidade 1:N:** Para ilustrar esse relacionamento usaremos a situação proposta na figura 4.32 com os respectivos atributos conforme demonstramos abaixo. Nesse tipo de cardinalidade podemos adotar 2 soluções:



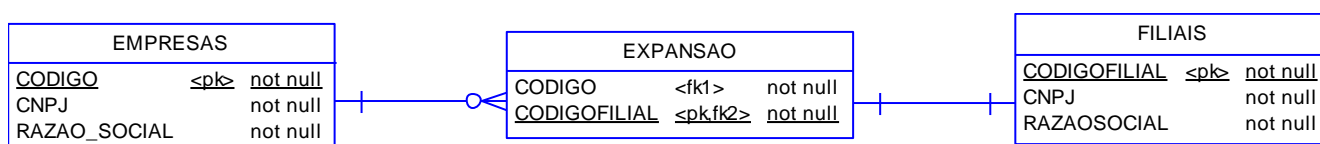
**FIGURA4.68** relacionamento 1:N

- Transposição de Chave Estrangeira → inserir na entidade que possui a cardinalidade N a Chave Primária da Entidade que possui cardinalidade 1 como Chave Estrangeira.



**FIGURA4.69** Tradução de Relacionamento 1:N – transposição de chave

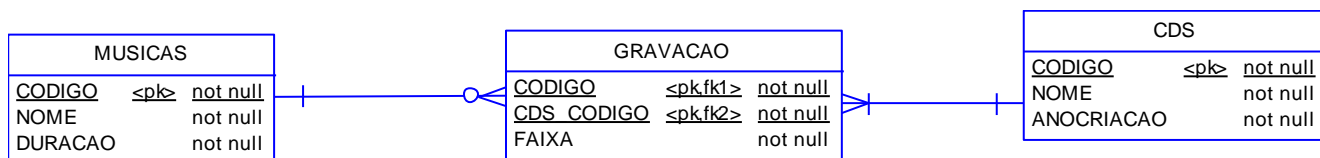
- Tabela de Relacionamento → Nessa solução cria-se uma nova tabela para abrigar o relacionamento, cujas colunas seriam duas Chaves Estrangeiras oriundas da Chave Primária das duas entidades participantes do relacionamento. A Chave Primária dessa nova tabela será a Chave Primária da entidade de cardinalidade N. Esta pode ser uma alternativa quando poucos elementos da entidade de cardinalidade N participam do relacionamento evitando-se assim excesso de valores nulos na Chave Estrangeira.



**FIGURA4.70** Tradução de Relacionamento 1:N – criação de tabela de relacionamento

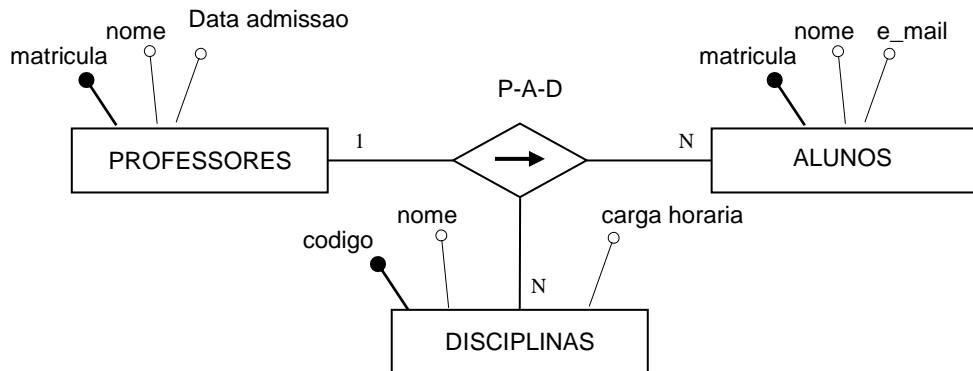
• **Relacionamento de Cardinalidade M:N:** Nesse tipo de cardinalidade temos somente uma solução:

- Tabela de Relacionamento → Nessa solução única cria-se uma nova tabela para abrigar o relacionamento, cujas colunas seriam duas Chaves Estrangeiras oriundas da Chave Primária das duas entidades participantes do relacionamento. A Chave Primária dessa nova tabela será formada por todas as Chaves Estrangeiras oriundas da Chave Primária das tabelas origem. Será utilizada a figura 4.47 para ilustrar esta situação.



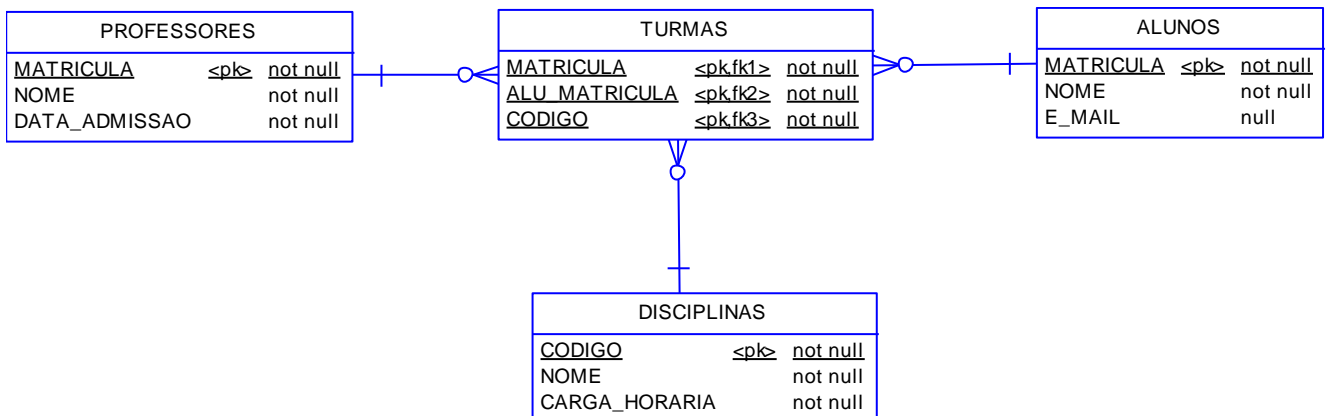
**FIGURA4.71** Tradução de Relacionamento 1:N – criação de tabela de relacionamento

**4.4.3.8 Relacionamentos enários** → Os relacionamentos serão traduzidos conforme a cardinalidade dos mesmos. Para ilustrar esse relacionamento usaremos a situação proposta na figura 4.30 com os respectivos atributos conforme demonstramos abaixo.



**FIGURA4.72** relacionamento ternário

Solução para traduzir relacionamento ternário. Observe que para dar maior semântica ao modelo lógico, renomeamos o relacionamento P-A-D pelo termo TURMAS. A solução seria exatamente a mesma se a multiplicidade fosse 1:1:1 ou 1:1:N ou M:N:N (SETZER, 2005, p.142).



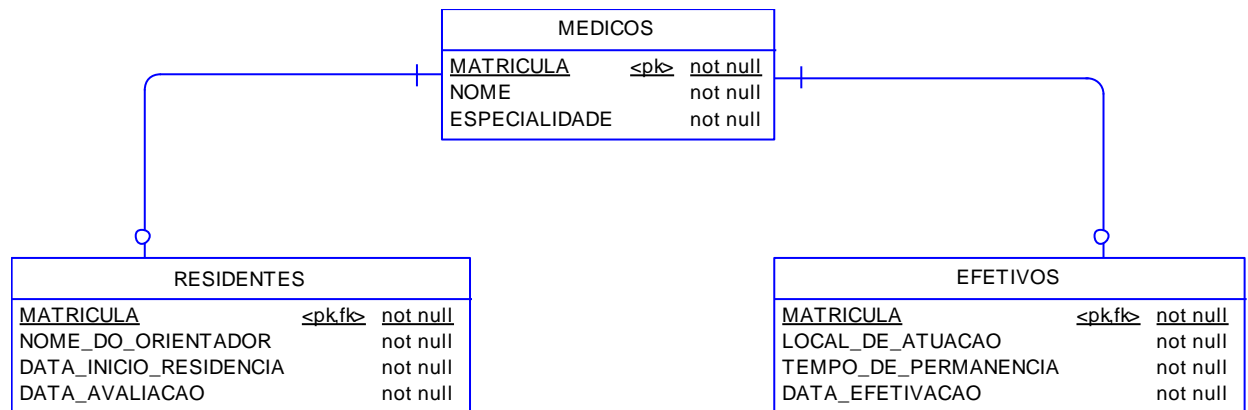
**FIGURA4.73** Tradução de Relacionamento ternário



Também será adotada a mesma solução, ou seja, gerar uma tabela para abrigar o relacionamento caso este seja de grau maior que 3 (SETZER, 2005, P.142).

**4.4.3.9 Generalização-Especialização** → A estrutura de Generalização-Especialização suporta 3 soluções de tradução (COUGO, 1997, p.237-246). Tomaremos como exemplo a figura 4.52.

- 1ª - Criar uma tabela para cada entidade generalizada e uma tabela para cada entidade especializada → nessa solução a transposição do MCD para o MLD é praticamente direta. A Chave Primária da tabela que representa a entidade genérica será a Chave Primária de cada tabela que representa cada entidade específica. Dessa maneira está se criando um “relacionamento artificial” entre a tabela genérica e as específicas de cardinalidade 1:1.



**FIGURA4.74** Tradução da Generalização-Especialização – uma tabela para cada entidade

- 2ª - Criar somente a tabela para a entidade generalizada e migrar todos os atributos e relacionamentos das entidades especializadas para ela → Nesse caso, existirá somente a tabela representando a entidade genérica e todos os atributos, dela e das entidades específicas que não estão sendo representadas por tabelas. Também os relacionamentos das entidades específicas deverão ser transferidos para a tabela gerada. Nesse caso, *os atributos das entidades específicas deverão ser opcionais (nulos) na tabela que representa a entidade genérica.*

MEDICOS		
<u>MATRICULA</u>	<pk>	not null
NOME		not null
ESPECIALIDADE		not null
NOME_DO_ORIENTADOR		null
DATA_INICIO_RESIDENCIA		null
DATA_AVALIACAO		null
LOCAL_DE_ATUACAO		null
TEMPO_DE_PERMANENCIA		null
DATA_EFETIVACAO		null

**FIGURA4.75** Tradução da Generalização-Especialização – fundir atributos na tabela genérica

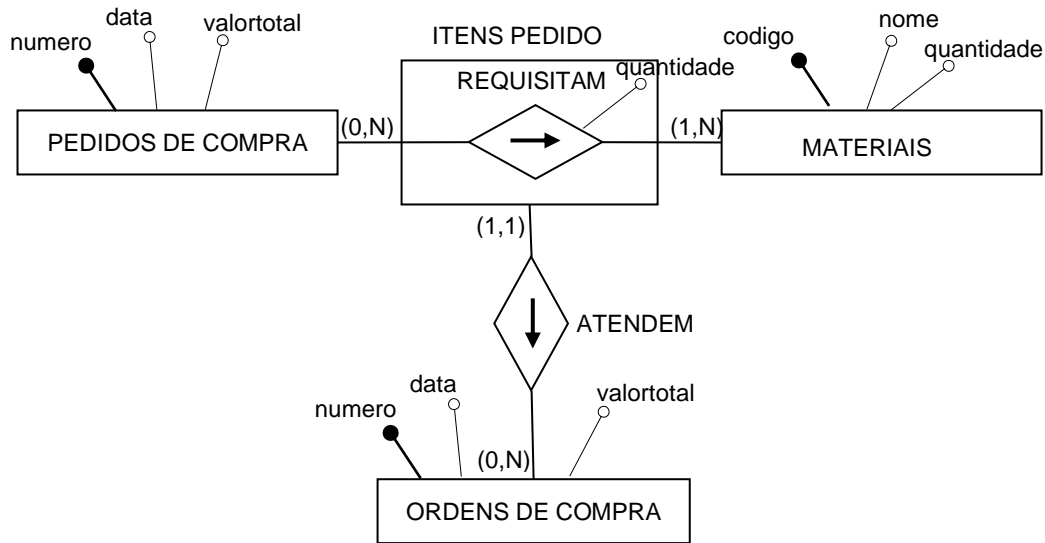
- 3ª - Criar somente tabelas para as entidades especializadas e migrar todos os atributos e relacionamentos da entidade generalizada para cada uma dessas tabelas → Esta solução evita os problemas de nulidade causada pela solução anterior. Nessa solução, fazemos o inverso da solução anterior, pois criamos uma tabela para cada entidade específica e migramos os atributos e relacionamentos da entidade genérica para cada tabela representativa de cada entidade específica criada.

RESIDENTES		
<u>MATRICULA</u>	<pk>	not null
NOME		not null
ESPECIALIDADE		not null
NOME_DO_ORIENTADOR		not null
DATA_INICIO_RESIDENCIA		not null
DATA_AVALIACAO		not null

EFETIVOS		
<u>MATRICULA</u>	<pk>	not null
NOME		not null
ESPECIALIDADE		not null
LOCAL_DE_ATUACAO		not null
TEMPO_DE_PERMANENCIA		not null
DATA_EFETIVACAO		not null

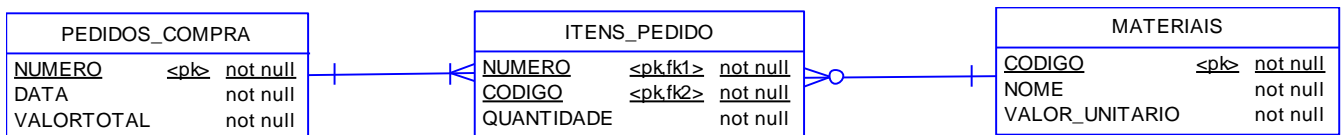
**FIGURA4.76** Tradução da Generalização-Especialização – Desmembrar somente nas específicas

**4.4.3.10 Agregação** → A tradução da Agregação se dará em duas partes: primeiro será tratado o aspecto relativo ao relacionamento (aquele que dará vida ao agregado), e em segundo lugar faremos a tradução do aspecto relativo a entidade, ou seja, ao agregado e seus vínculos com os demais elementos do modelo (COUGO, 1997, p.237-246). Para ilustrar esse relacionamento usaremos a situação proposta na figura 4.49 com os respectivos atributos conforme demonstramos abaixo



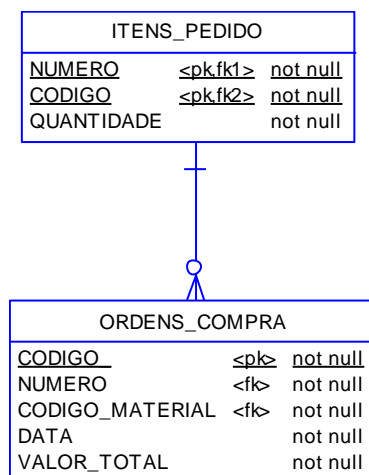
**FIGURA4.77** Agregação

- Aspecto relativo ao relacionamento → será analisado inicialmente o relacionamento e é claro a cardinalidade que compõe o agregado. A tradução dessa parte usará os mesmos critérios daquele adotado para qualquer relacionamento visto anteriormente. No caso, por tratar-se de um relacionamento M:N, traduz-se o relacionamento criando-se uma tabela para representar a referência cruzada das duas entidades, cuja Chave Primária será formada pelas Chaves Estrangeiras oriundas das tabelas participantes do relacionamento.



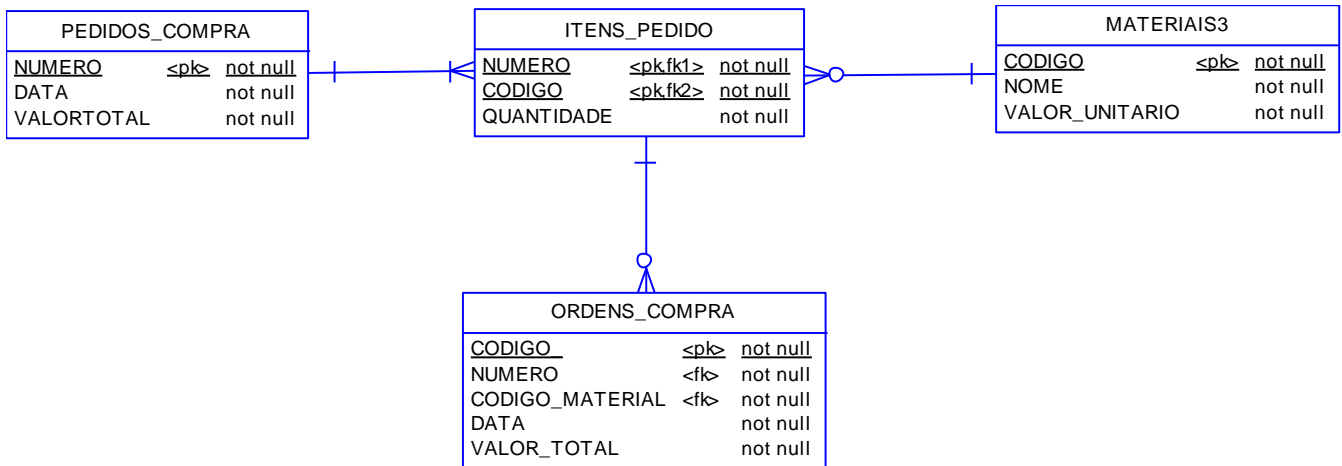
**FIGURA4.78** Tradução da Agregação – aspecto relativo ao relacionamento

- Aspecto relativo à Entidade → Após a derivação do relacionamento visto no item anterior, o agregado passará a ser representado pela tabela gerada da derivação do relacionamento. Nesse caso, analisam-se as demais partes do modelo conforme as regras de relacionamento estudadas anteriormente, sendo que agora, o agregado é reconhecido como uma tabela com identidade e para a qual poderão fluir relacionamentos ou de qual poderão sair relacionamentos.



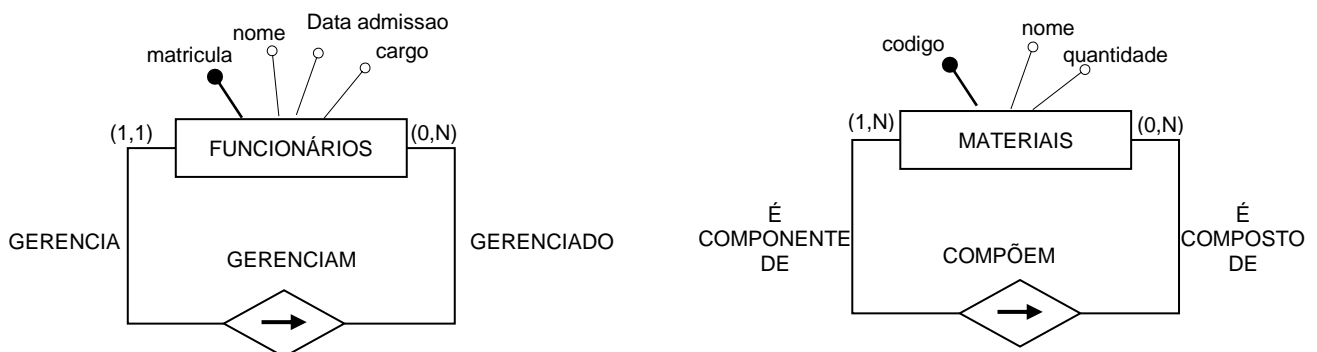
**FIGURA4.79** Tradução da Agregação – aspecto relativo a entidade

A visão completa do Agregado traduzido é conforme demonstrado abaixo.



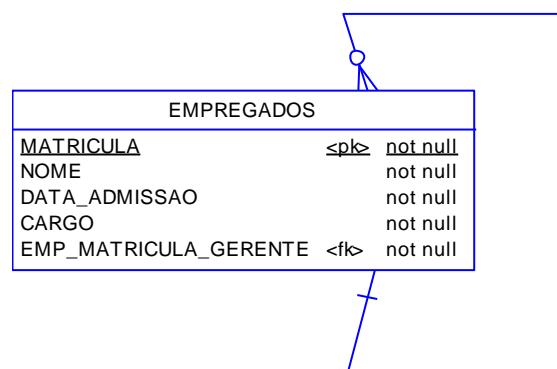
**FIGURA4.80** Tradução da Agregação – modelo completo

**4.4.3.11 Auto-relacionamento** → A tradução do autor relacionamento deverá considerar a cardinalidade do mesmo e para tanto existem duas soluções (COUGO, 1997, p.232-234; SETZER, 2005, p.140-141). Para ilustrar esse relacionamento usaremos a situação proposta na figura 4.19 com os respectivos atributos conforme demonstramos abaixo



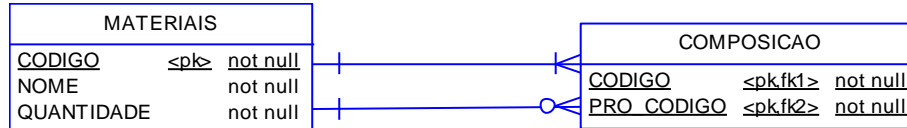
**FIGURA4.81** Exemplos de auto-relacionamento

- Para auto-relacionamento de cardinalidade 1:1 e 1:N, migrar a Chave Primária da tabela para ela mesma como chave estrangeira.



**FIGURA4.82** Tradução do Auto-Relacionamento cardinalidade 1:1 ou 1:N

- Para auto-relacionamento de cardinalidade M:N, criar uma nova tabela para comportar a referência cruzada entre os indivíduos da tabela original. A Chave Primária da tabela original deverá ser migrada duas vezes para a nova tabela para que se possa construir os pares.



**FIGURA4.83** Tradução do Auto-Relacionamento cardinalidade M:N

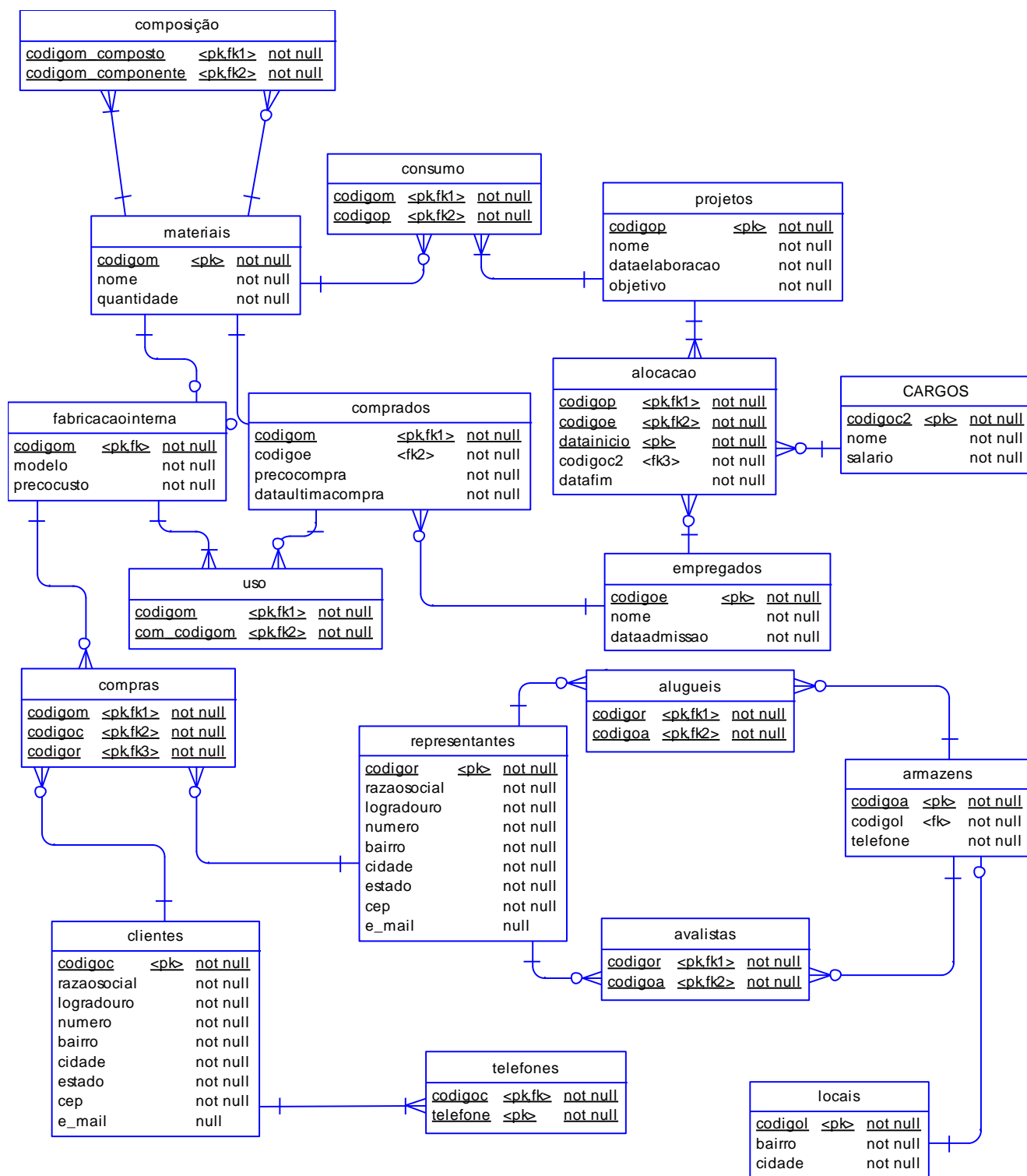
**4.4.3.12 Valores Nulos para a PK** → em algumas situações pode existir uma Foreign Key que suporte valores nulos. Tais situações estão descritas abaixo.

**A(1,1) B(0,N)** → Na situação acima, a FK nunca será nula pois cada elemento de B está associado a no mínimo 1 elemento de A, ou seja, adotando a nomenclatura relacional, “cada linha existente na tabela B deverá estar obrigatoriamente vinculada a uma linha de A, e como este vínculo se dá pelo preenchimento da coluna FK, então ela sempre terá que ter valor”.

**A(1,1) B(1,N)** → Semelhante ao caso anterior, a cardinalidade mínima de B com A é um, ou seja, toda linha de B deve estar associada através da FK a uma linha de A, fazendo-a ser obrigatória.

**A(0,1) B(0,N) ou B(1,N)** → Neste caso, a FK poderá ser nula, pois a cardinalidade mínima de B para A é 0, o que significa que pode existir alguma linha de B que não está ligada a nenhuma linha de A

**4.4.4 Exemplo de Modelo Conceitual – Entidades-Relacionamentos traduzido para o Modelo Lógico Relacional.** A figura 4.83 abaixo demonstra o MLD da tradução do MCD da figura 4.57.



**FIGURA4.83** Modelo Lógico obtido da tradução do MCD da figura 4.57.



## 4.5 NORMALIZAÇÃO

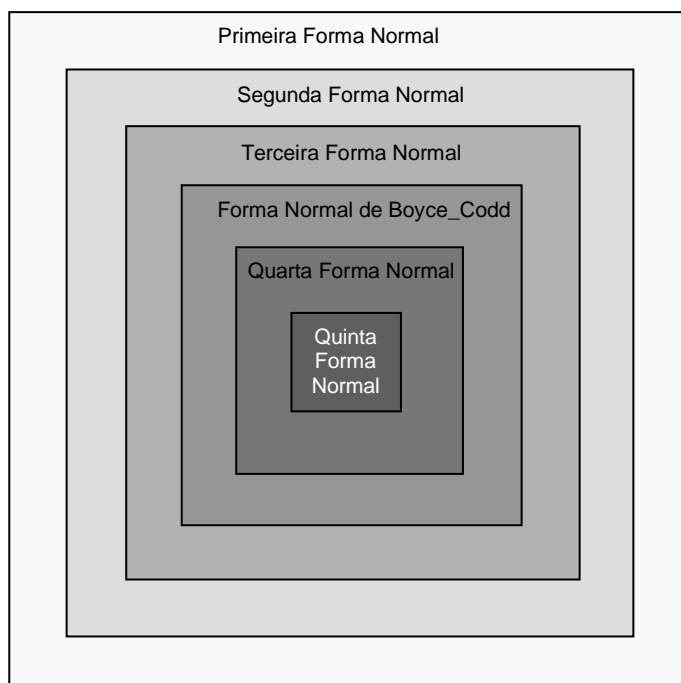
A Normalização é um processo surgido com o aperfeiçoamento do Sistema Relacional na década de 70, e tem por objetivo, simplificar as tabelas bem como, a obtenção do Modelo Lógico a partir do Modelo Conceitual. Este processo de simplificação segue um conjunto de regras, denominadas **Formais Normais**. O processo de adequação das tabelas a estas regras é chamado de Normalização.

A normalização busca evitar que dentro de uma tabela exista **mistura de assuntos** que podem provocar redundâncias desnecessárias nos dados (MACHADO, 2004, p. 181). Durante o processo de normalização, analisa-se a relação para determinar algum tipo de interdependência entre os atributos e nesse caso, transformar tais atributos em novas tabelas (TEOREY, LIGHTSTONE, NADEAU, 2007, p.111). Dessa maneira, alcançamos tabelas purificadas e imunes a anomalias de atualização que por sua vez podem causar problemas tais como: grupos repetitivos, dependências parciais de chave concatenada, redundâncias de dados desnecessária, etc (MACHADO, ABREU, 1996, p.165).

Para realizar a normalização é necessário que cada tabela a ser analisada tenha uma Chave Primária que permite identificar as demais colunas (MACHADO, 2004, p.182).

Atualmente são 6 as *formas normais*. Cada forma normal representa um conjunto cada vez mais rigoroso de regras, e que quanto mais alta a forma normal ao qual o projeto corresponda, melhor este será (HARRINGTON, 2002, p.85). No entanto, se você conseguir colocar suas tabelas até a 3ª Forma Normal, já terá resolvido a maioria dos problemas de projeto.

A figura 4.84 abaixo expressa estas formas normais. Nela é possível perceber que se uma tabela estiver em uma forma normal mais interna (mais alta) estará também em todas as formas normais abaixo dela (mais externa).



**FIGURA4.84** Formais Normais (Harrington, 2002, p.85)

### 4.5.1 Anomalias de Modificação

Considere estrutura de dados proposta na figura 4.85 abaixo. Nela é possível perceber que estamos abrigando dentro de uma única estrutura elementos de assuntos distintos (Pedidos, Fornecedores e Produtos) ainda que

relacionados. Uma estrutura dessa maneira está sujeita de um conjunto de **anomalias de modificação**, já que, se tentarmos **eliminar** o pedido numero “5”, estaremos eliminando por consequência todos os dados relativos ao Fornecedor “computer”. Se quisermos **incluir** um novo produto, não temos condições de fazê-lo sem que este esteja atrelado a um pedido. E por fim, se houver alguma **alteração** no endereço de um determinado Fornecedor, temos que atualizar em diversos registros.

No entanto, se for possível desmembrar os elementos de cada assunto em estruturas separadas, tais problemas serão amenizados, ou até mesmo resolvidos.

PEDIDO								
Num. Pedido	Data Emissão	Nome Fornec.	CNPJ	Endereço	Produtos			
					Cód. Produto	Nome Produto	Qtde.	Preço Unit.
003	20-jan	Software	12345-78	Av. Lapa 777, Lapa RJ 20000	033A	DOS	04	130,00
					002M	COREL	01	499,00
					145J	ABC	13	256,00
004	27-Jan	Brasoft	23456-89	R. Itu 49, Itu – SP 11000	002M	COREL	02	450,00
					083P	ZAPT	10	85,00
005	27-Jan	Computer	34567-00	R. Vitoria 12, Vitoria ES – 30000	033A	DOS	50	110,00
					145J	ABC	50	110,00
006	14-Mar	Brasoft	23456-89	R. Itu 49, Itu – SP 11000	029K	WIN	15	200,00
					083P	ZAPT	10	87,00

**FIGURA4.85** Estrutura Desnormalizada

## 4.5.2 Formas Normais

**Primeira Forma Normal (1ªFN):** Qualquer tabela que atenda a definição de uma **relação** está na Primeira Forma Normal. Uma tabela, para que possa representar uma relação deve ter nas suas células valores únicos, ou seja, não permitir colunas ou grupos de repetição ou arrays como valores. Todos os valores de uma coluna devem ser do mesmo tipo. Não importa a ordem das colunas nem das linhas contanto que cada coluna tenha um nome exclusivo e que as linhas não sejam idênticas (KROENKE, 1999, p.89). Por isso, diz-se que a 1ªFN faz parte da definição formal de uma tabela no modelo relacional (ELMASRI, NAVATHE, 2005, p.224).

Para Cougo, (1997, p.185), uma tabela está na primeira forma normal se:

- Está integrado por tabelas: Se estamos modelando para modelos lógicos relacionais devemos respeitar o conceito básico relativo ao objeto tabela, ou seja, uma estrutura bidimensional formado por linhas e colunas;
- As linhas da tabela são unívocas: Deve-se obrigatoriamente definir identificador (es) unívoco(s) ou chave(s) primária(s) da tabela de forma a garantir a diferenciação entre duas linhas de uma tabela;
- As linhas não contêm itens repetitivos: São valores ou grupos de valores que se repetem para determinado elemento da relação. São também chamadas colunas multivaloradas;
- Os Atributos são atômicos: Denominamos atômicos itens de dados indivisíveis. Conjuntos de itens atômicos podem formar um item composto;
- Os atributos não contêm valores nulos: Refere-se a não permitir que existam colunas com valor nulo na tabela. Embora, atualmente, não há mais restrição quanto à existência de valores nulos em colunas de uma tabela.

Processo de Obtenção da Primeira Forma Normal:

- **Definir as Chaves Candidatas e escolher a Chave Primária:** A partir da escolha de um ou mais atributos de uma tabela, deve-se buscar diferenciação entre uma linha e outra.

- **Transformar atributos compostos em atômicos:** Atributos compostos, que claramente tenham elementos distintos em sua estrutura, deverão ser separados.
- **Eliminar grupos repetitivos:** Significa desmembrar as colunas com valores repetidos de forma a termos somente atributos atômicos monovalorados.

#### Etapas do processo de normalização da 1ª Forma Normal:

- **Acabar com os itens de repetição** → As repetições ou multivalorações podem ser resolvidas de 3 formas diferentes: 1) Desmembramento em novas tabelas, 2) Desmembramento em novas linhas ou 3) Desmembramento em novas colunas. Analisemos cada elemento individualmente (ELMARSI, NAVATHE, 2005, p.225).

**1) Desmembramento em novas tabelas:** Nessa solução, os itens repetitivos saíram da tabela original (PEDIDO) e darão origem em uma nova tabela (ITENS\_PEDIDO). A Chave Primária dessa nova tabela (ITENS\_PEDIDO) será formada pela Chave Primária da tabela origem (PEDIDO) mais a coluna ou a coluna representativa do grupo que saiu da tabela origem. Aplicando na tabela desnormalizada da figura 4.85, temos a seguinte solução:

PEDIDO				
<u>Num. Pedido</u>	Data Emissão	Nome Fornec.	CNPJ	Endereço
003	20-jan	Software	12345-78	Av. Lapa 777, Lapa RJ 20000
004	27-Jan	Brasoft	23456-89	R. Itu 49, Itu – SP 11000
005	27-Jan	Computer	34567-00	R. Vitoria 12, Vitoria ES – 30000
006	14-Mar	Brasoft	23456-89	R. Itu 49, Itu – SP 11000

**FIGURA 4.86** Estrutura com desmembramento de atributos/grupos multivalorados – novas tabelas

ITEM_PEDIDO				
<u>Num. Pedido</u>	<u>Cód. Produto</u>	Nome Produto	Qtde.	Preço Unit.
003	033A	DOS	04	130,00
003	002M	COREL	01	499,00
003	145J	ABC	13	256,00
004	002M	COREL	02	450,00
004	083P	ZAPT	10	85,00
005	033A	DOS	50	110,00
005	145J	ABC	50	110,00
006	029K	WIN	15	200,00
006	083P	ZAPT	10	87,00

**FIGURA 4.87** Estrutura com desmembramento de atributos/grupos multivalorados – novas tabelas

**2) Desmembramento em novas linhas:** Nessa solução, os itens repetitivos serão desmembrados dentro da mesma tabela porem em linhas, uma para cada repetição. Nesse caso, a chave primária da tabela deverá ser revista para comportar a redundância da parte comum a ser repetida para formar as novas linhas. Aplicando na tabela desnormalizada da figura 4.85, temos a seguinte solução:

PEDIDO							
<u>Num. Pedido</u>	Data Emissão	Nome Fornec.	CNPJ	endereço	<u>Cód. Produto</u>	Nome Produto	Qtde. Preço Unit.

003	20-Jan	Software	12345-78	Av. Lapa 777, Lapa RJ 20000	033A	DOS	04	130,00
003	20-jan	Software	12345-78	Av. Lapa 777, Lapa RJ 20000	002M	COREL	01	499,00
003	20-jan	Software	12345-78	Av. Lapa 777, Lapa RJ 20000	145J	ABC	13	256,00
004	27-Jan	Brasoft	23456-89	R. Itu 49, Itu – SP 11000	002M	COREL	02	450,00
004	27-Jan	Brasoft	23456-89	R. Itu 49, Itu – SP 11000	083P	ZAPT	10	85,00
005	27-Jan	Computer	34567-00	R. Vitoria 12 Vitoria ES – 30000	033A	DOS	50	110,00
005	27-Jan	Computer	34567-00	R. Vitoria 12 Vitoria ES – 30000	145J	ABC	50	110,00
006	14-Mar	Brasoft	23456-89	R. Itu 49, Itu – SP 11000	029K	WIN	15	200,00
006	14-Mar	Brasoft	23456-89	R. Itu 49, Itu – SP 11000	083P	ZAPT	10	87,00

**FIGURA 4.88** Estrutura com desmembramento de atributos/grupos multivalorados – novas linhas

Observe que a Chave Primária dessa nova tabela será formada por um elemento da parte repetida (cabecalho do PEDIDO) e outra por um elemento da parte desmembrada (corpo do PEDIDO).

**3) Desmembramento em novas colunas:** Nessa solução, os itens repetitivos serão desmembrados dentro da mesma tabela em novas colunas. Vale lembrar que esta solução só deve ser adotada se o numero de ocorrências for conhecido e não muito grande. Aplicando na tabela desnormalizada da figura 4.85, temos a solução abaixo.

PEDIDO															
Num. Pedido				Cód. Produto1	Nome Produto1	Qtde1.	Preço Unit1.	Cód. Produto2	Nome Produto2	Qtde2.	Preço Unit2.	Cód. Produto3	Nome Produto3	Qtde3.	Preço Unit3.
003				033A	DOS	04	130,00	002M	COREL	01	499,00	033A	DOS	04	145J
004				002M	COREL	02	450,00	083P	ZAPT	10	85,00				
005				033A	DOS	50	110,00	145J	ABC	50	110,00				
006				029K	WIN	15	200,00	083P	ZAPT	10	87,00				

**FIGURA 4.89** Estrutura com desmembramento de atributos/grupos multivalorados – novas colunas

Observe que nessa solução, devemos considerar que em algumas linhas encontraremos colunas nulas, pois o numero de ocorrências é menor do que o previsto. Além disso, nessa solução, como pré-definimos a quantidade de colunas que serão construídas para abrigar o conteúdo, não temos a flexibilidade de armazenarmos um determinado pedido que tenha mais produtos do que o previsto no modelo. Para que não paire dúvidas, suprimimos do modelo os atributos “data emissão, nome fornec, CGC e endereço” simplesmente por questão de espaço. Os mesmos estão simbolicamente representados pelas 4 colunas em branco logo após a coluna “num.pedido”.



Em modelos de dados que possuam mais que um atributo ou grupo multivalorado, a melhor solução é desmembrá-los em novas tabelas (alternativa 1) descrita acima (ELMASRI, NAVATHE, 2005, p.227).

- **Transformar os Atributos Compostos em Atributos Atômicos.** Analisando a tabela, vamos considerar que Data de Emissão e CGC serão mantidos como Atômicos, e o Endereço, será considerado com composto e nesse caso, devemos decompô-lo conforme abaixo. Além disso, utilizamos como tabela fonte para esta decomposição a tabela da figura 4.86 oriunda da primeira solução para atributos compostos vistos anteriormente. A tabela da figura 4.87 não foi alterada pois não possui atributos compostos, mantendo-se inalterada conforme demonstra a figura 4.91.

PEDIDO								
Num. Pedido	Data Emissão	Nome Fornec.	CNPJ	Logradouro	Numero	Cidade	Uf	cep
003	20-jan	Software	12345-78	Av. Lapa	777	Lapa	RJ	20000
004	27-Jan	Brasoft	23456-89	R. Itu	49	Itu	SP	11000
005	27-Jan	Computer	34567-00	R. Vitoria	12	Vitoria	ES	30000
006	14-Mar	Brasoft	23456-89	R. Itu	49	Itu	SP	11000

**FIGURA 4.90** Estrutura com desmembramento de atributos compostos

ITENS_PEDIDO				
Num. Pedido	Cód. Produto	Nome Produto	Qtde.	Preço Unit.
003	033A	DOS	04	130,00
003	002M	COREL	01	499,00
003	145J	ABC	13	256,00
004	002M	COREL	02	450,00
004	083P	ZAPT	10	85,00
005	033A	DOS	50	110,00
005	145J	ABC	50	110,00
006	029K	WIN	15	200,00
006	083P	ZAPT	10	87,00

**FIGURA 4.91** Estrutura com desmembramento de atributos compostos – não sofreu alteração

- Definir uma **Chave** para que tenhamos unicidade nas linhas da tabela. Em algumas situações, pode ocorrer da perda da integridade da chave primária, especialmente quando se adota reduzir os atributos multivalorados desmembrando-os como linhas na mesma tabela (conforme visto anteriormente). Por isso, é necessário rever as colunas da tabela para se determinar a Chave Primária da mesma. No caso, somente na solução proposta na figura 4.88 seria necessário rever a Chave acrescentando coluna “cod.produto” conforme demonstrado abaixo.

PEDIDO												
Num. Pedido	Cód. Produto	Data Emissão	Nome Fornec.	CNPJ	Logradouro	Num	Cidade	Uf	cep	Nome Produto	Qtde.	Preço Unit.
003	033A	20-Jan	Software	12345-78	R. Lapa	777	Lapa	RJ	20000	DOS	04	130,00
003	002M	20-jan	Software	12345-78	R. Lapa	777	Lapa	RJ	20000	COREL	01	499,00
003	145J	20-jan	Software	12345-78	R. Lapa	777	Lapa	RJ	20000	ABC	13	256,00
004	002M	27-Jan	Brasoft	23456-89	R. Itu	49	Itu	SP	11000	COREL	02	450,00
004	083P	27-Jan	Brasoft	23456-89	R. Itu	49	Itu	SP	11000	ZAPT	10	85,00
005	033A	27-Jan	Computer	34567-00	R. Vitoria	122	Vitoria	ES	3000	DOS	50	110,00
005	145J	27-Jan	Computer	34567-00	R. Vitoria	122	Vitoria	ES	3000	ABC	50	110,00
006	029K	14-Mar	Brasoft	23456-89	R. Itu	49	Itu	SP	11000	WIN	15	200,00
006	083P	14-Mar	Brasoft	23456-89	R. Itu	49	Itu	SP	11000	ZAPT	10	87,00

**FIGURA 4.92** Redefinição de chave para alternativa de desmembramento em linhas para atributos/grupos multivalorados.

Uma vez realizada a Normalização da 1ª Forma Normal, chegamos a uma situação de modelo no qual os problemas de atributos compostos e multivalorados já não existem mais. No entanto, aquelas questões propostas anteriormente continuam em aberto. São elas:

- Somente pode-se incluir um novo fornecedor, se for incluído um novo pedido para este fornecedor.
- Se for eliminado o pedido número 005, perde-se junto, todas as informações do fornecedor “Computer”.
- Se for alterada a data do pedido 003, tem-se que alterar várias linhas da tabela (esta situação somente se manifesta quando se opta por desmembrar os atributos multivalorados em novas linhas).

Para tratarmos desses problemas, devemos avançar no processo de normalização. Porém, precisamos, antes disso, introduzir o conceito que irá sustentar estas outras Formas Normais, que é o de **Dependência Funcional** e sobre o qual a maior parte da teoria de normalização foi baseada (MACHADO, ABREU, 1996, p.172).

**Dependência Funcional** é a situação na qual, um atributo x de uma entidade X é funcionalmente dependente de outro atributo y também presente na mesma entidade X, ou seja, se a cada valor de x corresponde exatamente um valor de y. Por exemplo, suponha uma entidade CLIENTES que possuam diversos atributos e

entre eles o atributo ESTADO e DDD, representando a Unidade Federativa e o DDD onde ele mora. Nesse caso, sempre que o atributo ESTADO tiver o valor 'ES', o DDD terá o valor "27". Diz-se então que o atributo DDD é funcionalmente dependente do atributo ESTADO (MAYER, 2001, p.30). Em outras palavras, dizer que um atributo y é dependente de um atributo x significa dizer que basta conhecer o valor x para se chegar a um único valor do atributo y. O atributo x determina o valor do atributo y (MACHADO, 2004, p.187).

A Dependência Funcional pode ser **total** ou **transitiva**. Diz-se que há uma **dependência funcional total** quando existe uma Chave Primária composta e os atributos presentes na relação (e que não são chave) dependem da chave como um todo e não somente de parte da chave (MACHADO, ABREU, 1996, p172-173). A figura 4.87 referente a normalização de 1ª Forma Normal demonstra uma situação de dependência total de chave primária. Os atributos "qtde" e "preço unit" dependem totalmente da Chave Primária composta "num pedido e cod produto" para que se conheçam os seus valores. Podemos resumir então que a **dependência funcional total ou completa** só ocorre em tabelas que possuam Chave Primária composta.

Por outro lado, a **dependência funcional transitiva** ocorre quando a interdependência entre os atributos se dá sem que algum dos atributos seja Chave Primária da tabela. Observe a tabela da figura 4.86. Analisando os atributos "CNPJ" e "Nome Fornec" percebe-se que se conhecermos o valor do CNPJ conheceremos o valor do "Nome Fornec", no entanto, o CNPJ não é Chave Primária da tabela. Diz-se então que o atributo "Nome Fornec" depende transitivamente do atributo CNPJ. Este, o CNPJ, depende por sua vez da Chave Primária da tabela.

A determinação das dependências funcionais requer a interpretação semântica dos valores armazenados nos elementos da relação, ou seja, depende de uma análise humana de tais valores o que torna difícil a mecanização desse processo (MAYER, 2001, p.31).

**Segunda Forma Normal (2FN):** A Segunda Forma Normal é baseada no conceito de **Dependência Funcional Total** visto anteriormente. Uma tabela está na segunda forma normal, se (COUGO, 1997, p.200-202):

- Ela está na Primeira Forma Normal: Para se aplicar a 2FN, deve-se garantir que a tabela está na 1FN.
- Cada uma das colunas não pertencentes à Chave Primária, não for dependente parcialmente dessa chave: Havendo Chave Primária composta, ou seja, formada por mais de uma coluna, todas as demais colunas da tabela devem ser dependentes da chave como um todo e não só de parte da chave. Se uma tabela tiver somente uma coluna como Chave Primária, ela estará automaticamente, na 2FN .
- Na tabela da figura 4.87, tem-se a situação na qual as colunas "Cod. Produto" e "Nome Produto" possuem uma interdependência entre elas sem com isso depender obrigatoriamente da Chave Primária completa da tabela a qual estão associadas. Note que para conhecermos o "nome do produto" basta que se conheça o "Cód Produto". No entanto, a Chave Primária é formada por "Cod. Pedido + Cod. Produto".

#### Processo de Obtenção da Segunda Forma Normal:

- Identificar as Colunas que não participam da Chave Primária da Tabela.
- Para cada uma das colunas identificadas, analisar se o seu valor é determinado por parte ou pela totalidade da Chave Primária.
- Para as colunas que forem identificadas como sendo dependentes parcialmente da chave:
  - Criar novas tabelas (tantas quantas forem as dependências parciais) para abrigar as colunas que são dependentes parcialmente da Chave Primária;
  - Excluir essas colunas da tabela original;
  - A Chave Primária dessas novas tabelas será a parte da Chave Primária da tabela original que determinava o valor das colunas dependentes parcialmente.

Tomemos como exemplo a tabela da figura 4.87. Ao aplicarmos a regra definida acima, temos:

- Identificar as colunas não Chave: nome produto, qtde. e preço unit.



- Identificar as dependências parciais: Para cada uma das colunas descritas anteriormente, analisar as dependências:

COLUNA	DEPENDÊNCIA	CHAVE
NOME PRODUTO	PARCIAL	COD PRODUTO
QTDE	TOTAL	NUM PEDIDO + COD PRODUTO
PRECO UNIT	TOTAL	NUM PEDIDO + COD PRODUTO

**FIGURA 4.93** Análise de dependência parcial de chave composta

- Criar novas tabelas com as colunas dependentes parcialmente das chaves e excluir essas colunas da tabela original. O resultado é demonstrado na figura 4.94 abaixo:

ITEM_PEDIDO			
<u>Num. Pedido</u>	<u>Cód. Produto</u>	Qtde.	Preco Unit.
003	033A	04	130,00
003	002M	01	499,00
003	145J	13	256,00
004	002M	02	450,00
004	083P	10	85,00
005	033A	50	110,00
005	145J	50	110,00
006	029K	15	200,00
006	083P	10	87,00

**FIGURA 4.94** Tabela PEDIDO, renomeada para ITEM\_PEDIDO contendo somente as colunas com dependência total da Chave Primária composta.

PRODUTO	
<u>Cód. Produto</u>	Nome Produto
033A	DOS
002M	COREL
145J	ABC
083P	ZAPT
029K	WIN

**FIGURA 4.95** Tabela PRODUTO, contendo somente as colunas com dependência parcial da Chave Primária composta, no caso a coluna "cod. produto".



Se a Chave Primária tiver um único atributo, não precisa ser analisada para a 2ª Forma Normal (ELMASRI, NAVATHE, 2005, p.227).

Ao final da análise de 2ªFN a tabela inicial deu origem a uma nova tabela (PRODUTO, figura 4.95) para abrigar os atributos com dependência parcial.

**Terceira Forma Normal (3FN):** A Segunda Forma Normal é baseada no conceito de Dependência Funcional Transitiva visto anteriormente. Uma tabela está na terceira forma normal, se (COUGO, 1997, p.209-211)

- Está na segunda forma normal

- Nenhuma coluna não pertencente à chave, fica determinada transitivamente por esta. Por dependência transitiva, pode-se entender a situação onde um atributo depende de outro e este segundo depende de um terceiro. Pode-se também dizer que possui dependência transitiva, um atributo que depende de um outro atributo chave presente na Tabela, mas esta chave, não é a Chave Primária.
- A dependência transitiva de uma chave só será possível se a tabela tiver ao menos duas colunas não pertencentes à chave. Caso tenhamos somente uma coluna externa à chave, essa tabela já estará automaticamente na 3FN.

#### Processo de Obtenção da Terceira Forma Normal:

- Identificar as Colunas que não participam da Chave Primária da Tabela.
- Para cada uma das colunas identificadas, analisar se o seu valor é determinado por alguma outra coluna não pertencente à Chave.
- Para colunas dependentes transitivamente da chave, deve-se:
  - Criar novas tabelas para estas colunas;
  - Excluir da tabela origem as colunas dependentes transitivamente das chaves mantendo e que foram deslocadas para novas tabelas, porém, manter a coluna determinante da transitividade na tabela origem;
  - A chave primária dessas novas será(ão) a(s) coluna(s) que determinou(aram) o valor da coluna analisada.

Para analisar tais dependências conforme as regras acima tomaremos as tabelas das figuras 4.90, 4.94 e 4.95 (PEDIDO, ITEM\_PEDIDO e PRODUTO respectivamente). Vale destacar que estas 3 tabelas já se encontram na 2ª Forma Normal. A tabela da figura 4.90 não precisou ser analisada na 2ª FN, pois possui Chave Primária formada por um único atributo, ao passo que as tabelas das figuras 4.94 e 4.95 surgiram do desmembramento de 2ªFN realizado sobre a tabela da figura 4.91.

Aqui também encontramos nova restrição. Tabelas que possuam um único atributo além da Chave Primária estão automaticamente na 3ª Forma Normal. Por isso, não analisaremos a tabela da figura 4.95. Nossa análise então se centrará nas tabelas da figura 4.90 e 4.94. A figura 4.96 abaixo demonstra as tabelas que iremos analisar, cada atributo não chave dessas tabelas e se existe ou não dependência transitiva. Caso exista, a coluna que determina a dependência será demonstrada em DEPENDÊNCIAS TRANSITIVAS.

TABELA	COLUNAS	DEPENDÊNCIAS TRANSITIVAS
PEDIDO	data emissao	Não possui
PEDIDO	nome fornec	CNPJ
PEDIDO	cnpj	nome fornec
PEDIDO	Logradouro	CNPJ
PEDIDO	Numero	CNPJ
PEDIDO	Cidade	CNPJ
PEDIDO	Uf	CNPJ
PEDIDO	CEP	CNPJ
ITEM_PEDIDO	Qtde	Não possui
ITEM_PEDIDO	Preço	Não possui

**FIGURA 4.96** Análise de dependência transitiva

Analisar as Dependências Transitivas, individualmente:

- A coluna “data emissão” não é determinada, conhecida a partir de nenhuma outra coluna presente na tabela, ou seja, ela **depende** do conhecimento da Chave Primária para conhecermos o seu valor.



- A coluna “nome fornec” poderá ser conhecida a partir do conhecimento da coluna CNPJ. Dessa forma, ela **não depende** do conhecimento somente da Chave Primária para conhecermos o seu valor.
- A coluna “cnpj” também poderá ser conhecida a partir do conhecimento da coluna “nome fornec”. Dessa forma, ela **não depende** do conhecimento somente da Chave Primária para conhecermos o seu valor.
- A coluna “logradouro” também poderá ser conhecida a partir do conhecimento da coluna “cnpj”. Dessa forma, ela **não depende** do conhecimento somente da Chave Primária para conhecermos o seu valor.
- A coluna “numero” também poderá ser conhecida a partir do conhecimento da coluna “cnpj”. Dessa forma, ela **não depende** do conhecimento somente da Chave Primária para conhecermos o seu valor.
- A coluna “cidade” também poderá ser conhecida a partir do conhecimento da coluna “cnpj”. Dessa forma, ela **não depende** do conhecimento somente da Chave Primária para conhecermos o seu valor.
- A coluna “uf” também poderá ser conhecida a partir do conhecimento da coluna “cnpj”. Dessa forma, ela **não depende** do conhecimento somente da Chave Primária para conhecermos o seu valor.
- A coluna “cep” também poderá ser conhecida a partir do conhecimento da coluna “cnpj”. Dessa forma, ela **não depende** do conhecimento somente da Chave Primária para conhecermos o seu valor.
- A coluna “qtde” não é determinada, conhecida a partir de nenhuma outra coluna presente na tabela, ou seja, ela **depende** do conhecimento da Chave Primária para conhecermos o seu valor.
- A coluna “preco unit” também não é determinada, conhecida a partir de nenhuma outra coluna presente na tabela, ou seja, ela **depende** do conhecimento da Chave Primária para conhecermos o seu valor.

Após tal análise, caso existam colunas com **dependência transitiva**, deve-se criar novas tabelas para abrigá-las:

PEDIDO		
<u>Num. Pedido</u>	Data Emissão	CNPJ
003	20-Jan	12345-78
004	27-Jan	23456-89
005	27-Jan	34567-00
006	14-Mar	23456-89

**FIGURA 4.97** Tabela PEDIDO, contendo somente as colunas com dependência da Chave Primária.

FORNECEDOR						
<u>CNPJ</u>	Nome Fornec.	Logradouro	Numero	Cidade	Uf	cep
12345-78	Software	Av. Lapa	777	Lapa	RJ	20000
23456-89	Brasoft	R. Itu	49	Itu	SP	11000
34567-00	Computer	R. Vitoria	12	Vitoria	ES	30000

**FIGURA 4.98** Tabela FORNECEDOR, contendo as colunas que foram eliminadas da tabela PEDIDO devido a dependência transitiva

ITEM PEDIDO			
<u>Num. Pedido</u>	<u>Cód. Produto</u>	Qtde.	Preco Unit.
003	033A	04	130,00
003	002M	01	499,00
003	145J	13	256,00
004	002M	02	450,00
004	083P	10	85,00
005	033A	50	110,00
005	145J	50	110,00
006	029K	15	200,00

006	083P	10	87,00
-----	------	----	-------

**FIGURA 4.99** Tabela PEDIDO que não sofreu alteração pois nenhuma de suas colunas possuem dependência transitiva.

PRODUTO	
<u>Cód. Produto</u>	Nome Produto
033A	DOS
002M	COREL
145J	ABC
083P	ZAPT
029K	WIN

**FIGURA 4.100** Tabela PRODUTO (oriunda da tabela da figura 4.95) que não sofreu alteração, pois já estava na 3ª Forma Normal.

Ao analisarmos as tabelas descritas na 3ª Forma Normal, é possível validar se as anomalias de Inserção, Alteração e Exclusão definidas na seção **4.5.1 – Anomalias de Modificação**.

- Se desejarmos eliminar o pedido 5, não mais perderemos os dados do Fornecedor “computer”.
- Podemos incluir um novo produto sem ter que vinculá-lo a um pedido.
- Qualquer alteração nos dados de um determinado Fornecedor agira somente no registro a ele pertencente.

Dessa forma, podemos concluir que ao alcançarmos as 3 primeiras Formas Normais obtermos um modelo livre dessas anomalias.

O Quadro abaixo faz um resumo das três Formas Normais estudadas acima com as ações previstas para cada uma.

TESTE PARA A FORMA NORMAL		(AÇÕES) NORMALIZAÇÃO
1ª Forma Normal	A relação não deve conter atributos não atômicos ou relações aninhadas. Só deve conter atributos atômicos	Formar uma nova relação para cada atributo não atômico ou para cada relação aninhada.
2ª Forma Normal	Para as relações que possuam chave primária composta (com vários atributos), nenhum atributo externo deva ser funcionalmente dependente de parte da chave.	Decompor e montar uma nova relação para cada chave parcial com seus atributos dependentes. Manter, na relação original, a chave composta com todos os atributos que dependam totalmente dela.
3ª Forma Normal	As relações não devem ter atributos que dependam de qualquer outro atributo presente na mesma relação, sendo que esse atributo determinante não seja chave primária. Não deve haver dependência transitiva entre um atributo não-chave e a chave primária.	Decompor e montar uma nova relação para todos os atributos que tenham dependência transitiva. A chave da nova relação será o atributo que determinava a transitividade na relação origem. Este atributo permanecerá na relação origem para criar o vínculo com a nova relação.

**FIGURA 4.101** Resumo para Formas Normais e Ações Previstas (baseada em ELMARSI, NAVATHE, 2005, p.229)



**BIBLIOGRAFIA**

- [1] COUGO, P. **Modelagem Conceitual e Projeto de Banco de Dados**. Rio de Janeiro: Campus, 1997.
- [2] DATE, C.J. **Introdução a Sistemas de Banco de Dados**. 8ed. Rio de Janeiro: Elsevier, 2003.
- [3] ELMASRI, R., NAVATHE, S. B. **Sistemas de Banco de Dados**. 4ed. São Paulo: Adisson Wesley, 2005.
- [4] HEUSER, C. A. **Projeto de Banco de Dados**. Porto Alegre: Instituto de Informática da UFRGS: Sagra Luzzatto, 2001.
- [5] KORTH, H. F. & SILBERSCHATZ, A. **Sistema de Bancos de Dados**. Rio de Janeiro: Elsevier, 2006.
- [6] KROENKE, D.M. **Banco de Dados**. Fundamentos, projeto e implementação. Rio de Janeiro: LTC, 1999.
- [7] HARRINGTON, J. L. **Projetos de Bancos de Dados Relacionais**. Teoria e Prática. 2ed. Rio de Janeiro: Campus, 2002.
- [8] MACHADO, F. N. R., ABREU, M. **Projetos de Bancos de Dados**. Uma visão prática. São Paulo: Érica, 1996.
- [9] MAYER, R. C. **Otimizando a performance de Bancos de Dados Relacionais**. Rio de Janeiro: Axel Books, 2001.
- [10] POMPILO, S. **Análise Essencial – guia prático de Análise de Sistemas**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2002.
- [11] PRESSMAN, R.S. **Engenharia de Software**. 5ªed. Rio de Janeiro: McGraw-Hill, 2002.
- [12] SETZER, W. W. **Banco de Dados**. 1ed. São Paulo: Edgard Blucher, 2005.
- [13] SHLAER, S., MELLOR, S. J. **Análise de Sistemas Orientada para Objetos**. São Paulo: McGraw-Hill : Newstec, 1990.
- [14] SQL Magazine. Rio de Janeiro: DEVMEDIA Group, Ed. 16. Ano 2. 2004.
- [15] TEOREY, T., LIGHTSTONE, S., NADEAU, T. **Projeto e Modelagem de Banco de Dados**. Rio de Janeiro: Elsevier, 2007.
- [16] TOGNERI, D. F. **Notas de Aula**. Disciplina Analise e Projeto de Sistemas. 2005.
- [17] YOURDON, E. **Análise Estruturada Moderna**. Tradução da terceira edição americana. Rio de Janeiro: Campus, 1992.